

Supporting information for the manuscript entitled “Adaptive reinforcement learning with active state-specific exploration for engagement maximization during simulated child-robot interaction” by Velentzas, Tsitsimis, Rañó, Tzafestas and Khamassi, Paladyn. Journal of Behavioral Robotics, 2018, 9(1), 235–253

1 Replacing the mid-term reward with the myopic value function

From the update rule of the mid-term reward described in the first version of the algorithm [1] it follows

$$\Delta \bar{r}_t = (r_t - \bar{r}_t) / \tau_1 \Rightarrow \bar{r}_{t+1} = \bar{r}_t + \alpha_C (r_t - \bar{r}_t)$$

where $\alpha_C = 1/\tau_1$. Replacing this global approximation of mid-term reward to state-dependent mid-term reward (i.e., after observing quintuple (s, a, θ^a, r, s')) we get

$$\bar{r}(s) \leftarrow \bar{r}(s) + \alpha_C (r - \bar{r}(s))$$

Therefore, using state-specific mid-term rewards $\bar{r}(s)$ is equivalent with using the value function with updates of

$$V(s) \leftarrow V(s) + \alpha_C (r + \gamma V(s') - V(s))$$

with $\gamma = 0$. We used the notation $V_m(s)$ to state the myopic computation. Equivalently, the state-specific long term reward signals $\bar{r}(s)$ are then updated with

$$\bar{r}(s) \leftarrow \bar{r}(s) + \alpha_V (\bar{r}(s) - \bar{r}(s))$$

where $\alpha_V = 1/\tau_2$. Therefore $\bar{r}(s)$ is equivalent with $\bar{V}_m(s)$. The computations of $Q_m(s, a)$ and $\bar{Q}_m(s, a)$ are done equivalently.

2 Comparison with the previous non-state specific version of the active exploration algorithm

Here we compare our algorithm with the previous version presented in [1, 2] on a parameterized action Markov Decision Process of 5 states plus a final accepting state (Experiment 2, left MDP of Figure 3 in the paper).

The action space is $A = A_d \times A_p$, where $A_p = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ and $A_p = [-100, 100]$; simply there are 6 discrete actions, each one with a continuous parameter. We assume that for every state S_i at each timestep t , there is an optimal action $a_{S_i}^*$ which leads to the next state in a deterministic manner (i.e., $P(S_{i+1}|S_i, a_{S_i}^*) = 1$) independently of its parameter $\theta_{S_i}^{a_{S_i}^*}$, while all other actions $a \in A_d \setminus \{a_{S_i}^*\}$ result in no state change. However, each optimal discrete action $a_{S_i}^*$ is characterized by an optimal value $\mu_{S_i}^*$ as shown in Figure 2 of the main paper, and choosing a parameter $\theta_{S_i}^{a_{S_i}^*}$ such that $H(\theta_{S_i}^{a_{S_i}^*}) \geq 0$ will result to an increase of the reward signal r_t . This occurs when

$$\mu_{S_i}^* - \sigma^* \sqrt{2 \ln 2} \leq \theta_{S_i}^{a_{S_i}^*} \leq \mu_{S_i}^* + \sigma^* \sqrt{2 \ln 2}$$

where σ^* will be a global parameter for all states and actions, even though it could also be state-action dependent as a measure of specificity. This inequality specifies a tolerance interval with a fixed width for the parameter value of each optimal action, however the range may change since both the optimal action and the optimal parameter value are non-stationary in the general case.

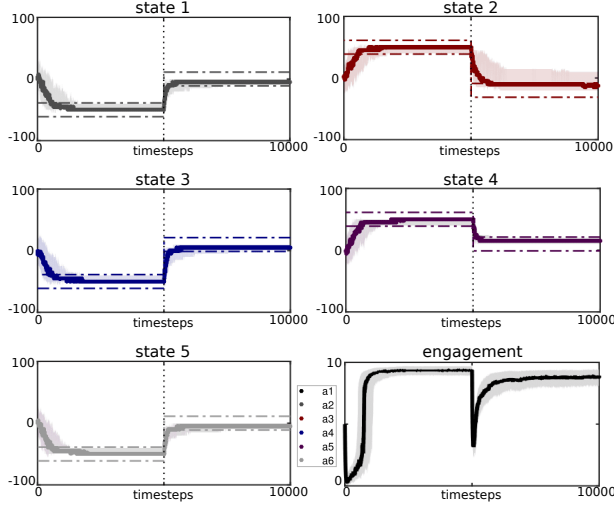


Figure 1: Experiment 2 Task 0. The median values of the chosen parameters for each action of interest at each state is shown, along with their interquartile range as shaded area. The color of the horizontal dash-dot lines represent the optimal actions while the range between them represents the tolerance interval. The engagement on each timestep is shown at the bottom right.

2.1 Global change-points

We first perform numerical simulations and measure the performance on two different tasks using the right MDP of Figure 3 in the paper for 10000 timesteps (we call this a hyper-session), where the structure of the MDP or the environmental reward feedback changes for all states at the same moment (global change-point). We first simulate *Task 0*, a simplified version of *Task 1* in the main paper where the optimal discrete actions are stationary, meaning that $a_{S_i,t}^*$ are constant in time, whereas the optimal parameter values $\mu_{S_i,t}^*$ abruptly change at timestep $t = 5000$. *Task 0* permits to compare with *Task 1* and analyze what happens in the non-state-specific old version of the active exploration algorithm [1] depending on the amount of change required by the task.

Figure 1 captures the results of *Task 0* after 200 hyper-sessions. The graphs show the median values for the actions of interest at each state as also the average engagement (bottom right of the figure) along with their interquartile range as shaded areas. Even though the algorithm cannot end up in more than one state on the same timestep, here due to the fact that we present the values over 200 hyper-sessions the lines seem continuous. Note that the sub-optimal actions are not shown here for clarity but we later present metrics on the probability of choosing a correct action. The color of the horizontal dashed-dotted lines represents the optimal action,

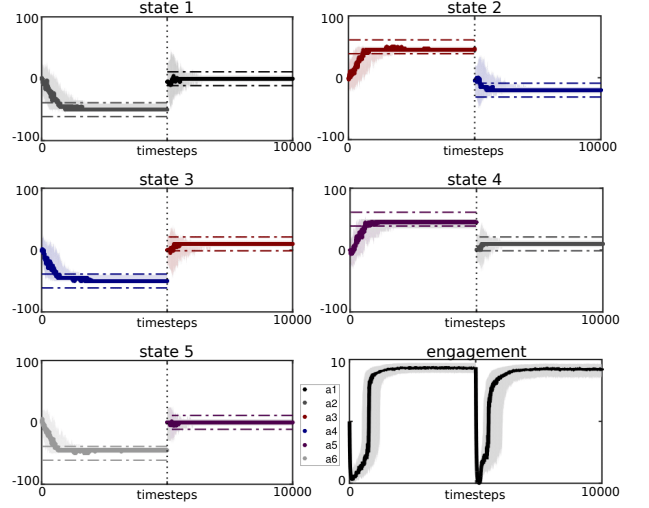


Figure 2: Experiment 2 Task 1. As in Figure 1, the median values of the chosen parameters for each action of interest at each state is shown, along with their interquartile range as shaded area. The color of the horizontal dash-dot lines represent the optimal actions while the range between them represents the tolerance interval. The engagement on each timestep is shown at the bottom right.

while the range between these lines represents the tolerance interval, for which $H(\theta^{a^*}) \geq 0$ (we keep $\sigma^* = 10$ for all cases). The optimal actions for each state are $\{a_{S_1,t}^*, a_{S_2,t}^*, a_{S_3,t}^*, a_{S_4,t}^*, a_{S_5,t}^*\} = \{a_2, a_3, a_4, a_5, a_6\}$ for all timesteps t , while the optimal parameters are $\{\mu_{S_1,t}^*, \mu_{S_2,t}^*, \mu_{S_3,t}^*, \mu_{S_4,t}^*, \mu_{S_5,t}^*\} = \{-50, 50, -50, 50, -50\}$ for $t < 5000$, and $\{\mu_{S_1,t}^*, \mu_{S_2,t}^*, \mu_{S_3,t}^*, \mu_{S_4,t}^*, \mu_{S_5,t}^*\} = \{0, -10, 10, 10, 0\}$ for $t \geq 5000$. In *task 2* the main difference is that the optimal actions also change so that $\{a_{S_1,t}^*, a_{S_2,t}^*, a_{S_3,t}^*, a_{S_4,t}^*, a_{S_5,t}^*\} = \{a_1, a_4, a_3, a_2, a_5\}$ for $t \geq 5000$. The results for *Task 1* are shown in Figure 2 and can be directly compared to those obtained with the new state-specific active exploration algorithm (Figure 4 in the main paper).

In both tasks the algorithm performs exploration on the first timesteps and then manages to approximate the optimal action-parameter tuple $(a_{S_i}^*, \mu_{S_i}^*)$ for each state S_i by the end of timestep 5000, as the greater part of the shaded interquartile regions of uncertainty fall in the tolerance interval. To be more precise, Figure 3 captures the numerically calculated probabilities of choosing the optimal action at each timestep as $P_t(a^*)$, the probability of choosing a parameter value inside the tolerance interval given that the chosen action is the optimal as $P_t(H(\theta^a) \geq 0 | a^*)$, as also their product $P_t(H(\theta^a) \geq 0 \cap a^*)$. At the bottom of the same figure the exploration level is also displayed with $\tau = 1/\beta$ as mean temperature for the softmax-Boltzmann function. All probabilities begin with small val-

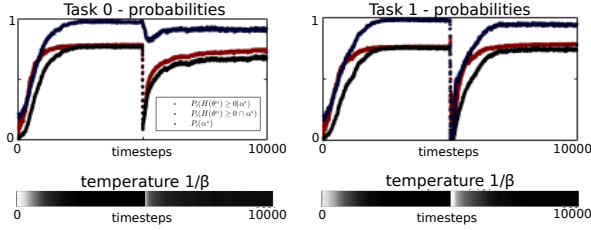


Figure 3: Experiment 2 Tasks 0 and 1. Probabilities of choosing the optimal action at each timestep, as also choosing an optimal parameter inside the tolerance interval for both tasks. At the bottom the temperature $1/\beta$ is shown, with higher values corresponding to white and lower values to black.

ues as the exploration level and the uncertainty of actions are high on the first timesteps. Gradually, the exploration level drops and the actions are learned with $P_t(a^*)$ being close to 1 right before the change-point occurrence, reaching an engagement of 9.2 out of 10 in both tasks. The tolerance region is learned with probability close to 0.78 while a random strategy would give 0.12 for $\sigma^* = 10$.

After the change-point occurrence the algorithm presents adaptivity in both tasks. This can be also seen by observing the probability $P_t(a^*)$ and $P_t(H(\theta^a) \geq 0|a^*)$ in Figure 3. For *Task 0*, $P_t(a^*)$ drops only slightly since the optimal actions are already learned. However, the algorithm does not reach the same levels of performance afterwards. The temperature rises (as β increases) but then drops again as the performance is stabilized. Even though the performance is not completely restored, the engagement drops at first but then rises up to high values of over 8 out of 10. For *Task 1*, the algorithm also relearns the new optimal actions. In fact it manages to achieve the same levels of performance with approximately the same levels of robustness. Observing the temperature at the bottom right of Figure 3 we can see that the temperature rises more than in *Task 0*. However, the performance is completely restored afterwards. A main reason for this feature is the larger stimulus (or “novelty”) resulting from the sudden drop of rewards r_t , which then results in large negative values of $\bar{r}_t - \bar{r}_t$ and larger drops in β as a consequence. This large stimulus therefore restores exploration to higher levels (also observed by the lighter area of the temperature in comparison with the temperature in *Task 1*) and the algorithm does not suffer from “inertia” by an already learned strategy. Nevertheless, the maximal engagement reached and the time needed to adapt after the change-point are not as good as with the new state-specific exploration algorithm (Figure 4 in the main paper).

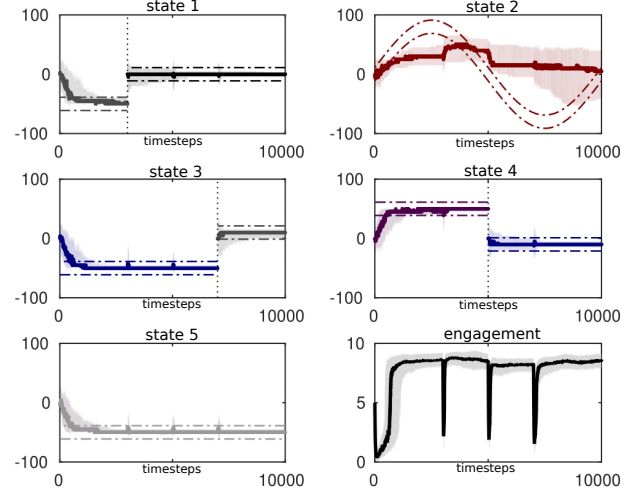


Figure 4: Experiment 2 Task 2. All states follow different dynamics with non-global change-points and continuously changing parameter values. When a change-point occurs in some state, exploration is influenced in all states (shown by the disturbances in all other parameter values chosen). Additionally, the dynamically changing tolerance interval of the optimal action parameter value in state 2 (top right) cannot be followed. These result in an overall lower engagement, shown at the bottom right.

2.2 Local change-points

Next, we test the algorithm on *Task 2*, for which states follow a different dynamics with local change-points as shown in Figure 4. In state S_1 there is a change-point at timestep $t = 3000$. The optimal action-parameter tuple for $t < 3000$ is $\{a_2, -50\}$ and changes to $\{a_1, 0\}$ for $t \geq 3000$. In state S_2 the optimal action is a_3 , however the optimal parameter is changing sinusoidally in time. In state S_3 a change-point occurs at timestep $t = 7000$, where the optimal action-parameter tuple is $\{a_4, -50\}$ for $t < 7000$ and $\{a_2, 0\}$ for $t \geq 7000$. State S_4 is also subject of an abrupt change, where the optimal tuple is $\{a_5, 50\}$ for $t < 5000$ and $\{a_4, -10\}$ for $t \geq 5000$. State S_5 is stationary, with $a_6, -50$ as an optimal action tuple.

From the results shown in Figure 4, we can see that the algorithm manages to adapt to the local change-points in states S_1 , S_3 and S_4 . Nevertheless, the algorithm fails to follow the sinusoidally drifting optimal parameter change in state S_2 . Also note that, because the active exploration is not state-specific, any local change-point in a given state transiently affects performance in all other states. This is particularly obvious in state S_5 where no change-point occurs but where a transient mild re-exploration of the continuous parameter is performed at timesteps $t = 3000$, $t = 5000$ and $t = 7000$. Overall, the algorithm performs quite well, with an engagement above 8 most of the time

and fast adaptations to the local change-points. Nevertheless, the performance is not as good as the one reached by the novel state-specific active exploration algorithm as illustrated by Figure 5 of the main paper.

References

- [1] M. Khamassi, G. Velentzas, T. Tsitsimis, C. Tzafestas, Robot fast adaptation to changes in human engagement during simulated dynamic social interaction with active exploration in parameterized reinforcement learning, *IEEE Transactions on Cognitive and Developmental Systems*, 2018 (in press)
- [2] M. Khamassi, G. Velentzas, T. Tsitsimis, C. Tzafestas, Active exploration and parameterized reinforcement learning applied to a simulated human-robot interaction task, In: *2017 IEEE Robotic Computing Conference*, Taipei, Taiwan, 2017, 28–35