Sao Mai Nguyen and Pierre-Yves Oudeyer. Interactive Learning Gives the Tempo to an Intrinsically Motivated Robot Learner, in Proceedings of the IEEE-RAS International Conference on Humanoid Robots, 2012, Osaka, Japan.

Interactive Learning Gives the Tempo to an Intrinsically Motivated Robot Learner

Sao Mai Nguyen and Pierre-Yves Oudeyer
Flowers Team, INRIA and ENSTA ParisTech, France
Email: nguyensmai at gmail.com, pierre-yves.oudeyer at inria.fr

***Abstract*—This paper studies an interactive learning system that couples internally guided learning and social interaction for robot learning of motor skills. We present Socially Guided Intrinsic Motivation with Interactive learning at the Meta level (SGIM-IM), an algorithm for learning forward and inverse models in high-dimensional, continuous and non-preset environments. The robot actively self-determines: at a meta level a strategy, whether to choose active autonomous learning or social learning strategies; and at the task level a goal task in autonomous exploration. We illustrate through 2 experimental set-ups that SGIM-IM efficiently combines the advantages of social learning and intrinsic motivation to be able to produce a wide range of effects in the environment, and develop precise control policies in large spaces, while minimising its reliance on the teacher, and offering a flexible interaction framework with humans.**

I. Introduction

A. Combining intrinsic motivation and social learning

Humanoid robots, similarly to animal or human infants, need learning mechanisms which enable them to control their numerous degrees of freedom in new activities and situations [1], [2] in order to adapt to their changing environment and their interaction with humans. Thus, constraints of time and resources makes the choice of tasks to learn and methods to adopt crucial. Exploration strategies developed in the recent years can be classified into two broad interacting families: 1) socially guided exploration [3]–[5]; 2) internally guided exploration and in particular intrinsic motivation [1], [6] for meta-exploration mechanisms monitoring the evolution of learning performances [7]–[9].

Intrinsic motivation and socially guided learning are often studied separately in developmental robotics. Indeed, many forms of socially guided learning can be seen as extrinsically driven learning. Yet, in the daily life of humans, the two strongly interact, and their combination could on the contrary push their respective limits (cf. table I).

	Intrinsically Motivated Exploration	Socially Guided Exploration
Pros	Independent from human, broad task repertoire	transfer knowledge from human to robot
Cons	High-dimensionality, unboundedness	Teacher's patience & ambiguous input, correspondence problem

TABLE I: Advantages and disadvantages of the two exploration strategies.

Social guidance can drive a learner into new intrinsically motivating spaces or activities which it may continue to explore alone and for their own sake, but might have discovered only due to social guidance. Robots may acquire new strategies for achieving intrinsically motivated activities while observing examples. One might either search in the neighbourhood of the good example, or eliminate from the search space the bad example.

Conversely, as learning that depends highly on the teacher is limited by ambiguous human input or the correspondence problem [10], and turn out to be too time-consuming. For example, while self-exploration fosters a broader task repertoire of skills, exploration guided by a human teacher tends to be more specialised, resulting in fewer tasks that are learnt faster. Combining both can thus bring out a system that acquires a wide range of knowledge which is necessary to scaffold future learning with a human teacher on specifically needed tasks, as proposed in [11]–[13].

Social learning has been combined with reinforcement learning [12], [13] to complete one task. However, we would like a system that learns not only for a single goal, but for a continuous field of goals, or every goals in the task space.

Such a multi-goal system has been presented in [11], [14], where unfortunately the representation of the environment and actions is symbolic and discrete in a limited and preset world, with few primitive actions possible. We would like to address the problem of multi-task learning in a complex, high-dimensional and continuous environment. A method for generalising movements for several tasks was proposed in [15], [16]. However, the example and target tasks are determined by the human engineer, and the learner assumes that it is sufficient to reshape the global movement instead of learning the whole movement. On the contrary, we learn to complete more different tasks, might they require different movements. Moreover, instead of relying entirely on the teacher, the SGIM-IM also actively determines which task is interesting to focus on, to better generalise for similar tasks.

B. Formalisation

Let us set the framework of our robot learning in such an environment. The agent can complete tasks parameterised by parameters $\tau \in T$, by carrying out policies π_θ, parameterised by $\theta \in \mathbf{R}^n$:

$$\begin{aligned} \pi_\theta : A &\rightarrow [0,1] \\ a &\mapsto \pi_\theta(a) \end{aligned}$$

which associates to an action a the probability that a is the right action to perform. The performance of a policy π_θ at completing a task τ is measured by:

$$\begin{aligned} J : T \times \mathbf{R}^n &\rightarrow [0, +\infty[\\ (\tau, \theta) &\mapsto J(\tau, \theta) \end{aligned} \tag{1}$$

We define a $skill$ as the function that maps to a task τ the best policy to complete it:

$$\begin{aligned} S : T &\rightarrow \mathbf{R}^d \\ \tau &\mapsto argmax_\theta J(\tau, \theta) \end{aligned}$$

The aim of the agent is to find the right policy to complete every task τ to maximise

$$I = \int_\tau P(\tau) J(\tau, S(\tau)) d\tau \tag{2}$$

where $P(\tau)$ is a probability density distribution over T. A priori unknown to the learner, $P(\tau)$ can describe the probability of τ occurring or the reachable space or a region of interest.

We assume that T can be partitioned into subspaces where the tasks are related, and in these subspaces our parametrisation allows a smooth variation of J with respect to τ most of the time, i.e. that S is a piecewise continuous function.

Our learner improves its skill S to maximise $I = \int_\tau P(\tau) J(\tau, S(\tau)) d\tau$ both by self-exploring the policy and task spaces and by asking for help to a teacher, who performs a trajectory ζ_d and completes a task τ_d.

Note that the observed trajectory might be impossible to the learner to reexecute, and he can only approach it best with a policy π_{θ_d}. We have also described our method without specifying a particular choice of learning algorithm or action, task or policy representation. These designs can indeed be decided according to the application at hand.

C. *Air Hockey Experimental Setup*

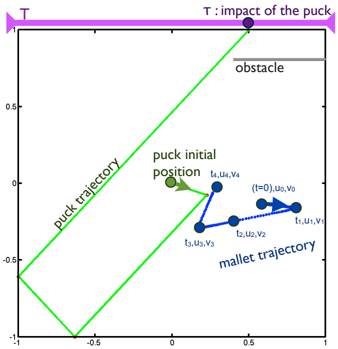


Fig. 1: Air Hockey Table: the task space is defined as the top border of the square. The puck moves in straight line without friction until it hits either the mallet, the table borders or the obstacle placed on the right.

1) Description of the Environment: Let us illustrate this formalisation with an example of a simulated square air hockey table that contains an obstacle (fig. 1). Always starting with the same position and velocity, the puck moves in straight line without friction. The effect is the position of the impact when the puck collides with the top border of the table. T is thus the top border of the table, mapped into the $[-1, 1]$ segment, where the subregion hidden by the obstacle is difficult to reach.

We control the mallet with a parameterised trajectory determined by 5 key positions $\theta_0, \theta_1, \theta_2, \theta_3, \theta_4 \in [-1, 1]^2$ at times $t_0 = 0 < t_1 < t_2 < t_3 < t_4$. The executed trajectory is generated by Gaussian distance weighting:

$$\zeta(\mathbf{t}) = \sum_{i=0}^{5} \frac{w_i(\mathbf{t})\theta_i}{\sum_{j=0}^{5} w_j(\mathbf{t})} \text{ with } w_i(\mathbf{t}) = e^{\sigma * |\mathbf{t} - \mathbf{t_i}|^2}, \sigma > 0 \tag{3}$$

Therefore, the policy parameter space $\mathbf{R}^n$ is of dimension $n = 14$ and T of dimension 1. The learner maps which trajectory of the mallet with parameter $\theta = (\theta_1, ..., \theta_{14})$ induces a collision with the top border at position τ. This is an inverse model of a highly redundant mapping, which is all the more interesting than the obstacle introduces discontinuities in the model.

2) Demonstrations and Evaluation: We can simulate a teacher by using the learning data (ζ_d, τ_d) taken from Random and intrinsically motivated learner based on SAGG-RIAC algorithm [17] as detailed later on. We choose 500 demonstrations so that τ_d is evenly distributed in $[0.5, 1]$. The teacher is thus specialised in a restricted domain of T. The demonstrations of that batch are given to the learner in a random order.

We assess our agent by measuring how close it can reach a benchmark set that defines the user's region of interest. In this case, the benchmark set is distributed over $T = [-1, 1]$ and placed every 0.05, to get the mean error at reaching these benchmark points.

D. *Interactive Learning*

In initial work to address multi-task learning in a continuous task space in the case of a complex, high-dimensional and continuous environment, we proposed in [18] the Socially Guided Intrinsic Motivation by Demonstration (SGIM-D) algorithm, where the agent learns by demonstration every M actions he performs, and otherwise learns by the SAGG-RIAC algorithm (cf. III-A). The SGIM-D could benefit from both strategies to explore the task space and the policy space as studied in [19]. Fig.2 plots at different stages of the learning, the mean error of the agent at reaching all the points of the benchmark set, with respect to M. The performance of the SGIM-D learner for the air hockey game depends on the period M of the demonstrations. Actually, the SGIM-D learner is passive with respect to the social interaction and the teacher, and does not optimise the timing of the interactions with the teacher. A mechanism that tunes the parameter M, and manages actively its interaction with the teacher can improve the performance of SGIM-D.

Indeed, the interaction between the learning agent and the teacher can be described as the way intentional or unintentional information flows from the human and the robot f_{HR}, and from the robot to the human f_{RH}. The

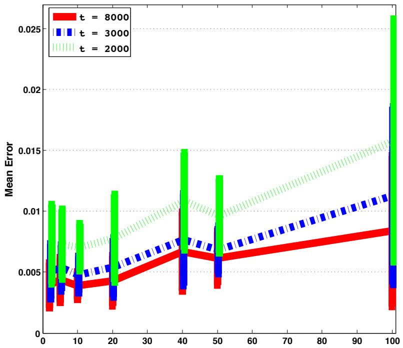


Fig. 2: Mean error of SGIM-D at reaching all the points of the benchmark set, with respect to the period of the demonstrations M. We plotted it for different stages of the learning, with the standard deviation.

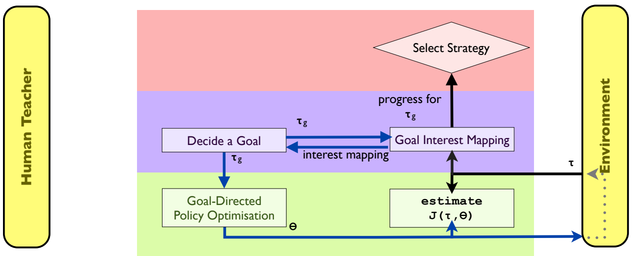


Fig. 3: Data Flow under the Intrinsic Motivation strategy

SGIM-D learner only took advantage of f_{HR}, and neglected any intentional communication from the robot to the human. However, an interactive learner who not only listens to the teacher, but actively requests for the information it needs and when it needs help, has been shown to be a fundamental aspect of social learning [14], [20], [21].

Under the interactive learning approach, the robot interacts with the user, combining learning by demonstration, learning by exploration and tutor guidance. Approaches have considered extra reinforcement signals [11], action requests [13], [22] or disambiguation among actions [20]. In [23] the comparison between a robot that has the option to ask the user for feedback, to the passive robot, show a better accuracy and fewer demonstrations. Therefore, requesting demonstrations when it is needed can lessen the dependence on the teacher and reduce the quantity of demonstrations required. This approach is the most beneficial to the learner, for the information arrives as it needs them, and to the teacher who does not need to monitor the learning process.

This is why we design an interactive learning algorithm with an intrinsically motivated robot learner, which decides itself when it is most beneficial to imitate the teacher. We first describe the design of our **SGIM-IM** (Socially Guided Intrinsic Motivation with Interactive learning at the Meta level) algorithm, which actively chooses the best learning strategy between intrinsically motivated exploration and imitation learning. Then we show that SGIM-IM efficiently requests for the teacher's demonstrations to complete a wide range of tasks, while being specialised in specific subspaces through 2 experimental setups: an air hockey game and a fishing skill learning.

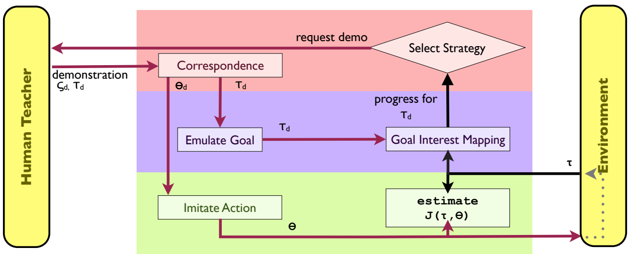


Fig. 4: Data Flow under the Social Learning strategy

II. SGIM-IM Algorithm

SGIM-IM (Socially Guided Intrinsic Motivation with Interactive learning at the Meta level) is an algorithm that merges interactive learning as social interaction, with the SAGG-RIAC algorithm of intrinsic motivation [17], to learn local inverse and forward models in complex, redundant, high-dimensional and continuous spaces.

A. SGIM-IM Overview

SGIM-IM learns by episodes during which it chooses actively which learning strategy to opt for each episode, between intrinsically motivated or social learning exploration.

In an episode under the intrinsic motivation strategy (fig. 3), it explores autonomously following the SAGG-RIAC algorithm [17]. It actively self-generates a goal τ_g where its competence improvement is maximal, then explores which policy π_θ can achieve τ_g best. The SGIM-IM learner explores preferentially goal tasks easy to reach and where it makes progress the fastest. It tries different policies to approach the self-determined task τ_g, re-using and optimising the estimation of J built through its past autonomous and socially guided explorations. The episode ends after a fixed duration

In an episode under the social learning strategy (fig. 4), our SGIM-IM learner observes the demonstration $[\zeta_d, \tau_d]$, memorises this task τ_d as a possible goal, and mimics the teacher by performing policies π_θ to reproduce ζ_d, for a fixed duration. This strategy highlights useful tasks, and teaches the learner at least one way to complete a new task, whereas self-exploration has low chance of discovering useful tasks.

The SGIM-IM learner actively decides on a meta level which strategy to choose according to the recent learning progress enabled by each strategy. If it has recently made the most progress in the intrinsic motivation strategy, it prefers exploring autonomously. Conversely, if the demonstrations do not enable him to make higher progresses than by autonomous learning (limited teacher, or inappropriate teacher) it would prefer autonomous exploration.

Its architecture is separated into three layers (fig. 5), that we describe in the following paragraphs. For the parts, which are common to SGIM-D, please refer to [18] for more details.

B. Task Space Exploration

This level of active learning drives the exploration of the task space. With the autonomous learning strategy, it sets goals τ_g depending on the interest level of previous goals (*Decide a Goal*). With the social learning strategy, it retrieves from the teacher information about demonstrated effects τ_d

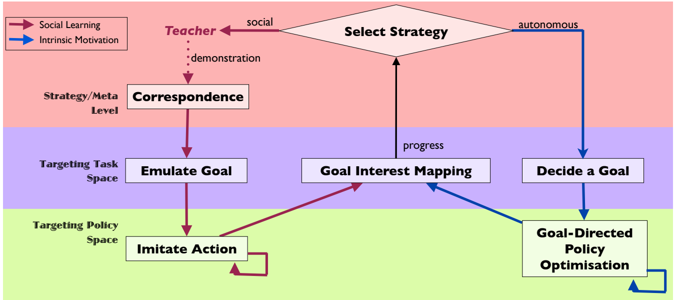


Fig. 5: Time flow chart of SGIM-IM, which combines Intrinsic Motivation and Social Learning into 3 layers that pertain to the human-machine interface, the task space exploration and the action space exploration respectively.

Algorithm II.1 SGIM-IM

```
Initialization: $\mathcal{R} \leftarrow$ singleton $T$
Initialization: $flagInteraction \leftarrow false$
Initialization: $Memo \leftarrow$ empty episodic memory
Initialization: $\Delta_S$ : progress values made by social learning
Initialization: $\Delta_A$ : progress values made by intrinsic motivation learning
loop
    $flagInteraction \leftarrow$ Select Strategy$(pref_S, pref_A)$
    if $flagInteraction$ then
        Social Learning Regime
        demo $\leftarrow$ ask and perceive demonstration
        $(\theta_d, \tau_d) \leftarrow$ Correspondence (demo)
        Emulate Goal: $\tau_g \leftarrow \tau_d$
        $\gamma_i \leftarrow$ Competence for $\tau_g$
        Memo $\leftarrow$ Imitate Action$(\theta_d)$
        $\gamma \leftarrow$ Competence for $\theta_g$
        Add $\gamma - \gamma_i$ to stack $\Delta_S$
    else
        Intrinsic Motivation Regime
        $\tau_g \leftarrow$ Decide a goal$(\mathcal{R})$
        $\gamma_i \leftarrow$ Competence for $\tau_g$
        repeat
            Memo $\leftarrow$ Goal-Directed Policy Optimisation$(\tau_g)$
        until Terminate reaching of $\tau_g$
        $\gamma \leftarrow$ Competence for $\tau_g$
        Add $\gamma - \gamma_i$ to stack $\Delta_A$
    end if
    $\mathcal{R} \leftarrow$ Update Goal Interest Mapping$(\mathcal{R}, Memo, \tau_g)$
end loop
```

(*Emulate a Goal*). Then, it maps T in terms of interest level (*Goal Interest Mapping*).

1) Goal Interest Mapping: T is partitioned according to interest levels. For each task τ explored, it assigns a competence γ_τ which evaluates how close it can reach τ: $\gamma_\tau = max_{(\theta \in Memo)} J(\tau, \theta)$ where $Memo$ is the list of all the policy parameters experienced in the past. A high value of γ_τ represents a system competent at reaching the goal y_g.

T is incrementally partitioned into areas of different size so as to maximise the difference in competence progress, as described in [24]. For a region $R_i \subset T$, we compute the interest as *the local competence progress, over a sliding time window of the δ most recent goals attempted inside R_i*:

$$interest_i = \frac{\left| \left(\sum_{j=|R_i|-\delta}^{|R_i|-\frac{\delta}{2}} \gamma_j \right) - \left(\sum_{j=|R_i|-\frac{\delta}{2}}^{|R_i|} \gamma_j \right) \right|}{\delta} \tag{4}$$

2) Decide a Goal: This function uses the interest level mapping to decide which goal is interesting to focus on. It stochastically chooses effects in regions for which its empirical evaluation of learning progress is maximal.

C. Action Space Exploration

This lower level of learning explores the policy parameters space $\mathbf{R}^d$ to build an action repertoire and local models. With the social learning strategy, it imitates the demonstrated actions ζ_d (*Imitate an Action*), while during self-exploration, the *Goal-Directed Policy Optimisation* function attempts to reach the goals τ_g set by the *Task Space Exploration* level, then, it returns the measure of competence at reaching τ_d or τ_g.

1) Imitate an Action: This function tries to imitate the teacher with policy parameters $\theta_{im} = \theta_d + \theta_{rand}$ with a random movement parameter variation $|\theta_{rand}| < \epsilon$ and π_{θ_d} the closest policy to reproduce the demonstration. After a short fixed number of times, SGIM-IM computes its competence at reaching the goal indicated by the teacher τ_d.

2) Goal-Directed Policy Optimisation: This function searches for policies π_θ that guide the system toward the goal $\tau_g$by 1) building local models during exploration that can be re-used for later goals and 2) optimising actions to reach for the current goal. In the experiments below, SGIM-IM uses uses locally weighted regression in order to infer the motor policy parameters corresponding to a given novel parametrized task, and based on the previously learnt correspondences between policy and task parameters. Policy parameters are learned using local optimisation with the Nelder-Mead simplex algorithm [25] and global random exploration to avoid local minima, in order to build memory-based local direct and inverse models, using locally weighted learning with a gaussian kernel such that presented in [26].

D. Select Strategy

A meta level actively chooses the best strategy based on the recent progress made by each of them. For each episode, the learner measures its progress as the difference of competence at the beginning and the end of the exploration for the self-determined or the emulated goal, and adds this progress value to stacks Δ_A or Δ_S. The preference for each strategy is computed as the average on a window frame of the last ns progress values of Δ_A and Δ_S. Besides, in order to limit the reliance on the teacher, we penalise the preference for social learning with a *cost* factor. The strategies are selected stochastically with a probability proportional to their preference (cf. Algorithm II.2). Therefore, autonomous exploration is preferred if it provided highest competence progress in the recent past, while social learning is preferred only if its progress were *cost* times higher.

Algorithm II.2 [flagInter] = SelectStrategy(Δ_S, Δ_A)

```
input: $\Delta_S$ : progress values made by social learning strategy
input: $\Delta_A$ : progress values made by intrinsic motivation learning strategy
output: $flagInter$ : chosen strategy
parameter: $nbMin$ : duration of the initiation phase
parameter: $ns$ : window frame for monitoring progress
parameter: $cost$ : cost of requesting a demonstration
Initiation phase
if Social Learning and Intrinsic Motivation Regimes have not been chosen each $nbMin$ times yet then
    $ps \leftarrow 0.5$
else
    Permanent phase
    $wa \leftarrow$ average(last $ns$ elements of $\Delta_A$)
    $ws \leftarrow \frac{average(\text{last } ns \text{elements of } \Delta_S)}{cost}$
    $ps \leftarrow min(0.9, max(0.1, \frac{ws}{ws+wa}))$
end if
flagInter ← true with probability $ps$
return flagInter
```

We applied our hierarchical **SGIM-IM** algorithm with 2 layers of active learning to 2 illustration experiments.

III. AirHockey Experiment

We first apply SGIM-IM to our air hockey game, described in I-C.

A. Experimental Protocol

To assess the efficiency of SGIM-IM, we decide to compare the performance of several learning algorithms (fig. 6):

- Random exploration: throughout the experiment, the robot picks policy parameters randomly in $\mathbf{R}^d$.
- SAGG-RIAC: throughout the experiment, the robot explores autonomously, without taking into account any demonstration, and is driven by intrinsic motivation.
- SGIM-IM: interactive learning where the robot learns by actively choosing between social learning strategy or intrinsic motivation strategy.
- SGIM-D: the robot's behaviour is a mixture between Imitation learning and SAGG-RIAC. When the robot sees a new demonstration, it imitates the trajectory for a short while. Then, it resumes its autonomous exploration, until it sees a new demonstration by the teacher, which occurs every M actions experimented by the robot.

For each experiment in our air hockey setup, we let the robot perform 8000 actions in total, and evaluate its performance every 1000 actions. For the air hockey experiment, we set the parameters of SGIM-IM to: $cost = 100$ and $ns = 20$, and those of SGIM-D to M= 10 and M= 100 which are the best and worst parameters of SGIM-D according to fig.2.

B. Results

Fig.7 plots the mean distance error of the attempts to hit the border at the benchmark points, with respect to the number of actions performed by the mallet. It shows that SGIM-IM performs significantly better, and faster than Random exploration or SAGG-RIAC (t-test on the final distance error with $p < 0.05$). It divides by a factor of 10 the final error value compared to SAGG-RIAC. Moreover, its error rate is smaller since the very beginning. SGIM-IM has taken advantage of the demonstrations very fast to be able to hit the puck and place it on the top border, instead of making random movements which have little probability of hitting the puck, let alone placing it at a desired position. Its performance is close to SGIM-D with the best parameters. SGIM-IM manages to tune its percentage of social interaction so as to take most advantage of the demonstrations.

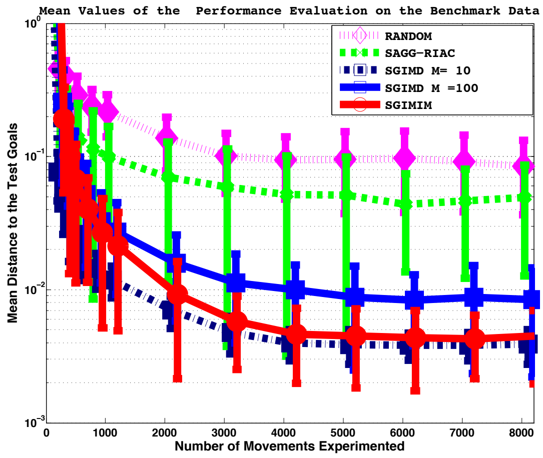


Fig. 7: Evaluation of the performance of the robot with respect to the number of actions performed, under different learning algorithms. We plotted the mean distance to the benchmark set with its standard deviation errorbar.

C. Active Choice of Strategy

As for the strategy adopted, fig.8 shows 3 phases. In the first part, demonstration requests are useful in the beginning, as each indicate to the learner which kind of actions can make the mallet hit the puck to place it in T and induce a high competence progress value. $\Delta_S \gg \Delta_A$, but autonomous learning still makes good progress. As the progress of autonomous learning decreases, the number of requests for demonstrations increase for $1500 < t < 4000$. . In the second part, the progress by the social learning strategy decreases and varies like the progress of autonomous learning. $\Delta_S \approx \Delta_A$. The bootstrapping effect enabled by demonstrations has decreased. Therefore, preference for autonomous exploration increases.

In this experimental setting, the learner can quickly improve its performance by a combination of demonstrations and autonomous exploration. When the demonstrations first bootstrap autonomous learning, then demonstrations are preferred to self-exploration and finally, as requests for demonstrations no longer help improve the robot's skill, and it prefers to improve its learning by intrinsic motivation. The SGIM-IM learner shows an improvement in both the decrease of the final error value, and the speed of learning, in this bounded and deterministic environment. Let us illustrate SGIM-IM in a stochastic environment.

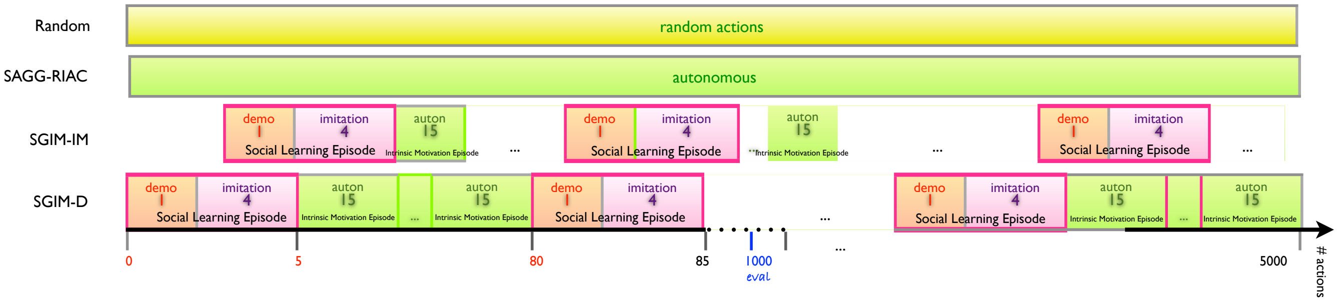


Fig. 6: Comparison of several learning algorithms. Each box represents the chronology of the adopted strategies (the figures correspond to the number of actions experimented in the episode). The figures here are given for the Fishing experiment).

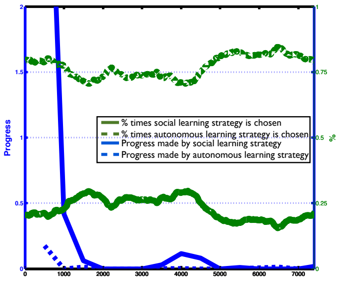


Fig. 8: 1/ Strategies chosen through time by SGIM-IM: percentage of times each strategy is chosen with respect to the number of actions performed (summed over 100 bins and averaged over several runs of SGIM-IM) 2/ The average progress made by social learning and intrinsic motivation strategies Δ_S and Δ_A

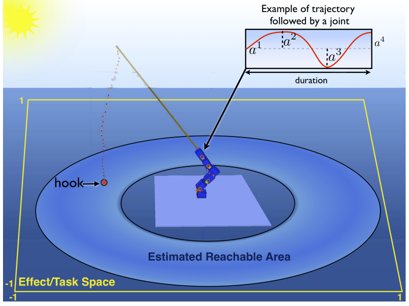


Fig. 9: Fishing experimental setup.

IV. Fishing Experiment

A. Experimental Setup

In this second experiment, we consider a simulated 6 degrees-of-freedom robotic arm holding a fishing rod (fig. 9). The aim is that it learns how to reach any point on the surface of the water with the hook at the tip of the fishing line.

$T = [-1, 1]^2$ is a 2-D space that describes the position of the hook when it reaches the water. The robot base is fixated at (0,0). The actions are parametrized motor primitives defined for each joint by 4 scalar parameters that represent the joint positions at $t = 0$, $t = \frac{\eta}{3}$,$t = \frac{2\eta}{3}$ and $t = \eta$. These 4 parameters $\theta_1, \theta_2, \theta_3, \theta_4$ generate a trajectory for the joint by Gaussian distance weighting.

Each of the 6 joints' trajectories is determined by 4 parameters. Another parameter sets τ. Therefore $\mathbf{R}^d$ is a 25-D space. The robot learns an inverse model in a continuous space, and deals with high-dimensional and highly redundant models. Our setup is all the more interesting since a fishing rod's and wire's dynamics are more difficult to model than a robotic arm inverse dynamics problem because the stochasticity distribution is hard to model as it depends on the actions dynamics. Thus learning directly the effect of one's actions is all the more advantageous. Moreover, its high-dimensionality, redundancy and stochasticity makes this simulation environment as challenging as a real robotics setup. A detailed analysis of this simulation environment can be found in [18].

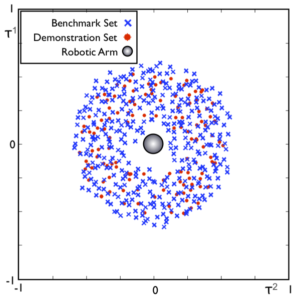


Fig. 10: Maps of the benchmark points used to assess the performance of the robot and of the demonstrated goals τ_d.

B. Experimental Protocol

1) Evaluation: After several runs of Random explorations, SAGG-RIAC and SGIM-D, we determined the apparent reachable space as the set of all the 300.000 reached points in the task space. We then tiled the reachable space into small rectangles, and generated a point randomly in each tile. Our benchmark set thus obtained is a set of 358 goal points in the task space, representative of the reachable space, and independent of the learning data used (fig. 10).

2) Demonstrations: The human teacher uses kinesthetics to teleoperate the model in a simulator with the physical robot (http://youtu.be/Ll_S-uO0kD0). The human subject is presented with a grid of points to reach on the surface of the water, and he has to place the simulator's hook nearest those points. After a habituation phase, we record the trajectories of each of the joints, and the position of the hook when touching the surface of the water. We obtained a teaching set (fig. 10) of 127 samples, that are demonstrated in a random order, and which the robot can not a-priori repeat exactly.

C. Results

1) Precision in the exploration of the reachable space: Our SGIM-IM learner parameters are set to: $cost = 2$ and

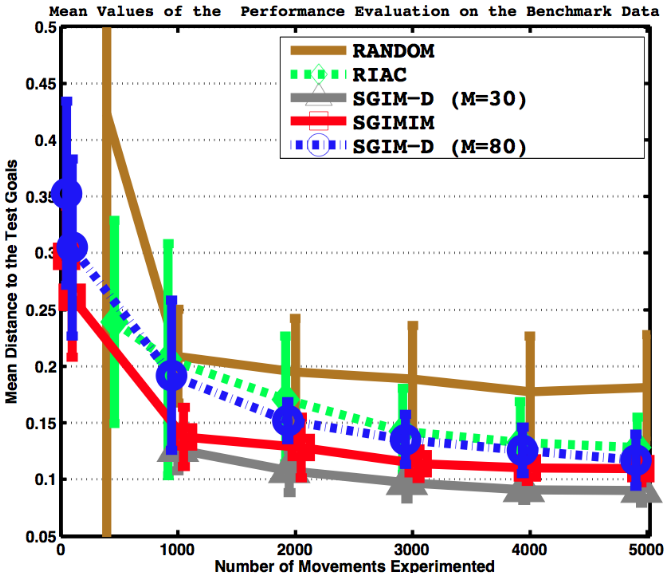


Fig. 11: Evaluation of the performance of the robot with respect to the number of actions performed, under the learning algorithms: random exploration, SAGG-RIAC, SGIM-IM, SGIM-D with a demonstration every M = 30 movements, and SGIM-D with a demonstration every M = 80 movements (to equal the total number of demonstrations of SGIM-IM). We plotted the mean distance with its standard deviation errorbar.

$ns = 15$. For every simulation on the fishing experiment setup, 5000 movements are performed. The error was assessed every 1000 movements. We examine how close the learner can get to any point of the reachable space in T, with respect to the number of actions performed by the robot (fig. 11), and with respect to the number of demonstrations given by the teacher (fig. 12b). RANDOM performs the worst, while SAGG-RIAC decreases significantly the error value compared to RANDOM (t-test with $p < 0.05$). Not only has the asymptotic performance improved, but SAGG-RIAC also learns faster from the beginning. Requesting demonstrations every 80 actions performed (SGIM-D M=80) bootstraps slightly the learning error. In this case, the social learning strategy only makes up 7% of the total time, with 61 demonstrations requests. SGIM-IM performs better than SAGG-RIAC (t-test with $p < 0.05$). The main difference lies in the beginning of the learning process, where it could take advantage of the teacher to guide him and discover the reachable space. With 52 demonstrations requested in average, SGIM-IM yet performs better than SGIM-D(M=80) with $p < 0.5$, owing to its active choice of strategy, that fits better its needs. If we increase the number of demonstrations to 162 (SGIM-D M=30), and let the robot adopt the social learning strategy 20% of the time, they indeed efficiently bootstrap the autonomous learning. SGIM-IM manages to request a fair amount of demonstrations and still obtain a performance in between the 2 SGIM-D parameters.

Not only has the error decreased, but the explored space has also increased. Fig. 12a plots the histogram of the positions of the hook $\tau \in T$ when it reaches the water. The first column shows that a natural position lies around $\tau_c = (-0.5, 0)$ in the case of random exploration : most actions map to a region around τ_c for the action space does not map linearly to the task space. As the initial position of the hook is close to the surface of the water, the robot needs to lift it with quite specific movements to throw it far away, whereas most movements would make the hook touch the water immediately, around the region of τ_c. The second column show that SAGG-RIAC has increased the explored space, and most of all, covers more uniformly the explorable space. SGIM-D and SGIM-IM emphasise the increase even further as a broader range of radius covered in the explored space.

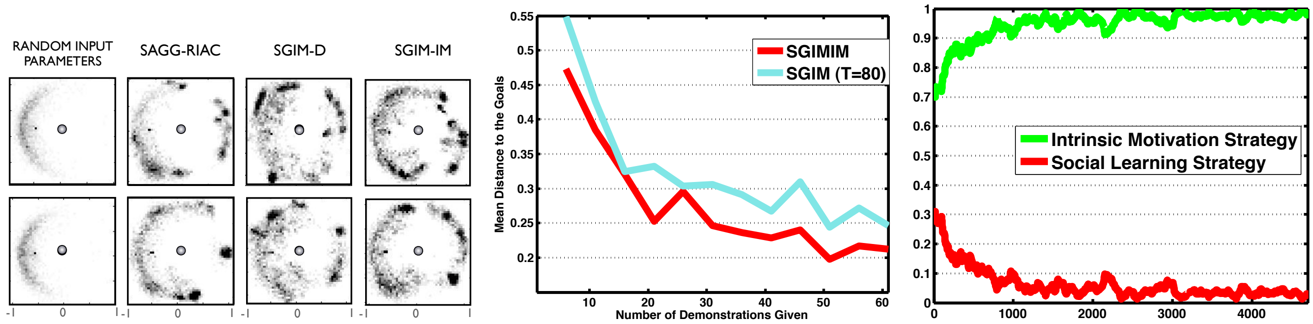
2) *Performance of the Interaction:* The simple consideration of performance with respect to time spent by the robot must be completed by considerations about the load of work for the teacher. A robot that constantly requests for help would quickly exceed the time and effort a user is ready to devote to teach. Therefore, we must examine the performance of the learner with respect to the number of the demonstrations given. Fig.12b shows that while for the first demonstrations SGIM-IM and SGIM-D(M=80) perform the same progress, a difference quickly as SGIM-IM requests fewer demonstrations. Each demonstration has a better impact on the performance of the robot, as its error plot in 12b is below the one of SGIM-D.

Indeed, fig.12c shows that the demonstrations are actively requested in the beginning of the learning process, when the demonstrations enhance the most progress by showing how to avoid the central region around τ_c. The requests then decrease as the robot acquires a good knowledge of the explorable space, and can autonomously search around the already explored localities.

In this fishing experiment, the SGIM-IM learner's active choice of learning strategy enabled it to take advantage of the teacher to request demonstrations, while carefully choosing when the teacher's demonstrations enhance the most learning progress, in order to lessen its dependence on the teacher.

V. Discussion and Conclusion

We showed through 2 illustration experiments that the Socially Guided Intrinsic Motivation with Interactive learning at the Meta level algorithm could learn to complete multiple tasks in both deterministic and stochastic environments. It can also manage the interaction with both a human teacher whose demonstrations can not be exactly reproduced by him, and a specialised teacher who only gives demonstrations in a restricted subspace of the task space. In both experiments, our robot learns efficiently and faster all possible tasks, in continuous task and action spaces. The robot could learn high-dimensional models for highly redundant problems, which constitute a typical issue for humanoid robots who evolve in continuous and unpreset environments and who have to control their numerous degrees of freedom with high redundancy. The **SGIM-IM** learner can handle its interaction with human users owing to interactive learning. It automatically balances learning by imitation and autonomous learning, by taking in account both its need and the cost of



(a) Histogram of the tasks explored by the fishing rod inside the 2D effects space. Each algorithm is illustrated by 2 example experiments.

(b) Comparison of the performance of the robot with respect to the number of actions performed, of SGIM-IM and SGIM-D (M=80)

(c) Strategy chosen though time: percentage of times each strategy is chosen with respect to the number of actions performed

Fig. 12: Analysis of the fishing experiment.

an interaction, so as to minimise the teacher's effort and maximise the impact of each demonstration. It thus offers a flexible interaction between a robot and the human users.

The Socially Guided Intrinsic Motivation with Interactive learning at the Meta level algorithm has a 3-layered hierarchical structure which includes two levels of active learning. Based on its exploration in the action space, it actively chooses in the task space which goals could be interesting to target, and selects on a meta level between autonomous learning or social learning strategies. It can actively interact with the teacher instead of being a passive system. This structure could easily be extended to take into account more complex social interaction scenarios, such as an interaction with several teachers, where the learner can choose who it should imitate. Future work will study possibilities for the robot to request for specific demonstrations (show me a specific kind of movements or show me how to complete a kind of goals). Moreover, we plan to extend this study so that humanoid robots can really evolve in our everyday environments and complete multiple tasks of different nature.

Acknowledgment

This research was partially funded by ERC Grant EXPLORERS 240007 and ANR MACSi.

References

[1] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, no. 599-600, 2001.

[2] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida, "Cognitive developmental robotics: A survey," *IEEE Trans. Autonomous Mental Development*, vol. 1, no. 1, 2009.

[3] A. Whiten, "Primate culture and social learning," *Cognitive Science*, vol. 24, no. 3, pp. 477–508, 2000.

[4] M. Tomasello and M. Carpenter, "Shared intentionality," *Developmental Science*, vol. 10, no. 1, pp. 121–125, 2007.

[5] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Handbook of Robotics*. MIT Press, 2007, no. 59, ch. Robot Programming by Demonstration.

[6] E. Deci and R. M. Ryan, *Intrinsic Motivation and self-determination in human behavior*. New York: Plenum Press, 1985.

[7] P.-Y. Oudeyer, F. Kaplan, and V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Transactions on Evolutionary Computation*, vol. 11(2), pp. pp. 265–286, 2007.

[8] J. Schmidhuber, "Formal theory of creativity, fun, and intrinsic motivation (1990-2010)," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 230–247, 2010.

[9] ——, "Curious model-building control systems," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 2, 1991, pp. 1458–1463.

[10] C. L. Nehaniv and K. Dautenhahn, *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*. Cambridge: Cambridge Univ. Press, March 2007.

[11] A. L. Thomaz and C. Breazeal, "Experiments in socially guided exploration: Lessons learned in building robots that learn with and without human teachers," *Connection Science*, vol. 20 Special Issue on Social Learning in Embodied Agents, no. 2,3, pp. 91–110, 2008.

[12] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, no. 4, pp. 682–697, 2008.

[13] M. Lopes, F. Melo, and L. Montesano, "Active learning for reward estimation in inverse reinforcement learning," in *European Conference on Machine Learning*, 2009.

[14] A. L. Thomaz, "Socially guided machine learning," Ph.D. dissertation, MIT, 5 2006.

[15] B. da Silva and G. B. A. Konidaris, "Learning parameterized skills," in *29th International Conference on Machine Learning (ICML 2012)*, 2012.

[16] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Autonomous Robots*, pp. 1–19, 2012, 10.1007/s10514-012-9290-3. [Online]. Available: http://dx.doi.org/10.1007/s10514-012-9290-3

[17] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, in press.

[18] S. M. Nguyen, A. Baranes, and P.-Y. Oudeyer, "Bootstrapping intrinsically motivated learning with human demonstrations," in *Proceedings of the IEEE International Conference on Development and Learning*, Frankfurt, Germany, 2011.

[19] S. M. Nguyen and P.-Y. Oudeyer, "Properties for efficient demonstrations to a socially guided intrinsically motivated learner," in *21st IEEE International Symposium on Robot and Human Interactive Communication*, 2012.

[20] S. Chernova and M. Veloso, "Interactive policy learning through confidence-based autonomy," *Journal of Artificial Intelligence Research*, vol. 34, 2009.

[21] M. Nicolescu and M. Mataric, "Natural methods for robot task learning: Instructive demonstrations, generalization and practice," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. ACM, 2003, pp. 241–248.

[22] D. H. Grollman and O. C. Jenkins, "Incremental learning of subtasks from unsegmented demonstration," 2010.

[23] M. Cakmak, C. Chao, and A. L. Thomaz, "Designing interactions for robot active learners," *Autonomous Mental Development, IEEE Transactions on*, vol. 2, no. 2, pp. 108–118, 2010.

[24] A. Baranes and P.-Y. Oudeyer, "Riac: Robust intrinsically motivated active learning," in *Proceedings of the IEEE International Conference on Development and Learning*, Shanghai, China, 2009.

[25] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions," *SIAM Journal of Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[26] C. Atkeson, M. Andrew, and S. Stefan, "Locally weighted learning," *AI Review*, vol. 11, pp. 11–73, April 1997.