A Reinforcement Learning Architecture that Transfers Knowledge between Skills when Solving Multiple Tasks

Paolo Tommasino, Daniele Caligiore, Marco Mirolli, and Gianluca Baldassarre

***Abstract*—When humans learn several skills to solve multiple tasks, they exhibit an extraordinary capacity to transfer knowledge between them. We present here the last enhanced version of a bio-inspired reinforcement-learning modular architecture able to perform skill-to-skill knowledge transfer and called 'TERL Transfer Expert Reinforcement Learning model'. TERL architecture is based on a reinforcement-learning actor-critic model where both the actor and the critic have a hierarchical structure, inspired by the mixture-of-experts model, formed by a gating network that selects experts specialising in learning the policies or value functions of different tasks. A key feature of TERL is the capacity of its gating networks to accumulate, in parallel, evidence on the capacity of experts to solve the new tasks so as to increase the responsibility for action of the best ones. A second key feature is the use of two different responsibility signals for the experts' functioning and learning: this allows the training of multiple experts for each task so that some of them can be later recruited to solve new tasks and avoid catastrophic interference. The utility of TERL mechanisms is shown with tests involving two simulated dynamic robot arms engaged in solving reaching tasks, in particular a planar 2-degrees-of-freedom arm, and a 3D 4-degrees-of-freedom arm.**

***Index Terms*—Cumulative learning, transfer reinforcement learning, catastrophic interference, bio-inspired modular neural architectures, mixture-of-expert networks, functioning and learning responsibility signals, autonomous robotics, reaching tasks**

I. Introduction

Humans, in particular children, learn multiple skills in a cumulative fashion, from simple to progressively more complex ones [1]. A striking feature of this cumulative learning process is its increasing speed. Thus, for example, children initially learn basic reaching and grasping skills in months [2], [3], but then later rapidly develop a repertoire of variants of the basic patterns in increasingly shorter times [4], [5].

This work is in part motivated by the aim to understand the mechanisms through which children learn skills in an increasingly fast fashion. In this respect, our leading hypothesis is that the increasing learning speed observed in children relies on the transfer of knowledge from already learned skills to new skills to be acquired. The increasing learning rates might thus result from the fact that, thanks to knowledge transfer, learning processes can focus on acquiring only the aspects of the new skills that are novel with respect to the already acquired ones.

In particular, this work describe a bio-inspired modelling architecture, developed within a reinforcement learning (RL) framework, that can be used to study skill-to-skill knowledge transfer. The system is called *Transfer Expert Reinforcement Learning* model (TERL). TERL has two capabilities that we deem essential for obtaining an increasingly fast learning of multiple skills (note that here we will use the term 'skill' as a synonym of the RL 'policy'): (a) The capacity to transfer knowledge from already acquired skills to new skills to be learned [6]; (b) The capacity to store knowledge of the newly acquired skills so that it does not damage the knowledge on the already acquired ones; in other words, it can avoid 'catastrophic interference' (or 'catastrophic forgetting' [7], [8]). The architecture, as further specified in Section II-A, is 'bio-inspired' in that it is based on principles suggested by behavioural and brain mechanisms operating in organisms, and at the same time meets some constraints also faced by organisms when acquiring multiple skills.

The challenge faced here contributes to the developmental robotics overall objective of endowing robots with 'developmental programs' supporting a prolonged autonomous development [9], [10], [11], [12], [13], [14]. In particular, within this overall objective we face here the sub-problem of how robots could acquire multiple skills in increasingly efficient ways.

The model we illustrate and analyse here is related to the machine learning literature on *transfer reinforcement learning (TRL)* investigating how transferring knowledge between different domains and tasks (see [6] and [15] for two reviews). The architecture described here focuses on a specific important problem of TRL concerning the development of algorithms capable of automatically selecting already solved 'source tasks' and the skills acquired to solve them, and to use such skills as starting points to best solve new 'target tasks'. We will refer to this problem as the 'source-task selection problem'. Here we sought solutions to the source-task selection problem for an important class of tasks: the tasks share the same primitive actions and the same environment transition function, but they involve different reward functions depending on the tasks [6], [15]. This class of tasks is biologically relevant as it reflects a common situation faced by real animals where they have to satisfy different needs (e.g., hunger, thirst, etc.) and needs generate different tasks in the same environment and these are

P. Tommasiono is with the Nanyang Technological University, Robotics Research Centre, Singapore. E-mail: paolo001@e.ntu.edu.sgz

D. Caligiore, M. Mirolli, and G. Baldassarre are with the Laboratory of Computational Embodied Neuroscience (LOCEN), Istituto di Scienze e Tecnologie della Cognizione (ISTC), Consiglio Nazionale delle Ricerce (CNR), Roma, Italy. E-mail: {daniele.caligiore,marco.mirolli,gianluca.baldassarre}@istc.cnr.it

Manuscript received ; revised .

Published as: Tommasino Paolo, Caligiore Daniele, Mirolli Marco, Baldassarre Gianluca (2019) A reinforcement learning architecture that transfers knowledge between skills when solving multiple tasks, IEEE Transactions on Cognitive and Developmental Systems, 11(2): 292-317. doi: 10.1109/TCDS.2016.2607018

solved using the same set of 'primitive actions' (i.e., muscle activations). Needs generate tasks as they represent reward functions that produce a reward signal when a need is present and is satisfied with the attainment of a suitable resource [16].

TERL architecture has been developed to its current form through a series of predecessor models. The idea of using the mixture-of-expert model to implement an actor-critic RL model was first proposed in [17] who used it to compose skills and to implement skill-based transfer between complex tasks (see Section IV). The idea of using the mixture-of-experts model principles and actor-critic models was also used in [18], [19] to implement automatic task decomposition and multi-task learning and was tested with a simple navigation task involving a 2D scenario and discrete actions. The architecture was successively developed to work with continuous actions to control a 2D dynamic simulated arm engaged in solving multiple reaching tasks [20] but its capacity to implement skill-to-skill transfer was not investigated. The latter system was further investigated by [21] and further developed by decoupling the responsibility signals of experts used for functioning and for learning: this decoupling greatly enhances the capacity of the system to face the skill-to-skill transfer RL problem (see Section IV). Such work was the first to use the 'TERL' acronym to indicate the system. Two previous works [21], [22] used TERL to show that the principles of the mixture-of-experts system applied to RL can be used to model and investigate the processes of *assimilation* and *accommodation* studied in developmental psychology [1].

The system described in the current article represents the culmination of these previous efforts into a new architecture that resembles the previous systems in many ways but also presents some innovations, is described in a novel and comprehensive way, and is tested in depth in relation to the skill-to-skill transfer problem. More in detail, the work illustrated here presents the following advancements with respect to the previous works: (a) a systematic justification of the model ingredients from a computational perspective, and their link to relevant features of brain (Section II-A); (b) a fully revised formal description of the model (Section II-B); (c) a refinement of the new mechanism for decoupling the experts' responsibility signals used for functioning and for learning, and other improvements (e.g., noise generation and a best-expert freezing mechanism; Section II-B): these mechanisms allow the system to train multiple experts (that we call 'backgroud copies') for each task, so some of them can be later recruited to solve new tasks while avoiding catastrophic interference; (d) a systematic study and quantification of the skill-to-skill transfer capabilities of TERL involving sequential trials (Section III-B) or interleaved trials (Section III-D) related to different tasks; (e) tests on the scalability of the system to a large number of sequential tasks (Section III-C); (f) tests with the 4DoF redundant robotic arm in addition to the planar arm (Section III-E); (g) a throughout discussion of other systems related to skill-to-skill transfer, from the option-framework systems to systems directly storing experience or systems belonging to the MOSAIC-model family, and a clarification of their differences with respect to TERL (Section IV). The article is closed with the summary of the main results presented here and the possible future developments of the system (Section V).

II. TERL: RATIONALE, ARCHITECTURE, FUNCTIONING, AND LEARNING

A. Principles used to build TERL

The principles followed to build TERL are inspired by the way in which animals' brain and behaviour might solve the source-task selection problem. At the same time, these principles allow TERL to face such a problem by fulfilling some constraints undergone by animals. This is intended to facilitate the future use of TERL as a model of multiple-skill learning in animals.

1) Continuous states and actions, and function approximation: TERL is a RL system using function approximators. Function approximation is needed for RL to work in realistic setups, e.g. with embodied models and robots involving continuous state and action spaces [23], [24], [25]. The system described here is suitable to tackle such continuous state and action spaces (but not discrete actions as it is based on mechanisms mixing different actions). In particular, TERL uses linear function approximators (here also called experts) to encode policies and value function estimations. Linear function approximators have the computational advantages of simplicity of implementation, learning speed, convergence properties, and stability [24], [26]. However, they cannot solve linearly-separable problems: this limitation is commonly solved, as here, by re-coding the input space with kernel functions [24], [27] and by leveraging modularity.

From a biological perspective, building systems capable of working with continuous state and action spaces captures the constraint faced by organisms that interact with the world through sensors and actuators involving continuous output and input signals. The use of linear function approximators increase the biological plausibility of models as they can use learning rules based on locally-available information [28].

2) Actor-critic architecture: TERL overall architecture is based on the *actor-critic* RL model [29], [24]. Actor-critic models use separate data structures for storing the action policy and the value function learned on the basis of the Temporal Difference (TD) algorithm [24]. The use of different data structures to represent policies and value functions was done as the latter ones often require different segmentations of the problem space. Moreover, since the system 'compiles' t the gathered knowledge related to policies and evaluations into neural network-data structures (respectively mapping states into actions or into state-values), it is less memory demanding and more computationally efficient than other transfer models directly storing experience in the form of 'state, action, state, reward' tuples (we will further explain this in Section IV that reviews some examples of the latter models). Actor-critic models, which explicitly represent policies, are also suitable to face continuous action problems where drawing the policy from value functions is costly or complex [26].

From a biological perspective, the actor-critic architecture is considered one of the best models of the brain mechanisms underlying trial-and-error in organisms, in particular of basal ganglia [30], [31], [32], [33]. Moreover, the TD learning rule at the core of the architecture learning processes accurately reproduces the dynamics of phasic dopamine, a neuromodulator playing a major role in trial-and-error learning of organisms [34], [35].

3) Mixture-of-experts to solve the source-task selection problem: The actor and the critic components of TERL are each based on the mixture-of-experts neural-network model [36], [37]. This model, proposed to solve supervised learning problems, was modified to work within a RL framework in the models preceding TERL (starting from [17] and [18]). The use of the modified principles of the mixture-of-expert architecture is at the basis of the capacity of TERL to solve the source-tasks selection problem. For this purpose, the actor and the critic are each formed by a hierarchy having two levels: low-level 'experts', learning the policies or value functions needed to solve the novel tasks, and a high-level 'gating network', learning to select the experts (the term 'hierarchy' is used here to refer to the feature of the architecture for which a high-level module controls the selection of lower-level modules). The two gating networks of the actor and the critic select experts encoding already acquired skills to solve a new task on the basis of the accumulation of evidence related to their actual capacity to solve such new tasks. This accumulation of evidence is done in parallel for all available experts, so it scales up well with the number of experts. We shall see that this process results in an effective mechanism capable of quickly identifying the best expert for a given task, so after an initial phase of exploration the system will assign all the responsibility for action to that expert.

4) No prior information about the similarity between source and target tasks: The scenario that was used here to test TERL capacity to solve the source-task selection problem has an important feature: the system is not given any prior information about the similarity between the source and the target tasks, so the only information it can use to solve the source-task selection problem is to sample the performance of the already acquired skills in the new target task. In particular, the model is tested here with robot arms that have to learn to solve different reaching tasks where the target object is located in different positions; these positions have different degrees of similarity between them, so they offer different opportunities for knowledge transfer: the robot is, however, informed only on the task identity (i.e., at different trials it is told 'this is task 1' or 'this is task 2', etc.), but not on the position of the target object, so the only way it has to select for re-use the already acquired skills is to try them out in the new target task. We adopted this demanding condition to be sure to find algorithms that are able to best use the knowledge on the performance of the already-acquired skills in the new task. Indeeed, in some domains this is the only available information. As we further discuss in Section IV, the mechanisms examined here to exploit this type of information can be integrated with other mechanisms capable of exploiting information about the similarity between tasks that is available in some domains.

Biologically, the condition where the agent has no prior knowledge on the similarity between the source and target tasks captures the situations in which animals have information about their internal needs but have no clue on the external world conditions where they have to satisfy them [38], [39]. In these cases, the animals know that they have to solve different tasks (as they have to satisfy different needs, e.g. extinguish hunger or thirst), but they do not know if the behaviours to learn to solve them have to be similar or different (for example, if they have to perform similar behaviours to 'collect' fruits with high content of nutrient or water; or if they have to perform very different behaviours to collect nutrient fruits or water from a lake). Instead, the condition in which the agent has information about the similarity between source and target tasks reflects situations where the tasks are defined in terms of 'goals' (i.e., desired states of the external world) rather than 'needs': in this case, the similarities between the goals of tasks can be heuristically used to infer that the behaviours to learn to solve them might be similar or different [40]. Note that the restrictive condition faced here, where the system has no information about the similarity between tasks, is also relevant for robots as it reflects situations where the robot tasks are defined in terms of very abstract goals or in terms of internal variables of the robot (e.g., 'satisfy the user's request to drink' or 'increase your energy level'), rather than specific goals (e.g., 'reach the glass located in position x_1, y_1 on the table' or 'reach the glass located in position x_2, y_2 on the table'): indeed, abstract goals and internal variables give little or no information about the similarity between the behaviours needed to solve the tasks, a situation captured by the restrictive condition used used here.

5) Redundant experts to face catastrophic interference: The system is based on multiple possibly redundant, low-level experts, which are individually very simple but collectively capable of solving multiple, complex tasks. Computationally, this solution has several advantages such as fault tolerance and the possibility of using simple linear experts [41], [42], [43]. A further important advantage of this solution, exploited here, is the possibility to reduce catastrophic interference. Here the problem of catastrophic interference is generated by the requirement that the system has to use the same experts during its whole learning life, so knowledge of new tasks can disrupt knowledge of already solved tasks. This requirement captures an important constraint undergone by brain, namely the fact that it cannot generate 'new experts' (new neural resources) to face new tasks (note that for simplicity here we do not consider the developmental processes involving the physical structuring of the brain in time guided by epigenetic programs).

TERL faces the catastrophic interference problem on the basis of a novel mechanism: the decoupling of the responsibility signals used to select the experts for functioning and the responsibility signals used to modulate their learning. This innovation, introduced by TERL models and departing from other models based on the mixture-of-experts principles, allows the regulation of the learning rate of experts as desired, in particular independently of the size of their responsibility for acting, or evaluating states, in the task (the decoupling is explained in detail in Section II-B). This decoupling allows the formation of copies of the best policy expert and value expert trained when solving a given task. These 'copy experts' can then be used to solve new tasks without disrupting the capacity of previously-trained experts to solve previous tasks.

In this respect, we mentioned above that the system is informed on the 'identity' of the task being faced, so the reader might wonder what are the advantages of duplication with respect to a straightforward solution creating one new expert for each new task. These advantages are several (see also [42]). First, the system could be used to model the brain as this has most neural resources since birth and so it must be endowed with mechanisms to use them when learning an increasing number of tasks. Second, having all modules since the beginning allows the system to use experts already trained with similar tasks to solve new tasks rather than starting from scratch: this is at the basis of the transfer learning capabilities of the system. Third, the presence of copy experts gives robustness to the system, e.g. in case of failure of some experts. Fourth, although not done here, using the same set of experts from the beginning might allow the system to progressively organise them in space (e.g. within a 2D grid) on the basis of their specialisation and temporal activation [43].

We also anticipate that since the task tackled by the system changes repeatedly the model has to suitably regulate the exploratory noise. Indeed, noise should progressively decrease with the learning of a task and increase again in correspondence to new tasks. Since this problem was not central for this work, we solved it with a simple approach that uses high levels of noise only in correspondence to tasks that the system finds difficult to solve and where a high degree of exploration is needed, and a low level of noise for tasks that are readily solved by the system.

From a biological perspective, although we do not have direct empirical evidence for neural duplication (i.e. the fact that a piece of knowledge encoded in a neural structure is reproduced in another neural structure; see [44]) we do have evidence for the partially modular organisation of the brain [41], in particular: (a) motor cortex is based on neural columns, where different assemblies of columns might participate to encode multiple repertoires of skills, from very similar to very different [45], [46]; (b) basal ganglia are organised in channels supporting the trial-and-error learning and selection of different actions and mental contents [47], [48].

B. TERL architecture and functioning

1) Architecture components: The architecture of TERL, shown in Figure 1, is based on two main components: the *actor*, responsible for learning the action policy, and the *critic*, responsible for learning to approximate the value function. Each of the two components is formed by a number of *expert networks*, learning the policy or the value function, and a *gating network*, learning to select the experts given the task (note that here for simplicity the number of experts of the actor and of the critic is the same but it could be different). The functioning and learning processes of the architecture are now explained in detail (the Appendix gives its parameters).

2) Inputs and outputs of the architecture components: The experts receive as input the current environment state encoded in an expanded space of features (here Gaussian basis functions with centres equally distributed over the input space, see below for details) and have no information about the goal or task. In particular, each environmental state is encoded with an I-dimensional vector of continuous variables $\mathbf{s} =< s_1, s_2, ..., s_i, ..., s_I >$. In the experiments reported here, $\mathbf{s}$ encoded the arm joint angles, so $I = 2$ in the 2D planar arm simulations, and $I = 4$ in the simulations involving the 3D humanoid robot arm. The state vector is expanded into a D-dimensional vector of continuous variables (features) $\mathbf{f} =< f_1, f_2, ..., f_d, ..., f_D >$ where $D >> I$. Features are computed through normalised Gaussian basis functions:

$$f_d = \frac{e^{-\frac{||\mathbf{s}-\mathbf{s}_d||}{\sigma_f^2}}}{\sum_{d=1}^{D} e^{-\frac{||\mathbf{s}-\mathbf{s}_d||}{\sigma_f^2}}} \tag{1}$$

where $\mathbf{s}_d$ is the preferred state vector of feature d and σ_f^2 is the width of the Gaussians (in degrees). The preferred vectors of the features lay on the vertices of a regular grid overlapped with the state space. Notice that the experts are not informed on the task to solve but only on the posture of the arm, so different tasks require different experts. This limited input eased the analysis of the transfer capabilities of the system, and reflects the organisation of the brain where primary motor cortex is mainly informed as to the limb posture but not as to the overall tasks the system is accomplishing [49], [50], [51]. This issue is further discussed in Section V.

The output of the actor experts is a J-dimensional vector $\mathbf{a} =< a_1, a_2, ..., a_j, ..., a_J >$ encoding the controlled continuous variables. Here, the vector encoded the angles of the arm joints, so $J = 2$ in the 2-D planar arm simulations, and $J = 4$ in those involving the 3D humanoid robot arm. As further explained below, these desired angles, varying at each time step, are used as the equilibrium points of proportional derivative controllers (PD) that produce the torques controlling the dynamic-arm joints.

The gating networks are informed only as to the identity of the task to pursue. The tasks used here are of the type 'reach and touch object A in space' through a dynamic arm. This tells the gating networks which task is being solved (e.g., 'Task A', 'Task B', etc.) but it does not furnish any information about the similarity between tasks. Formally, the current task identity is encoded as a K-dimensional vector $\mathbf{z} =< z_1, z_2, ..., z_k, ..., z_K >$. The $\mathbf{z}$ vectors, as many as the tasks, are orthogonal and are formed by binary elements all activated with 0 with the exception of one element, activated with 1, corresponding to the current task. Notice that the gating networks are only informed on the task but not on the posture of the arm. Again, this limited input eased the analysis of the transfer capabilities of the system and reflects the organisation of the brain where higher cortical areas are mainly informed on abstract goals and overall motivations [52], [53], [54].

The output of the gating networks are two E-dimensional vectors, where E represents the number of experts: $\mathbf{g}^{CG} =< g_1^{CG}, g_2^{CG}, ..., g_e^{CG}, ..., g_E^{CG} >$ for the critic gating network and $\mathbf{g}^{AG} =< g_1^{AG}, g_2^{AG}, ..., g_e^{AG}, ..., g_E^{AG} >$ for the actor gating network. These vectors encode the responsibility signals of the respective experts.

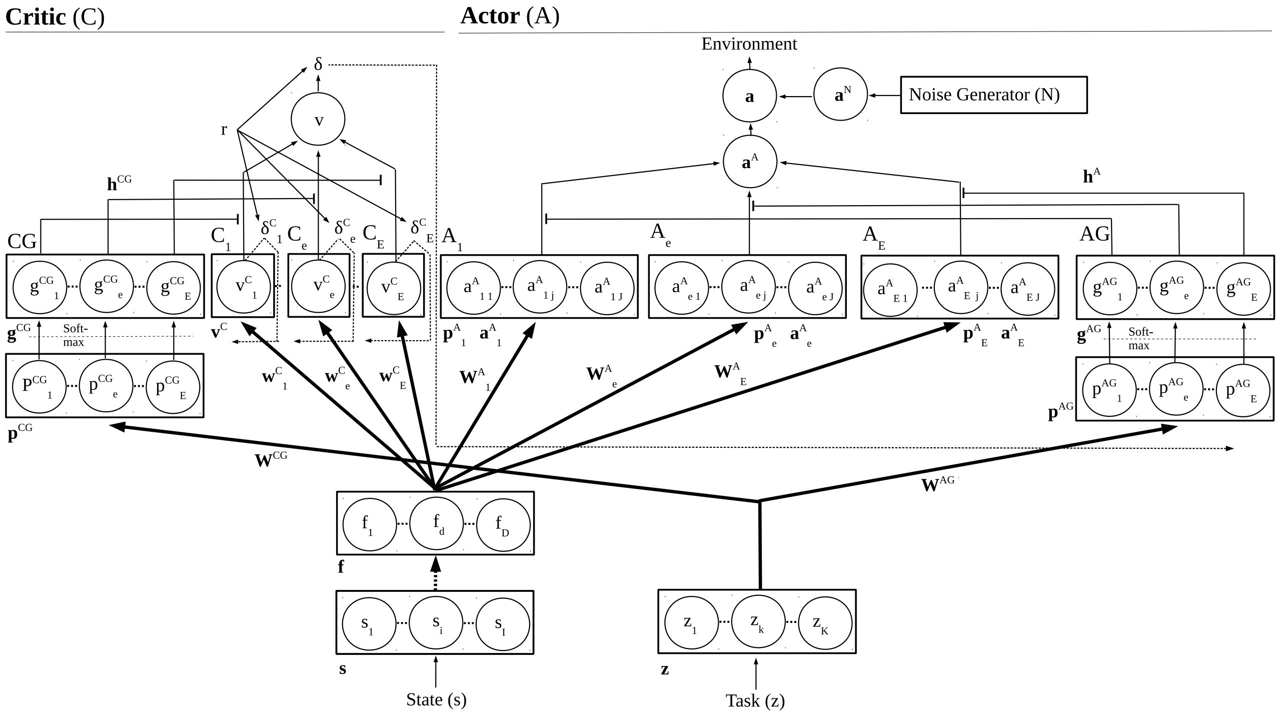


Fig. 1. The architecture of TERL, formed by a modular critic and a modular actor. The figure gives an overview of the elements composing the system that are explained in detail in the main text. The critic is formed by a gating network (CG) and a number of experts (C$_e$). Also the actor is formed by a gating network (AG) and a number of experts (A$_e$). Thin arrows: one-to-one or many-to-one hardwired connections set to one (for information relay); dotted bold arrow: all-to-all hardwired connections (for computing features); thin dashed arrows: learning signals that cause the update of all crossed connections (notice how the TD-errors of the critic experts are locally used to train themselves, whereas the critic selector does not need any TD-error to learn as it learns based on the mixture-of-experts supervised learning rule; the general TD-error is used to train the actor experts and the actor selector); bold arrows: trained all-to-all connections; flat-head arrows: one-to-one hardwired multiplicative gating connections set to one (for carrying the responsibility signals). Letters indicate the symbols used in the mathematical description of the system in the text. Bold letters represent vectors (for simplicity, the vectors of biases are omitted).

Finally, the system receives a reward $r = 1$ when the task is accomplished with success (here when the arm succeeds to touch the target object). Otherwise, the system receives a reward $r = 0$ at any other time step.

3) Actor gating network: functioning: The actor gating network, denoted with AG, receives as input the task identity **z** and returns as output the responsibility signals of the experts. These are encoded in the vector $\mathbf{g}^{AG}$ with elements g_e^{AG} corresponding to experts $e = 1, 2, ..., E$. The responsibility signal of an expert represents the 'prior probability' that such expert is the best one to solve the current task among the actor experts. The priors are used: (a) to set the contribution of each expert to produce actions; (b) to establish, after a suitable transformation (see below), the size of its learning update.

The activation potential of the output units of AG, denoted with the vector $\mathbf{p}^{AG}$ with elements p_e^{AG}, is computed as:

$$\mathbf{p}^{AG} = \mathbf{W}^{AG} \cdot \mathbf{z} \tag{2}$$

where $\mathbf{W}^{AG}$ is the matrix of the connection weights of the network. The gating networks of the model do not have a bias as they should not have a tendency to select specific default experts.

As in the mixture-of-experts model, the prior responsibilities, encoded in the vector $\mathbf{g}^{AG}$ with elements g_e^{AG}, are computed with the soft-max function:

$$g_e^{AG} = \frac{e^{p_e^{AG}/\kappa}}{\sum_{e=1}^{E} e^{p_e^{AG}/\kappa}} \tag{3}$$

where κ is the temperature regulating the slope of the soft-max. The soft-max guarantees that $\sum_{e=1}^{E} g_e^{AG} = 1$ so the responsibility signals can be interpreted as probabilities.

4) Actor experts: functioning: The actor experts, denoted with A_e, are a set of E neural networks each of which gets as input the state features **f** and encodes a possible action with logistic output units. The activation potential of the output units of expert A_e, denoted with the vector $\mathbf{p}_e^A$ with elements p_{ej}^A (e.g., corresponding to the robot joint angles), is computed as:

$$\mathbf{p}_e^A = \mathbf{W}_e^A \cdot \mathbf{f} + \mathbf{b}_e^A \tag{4}$$

where $\mathbf{W}_e^A$ is the connection weight matrix of the expert A_e and $\mathbf{b}_e^A$ the vector of biases. The activation of the output units of expert A_e, denoted with the vector $\mathbf{a}_e^A$ with elements a_{ej}^A, are computed with a logistic function:

$$\mathbf{a}_e^A = \frac{1}{1 + e^{-\mathbf{p}_e^A}} \tag{5}$$

As we shall see below, $\mathbf{a}_e^A$ is mixed with noise and so should be interpreted as the mean of a probability distribution of the system action.

5) Global actor action: The global action $\mathbf{a}^A$ of the actor (before the addition of noise) is formed by mixing the actions $\mathbf{a}^A_e$ of experts based on their priors:

$$\mathbf{a}^A = \sum_{e=1}^{E} \left[g^{AG}_e \cdot \mathbf{a}^A_e \right] \tag{6}$$

To foster exploration, the action of the system to be executed, denoted with $\mathbf{a}$, is obtained by mixing the global actor action $\mathbf{a}^A$ with a noisy action, denoted with $\mathbf{a}^N$, produced by the Noise Generator component explained below. Thus, at a specific time t:

$$\mathbf{a}_t = u_t \cdot \mathbf{a}^A_t + (1 - u_t) \cdot \mathbf{a}^N_t \tag{7}$$

where u_t is a variable regulating the exploitation-exploration level in different moments of the trial and explained below in the paragraph on the Noise Generator. This equation implies that with high values of u_t the performed action is strongly based on the actor action, whereas with low values it is strongly based on noise.

6) Critic gating network: functioning: The critic gating network, denoted with CG receives as input the task identity $\mathbf{z}$ and returns as output the responsibility signals of the experts of the critic. These responsibility signals are encoded in the vector $\mathbf{g}^{CG}$ with elements g^{CG}_e corresponding to the experts e. Similar to what happens for the actor, these responsibility signals represent the prior probability that each expert is the best one in estimating the value function, given the current task and the current policy, and are used for the critic functioning and, after a suitable transformation illustrated below, for its learning.

The activation potential of the output units of CG, encoded in the vector $\mathbf{p}^{CG}$ with elements p^{CG}_e, is computed as:

$$\mathbf{p}^{CG} = \mathbf{W}^{CG} \cdot \mathbf{z} \tag{8}$$

where $\mathbf{W}^{CG}$ is the matrix of connection weights of the network.

As with the actor, the prior responsibilities, encoded in the vector $\mathbf{g}^{CG}$ with elements g^{CG}_e, are computed with the soft-max function:

$$g^{CG}_e = \frac{e^{p^{CG}_e/\kappa}}{\sum_{e=1}^{E} e^{p^{CG}_e/\kappa}} \tag{9}$$

7) Critic experts: functioning: The critic experts, denoted with C_e, are a set of E neural networks each of which gets as input the state features $\mathbf{f}$ and returns the state value with a linear output unit. The activation v^C_e of the output unit of expert C_e is computed as:

$$v^C_e = \mathbf{w}^C_e \cdot \mathbf{f} + b^C_e \tag{10}$$

where $\mathbf{w}^C_e$ is a row vector encoding the connection weights of the expert and b^C_e its bias. Note how the actor experts use a logistic function to produce the output so this can be mapped to a limited range of the robot's joint angles (see Sections III-B, III-E). The critic experts instead use a linear output unit so they can produce state values outside the $(0, 1)$ range.

8) Critic: global value: The global value of the critic, denoted with v, is computed by mixing the values v^C_e of the experts based on their priors:

$$v = \sum_{e=1}^{E} \left[g^C_e \cdot v^C_e \right] \tag{11}$$

C. TERL learning

1) Critic: global TD-error: The global value v_t at time t is used to compute the global TD-error δ_t of the critic:

$$\delta_t = \begin{cases} 0 & \textit{if } t = 0 \\ (r_t + \gamma \cdot v_t) - v_{t-1} & \textit{if } 0 < t < T \\ r_t - v_{t-1} & \textit{if } t = T \end{cases} \tag{12}$$

where r_t is the reward at time t, and γ is a discount factor. In this formula, v_t is set to zero at the end of the trial ($t = T$) to take into account the episodic nature of the RL problems considered here. As we shall see below, δ_t, related to the performed actions, is used to train the actor experts and the actor gating network.

2) Critic experts: TD-error: The TD-error of each critic expert, denoted with $\delta^C_{e_t}$, is computed as follows on the basis of the expert value $v^C_{e_t}$ at time t:

$$\delta^C_{e_t} = \begin{cases} 0 & \text{if } t = 0 \\ \left(r_t + \gamma \cdot v^C_{e_t}\right) - v^C_{e_{t-1}} & \text{if } 0 < t < T \\ r_t - v^C_{e_{t-1}} & \text{if } t = T \end{cases} \tag{13}$$

This value represents the expert error in estimating the current state value and so it is computed on the basis of the expert current and past values and the experienced reward. For this reason, as we shall see below, it is used both to train the critic experts and to update the prior probability estimate g^{CG}_e that each critic expert is the best to evaluate the actor policy used to solve the current task.

3) Actor gating network: learning: We now show how the estimate of the goodness of experts in solving the new task is updated on the basis of a process of accumulation of information and the TD-error. At each time step, the each expert's responsibility signal is updated on the basis of the new evidence (likelihood) that the expert contributed to determine the executed action $\mathbf{a}$ in the current task (recall that such action is noisy due to the $\mathbf{a}^N$ component). Formally, the likelihood $l^{AG}_{e_t}$ is computed on the basis of a Gaussian function: this makes the likelihood of each expert inversely related to the distance between that expert action $\mathbf{a}^A_e$ and the executed noisy action $\mathbf{a}$:

$$l^{AG}_{e_t} = e^{-\frac{1}{2\cdot\sigma^2_{AG}} \left|\left| \mathbf{a}_{t-1} - \mathbf{a}^A_{e_{t-1}} \right|\right|^2} \tag{14}$$

where σ^2_{AG} is the Gaussian width parameter. Thus, the more similar the expert action to the executed action, the higher the likelihood that it had a large responsibility in generating it.

The likelihood is then used to compute the posterior probabilities with the Bayes rule:

$$h^{AG}_{e_t} = \frac{l^{AG}_{e_t} \cdot g^{AG}_{e_{t-1}}}{\sum_{e=1}^{E} [l^{AG}_{e_t} \cdot g^{AG}_{e_{t-1}}]} \tag{15}$$

Notice that this formula implies that $\sum_{e=1}^{E}\left[h_e^{AG}\right]=1$ and that $h_{e_t}^{AG} > g_{e_{t-1}}^{AG}$ for experts whose action $\mathbf{a}_e^A$ was more similar to the executed action **a** relative to the action of all other experts.

The update of the connection weights of the actor gating network AG, and hence the responsibility signals of the actor experts, is based on the difference between the posteriors and the priors but also on the overall TD-error δ:

$$w_{ek_t}^{AG} = w_{ek_{t-1}}^{AG} + \eta^{AG} \cdot \delta_t \cdot \left(h_{e_{t-1}}^{AG} - g_{e_{t-1}}^{AG}\right) \cdot z_{k_{t-1}} \qquad (16)$$

where η^{AG} is a learning rate. The formula, which is based on the mixture-of-expert formula changed to take into consideration the RL framework used here (in particular the TD-error), changes the connection weights so that the prior responsibility signal of experts, $g_{e_{t-1}}^{AG}$, gets closer to or away from the posterior responsibility signal, $h_{e_{t-1}}^{AG}$, depending on the sign and size of the TD-error, δ_t. In particular, the formula implies that the responsibility signals of experts whose actions are similar to the executed action (hence have a high $h_{e_{t-1}}^{AG}$) are increased, but only if the executed action produced a *positive TD-error*, i.e. if the executed action had positive effects. The responsibility signals of experts that produced an action dissimilar from the executed one are instead decreased. On the contrary, in the case of a *negative TD-error* the responsibility signals are updated in the *opposite direction*, yielding a lower responsibility for experts whose action was more similar to the executed one as the latter had negative effects. The responsibility signals of experts that produced an action dissimilar from the executed one are instead increased. Overall, the rule ensures that the responsibility signals are progressively increased for the experts that contribute to achieve the highest reward in the current task as the effect of an accumulation of evidence on the fact that they are better than the other experts in solving such task.

4) Actor experts: learning: The mixture-of-experts model uses the posterior probabilities to scale the learning rate of each expert so that the best experts are not only assigned the highest responsibility during functioning, but also an update proportional to such posteriors. One of the most important departures of the current system from this strategy is based on the decoupling of the responsibility signals used for functioning and those used for learning. Indeed, a number of pilot experiments using several variants of the mixture-of-experts strategy showed that the classical strategy leads not only to a desirable rapid increase of the responsibility of the best expert towards the maximum possible value of one, but also to a rapid decrease to zero of the learning rate of all other experts. In our case, the latter outcome is detrimental because when the system has to learn a new task similar to a previously acquired one, it recruits the expert trained to solve the latter, modifies it, and so loses the capacity to solve it (catastrophic forgetting). Instead, the use of fixed responsibilities for learning allows the training of multiple experts similar to the best one: the size of those responsibilities allows the regulation of the number and learning rate of the copy experts. These copy experts can be later recruited to solve similar tasks without destroying the capacity to solve the previously solved source tasks.

To implement this idea, we ranked the experts on the basis of the decreasing value of their priors g_e^{AG} and then assigned them fixed learning rates based on the ranks. In particular, here we used the experts' ranks, encoded with $g_e^{rAG} \in \{0,1,2,...,E-1\}$, to compute their learning responsibility signals g_e^{lAG} as follows:

$$g_e^{lAG} = \frac{\zeta^{-g_e^{rAG}}}{\sum_{e=1}^{E}\left[\zeta^{-g_e^{rAG}}\right]} \qquad (17)$$

where ζ was a coefficient. For example, setting $\zeta = 6$, as done here, implies that the learning responsibility signals g_e^{lAG} are, when ordered by rank, equal to the following values: $(0.834, 0.139, 0.023, 0.004, 0, 0, 0, 0, ...)$

Note that: (a) the algorithm will train one main expert and a given number of copies (here three) of it, whereas other experts will not be trained so avoiding disrupting their knowledge; (b) the experts that are trained are those that have the highest priors, i.e. those that are the best in solving the current task; (c) responsibility signals different from those used here can be used, including some with $\sum_{e=1}^{E}[g_e^{lAG}] \neq 1$, to obtain a desired number of background expert copies and set their learning rates; for example, one might establish a fixed learning rate for the first k experts. The higher the number of background experts copies are formed, the higher the redundancy of the system and all related advantages and disadvantages.

The connection weights of experts are then updated on the basis of g_e^{lAG} as follows:

$$\begin{aligned} m_{ejd_t}^{A} &= \left(a_{j_t} - a_{ej_t}^{A}\right) \cdot \left(a_{ej_t}^{A} \cdot \left(1 - a_{ej_t}^{A}\right)\right) \cdot f_{d_t} \\ w_{ejd_t}^{A} &= w_{ejd_{t-1}}^{A} + \eta^{A} \cdot \delta_t \cdot g_{e_{t-1}}^{lAG} \cdot m_{ejd_{t-1}}^{A} \end{aligned} \qquad (18)$$

where $\left(a_{ej_t}^{A} \cdot \left(1 - a_{ej_t}^{A}\right)\right)$ is the derivative of the logistic transfer function, and η^A is a learning rate. The rule is based on the standard gradient-descent formula of the mixture-of-experts model applied to neural logistic output units, but the size of the update is also weighted by the RL TD-error. The rule implies that: (a) the size of the update is higher for experts that produce an action more similar to the action actually executed (i.e. a higher $|a_{j_t} - a_{ej_t}^{A}|$); (b) $\mathbf{a}_e^A$ moves towards **a** when $0 < \delta_t$ as the noisy action **a** has been better than expected; (c) $\mathbf{a}_e$ moves away from **a** when $\delta_t < 0$ as the noisy action **a** has been worse than expected; (d) if δ_t is close to zero, for example when the system has converged to good solutions, the action is not updated because the noisy action **a** did not perform better or worse than expected; (e) if some of the expert copies developed for the task are later recruited to learn a similar new task, the capacity to solve the current task tend to not be impaired as the main expert used to solve it will not be recruited (see also the sub-section 'Additional mechanisms to preserve the best experts' below).

5) Critic gating network: learning: We now show how the responsibility signals of the experts of the critic are updated on the basis of mechanisms that accumulates evidence on the capacity of experts to face the given task similar to what is done in the mixture-of-experts model for supervised learning problems. This is possible as the problem of learning the value function can still be seen as a supervised learning problem with the difference being that the value to use as the desired output is not given and so, as common in RL [24], has to be formulated on the basis of the reward signal and the estimate of the next state value.

The new evidence (likelihood) that an expert is the best one to estimate the value function is computed as a Gaussian function of that expert TD-error, δ^C_e, as this represents its error in predicting the incoming rewards:

$$l^{CG}_{e_t} = e^{-\left(\frac{\left(\delta^C_{e_t}\right)^2}{2\cdot\sigma^2_{CG}}\right)} \tag{19}$$

where σ^2_{CG} is the Gaussian width.

Similarly to the actor gating network, the likelihood is used to compute the posterior probabilities with the Bayes rule:

$$h^{CG}_{e_t} = \frac{l^{CG}_{e_t}\cdot g^{CG}_{e_{t-1}}}{\sum_{e=1}^{E}\left[l^{CG}_{e_t}\cdot g^{CG}_{e_{t-1}}\right]} \tag{20}$$

The formula implies that $h^{CG}_{e_t}$ is higher than $g^{CG}_{e_{t-1}}$ for experts that have a low TD-error in comparison to the TD-error of the other experts. Also here $\sum_{e=1}^{E}\left[h^{CG}_{e}\right] = 1$.

The update of the connection weights of the critic gating network is then computed as in the mixture-of-expert model:

$$w^{CG}_{ek_t} = w^{CG}_{ek_{t-1}} + \eta^{CG}\cdot\left(h^{CG}_{e_t} - g^{CG}_{e_{t-1}}\right)\cdot z_{k_{t-1}} \tag{21}$$

where η^{CG} is a learning rate. The formula implies that the responsibility signals of experts with relatively smaller TD-errors are increased while those of experts with larger TD-errors are decreased, leading to larger responsibilities for experts that produce more accurate estimates of the state values for the current task. Overall, as for the experts: the rule ensures that the responsibility signals are progressively increased for the experts that contribute to achieving the highest reward in the current task as the effect of an accumulation of evidence that they are better than the other experts in solving such task.

6) Critic experts: learning: Due to the same reasons explained for the actor experts, we also implemented a decoupling between the functioning and the learning responsibility signals of the critic experts. The expert ranks g^{rCG}_e were computed as for the actor and used in the same way (Equation 17) to compute the learning responsibility signals g^{lCG}_e used to modulate the learning rate of the experts.

Each expert was then trained using the TD-learning RL formula and its own TD-error, δ^C_e, to enforce the self-consistency of the value estimates produced by the expert:

$$w^{C}_{ed_t} = w^{C}_{ed_{t-1}} + \eta^{C}\cdot\delta^{C}_{e_t}\cdot g^{lCG}_{e}\cdot f_{d_{t-1}} \tag{22}$$

7) Additional mechanisms to preserve the best experts: Note that in some conditions that are more challenging for catastrophic forgetting, additional mechanisms can be used to further reduce interference between tasks. For example, in the 'sequential conditions' tested here (Section III) we found it very useful to use a mechanism for which when the prior of an expert overcomes a certain (high) threshold, indicating that much evidence has been accumulated that such expert is the best for the current task, its learning rate is reduced to low values (here to zero for simplicity). Such low learning rate facilitates the preservation of the best experts found for previous tasks thus leading the system to update especially the 'background copies' to solve the new tasks. Notice that alternative mechanisms might be investigated in the future to preserve the best expert found for a given task.

8) Noise generator and executed noisy action: One of the major problems of RL is the regulation of exploratory noise, also known as the exploration-exploitation dilemma. The heuristic solution most commonly used in the literature is to progressively lower exploration noise with learning so as to augment the exploitation versus exploration while the system becomes progressively more skilled (i.e., progressively better at achieving reward) [24]. Here we adapted a very simple mechanism that is based on this idea, but note that the other mechanisms of TERL could work with any other method one might use for noise generation. The mechanism takes into consideration the fact that the new task to solve can be very similar, or even identical, to already solved tasks: in these cases, exploratory noise has to be very low since the initial trials of learning of the new task because the system already has some experts that can be readily used to solve it. For this purpose, we started each trial with low noise and then increased it during the trial. In this way, if the system is capable of solving the task quickly, it is not disturbed by noise; instead, if it takes a long time to solve it during the trial, then noise increases causing high exploration. Overall, the mechanism implies that with not-yet-learned tasks noise decreases progressively, on average, over the tasks (as in standard RL problems). Instead, with already-learned or easy tasks, for example with tasks very similar to already solved ones, noise is immediately low or decreases fast, on average over the trials, with learning.

To implement this idea in detail, we divided each trial into two phases. In a first 'exploitation phase' ($t = 0, 1, ..., M$) noise is very low. In a second 'exploration phase' ($t = M+1, M+2, ..., T'$, where T' is the trial timeout; note that T, the trial duration, is possibly shorter than T' in case of reward accomplishment) noise gets progressively higher. Notice that M should be set to a value equal or higher than the time assumed to be sufficient to solve the task (a similar parameter has to be set when using the progressive decrease of noise mechanism commonly employed in the RL literature and that could not be used here). In detail, the mechanism is implemented by setting the value of u_t regulating the mixture between the noisy action and the actor action (see Equation 7) as follows:

$$u_t = \begin{cases} \upsilon & \text{if } 0 < t \leq M \\ u_{t-1} - \beta u_{t-1} & \text{if } M < t \leq T' \end{cases}$$

where υ is the value of u in the exploitation phase and β is a time constant regulating the dynamics of u in the exploration phase. A small amount of noise is still present in the initial exploitation phase to have some refinement of the policy even during such phase, whereas β is set to a value that implies that noise rapidly increases during the second exploration phase.

As done in [55], using a noise filter is important to control dynamic robotic arms because their physical inertia tends to cancel out white noise and also avoids issuing jerky commands to the robot. For this reason, the noisy action $\mathbf{a}^N$ that is mixed with the actor action to produce the executed action (Equation 7) is produced by a Noise Generator component, denoted with N, through a first order filter:

$$\mathbf{a}_t^N = \mathbf{a}_{t-1}^N + \tau \left(-\mathbf{a}_{t-1}^N + \mathbf{n}_t \right) \tag{23}$$

where τ represents the filter time constant ranging in $(0, 1)$ and $\mathbf{n}_t$ is a vector having each element randomly drawn on the basis of a uniform probability distribution in $[-\epsilon, +\epsilon]$ at each time step.

III. Results

A. Measuring transfer quality

To measure the potential for transfer of TERL, its performance was compared with two alternative systems furnishing the upper and lower bounds of learning speed curves (cf. [6]), namely: (a) a system that learns to solve the new task based on a certain source task: this may result in an advantage or a disadvantage for learning the new task depending on its similarity to the source task; (b) a system that learns to solve any new task from scratch, without the possibility of skill transfer. The two systems are also important as they view the multiple-task problem as either a set of separated Markov Decision Processes (MDPs; [24]), each corresponding to a single task, or as a whole MDP, where the problem is to maximise the reward given multiple tasks. The possibility of viewing the problem in these two ways is important for distinguishing the system discussed here from other related systems (see Section IV).

The first system, called 'SIM' (which stands for '*sim*ple system'), is simply formed by one critic expert and one actor expert and no gating networks (each of the two experts is a linear function approximator getting as input the Gaussian features **f** used to encode the arm posture). SIM is capable of learning only a policy mapping $\pi_k : S \times A \to [0, 1]$ to solve one specific MDP problem k denoted with $< S, A, T, R_k >$ (where S denotes the set of states, A the set of actions, T the transition function mapping the current state and action to a probability distribution over the next states, and R_k the reward function of task k mapping the current state, action and resulting state to a reward value). This, and the fact that SIM did not receive an input on the solved task, implies that when it learns a new task it fully forgets the previously one. Most tests illustrated below required the solution of tasks in sequence: in this condition, SIM allowed the identification of the advantages and disadvantages of attempting to solve each new task *starting from the previous one*. SIM used identical parameters as the experts of TERL.

The second system, called 'EXP' (which stands for 'system with an input *exp*anded with information about the identity of the task'), as SIM is again formed only by one actor expert and one critic expert and no gating networks. However, in EXP such experts have enough computational capabilities to solve all tasks. In particular, in EXP the two experts receive information not only on the arm posture (features **f**), but also on the task identity (**z**): EXP uses the information about the task identity as an additional input to the experts together with the posture features (hence forming an 'expanded' input) rather than as information sent to the gating networks (not present in EXP) as in TERL. Technically, the task identity is used as a further dimension of the input problem space so that the Gaussian basis functions encoding the features (**f**) are reproduced for a number of times equal to Z (the number of tasks) and activated only when the corresponding feature z is active. For this reason, in the tests EXP allowed the identification of the advantages/disadvantages of solving new tasks *starting from scratch* without any opportunity of transferring knowledge from previously acquired ones (no generalisation). At the same time, the system allow the measure of performance in complete absence of catastrophic interference. EXP used identical parameters as the experts of TERL.

Differently from SIM, but the same as EXP, TERL is informed of the task at hand. However, unlike EXP, TERL uses the information regarding the task identity to take advantage of the fact that different tasks share the same states, actions, and transition function to transfer knowledge from acquired tasks to new tasks to be learned.

The minimal performance that we require from TERL is that it converges towards the best MDP solution, i.e. *performance optimality* [56]. The core of the skill-transfer problem is however captured by *learning optimality* [56], namely the fact that an algorithm achieves a high (in theory the best) learning speed on new tasks thanks to the previously acquired ones. The literature on TRL measures this capacity for transfer through three metrics [6], [15]: (a) the *jumpstart*: i.e., a higher initial performance when learning a new task based on transfer with respect to a non-transfer condition; (b) the *learning speed*: i.e., a faster learning with respect to a non-transfer condition; (c) the *asymptotic improvement*: i.e., an improvement in asymptotic performance (i.e., the performance when the learning process achieves its maximum after a sufficiently long training). This is relevant only when a difficult task can be learned only after having previously learned other tasks, otherwise this criterion is not distinct from performance optimality.

Here we required not only that TERL approaches performance optimality (solution of the tasks), but also that it approaches the performance of SIM when transfer is advantageous and the performance of EXP when starting from scratch is the best thing to do, while at the same time avoiding catastrophic forgetting. A comparison between the expected performance of SIM and EXP and the desired performance of TERL in different conditions is summarised in table I.

B. Tests with the planar arm: sequential learning of tasks

TERL was tested with two setups requiring simulated robotic dynamical arms to reach targets positions in space. These setups were chosen because: (a) they involve continuous state and action spaces and the control of dynamical plants; (b) they allow a useful visualisation and analysis of the performance of the system, in particular the parallel visualisation of the postures/movements of different actor experts; (c) they involve limb movements, a category of biological phenomena that could be investigated with TERL in future work.

TABLE I
Expected performance of SIM and EXP, and desired performance of TERL, when learning a new task after having previously learned a task requiring the *Same* skill, a *Similar* skill, or a *Different* skill. In particular, SIM starts from the previous skill, so it has an advantage if transfer is possible but at the same time always loses the capacity to solve the previously solved task (catastrophic forgetting). EXP learns each task from scratch, so it cannot have any advantage of transfer nor any disadvantage in terms of catastrophic interference. TERL was designed to produce the best outcome in all conditions: that is, to take advantage of transfer whenever possible, to avoid wasting time when transfer is not possible, and to always avoid catastrophic forgetting. *L*: expected learning speed; *C.f.*: expected effects of catastrophic forgetting; *N*: 'neutral' performance, as in learning from scratch (without transfer); *G*: good performance, benefiting of transfer or having no catastrophic forgetting; *B*: bad performance, worse than starting from scratch, or presence of catastrophic forgetting. A bold letter indicates the best outcome in the given condition.

Prev.	SIM		EXP		TERL	
task	L.	C.f.	L.	C.f.	L.	C.f.
Same	**G**	**G**	N	**G**	**G**	**G**
Similar	**G**	B	N	**G**	**G**	**G**
Different	B	B	**N**	**G**	**N**	**G**

The first set-up involved a 2D 2DoF simple simulated dynamic arm working on a plane containing four 'object' goals each having a radius of $3cm$ (Figure 2). The planar arm was very useful for developing the algorithm and also facilitates the explanation of its functioning (see examples below). The second setup involves a 3D, redundant, 4DoF simulated dynamic robotic arm and allows the test of the capacity of the system to scale up to more complex tasks (Section III-E).

Figure 2 shows the 2D arm used to test the model. The arm was formed by two links: an upper arm measuring $25cm$ and a forearm measuring $35cm$. The movement range of the joints was set to [-100°;+30°] for the shoulder (0° corresponding to the upper limp located forward the robot and the angle was measured anticlockwise) and to [0°;+160°] for the elbow (0° corresponding to the straight arm and the angle was measured anticlockwise).

The dynamics of the arm were simulated based on the following equations [57]:

$$\begin{aligned} q_s &= (I_s + I_e + 2M_eL_sS_e\cos\theta_e + M_e{L_s}^2)\ddot{\theta}_s \\ &\quad +(I_e + M_eL_sS_e\cos\theta_e)\ddot{\theta}_e - M_eL_sS_e(2\dot{\theta}_s \\ &\quad +\dot{\theta}_e)\dot{\theta}_e\sin\theta_e + B_s\dot{\theta}_s \qquad (24) \\ q_e &= (I_e + M_eL_sS_e\cos\theta_e)\ddot{\theta}_s + I_e\ddot{\theta}_e + \\ &\quad M_eL_sS_e\dot{\theta}_s^{\,2}\sin\theta_e + B_e\dot{\theta}_e. \end{aligned}$$

where q_s and q_e are the actuated torques of the shoulder and elbow joints, respectively, and the parameters M, L, S, I, and B are respectively the mass, the length, the distance from the centre of mass to joint, the rotational inertia of links, and the coefficient of viscosity (these parameters were set to $\{0.9, 0.25, 0.11, 0.065, 0.08\}$ for the shoulder joint and to $\{1.1, 0.35, 0.15, 0.1, 0.08\}$ for the elbow joint as in [57]). The equations were integrated with a 4^{th} order Runge-Kutta method using a time step of $0.01sec$.

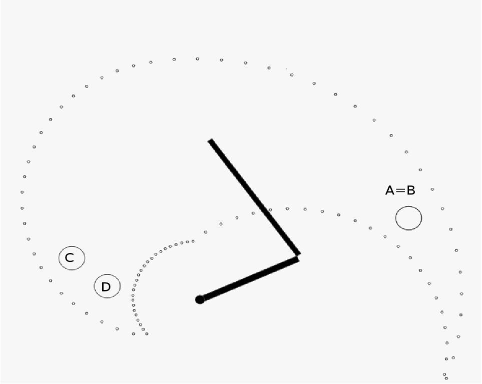


Fig. 2. The dynamic 2D arm and the four target objects A, B, C and D (A=B indicates that the location of A coincides with the location of B). Dots represent the borders of the workspace established by the length of the arm links and the range of its joints. The work space is asymmetric as the 'elbow' joint range is asymmetric with respect to the upper arm as in a humanoid robot arm.

The arm had two actuated DoF: one for the shoulder joint (θ_s) and one for the elbow joint (θ_e). In particular, the robot controllers used here set the desired posture (or 'desired angles', or 'action') of the arm, and 'skills' represented mappings from sensory input to desired postures. A proportional derivative controller (PD) was used to generate the torque of each of the two arm joints [58]:

$$q = K_p(\theta_{des} - \theta) - K_d\dot{\theta} \qquad (25)$$

where θ and θ_{des} are respectively the current and desired joint angles, and K_p and K_d are respectively the proportional and damping gains (K_p was set to 25 and K_d to 4 for both joints).

The objects shown in Figure 2 allow the specification of four reaching tasks, each requiring that the system learns to reach one specific object. We will call these tasks 'Task A', 'Task B', etc., depending on the target object. The solution of each task requires the system to acquire a skill that allows the system to reach one of these objects starting from any initial posture. All trials involving the solution of the tasks started with the arm set at a random posture. Each trial terminated either with the achievement of the goal (target object) or with a time out of 6 sec (i.e., 600 steps of 0.01 sec each; 0.01 is also the integration time step used for dynamical equations of the arm). The models got a reward of 1 when the hand touched any point of the target object, and 0 otherwise. This implies that the system can work with tasks defined in terms of reaching a set of states, rather than a single state, to the extent that a suitable reward function can be associated with them.

The tasks were used to build three tests corresponding to the three conditions of the rows of Table I. In each, the system had to first learn a task during n trials, and then another task for further n trials (this is what we call here a 'sequential test'). The first test involves first learning Task A and then Task B each for 200 trials: as these tasks require the same sensorimotor mapping this allows us to check if TERL is able to re-use experts used to solve Task A when learning to solve Task B and how efficient it is in doing so (by comparing TERL with SIM, which performs an instantaneous transfer, this allows us to quantify the 'overhead' cost of solving the task with the hierarchical architecture of TERL rather than with a simple expert). We set the number of trials for this and other experiments through pilot experiments that showed the amount of learning allowing the best tested systems for each condition to achieve steady-state performance. The second test involves first learning Task C and then Task D each for 400 trials: as these tasks require similar sensorimotor mappings, TERL should exhibit a transfer advantage when learning D in comparison to EXP that learns all tasks from scratch. Finally, the third test involves learning first Task A and then Task D each for 700 trials: as these tasks require very different sensorimotor mappings, TERL should realise that no transfer has to be attempted, and show a performance similar to EXP, that learns from scratch, and superior to SIM, that starts to learn Task D from the different sensorimotor mapping learned for Task A.

The avoidance of catastrophic forgetting can be ascertained by comparing the performance in the first task (e.g., C) after learning it, with the performance in the same task after the system has learned the second task (e.g., D) and without re-learning the first one. The system is robust to forgetting if the performance in solving the first task does not decrease after learning the second task.

Overall, the tests involve these challenges: (a) The experts have to learn to associate, at each time step of the trial, a suitable desired posture (output) to the current posture of the robot (input): the dynamics of the arm will then generate the actual trajectory of the arm given the desired equilibrium points selected by the experts; no cost is given for performing movements, but due to the discount of the reward (see Section II-B) the RL algorithm tries to find trajectories of equilibrium points that minimise the time taken by the dynamic arm to touch the target object. This aspect is not further discussed here, but in previous work we have shown how one expert of the type used here can learn to produce quite complex equilibrium-point trajectories to minimise such time and possibly move around obstacles [59], [60]. (b) The gating networks are only informed about the identity of the currently solved task (Task A, Task B, etc.) and on this basis they have to learn to select the best expert(s) to solve it. (c) The management of the expert copies by the learning algorithms have to support the exploitation of the opportunities for transfer between tasks while at the same time avoiding catastrophic interference.

The initial connection weights of the expert actors were randomly drawn from a uniform distribution ranging in [-0.2, 0.2], and the connection weights of the critic experts were set to zero for SIM, EXP, and TERL The connection weights of TERL gating networks were set to zero. The actor and the critic of TERL were each formed by ten experts each. This number was chosen to give the system enough resources to form main experts (i.e., experts with the highest functioning responsibilities) and copies for the three tasks and to study the selective recruitment of redundant resources by the system's gating networks. Other parameter values of TERL are summarised in the Appendix.

Figure 3 illustrates the learning curves of SIM, EXP and TERL in the three tests. In particular, for each test the figure reports the learning curves related to the first and second task forming each test.

The results of the first test, formed by two tasks involving the same sensorimotor mapping, show that TERL manages to quickly discover that in order to solve the new (second) Task B it can reuse the skill previously acquired by solving Task A. The overhead cost for this discovery is rather low (compare TERL performance with SIM performance in Task B). TERL has also some advantage on the other models in the Task A solved as first: the reason for this is that TERL experts are initialised to incorporate different initial behaviours from which the selection mechanism can draw one that is most appropriate for the current task (this is further explained on the basis of some figures in Section III-C).

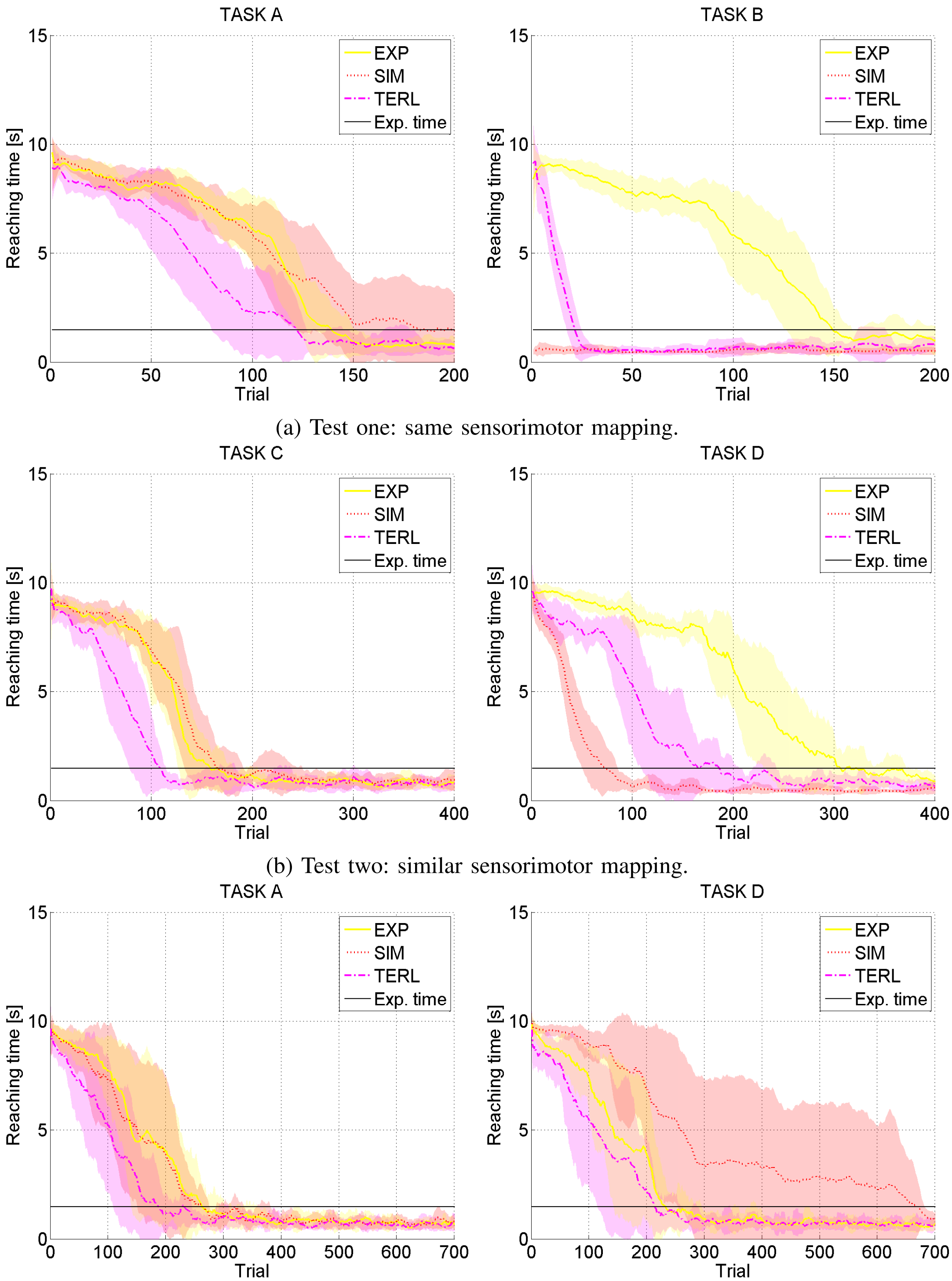
The results of the second crucial test, formed by two tasks involving similar sensorimotor mappings, show that TERL is capable of discovering that in order to solve the second task (Task D) it can start from the skill previously acquired by solving Task C and this produces a notable advantage in its learning speed. The difference in performance between EXP (starting from scratch) and TERL in Task D shows the transfer advantage for TERL. The difference in performance between TERL and SIM (transferring immediately) in Task D shows the overhead cost that TERL pays to understand the possibility of exploiting transfer.

Finally, the results of the third test, formed by two tasks involving very different sensorimotor mappings, show that TERL manages to avoid using the skill previously learned for Task A to solve the new task D as the two are quite different. The difference in performance between SIM and TERL in Task D shows the cost that TERL avoids by not attempting to transfer from the different previously-learned Task A.

While having these transfer strengths, TERL also manages to avoid catastrophic forgetting. Figure 4 shows the comparison between (a) the performance of the three models in the first task (in each of the three tests) right after such task has been learned and (b) the performance right after the model has learned the second task (with the learning rates set to zero, so as to avoid re-learning). The comparison of the performance in the first task, before and after learning the second task, allows the evaluation of how the learning of the second task interferes with the performance of the first one.

As expected, SIM suffers catastrophic forgetting both when the second task is similar but not identical to the first one and when the two tasks are very different. Again as expected, EXP performs very well since it learns every task from scratch, and previous acquired skills are not affected by the learning of the new ones. What is most important is that, as expected TERL does not suffer of the problem of catastrophic forgetting in any of the three tests. The reason is that when a new task is the same or similar to the previously learned task, TERL uses a copy of the previously acquired skill to solve the new task thus not damaging the capacity of the best expert trained to solve the previous task. When the new task is different, TERL does not transfer but rather uses a completely new expert, thus avoiding interference.

To show the actual formation and use of background copies

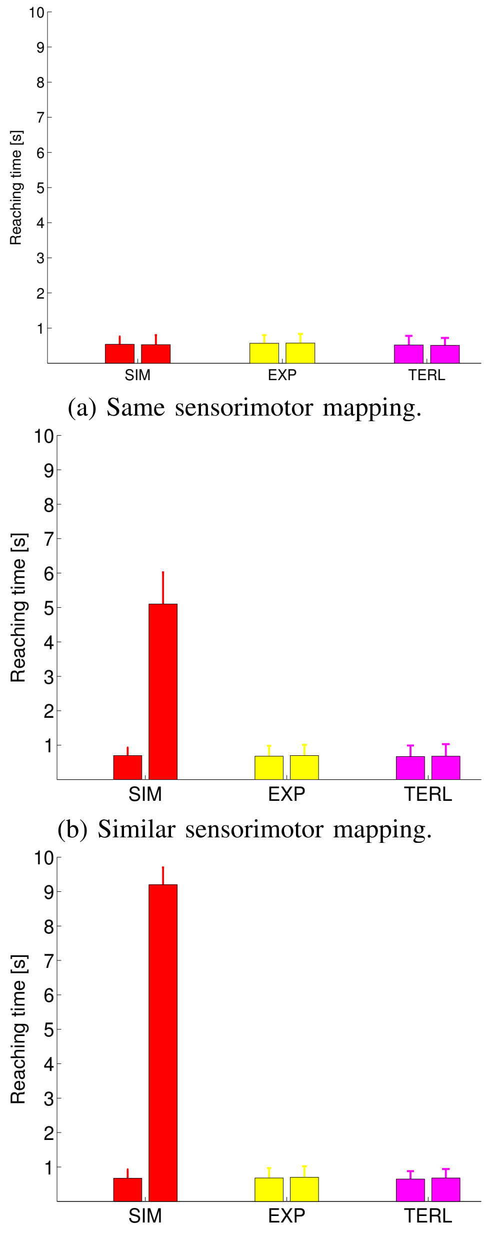


(a) Test one: same sensorimotor mapping.

(b) Test two: similar sensorimotor mapping.

(c) Test three: different sensorimotor mapping.

Fig. 3. Performance (reaching time) during learning in three 'sequential tests' in each of which the systems TERL, SIM, and EXP learn a first task for n trials (200; 400, 700 in the three conditions; left panels) and then a second task for further n trials (right panels). Each curve represents the average and standard deviation of ten repetitions of each experiment. (a) Test one: test with two tasks requiring the same sensorimotor mapping (Tasks A and Task B). (b) Test two: test with two tasks requiring a similar sensorimotor mapping (Tasks C and Task D). (c) Test three: test with two tasks requiring a different sensorimotor mapping (Task A and Task D). 'Exp. time' represents the 'exploitation time' of the noise generator (as explained in detail in Section II-B, this represents an initial period of time at the beginning of each trial during which exploratory noise is kept at a minimum to foster exploitation).

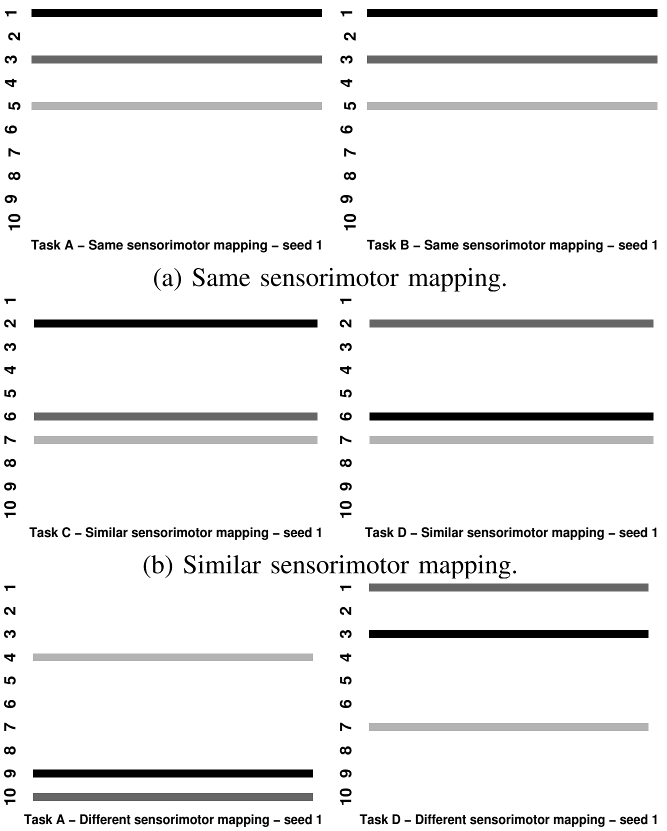


(a) Same sensorimotor mapping.

(b) Similar sensorimotor mapping.

(c) Different sensorimotor mapping.

Fig. 4. Test on catastrophic forgetting. Each graph shows the average and standard deviation of reaching time in the first task before (bars on the left) and after (bars on the right) learning the second task. (a) Same sensorimotor mapping (Task A then Task B). (b) Similar sensorimotor mapping (Task C then Task D). (c) Different sensorimotor mapping (Task A then Task D).



(a) Same sensorimotor mapping.

(b) Similar sensorimotor mapping.

(c) Different sensorimotor mapping.

Fig. 5. Use of the actor experts by TERL in three tests, each involving the solution of two tasks in sequence, characterised by: (a) the same sensorimotor mapping; (b) a similar sensorimotor mapping; (c) a different sensorimotor mapping. Each of the three graphs shows the three highest prior responsibility signals of the ten experts in one trial after learning the first task (left panels) and in one trial after learning the second task (right panels). In each graph the highest, second highest, and third highest priors are respectively marked with a black, dark gray, and light gray strip, while all other priors are not marked (white). The data refer to one specific system (seed 1): in other replications of the experiment, or if the analysis is repeated for the critic experts, the results are the same but of course involve different experts. Note that this type of graphs give information on the selection of experts during the whole trial: here the graphs simply show that TERL reliably selected specific experts during the whole trial; in other conditions the graphs are very useful to analyse expert-based models similar to TERL, especially at the beginning of the learning process or when using gating networks having highly varying input.

by TERL, Figure 5 shows the responsibilities that it assigns to actor experts during learning in the three tests involving same (a), similar (b) or different (c) sensorimotor mappings (left panels show responsibilities of actor experts for the first task of each test; right panels for the second one; data are qualitatively similar for the responsibility signals of the critic experts, data not shown). The graphs show that, when the two tasks require the same sensorimotor mapping (Figure 5a), TERL uses the same experts with the same responsibilities, i.e. it re-uses the skill developed for Task A to solve Task B. When the two tasks require similar sensorimotor mappings (Figure 5b), TERL solves the second task, e.g. Task D, by re-using a copy expert developed in background while solving the previous task, e.g. task C. Thus TERL is able to exploit previously acquired knowledge to solve new tasks, and at the same time to avoid catastrophic interference, by recruiting and modifying expert copies developed for previously solved tasks. Finally, when the two tasks require very different sensorimotor mappings (Figure 5c), TERL uses different experts as it realises that trying to transfer knowledge would be useless or even detrimental.

C. Tests with the planar arm: scaling-up to several tasks and experts

The experiments reported in this section test the capacity of TERL to scale up to learn a larger number of tasks with respect to the previous sections, and in particular show how the mechanism of recruitment of already trained experts, or of still untrained experts to solve new tasks, continues to work with a large number of available experts. The experiments also show the capacity of TERL to recruit a restricted subset of experts among the multiple available ones, for both functioning and learning, based on the responsibility signals. For this purpose, we trained the system in sequence with the six tasks shown in Figure 6, each for 500 trials, using 30 experts for the critic and 30 for the actor (we set this large number to test the scaling capabilities of the system; the number of experts of the critic and actor do not need to be the same).

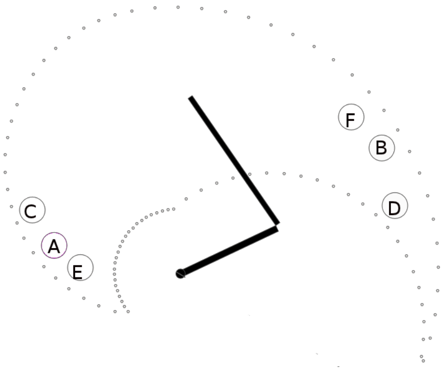


Fig. 6. Configuration of targets (tasks) used to test the scaling-up capacities of TERL.

Before illustrating the results of these experiments, we also indicate how they were also used to quantify the advantage of TERL over other systems due to the possibility of differentially initialising its experts. For this purpose, in these experiments TERL was also tested in a version where the connection weight of the bias of each actor expert network was randomly generated in [-1,+1] with the effect that the initial actions (here desired arm postures, or ‘equilibrium points – EPs’, pursued by the robot PD or PID) of all actor experts were more uniformly distributed in the work space (as usual, all other connection weights were set to zero). In the graphs we will refer to this condition with ‘TERLs’ (‘s’ refers to the ‘high initial *s*cattering’) to distinguish it from the condition with a low initial scattering used so far where the bias connection weight was drawn in [-0.1, +0.1] to have some differentiation. Figure 7 shows the postures of the actor experts in the low and high scattering condition in the initial phase of training. Having initially scattered actor experts is advantageous as it produces different initial postures/behaviours and this facilitates the selection of them by the information-accumulation mechanism of the actor gating network. Although low, the initial (small) scattering of the actor experts used in TERL also explains the advantage it has with respect to SIM and EXP in learning the fist task in the tests shown in Figure 3. Notice that the possibility of exploiting the best skill from a rich initial repertoire of skills is general as TERL samples in parallel the goodness of all available experts for the current task (since it compares the action of all experts with the action actually performed) and so it rapidly focuses the selection on the best available one. Without such a mechanism (e.g., as is in SIM and EXP) the initial repertoire of experts, and in general any repertoire of experts available at a later time, could not be probed to isolate the best expert usable to solve the current task. Note that TERL rather than TERLs was used throughout the paper to avoid that the additional advantage given by scattering confounded the advantage of the system based on the mechanisms identifying the best experts from which to transfer knowledge (below we present some other results on TERLs alongside TERL to show its advantages).

Figures 8 and 9 shows the results of the test on the six tasks learned in sequence. TERL performs better than both the EXP and the SIM architectures. This confirms the capacity of TERL to scale up to a higher number of experts, 30 in this case, and to exploit transfer opportunities when this is possible (e.g., when learning Tasks C and E after Task A, and Tasks D and F after Task B). When the actor experts start with a higher scattering of the initial EPs (TERLs), the system has an even higher performance due to the advantages mentioned above.

A larger number of experts involves the following effects on computational costs: (a) the functioning and learning responsibility signals of experts are computed in parallel on the basis of the action/value errors of the experts: this implies that the number of tests to do with a task in order to evaluate the experts does not increase with the number of experts; (b) the computation of the activation of experts increases linearly with the number of experts (but notice that the activation of different experts is independent, so it could be implemented in parallel hardware); (c) the learning processes involve only a given subset of experts for each step (here eight in total) so it does not depend at all on the total number of experts (recall that thanks to the functioning/learning decoupling, the number of experts learning at each trial is fixed and depends on how many background copy experts one wants to train, not on the total number of available ex- perts). Importantly, this also means that the computation time needed by the system functioning and learning is fully independent of the experience already acquired, contrary to most TRL systems that become slower with the accumulation of experience (see Section IV).

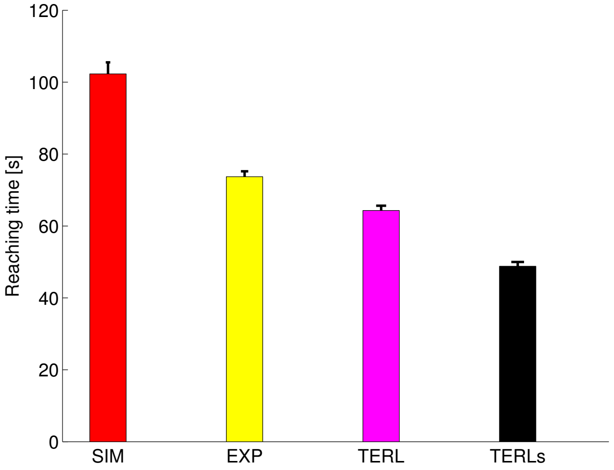


Fig. 9. Overall performance of SIM, EXP, TERL, and TERLs during learning of six tasks in sequence: sums of the integrals, one for each model, of the mean curves of the six graphs plotted in Figure 8.

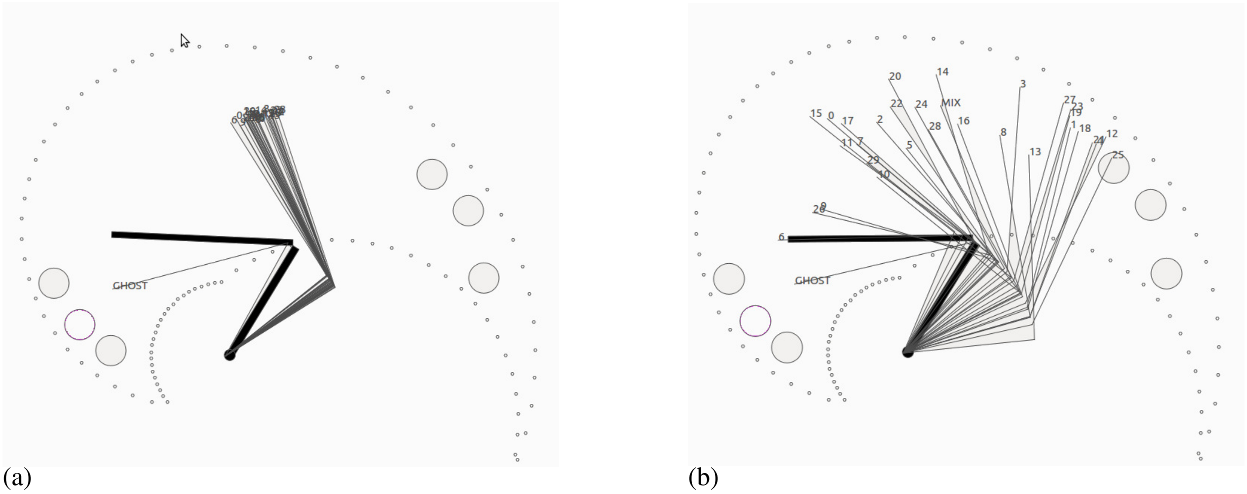


Fig. 7. (a) Snapshot of the postures (actions) suggested by the actor experts in the low scattering condition (TERL). (b) Snapshot of the postures (actions) suggested by the actor experts in the high scattering condition (TERLs). The thick stylized 2D arm indicates the current arm posture whereas the other thin stylized 2D arms represent the arm postures suggested by the 30 actor experts (the number on each arm drawing indicates the expert). The thin arm marked with ‘MIX’ represents the overall arm posture suggested by the mixture of experts; the thin arm marked with ‘GHOST’ indicates the arm posture suggested by the exploration noise added to the mixture posture.

D. Tests with the planar arm: interleaved tasks

We also tested the capacity of TERL and TERLs to solve multiple tasks when these are learned in an interleaved fashion rather than in a sequence of blocks each focused on a task, as done in the previous tests. For this purpose, the six tasks considered in the previous subsection were learned during 1500 trials where at each trial the task to be learned was randomly selected. As before, TERL and TERLs used 30 experts for both the actor and the critic.

Figure 10 shows the results of the test. The first interesting point is the advantage of TERL on EXP, which learns from scratch. Although not large, this advantage is important as it shows that when TERL learns similar tasks at the same time with interleaved trials the information gained is shared among the experts. For example, when the system has to learn to reach tasks A, C, E starting from EPs initially set at the centre of the work space, the experts that it forms can initially be the same, so that up to a certain point modifying them for solving one task also improves them to accomplish similar tasks. A second relevant point is the large advantage of TERLs versus all other models, confirming that the initial differentiation of the experts can significantly boost learning speed. Last, for all conditions a higher performance also tends to produce a lower statistical variation of the results of different repetitions of the tests.

E. Test with a 3D redundant simulated 4-DOF arm (iCub)

This section illustrates a test to evaluate whether TERL was capable of exploiting between-skill transfer while avoiding catastrophic forgetting in more complex conditions. The test required in particular to control the 3D 4DoF simulated robotic dynamic arm of the humanoid robotic platform *iCub*, an open-source robot built for studying cognitive development in humans [61].

Figure 11 shows the simulated setup used in this test formed by the iCub robot and a 3D environment containing 3 spherical target objects (in the simulator, these three objects can be anchored to the world without an actual physical support). Each arm of the iCub has 16 joints: 3 for the shoulder (J_{0-2}), 1 for the elbow (J_3), 3 for the wrist (J_{4-6}) and 9 for the hand (J_{7-15}) [1]. Here we used TERL to control the movements of four joints of the right arm, in particular: J_0, the ‘shoulder pitch’, responsible for the front-back movement when the arm is aligned with gravity; J_1, the ‘shoulder roll’, affecting the adduction-abduction movement of the arm; J_2, the ‘shoulder yaw’, affecting the yaw movement when the arm principal

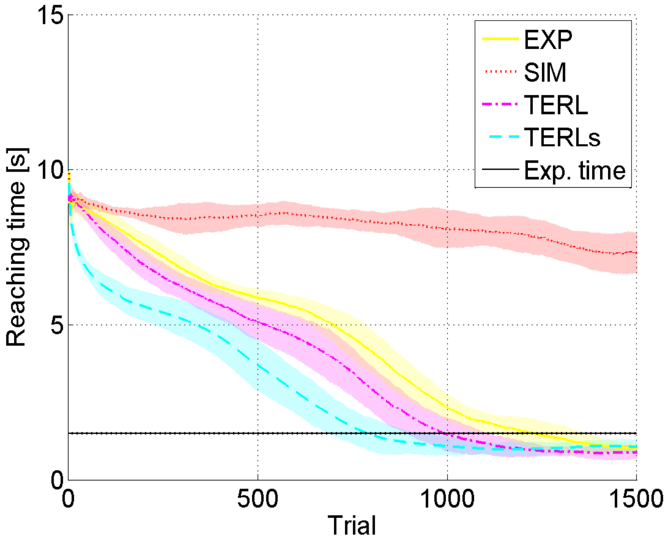


Fig. 10. Performance of the different systems during learning (trials) in the interleaved test where six tasks changed at each trial and for several times. The performance was measured as the time taken by the systems to reach and touch the target during a trial. The six tasks are those of Figure 6. Curves are averages of ten repetitions of the simulations.

[1] $http://wiki.icub.org/wiki/ICub\_joints$

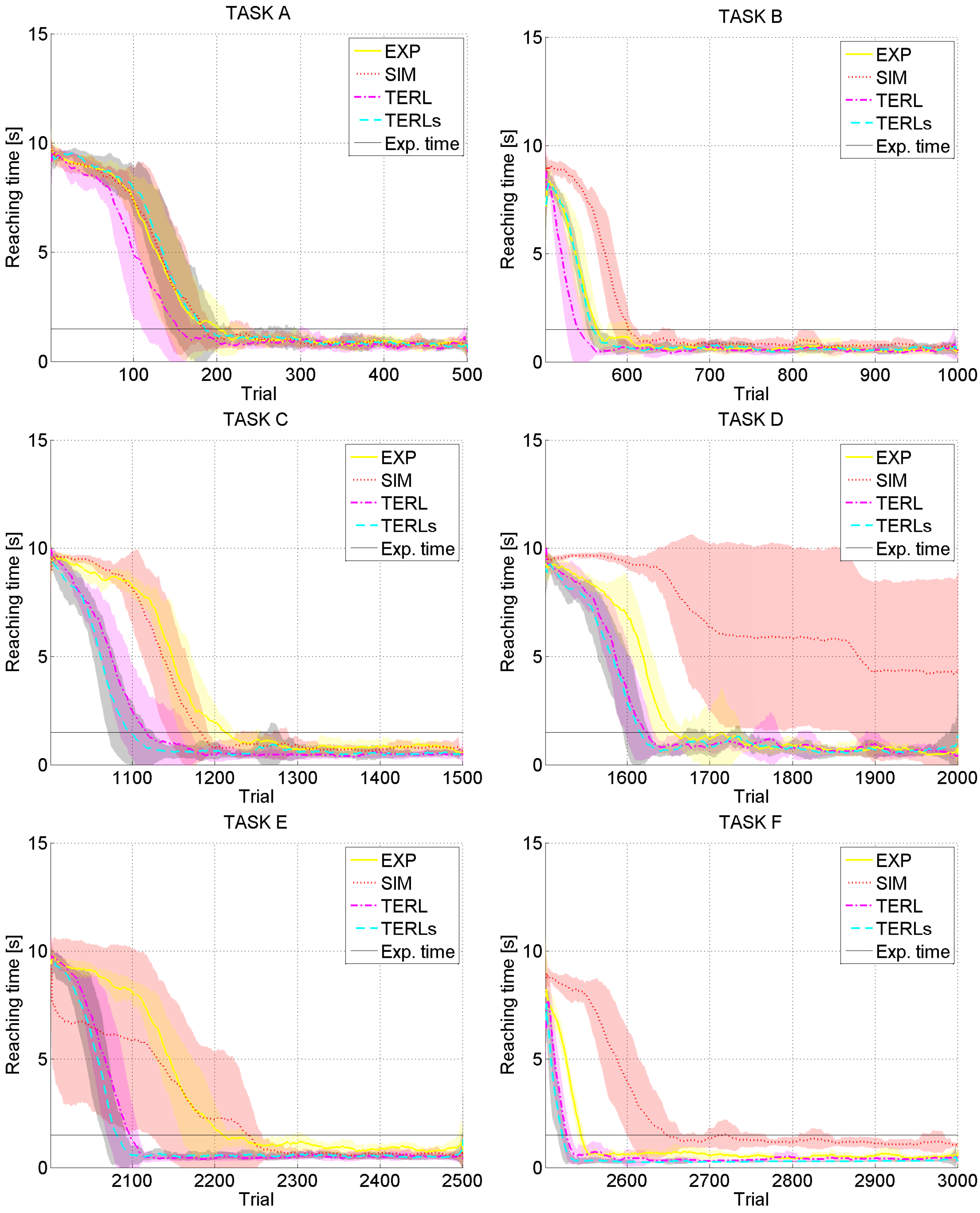


Fig. 8. Reaching time of SIM, EXP, TERL, and TERLs during learning of six tasks in sequence. The test involved learning one after the other the targets A, B, C, D, E, and F shown in Figure 6, each for 500 trials. Each curve and shadow represent the average and standard deviation computed over ten different repetitions of the simulation.

axis is aligned with gravity; J_3, the joint related to the elbow. During the simulation J_0 could assume values in the range [-80°;-15°], J_1 in the range [10°;110°], J_2 in the range [-10°;75°], and J_3 in the range [20°;85°]. The positions of the remaining joints were set at fixed values ($J_4 = -10°$; $J_5 = -30°$; $J_9 = 80°$; $J_{6-8} = J_{10-15} = 0°$). The torso joint affecting the yaw with respect to the vertical axis was fixed to $-30°$. All trials started with the arm set at a fixed posture so that the obstacle was in the way of the targets at each trial: $J_0 = -90°$, $J_1 = 100°$, $J_2 = 90°$, $J_3 = 6°$.

The three spherical objects in the environment had a diameter of $3.5cm$ and were set in front of the robot (Figure 11). The objects allowed the implementation of two reaching tasks, each requiring that TERL learn how to control the right arm of the iCub in order to reach either object B ('Task B') or object C ('Task C') while avoiding hitting the 'obstacle' A.

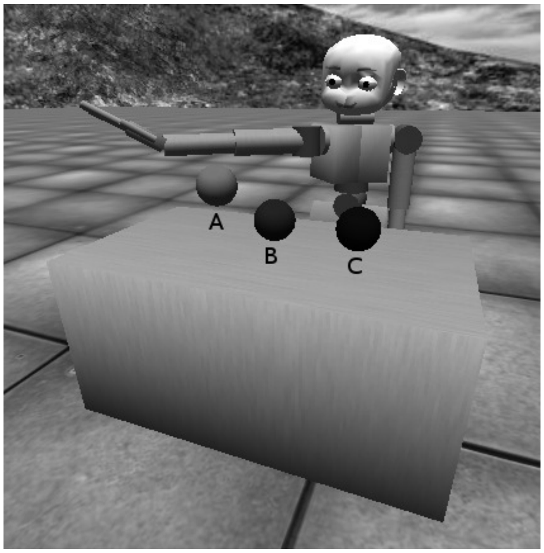


Fig. 11. The iCub robot and the environment used to test TERL. The picture refers to the beginning of a trial. Object A represents an obstacle, whereas objects B and C represent the target objects that the robot has to reach.

The system had the same architecture and functioning of the system of the previous sections, and was trained as in the sequential test involving the 2D arm with a few differences. First, all trials involving the solution of the tasks started with the arm set at a fixed posture at the right of object A. This made the task more difficult as object A was always an obstacle for reaching the two targets B and C (Figure 11) and so the robot had to perform curved trajectories around the obstacle to reach the targets. Second, each trial ended when the iCub hit any one of the three objects with the hand, or after a timeout of 8 sec (in the 2D tests the time out was 6 sec; as before, the integration time-step of the model and robot equations was 0.01 sec). The longer trial duration allowed a longer exploration necessary as the model had to discover a more complex trajectory to reach the targets while controlling a redundant arm (redundancy required the algorithm to autonomously converge to one possible solution among all those explored). Third, when the iCub hand hit the obstacle the model got a negative reward signal set to -0.5, whereas if it touched the target object received a reward equal to 1. The reward was 0 otherwise. Fourth, TERL was endowed with 10 experts for both the actor and the critic. Last, the values of a few parameters of the model were changed to take into account the different setup, in particular the higher number of controlled degrees of freedom that required a different number of output units, and thus a different number of Gaussian functions used by the algorithm as features (see Table II in Appendix).

In the test, the system first learned Task B and then Task C, each for 5000 trials. As the tasks require a similar sensorimotor mapping, they allowed us to evaluate whether TERL was capable of re-using the experts employed to solve Task B when learning to solve Task C. To show the capacity of TERL to exploit transfer we also tested EXP in the same experiment involving first learning of Task B and then of Task C.

Figure 12a shows that TERL learns the second task (Task C) much faster then the first task (Task B) as it can partially transfer the ability acquired for the first task to the second one. The fact that the advantage in task C is actually due to TERL's transfer capacity rather than a different difficulty of the Task C with respect to Task B is demonstrated by the fact that TERL is much faster in learning Task C than EXP (Figure 12b). The same result is shown by the reaching time instead of the reward (Figure 13).

The curve trajectories performed by TERL to solve the obstacle avoidance Task B and Task C are shown in Figure 14. These curves show how the sensorimotor mappings needed to solve the two tasks are similar which explains the higher performance of TERL capable of taking advantage of this.

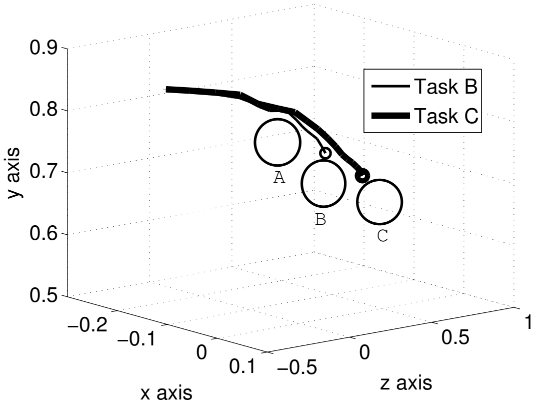


Fig. 14. Trajectories of the iCub hand produced by TERL after learning two 3D tasks requiring a similar sensorimotor mapping (Tasks B and Task C while avoiding A, see Figure 11). The trajectories refer to one run of the simulation but other runs produced qualitatively similar results.

IV. Discussion: TERL comparison with other models

This section compares TERL with other models illustrating its similarities and novelties with respect to them and the different sub-problems of transfer reinforcement learning they face.

a) The option framework and off-policy learning: Several systems from the RL literature perform transfer of knowledge between tasks by relying on the option framework ([62] for a review). An option is a data structure that encapsulates a policy (a skill), an initiation set (states where the option can be se- lected), and termination condition (establishing when the option execution terminates, e.g. when some subgoal states are achieved). Options can be used as building blocks to solve different complex tasks through their suitable combination [63]. Option-based systems have been mainly used to face compositionality problems rather than skill-to-skill knowledge-transfer problems as here. These are two very

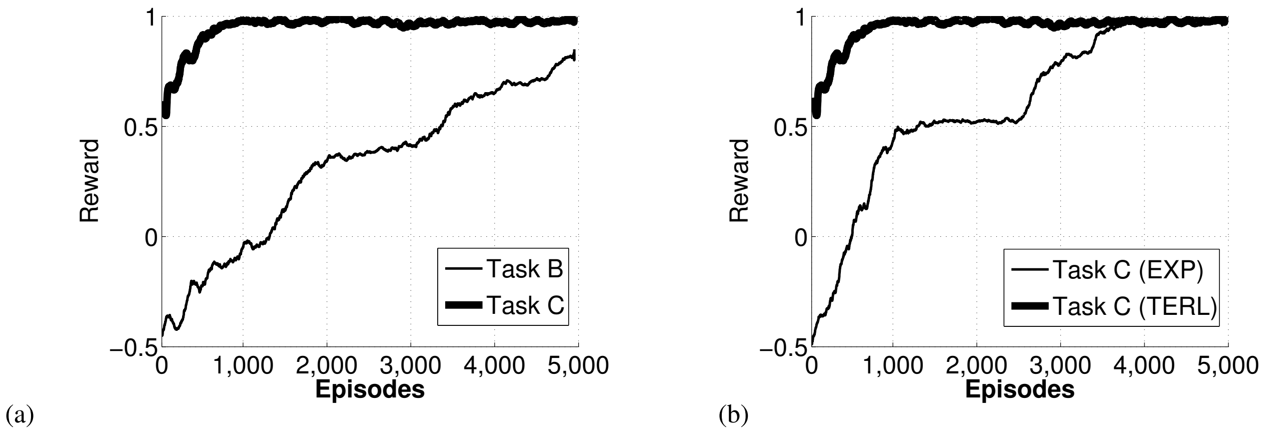


Fig. 12. Performance of TERL and EXP where the systems are first trained on Task B for 5000 trials and then on Task C for another 5000 trials. (a) Reward acquired by TERL during the sequential learning of the two 3D tasks requiring a similar sensorimotor mapping (Tasks B and C): although Task B and Task C are learned sequentially, the learning curves are represented in the same graph to ease comparison. (b) Reward acquired by TERL and EXP during learning of Task C and showing the advantage of TERL over EXP due to its capacity for transfer. Each curve represents an average over ten repetitions of the experiment, and data are smoothed over 100 trials. Notice that TERL solving task B (graph (a)) and EXP solving task C (graph (b)) have an initial negative reward indicating that they often hit the obstacle.

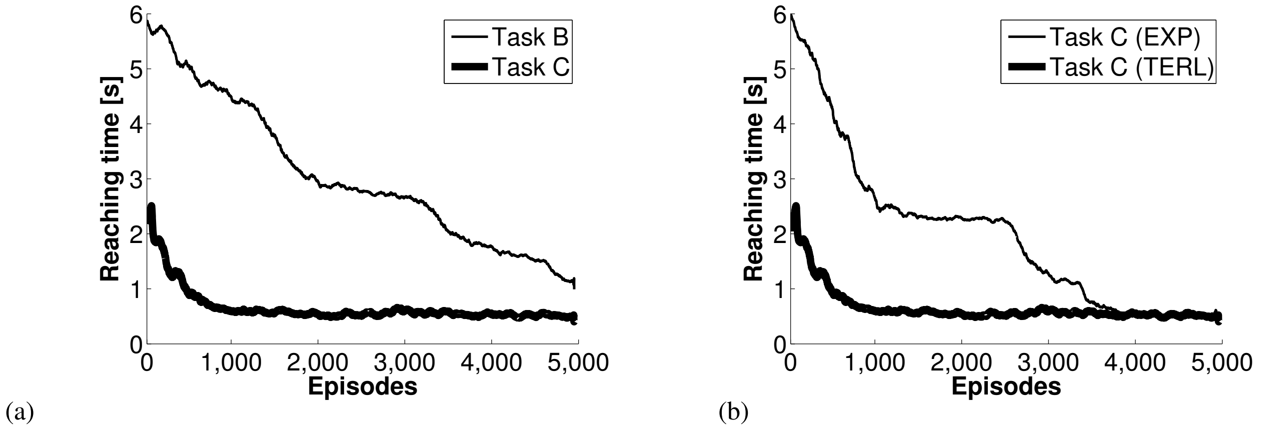


Fig. 13. (a) Reaching time of TERL during the sequential learning of the two 3D tasks requiring a similar sensorimotor mapping (Tasks B and C). TERL first learns Task B for 5000 trials and then learns the second Task C for other 5000 trials. (b) Reaching time of TERL and EXP during learning of Task C. Each curve represents an average over ten repetitions of the experiment, and data are smoothed over 100 trials with a moving average.

different problems. Compositionality problems concern how assemble multiple skills to solve complex tasks that require more than one options (skills) to be solved. For example, the option framework allows the formation of policies that select multiple options in sequence, together with primitive actions, to accomplish complex goals. Compositionality transfers knowledge from a (complex) task to a (complex) task by re-using whole skills (or part of them), rather than from acquired skills to a newly learned skill as required in skill-to-skill transfer problems. For example, the complex skill of ‘pressing a button and the complex skill of ‘scratching with the finger’ might share a very similar component-skill of ‘reaching with the finger’ performed before performing respectively the component-skills ‘pushing with the finger’ and ‘scratching’. A precursor of these systems is *Compositional Q-learning model* (*CQ-learning*) [17], relevant here also because it proposed the idea of adapting some principles of the mixture-of-experts model to a RL context. As another example, recently it has been shown [64] that the transfer of whole options between different grid-world environments can be facilitated when options are based on the agent’s sensations (e.g., egocentric position of objects) rather than on information related to the specific problem (e.g., absolute spatial relations between objects).

A key idea that can be exploited within the option framework [62], and more generally in systems formed by multiple RL experts, is ‘off-policy learning’ [63]. Off-policy learning is employed by several TRL systems reviewed below and allows RL algorithms, for example Q-learning [65], [24], to train a ‘target policy’ (and/or a value function) on the basis of actions selected by another ‘behaviour policy’. Off-policy learning has been used for ‘intra- option learning’ to let options learn from their partial execution or from the execution of actions by other components of the system [62]. A system that makes massive use of off-policy learning is the Horde architecture [66]. This architecture is also related to TERL as it is formed by multiple individually simple experts (‘demons’) that can be used for prediction or control purposes. Each demon learns an approximation (e.g., based on features and linear functions) of a *generalised value function* (GVF). A GVF stores knowledge and learns through mechanisms analogous to those used to estimate RL conventional value functions, but it can refer to elements different from rewards such as the activation of a robot sensor. A demon is based on a reward function, a termination function, a terminal-reward function, a policy (which is given in the case of prediction) and learns the related GVF. Such functions can be structured so that, for example, a demon can be used to learn to predict the time when a particular sensor will be activated, or to learn a policy to accomplish a certain activation of a sensor through an off-policy RL algorithm. TERL shares with Horde the idea of behaviour or value prediction based on multiple experts and their parallel learning, and the background objective of life-long learning, but differs from it in many respects, in particular it pivots on transfer learning to improve learning speed whereas Horde relies on off-policy learning.

b) Models from the literature on transfer reinforcement learning: Among the several systems proposed within the literature on transfer RL (TRL), we focus here on those closer to TERL, i.e. those that address the source-task selection problem by explicitly reasoning about 'libraries' of already solved tasks to decide from which of them to transfer knowledge to the new task [6]. One of these systems, called *Policy Reuse Q-learning – PRQ-learning* [67], uses off-policy learning, based on Q-learning [65] to transfer knowledge. In particular, it uses 'source policies' (i.e., previously learned policies form which to possibly transfer knowledge) to train (off-line) a new optimal policy for each new task: the source policy to use for transfer is selected at each trial on the basis of a soft-max function applied to the 'reuse gains' of all source policies (the reuse gain of a policy is equal to the average reward obtained in the the new task). The newly learned policy is added to the policy library if its reuse gain is larger than a certain threshold with respect those of existing policies. With respect to TERL, PRQ-learning has the advantage of computing an explicit metric of the similarity between tasks and the capacity to identify a core set of policies for a domain. However, at each trial it requires the selection of only one policy to use (the system is tested in a grid world) and evaluates the reuse gain of only that policy: this implies the need for an increasing number of trials to evaluate the source policies as their number increases. Instead, TERL evaluates the ability of experts to solve/evaluate the new task in parallel at each step. In addition, regarding brain modelling, PRQ-learning solves the interference problem by building a new data structure for each new task and so it could not be usable to model how the brain faces such problem.

Another class of TRL systems transfers knowledge encoded as 'experience samples', in particular tuples $< s, a, s', r >$ (i.e., the state and action at a certain time step and the state and reward at the following time step), rather than 'compiling' it into function approximators. For example, the system proposed in [68] stores the experience samples during a 'sampling phase' and then, during an off-line learning phase, computes the approximation of the action-value function on the basis of the stored samples. The key idea of the system is to implement knowledge transfer by using samples from source tasks that are similar to the target task. The selection of the source tasks and of their tuples is done on the basis of the 'compliance' (similarity) of such tuples with those experienced in the target task. The system has the advantages of basing transfer on the similarity between the dynamics and reward functions of different tasks and to be applicable to problems where both reward and transition functions change. However, its batch nature and the computation of compliance requires it to explicitly record experienced tuples, which is memory intensive, and to directly compare tuples, which implies a computational cost that increases with the number of experienced tuples. Moreover, as these mechanisms directly store tuples of experience they could not be used for modelling 'compiled', semantic knowledge as done in biological neural networks. TERL, instead, tests the experts for selection in parallel and 'compiles' the acquired information into the parameters of the experts (function approximators), so it requires computation time and memory resources that do not increase with the length of past experience.

c) The MOSAIC models: The systems most similar to TERL are those related to the *MOSAIC* model [69], so these are considered more in depth. TERL and some MOSAIC models have important similarities but also differences that make them suitable to face different problems. A first MOSAIC model similar to TERL is the *Multiple Model-based RL model (MMRL)* [70]. MMRL is composed of modules formed by a predictor of the world dynamics (taking as input the current state and planned action, and returning as output the next state) and a RL controller. During each trial, the errors of the predictors determine the responsibility signals of the modules: for this purpose, each module predictor error is passed through a Gaussian function (measuring its 'correctness') and then all the Gaussian activations are passed through a soft-max function. The responsibility signals are used to both weight-average the contribution of the RL controllers to act, and to regulate the learning rate of both the predictors and the related RL controllers. In this way, different modules specialise in predicting and acting in sub-portions of the whole problem space based on the correctness of the predictors.

These ideas are further developed in the MOSAIC-RL model [71], the MOSAIC-family model most similar to TERL. MOSAIC-RL is formed by three sets of 'modules': 'forward modules', predicting the environment dynamics (i.e., models of the state transition function), 'reward modules', predicting the 1-step reward (i.e., models of the reward function), and RL controllers (each based on a value-function approximator and a policy directly generated from the value approximator through control theory methods). With respect to previous MOSAIC models, in MOSAIC-RL the three types of modules are decoupled, so they segment the problem space differently on the basis of their respective errors: the world-dynamics prediction errors, the 1-step reward prediction errors, and the TD-errors. As in MMRL, these errors are used to compute the responsibility signals of the related type of modules on the basis of Guassian and soft-max functions. However, now the three types of modules have distinct responsibility signals, so the RL controllers deciding the action are selected on the basis of their TD error, not on the basis of the world dynamics as in MMRL. This allows the model to face not only problems with non-stationary (i.e., hidden) dynamics, as MMRL, but also problems involving 'a non-stationary reward function', i.e. multiple reward functions (multiple tasks).

Given the similarities between TERL and MOSAIC-RL, in terms of the responsibility signals computed in parallel on the basis of a softmax of Gaussians of errors, and the segmentation of the whole problem into sub-problems ('tasks'), it is important to see how they can be used to face different problems. The key difference between the two systems is that TERL uses the softmax of Gaussians to train the gating networks, not only to select the experts to act: the knowledge acquired with this learning process during the first trials in which a new task is faced (when TERL selects multiple experts for action as it still does not know which one is the best) is compiled into the parameters of the gating networks. Thus once acquired it can be used to immediately select the best expert to use, based on the information regarding the identity of the task to solve, since the very first step of the trial. Instead, MOSAIC-RL uses the softmax of Gaussians to dynamically accumulate evidence on the experts to select step-by-step during the trial, or during multiple trials, when these address the same task multiple times in sequence, even after the system has learned to solve the task. This implies an important feature of the problems that MOSAIC-RL can solve. In trial-based RL problems, relevant for transfer RL, reward is often zero during the trial and high at the end of the trial when the task 'goal' is accomplished, so the TD-error can be high only at the end of the trial. In these cases, the responsibility signals of controllers computed by MOSAIC-RL on the basis of their TD-errors are similar during the first part of the trial and differentiate only at the end of it, so they can start to have an effect on the selection of controllers only from the second trial onward, and only when the same task is experienced for more than one trial in sequence. In this respect, in commenting on the performance of MOSAIC-RL [71], it was reported that 'after learning, the RL modules also successfully switched within a few (one or two) trials when the subenvironment changed' (note that the training and test of MOSAIC-RL was done in blocks each formed by 100 trials involving the same 'subenvironment', i.e. task).

This difference implies that MOSAIC-RL is not suitable to face problems as those used here to test TERL where the task is switched at each trial and its identity is known (interleaved condition tests, see Figure 10). This situation is for example common in animals where motivation changes continuously after being 'satisfied in one trial' and is known to the animal. After learning, for example, when an animal is hungry (hunger signals a first task identity) it is able to directly move to a food dispenser, and when it is thirsty (thirst signals a second task identity) it is able to directly move to a water dispenser: this without the need to sample each time the reward given by food or water. The same holds when animals pursue a goal: also in this case the task identity is known and so the animal can immediately recall the behaviour to accomplish it (after this behaviour has been acquired and associated to the goal). Note that this behaviour of MOSAIC-RL is expected and *is not a drawback* of the system. Indeed, MOSAIC-RL has been designed to solve problems where the task identity is hidden and so has to be identified by repeated sampling: the mechanisms of the system are thus very good for facing these problems. Instead, for the TRL problems considered here, for which TERL has been designed, the information about the identity of the task to solve is clear and available before each trial. Since it addresses a different problem with respect to TERL, MOSAIC-RL does not use different responsibility signals for the evaluation function and the policy 'experts', nor does it use different responsibility signals for functioning and for learning as done by TERL. These features are very important, as shown here, when the system is used to address the skill-to-skill knowledge-transfer problem and the catastrophic-interference problem.

Overall, it can be said that MOSAIC models and TERL are best suited to solve different complementary problems: MOSAIC models are best suited to face the problem of segmentation of whole MDP problems into sub-tasks on the basis of the progressive on-line accumulation of evidence on the hidden (non-observable) world features based on the errors of predictors of the world dynamics or the world reward. Instead, TERL can be used to solve the skill-to-skill transfer problem without incurring catastrophic interference on the basis of gating networks learning to map goals to responsibilities of experts. For these reasons, the two systems might be suitably integrated in the future.

d) Other algorithms, evolutionary duplication, and task-policy mapping: The generation of multiple neural copies by TERL links it to another class of models that implement neural duplication to solve different tasks or sub-parts of the same task. One of these models [42] is based on RL experts called 'mots' that self-organise in a 2D spatial grid in a way that is reminiscent of self-organising neural networks. Mots are selected for action and for learning based on a previously proposed model called *SERL – Selected Expert Reinforcement Learner* [72]. In particular, each mot implements a RL policy – e.g., in the form of Q-learning – and at the same time learns to estimate the inaccuracy of that policy – e.g., in the form of error of the Q-learning values. At each time step, a 'winning mot' is selected for action and for learning if it has the highest Q-value reduced by the Q-value estimated error. The mots close to the winning mot also learn and this leads to the emergence of spatially organised mots that tend to specialise on the same parts of the problem space, and thus to form 'copies' of the same portion of behaviour similar to TERL. Later [43], mots maps have also been develop to capture in space the temporal relations (sequential activation) of mots by training not only the winning mot and its neighbours but also the previous and/or following winner(s) and neighbours. Future work might aim to obtain a spatial organisation of TERL experts by employing similar mechanisms as those used by mots, thus benefiting from the advantages that this carries in terms of smoothness, robustness, hierarchy by region, dimensionality reduction, and reuse (see [43]).

Another relevant class of models that implement neural duplication has been proposed in [44], [73]. These models are developed within an evolutionary framework, so they can be used to model neural duplication based on genetic processes [73] whereas their use to model learning processes related to the single individuals, as in TERL, has been proposed only recently [74], [75]. In this respect, TERL represents a relevant hypothesis as to how neural duplication might take place within the single animal behaviour and brain when trial-and-error learning mechanisms are involved [21], [22].

Another class of systems learns to map tasks to solutions, in particular it uses the information regarding features of the goals, or reward functions, to best initialise the value functions or policies to solve new tasks [76], [77]. As in TERL, in these models the transition function is assumed to be constant. While solving different tasks, these systems learn a mapping between the goal features and the parameters of the value function or policy approximators. This mapping allows the systems to generate the initial policies and value functions to solve new tasks: later these policies and value functions can be refined with further training experience. These systems face a TRL problem that is different, in particular complementary, with respect to the one faced by TERL. Indeed, they can exploit information (features) on the goals of the new tasks *before* solving them, but then when they further refine the initial solutions they cannot further transfer knowledge from previously solved tasks. On the contrary, TERL cannot benefit from information (features) on the new tasks before starting to solve them, but on the other hand it can transfer information from previously solved tasks *after* it starts to solve the new tasks based on the effectiveness of existing experts in solving them (rather than based on similarities between goals). To further clarify this difference, consider these examples. When the system faces a new task where the goal is very similar to the goal of a previously solved task, and indeed the two tasks require a very similar sensorimotor mapping to be solved, then the systems under discussion can formulate a good first guess on the solution to use. If instead the new task requires a very different sensorimotor mapping, and such sensorimotor mapping (or a similar one) was previously learned to accomplish a different goal, then the mechanisms of TERL would rapidly identify the expert encoding the latter mapping and use it to solve the new task. Similarly, the mechanisms of TERL would be useful when solving a new task where the goal does not give any useful information about the type of sensorimotor mapping to use. For these reasons, the mechanisms proposed by the systems discussed here and TERL might be integrated in future work (Section V sketches how TERL might do this).

V. Conclusions and future work

A. Main achievements

This paper has described and tested a reinforcement learning architecture ('TERL – Transfer Expert Reinforcement Learning architecture') to learn multiple skills solving different related tasks. The architecture offers a solution to the reinforcement-learning source-task selection problem [6] requiring the identification of the skills acquired to solve previous tasks from which to transfer knowledge to best solve new tasks.

We sought a solution to this problem under two stringent conditions. First, TERL was not given any information regarding the similarity between the solved tasks and the new tasks and so it had to sample the actual performance of the possessed skills in the new task to infer their relevance for it. This condition was introduced to ensure the development of algorithms that are able to exploit the knowledge acquired while solving the new task without having any prior information on the similarity between the solved tasks and the new task. The solution proposed here is important because such knowledge on the effectiveness of previous solutions to solve the new task is always available in source-task selection problems. Once we have developed efficient algorithms capable of fully exploiting such knowledge, we can develop and add to them other mechanisms able to use additional information, for example related to the similarity between tasks (see below). The second stringent condition was that the computational resources used by the system (here the experts) were constant. The use of the same resources to solve multiple tasks can support generalisation and knowledge transfer but also introduces the well-known problem of catastrophic interference. The use of the same resources was dictated by our objective of building solutions that have biological plausibility.

Seeking the solution to the skill-to-skill transfer problem under the two conditions just described led to the development of two core mechanisms incorporated in TERL. The first mechanism accumulates evidence on the goodness of the skills, measured in terms of collected reward, during the solution of the new task. This mechanism accumulates such evidence in parallel for all the experts, so continues to work well when the number of experts increases (it does not require the test of single experts separately one after the other). The second mechanism involves the decoupling of the responsibility signals used for the functioning and for the learning processes of the experts. This decoupling allows TERL to overcome the catastrophic interference problem.

The tests of TERL with a 2D robotic dynamic arm showed that the system works both in conditions where tasks have to be learned and performed in sequence (i.e., with several blocks of trials for each task: this challenges the robustness of the system to catastrophic forgetting) or in random order (i.e., with one different task per each trial: this challenges the capacity of the system to rapidly decide which experts to use). Other tests showed how TERL scales up to larger numbers of experts and tasks with little additional computational costs. Also, in contrast to several transfer reinforcement learning systems that explicitly store experience in the form of state-action-state-reward tuples, the speed of functioning of TERL mechanisms is fully independent of the amount of knowledge already acquired. Preliminary experiments also showed that TERL can scale up to control a 3D redundant dynamic robotic arm.

B. Future work

The architecture of TERL could be further tested and improved in different ways in future work. In the current work, the experts were given only information about the task identity, but not on task features or task goals, to facilitate the development and study of the transfer algorithms tested here. For the same reasons, the gating network was informed only as to the task identity (with this the system only knows if two tasks are the same or different), and in particular it was not given a richer description of the task, or task goal, in terms of features (which might hint to the possible similarities between two tasks). In future work, it will be interesting (a) to give the experts further information regarding the task to solve, and (b) to give both the experts and the gating networks a rich description of the task/goal (e.g., as in the system proposed in [77] and [78], where a robot is informed on the position in space of a target to be hit with a dart or a ball). We speculate that this would have important consequences. Giving information about tasks to the experts might allow the system to encode knowledge of more than one task in the same expert, now not possible. In turn, this might allow the study of how the system encodes similar tasks in the same expert, and also allow the system to learn a number of tasks greater than the number of available experts. Giving the system a rich description of the task/goal might allow the gating networks to use such information to create useful predictions about the responsibility to assign to experts for each new task before experiencing it (here such prior signals were necessarily equal between them as the system knew only the identity of tasks and has no information about their similarity). The mechanisms described here could then be used to accumulate evidence about the actual capacity of experts to solve the new task based on their test in the new task, so as to strengthen or weaken the responsibility signals predicted initially. The output of the gating networks before and after this learning and accumulation of evidence could thus be interpreted as prior and posterior probability estimates of which experts are best to solve the new task similar to Bayesian approaches [79].

Future work might also further improve the mechanism that accumulates evidence on the goodness of experts for new tasks. In particular, here the mechanism searches only the average parameters of the Gaussian functions used to update the responsibility signals of experts, while its size (σ_A^2 and σ_C^2) is fixed. It might instead be possible to find a way to also estimate such parameters, similar to what is done in the mixture of Gaussian models [80], but taking into account the RL context considered here.

Future work should also further study and improve the mechanism used here to preserve the best experts against interference when facing some challenging conditions (e.g., when tasks are learned in sequence). Indeed, here we used a heuristic mechanism that lowers the learning rates of experts that achieve a high functioning responsibility in given tasks as this indicate that they can be reliably considered the best experts in those tasks. More principled mechanisms should hence be developed for this purpose.

Another aspect of the architecture to improve involves the treatment of noise. Here we adopted a simple solution as this was not the focus of this work. Noise was kept low at the beginning of each trial during a fixed time interval that was sufficient to solve the task if the system had already learned it, and then noise was increased to let the system explore new solutions in case of failure to solve the task during such interval. Future work should find new solutions to this problem, in particular to regulate the level of noise depending on the system's performance suitably estimated on the basis of a meta-learning process (e.g., see [81]).

Another aspect of TERL that might be developed in future work is that it currently assumes the existence of tasks to be solved, and that it receives information about the identity of the task to solve within each trial. These assumptions have important consequences on the system functioning, for example they gives an episodic nature to the reinforcement learning algorithms used by the system and this affects the algorithms to compute the global and local TD-errors, the exploration-exploitation noise, and the reset of some variables. Future work could make the system fully autonomous by endowing it with a component that is able to self-generate goals, tasks, and learning trials. For example, a recent work [82] has proposed a system that self-generates goals when the exploratory action of the simulated robot controlled by the model causes important effects in the environment, for example it turns on a spherical light by touching it. This system also self-generates 'trial-terminations' either when it successfully accomplishes the currently pursued goal or when a time out elapses. TERL might be suitably integrated with similar mechanisms for goal (task) and trial self-generation.

Given the bio-inspired nature of the ingredients used to build it, TERL might also be used for investigating important behavioural and brain phenomena. One possible application is the study of the psychological processes of *assimilation and accommodation* postulated by Piaget [1], as started to do by [21] and [22]. Along this line, the system might also be used to investigate the open-learning processes characterising children [83] and leading them to progressively acquire a repertoire of increasingly sophisticated skills (as mentioned in Section I, this was one main motivation for designing TERL). In this respect, it would be interesting to endow the system with the autonomous capacities of self-generating goals and tasks, as mentioned above, and focusing on those that are learned with the highest learning rate [84], [85], [86], [87], [82].

TERL can also be used to study psychophysical issues related to human motor learning, in particular problems related to how the motor system transfers knowledge between similar motor tasks. For example, the seminal work of [88] showed that learning and transfer of reaching movements are strongly facilitated when different tasks are learned in interleaved random trials rather than in sequential whole blocks of trials each focused on different tasks. Other studies have investigated how humans generalise a newly learned reaching skill, acquired to reach a given target point while compensating a disturbing force field, to other target points laying on a circumference centred on the (fixed) starting point [89], [90], [91]. These studies show how transfer and generalisation benefits only tasks involving target points that are very close in space. The architecture and transfer capabilities of TERL seem ideally suited to investigate these phenomena because they support transfer learning between similar tasks, and these transfer processes are parameterised under many respect, and because they can be linked to the architecture and functioning of the brain (see below; see also [92], for a review on experiments, models, and possible approaches to investigate these issues).

Some aspects of the architecture and functioning of TERL have been inspired by the brain organisation and so they might be leveraged to model and investigate some open issues in neuroscience. In general, reinforcement learning algorithms are suitable to study the organisation and learning mechanisms of hierarchical architecture [93]. Within a biological perspective, TERL architecture might be used to model basal ganglia-cortical loops [94], [95], a brain hierarchical system playing a key role in trial-and-error learning in organisms [96], [48], [97], [98]. For example, following previous works [18], [99], the capacity of TERL gating networks to assign different tasks/goals to different experts might be used to model and study he mechanisms with which basal ganglia and cortex form separated loops and channels dedicated to different sensory inputs, actuator outputs, and input-output mappings [96], [100].

The capacity of TERL to form copies of experts on the basis of reinforcement learning processes could be used to model the duplication of neural modules in the brain [41], a process that has been obtained through evolutionary algorithms mimicking DNA-based neural duplication [73], [44] and has also been previously observed in other systems [70], [72], [42]. In this respects, we are not aware of previous systems regulating the duplication process during learning on the basis of dedicated mechanisms such as the ranking and learning-responsibility mechanisms described here.

APPENDIX

Parameters setting: The parameters of the model were set as indicated in Table II.

TABLE II
PARAMETERS OF THE MODEL USED TO CONTROL THE 2D AND THE 3D ROBOTIC ARM. THE TABLE REPORTS IN PARENTHESES THE PARAMETERS USED WITH THE 3D ARM THAT DIFFER FROM THOSE USED WITH THE 2D ARM.

Parameter	**Value**
State features	
D	21^2 (8^4)
σ_f^2	1
Responsibility accumulation process	
κ	0.01
σ_{AG}^2	0.5 (0.3)
σ_{CG}^2	0.5 (0.3)
Discount factor	
γ	0.99
Learning rates	
η^{AG}	3.0
η^{CG}	1.0
η^{A}	1.2 (2.0)
η^{C}	0.012 (0.2)
Noise generation	
ϵ	20
τ	0.01
M	100
T'	600 (800)
υ	0.95
β	0.03

ACKNOWLEDGMENT

This research has received funds from the European Commission under the 7th Framework Programme (FP7/2007-2013), ICT Challenge 2 ‘Cognitive Systems and Robotics’, project ‘IM-CLeVeR - Intrinsically Motivated Cumulative Learning Versatile Robots’, Grant Agreement number ICT-IP-231722.

REFERENCES

[1] J. Piaget, *The Origins of Intelligence in Children*. London: Routledge and Kegan Paul, 1953.

[2] N. E. Berthier and R. Keen, “Development of reaching in infancy.” *Experimental Brain Research*, vol. 169, pp. 507–518, 2006.

[3] H. Forssberg, A. C. Eliasson, H. Kinoshita, R. S. Johansson, and G. Westling, “Development of human precision grip. i: Basic coordination of force.” *Experimental Brain Research*, vol. 85, no. 2, pp. 451–457, 1991.

[4] E. Thelen, “Rhythmical stereotypies in normal human infants,” *Animal Behaviour*, vol. 27, no. Pt 3, pp. 699–715, 1979.

[5] W. K. Geerts, C. Einspieler, J. Dibiasi, B. Garzarolli, and A. F. Bos, “Development of manipulative hand movements during the second year of life,” *Early Human Development*, vol. 75, no. 1-2, pp. 91–103, 2003.

[6] M. Taylor and P. Stone, “Transfer learning for reinforcement learning domains: A survey,” *The Journal of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009.

[7] M. McCloskey and N. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” *The psychology of learning and motivation*, vol. 24, pp. 109–165, 1989.

[8] R. M. French, “Catastrophic forgetting in connectionist networks,” *Trends in Cognitive Sciences*, vol. 3, pp. 128–135, 1999.

[9] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, “Autonomous mental development by robots and animals,” *Science*, pp. 599–600, 2001.

[10] A. G. Barto, S. Singh, and N. Chentanez, “Intrinsically motivated learning of hierarchical collections of skills,” in *International Conference on Developmental Learning (ICDL2004)*, J. Triesch and T. Jebara, Eds. New York, NY: IEEE, 2004, pp. 112–119.

[11] A. Stout, G. D. Konidaris, and A. G. Barto, “Intrinsically motivated reinforcement learning: A promising framework for developmental robot learning,” in *Proceedings of the AAAI Spring Symposium on Developmental Robotics*. Stanford, CA: Stanford University, 2005.

[12] P. Oudeyer, F. Kaplan, and V. Hafner, “Intrinsic motivation system for autonomous mental development,” *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 703–713, 2007.

[13] S. Thrun and T. Mitchell, “Lifelong robot learning,” *Robotics and autonomous systems*, vol. 15, pp. 25–46, 1995.

[14] G. Baldassarre and M. Mirolli, “What are the key open challenges for understanding the autonomous cumulative learning of skills?” The Newsletters of the Autonomous Mental Development Technical committee (IEEE CIS AMD Newsletters). Vol. 7 N. 1, pp. 11., 2010.

[15] A. Lazaric, “Transfer reinforcement learning: A framework and a survey,” in *Reinforcement Learning: State of the Art*, M. Wiering and M. van Otterlo, Eds. Berlin: Springer, 2012, pp. 143–174.

[16] J. Alcock, *Animal behavior: an evolutionary approach*, 6th ed. Sunderland, MA: Sinauer Associated, 1998.

[17] S. P. Singh, “Transfer of learning by composing solutions of elemental sequential tasks,” *Machine Learning*, vol. 8, pp. 323–339, 1992.

[18] G. Baldassarre, “A modular neural-network model of the basal ganglia’s role in learning and selecting motor behaviours,” *Journal of Cognitive Systems Research*, vol. 3, pp. 5–13, 2002.

[19] ——, “Planning with neural networks and reinforcement learning,” PhD Thesis, Computer Science Department, University of Essex, Colchester, UK, 2002.

[20] 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 *Proceedings of the Tenth International Conference on Epigenetic Robotics (EpiRob2010)*, vol. 149, Lund University, Sweden, 2010, pp. 27–34.

[21] P. Tommasino, D. Caligiore, M. Mirolli, and G. Baldassarre, “Reinforcement learning algorithms that assimilate and accommodate skills with multiple tasks,” in *IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob2012)*, J. Movellan and M. Schlesinger, Eds. New York, NY: IEEE, 2012, pp. e1–8.

[22] D. Caligiore, P. Tommasino, V. Sperati, and G. Baldassarre, "Modular and hierarchical brain organization to understand assimilation, accommodation and their relation to autism in reaching tasks: a developmental robotics hypothesis," *Adaptive Behavior*, vol. 22, no. 5, pp. 304–329, 2014.

[23] R. Sutton, "Generalization in reinforcement learning: Successful examples using sparse coarse coding," in *Advances in neural information processing systems*. Cambridge, MA: The MIT Press, 1996, pp. 1038–1044.

[24] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge MA, USA: The MIT Press, 1998.

[25] J. Kober and J. Peters, "Reinforcement learning in robots: A survey," in *Reinforcement Learning: State of the Art*, M. Wiering and M. van Otterlo, Eds. Berlin: Springer, 2012, pp. 579–610.

[26] H. van Hasselt, "Reinforcement learning in continuous state and action spaces," in *Reinforcement Learning: State of the Art*, M. Wiering and M. van Otterlo, Eds. Berlin: Springer, 2012, pp. 143–174.

[27] R. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*. Cambridge, MA: The MIT Press, 2000, pp. 1057–1063.

[28] R. C. O'Reilly, "Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm," *Neural computation*, vol. 8, no. 5, pp. 895–938, 1996.

[29] A. Barto, R. Sutton, and C. Anderson, "Neuron-like adaptive elements that can solve difficult learning control problems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, pp. 834–846, 1983.

[30] A. G. Barto, "Adaptive critics and the basal ganglia," in *Models of Information Processing in the Basal Ganglia*, J. Houk, J. Davis, and D. Beiser, Eds. Cambridge MA, USA: The MIT Press, 1995, pp. 215–232.

[31] K. Doya, "Complementary roles of basal ganglia and cerebellum in learning and motor control," *Current Opinions Neurobiology*, vol. 10, no. 6, pp. 732–739, 2000.

[32] D. Joel, Y. Niv, and E. Ruppin, "Actor-critic models of the basal ganglia: new anatomical and computational perspectives," *Neural Networks*, vol. 15, no. 4-6, pp. 535–547, 2002.

[33] M. Khamassi, L. Lacheze, B. Girard, A. Berthoz, and A. Guillot, "Actor-critic models of reinforcement learning in the basal ganglia: From natural to artificial rats," *Adaptive Behavior*, vol. 13, pp. 131–148, 2005.

[34] W. Schultz, P. Dayan, and R. P. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, pp. 1593–1599, 1997.

[35] W. Schultz, "Getting formal with dopamine and reward," *Neuron*, vol. 36, pp. 241–263, 2002.

[36] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.

[37] R. Jacobs, M. Jordan, and A. Barto, "Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks," *Cognitive Science*, vol. 15, pp. 219–250, 1991.

[38] D. McFarland, *Animal behavior*, 2nd ed. Harlow, UK: Longman Group, 1993.

[39] M. Mirolli, F. Mannella, and G. Baldassarre, "The roles of the amygdala in the affective regulation of body, brain and behaviour," *Connection Science*, vol. 22, no. 3, pp. 215–245, 2010.

[40] R. E. Passingham and S. P. Wise, *The neurobiology of the prefrontal cortex: anatomy, evolution, and the origin of insight*. Oxford: Oxford University Press, 2012, vol. 50.

[41] D. Meunier, R. Lambiotte, and E. T. Bullmore, "Modular and hierarchically modular organization of brain networks," *Frontiers in Neuroscience*, vol. 4, p. 200, 2010.

[42] M. Ring, T. Schaul, and J. Schmidhuber, "The two-dimensional organization of behavior," in *Proceedings of the International Conference on Development and Learning and Epigenetic Robotics (ICDL2011)*, A. Cangelosi, J. Triesch, I. Fasel, K. Rohlfing, F. Nori, P.-Y. Oudeyer, M. Schlesinger, and Y. Nagai, Eds. New York, NY: IEEE, 2011, pp. E1–6.

[43] M. Ring and T. Schaul, "The organization of behavior into temporal and spatial neighborhoods," in *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–6.

[44] C. Fernando, K. K. Karishma, and E. Szathmáry, "Copying and evolution of neuronal topology," *PloS one*, vol. 3, p. e3775, 2008.

[45] M. S. A. Graziano, *The intelligent movement machine: an ethological perspective on the primate motor system*. Oxford: Oxford University Press, 2009.

[46] T. N. Aflalo and M. Graziano, "Organization of the macaque extrastriate visual cortex re-examined using the principle of spatial continuity of function," *Journal of Neurophysiology*, vol. 105, pp. 305–320, 2011.

[47] K. Gurney, T. J. Prescott, and P. Redgrave, "A computational model of action selection in the basal ganglia. I. a new functional anatomy." *Biological Cybernetics*, vol. 84, pp. 401–410, 2001.

[48] A. M. Graybiel, "The basal ganglia and chunking of action repertoires." *Neurobiology of Learning and Memory*, vol. 70, no. 1-2, pp. 119–136, 1998.

[49] G. Rizzolatti and G. Luppino, "The cortical motor system." *Neuron*, vol. 31, pp. 889–901, 2001.

[50] M. S. A. Graziano, C. S. R. Taylor, T. Moore, and D. F. Cooke, "The cortical control of movement revisited," *Neuron*, vol. 36, pp. 349–362, 2002.

[51] S. Thill, D. Caligiore, A. M. Borghi, T. Ziemke, and G. Baldassarre, "Theories and computational models of affordance and mirror systems: an integrative review," *Neuroscience and Biobehavioral Reviews*, vol. 37, pp. 491–521, 2013.

[52] E. K. Miller and J. D. Cohen, "An integrative theory of prefrontal cortex function," *Annual Reviews Neuroscience*, vol. 24, pp. 167–202, 2001.

[53] J. M. Fuster, "Upper processing stages of the perception-action cycle." *Trends in Cognitive Sciences*, vol. 8, pp. 143–145, 2004.

[54] D. Caligiore, A. M. Borghi, D. Parisi, and G. Baldassarre, "TRoPICALS: A computational embodied neuroscience model of compatibility effects," *Psychological Review*, vol. 117, pp. 1188–1228, 2010.

[55] K. Doya, "Reinforcement learning in continuous time and space," *Neural Computation*, vol. 12, pp. 219–245, 2000.

[56] M. van Oterlo and M. Wiering, "Reinforcement learning and Markov decision processes," in *Reinforcement Learning: State of the Art*, M. Wiering and M. van Otterlo, Eds. Berlin: Springer, 2012, pp. 3–42.

[57] Y. Uno, M. Kawato, and R. Suzuki, "Formation and control of optimal trajectory in human multijoint arm movement. minimum torque-change model." *Biological Cybernetics*, vol. 61, no. 2, pp. 89–101, 1989.

[58] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Berlin: Springer-Verlag, 2000.

[59] D. Caligiore, E. Guglielmelli, D. Parisi, and G. Baldassarre, "A reinforcement learning model of reaching integrating kinematic and dynamic control in a simulated arm robot," in *IEEE International Conference on Development and Learning (ICDL2010)*. New York, NY: IEEE, 2010, pp. 211–218.

[60] D. Caligiore, D. Parisi, and G. Baldassarre, "Integrating reinforcement learning, equilibrium points and minimum variance to understand the development of reaching: A computational model," *Psychological Review*, vol. 121, no. 3, pp. 389–421, 2014.

[61] G. Sandini, G. Metta, and D. Vernon, "The iCub cognitive humanoid robot: an open-system research platform for enactive cognition," in *50 Years of Artificial Intelligence. Essays Dedicated to the 50th Anniversary of Artificial Intelligence*, M. Lungarella, F. Iida, and R. Bongard J.and Pfeifer, Eds. Berlin: Springer-Verlag, 2007, vol. 4850.

[62] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete Event Dynamic Systems*, vol. 13, pp. 341–379, 2003.

[63] R. S. Sutton, D. Precup, and S. P. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[64] G. Konidaris and A. Barto, "Building portable options: Skill transfer in reinforcement learning," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. Cambridge, MA: The MIT Press, 2007, vol. 7, pp. 895–900.

[65] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.

[66] R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, and D. Precup, "Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction," in *The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS2011)*, 2011, pp. 761–768, 26 May 2011, Taipei Taiwan.

[67] F. Fernández and M. Veloso, "Probabilistic policy reuse in a reinforcement learning agent," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 2006, pp. 720–727.

[68] A. Lazaric, M. Restelli, and A. Bonarini, "Transfer of samples in batch reinforcement learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 544–551.

[69] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural Networks*, vol. 11, pp. 1317–1329, 1998.

[70] K. Doya, K. Samejima, K.-I. Katagiri, and M. Kawato, "Multiple model-based reinforcement learning," *Neural Computation*, vol. 14, pp. 1347–1369, 2002.

[71] N. Sugimoto, M. Haruno, K. Doya, and M. Kawato, "Mosaic for multiple-reward environments," *Neural Computation*, vol. 24, pp. 577–606, 2012.

[72] M. Ring and T. Schaul, "Q-error as a selection mechanism in modular reinforcement-learning systems," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI2011)*, 2011, vol. 22, no. 1, pp. 1452–1458.

[73] R. Calabretta, S. Nolfi, D. Parisi, and G. P. Wagner, "Duplication of modules facilitates the evolution of functional specialization," *Artificial Life*, vol. 6, pp. 69–84, 2000.

[74] C. Fernando, "Design for a darwinian brain: Part 1. philosophy and neuroscience," arXiv, arXiv preprint 1303.7200, 2013.

[75] C. Fernando and V. Vasas, "Design for a darwinian brain: Part 2. cognitive architecture," arXiv, arXiv preprint 1303.7201, 2013.

[76] N. Mehta, S. Natarajan, P. Tadepalli, and A. Fern, "Transfer in variable-reward hierarchical reinforcement learning," *Machine Learning*, vol. 73, pp. 289–312, 2008.

[77] B. Castro da Silva, G. Konidaris, and A. G. Barto, "Learning parameterized skills," in *Proceedings of the 29th International Conference on Machine Learning (ICML2012)*, J. Langford and J. Pineau, Eds. New York, NY: Omnipress, 2012, pp. 1679–1686.

[78] B. Castro da Silva, G. Baldassarre, G. Konidaris, and A. Barto, "Learning parameterized motor skills on a humanoid robot," in *IEEE International Conference on Robotics and Automation (ICRA2014)*, N. Xi, W. R. Hamel, J. Tan, V. N. Krovi, M. Buss, I. Elhajj, and L.-C. Fu, Eds. New York, NY: IEEE, 2014, pp. e1–6.

[79] A. Wilson, A. Fern, S. Ray, and P. Tadepalli, "Multi-task reinforcement learning: a hierarchical bayesian approach," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 1015–1022.

[80] R. Jacobs, "Mixture models," Department of Brain & Cognitive Sciences, University of Rochester, Rochester, NY 14627, USA, Technical report, 2008.

[81] 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 *The Fourth Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob2014)*, G. Metta, M. Lee, and I. Fasel, Eds. New York, NY: IEEE, 2014, pp. 322–327, 13–16 October, Genoa, Italy.

[82] ——, "Grail: a goal-discovering robotic architecture for intrinsically-motivated learning," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 9, 2016.

[83] G. Baldassarre and M. Mirolli, Eds., *Intrinsically motivated learning in natural and artificial systems*. Berlin: Springer, 2013.

[84] A. McGovern and A. Barto, "Automatic discovery of subgoals in reinforcement learning using diverse density," Computer Science Department, Faculty Publication Series, 2001.

[85] G. D. Konidaris and A. G. Barto, "Skill discovery in continuous reinforcement learning domains using skill chaining," in *Advances in Neural Information Processing Systems 22 (NIPS09)*, Y. e. a. Bengio, Ed., 2009, pp. 1015–1023.

[86] V. G. Santucci, G. Baldassarre, and M. Mirolli, "Intrinsic motivation mechanisms for competence acquisition," in *IEEE International Conference on Development and Learning-EpiRob 2012 (ICDL-EpiRob-2012)*, J. Movellan and M. Schlesinger, Eds. New York, NY: IEEE, 2012, pp. e1–6.

[87] ——, "Which is the best intrinsic motivation signal for learning multiple skills?" *Frontiers in Neurorobotics*, vol. 7, pp. e1–14, 2013.

[88] J. B. Shea and R. L. Morgan, "Contextual interference effects on the acquisition, retention, and transfer of a motor skill," *Journal of Experimental Psychology: Human Learning and Memory*, vol. 5, no. 2, pp. 179–187, 1979.

[89] R. Shadmehr and F. A. Mussa-Ivaldi, "Adaptive representation of dynamics during learning of a motor task," *Journal of Neuroscience*, vol. 14, pp. 3208–3224, 1994.

[90] R. Shadmehr and S. P. Wise, Eds., *The Computational Neurobiology of Reaching and Pointing*. Cambridge, MA: The MIT Press, 2005.

[91] R. Shadmehr and S. Mussa-Ivaldi, *Biological Learning and Control: How the Brain Builds Representations, Predicts Events, and Makes Decisions*. Cambridge, MA: The MIT Press, 2012.

[92] L. Lonini, C. Dimitrakakis, C. A. Rothkopf, and J. Triesch, "Generalization and interference in human motor control," in *Computational and Robotic Models of the Hierarchical Organisation of Behaviour*, G. Baldassarre and M. Mirolli, Eds. Berlin: Springer-Verlag, 2013, pp. 155–176.

[93] G. Baldassarre and M. Mirolli, Eds. Berlin: Springer-Verlag, 2013.

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

[95] V. G. Fiore, V. Sperati, F. Mannella, M. Mirolli, K. Gurney, K. Firston, R. J. Dolan, and G. Baldassarre, "Keep focussing: striatal dopamine multiple functions resolved in a single mechanism tested in a simulated humanoid robot," *Frontiers in Psychology – Cognitive Science*, vol. 5, no. 124, pp. e1–17, 2014.

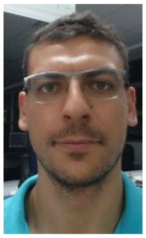
[96] G. E. Alexander, M. R. DeLong, and P. L. Strick, "Parallel organization of functionally segregated circuits linking basal ganglia and cortex," *Annual Review of Neuroscience*, vol. 9, pp. 357–381, 1986.

[97] H. Yin and B. Knowlton, "The role of the basal ganglia in habit formation," *Nature Review Neuroscience*, vol. 7, no. 6, pp. 464–476, 2006.

[98] M. M. Botvinick, Y. Niv, and A. Barto, "Hierarchically organized behavior and its neural foundations: A reinforcement-learning perspective," *Cognition*, vol. 113, no. 3, pp. 262–280, 2008.

[99] M. Khamassi, L. Lachèze, B. Girard, A. Berthoz, and A. Guillot, "Actor–critic models of reinforcement learning in the basal ganglia: from natural to artificial rats," *Adaptive Behavior*, vol. 13, no. 2, pp. 131–148, 2005.

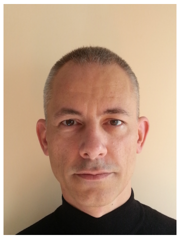
[100] J. W. Mink, "The basal ganglia: focused selection and inhibition of competing motor programs," *Progress in Neurobiology*, vol. 50, no. 4, pp. 381–425, 1996.



Paolo Tommasino received the B.S. and M.S. degrees in Biomedical Engineering at the Universita' Campus Bio-Medico, Roma, Italy, in 2008 and 2011, respectively. From May 2011 to May 2012, he was a Research Fellow at the Laboratory of Computational Embodied Neuroscience (LOCEN), Institute of Cognitive Science and Technologies, Rome, where he was involved in the project founded by the European Commission IM-CLeVeR - Intrinsically Motivated Cumulative Learning Versatile Robots. Since August 2012, he joined the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore where he is currently a PhD Candidate. His research interests include the areas of mechatronic technologies for the study of human motor control, novel robotic devices for rehabilitation and motor therapy after neurological injury, bio-inspired algorithms for controlling redundant manipulators, bio-inspired actuators, reinforcement and Hebbian learning, chaotic neural networks and reservoir computing.

Daniele Caligiore (http://www.istc.cnr.it/people/daniele-caligiore) received a Master Degree in Electronics Engineering at the University of Catania (Italy) in 2003, and a PhD in Biomedical Engineering at the University Campus Bio-Medico di Roma (Italy) in 2011. During his PhD he was visiting scholar at the Centre for Robotics and Neural Systems and at the School of Psychology (University of Plymouth, UK) and at Embodied Cognition Lab (Universita' of Bologna, Italy). Since 2004 he works as a researcher with the Institute of Cognitive Sciences and Technologies, National Research Council (Italy). He has participated to several European projects in the field of embodied cognition and developmental robotics: MindRACES – from Reactive to Anticipatory Cognitive Embodied Systems; ROSSI – Emergence of communication in RObots through Sensorimotor and Social Interaction; IM-CLeVeR – Intrinsically Motivated Cumulative Learning Versatile Robots. His research interests include developmental robotics, embodied cognition, system-level computational neuroscience, brain cortical and sub-cortical hierarchies; reinforcement learning. He has authored/co-authored about 70 peer-reviewed publications appeared in international journals, books and conference proceedings. Recently, he was guest-editor for a consensus paper of the journal 'Cerebellum' titled 'Towards a systems-level view of cerebellar function: the interplay between cerebellum, basal ganglia and cortex'.

Marco Mirolli (http://www.istc.cnr.it/people/marco-mirolli) is a Researcher at the Institute of Cognitive Sciences and Technologies, CNR, Rome. He holds a degree in Philosophy (Siena, 2001) and a Ph.D. in Cognitive Sciences (Siena, 2006). He has co-edited 4 books, including ”Intrinsically Motivated Learning in Natural and Artificial Systems” (Springer) and published more than 60 peer-reviewed papers. Throughout his career he has been (mostly) studying brain and behaviour through computer simulations and robotic models. In particular, he has been working 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. Currently, his main interest is in understanding the relationships between the Body and the Mind through theoretical analysis, computational modeling, and empirical research.



Gianluca Baldassarre (http://www.istc.cnr.it/people/gianluca-baldassarre) got a BA and MA in Economics in 1998 and a MSc in Cognitive Psychology and Neural Networks at the University of Rome 'La Sapienza' in 1999. Then he got a PhD in Computer Science at the University of Essex, UK (research on 'Planning with Neural Networks') in 2003. He successively did a postdoc at the Italian Institute of Cognitive Sciences and Technologies, National Research Council, Rome, working on swarm robotics until 2005. From 2006 he is a Researcher at this same Institute and coordinates the Research Group called 'LOCEN – Laboratory of Computational Embodied Neuroscience'. In 2006-2009 he was Team Leader of the EU project 'ICEA – Integrating Cognition Emotion and Autonomy', in 2009-2013 he was the Coordinator of the European Integrated Project 'IM-CLeVeR – Intrinsically-Motivated Cumulative-Learning Versatile Robots', in 2016-2020 he is being the Coordinator of the European FET-OPEN project 'GOAL-Robots – Goal-based Open-ended Autonomous Learning Robots'. His research interests are on cumulative learning of multiple sensorimotor skills driven by extrinsic and intrinsic motivations, and on higher-level cognition grounded on sensorimotor behaviour with a particular focus on goal-directed behaviour. He studies these topics with LOCEN by following two interdisciplinary research approaches: (a) through computational models aiming to understand the functioning of brain and behaviour; (b) through machine-learning/autonomous-robotics approaches aiming to produce technologically useful robots. He has over 100 international peer-review publications.