



HAL
open science

Reducing computational cost during robot navigation and human-robot interaction with a human-inspired reinforcement learning architecture

Rémi Dromnelle, Erwan Renaudo, Mohamed Chetouani, Petros Maragos, Raja Chatila, Benoît Girard, Mehdi Khamassi

► **To cite this version:**

Rémi Dromnelle, Erwan Renaudo, Mohamed Chetouani, Petros Maragos, Raja Chatila, et al.. Reducing computational cost during robot navigation and human-robot interaction with a human-inspired reinforcement learning architecture. *International Journal of Social Robotics*, 2023, 15, pp.1297-1323. 10.1007/s12369-022-00942-6 . hal-03829879

HAL Id: hal-03829879

<https://hal.sorbonne-universite.fr/hal-03829879>

Submitted on 24 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

REDUCING COMPUTATIONAL COST DURING ROBOT NAVIGATION AND HUMAN-ROBOT INTERACTION WITH A HUMAN-INSPIRED REINFORCEMENT LEARNING ARCHITECTURE

PREPRINT OF THE PAPER PUBLISHED IN IJSR (2022), SPECIAL ISSUE ON ‘HUMAN-LIKE BEHAVIOR AND
COGNITION IN ROBOTS’

Rémi Dromnelle

Institute of Intelligent Systems and Robotics
Sorbonne Université, CNRS
Paris, France
remi.dromnelle@gmail.com

Erwan Renaudo

Intelligent and Interactive Systems Group,
Universität Innsbruck
Innsbruck, Austria
erwan.renaudo@uibk.ac.at

Mohamed Chetouani

Institute of Intelligent Systems and Robotics
Sorbonne Université, CNRS
Paris, France
mohamed.chetouani@sorbonne-universite.fr

Petros Maragos

1 Athena Research and Innovation Center
2 School of ECE, National Technical Univ. of Athens
Athens, Greece
maragos@cs.ntua.gr

Raja Chatila, Benoît Girard, Mehdi Khamassi

Institute of Intelligent Systems and Robotics
Sorbonne Université, CNRS
Paris, France
firstname.lastname@sorbonne-universite.fr

November 24, 2022

ABSTRACT

1 We present a new neuro-inspired reinforcement learning architecture for robot online learning and
2 decision-making during both social and non-social scenarios. The goal is to take inspiration from the
3 way humans dynamically and autonomously adapt their behavior according to variations in their own
4 performance while minimizing cognitive effort. Following computational neuroscience principles,
5 the architecture combines model-based (MB) and model-free (MF) reinforcement learning (RL). The
6 main novelty here consists in arbitrating with a meta-controller which selects the current learning
7 strategy according to a trade-off between efficiency and computational cost. The MB strategy, which
8 builds a model of the long-term effects of actions and uses this model to decide through dynamic
9 programming, enables flexible adaptation to task changes at the expense of high computation costs.
10 The MF strategy is less flexible but also 1000 times less costly, and learns by observation of MB
11 decisions. We test the architecture in three experiments: a navigation task in a real environment with
12 task changes (wall configuration changes, goal location changes); a simulated object manipulation
13 task under human teaching signals; and a simulated human-robot cooperation task to tidy up objects
14 on a table. We show that our human-inspired strategy coordination method enables the robot to
15 maintain an optimal performance in terms of reward and computational cost compared to an MB
16 expert alone, which achieves the best performance but has the highest computational cost. We also
17 show that the method makes it possible to cope with sudden changes in the environment, goal changes
18 or changes in the behavior of the human partner during interaction tasks. The robots that performed
19 these experiments, whether real or virtual, all used the same set of parameters, thus showing the
20 generality of the method.

21 **Keywords** strategy coordination, cognitive monitoring, reinforcement learning, robot cognitive architecture, navigation,
22 HRI, neuro-inspiration

23 1 Introduction

24 The field of robot reinforcement learning (RL) has seen a fast growth in the last decade [Kober et al., 2013, Khamassi
25 et al., 2018, Ibarz et al., 2021]. In particular, notable progresses have been made with the use of deep RL algorithms
26 [Mnih et al., 2015], which enable to deal with large continuous state and action spaces. Nevertheless, these methods
27 are computationally very costly, requiring millions of iterations before convergence [Justus et al., 2018, Strubell et al.,
28 2019]. Moreover, they are most of the time designed specifically for a given scenario, thus preventing generalization.
29 More precisely, the human designer either goes for a model-based (MB) RL, when it seems feasible for the robot to try
30 and estimate a model of the effect of its actions, or for a model-free (MF) RL one, when it does not seem feasible [Wang
31 et al., 2019]. Overall, a wide variety of algorithmic solutions exist (some being value-based, other being policy-based),
32 each being more appropriate to specific experimental scenarios [Kober et al., 2013]. While recent hybrid MB/MF robot
33 learning methods have been proposed [Caluwaerts et al., 2012b, Chebotar et al., 2017], it is not clear if they could cope
34 on-the-fly with the high degree of variability and non-stationarity of human-robot interaction (HRI), and at the same
35 time minimize computational cost. To our knowledge, no generic solution exists that enable robots to automatically
36 choose the most efficient and least costly learning algorithm in a variety of contexts depending on the characteristics of
37 the task at hand.

38 In contrast, humans, and more generally mammals, are endowed with behavioral flexibility which enable them to
39 adapt to a variety of contexts and situations. One of the key ingredients of this behavioral flexibility is thought to
40 be a certain degree of modularity within their cognitive architecture, so that learning and decision-making processes
41 rely on the alternation and sometimes combination of different learning strategies [Hikosaka et al., 1999, Daw et al.,
42 2005, Dollé et al., 2008, 2010, Khamassi et al., 2011, Khamassi and Humphries, 2012, Van Der Meer et al., 2012,
43 O'Doherty et al., 2017]. In other words, humans have different cognitive tools within their mental toolbox, and can
44 reuse the tools they think are appropriate in new situations while minimizing cognitive effort [Shenhav et al., 2013,
45 Zenon et al., 2019]. More precisely, it has been shown that humans rely on a mixture of MB and MF RL processes when
46 facing contexts requiring repeated decisions [Daw et al., 2011, Lee et al., 2014, Viejo et al., 2015]. They are moreover
47 able to recognize the degrees of stability and familiarity of a given task to decide when to shift between these two
48 behavioral modes. Importantly, these human cognitive abilities have recently been modeled with the deep reinforcement
49 learning framework [Wang et al., 2018]. However, these approaches still rely on task-specific parameterization and
50 computationally heavy pretraining, and do not explicitly address genericity nor cost reduction.

51 The idea of taking inspiration from how the brain coordinates multiple learning systems to enable more flexibility
52 in robots has received increased attention in the robotics community during the last couple of decades [Girard et al.,
53 2005, Meyer and Guillot, 2008, Caluwaerts et al., 2012b, Zambelli and Demiris, 2016, Banquet et al., 2016, Lowrey
54 et al., 2019]. Furthermore, robot cognitive architectures combining both MB and MF learning processes have started
55 to be studied in recent years [Caluwaerts et al., 2012b, Jauffret et al., 2013, Renaudo et al., 2014, 2015b, Llofri
56 et al., 2015, Maffei et al., 2015, Chatila et al., 2018, Sheikhezahad Fard and Trappenberg, 2019, Hafez et al., 2019,
57 Rojas-Castro et al., 2020, Hangl et al., 2020]. Among these proposals, we have previously proposed a way to implement
58 these principles within a classical three-layered robot cognitive architecture, to facilitate integration with other sensing
59 and control components, as well as to permit future transfer to different robotic platforms [Renaudo et al., 2015c].
60 Nevertheless, to our knowledge, none of these recent projects have studied (1) the extent to which combining MB
61 and MF RL can provide *behavioral flexibility* and simultaneously *reduce computational cost*, by enabling robots to
62 autonomously determine when to avoid the high cost of MB planning when an MF strategy is considered sufficient;
63 and (2) the extent to which such a multi-strategy architecture is effective in a variety of tasks, including social and
64 non-social ones, and thus can be generalized to different scenarios and situations.

65 Here, we present a novel robot reinforcement learning architecture which display behavioral flexibility by dynamically
66 shifting between MB and MF RL through the arbitration of a trade-off between performance and computation cost.
67 We test the new algorithm during simulated and real robot experiments, and test its generalizability without parameter
68 re-tuning in three different scenarios: a navigation task involving paths of different lengths to the goal, dead-ends, and
69 non-stationarity; a human-robot interaction task where the robot learns to put objects in the rights containers under
70 human teaching signals; a human-robot cooperation task where both human and robot have to hand-over some objects
71 to the other agent in order to put them in their respective containers. We find that the proposed architecture flexibly and
72 consistently switches to MB control after environmental changes in any of the three scenarios. It moreover efficiently
73 switches to MF control when the task is recognized as stationary. Overall, the robot achieves the same performance as
74 optimal MB control in the three scenarios, while dividing computation time by more than two.

75 Part of the results in the navigation scenario (Experiment 1), those with change in reward location, but not those with
 76 change in the wall configuration, have been published in a conference paper [Dromnelle et al., 2020b]. Part of the results
 77 in the HRI scenario (Experiment 2) have been published in a second conference paper [Dromnelle et al., 2020a]. We
 78 present new unpublished results in both experiments, new extended analyses of the properties of the robotic architecture
 79 which explain these results, and a thorougher description of the methods. Experiment 3 is completely new.

80 In summary, we propose an original and efficient human-inspired mechanism for the coordination of robot learning
 81 systems in a variety of scenarios. To our knowledge, this is the first robotic implementation of a hybrid MB/MF
 82 algorithm that efficiently reduces computation cost while maintaining performance, and which can cope with human
 83 behavioral variability during HRI. This feature can be a key advantage from an ecological point of view and for robots
 84 that can only rely only on their limited internal computational and energetic resources to achieve their objectives.

85 2 Material and Methods

86 2.1 Markov Decision Problem

87 In the three scenarios considered in this work, we systematically consider the robot as an RL agent facing a Markov
 88 decision problem (MDP) [Sutton and Barto, 1998]. This means that the robot will experience a series of discrete states
 89 $s \in \mathcal{S}$, choosing what to do at each iteration t (*i.e.*, timestep) within a finite set of discrete actions $a \in \mathcal{A}$, with the goal
 90 of maximizing the sum of cumulative reward $r \in \mathbb{R}$ over a potentially infinite horizon (the robot does not know in
 91 advance how long the task will last): $f(t) = \sum_{t=0}^{\infty} \gamma^t r_t$ with $0 \leq \gamma \leq 1$.

92 The MDP can be described by the n-uplet $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$ where $T : (\mathcal{S}, \mathcal{A}) \rightarrow \mathcal{S}$ is the transition function, which
 93 represents the probability $P(s'|s, a)$ of arriving in state s' after executing action a in state s , and $R : \mathcal{S} \rightarrow \mathbb{R}$ is the
 94 reward function, which represents the scalar reward r that the robot can get after reaching state s' .

95 It is important to note that using a discrete state space does not necessarily mean that the human designer always
 96 has to pre-define in advance the decomposition of the task into discrete states. As we will see in the navigation
 97 scenario (Experiment 1), we propose a method for the autonomous decomposition of states from the data acquired
 98 through a Simultaneous Localization and Mapping Algorithm (SLAM, Grisetti et al. [2007]) by the real robot during
 99 initial random navigation within the environment. In that case, the states will represent unique locations in space,
 100 and the actions allowed to the robot represent moves in eight cardinal directions: north, north-east, east, etc. In the
 101 Human-Robot Interaction (HRI) scenarios (Experiments 2 and 3), the states will represent the configuration of cubes on
 102 a table and the possible actions will be: pick a cube, place a cube in a container, hand-over a cube to the human, take
 103 the cube that the human is handing over. Moreover, we will present our method for the robot to autonomously learn a
 104 world model from the data it collects during initial exploration, this model consisting in the estimations \hat{T} and \hat{R} of the
 105 transition and reward functions T and R , respectively. The robot will then use this learned world model to perform
 106 mental simulations through Dynamic Programming [Sutton and Barto, 1998], and hence bootstrap learning within a
 107 few hundreds of iterations, thanks to such an MB strategy.

108 The rationale here for using discrete state and action spaces, and addressing them with a hybrid MB/MF learning
 109 strategy, is to test in a robot the performance, computational cost and generalizability of a human-inspired model. We
 110 thus want to evaluate to which extent it enables robot fast adaptation and quick (in the order of thousands of iterations)
 111 reaching of an optimal performance at a low computational cost, inspired by human ability to quickly adapt in new
 112 situations. This human ability is currently thought to rely on the combination of MB and MF RL applied to such
 113 discrete representations of the task at hand [Daw et al., 2011, Lee et al., 2014, Viejo et al., 2015]. In contrast, current
 114 deep RL methods are computationally heavy and cannot achieve an optimal performance in these simple tasks within a
 115 few thousands of iterations (we will even show cases of adaptations to task changes within a few hundreds of iterations),
 116 but rather require millions of iterations [Wang et al., 2019]. We will illustrate in the navigation scenario that at the end
 117 of the experiment, after the robot has performed 6400 actions, that a Deep Q-Network (DQN) [Mnih et al., 2015] barely
 118 had time to slightly improve its performance, compared to the other tested algorithms.

119 2.2 A robot cognitive architecture with a dual decision-making process

120 The present work implements a classical three-layer robot cognitive architecture [Gat, 1998, Alami et al., 1998]
 121 composed of a decision, an executive and a functional layer. The decision layer of the proposed architecture (Fig. 1) is
 122 composed of two competing experts which generate action propositions, each with its own method and with its own
 123 advantages and disadvantages. These two experts are directly inspired by current computational neuroscience models
 124 which combine MB and MF RL strategies for navigation [Khamassi and Humphries, 2012], and more generally for
 125 decision-making tasks [Daw et al., 2005, 2011]. Hereafter, we follow the decomposition of the computations of each

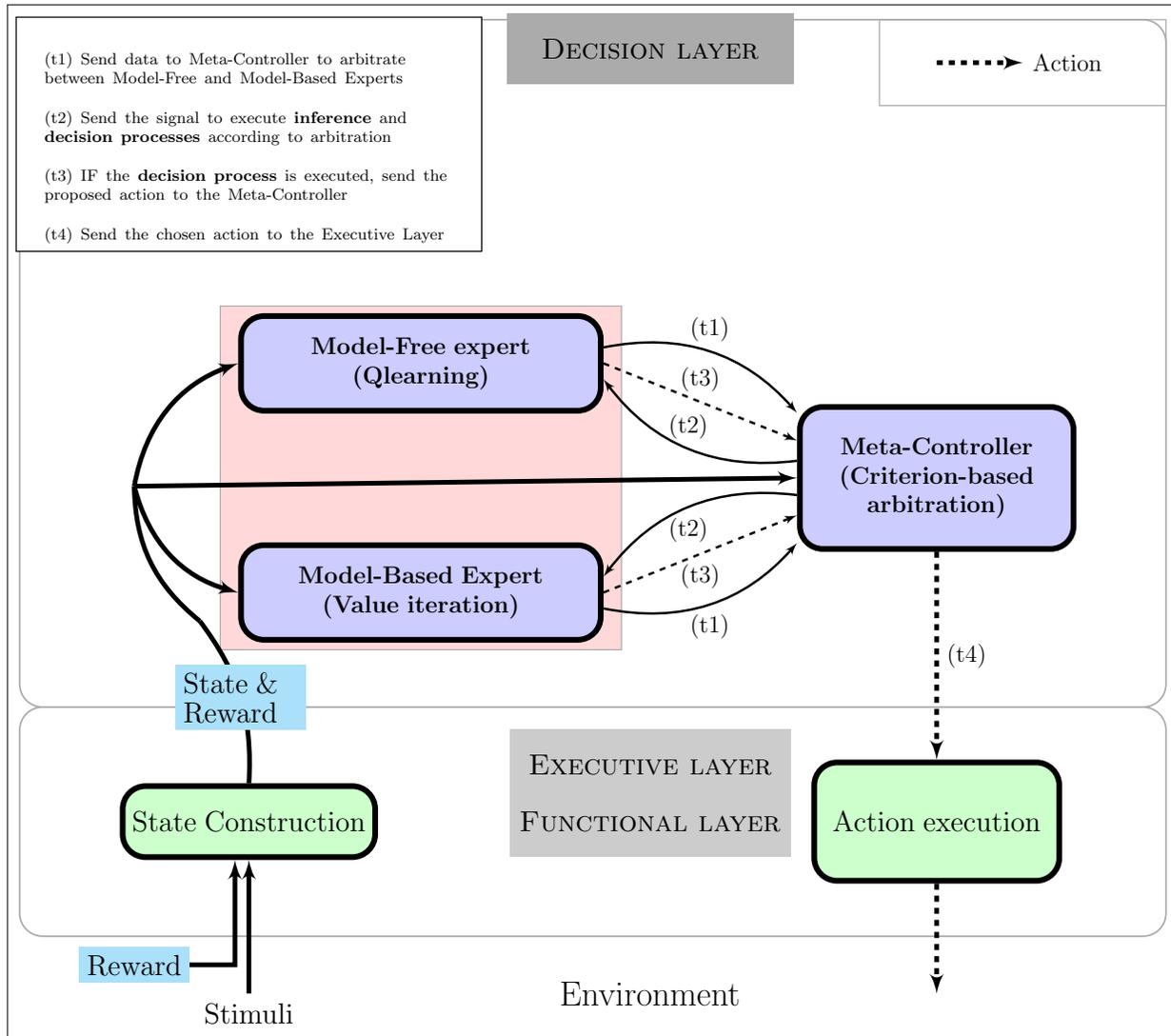


Figure 1: General structure of the architecture. Two experts having different properties are computing the next action to do in the current state s . They each send monitoring data to the meta-controller (MC) about their learning status and inference process (t1). The MC chooses an expert according to a criterion that uses this data and authorizes it to carry out its inference and decision processes (t2). After the decision, the chosen expert sends its proposition to the MC (t3), which sends the action to the Executive Layer (t4). The effect of the executed action generates a new perception, transformed into an abstract Markovian state, and eventually a non null reward r , that are sent to the experts. Each expert learns according to the action chosen by the MC, the new state reached and the reward. Figure by Dromnelle, Renaudo, Khamassi and Girard (2022); available under a CC-BY4.0 licence (<https://doi.org/10.6084/m9.figshare.21031723>).

126 expert into three processes, namely learning, inference and decision [Cazé et al., 2018], in order to more clearly identify
 127 what is the respective computational cost of each of these processes.

128 The decision layer is also equipped with a meta-controller (MC) in charge of arbitrating between experts. The MC
 129 determines which expert will perform inference and decision steps in the current state, according to an arbitration
 130 criterion. After that, the decision layer sends the chosen action to the executive layer, who ensures its accomplishment
 131 by recruiting robot's skills from the functional layer. The latter consists of a set of reactive sensorimotor loops that
 132 control actuators during interaction with the environment. The robot reaches a new state and obtains or not a reward.
 133 The two experts use the new state and the reward information to update their knowledge about the executed action. This
 134 allows MB and MF experts to cooperate by learning from each others' decision.

135 Compared to our previous architecture [Renauldo et al., 2015b], several changes have been made: The overall organiza-
 136 tion of the decision-making layer and the prioritization of communication between modules have been changed; The
 137 MF expert is no longer built as a neural network but as a tabular algorithm; The MC chooses which expert is the most
 138 suitable at a given time and in a given state, and no longer simply at a given time; And above all, we have defined a
 139 novel arbitration criterion that not only compares experts’ performance, but also their estimated computational cost.

140 2.3 The decision layer

141 2.3.1 Model-based (MB) expert

142 The MB expert learns a transition model T and a reward model R of the problem, and uses them to compute the
 143 values of actions in each state. These models allow to simulate over several steps the consequences of following a
 144 given behavior and to look for desirable states to reach. Consequently, when the task changes, the robot can use this
 145 knowledge to find the new relevant behavior with little actual interactions with the world. However, this search process
 146 is costly in terms of computation time as it needs to simulate several value iterations [Sutton and Barto, 1998] in each
 147 state to find the correct solution.

148 **Learning process.** The learning process of the MB consists in updating the reward and the transition models by
 149 interacting with the world. The transition model T is learnt by counting occurrences of transitions (s, a, s') . A
 150 pretraining phase can take place to improve the robot’s transition model before the beginning of task. Nevertheless, the
 151 transition model is updated all along the experiment, so that the robot can adapt to task changes.

152 The transition model T is updated using the number of visits $V_N(s, a)$ of state s and action a . $V_N(s, a)$ has a maximum
 153 value of N and $V_N(s, a, s')$ is the number of visits of the transition (s, a, s') in the last N visits of (s, a) . The transition
 154 probability $T(s, a, s')$ is defined in Eq. 1. This leads to an estimation of the probability to the closest multiple of $1/N$:

$$T(s, a, s') = \frac{V_N(s, a, s')}{V_N(s, a)} \quad (1)$$

155 The reward model R stores the most recent reward value r_t received for performing action a in state s and reaching the
 156 current state s' , multiplied by the probability of the transition (s, a, s') .

157 **Inference process.** Performing the process of inference consists in planning using a tabular Value Iteration algorithm
 158 [Sutton and Barto, 1998]:

$$Q(s, a) \leftarrow \sum_{s'} T(s, a, s') [R(s') + \gamma \max_{k \in \mathcal{A}} Q(s', k)] \quad (2)$$

159 $Q(s, a)$ is the action-value estimated by the agent for performing the action a in the state s , $R(s')$ the probabilistic
 160 reward of the reward model R associated with the state (s') and γ the decay rate of future rewards.

161 **Decision process.** Performing the decision process consists in converting the estimation of action-values into a
 162 distribution of action probabilities using a Boltzmann softmax function, and drawing the action proposal from this dis-
 163 tribution. We moreover introduce the possibility of human interventions under the form of a bias $Q_H(s, a)$ representing
 164 the human’s preferences for action (these will be used for HRI tasks in Experiments 2 and 3, but not in the navigation
 165 task of Experiment 1):

$$P(a|s) = \frac{\exp((Q(s, a) + \alpha_H * Q_H(s, a))/\tau)}{\sum_{b \in \mathcal{A}} \exp((Q(s, b) + \alpha_H * Q_H(s, b))/\tau)} \quad (3)$$

166 where τ is the exploration/exploitation trade-off parameter, and where the human-predicted preference (bias) $Q_H(s, a)$
 167 equals 1 if the human praised the robot the last time it performed the action a in state s , and 0 otherwise. For the sake
 168 of parsimony, the weight of the human bias α_H is identical to the learning rate of the robot α .

169 2.3.2 Model-free (MF) expert

170 The MF algorithm does not use models of the problem to decide which action to do in each state, but directly learns
 171 the state-action associations by caching in each state the earned rewards in the value of each action (action-values).
 172 Because updating the action-values is local to the visited state, the learning process is slow and the robot cannot learn

173 the topological relationships between states. Consequently, when the task changes, the robot takes many actions to
 174 adopt the new relevant behavior. On the other hand, this method is less expensive in terms of inference duration.

175 **Learning process.** Performing the learning process consists in estimating the action-value $Q(s, a)$ using a tabular
 176 Q-learning algorithm:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[R(s) + \gamma \max_k Q(s', k) - Q(s, a)] \quad (4)$$

177 where α is the learning rate, $R(s)$ is the scalar reward received for reaching the state s , γ is the decay rate of future
 178 rewards (same as γ used by MB in Eq. 2), and s' is the state reached after executing a .

179 **Inference process.** Since the MF expert does not use planning, its inference process consists only in reading from the
 180 table that contains all the action-values the one that corresponds to performing the action a in the state s .

181 **Decision process.** The decision process is the same as for the MB expert (Eq. 3).

182 2.3.3 Meta-controller and arbitration method.

183 The MC is in charge of selecting which expert will generate the behavior. For each state s , it computes the entropy
 184 of the action probability distribution $H(s, E)$ of expert E [Viejo et al., 2015], which is close to the notion of trust in
 185 [Rutard et al., 2020]:

$$H(s, E, t) = - \sum_{a=0}^{|A|} g(P(a|s, E, t)) \cdot \log_2(g(P(a|s, E, t))) \quad (5)$$

186 where $g(P(a|s, E, t))$ is a low-pass filtered action probability distribution, estimated from the past inferences performed
 187 by expert E , with time constant $\tau = 0.67$, which has previously been found to reflect the quality of learning in humans
 188 [Viejo et al., 2015]. The lower the entropy, the lower the uncertainty of the agent about the action to choose. So the
 189 lower the entropy, the higher the quality of learning. The action selection probabilities used to compute the entropy are
 190 averaged over time, per state, using an exponential moving average.

191 For each state, the MC also computes the low-pass filtered duration of the previous inference processes $C_T(s, E, t)$ of
 192 expert E , measured in actual simulation time. The novel arbitration criterion that we propose here is a trade-off between
 193 the quality of learning and the cost of inference. By using it, the MC can decide between favouring the most certain
 194 expert (the most efficient) and the cheapest expert in terms of computations. Note that the inference process of an expert
 195 does need to be run before the meta-controller’s arbitration since it relies on a low-pass filtered memory of the past
 196 costs of each expert in each state. The meta-controller computes the expert-value $Q(s, E)$ for each expert as following:

$$Q(s, E, t) = - [H(s, E, t) + \exp(-\kappa H(s, MF, t)) C_T(s, E, t)] \quad (6)$$

197 where the term $\exp(-\kappa H(s, MF, t))$ allows to weight the impact of computation costs in the criterion: The lower the
 198 entropy of the MF distribution of action probabilities, the more the computation cost of the inference process weights in
 199 the equation. We have chosen the value (here $\kappa = 7$) of the weighting of $-H(s, MF, t)$ according to a Pareto front
 200 analysis [Powell and Sammut-Bonnicci, 2015] (Figure 2, left). We were looking for a κ that minimizes the cost of
 201 inference, while maximizing the agent’s ability to accumulate reward over time (here we tried to loose less than 1%
 202 of the maximum, dashed line on fig. 2, left), in the two non-stationary navigation tasks detailed in the next section.
 203 Figure 2, right, illustrates this process by showing the way $\exp(-\kappa H(s, MF, t))$ evolves as a function of the value of
 204 the entropy $H(s, MF, t)$ and parameter κ .

205 Finally, the MC converts the estimation of expert-values $Q(s, E)$ into a distribution of expert probabilities using a
 206 softmax function (Eq. 3), and samples the activated expert from this distribution. The inference process of the unchosen
 207 expert is inhibited, which thus allows the system to save the corresponding computation time.

208 2.4 World-model building

209 In this work, we alternate experiments in simulation and with the real robot. This is to enable the robot to learn a world
 210 model of the task in reality, then use this world model for simulations permitting to tune the parameters and evaluate the

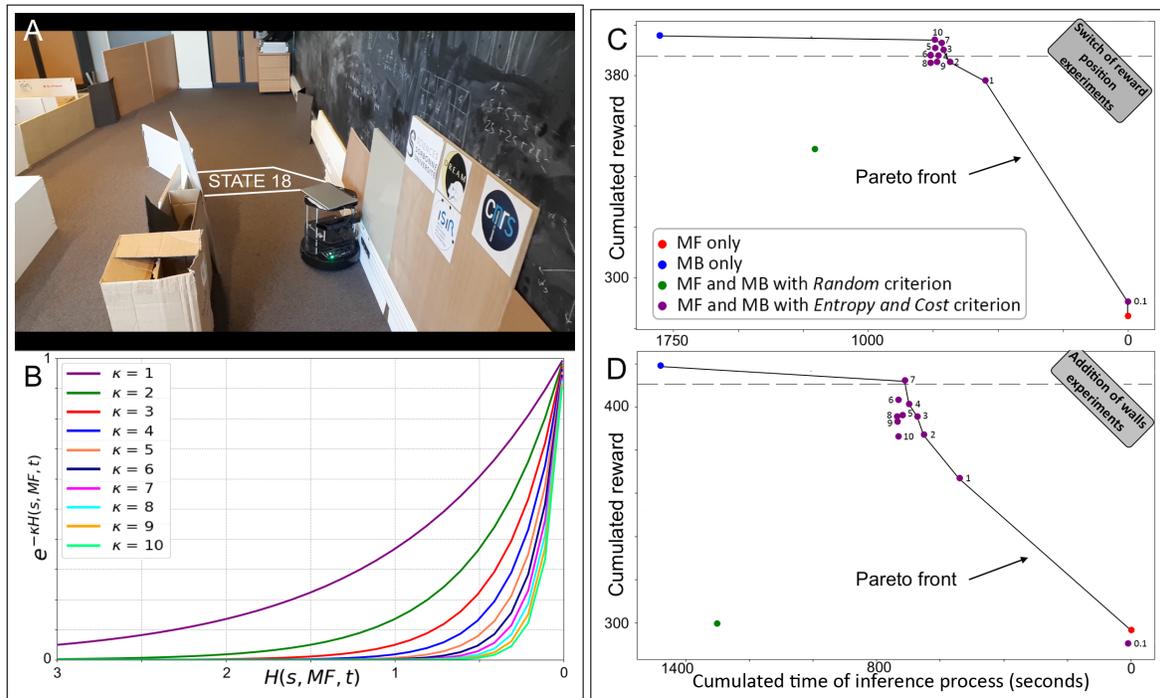


Figure 2: Selection of the value of the κ parameter in simulations of the navigation task (Experiment 1). **A**. Indoor arena used for the navigation task with the real robot. State 18 depicts the initial reward location. The robot learned a discrete map of the environment which was then used for parameter optimization in simulation. **B**. Shape of the $\exp(-\kappa H(s, E, t))$ function for various values of the κ parameter. **C,D**. Cumulated reward and cumulated computational cost obtained with various values of κ (Eq. 6) in the MC-EC architecture (purple), versus the MF-only (red), MB-only (blue) and MC-Rnd (green) controls. The dashed line represents 0.99% of the maximal cumulated reward measured. The analysis was performed on data collected in the two non-stationary navigation scenarios (top: displace reward scenario; bottom: added wall scenario). Figure by Dromnelle, Renaudo, Khamassi and Girard (2022); available under a CC-BY4.0 licence (<https://doi.org/10.6084/m9.figshare.21031723>).

211 proposed robot cognitive architecture. And finally perform the learning experiments with the real robot under various
 212 conditions: Change in the reward function R of the MDP, change in the transition function T of the MDP.

213 Figure 3 illustrates the method. The robot first learns a world model from real data collected during initial exploration.
 214 Then the world model is used as a new approximate but realistic MDP to perform offline simulations. These simulations
 215 serve to evaluate the robot cognitive architecture, measure its performance and cost in different conditions, and optimize
 216 its parameters in simulation, thus more quickly than with a real robot. Finally, the parameterized architecture can be
 217 tested again on the real robot, where MB and MF RL strategies can learn in parallel the new task conditions imposed to
 218 the robot.

219 The method is here illustrated with a navigation scenario, easy to conceptualize and visualize. But it is a generic method
 220 which can be used in other scenarios, such as MDPs for HRI with humans.

221 2.5 General information

222 Similarly to the Rmax algorithm [Sutton and Barto, 1998], we initialized the action values to non-zero values so to
 223 help exploration of non-previously selected actions, since the action values are updated according to the previous ones.
 224 Thus, in any non-rewarded states, having previously selected at least one action results in a non-flat action probability
 225 distribution, and thus more chances to select another one (exploration). More precisely, the initial action values are set
 226 to 1 for both experts.

227 For the MF expert, we conducted a grid search to find the best parameter-set, *i.e.*, parameters maximizing the total
 228 accumulated reward over a fixed duration of 1600 timesteps (which is the duration of the first phase of the navigation
 229 phase, before task changes occur). As this expert is very slow to learn compared to the MB expert, it is important to
 230 ensure that it can display a beginning of performance improvement within this duration. We found $\alpha = 0.6$, $\gamma = 0.9$ and

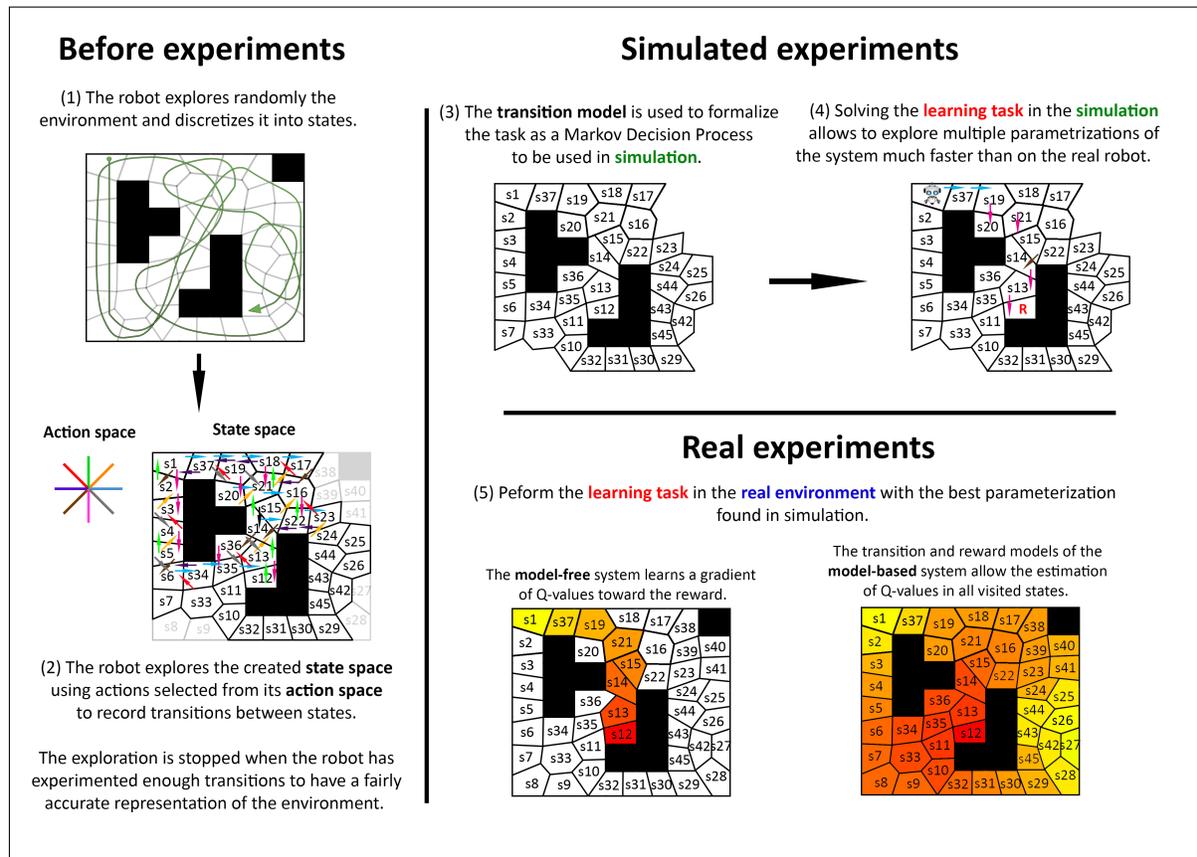


Figure 3: The different phases of the method used for world model building and offline usage. We illustrate the method with a navigation scenario, easy to conceptualize and visualize, but the method is generic and can be used in other scenarios, such as MDPs for HRI. Figure by Dromnelle, Renaudo, Khamassi and Girard (2022); available under a CC-BY4.0 licence (<https://doi.org/10.6084/m9.figshare.21031723>).

231 $\tau = 0.02$. For the MB expert, we chose $\gamma = 0.95$. For the MB expert and the MC, we chose the same value of τ as the
 232 MF expert. Finally, for the MC, we choose a gating parameter $\kappa = 7$.

233 3 Experiment 1: Navigation task

234 The work described in this section presents extended analyses of the results of Dromnelle et al. [2020b], plus unpublished
 235 results in a new condition of the task (changes in wall configuration). Finally, we also provide more details about
 236 the world model building method, because it will also be used in Experiments 2 and 3. We will refer to Dromnelle
 237 et al. [2020b] for previously published results, which can be accessed from: [https://hal.archives-ouvertes.](https://hal.archives-ouvertes.fr/hal-02883717v3/document)
 238 [fr/hal-02883717v3/document](https://hal.archives-ouvertes.fr/hal-02883717v3/document).

239 3.1 Methods

240 We first evaluated our cognitive architecture in a navigation task. Since running 1600 actions on the robot takes about
 241 six hours, we have created a simulation of the task where the probabilities of transitions are derived from a world model
 242 learned by the real robot during a 13 hours exploration of the real arena (Section 2.4). This simulation allowed us to
 243 quickly test multiple coordination criteria and parameterizations, before evaluating them on a real robot.

244 We used a 2.6 m x 9.5 m arena containing obstacles (Fig 2A), and a turtlebot. The computer uses ROS [Quigley et al.,
 245 2009] to process the signals from its sensors, controls the mobile base and interfaces with our architecture. A Kinect-1
 246 sensor returns an estimate of distance to obstacles in its field of view, completed by contact sensors at the front and sides
 247 of the mobile base. The robot localizes itself using the gmapping Simultaneous Localization and Mapping Algorithm
 248 (SLAM, [Grisetti et al., 2007]). During a preliminary environmental exploration phase, the robot incrementally builds a

249 discretized map by creating a new nodes every time its minimal distance with all existing nodes is larger than 35 cm, and
 250 thus autonomously creating new Markovian states (Fig. 4). The current state (of the corresponding MDP) is the closest
 251 node from the robot when its previous action is completed and it evaluates the consequences. We chose to build this
 252 map beforehand and to reuse it for each of the learning experiments, so as to reduce the sources of behavioral variability.
 253 However, note that with the present method the system could start with an empty map and build it incrementally, and
 254 that a new map could be used for each experiment.

255 In this experiment, the robot must learn to reach a specific state of the environment (state 18 – see Fig. 2A). When
 256 it succeeds, it receives a unitary reward and is randomly returned to one of the two initial positions, located in the
 257 extremities of the arena (states 0 and 32), to start over. The goal of the robot is first to reach state 18. Thus the reward
 258 used here could represent the energy that the robot gets when it reaches its battery recharging station, or it could
 259 represent the success for achieving the instruction given by a human to the robot to go to its home base.

260 Performing an action consists of moving in a certain direction and changing state. The robot can move along 8 equally
 261 distributed allocentric directions (Fig. 4). When the contact sensors are activated, the robot moves back 0.15 meters.
 262 Finally, according to the exact position in which the robot is located within a state, the arrival state will not necessarily
 263 be identical for the same action performed. The environment is therefore probabilistic, which multiplies the possibilities
 264 for the robot. For the MB expert, this specificity implies that the transitions $T(s, a, s')$ and the rewards $R(s, a)$ are
 265 stored respectively in the model of transition T and the model of reward R as probability distributions.

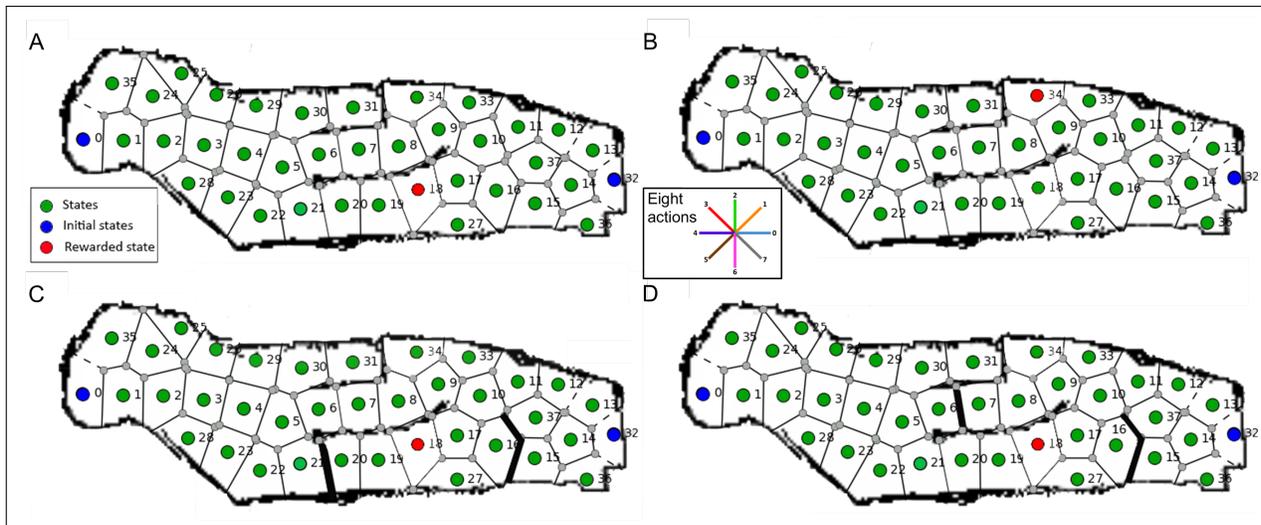


Figure 4: Configurations of the navigation task. **A.** Starting condition: The rewarding state is state #18 (red), the departure states are #0 and #32 (blue), all other states are in green. **B.** Goal-location change condition (after 1600 actions) used in [Dromnelle et al., 2020b]: The reward location is moved to state #34. The inset figure shows the eight actions available to the robot. **C&D.** Wall configuration change conditions (after 1600 actions): Obstacles are added that forbid the transitions between state #16 and states #15 and #37 (C&D), and either between states #20 and #21 (C) or states #6 and #7 (D).

266 The experiment involves a stable period during which the environment and reward do not change (Fig. 4A). Then, after
 267 the 1600th action a task change is imposed where the reward is moved from state 18 to state 34 (Fig. 4B). We also made
 268 a second series of experiments where the reward is fixed but some wall configurations are changed in the environment,
 269 either in the lower corridor (Fig. 4C) or in the middle corridor (Fig. 4D) depending which of these is preferentially used
 270 by the robot, when starting from state 0, so as to maximize the induced perturbation. We chose this duration of 1600
 271 actions (in the order of a few hours with the real robot, as mentioned above), so as to represent a realistic scenario in the
 272 context of HRI. In this situation, the human's instructions to the robot may change during the day: the robot may have
 273 to complete a task with a specific configuration of the environment in the morning, and then in the afternoon it has to
 274 learn a new goal location, or the configuration of the environment changes (*e.g.*, one of the corridors is obstructed while
 275 a human is repairing a light in the ceiling). Under these conditions, we cannot afford to use a learning algorithm which
 276 requires millions of actions before converging.

277 To evaluate the performance of the virtual robot, we studied four combinations of experts : (1) a MF-only robot using
 278 only the MF expert to decide, (2) an MB-only robot using only the MB expert to decide, (3) a random coordination
 279 robot which coordinates the two experts randomly and (4) an Entropy and Cost robot which coordinates the two experts

280 using the model of arbitration presented in 2.3.3. In Dromnelle et al. [2020b], we also compared our algorithm to a
 281 reference learning algorithm in the literature, a DQN deep neural network [Mnih et al., 2015], to show that our method
 282 outperforms it in terms of cumulated reward with very limited computational cost.

283 We define the “optimal behaviour” as the behaviour that allows the robot to accumulate the most reward over time.

284 The navigation task does not involve any human intervention, in contrast to the HRI tasks of Experiments 2 and 3. Thus,
 285 all the results of Experiment 1 were obtained with $\alpha_H = 0$ in the robot’s decision-making equation through softmax
 286 (Eq. (3)).

287 3.2 Results

288 Overall, the navigation experiment (Experiment 1) consists of two conditions:

- 289 • Condition 1 (simulation + real robot): initial learning followed by changes in goal location (published in
 290 Dromnelle et al. [2020b]).
- 291 • Condition 2 (simulation + real robot): initial learning followed by changes in wall configuration (unpublished).

292 We mainly focus on the presentation of the new results in Condition 2, while referring to Dromnelle et al. [2020b] and
 293 to the supplementary material to show that the global pattern of the results is similar between the two conditions. We
 294 moreover show replications of the simulated results in the real environment with a Turtlebot.

295 3.2.1 Trade-off between learning flexibility and computational cost

296 The first important result that we illustrate here with the wall configuration change condition (Fig. 5A,B) is that the MB
 297 and MF expert show complementarity in the trade-off between learning flexibility and computational cost:

- 298 • The MF-only robot (red) takes longer to reach the optimal behaviour during initial learning, is even slower
 299 to adapt to the task change after the 1600th action (Fig. 5A), but achieves this performance at a negligible
 300 computational cost (Fig. 5B). This is because its inference process simply consists in reading from the table
 301 that contains all the actions-values.
- 302 • In contrast, the MB-only robot (blue) has the best performance (Fig. 5A), but also the highest computational
 303 cost due to the planning process (about 1000 times higher than the MF-only robot) (Fig. 5B).

304 The Entropy and Cost (EC) robot (purple), which combines MB and MF experts through the meta-controller proposed
 305 in the present cognitive architecture (Fig. 1), manages to reach a non-significantly different performance from the
 306 MB-only robot (Mann-Whitney test, $df = 1$, $p = 0.171$), showing that our coordination method does not penalize the
 307 robot in terms of cumulated reward. This good performance is obtained despite the fact that the EC robot chooses
 308 the MF strategy more than 50% of the time after the 800th action (Fig. 5.C). This means that the MF strategy in the
 309 EC robot has learned faster than in the MF-only robot, taking advantage of the demonstrations provided by the MB
 310 expert. The activation of the MB expert is thus limited, which drastically reduces the computation cost (more than two
 311 times smaller than the MB-only robot at the end of the experiment, Fig. 5B). In addition, the EC robot performs better
 312 than the random coordination robot (green) suggesting that our coordination method is more efficient than randomly
 313 alternating between MB and MF control.

314 Thus in this task, the proposed architecture enables to benefit from the high learning flexibility of the MB-RL expert,
 315 with a limited computational cost thanks to the cheap MF-RL expert. These results replicate what we previously
 316 obtained in the change in goal location condition [Dromnelle et al., 2020b], and show similar properties when tested in
 317 the real robot (Online Resource Suppl. Fig. S4).

318 3.2.2 Emergent temporal pattern of expert selection

319 The second important result is the consistent temporal pattern of expert selection that emerges from the meta-controller’s
 320 expert selection rule (Equation 6). This pattern was observed (1) in the change in goal location condition [Dromnelle
 321 et al., 2020b], (2) in the simulated version of the change in wall configuration condition (Fig. 5.C), and (3) in the version
 322 with the real robot (Fig. 5.D), thus showing the robustness of the pattern. This pattern consists in:

- 323 • **The MF exploring phase (1 on Fig. 5.C):** Before the discovery of the position of the reward, the robot uses
 324 mainly the MF expert. This is due to the difference in the method for updating action-values between the
 325 two experts. With the same initial values and the set of parameters we have defined, the action-values of the
 326 MF expert decrease slightly more than those of the MB expert, which drives a more pronounced decrease of

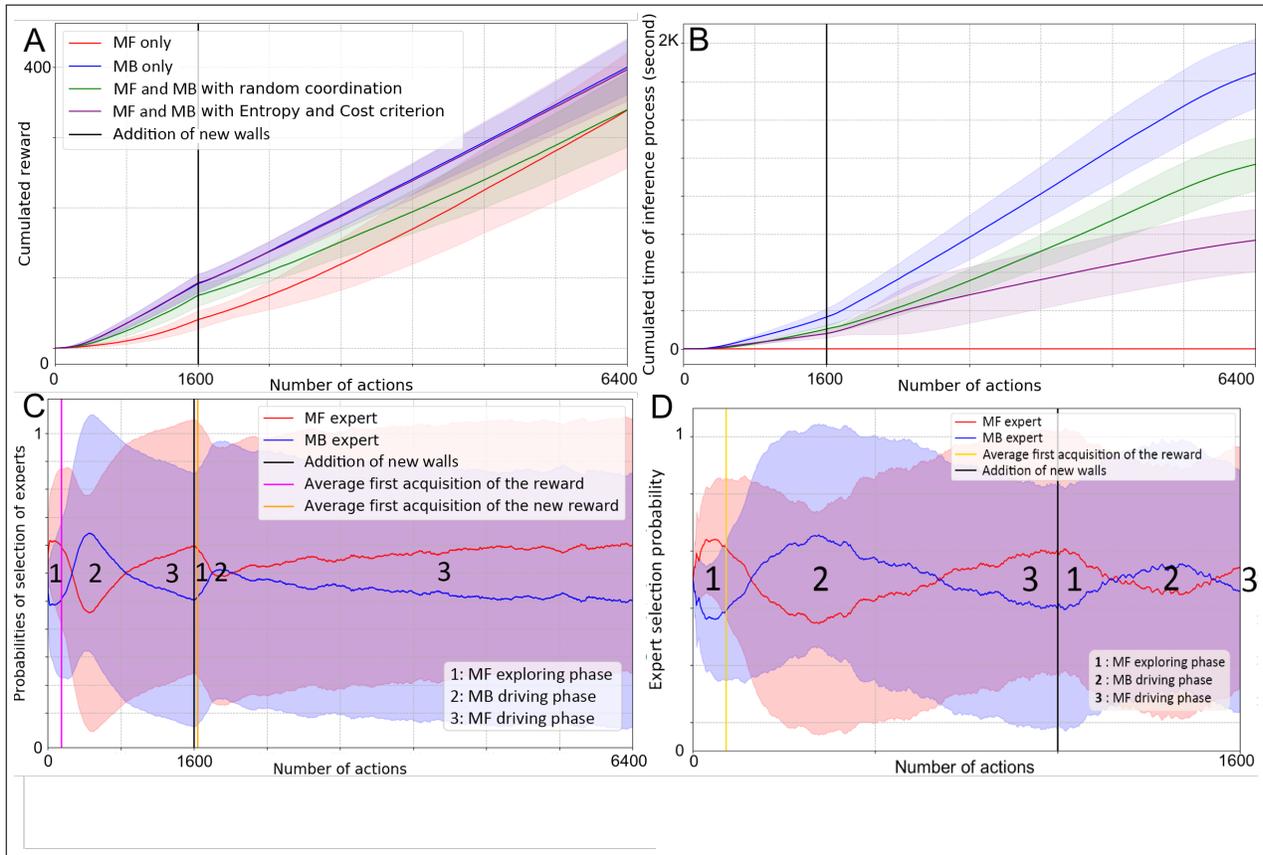


Figure 5: Simulation results of the wall configuration change condition of the navigation experiment: **A**. Mean performance for 100 simulated runs of the task. The performance is measured as the cumulative reward obtained over the duration of the experiment. The duration is represented as the number of actions performed by the robot. We use standard deviation as dispersion indicator. At the 1600th action, new walls are introduced in the arena, as illustrated in Fig. 4C-D. **B**. Mean computational cost for 100 simulated runs of the task. The computational cost is measured as the cumulative time of the inference process over the duration of the experiment in seconds. The duration is represented as the number of actions performed by the robot. **C**. Mean probabilities of selection of experts by the MC using the Entropy and Cost criterion for 100 simulated runs of the task. These probabilities are defined by the softmax function of each expert. The duration is represented as the number of actions performed by the robot. We use standard deviation as dispersion indicator. **D**. Mean probabilities of selection of experts by the MC-EC robot for 10 runs of the wall configuration change task with the real robot.

327
328
329

the entropy of the action probability distribution. In addition, since we do not have an expert specialized in exploration, it makes sense to use the computationally cheapest expert until the position of the reward has been discovered.

330
331
332
333

- **The MB driving phase (2 on Fig. 5.C):** After finding the first reward the MB expert progressively takes the lead on the decisions because its inference process needs only to find the reward once to spread action-values to all states of the environment thanks to its transition model. It can thus find the reward more easily than the MF expert, and so, its performance increases.

334
335
336
337

- **The MF driving phase (3 on Fig. 5.C):** The MF expert learns by demonstration from the MB expert, and thus spreads action-values from state to state and eventually, towards the 800th action, it reaches the performance of the MB expert. Because the MF expert is less expensive, the arbitration criterion (Eq. 6) gives it the lead over decisions.

338
339

- Interestingly, when a change in the task occurs (At the 1600th action on Fig. 5.C), the sequence of three phases appears again.

340 The large standard deviation shown in the figures is explained by the fact that for each experiment, the robot's strategy
 341 and behaviour can be very different, notably due to the large number of states and possible actions, but also to the
 342 probabilistic nature of the environment. As a result, the time of the switches from one phase to another varied a lot
 343 from one individual to another. Nevertheless the individual behavior of each run is consistent with the average behavior
 344 presented here (Online Resource Suppl. Fig. S1.B). Importantly, experiments with the real robot replicated the expert
 345 selection pattern obtained in simulation (Fig. 5D).

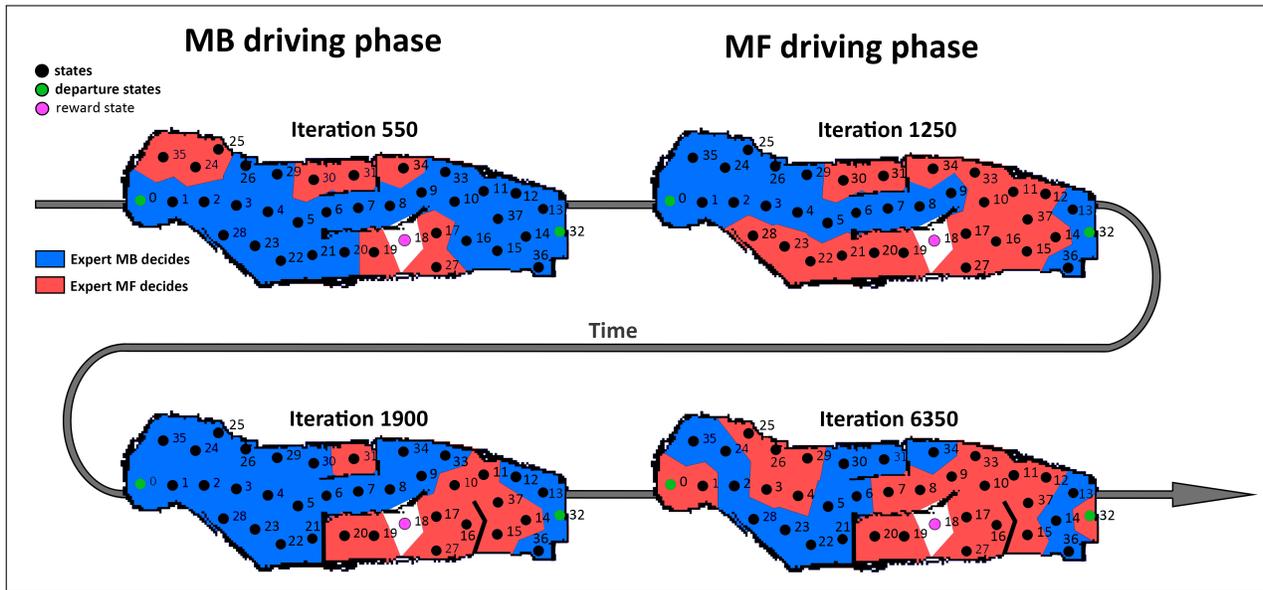


Figure 6: Evolution of the expert spatial preferences in the wall configuration change condition of the navigation experiment. Expert selection maps of the MC-EC robot for one of the hundred simulations: in red, states where the MF was the last chosen expert, in blue, where the MB was last chosen. after 1600 actions, new walls are introduced that, here, forbid the transitions between states between state #16 and states #16 and #37, and between states #20 and #21. The MF driving phase and the MB driving phase correspond to the behavioral phases identified in Fig. 5C.

346 3.2.3 Spatial pattern of expert selection

347 The last important result is the spatial pattern of expert selection: The MB and MF selection probabilities reported
 348 earlier were not the same in all states of the environment; The meta-controller (MC) turned out to stably prefer the MB
 349 expert in specific parts of the environment at different stages of learning, and preferred the MF expert in other parts or
 350 at different stages.

351 Figure 6 illustrates the expert selection map by the MC of the EC robot at different periods of the experiment. These
 352 maps show the relative dominance of MB and MF experts over the robot's decisions in different parts of the environment.
 353 They enable us to shed a different light on the emergence of the temporal pattern of expert selection reported in the
 354 previous subsection. During the MB driving phase, the map is mainly colored in blue, indicating a dominance of MB
 355 decisions, while during the MF driving phase, it is the opposite and the states are mostly colored in red. Interestingly,
 356 we can see with these maps how a spatial coordination pattern of MB and MF experts evolves with time: during the MF
 357 driving phase, paths composed of mostly red states start to appear. These paths approximately end up connecting the
 358 departure states to the rewarding state, although the states at the extremities of this path (states 0, 1, 2 and 32) are still
 359 preferentially controlled by the MB expert at the 1250th iteration in the example shown in Fig. 6. After the 1600th
 360 action, where a change in the wall configuration along the south corridor occurs in the example shown in the figure, the
 361 extremities of the red path vanish progressively, before re-forming themselves along the central corridor. This illustrates
 362 the new preference of the robot for the central corridor instead of the south one, because it is now the optimal path to
 363 the reward.

364 This leads to the distinction between two types of states: (1) states located on the optimal path, where the MF expert is
 365 well trained, and where the robot often goes; (2) states located at the border of the optimal path, where the MF expert
 366 received little training, and thus where the MB expert remains dominant. Because the robot does not often go outside
 367 the optimal paths after learning, the MF expert remains the most often selected. Nevertheless, when occasionally the
 368 robot gets outside the optimal path, the MC reacts by giving the lead to the MB expert which will bring the robot back

369 on track. This illustrates another important aspect of the behavioral flexibility produced by the architecture, which could
 370 contribute in explaining flexibility in humans, while neuroscience experiments usually cannot tell whether the biological
 371 “MB expert” is completely deactivated after learning or whether it remains potentially reactive to similar situations. This
 372 leads to a model-driven prediction which could be tested with future human experiments: An MB process should guide
 373 humans back to their familiar sequence of states and actions, after they got out of their optimal path in a given task.

374 Similar results were obtained in the change in goal location condition of the task (Online Resource Suppl. Fig. S2).
 375 Finally, Online Resource Suppl. Figures S5 and S6 show that the same pattern of spatial coordination of experts that we
 376 observed previously in simulation, also emerged over time with the real robot in the two types of experiments. However,
 377 one can note that the red paths are less complete than they were in the simulation results. This is a sign of a reality gap
 378 [Koos et al., 2012], meaning that the experiments with the real robot were more difficult, which impacted the robot’s
 379 ability to achieve the task.

380 Another interesting prediction for neuroscience from these results is that a situation with more difficulty, more volatility
 381 and uncertainty, could involve a more intertwined contribution of both MB and MF experts, even after a long training.
 382 In such cases, rather than observing a continuous activation, from departure until reward, of a putative MF expert in the
 383 brain, one would expect to observe intermittent activations of a putative MB expert along the robot’s trajectory.

384 Overall, the important thing to note is that the proposed robot architecture enables to adapt to different situations
 385 (different types of task changes), with different degrees of difficulty and uncertainty (simulation versus reality), with the
 386 same principle for expert coordination by the meta-controller. This enables to achieve a performance in these simple
 387 navigation tasks which is not different from optimality, at a drastically reduced computational cost.

388 4 Experiment 2: Human-robot interaction with human as teacher

389 In this section, we evaluate our robotic architecture and coordination system in a human-robot interaction task. First,
 390 we present the simulated task, consisting in putting colored cubes in colored containers on a table. Then we present the
 391 two types of simulated humans that we defined to interact with the robot. In the second part, we present the results
 392 obtained and show how our coordination system allows the robot, in a task with more states, and without major change
 393 in our architecture, to maintain again a high level of performance while decreasing greatly its computational cost, but
 394 also to deal with the volatility of human behavior. The work presented in this section is an extended version of the
 395 publication Dromelle et al. [2020a], to which we will refer when mentioning previously published results. The pdf of
 396 the publication can be accessed from: <https://hal.archives-ouvertes.fr/hal-02899767v2/document>.

397 4.1 Material and Methods

398 4.2 Simulated environment and robot

399 Unlike Experiment 1, this experiment was performed only in simulation. Here, a robot having at least one mobile
 400 arm, a visual sensor and a sound sensor faces a table. On the table, three containers and three cubes of different colors
 401 are placed. The robot is able to distinguish the colors of cubes and containers, and to manipulate each of the cubes.
 402 On the other side of the table, a human can interact verbally with the robot, but can also take control of the robot’s
 403 arm. We consider that the robot is able to interpret the very simple human messages consisting in either congratulating
 404 it, thus constituting a reward signal for the robot, or telling it to observe human demonstrations, thus constituting an
 405 observation of action by the robot. Figure 7 illustrates the experiment.

406 As for the navigation task, we represent the environment by a model of transitions between Markovian states. The
 407 transition model representing the simulated environment is not generated by a robot in the real world, since there is no
 408 real experience, but predefined by the experimenter. This model is deterministic: Each action carried out in each state
 409 by the robot leads to a single terminal state. It would undoubtedly be more complex if it had been generated by a robot
 410 carrying out this task in the real world, as for the navigation task of Experiment 1 (Section 3). Initially, we had planned
 411 to carry out the task with real human subjects and a *Baxter* robot, but the various lockdowns and the sanitary conditions
 412 in 2020 made us abandon this project and stick to simulations [Feil-Seifer et al., 2020]. Nevertheless, this HRI task
 413 model is in a sense already more complex than the navigation environment, as we will see in the next two subsections.

414 In this HRI task, the robot’s objective is to learn how to put each of the cubes, initially placed on the table, in the
 415 container of the corresponding color. When this is done, the robot gets a scalar reward, and the cubes are automatically
 416 put back on the table. Because real naive humans playing with the robot could have wanted the robot to achieve any
 417 possible configuration (*i.e.*, not always simply to put the red cube into the red container, and so on, as required here,
 418 but also sometimes to put the red cube into the blue container, the blue one into the green container, etc., or to put all
 419 cubes into the red container), the robot will have to learn by trial and error the configuration desired by the human.

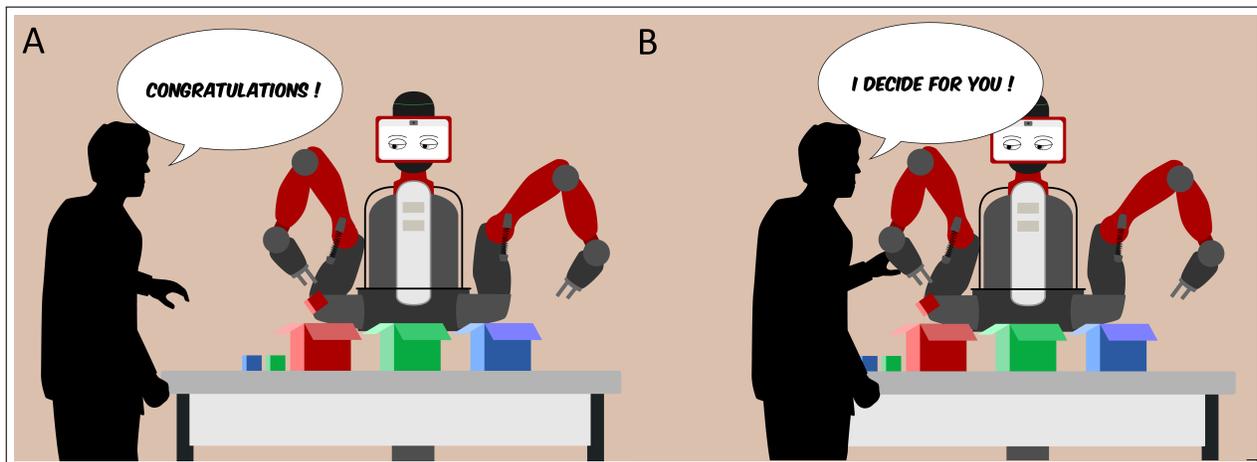


Figure 7: Human-Robot interaction task teaching signals. **A.** Human provides the robot with evaluative feedback (Human intervention type: *Congratulation*). **B.** Human provides the robot with demonstrations (Human intervention type: *Takeover*). Adapted from [Dromnelle et al. \[2020a\]](#), with permission from IEEE.

420 Importantly, the robot will have to learn this quickly, and to maintain a correct performance throughout the trials, in
 421 order to make the duration of the experiment consistent with real human-robot interactions, and to prevent humans
 422 from getting bored. Thus, even if the task is simple, we want the robot to quickly achieve an optimal performance at a
 423 low computational cost. This is the reason why we are interested in testing whether the same generic robot cognitive
 424 architecture can produce human-inspired behavioral flexibility also in this HRI task.

425 4.3 State and action spaces

426 As for the navigation experiment, the robot state space is discrete. Here, a state represents the position of the three
 427 colored cubes: In the red container, in the green container, in the blue container, on the table, in the robot's hand, or
 428 in the human's hand. If we remove the states where the robot and the human hold several cubes at the same time,
 429 there remains a total of 112 states, *i.e.*, three times as many states as in the navigation experiment. These 112 states
 430 correspond to $5 \times 5 \times 5 - 13$, because the 3 cubes can be put in 5 different positions (hand, table, red container, blue
 431 container, green container), from which we subtract the 13 configurations corresponding to the robot's hand having
 432 several cubes simultaneously.

433 Regarding the action space, the robot can perform 7 different actions: Take the red cube, take the green cube, take the
 434 blue cube, put the cube held in its hand into the red container, into the green container, into the blue container and onto
 435 the table.

436 While other ways of modeling the task would have been possible, such as with relational RL [[Džeroski et al., 2001](#)], we
 437 chose this state decomposition for several reasons: To remain in line with the representation used in the navigation
 438 experiment; For its ease of use; As a proof of concept of the interest of combining MF and MB learning strategies also
 439 in the field of human-robot interaction.

440 4.4 Pre-experimental babbling phase

441 A babbling phase precedes the experiment, where the robot can manipulate the cubes without getting rewarded. We
 442 defined this pre-experimental phase because in this task, the robot explores its environment much less than in the
 443 navigation task (at an equivalent exploration parameter τ), which may have significant repercussions on the performance
 444 of the robot. The reasons for this less extensive exploration are as follows:

- 445 • The environment of this HRI task is defined by approximately three times as many states as in the navigation
 446 task (112 states for the former, 38 for the latter),
- 447 • Only 6 actions must be performed from the initial state to reach the final state (*i.e.*, approximately 5% of the
 448 total number of states), against 9 in the navigation task (*i.e.*, approximately 24% of the total).

449
450
451

- The environment is not probabilistic, each action performed by the robot in each state of this task leads to a single terminal state. If the probabilistic environment in the navigation task made it more complicated for the robot to traverse, it also allowed it to discover unexplored states by chance.

452
453
454
455
456
457
458
459
460
461

First, we evaluated the performances of the robot in the HRI task after several babbling durations, using our arbitration criterion (MC-EC) and without human intervention (Fig. 8). We found an optimal babbling duration of 1200 iterations. Beyond that, babbling no longer improved the performance of the robot. Of course, we could also choose to give the robot a more or less complete transition model before the start of the experiment. We consider here the case where the robot has no a priori knowledge about the environment, apart from predefined state and action spaces. In the same way, we could very well imagine that the transition model built by the robot before the first experiment could be reused for all the following experiments. This would be particularly useful in the case of real experiments, where pretraining the robot can accelerate its performance for the next interactions with human participants. Nevertheless, in the present simulations, including a babbling phase enables to estimate how many iterations are required by the robot to learn a correct transition model.

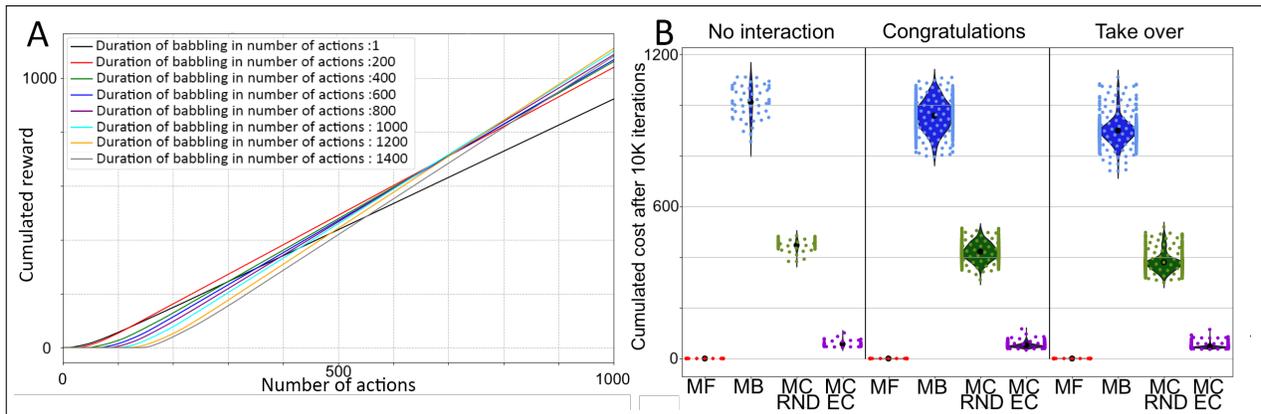


Figure 8: Results in the HRI teaching task. **A.** Average performance of the MC-EC robot for different babbling durations. For each duration, 50 simulated experiments were performed. Performance is defined as the robot’s ability to accumulate reward over the duration of the experiment that follows the babbling phase. The duration is represented by the number of actions performed by the robot. **B.** Costs of the inference processes accumulated at the 10000th iteration by the different robots and for the different types of intervention. The colored dots represent the unit performances of the different experiments and the black dots the average performances for all the experiments and all durations of interventions combined, that is to say 600 experiments per type of robot.

462 4.5 Simulated humans

463
464
465
466

A simulated human able to interact with the robot faces the table. We have defined two ways for the robot to learn from humans, drawing inspiration from the concepts of *learning by evaluative feedback* and *learning by demonstration* [Knox and Stone, 2009, Judah et al., 2010, Griffith et al., 2013]. We name respectively the two types of underlying interventions: Intervention of the type *congratulation* and intervention of the type *takeover*. More precisely:

467
468
469
470
471
472
473
474
475
476
477
478
479

- In the case of the *congratulation* type intervention, the human can congratulate the robot after it has put a cube in the correct container, for example the red cube in the red container (Fig. 7A). The effect of the intervention will be effective the next time the robot is again in the same situation (when it holds the red cube again). Knox and Stone [2012] have previously shown that the more human praise directly affects the robot’s action selection process, the better the robot. Conversely, the more human praise affects the update of state-action values for each experienced transition, the worse it is. Thus, in our work, we model the human’s congratulation, and therefore his/her preference, as a positive bias (a bonus) of an state-action value valid only during the decision process, rather than as a direct modification direct of state-action values. Concretely, we are inspired by the *policy shaping* method named *Action Biasing* [Knox and Stone, 2012], and thus use a non-null parameter α_H to weight the human-predicted preference (bias) $Q_H(s, a)$ in the *softmax* function (Eq. (3)).
- In the case of the *takeover* type intervention, the human can override the choice of the robot, when a cube is held by it, by choosing the place where it will be placed (Fig. 7B). As for the congratulation, the demonstration of the human is associated with a single state-action pair (s_0, a_0) . Note that compared to the congratulation,

480 the demonstration has an instantaneous effect on the robot. And even if it cannot act during these moments,
481 the robot still learns from observing the consequences of the actions chosen by the human.

482 We note that in both cases, no human intervention memorization process was modeled. By interacting with the robot
483 to influence its decisions, the human biases the updating of its action-state values. Therefore, the consequence of the
484 intervention is incorporated into the robot’s state-action value model, which illustrates both the robot’s choices and the
485 human’s preference, even if it is not possible to separate them.

486 4.6 Expert parameters

487 In order to show the generic and task-independent nature of our learning and meta-control system, we reused the same
488 set of parameters as the one used in the navigation task for each of the experts and for the meta-controller (Table 1).

Table 1: Chosen values of experts and meta-controller parameters in the cube ordering task.

Param	MB	MF	MC
α	n.a.	0.6	n.a.
τ	0.02	0.02	0.02
γ	0.9	0.9	n.a.
κ	n.a.	n.a.	7.0

489 In contrast to the navigation task, the state-action values of the experts are not initialized to a positive value, and are
490 worth 0.0 at the start of the experiment.

491 4.7 Results of the experiments without human intervention

492 To evaluate the performance of the simulated robots, we reuse the color code of the navigation experiment: Red for the
493 MF-only robot, blue for the MB-only robot, green for the random coordination robot (MC-Rnd) and purple for the
494 robot that coordinates the two experts using the arbitration criterion that we have proposed (MC-EC).

495 The interest of this experiment is to evaluate the contribution of meta-control in a task where a robot can interact with a
496 human. We will start by evaluating the performance of the robots without human intervention, then with the two types
497 of human intervention defined above.

498 In [Dromnelle et al. \[2020a\]](#) we studied the evolution of the average performance of the different robots when the
499 human does not interact with them. As in the navigation experiments, the MF-only robot was the one with the
500 worst performance. Interestingly and in contrast with the navigation experiment, we had observed that the maximum
501 performance was achieved by robots doing meta-control (MC-EC and MC-Rnd) rather than by the MB-only robot.
502 Importantly, the MC-EC robot displayed a much lower computational cost than that of the MC-Rnd robot. Finally, we
503 found that these properties were obtained through a different temporal pattern of expert selection: We observed a very
504 short guidance phase by the MB expert, followed by the guidance phase of the MF expert. Because the state-action
505 values were initialized to 0.0 at the beginning of the experiment, we did not observe the exploratory phase of the MF
506 expert that we observed during the navigation experiment.

507 These results thus constituted a first step of validation of the genericity of the proposed method in a simple HRI task. In
508 such a case, when the robot has to learn on its own without human intervention, it can be useful to combine MB and
509 MF RL to get an optimal performance while minimizing the computational cost.

510 4.8 Meta-control provides robustness to errors in humans’ teaching signals

511 Next, we evaluate the architecture when the human intervenes in the form of two possible types of teaching signals:
512 *Congratulations* or *Takeover*. The main messages from the analyses that will be presented hereafter are that:

- 513 • The meta-controller of MC-EC robots enables them to get a robust performance in the task independent from
514 whether the human intervenes or not. Only MF-only robots require human intervention to bootstrap their
515 learning performance in this task, while all robots with an MB expert can already learn fast (but note that
516 human interventions are still beneficial in the *Takeover* case, see Fig. S10).
- 517 • The meta-controller of MC-EC robots provides them with robustness with respect to errors that humans can
518 make during their interventions (Fig. 9): We tested different percentages of errors made by the humans when
519 congratulating the robot or when taking-over to show the robot was is the right action to perform; We also
520 tested different omission rates in human’s teaching signals. The deterioration of performance caused by

omitted (Fig. 9C) or misleading (Fig. 9B) interventions was mostly penalizing the MF-only robot, while being mitigated in the MB-only, MC-Rnd and MC-EC robots, thanks to the MB expert.

- The meta-controller of MC-EC robots minimizes computational cost: Its cost was more than four times lower than that of the MC-Rnd, and ten times lower than the one of the MB-only. (Fig. 8B).
- Finally, overall the *takeover* human interventions were more efficient than the *congratulation* ones (compare Online Resource Suppl. Fig. S10 with Online Resource Suppl. Fig. S8), as they allowed to reach larger cumulated reward levels for all the configurations of the architecture (MF-only, MB-only, MC-Rnd and MC-EC). This required 300 iterations in the worse case (MF-only) but was faster for robots incorporating a MB expert (150 interactions). Quite naturally, increasing the number of such interventions increased the cumulated reward up to a ceiling value (Online Resource Suppl. Fig. S10).

In the next subsections, we present more detailed analyses of these results to illustrate the task-independent nature of our coordination model, its generalization to an environment composed of about three times more states than for the navigation task (Section 3), as well as its ability to cope with the volatility of human behavior. Despite these many differences, we reused the same parameters that were optimized for the navigation task, in order to show the generic and task-independent nature of our learning and meta-control system.

4.8.1 Results with human intervention of the *Congratulation* type

Cumulative reward. In Online Resource Suppl. Fig. S8, we can visualize the performance of the different robots at the last iteration (the 10000th) for different durations of human interventions of *Congratulation* type. The human begins to intervene directly after the end of the babbling period. We notice that only the MF-only robot seems to be strongly impacted by human intervention. The other robots have their performance slightly improved for long human interventions, but not for null and short human interventions. A *Kruskal-Wallis* test determined that, for the MB-only and MC-Rnd robots, at least some performances for different intervention durations were significantly different (Kruskal-Wallis test, p-value MB-only = 5.66×10^{-5} and p-value MC-Rnd = 0.002). In order to identify which performances were significantly different from the others, we performed multiple comparison procedures through the *Dunn* test [Dunn, 1964] with *Bonferroni corrections* (Online Resource Suppl. Fig. S7). If four performance comparison tests for the MB-only and MC-Rnd robots indeed had a p-value below the significance threshold of 0.05, we note that the effect seems above all to be due to the variability of the data. This is evidenced by the proximity of these p-values to the threshold of 0.05 compared to those of the MF-only robot. For example, for the MB-only robot, the performance relative to the duration of 10 interventions stands out, for no specific reason. Conversely, the effect of the *Congratulation* type intervention on the performance of the MF-only robot had an effect proportional to the duration of the intervention, which makes sense.

We then compared the performance between MF-only, MB-only, MC-EC and MC-Rnd robots. A *Kruskal-Wallis* test between the performances of the four robots for an intervention duration of 500 iterations confirms that at least one of the performances was significantly different from the others (p-value = 2.99×10^{-7}). Finally, a *Dunn* test allows us to see that the performance of the MC-EC robot at an intervention time of 500 iterations was significantly different from the performance of the MC-Rnd robots (p-value = 0.0408), MB-only (p-value = 9×10^{-5}) and MF-only (p-value = 3.94×10^{-7}) at the same duration of intervention. The performance of the MC-Rnd robot was also significantly different from the performance of the MF-only robot (p-value = 0.042) while the MF-only and MB-only robots had indistinguishable performances (p-value = 1.0).

For the moment, we have therefore shown that human intervention of the *Congratulation* type seems to be useful only to the MF-only robot, which only embeds a model-free expert. In contrast, only the MC-EC robot achieves maximal performance. Importantly, the MB-only, MC-Rnd and MC-EC robots, which all embed a model-based expert, do not need human intervention to improve their performance. In other words, the interest of the hybrid MB-MF architecture that we propose here is to be more robust to short human teaching interventions, and thus to produce optimal performance in this simple cube tidying task even for cases where real human participants were bored to provide the robot with a long supervision.

Computational cost. Next, we examine the advantages of the proposed architecture in terms of computational cost reduction. Figure 8B allows us to compare the cumulative costs of the inference processes of the different robots at the end of the experiment in the case where the human does not interact with the robot, and in the case where the human congratulates the robot or takes over. Overall, we can say that the help provided by the human seems to slightly offload the robot in computational cost. This is especially observable for the MB-only robot (which in fact performs more expensive computations than the other robots). In any case, the displayed cost of the MC-EC robot is again extremely low compared to those of the MB-only and MC-Rnd robots.

574 Overall, we can conclude that the MC-EC robot is capable, at minimal cost, of compensating for the absence of human
 575 intervention. When the human is present and interacts with the robot, the cost of the MB expert decreases, a sign that it
 576 performs less expensive computation. When the duration of the intervention is long, the MF-only robot is fully capable
 577 of performing the task efficiently at a very low computational cost. However, as soon as the duration of the intervention
 578 decreases, its performance drops. This is when the MB expert behaves like a “backup expert”, which allows the robot
 579 not to be dependent on the human. In a situation where the presence of the human is uncertain, the MC-EC robot is
 580 therefore the ideal robot.

581 **Humans that make omissions.** In order to confirm this reasoning, we performed another set of simulations where
 582 the simulated humans had a tendency to omit to congratulate the robot from time to time. In other words, the human
 583 behavior is now simulated with a certain degree of stochasticity, so that the robot is rewarded by the human only a
 584 proportion of the required feedback (from 0%, 10%, .. up to 100% of the time). If omitting has a clear effect on the
 585 performance of the MF-only robot (Online Resource Suppl. Fig. S9, first row), bringing it back to the performance of
 586 non-intervention, the other robots deal with it without much concern (Online Resource Suppl. Fig. S9, three bottom
 587 rows). This is because, as we have previously seen, their performance is already high without intervention, and remains
 588 here largely unaffected by the intermittent absence of human feedback.

589 **Humans that make mistakes.** Finally, to test the adaptability of these different robots to slightly more realistic
 590 humans, we made a last series of simulations where humans could make errors. Within the framework of the
 591 *Congratulation* type intervention, an error consists in congratulating a bad action of the robot (for example putting
 592 the red cube into the green container). All the system configurations suffer a performance degradation (Fig. 9A), the
 593 MF-only configuration is the most affected one. This corroborates our previous observations regarding the dependence
 594 of the MF-only robot, and therefore that of the MF expert, on human intervention. Again, using an MB expert is very
 595 beneficial for the robot. In all four cases, and even if the performance degradation of the other robots is minimal, we
 596 observe that at very high human error rates, the quantity of cumulative rewards at the end of the experiment remains
 597 lower than when the human never makes mistakes or never interacts with the robot. This is because during this 500
 598 iterations period of interventions, all the system configurations struggle to accumulate the reward despite human
 599 detrimental interventions, which therefore creates a performance delay compared to the robots not interacting with the
 600 human or with a human not making mistakes.

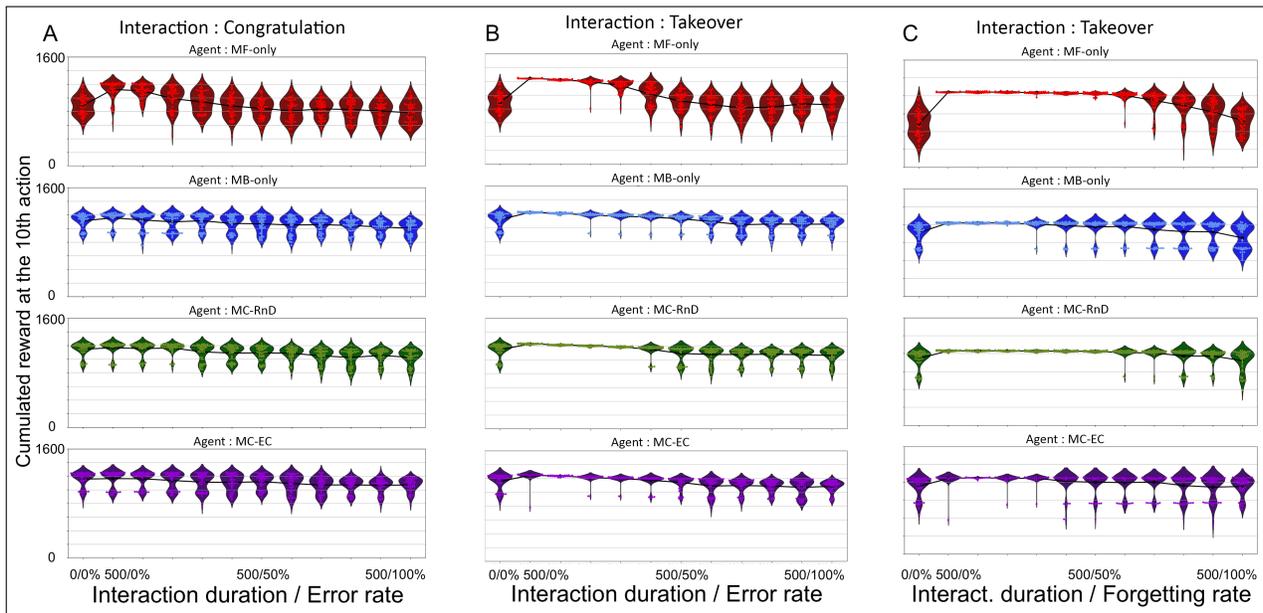


Figure 9: Reward accumulation results in the HRI teaching task. **A.** Case where humans provide erroneous *congratulation* feedback with increasing error rates. **B.** Case where humans provide erroneous *takeover* feedback with increasing error rates. **C.** Case where humans omit to provide *takeover* feedback with increasing omission rates. Dots report the accumulated after 10,000 simulation timesteps, for 50 simulations. First row (red): MF-only robot; second row (blue) MB-only robot; third row (green): MC-Rnd robot; fourth row (purple): MC-EC robot.

601 Importantly, using our arbitration criterion allows the MC-EC robot not to be dependent on the human to achieve the
 602 objective that has been set for it, but also to absorb its potential errors more effectively. In other words, the proposed
 603 architecture allows the simulated robot to be more robust to human errors in this task.

604 4.8.2 Results with human intervention of the *Takeover* type

605 Unlike the *Congratulation* type intervention, we can see in Online Resource Suppl. Fig. S10 that the *Takeover* type
 606 intervention has an effect on the performance of each robot, although the performance effect on the MF-only robot
 607 remains larger. For the other three robots, we can see that intervening over a period of more than 100 iterations no
 608 longer significantly increases performance. A *Kruskal-Wallis* test between the performances of the four robots for an
 609 intervention duration of 500 iterations confirms that at least one of the performances is significantly different from
 610 the others ($p\text{-value} = 6.10 \times 10^{-35}$). A *Dunn* test finds that at an intervention time of 500 iterations the performance
 611 of the MC-EC robot is significantly different from the performance of the MC-Rnd robot ($p\text{-value} = 5.84 \times 10^{-16}$),
 612 MB-only ($p\text{-value} = 1.73 \times 10^{-32}$) and MF-only ($p\text{-value} = 2.50 \times 10^{-04}$). The performance of the MC-Rnd robot is
 613 also significantly different from the performance of the MF-only ($p\text{-value} = 1.53 \times 10^{-04}$) and MB-only ($p\text{-value} =$
 614 1.25×10^{-03}), which both also have a significantly different performance ($p\text{-value} = 1.44 \times 10^{-04}$). These performances
 615 exceed on average the 1200 accumulated rewards, *i.e.*, more than the maximum performances obtained by the robots
 616 within the framework of the *Congratulation* type intervention (Online Resource Suppl. Fig. S8). In summary, all robots
 617 have different performances, and again, the MC-EC robot is the best of all.

618 We can explain the high performance of the *Takeover* type intervention by the fact that the decision of the human replaces
 619 that of the robot in 100% of cases, whereas in the case of the intervention of *Congratulation* type, the decision-making
 620 process, although biased in favor of the human, is still subject to a probabilistic treatment through the *softmax* function
 621 (3), which can at times select a non-optimal action. In addition, the *Takeover* type intervention acts on the behavior of
 622 the robot at the iteration on which it is performed, while the *Congratulation* type intervention has an influence on the
 623 robot behavior only the next time the robot performs the state-action combination that the human praised.

624 In Figure 8B, we can see that the cumulative cost values are as low as in the *Congratulation* type intervention: The more
 625 efficient the human intervention, the less the MB expert needs to do expensive calculations. Finally, in [Dromnelle et al.](#)
 626 [\[2020a\]](#) we observed the same guidance phases of the two experts as for the *Congratulation* and No-intervention cases.

627 If we observed previously that the robots MB-only, MC-Rnd and MC-EC were not impacted by humans omitting to
 628 intervene, because the human did not provide any significant assistance to the robots equipped with an MB expert,
 629 things are logically different here since the intervention brings clearer help. Indeed, we can see in Figure 9C that at high
 630 omission rates, the performance of all the robots degrades, even if again, the degradation of the performance of the
 631 robot MF-only remains much more important. Of the three other robots, the MC-EC robot seems to be the one doing
 632 the best when faced with the oversights of its human partner.

633 Finally, we again put the robots in front of humans making mistakes (Fig.9B). In the context of the *Takeover* type
 634 intervention, this means that the human takes control of the robot arm to put the cube in the wrong container, or to
 635 remove the cubes from the containers of the right color. Here the results are quite close to those observed in Figure 9A:
 636 we observe an overall degradation of the robots' performance, again much more intensive in the case of the MF-only
 637 robot. As before, at a very high human error rate, the quantities of cumulative rewards at the end of the experiment are
 638 lower than these same quantities when the human never interacts with the robots. This is due to the performance lag
 639 accumulated during the 500 iterations of erroneous interventions.

640 With our arbitration criterion, the robot benefits from the human performing a *Takeover* to even better achieve the
 641 objective that has been assigned to it, contrarily to *Congratulation* interventions, that are less effective. This superiority
 642 of *Takeover* over *Congratulation* has been observed in other studies [[Knox et al., 2011](#)]. It is therefore to be preferred.
 643 Nevertheless, as with the *Congratulation* type intervention, the combination of MF and MB experts can absorb human
 644 errors more effectively.

645 5 Experiment 3: Human-robot interaction with human as cooperator

646 In the third experiment, we evaluate our coordination system in a human-robot cooperation task different from the
 647 previous one: While in Experiment 2 the robot could learn with or without human intervention, here the robot necessarily
 648 needs help from the human. All the following results are previously unpublished.

649 We first present the new version of the simulated cube storing task, and the way in which we modeled the human partner
 650 with whom the robot must now cooperate to achieve its goal. In the second part, we present the results obtained and
 651 show that in a situation where the partner can turn into an adversary, our coordination system is no longer able to
 652 maintain a high level of performance. To circumvent this problem linked to a natural algorithmic asymmetry between

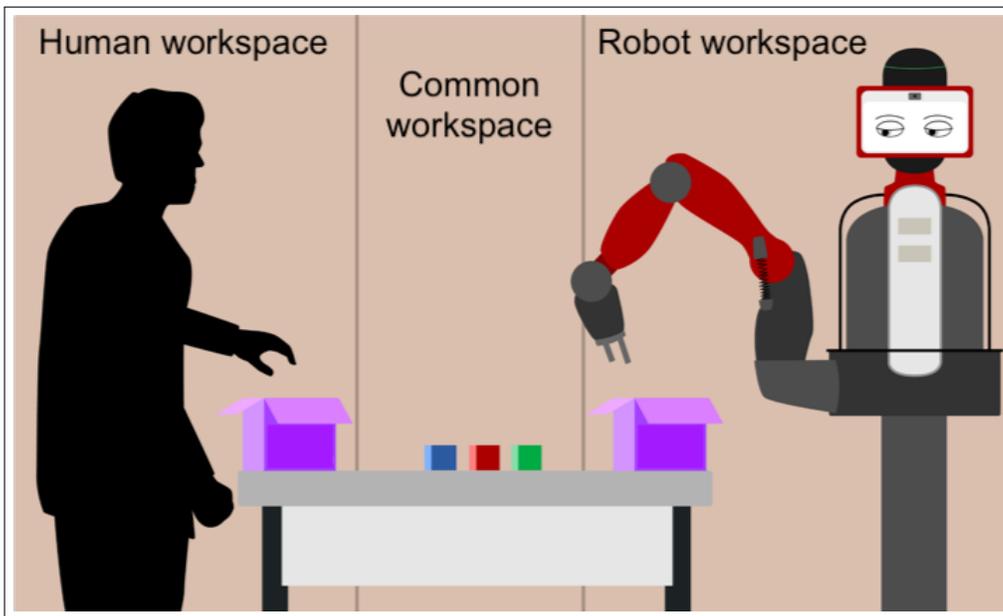


Figure 10: Illustration of the Human-Robot Cooperation task. Figure by Dromnelle, Renaudo, Khamassi and Girard (2022); available under a CC-BY4.0 licence (<https://doi.org/10.6084/m9.figshare.21031723>).

653 the MF and MB experts, and not to the human partner, who is only the revealer, we propose an inexpensive solution,
 654 under the form of adding a context switching detection mechanism to the robot. With this mechanism, the robot is again
 655 able to maintain a high level of performance while still greatly reducing its computational cost.

656 **5.1 Material and methods**

657 **5.1.1 Simulated environment and robot**

658 This experiment is also carried out in simulation only. The same robot as the one presented in Experiment 2 faces a
 659 table. This time, the table is divided into three distinct spaces: A space accessible to the human only, a common space
 660 and a space accessible to the robot only. The human space and the robot space each contain a container, referred to as
 661 the human’s container and the robot’s container. Three colored cubes are available on the table (Fig. 10). This task is
 662 inspired by those of Alami et al. [2011] and Renaudo et al. [2015a].

663 Unlike in Experiment 2, here the robot’s first objective is to learn how to put each cube in its own container. When
 664 this is done, the robot gets a scalar reward, and the cubes are automatically returned to the human’s container. Like
 665 in Experiment 1, we make the task non-stationary by introducing a change of objective during the experiment. More
 666 precisely, at the 5000th iteration, the robot must now learn to put each cube in the human’s container. When this is done,
 667 the cubes are automatically returned to the robot’s container.

668 We also test a variant of this experiment with another pair of objectives. The cubes’ position has to be swaped: first, the
 669 red and the blue start in the robot container and have to be put in the human container, while the green starts in the
 670 human container and must end in the robot container; then, the starting position is reversed (red and blue in the human
 671 container, green in the robot container) and positions still have to be swaped.

672 Unlike the task in Experiment 2, where the robot could carry out the experiment without the help of the human, the
 673 participation of the human is essential here, since the robot does not have access to the human’s side of the table. For
 674 this reason, we speak here of *cooperation with humans*, and no longer just of *human intervention*.

675 **5.1.2 Robot state and action spaces**

676 The state space is again a discrete state space. A state always represents the position of the three colored cubes. Each of
 677 the cubes can be located: In the human’s container, in the common space, in the robot’s container, in the human’s hand
 678 and in the robot’s hand. If we remove the states where the robot and the human are holding several cubes at the same
 679 time, this represents a total of 99 states, which is 13 less than the task of Experiment 2.

680 Concerning the action space, the robot can perform 6 classic actions: take the red cube, take the green cube, take the
 681 blue cube, place the cube held in hand in its container, place the cube held in hand in the common area, skip its turn. In
 682 addition, there are 2 interactive actions, allowing the robot to give the cube held in hand directly to the human (if his
 683 hand is empty) or, conversely, to ask the human to give the cube he is holding (if the robot's hand is empty), leading to a
 684 total of 8 actions.

685 As we will see in the next subsection, the human is considered in this experiment as a decision-making agent, and
 686 therefore has its own state space equivalent to that of the robot.

687 5.1.3 Simulated human

688 In Experiment 2, the human could from time to time interact with the robot. Here, its participation in the task is essential
 689 to the success of the robot. To model human behavior, we opted for a version of our MB-only robot with a complete
 690 transition model. We consider that if the robot must first learn the consequences of its actions during the babbling phase,
 691 the human already knows, for example, that when he takes the red cube from his container, the cube is now located in
 692 his hand.

693 5.2 Pre-experimental babbling phase

694 A babbling phase, where the robot and the human can manipulate the cubes in the absence of reward precedes the
 695 experiment. We chose to add this pre-learning phase for the same reasons as those mentioned in Experiment 2. This
 696 time, on the other hand, rather than evaluating the robot's performance using our arbitration criterion (MC-EC) at
 697 different babbling durations, we evaluate them at different percentages of transitions explored (Fig. 11). We choose
 698 an exploration percentage of 80% (yellow curve) for the first pair of objectives and an exploration percentage of 70%
 699 (orange) for the second. These values correspond to those above which continuing to explore no longer allows the
 700 reward to accumulate quicker over time. Again, we could choose to give the robot a more or less complete transition
 701 model before the start of the experiment or to reuse the transition model built by the robot before the first experiment
 702 for all subsequent ones, in the case of real experiences where time is not an unlimited resource.

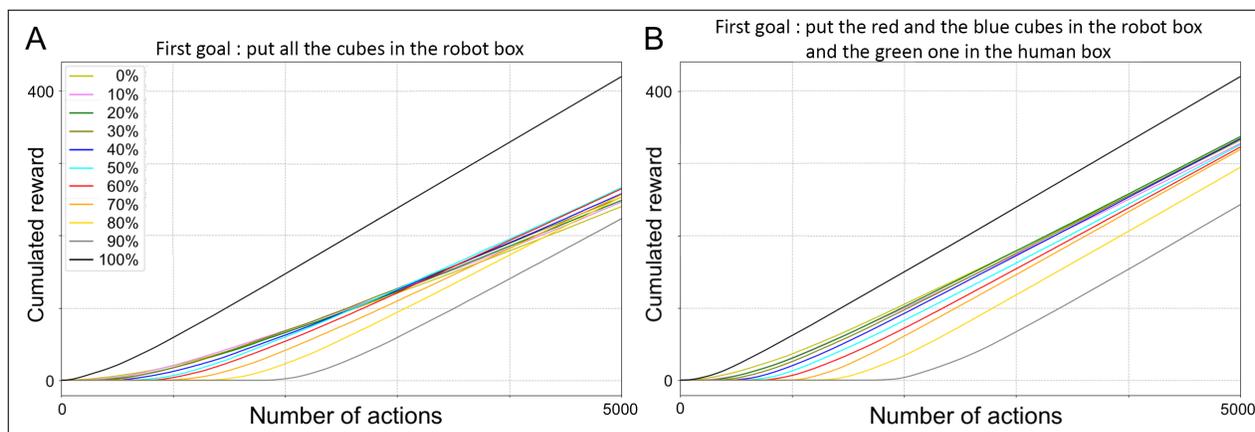


Figure 11: Sizing the babbling phase. **A.** Average performance of 50 simulations of the MC-EC robot for different percentages of transitions explored during the babbling phase and for the first combination of objectives (tidying task). **B.** Average performance of 50 simulations of the MC-EC robot for different percentages of transitions explored during the babbling phase and for the second combination of objectives (swapping task). Performance is defined as the robot's ability to accumulate reward over the duration of the experiment (5000 actions).

703 5.2.1 Expert parameters

704 We reuse again the same set of parameters used in the navigation task and the human-robot interaction task for each
 705 of the experts and for the meta-controller (Table 2), in order to show the robustness of our learning and meta-control
 706 system. The parameters of the simulated human are identical to those of the robots.

707 The action-state values of the experts and the human are again initialized to 0.0 at the start of the experiment.

Table 2: Selected values of expert and meta-controller parameters in the tidying task in cooperation with a human.

Param	MB	MF	MC
α	n.a.	0.6	n.a.
τ	0.02	0.02	0.02
γ	0.9	0.9	n.a.
κ	n.a.	n.a.	7.0

708 5.3 Results

709 To evaluate the performance of simulated robots, we reuse the color code from previous experiments: Red for the
 710 MF-only robot, blue for the MB-only robot, green for the random coordination robot (MC-Rnd) and purple for the
 711 robot that coordinates the two experts using the arbitration criterion that we have proposed (MC-EC).

712 The interest of this experiment is to evaluate the contribution of meta-control (expert coordination) in a task where a
 713 robot must necessarily cooperate with a human to progress, but also to push our architecture to its limits.

714 5.3.1 When the partner becomes an adversary

715 With the first pair of objectives (tidying task) during the first phase of the experiment, the performance of the MC-EC
 716 robot again equals that of the MB-only robot (Fig. 12.B), for a computational cost divided by three (Fig. 12.D).
 717 Unfortunately, as soon as the objective changes, the MC-EC robot no longer manages to accumulate as many rewards
 718 as the MB-alone robot, and is even caught up by the MF-only robot, hitherto considered to be the less efficient. We
 719 observed exactly the same tendencies with the second pair of objectives (swapping task, Online Resource Suppl.
 720 Fig. S11.A and B). In previous experiments, we had never faced such a drop in performance of the MC-EC robot. To
 721 explain it, we need to look at what exactly happens at the 5000th iteration.

722 For the robot and the human, the 5000th iteration is just another iteration: The objective changes without them being
 723 informed. Not knowing that the objective has changed, the two partners will continue to pass the cubes as if nothing
 724 had happened. When they finally manage, for example, to put all the cubes in the robot’s container (in the case of the
 725 first pair of objectives), no reward is issued to them and their R reward models are therefore modified accordingly.
 726 Following this, as soon as the inference processes of the MB experts of the MC-EC robot and the human are activated,
 727 the state-action values of the MB experts get reset to 0.0 via the natural action of the dynamic programming algorithm
 728 *Value Iteration* (Eq. 2).

729 However, before the 5000th iteration, the behavior of the MC-EC robot is mainly directed by the MF expert (Fig. 12F
 730 and Online Resource Suppl. Fig. S11.C), which is not able to reset its action-state values in one go. Indeed, it will take
 731 many iterations and passages through the states leading to the rewarded state for the action-state values to decrease
 732 following the absence of reward. The problem is therefore the following: after realizing that the objective has changed,
 733 the simulated human will go back to exploring the environment in order to find the new rewarded state, or even try
 734 to fulfill the new objective if he succeeds. To discover it, while the robot MC-EC, whose behavior is directed at this
 735 moment of the experiment mainly by its expert MF, will continue to try to achieve the first objective, resulting in
 736 destructive interferences. The robot will, for example, ask the human to give the currently held cube, so as to put it in
 737 the robot’s container, before the human can put it in its own container, therefore preventing the obtention of reward (and
 738 thus the identification of a new goal). On the contrary, the human may manage to put some cubes in his own container,
 739 preventing the robot to reach the previously rewarded state, where it would observe the absence of reward, generating
 740 large negative reward prediction errors that would start to modify the behavior of his MF expert. Here, the partner
 741 turned adversary highlights an algorithmic difference whose effect we had already observed in the navigation task of
 742 Experiment 1.

743 Indeed, this inability of the MF expert to reset his state-action values in the same way as the MB expert was the cause
 744 of a "spike" in the selection probability of the MF expert (Fig. 12C) which correlated with the very slight lag in reward
 745 accumulation that the MC-EC robot took on the MB-only robot (Fig. 12A). As a reminder, our arbitration criterion is
 746 a compromise between the cost of the inference process and the quality of the learning defined as the entropy of the
 747 distribution of the probabilities of selection of actions. Concretely, the closer the state-action values of a state are to
 748 each other, the greater the entropy will be, and the lower the learning quality will be. When the MB expert resets his
 749 state-action values, he also resets his learning quality. The MF expert not being able to do so, he will de facto become
 750 the expert with the best learning quality, and therefore the expert controlling the behavior of the robot, whereas the
 751 judicious behavior would be precisely to stop playing the first objective.

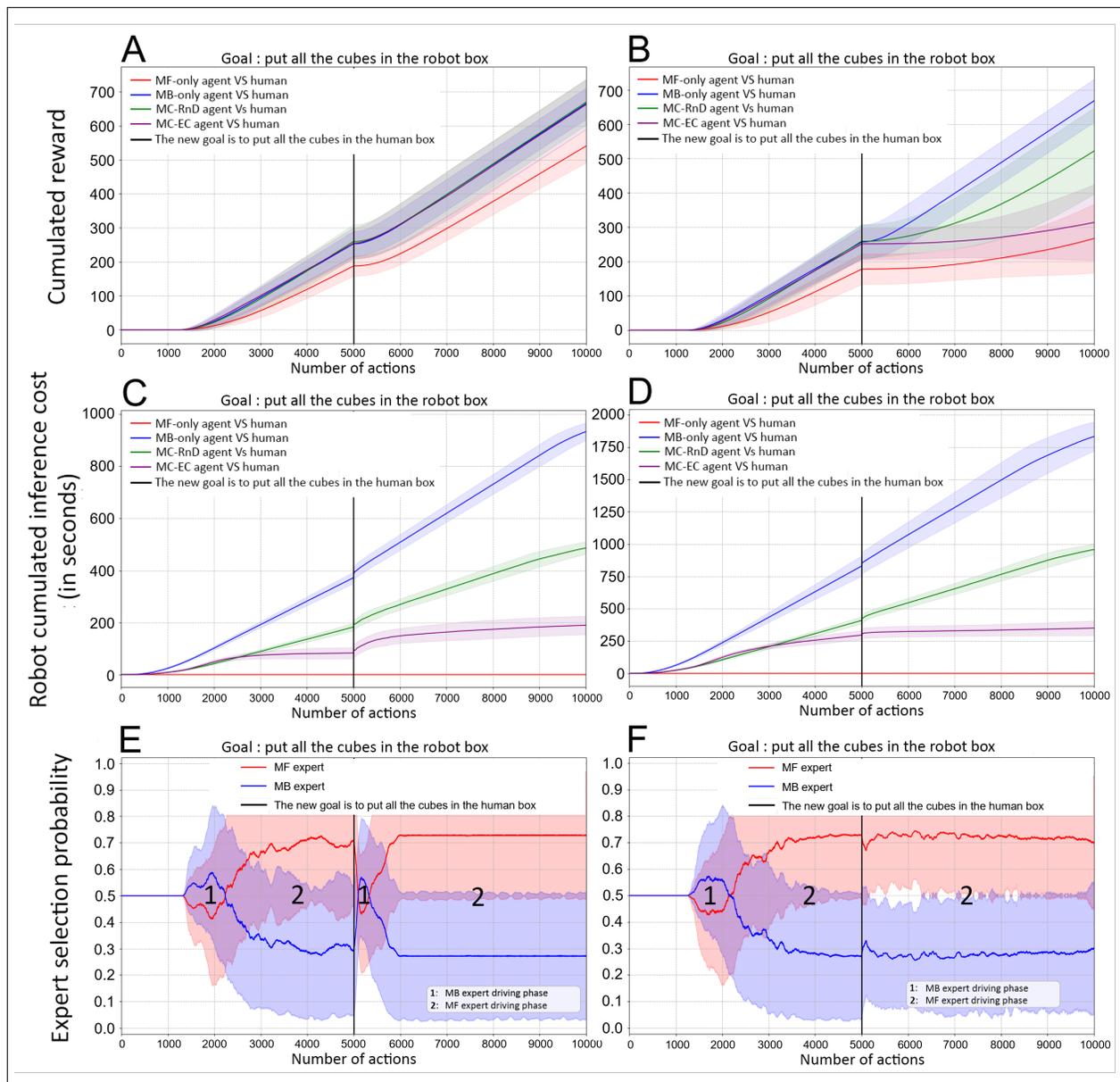


Figure 12: Tidyng task results with (A, C,E) or without (B, D, F) a context change detection mechanism: **A,B**. Average performance for 50 simulated experiments. **C,D**. Average computational cost for 50 simulated experiments. **E,F**. Average probability of selection of experts by the meta-controller of the MC-EC robot for 50 simulated experiments. We use standard deviation as an indicator of dispersion in all three figures.

752 In both experiments, the observation is therefore the same: if the environmental change implies a modification of the
 753 reward models of the MB experts, the algorithmic asymmetry of the MF and MB experts gives rise to a period when
 754 the MF expert directs the behavior of the robot more than it should. If this did not prevent the robot from maintaining
 755 good performance in the navigation task, the MB expert is no longer able to regain control of the robot’s behavior here
 756 (Fig. 12F and Online Resource Suppl. Fig. S11.C) and therefore remains stuck in MF expert guidance phase 2.

757 Note that, compared to the navigation task of Experiment 1, we do not observe here the exploratory phase of the MF
 758 expert. As a reminder, the existence of this phase was due to the difference in learning methods of the two experts, at
 759 the origin of the fact that the state-action values of the expert MF decreased slightly more than those of the expert MB
 760 expert. Here, the state-action values of the experts being initialized at 0.0 at the start of the experiment, and not at 1.0
 761 as in the browsing experiment, this effect of algorithmic asymmetry is not observed.

762 5.3.2 Context change detection

763 To counter this problem, we equipped our robot with a mechanism allowing it to automatically detect changes in
 764 goals by taking into account only the evolution of its action-state value models. To do this, we relied on the *cosine*
 765 *similarity* to evaluate the similarity of two n-dimensional vectors by determining the cosine of their angle. Generally
 766 used as a measure of similarity between two documents, we use it here to measure the similarity between two vectors of
 767 state-action values:

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \quad (7)$$

768 where A is the state-action value vector of the previous state before it was updated by the MB expert and B is the
 769 state-action value vector of the previous state after its update by the MB expert. Note that the vector values have all
 770 been multiplied by 100 and the null values have been replaced by very small values to avoid division by 0. If the two
 771 vectors are identical, θ is 1.

772 We already used this measure in [Caluwaerts et al., 2012b], where the *cosine similarity* was computed on vectors
 773 containing the Q-values of the MB expert. Concretely, every time the MB expert carries out its inference process, it also
 774 computes the *cosine similarity* θ of the Q-value vectors before and after this update, and compares it to a threshold. If θ
 775 is lower than this threshold, the MB expert then sends an additional signal to the meta-controller (arrow t1 in Fig. 1),
 776 which will take care of sending a signal to the MF expert to request a reset of its Q-values (arrow t2). The value of θ
 777 will necessarily decrease due to updates of state-action values in three cases:

- 778 • When the robot first finds the reward. In this case, resetting the state-action values of the MF expert is not a
 779 problem, since all of them are already null.
- 780 • When the robot reaches the previously rewarded state and does not obtain a reward, the moment we are most
 781 interested in.
- 782 • When the robot first finds the new reward. In this case, resetting the state-action values of the MF expert again
 783 is not a problem, since they have all been reset previously.

784 In the end, more than a mechanism allowing it to automatically detect a change of objective, the *cosine similarity*
 785 also allows the robot to detect the appearance of a new objective: it is therefore a mechanism for detecting changes of
 786 context, as pointed out by Caluwaerts et al. [2012b]. In our algorithm, when the robot discovers that the rewarded state
 787 no longer yields a reward, the action-state values of its expert MF are reset. Instead, we could allow it to store them
 788 in memory, so that we can potentially reuse them if the formerly rewarded state becomes rewarded again later in the
 789 experience, which is not the case here.

790 Of course, the functionality of the mechanism depends on the threshold against which the value of the *cosine similarity*
 791 θ will be compared. To define it, we looked over 200 simulations at the value of the *cosine similarity* at the iteration
 792 following that in which the robot reaches the formerly rewarded state for the first time. θ was in 100% cases less than
 793 or equal to 0.611 for the first pair of objectives (100 simulations), and 0.706 for the second (100 simulations). We
 794 have chosen a common threshold of 0.7. The histograms of the frequencies of the different values obtained from θ
 795 for an experiment of each of the pairs of objectives (Fig. 13) reveal that most of the time, the values of θ are worth
 796 1.0, a sign that during the experiments, the values of the state-action pairs of the MB expert do not evolve much. In
 797 the tidying task (Fig. 13A), the values of θ were lower than 0.7 four times (3 of 0.558 and 1 of 0.611), and in the
 798 swapping task (Fig. 13B), it happened five times (2 of 0.61 and 3 of 0.666). In both cases, this therefore corresponds to
 799 more event than the 3 ones we identified above as being actual context changes (discovery of reward, discovery of the
 800 disappearance of the reward, discovery of the new reward). This means that sometimes, the values of state-actions of

801 the expert MB strongly evolve without this being linked to a change of context, but for example rather to the discovery
 802 of a new unexplored state or transition. Depending on the defined threshold, the robot can therefore trigger false alarms,
 803 mistaking this “brutal” update for a change of context and reset the state-action values of the MF expert when it should
 804 not.

805 However, a these rare false alarms do not seem to have any negative effect on the robot’s performance: With the context
 806 change detection mechanism and a threshold of 0.7, the performance of the MC-EC robot is now identical to that of
 807 the MB-only robot (Fig. 12A). Just after the goal change, the computational cost of the inference process increases
 808 (Fig. 12C), a sign that the MB expert takes control of the robot’s behavior to enable it to better cope with environmental
 809 change. Figure 12E confirms this with the reappearance of the second guidance phases of the MB expert, absent from
 810 the experiments carried out without the detection mechanism context changes. Again, these results were replicated with
 811 the second pair of objectives (swapping task, Online Resource Suppl. Fig. S12).

812 5.4 Conclusion

813 In this last experiment, we evaluated our learning expert coordination model in a simulated human-robot cooperation
 814 task where the robot must actively cooperate with the human to achieve its objectives. The human is no longer simply
 815 present to help the robot improve its performance, but becomes a real partner. Again, we reused the parameters
 816 optimized for the navigation task, in order to show the robustness of our learning and meta-control system.

817 In this experiment, the robot was confronted with a problem already observed in the navigation task, but which until
 818 now did not prevent it from progressing: the inability of the MF expert to reset its action-state values after the change of
 819 objective compared to the MB expert. Here, due to the presence of a human not being affected by this problem, the two
 820 partners can become adversaries for a time, which leads to a drastic drop in the robot’s performance. Here, the human
 821 is not the problem, but simply its revelator. To counter this, we have therefore added a mechanism to detect context
 822 switches, allowing the robot to automatically reset the state-action values of its MF expert when necessary. With this
 823 mechanism, the robot using our arbitration criterion, once again obtains the same level of performance as that of a robot
 824 controlled solely by a model-based learning algorithm, while drastically reducing its computational cost (Fig. 12B,D).

825 Finally, we illustrated again with this human-robot cooperation task the generic and task-independent nature of our
 826 coordination model, and an efficient and inexpensive solution allowing it to circumvent a problem that can arise during
 827 abrupt changes in the task objectives. These results further highlight the robustness of the proposed method.

828 6 Discussion

829 We analyzed the behavior of a three-layered robot cognitive architecture integrating human-inspired mechanisms for the
 830 coordination of model-based (MB) and model-free (MF) reinforcement learning modules. Its main novelty lies in the
 831 use of the explicit online measure of both performance and computational cost of each system, so as to give control
 832 to the system with the best current trade-off between the two. The goal of this approach is to maximize behavioral
 833 flexibility, while enforcing computational (and thus energetic) frugality.

834 Behavioral flexibility was assessed in three main experiments: an indoor navigation task, a HRI task where the human
 835 teaches the robot and a HRI task where the human and the robot must cooperate. All these tasks were non-stationary, as
 836 an unsignalled change of the goal or of the available transitions, always happened in the course of learning. We kept the
 837 parameters of the system identical from one task to another.

838 Heavy computations consume both time and energy, resources that can be essential for robots: autonomous robots that
 839 rely on their sole (and usually limited) computational resources cannot always afford the time required by a complex
 840 computation, fast reactions can be necessary in many realistic settings, to avoid damaging the environment or oneself;
 841 even when time is not a crucial issue, heavy computations consume energy, a resource that is even more crucial to a
 842 mobile robotic platform. Our RL module coordination system is the first one in robotics, to our knowledge, to explicitly
 843 take into account the actual computational costs to arbitrate between modules. In computational neuroscience, some
 844 earlier models [Keramati et al., 2011, Pezzulo et al., 2013] proposed to evaluate the value of gaining better information
 845 from a MB module, versus the cost of performing inference with this MB module, but they were tested in toy problems,
 846 with shallow MDPs, with deterministic transitions, and with the model already knowing the transition function. Here
 847 we used a more empirical approach, by evaluating the real temporal costs induced by the use of MF and MB learning
 848 modules.

849 The comparison with DQN, made in the navigation experiment, showed that using end-to-end RL has a computational
 850 cost not compatible with robotic constraints, and that thus building and using a data representation adapted to the task
 851 at hand reduces the burden on the RL part of the system, allowing for low-cost on-the-fly learning. Nevertheless, the

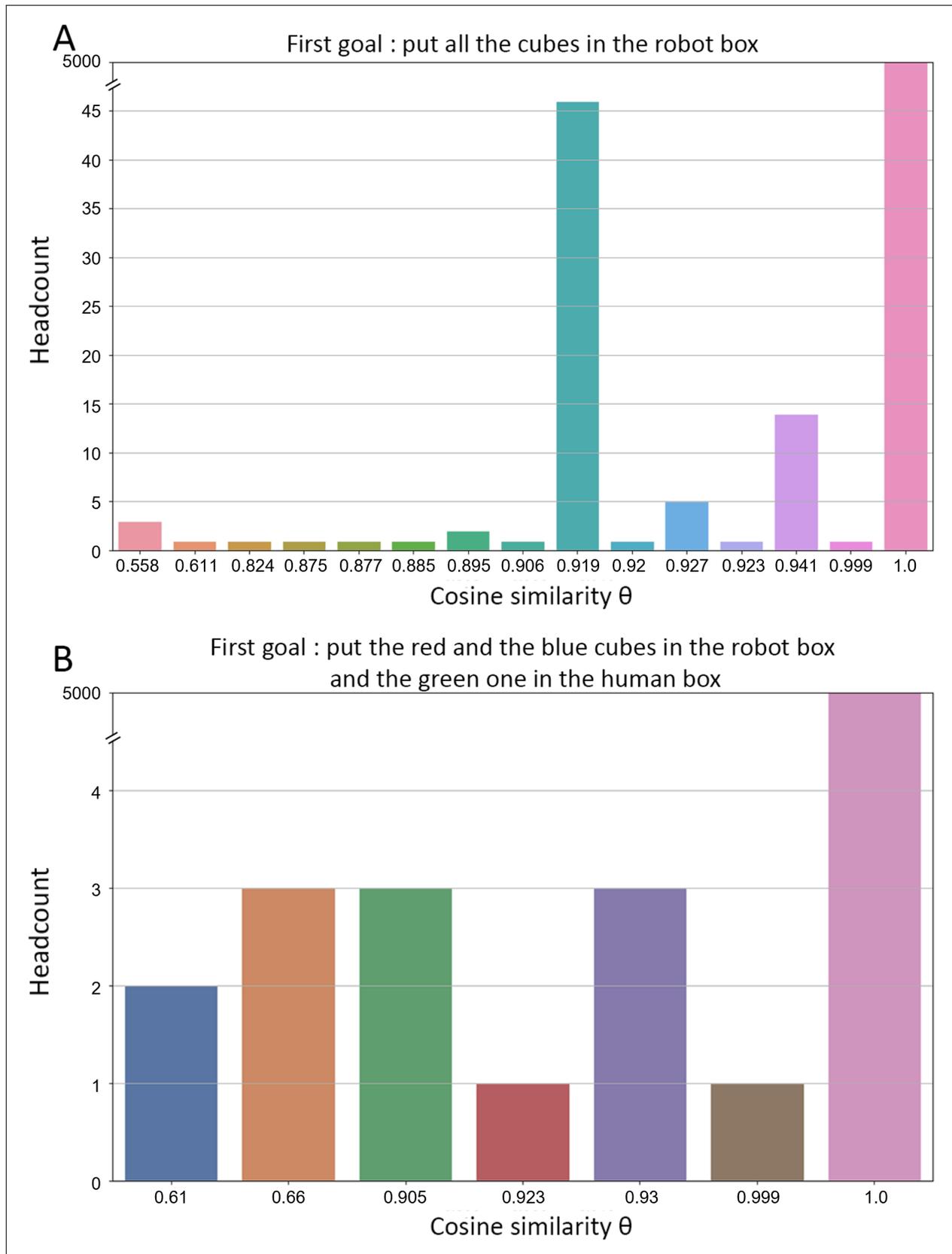


Figure 13: Values taken by the cosine similarity θ , used to parameterize the context change detection threshold. Histograms report the frequency of θ values measured in two 10,000 iteration-long simulations, using: in **A**, the first pair of objectives; in **B**, the second pair of objectives. The robot and the human play on a turn-based basis, so that makes a total of 5000 values of θ per experiment.

852 discrete state and action spaces used here for RL may partly limit the generality of the method, and prevent it from
853 tackling more complex high dimensional problems. Indeed, as designers of the system, we chose a representation
854 (discretization of the output of a SLAM algorithm) adapted to the problem at hand (a navigation problem). However
855 the context of this proposal is to build on the representation redescription framework [Doncieux et al., 2018, 2020] to
856 ultimately design systems that autonomously determine the representations adapted to the task. The modularity of the
857 present architecture also enables to extend it to the continuous case by replacing tabular value functions with neural
858 network implementations. Nevertheless, there is actually a trade-off between quickly learning an efficient (even if not
859 optimal) solution to coarsely represented or even discretized problem, versus slowing acquiring a more precise and
860 optimal solution using continuous representations and deep function approximators. In particular, humans are able to
861 alternate between contexts in which learning a discrete action plan is sufficient, versus contexts requiring the slower
862 acquisition of more fine-grained plans, especially motor plans like riding a bicycle, learning to play a music instrument,
863 etc [Haruno and Kawato, 2006, Hikosaka et al., 1999]. Thus, rather than having robots tackle any new problem with
864 computationally heavy deep RL methods, a promising direction for future work could be to add yet another expert to
865 our architecture, composed of a deep network, that the meta-controller will coordinate and compare to the other experts.
866 This way, when the meta-controller detects that a simpler solution is sufficient, it could avoid heavy computation and
867 would both reduce learning time and energy consumption. Moreover, because in our architecture each expert learns
868 from observing what the other experts are doing, initial MB control could bootstrap initial learning and exploration in
869 the deep network composing the new expert.

870 The arbitration criterion proposed in this work allowed the robots to autonomously determine when to shift between
871 systems during learning, generating coherent temporal decision-making patterns that alternates between strategies over
872 time. This promoted more flexibility than pure MF control in response to task changes, and permitted to reach the
873 same level of performance than pure MB control, while drastically reducing the computational cost. The HRI teaching
874 task revealed an interesting property of our system: Its ability to compensate for the imperfections of the human
875 feedbacks (when they were either omitted or erroneous). This suggests that our method is promising for experiments
876 involving interactions between robot and naive human users. In that case, our architecture can automatically cope with
877 human errors by relying more on its MB component. This enables to avoid redesigning or retuning the robot learning
878 parameters to different situations, and thus make the approach more realistically applicable to real-world HRI.

879 The meta-controller proposed here often produced a sequence of three behavioral phases with different expert selection
880 patterns: Initial MF-driven exploration, MB-driven decisions once the internal model has included reward information,
881 MF-driven less costly decision-making once the MF expert has been sufficiently trained. Such a pattern is similar to
882 the one observed in humans in an instrumental task [Viejo et al., 2015]. In that task, humans had to learn through
883 trial-and-error to associate different colored stimuli (considered as Markovian states) to different fingers of the hand
884 (considered as actions). After learning and stabilizing these associations (exploitation), the task conditions were
885 changed so that the humans had to learn new associations. Different computational models had been fitted to human
886 subjects' behavior, in order to determine the best model: An MF-only model, and MB-only model, and different ways
887 of coordinating MB and MF. Not only did the authors find that an entropy-based MB-MF coordination model best
888 explained humans' behavior in this task. They also found during subsequent analysis of the model fitted to human
889 behavior that it displayed a sequence of three behavioral phases: Initial quick responses by the humans when exploring
890 (where both MF and MB experts contributed), then an increase in decision time due to the MB contribution, and then
891 a progressive reduction of decision time as the MF increased its contribution. It is thus striking that despite a task
892 difference between humans and robots, and despite the fact that the present entropy-based coordination method has
893 been extended from [Viejo et al., 2015] by adding a cost term, we can still replicate on the robot a similar behavioral
894 pattern than the one experimentally observed in humans.

895 A system able to detect context changes was added in the last experiment, in order to allow for re-learning when
896 the goal-change occurred. It was inspired by such a system developed in our previous MF-MB coordination system
897 [Caluwaerts et al., 2012a]. Explicitly detecting task changes did not prove necessary in the navigation nor in the
898 teaching task, nevertheless, it should also improve the performance in these two tasks. In future work, we could study
899 to which extent the context change detector produces similar performance in these other tasks, and whether it allows in
900 general to cope with a wider variety of non-stationary tasks.

901 Acknowledgements

902 This work was supported by the Délégation Générale de l'Armement (ER, RD), by the Agence Nationale de la
903 Recherche (ANR-12-CORD-0030 Roboergosum Project), by joint funding from ANR and the Austrian Science Fund
904 FWF (ANR-21-CE33-0019-01), by the Centre National de la Recherche Scientifique (INS2I Appel Unique programme;
905 MK), and by the European Union Horizon 2020 research and innovation programme under grant agreement No 761758

906 “HumanE-AI-Net” (H2020-ICT-48 Network of Centers of Excellence). The authors would like to thank Romain
907 Retureau and Camille Lakhli for their help with some of the figures.

908 **Data and code availability**

909 The code related to this work will be made available in an open source repository like github upon publication. The
910 datasets generated during and/or analysed during the current study are available from the corresponding author on
911 reasonable request.

912 **Competing interest**

913 The authors have no competing interests to declare that are relevant to the content of this article.

914 **References**

- 915 R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *IJRR Journal*, 17:315–337,
916 1998.
- 917 R. Alami, M. Warnier, J. Guitton, S. Lemaignan, and E. A. Sisbot. When the robot considers the human... In *Proceedings*
918 *of the 15th International Symposium on Robotics Research*, 2011.
- 919 Jean-Paul Banquet, Souheil Hanoune, Philippe Gaussier, and Mathias Quoy. From cognitive to habit behavior during
920 navigation, through cortical-basal ganglia loops. In *International Conference on Artificial Neural Networks*, pages
921 238–247. Springer, 2016.
- 922 K. Caluwaerts, A. Favre-Félix, M. Staffa, S. N’Guyen, C. Grand, B. Girard, and M. Khamassi. Neuro-inspired
923 navigation strategies shifting for robots: Integration of a multiple landmark taxon strategy. In T.J. et al. Prescott,
924 editor, *Living Machines 2012, LNAI*, volume 7375/2012, pages 62–73. 2012a.
- 925 K. Caluwaerts, M. Staffa, S. N’Guyen, C. Grand, L. Dollé, A. Favre-Félix, B. Girard, and M. Khamassi. A biologically
926 inspired meta-control navigation system for the psikharpax rat robot. *Bioinspiration & Biomimetics*, 7: 025009,
927 2012b.
- 928 Romain Cazé, Mehdi Khamassi, Lise Aubin, and Benoît Girard. Hippocampal replays under the scrutiny of reinforce-
929 ment learning models. *Journal of neurophysiology*, 120(6):2877–2896, 2018.
- 930 Raja Chatila, Erwan Renaudo, Mihai Andries, Ricardo Omar Chavez-Garcia, Pierre Luce-Vayrac, Raphaël Gottstein,
931 Rachid Alami, Aurélie Clodic, Sandra Devin, Benoît Girard, and Mehdi Khamassi. Toward self-aware robots.
932 *Frontiers in Robotic and AI*, 5(1):88–108, 2018.
- 933 Yevgen Chebotar, Karol Hausman, Marvin Zhang, Gaurav Sukhatme, Stefan Schaal, and Sergey Levine. Combining
934 model-based and model-free updates for trajectory-centric reinforcement learning. In *International conference on*
935 *machine learning*, pages 703–711. PMLR, 2017.
- 936 Nathaniel D Daw, Samuel J Gershman, Ben Seymour, Peter Dayan, and Raymond J Dolan. Model-based influences on
937 humans’ choices and striatal prediction errors. *Neuron*, 69(6):1204–1215, 2011.
- 938 N.D. Daw, Y. Niv, and P. Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for
939 behavioral control. *Nat. Neurosci.*, 8(12):1704–1711, 2005.
- 940 L. Dollé, D. Sheynikhovich, B. Girard, R. Chavarriaga, and A. Guillot. Path planning versus cue responding: a
941 bioinspired model of switching between navigation strategies. *Biological Cybernetics*, 103(4):299–317, 2010.
- 942 Laurent Dollé, Mehdi Khamassi, Benoît Girard, Agnes Guillot, and Ricardo Chavarriaga. Analyzing interactions
943 between navigation strategies using a computational model of action selection. In *International Conference on*
944 *Spatial Cognition*, pages 71–86, 2008.
- 945 Stephane Doncieux, David Filliat, Natalia Díaz-Rodríguez, Timothy Hospedales, Richard Duro, Alexandre Coninx,
946 Diederik M Roijers, Benoît Girard, Nicolas Perrin, and Olivier Sigaud. Open-ended learning: a conceptual framework
947 based on representational redescription. *Frontiers in neurorobotics*, page 59, 2018.
- 948 Stephane Doncieux, Nicolas Bredeche, Léni Le Goff, Benoît Girard, Alexandre Coninx, Olivier Sigaud, Mehdi
949 Khamassi, Natalia Díaz-Rodríguez, David Filliat, Timothy Hospedales, et al. Dream architecture: a developmental
950 approach to open-ended learning in robotics. arXiv preprint arXiv:2005.06223, 2020.

- 951 Rémi Dromnelle, Benoît Girard, Erwan Renaudo, Raja Chatila, and Mehdi Khamassi. Coping with the variability in
952 humans reward during simulated human-robot interactions through the coordination of multiple learning strategies.
953 In 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pages
954 612–617. IEEE, 2020a.
- 955 Rémi Dromnelle, Erwan Renaudo, Guillaume Pourcel, Raja Chatila, Benoît Girard, and Mehdi Khamassi. How to
956 reduce computation time while sparing performance during robot navigation? a neuro-inspired architecture for
957 autonomous shifting between model-based and model-free learning. In Conference on Biomimetic and Biohybrid
958 Systems, pages 68–79. Springer, 2020b.
- 959 OJ Dunn. Multiple comparisons using rank sums technometrics 6: 241–252. Find this article online, 1964.
- 960 Sašo Džeroski, Luc De Raedt, and Kurt Driessens. Relational reinforcement learning. Machine learning, 43(1-2):7–52,
961 2001.
- 962 David Feil-Seifer, Kerstin S Haring, Silvia Rossi, Alan R Wagner, and Tom Williams. Where to next? the impact of
963 covid-19 on human-robot interaction research, 2020.
- 964 E. Gat. On three-layer architectures. In Artificial Intelligence and Mobile Robots. MIT Press, 1998.
- 965 B. Girard, D. Filliat, J-A. Meyer, A. Berthoz, and A. Guillot. Integration of navigation and action selection functionalities
966 in a computational model of cortico-basal ganglia-thalamo-cortical loops. Adaptive Behavior, 13:2, 2005.
- 967 Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. Policy shaping:
968 Integrating human feedback with reinforcement learning. Advances in neural information processing systems, 26,
969 2013.
- 970 G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters.
971 Trans. Rob., 23(1):34–46, February 2007. ISSN 1552-3098. doi: 10.1109/TRO.2006.889486.
- 972 Muhammad Burhan Hafez, Cornelius Weber, Matthias Kerzel, and Stefan Wernter. Curious meta-controller: Adaptive
973 alternation between model-based and model-free control in deep reinforcement learning. In 2019 International Joint
974 Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2019.
- 975 Simon Hangl, Vedran Dunjko, Hans J Briegel, and Justus Piater. Skill learning by autonomous robotic playing using
976 active learning and exploratory behavior composition. Frontiers in Robotics and AI, 7:42, 2020.
- 977 Masahiko Haruno and Mitsuo Kawato. Heterarchical reinforcement-learning model for integration of multiple cortico-
978 striatal loops: fmri examination in stimulus-action-reward association learning. Neural networks, 19(8):1242–1254,
979 2006.
- 980 Okihide Hikosaka, Hiroyuki Nakahara, Miya K Rand, Katsuyuki Sakai, Xiaofeng Lu, Kae Nakamura, Shigehiro
981 Miyachi, and Kenji Doya. Parallel neural networks for learning sequential procedures. Trends in neurosciences, 22
982 (10):464–471, 1999.
- 983 Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot
984 with deep reinforcement learning: lessons we have learned. The International Journal of Robotics Research, 40(4-5):
985 698–721, 2021.
- 986 Adrien Jauffret, Nicolas Cuperlier, Philippe Gaussier, and Philippe Tarroux. From self-assessment to frustration, a
987 small step toward autonomy in robotic navigation. Frontiers in neurorobotics, 7:16, 2013.
- 988 Kshitij Judah, Saikat Roy, Alan Fern, and Thomas Dietterich. Reinforcement learning via practice and critique advice.
989 In Proceedings of the AAAI Conference on Artificial Intelligence, volume 24, pages 481–486, 2010.
- 990 Daniel Justus, John Brennan, Stephen Bonner, and Andrew Stephen McGough. Predicting the computational cost of
991 deep learning models. In 2018 IEEE international conference on big data (Big Data), pages 3873–3882. IEEE, 2018.
- 992 M. Keramati, A. Dezfouli, and P. Piray. Speed/accuracy trade-off between the habitual and goal-directed processes.
993 PLoS Comp. Biol., 7(5):1–25, 2011.
- 994 M. Khamassi and M.D. Humphries. Integrating cortico-limbic-basal ganglia architectures for learning model-based and
995 model-free navigation strategies. Frontiers in Behavioral Neuroscience, 6:79, 2012.
- 996 Mehdi Khamassi, Charles Wilson, R Rothé, René Quilodran, Peter F Dominey, and Emmanuel Procyk. Meta-learning,
997 cognitive control, and physiological interactions between medial and lateral prefrontal cortex. Neural basis of
998 motivational and cognitive control, pages 351–370, 2011.
- 999 Mehdi Khamassi, George Velentzas, Theodore Tsitsimis, and Costas Tzafestas. Robot fast adaptation to changes in hu-
1000 man engagement during simulated dynamic social interaction with active exploration in parameterized reinforcement
1001 learning. IEEE Transactions on Cognitive and Developmental Systems, 10(4):881–893, 2018.

- 1002 W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In
 1003 Proceedings of the fifth international conference on Knowledge capture, pages 9–16, 2009.
- 1004 W Bradley Knox and Peter Stone. Reinforcement learning from simultaneous human and mdp reward. In AAMAS,
 1005 pages 475–482, 2012.
- 1006 W Bradley Knox, Matthew E Taylor, and Peter Stone. Understanding human teaching modalities in reinforcement
 1007 learning environments: A preliminary report. In IJCAI 2011 Workshop on Agents Learning Interactively from
 1008 Human Teachers (ALIHT), 2011.
- 1009 Jan Kober, Andrew J. Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. IJRR Journal, 32(11):
 1010 1238–1274, 2013. doi: 10.1177/0278364913495721.
- 1011 Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. The transferability approach: Crossing the reality gap in
 1012 evolutionary robotics. IEEE Transactions on Evolutionary Computation, 17(1):122–145, 2012.
- 1013 Sang Wan Lee, Shinsuke Shimojo, and John P O’Doherty. Neural computations underlying arbitration between
 1014 model-based and model-free learning. Neuron, 81(3):687–699, 2014.
- 1015 Martin Llofriu, Gonzalo Tejera, M Contreras, Tatiana Pelc, Jean-Marc Fellous, and Alfredo Weitzenfeld. Goal-oriented
 1016 robot navigation learning using a multi-scale space representation. Neural Networks, 72:62–74, 2015.
- 1017 Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan online, learn offline:
 1018 Efficient learning and exploration via model-based control. In International Conference on Learning Representations,
 1019 2019.
- 1020 Giovanni Maffei, Diogo Santos-Pata, Encarni Marcos, Marti Sánchez-Fibla, and Paul FMJ Verschure. An embodied
 1021 biologically constrained model of foraging: from classical and operant conditioning to adaptive real-world behavior
 1022 in dac-x. Neural Networks, 72:88–108, 2015.
- 1023 J.-A. Meyer and A. Guillot. Biologically-inspired robots. In B. Siciliano and O. Khatib, editors, Handbook of robotics,
 1024 pages 1395–1422. Springer-Verlag, Berlin, 2008.
- 1025 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves,
 1026 Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis
 1027 Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control
 1028 through deep reinforcement learning. Nature, 518(7540):529–533, February 2015. ISSN 00280836.
- 1029 John P O’Doherty, Jeffrey Cockburn, and Wolfgang M Pauli. Learning, reward, and decision making. Annual review
 1030 of psychology, 68:73–100, 2017.
- 1031 Giovanni Pezzulo, Francesco Rigoli, and Fabian Chersi. The mixed instrumental controller: Using value of information
 1032 to combine habitual choice and mental simulation. Frontiers in Psychology, 4(92), 2013. ISSN 1664-1078. doi:
 1033 10.3389/fpsyg.2013.00092. URL [http://www.frontiersin.org/cognition/10.3389/fpsyg.2013.00092/](http://www.frontiersin.org/cognition/10.3389/fpsyg.2013.00092/abstract)
 1034 [abstract](http://www.frontiersin.org/cognition/10.3389/fpsyg.2013.00092/abstract).
- 1035 Taman Powell and Tanya Sammut-Bonnici. Pareto Analysis. 01 2015. ISBN 9781118785317. doi: 10.1002/
 1036 9781118785317.weom120202.
- 1037 M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot
 1038 operating system. In ICRA Workshop on Open Source Software, 2009.
- 1039 Erwan Renaudo, Benoît Girard, Raja Chatila, and Mehdi Khamassi. Design of a control architecture for habit
 1040 learning in robots. In Biomimetic and Biohybrid Systems, LNAI Proceedings, pages 249–260, 2014. doi: 10.1007/
 1041 978-3-319-09435-9_22.
- 1042 Erwan Renaudo, Sandra Devin, Benoît Girard, Raja Chatila, Rachid Alami, Mehdi Khamassi, and Aurélie Clodic.
 1043 Learning to interact with humans using goal-directed and habitual behaviors, 2015a.
- 1044 Erwan Renaudo, Benoît Girard, Raja Chatila, and Mehdi Khamassi. Which criteria for autonomously shifting between
 1045 goal-directed and habitual behaviors in robots? In 5th International Conference on Development and Learning and
 1046 on Epigenetic Robotics (ICDL-EPIROB), pages 254–260, Providence, RI, USA, 2015b.
- 1047 Erwan Renaudo, Benoît Girard, Raja Chatila, and Mehdi Khamassi. Respective advantages and disadvantages of model-
 1048 based and model-free reinforcement learning in a robotics neuro-inspired cognitive architecture. In Biologically
 1049 Inspired Cognitive Architectures BICA 2015, pages 178–184, Lyon, France, 2015c.
- 1050 Dalia Marcela Rojas-Castro, Arnaud Revel, and Michel Menard. Rhizome architecture: An adaptive neurobehavioral
 1051 control architecture for cognitive mobile robots—application in a vision-based indoor robot navigation context.
 1052 International Journal of Social Robotics, 12(3):659–688, 2020.

- 1053 Felix Rutard, Olivier Sigaud, and Mohamed Chetouani. Tirl: enriching actor-critic rl with non-expert human teachers
1054 and a trust model. In 2020 29th IEEE International Conference on Robot and Human Interactive Communication
1055 (RO-MAN), pages 604–611. IEEE, 2020.
- 1056 Farzaneh Sheikhezahad Fard and Thomas P Trappenberg. A novel model for arbitration between planning and habitual
1057 control systems. Frontiers in neurorobotics, (13):52, 2019. doi: 10.3389/fnbot.2019.00052.
- 1058 Amitai Shenhav, Matthew M Botvinick, and Jonathan D Cohen. The expected value of control: an integrative theory of
1059 anterior cingulate cortex function. Neuron, 79(2):217–240, 2013.
- 1060 Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp.
1061 arXiv preprint arXiv:1906.02243, 2019.
- 1062 Richard S. Sutton and Andrew G. Barto. Introduction to Reinforcement Learning. MIT Press, Cambridge, MA, USA,
1063 1st edition, 1998. ISBN 0262193981.
- 1064 Matthijs Van Der Meer, Zeb Kurth-Nelson, and A David Redish. Information processing in decision-making systems.
1065 The Neuroscientist, 18(4):342–359, 2012.
- 1066 Guillaume Viejo, Mehdi Khamassi, Andrea Brovelli, and Benoît Girard. Modelling choice and reaction time during
1067 arbitrary visuomotor learning through the coordination of adaptive working memory and reinforcement learning.
1068 Frontiers in Behavioral Neuroscience, 9(225), 2015. ISSN 1662-5153. doi: 10.3389/fnbeh.2015.00225.
- 1069 Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis,
1070 and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. Nature neuroscience, 21(6):
1071 860–868, 2018.
- 1072 Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong
1073 Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. arXiv preprint
1074 arXiv:1907.02057, 2019.
- 1075 Martina Zambelli and Yiannis Demiris. Online multimodal ensemble learning using self-learned sensorimotor represen-
1076 tations. IEEE Transactions on Cognitive and Developmental Systems, 9(2):113–126, 2016.
- 1077 Alexandre Zenon, Oleg Solopchuk, and Giovanni Pezzulo. An information-theoretic perspective on the costs of
1078 cognition. Neuropsychologia, 123:5–18, 2019.

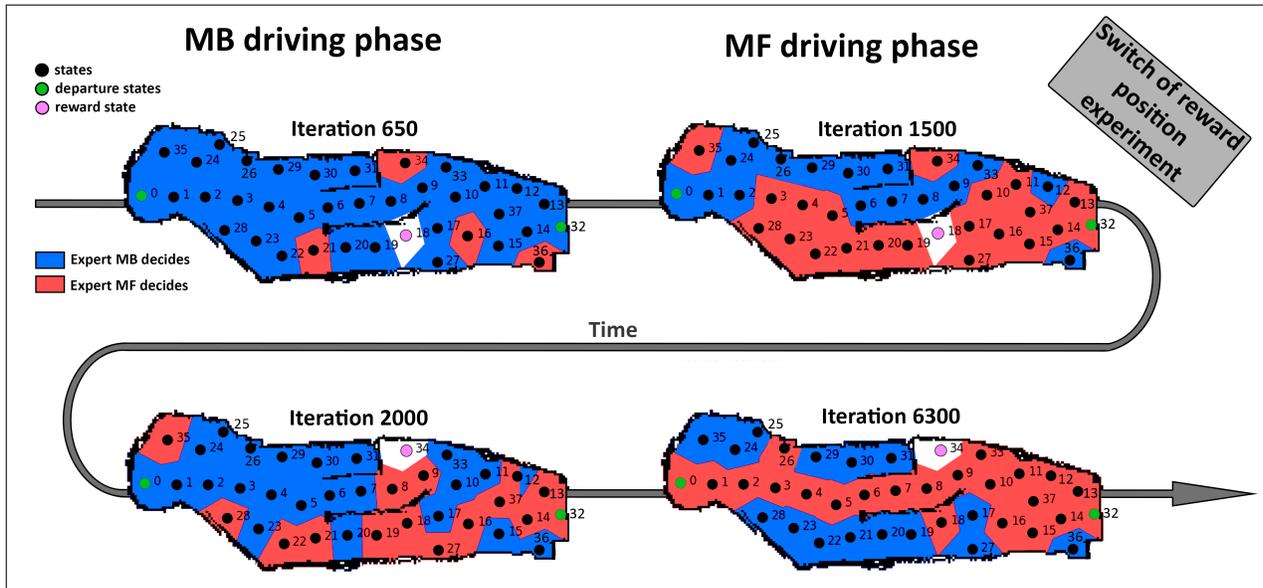


Figure 2: Evolution of the expert spatial preferences in the reward location change navigation experiment. Expert selection maps of the MC-EC agent for one of the hundred simulations: in red, states where the MF was the last chosen expert, in blue, where the MB was last chosen. The MF driving phase and the MB driving phase correspond to the behavioral phases identified in Fig. 5C in the main manuscript. Same conventions as in Fig. 6 in the main manuscript.

1.2 Additional simulation results of the task with change in wall configuration

1.3 Additional results of the navigation task with the real robot

2 Experiment 2: Human-robot interaction with human as teacher

2.1 Results with human intervention of the type *Congratulations*

2.2 Results with human intervention of the type *Takeover*

3 Experiment 3: Human-robot interaction with human as cooperater

3.1 When the partner becomes an adversary

3.2 Context change detection

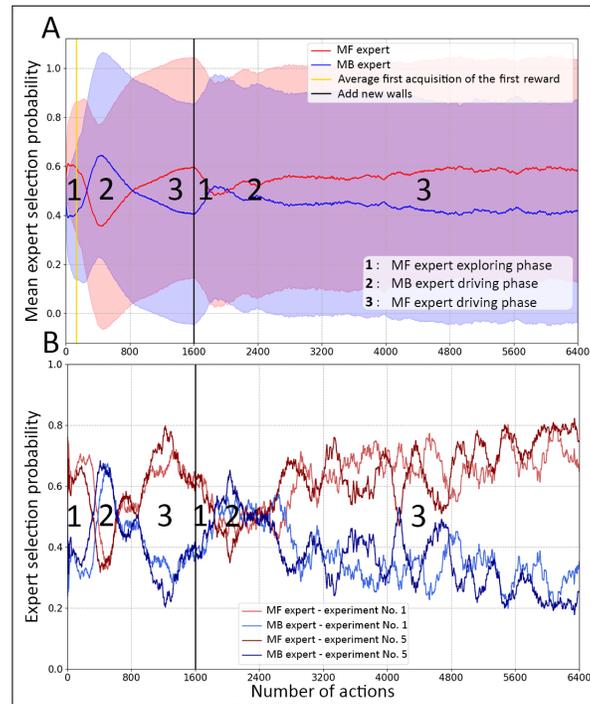


Figure 3: **Simulation results of individual runs of the navigation task with change in wall configuration.** **A.** Mean probabilities of selection of experts by the MC using the Entropy and Cost criterion for 100 simulated runs of the task. **B.** Probabilities of selection of experts by the MC using the Entropy and Cost criterion for 2 simulated runs of the task. Same conventions as Suppl. Fig. 1.

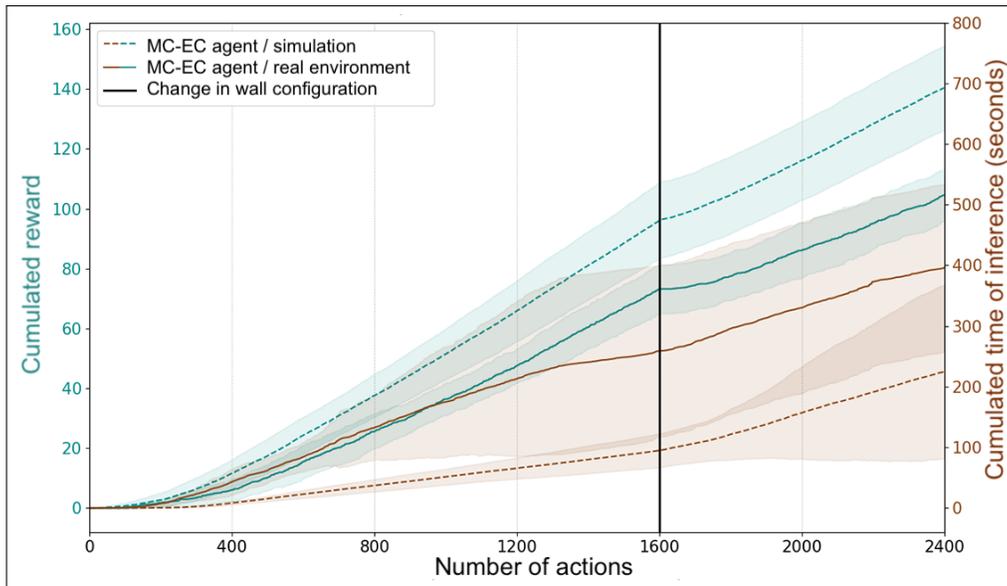


Figure 4: Real robot dynamics of expert selection in the wall configuration condition of the navigation task: Mean performance (in cyan) and computational cost (in brown) of the MC-EC robot. Dashed lines: simulation results; full lines: real robot results.

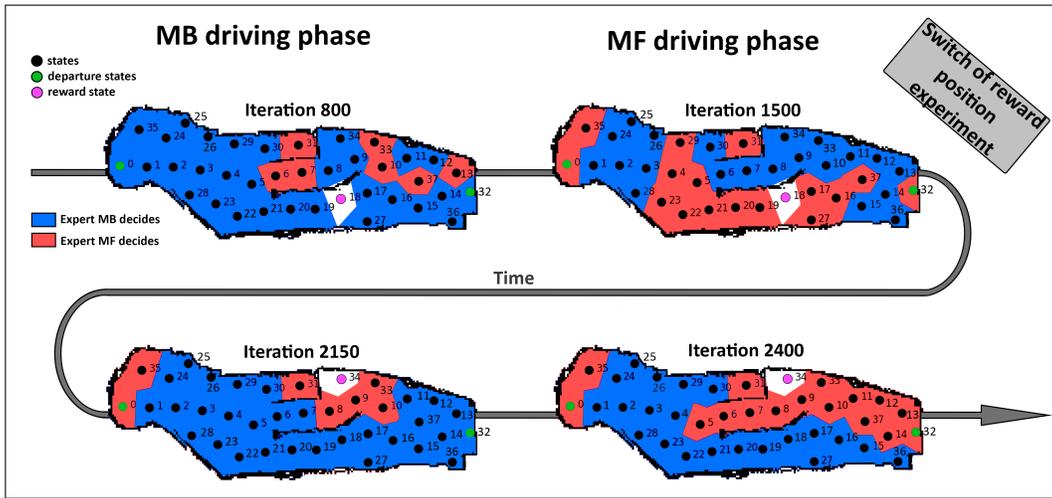


Figure 5: Expert selection map by the MC of the MC-EC robot for one of the navigation experiments with the real robot and with change in reward location. Same conventions as in Fig. 6 in the main manuscript.

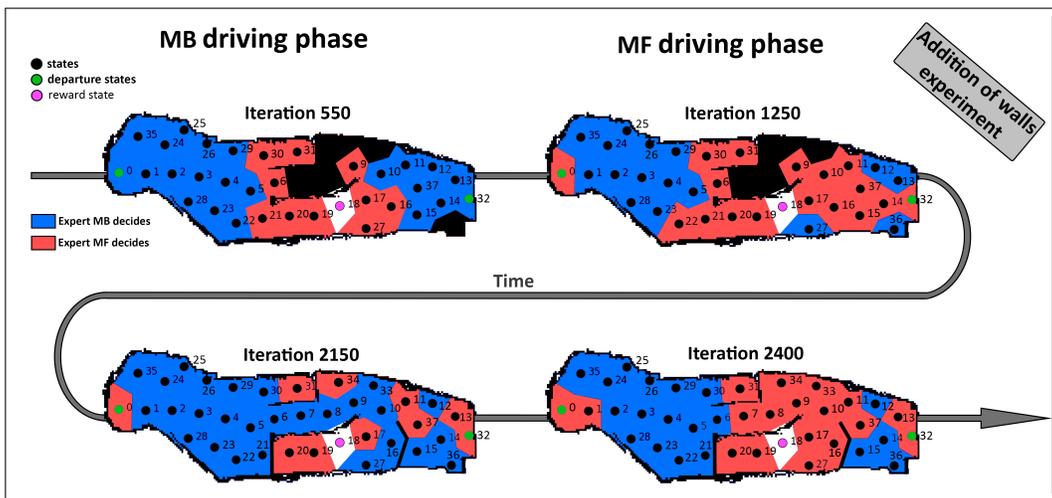


Figure 6: Expert selection map by the MC of the MC-EC robot for one of the navigation experiments with the real robot and with change in wall configuration. Same conventions as in Fig. 6 in the main manuscript.

MF MB	0	10	20	30	40	50	100	150	200	300	400	500
0	0	1.0	1.0	1.0	1.0	1.0	1.0	0.408531	0.030455	0.000016	1.626891e-08	1.572634e-13
10	1.0	10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.008622	3.915252e-05	3.025572e-09
20	1.0	1.0	20	1.0	1.0	1.0	1.0	0.563601	0.045360	0.000029	3.248472e-08	3.714620e-13
30	1.0	1.0	1.0	30	1.0	1.0	1.0	1.0	0.180889	0.000215	3.829072e-07	8.160585e-12
40	1.0	1.0	1.0	1.0	40	1.0	1.0	1.0	1.0	0.155930	1.696102e-03	4.402584e-07
50	1.0	1.0	1.0	1.0	1.0	50	1.0	1.0	1.0	0.203656	2.423178e-03	7.127423e-07
100	1.0	1.0	1.0	1.0	1.0	1.0	100	1.0	1.0	0.074458	6.376559e-04	1.186862e-07
150	1.0	1.0	1.0	1.0	1.0	1.0	1.0	150	1.0	1.0	2.169030e-02	1.448447e-05
200	0.568655	0.033144	0.406211	1.0	1.0	1.0	1.0	1.0	200	1.0	3.101433e-01	6.592969e-04
300	0.132280	0.005327	0.090044	1.0	1.0	0.478832	1.0	1.0	1.0	300	1.0	3.831200e-01
400	0.474306	0.026334	0.336678	1.0	1.0	1.0	1.0	1.0	1.0	1.0	400	1.0
500	0.172429	0.007403	0.118358	1.0	1.0	0.605852	1.0	1.0	1.0	1.0	1.0	500

RND EC	0	10	20	30	40	50	100	150	200	300	400	500
0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.890272	1.0	1.0
10	1.0	10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.455387	1.0	0.632360
20	1.0	1.0	20	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	30	1.0	1.0	1.0	1.0	1.0	0.029717	0.655518	0.044854
40	1.0	1.0	1.0	1.0	40	1.0	1.0	1.0	1.0	0.017303	0.431479	0.026507
50	1.0	1.0	1.0	1.0	1.0	50	1.0	1.0	1.0	0.819644	1.0	1.0
100	1.0	1.0	1.0	1.0	1.0	1.0	100	1.0	1.0	1.0	1.0	1.0
150	1.0	1.0	1.0	1.0	1.0	1.0	1.0	150	1.0	1.0	1.0	1.0
200	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	200	1.0	1.0	1.0
300	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	300	1.0	1.0
400	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	400	1.0
500	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	500

Figure 7: Results of *Dunn's* multiple comparison tests for the performance of the four robots in the *congratulation* type intervention of Experiment 2. P-values below the significance threshold 0.05 are colored in red. The significance level has been corrected with the *Bonferroni correction*.

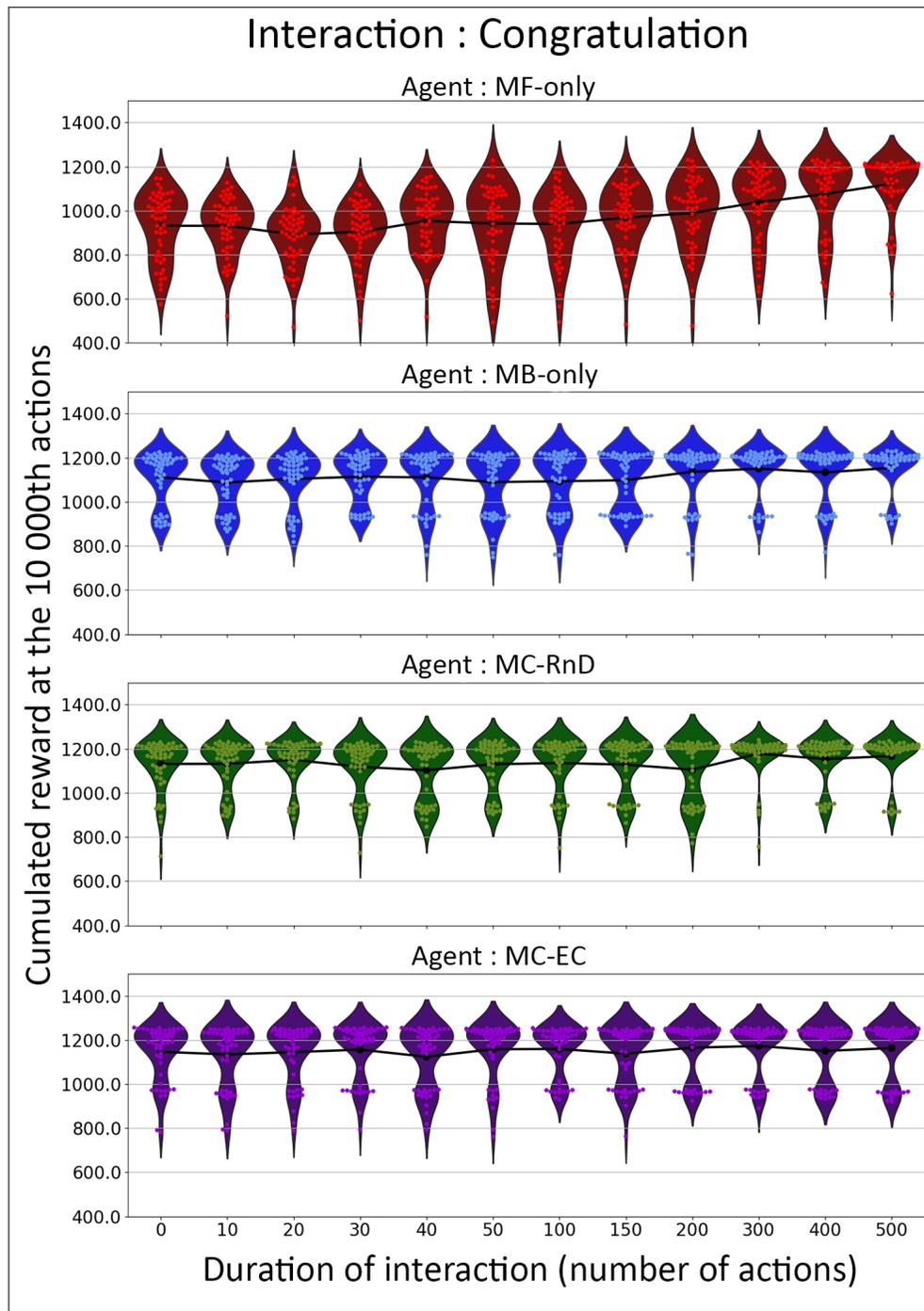


Figure 8: Reward accumulation in Experiment 2 when humans provide *congratulation* feedback for various durations (from 0 to 500 timesteps). Dots report the accumulated after 10,000 simulation timesteps, for 50 simulations. First row (red): MF-only agent; second row (blue) MB-only agent; third row (green): MC-Rnd agent; fourth row (purple): MC-EC agent.

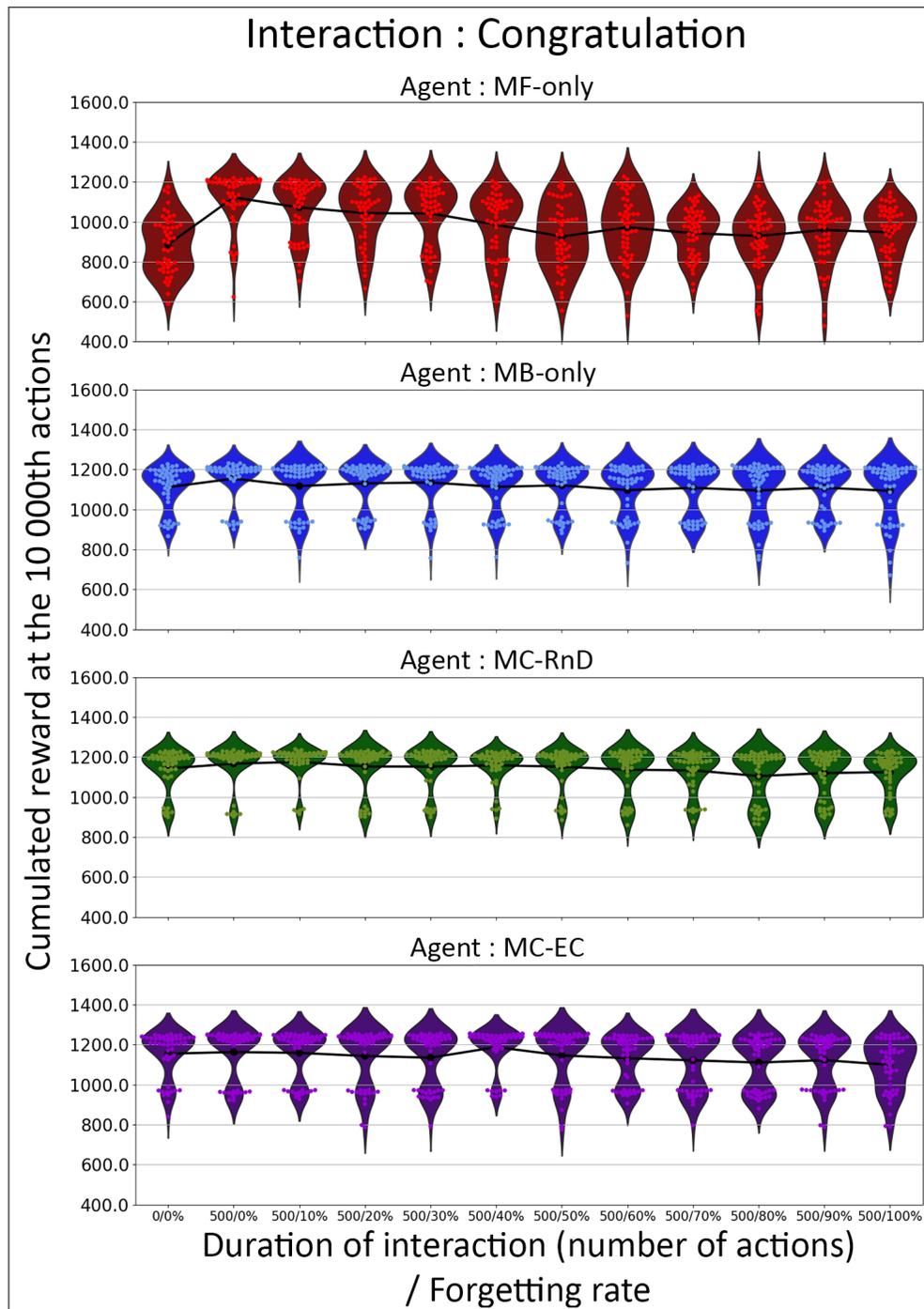


Figure 9: Reward accumulation in Experiment 2 when humans omit to provide *congratulation* feedback with increasing omission rates. Dots report the accumulated after 10,000 simulation timesteps, for 50 simulations. First row (red): MF-only agent; second row (blue) MB-only agent; third row (green): MC-Rnd agent; fourth row (purple): MC-EC agent.

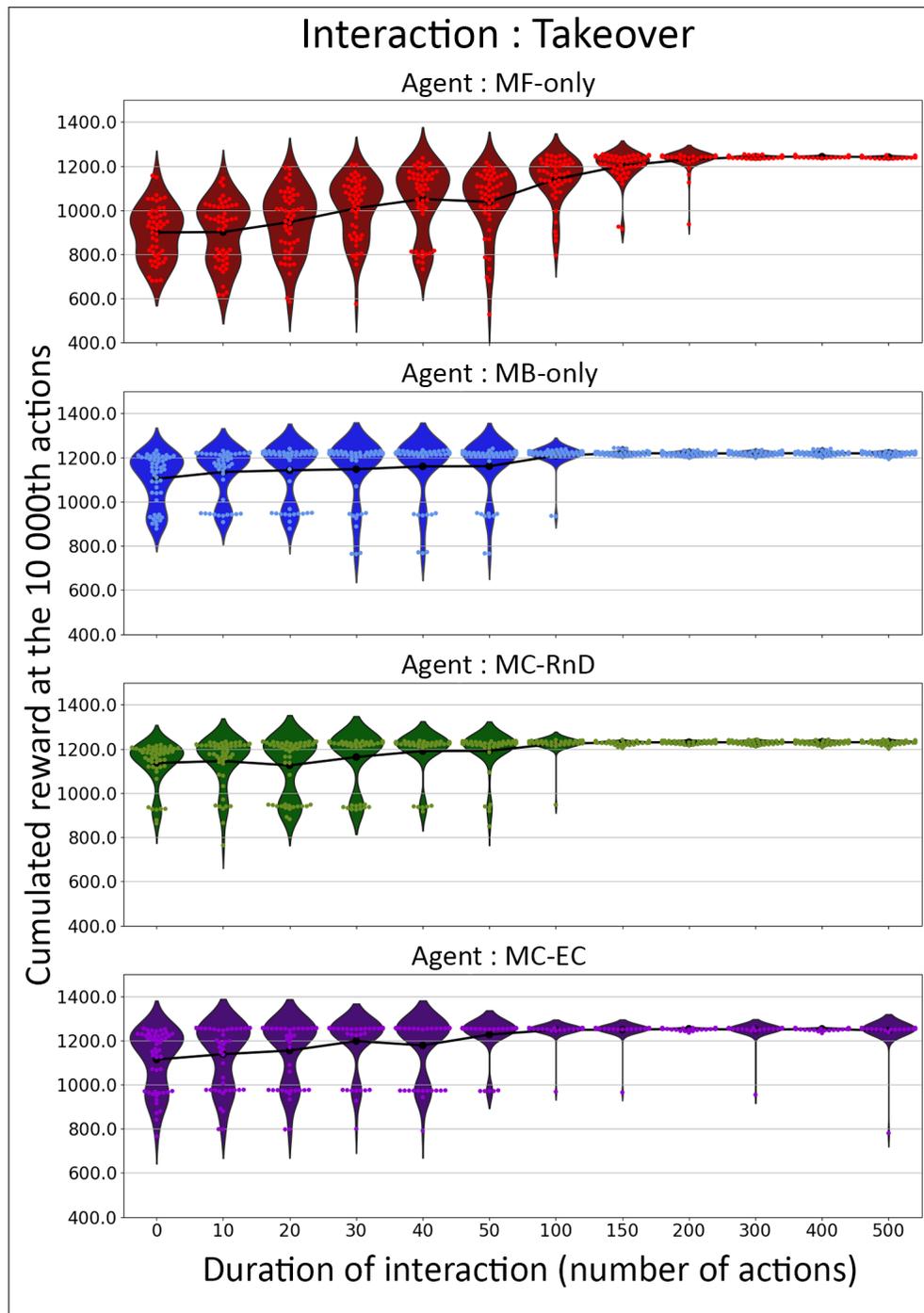


Figure 10: Reward accumulation in the HRI teaching task, when humans provide *takeover* feedback for various durations (from 0 to 500 timesteps). Dots report the accumulated after 10,000 simulation timesteps, for 50 simulations. First row (red): MF-only agent; second row (blue) MB-only agent; third row (green): MC-Rnd agent; fourth row (purple): MC-EC agent.

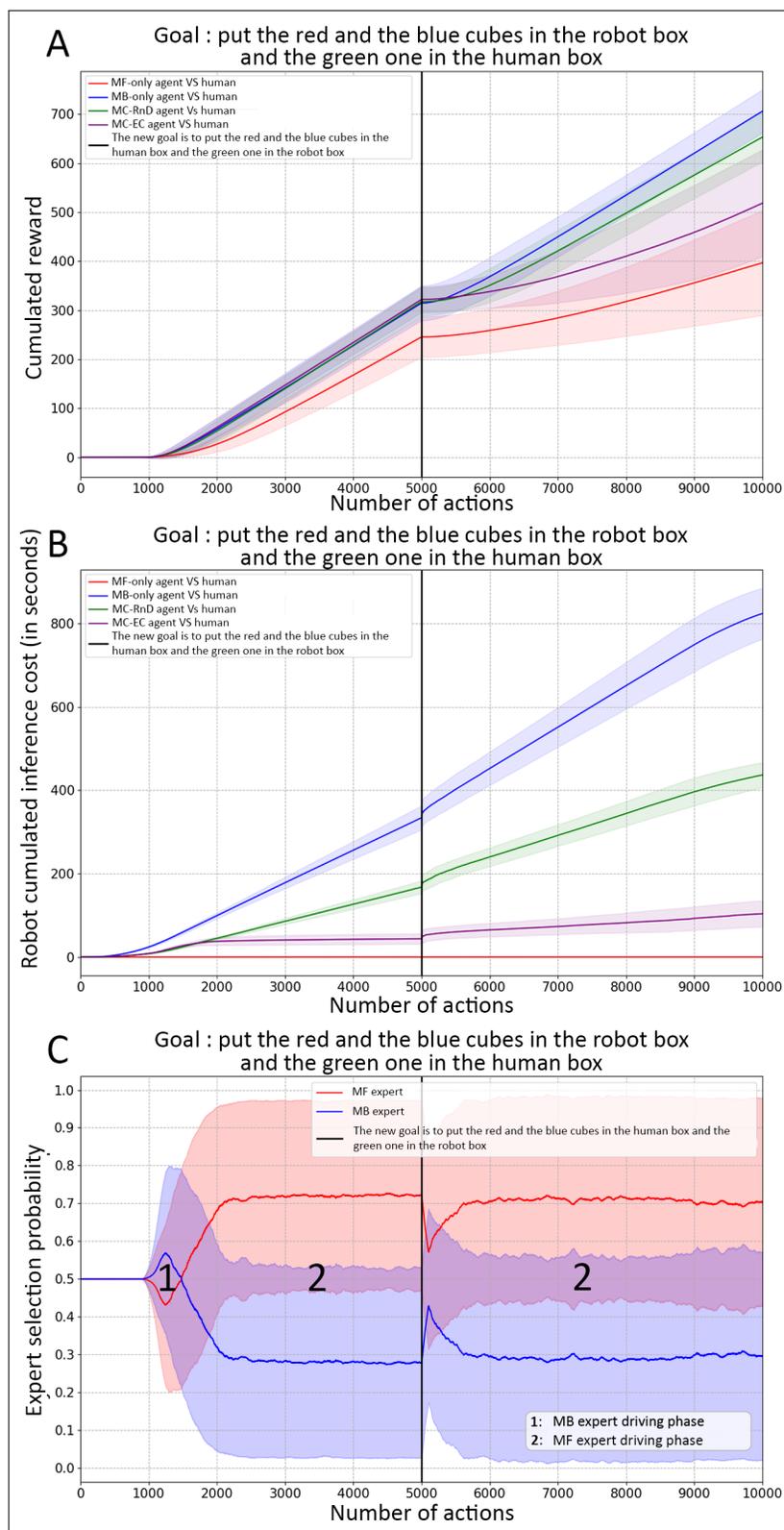


Figure 11: **Simulation results of the human-robot cooperation task (Experiment 3) with the second pair of objectives.** A. Average performance for 50 simulated experiments. B. Average computational cost for 50 simulated experiments. C. Average probability of selection of experts by the meta-controller of the MC-EC robot for 50 simulated experiments. We use standard deviation as an indicator of dispersion in all three figures.

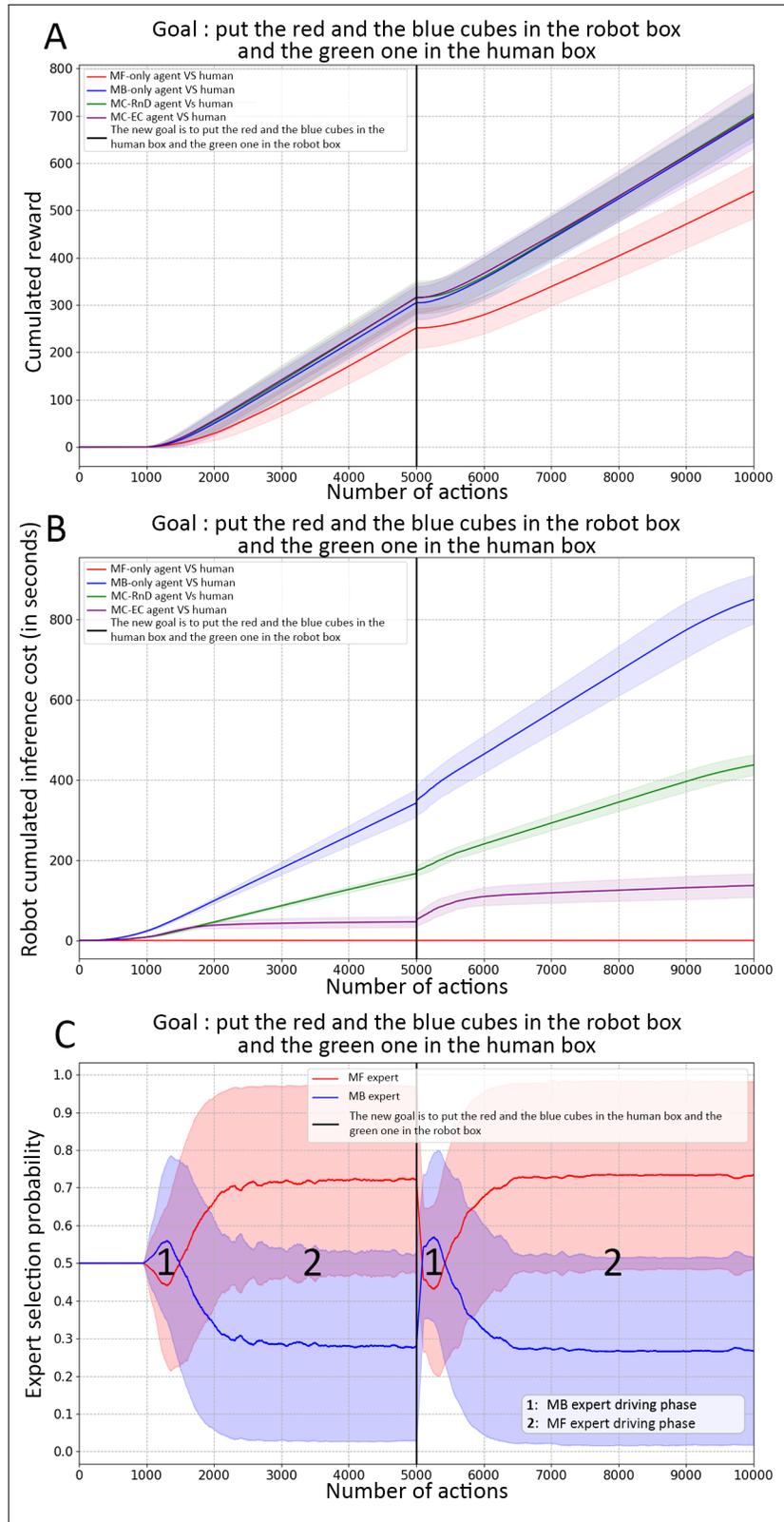


Figure 12: **Simulation results of the human-robot cooperation task (Experiment 3) with the second pair of objectives with context change detection.** **A.** Average performance for 50 simulated experiments. **B.** Average computational cost for 50 simulated experiments. **C.** Average probability of selection of experts by the meta-controller of the MC-EC robot for 50 simulated experiments. We use the standard deviation as an indicator of dispersion in all three figures. In these experiments, robots are able to detect context switches.