Supplementary Information

Specification of the Artificial agent algorithm

The artificial agent (AA) selected its target according to the probability for the player to choose a specific color after a given history. It stored the frequency that the participant chose each target after each possible history of four elements composed by two choices and two outcomes (**see Extended table 1**). We call the probability of the player choosing the black card P_{black}. In competitive trials, the AA will choose the black card with probability $1 - P_{black}$, while in cooperative trials, it will choose the black card with probability P_{black}. A cooperative choice of the AA is defined as an AA choice following the most likely target chosen by the participant. Thus, even in competitive trial blocks, the AA can make a cooperative choices. Since the algorithm needs to be initialized, we arbitrarily defined the first five trials as random (the AA plays the black target with probability 0.5). The possible combinations that are not encountered during these initialization trials were assigned with a probability of choosing the black target of 0.5.

H1: $BWBW$	H2: $BWB\bar{W}$	H3: $BWRW$	H4: $BWR\bar{W}$
H5: $B\bar{W}BW$	H6: $B\bar{W}B\bar{W}$	H7: $B\bar{W}RW$	H8: $B\bar{W}R\bar{W}$
H9: $RWBW$	H10: $RWB\bar{W}$	H11: $RWRW$	H12: $RWR\bar{W}$
H13: $R\bar{W}BW$	H14: $R\bar{W}B\bar{W}$	H15: $R\bar{W}RW$	H16: $R\bar{W}R\bar{W}$

$\mathrm{W} = \mathbf{Win}$ $\quad \mathrm{B} = \mathbf{Black\ target\ chosen}$
$\bar{\mathrm{W}} = \mathbf{Lose}$ $\quad \mathrm{R} = \mathbf{Red\ target\ chosen}$

Table S1. Exhaustive (16) possible histories Hi of outcomes (Win/Lose) and choices (Black chosen/red chosen) used by the algorithm to track the probability that the participant plays black.

Mixed-Intention Influence model: complementary analyses

To determine whether the Mixed-Intention Influence model could recover the observed behavioral Cooperativity signature (interaction between the participant's outcome and the following choice by the AA to switch or not), we simulated one hundred behavioral sets using the influence model from each of the 3 versions (i.e. the competitive expert alone, the cooperative expert alone and the mixed intentions version arbitrating between the cooperative and competitive experts). Only the Mixed-Intention Influence model was able to reproduce a similar effect of the Cooperativity signature on the probability to stay on the same target. This confirms the validity and specificity of the winning model (for

behavior generated by the cooperative expert alone or competitive expert alone, see **Extended Fig. 1 a,b**).

While our results suggest that the Mixed-Intention Influence model best predicts and reproduces behavior of participants. It is possible that the tracking was not dynamic across trials but fixed throughout the entire experiment (for example with the same ratio of competitive/cooperative numbers of trial). To test this hypothesis, of a fixed arbitrator deciding between the two experts, we also tested a model using a static free parameter arbitrating between the two experts. The results of the Bayesian model selection still assigned the Mixed-Intention Influence model to most participants, even after addition of this free parameter, which demonstrate the importance of the dynamic aspect of the tracking. Finally, because only the second order term of the influence model differentiates the two modes of interactions, we tested the contribution of the mentalizing term to the fit by removing this parameter. This led to a decrease of 5.39% (95% confident interval [2.47;-8.31]) in the log-likelihood, indicating the importance of the mentalizing term.

An important parameter of the model is the competitive bias (δ), representing the participant's prior on the other's intentions. We observed a significant bias (δ) towards the competitive framework (mean=0.2247, p= 0.0181). This competitive bias, of the Mixed-Intention Influence model has an observable behavioral effect: a higher competitive bias increases the randomness of switching targets between two consecutive trials, therefore making participants behavior more unpredictable ($\frac{d(switch)}{d(\delta)} = 0.13$, CI=[0.06; 0.20], p<0.0005, χ^2 test when δ is included in the logistic regression on the probability to stay).

Neural correlates of higher order feature of the interaction

We searched for brain regions computing the expected value for staying on the same target for the Mixed-Intention Influence model and for classic reinforcement learning to compare them for higher order inferences. To do this, we ran two GLMs (GLM5 and GLM6) containing the expected reward for staying on the same target as the only parametric regressor at the time of choice. In GLM5, the expected reward for staying was computed with the Mixed-Intention Influence model whereas in GLM6 it was computed with a reinforcement learning model. We found that the ventral striatum coded the expected reward for staying on the same target positively, as computed by the Mixed-Intention Influence model (x,y,z = 14, 11,-2; p < 0.05 whole-brain FWE corrected at the cluster level, threshold at p<0.001, see **Extended Fig. 3a**). Comparison of the neural correlates of the expected reward of the two models with a paired t-test revealed that Ventral Striatum (x,y,z=6,12,0), bilateral dlPFC (x,y,z=-36, 33, 44 and x,y,z= 30,24,42) and MTG (x,y,z=65,-56,-8, p<0.05 FWE corrected threshold at p<0.001) were encoding higher order features of the task (see **Extended Fig. 3b**).

Neural activaty is specific to the attribution of intention, but not the real intention.

It is possible that the observed DLPFC/IPS region activations are only due to the difference in behavior of the artificial agent between the competitive and cooperative blocks. To test this hypothesis, we conducted another GLM (GLM3), separating trials according to the cooperative blocks vs competitive blocks (and not according to classification of trials by the controller). We observed no difference in brain activation, even at a lower threshold ($p>0.01$), indicating that the PE difference observed between the trials classified as competitive by the controller and the trials classified as cooperative were due to the effective tracking of attribution of intentions, and not to PE differences between the two (unsignaled) types of trial blocks. Direct paired t-test comparison between PE for trials classified as competitive > cooperative (i.e. ($\Delta>0$)>($\Delta<0$)) and PE from the comparison between blocks of competitive and cooperative trials showed the engagement of the same regions (right angular gyrus x,y,z=48,-50,32, right dlPFC x,y,z= 30,8,53 and left angular gyrus x,y,z=-33,-59,44 ($p<0.05$ FWE). The left dlPFC was also engaged at a lower threshold (x,y,z=-39,-2,26; $p<0.01$ FWE threshold at $p<0.005$).

The activities we observed could be due to more volatility in the rewarded target, therefore we ran a fourth GLM (GLM4) to control for the volatility of the Artificial Agent. Indeed, the probability that the computer switches target was 11% higher in competitive than in cooperative trials ($p<0.0001$, 95%CI [8.8%; 14%]). However, when adding the trial by trial probability, that the artificial agent switches target as a non-orthogonalized regressor, the right dlPFC (x,y,z=30,9,42) and angular gyrus (x,y,z=51,-50,33) were still more positively correlated with PE in trials classified as competitive by the controller compared to those classified as cooperative. This result indicates that those activations are not due to the volatility of the competitive condition.

Description of computational models

Reinforcement learning model

Reinforcement learning (RL) consisted of directly linking action or state and outcome to predict future rewards after performing a particular action or being in a particular state. In our experiment, we updated action value with the Rescola-Wagner rule:

$$V_{t+1}^{a} = V_{t}^{a} + \alpha * \delta_{t} \quad \text{Eq 1}$$

$$\delta_{t} = R_{t} - V_{t}^{a} \quad \text{Eq 2}$$

Where α is the learning rate. The reward prediction error δ_t is defined as the difference between the reward at trial *t*, R_t and the expected value of the choice *a* at trial *t,* V_t^a. Then the probability to choose action *a* is:

$$p^a = s\left(V^a - V^b\right) \quad \text{Eq 3}$$

With $s(z) = {}^{1}/_{1+e^{-\beta z}}$ the sigmoid function when β is a free parameter to capture the stochasticity of the participant's behavior (i.e. the exploration/exploitation trade-off). We defined the probability to stay as $\begin{cases} p^{red} \; if \; previous \; choice \; was \; red \\ p^{black} \; if \; previous \; choice \; was \; black \end{cases}$. We used the same definition of the probability to stay for other models.

Fictitious play

In game theory, one can infer the probability that the other chooses one particular action and choose one's own action to maximize one's expected reward. This model is called a first order fictitious play model. Thus, the opponent's probability p^* of choosing an action *a* is dynamically updated by tracking the choice history of the opponent:

$$p_{t+1}^* = p_t^* + \eta * \delta_t^p \quad \text{Eq 4}$$

$$\delta_t^p = P_t - p_t^* \quad \text{Eq 5}$$

Where η is the learning rate. The reward prediction error δ_t^p is defined as the difference between the expected action of the opponent at trial *t,* p_t^*, and the actual other's choice on trial *t*, $(P_t = 1)$ if the other's action is *a* and $(P_t = 0)$ if it is *b*. Then the probability to choose action *a* depends on the payoff matrix. In the competitive setting of our game we can derive the probability *q* that the participant chooses action $a = \text{“}Red\ card\text{”}$ using the sigmoid function, the payoff matrix, and the probability that the other chooses action $a = \text{“}Red\ card\text{”}$:

$$p = s(2q^* - 1) \quad \text{Eq 6}$$

p^* is the inferred probability that the other chooses $a = \text{“}Red\ card\text{”}$. Because the payoff matrix is the same for the participant in both competitive and cooperative trials, the mode of interaction has no impact on the decision stage. However, considering the other's decision rule, it would be different in the competitive or cooperative modes:

$$\begin{array}{ll} q = & s(2p^* - 1) \quad \text{in cooperative mode} \\ q = & s(1 - 2p^*) \quad \text{in competitive mode} \end{array} \quad \text{Eq 7}$$

Influence model

Another strategy could be to take into account how one's own actions influence the other's future actions. Thus, to compute the probability of updating of the other's strategy, we replaced update of opponent strategy (Eq. 4) in the player decision rule (Eq. 7). Then with a Taylor expansion taking η close to 0, we added the influence terms (Δp : influence update signal of the participant, Δq : influence update signal of the other):

$$\Delta q \approx +\eta 2\beta q_t(1-q_t)(P_t - p_t^*) \qquad \text{Eq 8}$$

$$\begin{aligned} \Delta p &\approx +\eta 2\beta p_t(1-p_t)(Q_t - Q_t^*) \quad \text{in cooperative} \\ \Delta p &\approx -\eta 2\beta p_t(1-p_t)(Q_t - Q_t^*) \quad \text{in competitive} \end{aligned} \qquad \text{Eq 9}$$

Thus, in the competitive mode, there is only a sign difference between the term of influence of the two players which is not the case in the cooperative setting. A player can thus incorporate the influence of his/her action on the strategy of the other player:

$$\begin{aligned} p_{t+1}^* &= p_t^* + \eta_1(P_t - p_t^*) + \eta_2 2\beta p_t^*(1-p_t^*)(Q_t - q_t^{**}) \quad in\ cooperative \\ p_{t+1}^* &= p_t^* + \eta_1(P_t - p_t^*) - \eta_2 2\beta p_t^*(1-p_t^*)(Q_t - q_t^{**}) \quad in\ competitive \end{aligned} \qquad \text{Eq 10}$$

$$q_{t+1}^* = q_t^* + \eta_1(Q_t - q_t^*) + \eta_2 k_1 2\beta q_t^*(1-q_t^*)(P_t - p_t^{**}) \qquad \text{Eq 11}$$

The Influence model update rules. p_t^* Is the predicted opponent strategy. P_t is the opponent choice and then $(P_t - p_t^*)$ is the action prediction error. The influence update is due to the $(Q_t - q_t^{**})$ term. Thus Q_t is the player's own action and q_t^{**} the inferred probabilities that the opponent has of the player themself (second-order beliefs). Thus, in the cooperative and competitive modes the influence will occur in the opposite directions. In the Mixed-Intention Influence model, we decline p_{t+1}^* in $p_{t+1}^{coop\,*}$ and $p_{t+1}^{comp\,*}$.

To compute the p_t^{**} and q_t^{**}, we invert the decision function (Eq. 7):

$$\begin{aligned} q_t^{**} &= \frac{1}{2} - \frac{1}{2\beta}\ln\left(\frac{1-p^*}{p^*}\right) \\ p_t^{**} &= \frac{1}{2} + \frac{1}{2\beta}\ln\left(\frac{1-p^*}{p^*}\right) \quad \text{in competitive} \\ p_t^{**} &= \frac{1}{2} - \frac{1}{2\beta}\ln\left(\frac{1-q^*}{q^*}\right) \quad \text{in cooperative} \end{aligned} \qquad \text{Eq 12}$$

We define the probability to stay as $\begin{cases} p^{red}\ if\ previous\ choice\ was\ red \\ p^{black}\ if\ previous\ choice\ was\ black \end{cases}$.

k-ToM model

The k-ToM model is defined as in (Devaine et al., 2014a). An economic game under game theory is defined by a utility table $U(a^{self}, a^{other})$ to represent the payoff to players according to the actions of self,(a^{self}) and the other player (a^{other}). In our experiment, this utility table varies between competitive and cooperative blocks (**see Fig 1a**.). Because participants make a binary choice, a^{self} and a^{other} take the value of a=0 for the red option and a=1 for the black option. According to Bayesian

decision theory, agents try to maximize their expected value V = E[$U(a^{self}, a^{other})$]. We assume that agents use a softmax function as a decision rule:

$$P\left(a^{self} = 1\right) = s\left(\frac{V^1 - V^2}{\beta}\right) \quad \text{Eq 13}$$

$P\left(a^{self} = 1\right)$ is the probability that the agent chooses option $a^{self} = 1$. s is the sigmoid function and β is a free parameter called inverse temperature and controls for the magnitude of behavioral noise. The value of each action depends on the probability of other's action with $p^{other} = P(a^{other} = 1)$ and the utility table $U(a^{self}, a^{other})$:

$$V^i = p^{other} * U\left(a^{self} = i, a^{other} = 1\right) + (1 - p^{other}) * U\left(a^{self} = i, a^{other} = 1\right) \quad \text{Eq 14}$$

One key hypothesis of this model is that we consider that the other agent is itself a k-ToM agent. It means that the other agent has the same decision policy as equation 13. Thus, while the agent tracks p^{other}, the other track p^{self} to construct a recursive reasoning. This recursion induces distinct levels of ToM sophistication between the two agents, impacting how agents update their subjective prediction of p^{other}. (Devaine et al., 2014a) k-ToM agents are defined according to the way they update this prediction of p^{other} starting from 0-ToM. Definition of higher level of reasoning is based on the level 0, for which $P\left(a^{other} = 1\right) = s(x_t^0)$, with the log-odd x_t^0 varying with a volatility σ^0. The updating rule for the hidden state x_t^0 follows the Bayes-optimal probabilistic scheme :

$$q(x_{t+1}^0) \propto p(a_{t+1}^0 \mid x_{t+1}^0) \int q(x_t^0) p(x_{t+1}^0 | x_t^0) dx_t^0 \quad \text{Eq 15}$$

With $p(x_{t+1}^0 | x_t^0)$ the 0-ToM's prior belief on the volatility of the log-odd, and $q(x_t^0) \equiv p\left(x_t^0 | a_{1:t}^{other}\right)$, the posterior belief about the log-odds x_t^0 at trial t after the observation of all previous actions a^{other}. Thus, one can derive the 0-ToM's learning rule:

$$\hat{p}_{t+1}^{other} \approx s\left(\mu_t^0 \Big/ \sqrt{1 + {3(\Sigma_t^0 + \sigma^0)}\Big/{\pi^2}}\right) \quad \text{Eq 16}$$

$$\mu_t^0 \approx \mu_{t-1}^0 + \Sigma_t^0 (a_t^{other} - s(\mu_{t-1}^0)) \quad \text{Eq 17}$$

$$\Sigma_t^0 \approx \frac{1}{\frac{1}{\Sigma_{t-1}^0 + \sigma^0} + s(\mu_{t-1}^0)(1 - s(\mu_{t-1}^0))} \quad \text{Eq 18}$$

Where μ_t^0 is the approximate mean of 0-ToM posterior distribution of $q(x^0)$ and Σ_t^0 is it's approximate variance. Thus μ_t^0 is the estimate of the 0-ToM log-odds at trial t and Σ_t^0 her subjective uncertainty about it.

A 1-ToM agent assumes that the other agent reasons with a 0 depth ToM. Thus, with the decision policy of a 0-ToM agent we can construct a 1-ToM agent. More specifically, in combining

equation 13, 14 and 15, 1-ToM agent assumes that the probability for a 0-ToM agent to emit action $a^{other} = 1$ is $p^{other} = s \circ v^1(x_t^1)$ (we use the symbol ∘ to refer to the composition of two functions defined as (g∘f)(x)=g(f(x))) with s the sigmoid function and v^1 the value for 0-ToM agent to choose option 1 :

$$v^1(x_t^1) = \frac{p_t^{self} * \Delta U_t^1 + \left(1 - p_t^{self}\right) * \Delta U_t^0}{\beta_t} \quad \text{Eq 19}$$

With $\Delta U_t^i = U\left(a^{self} = i, a^{other} = 1\right) - U\left(a^{self} = i, a^{other} = 0\right)$ which represents the incentive for the 1-ToM agent to choose option one if 1-ToM agent chooses option $a^{self} = i$. p_t^{self} for a 1-ToM agent is the same as p_t^{other} for 0-ToM agent, thus :

$$p_t^{self} \approx s\left(\mu_{t-1}^0 \Big/ \sqrt{1 + {3(\Sigma_{t-1}^0 + \sigma_t^0)}\Big/{\pi^2}}\right) \quad \text{Eq 20}$$

To let the 1-ToM agent eventually learn how 0-ToM agent learns about herself, and act in consequence, the 1-ToM agent assumes that hidden states x_t^1 vary across trials with volatility σ^1, which leads to a meta learning rule similar to equation 16, 17, 18.

In a more general fashion, an agent of depth $k \geq 2$ considers that the other agent is a lower sophistication κ-ToM agent $(\kappa \leq k)$, but this sophistication has to be learned in addition to the hidden states x^κ that track the opponent's learning and decision making. Thus a k-ToM agent continuously tracks all possible other's sophistication levels and it's associate action probability $p^{other,\kappa} = s \circ v^\kappa(x^\kappa)$ that he will choose $a^{other} = 1$.

$$p_t^{other} = \sum_{l<\kappa} \lambda_t^{k,\kappa} * p_t^{other,\kappa} \quad \text{Eq 21}$$

$$p_t^{other} \approx s \circ \tilde{v}^\kappa\left(\mu_{t-1}^{k,\kappa}, \Sigma_{t-1}^{k,\kappa}\right) \quad \text{Eq 22}$$

$$\lambda_t^{k,\kappa} \approx \left[\frac{\lambda_{t-1}^{k,\kappa} * p_t^{other,\kappa}}{\sum_{\kappa'<\kappa} \lambda_{t-1}^{k,\kappa'} * p_t^{other,\kappa'}}\right]^{a_t^{other}} \left[\frac{\lambda_{t-1}^{k,\kappa} * (1 - p_t^{other,\kappa})}{\sum_{\kappa'<\kappa} \lambda_{t-1}^{k,\kappa'} * (1 - p_t^{other,\kappa'})}\right]^{1 - a_t^{other}} \quad \text{Eq 23}$$

$$\mu_t^{k,\kappa} \approx \mu_{t-1}^{k,\kappa} + \lambda_t^k \Sigma_t^{k,\kappa} W_{t-1}^\kappa \left(a_t^{other} - s \circ v^\kappa(\mu_{t-1}^{k,\kappa})\right) \quad \text{Eq 24}$$

$$\Sigma_l^{k,\kappa} \approx \left[\left(\Sigma_{l-1}^{k,\kappa} + \sigma^k\right)^{-1} + s' \circ v^k\left(\mu_{t-1}^{k,\kappa}\right) \lambda_t^\kappa {W_{t-1}^\kappa}^T W_{t-1}^\kappa\right]^{-1} \quad \text{Eq 25}$$

Where λ_t^k is k-ToM's probability that her opponent is κ-ToM, W^κ is the gradient of v^κ with respect to the hidden states x^κ. Here, v^k is obtained by the recursive injections of equation 5 in equation 1, as we have already done to obtain equation 4. $\tilde{v}^\kappa$ is defined in terms of the expectation operator: $E\left[s \circ v^\kappa\left(\mu_{t-1}^{k,\kappa}, \Sigma_{t-1}^{k,\kappa}\right)\right] = s \circ \tilde{v}^\kappa\left(\mu_{t-1}^{k,\kappa}, \Sigma_{t-1}^{k,\kappa}\right)$. Equation 3 and 5 have been estimated using a Variational approach to approximate Bayesian inference.

Two Experts model

For models using the payoff matrix to update hidden states, making a difference between the cooperative and competitive modes (i.e. k-ToM and the Influence model), we fitted three models in different settings: competitive, cooperative or mixed intentions. For the mixed intention setting, we ran the competitive and cooperative models in parallel, generating a probability that the other will choose option *a* for each possible mode of interactions, P^a_{comp} and P^a_{coop}. We then transformed the probability with the sigmoid function to have values ranging from $-\infty$ to $+\infty$. We have a binomial choice configuration thus $V^a = -V^b$ in both competitive and cooperative settings. Thus, as V_i^a and V_i^b get close to zero, uncertainty for *i*, the other's intention, increases. We defined the reliability of the intention *i* as the absolute value of V_i^a and the probability that the other intention is cooperative as the sigmoid function of the difference in reliability between the two modes:

$$P^t_{coop} = \frac{1}{1+e^{\beta\left(|V^t_{coop}|-|V^t_{comp}|-\delta\right)}} \quad \text{Eq 26}$$

where β is the inverse temperature controlling for the stochasticity of the mode of interaction and δ is the bias towards cooperative mode. Then, with $P^{a,t}_{comp}$ and $P^{a,t}_{coop}$ we computed the decision value given the competitive and cooperative payoff matrix, DV^a_{comp} and DV^a_{coop} respectively and weighted them by the probability of the corresponding mode of interaction to compute the total decision value:

$$DV^t = P^t_{coop} * DV^a_{coop} + \left(1 - P^t_{coop}\right) * DV^a_{comp} \quad \text{Eq 27}$$

The sigmoid function s generated the probability of selecting choice *a* at trial t:

$$p^{a,t} = s(DV^t) \quad \text{Eq 28}$$

Finally, the reward prediction error was defined as the reward at trial t for action a:

$$PE = R^{a,t} - p^{a,t} \quad \text{Eq 29}$$

Active inference model

For this model based on (Friston et al., 2016), we adopted the partially observable Markov decision process (POMDP) framework which is a way of describing transitions among states under the hypothesis that the probability of the next state depends only on the current state. The partially observed aspect of the Markovian process means that states are not directly observable and have to be inferred though a set of (noisy) observations.

Active inferences are composed of a tuple (P, Q, R, S, A, U , Ω) :

- Ω is a finite set of possible observations
- A is a finite set of possible action
- S is a finite set of hidden states
- U is a finite set of control states
- R is the *generative process* over observation $\tilde{o} \in \Omega$, hidden states $\tilde{s} \in S$, and action $\tilde{a} \in A$

$$R(\tilde{o}, \tilde{s}, \tilde{a}) = Pr(\{o_0, \dots, o_t\} = \tilde{o}\,, \{s_0, \dots, s_t\} = \tilde{s}\,, \{a_0, \dots, a_{t-1}\} = \tilde{a})$$

- P is the *generative model* over observation $\tilde{o} \in \Omega$, hidden states $\tilde{s} \in S$, and control states $\tilde{u} \in U$

  $P(\tilde{o}, \tilde{s}, \tilde{u}|m) = Pr(\{o_0, \dots, o_T\} = \tilde{o}\,, \{s_0, \dots, s_T\} = \tilde{s}\,, \{u_0, \dots, u_T\} = \tilde{u})$ with parameters θ.
- Q is the *approximate posterior* over hidden and control states

  $Q(\tilde{s}, \tilde{u}) = Pr(\{s_0, \dots, s_T\} = \tilde{s}, \{u_0, \dots, u_T\} = \tilde{u})$ with parameters or expectation $(\hat{s}, \hat{\pi})$, where $\pi \in \{1, \dots, K\}$ is a policy that indexes a sequence of control states
  $\widetilde{(u}|\pi) = (u_t, \dots, u_T|\pi)$

Firstly, *generative process* describes the transition probabilities among hidden states which generate observations. Transition probabilities depend on actions which are sampled from *approximate posterior* belief about control states. Belief is formed using the *generative model* (denoted by *m*) of how observations are generated by hidden states. The G*enerative model* encodes belief and hidden states of the agent in term of expectation.

The active inference model assumes that both action and expectation minimize the free energy of observations. That is, expectation minimizes free energy and expectation of control states prescribes actions in each trial.

$$(\hat{s}, \hat{\pi}) = argmin\, F(\tilde{o}, \hat{s}, \hat{\pi}) \qquad \text{Eq 30}$$

$$\Pr(a_t = u_t) = Q(u_t|\hat{\pi}^*) \qquad \text{Eq 31}$$

With:

$$\begin{aligned} F(\hat{o}, \hat{s}, \hat{\pi}) &= E_Q[-\ln P(\tilde{o}, \tilde{s}, \tilde{u}|m)] - H[Q(\tilde{s}, \tilde{u})] \\ &= -\ln P(\tilde{o}\,|m) + D_{KL}\big[Q(\tilde{s}, \tilde{u}) \,||\, P\big(\tilde{s}, \tilde{u}|\widetilde{o}\big)\big] \end{aligned} \qquad \text{Eq 32}$$

The generative mode could be defined as three marginal distributions:

$$P(\tilde{o}, \tilde{s}, \tilde{u}|m) = P(\tilde{o}|\tilde{s})\, P(\tilde{s}|\tilde{u})\, P(\tilde{u}|m) \qquad \text{Eq 33}$$

Thus, heuristically, the decision consists firstly of figuring out which state is the most likely by optimizing its expectation according to free energy and the generative model. Then, after optimizing its posterior beliefs, an action is sampled from the posterior probability distribution over the control state. The environment generates a new observation given the selected action using the generative process and a new decision cycle begins.

For our experimental design, we have 8 possible observations:

Ω = { “Previous black loose”; “Previous black win”; “Previous red loose”; “Previous red win”; “Current black win”; “Current black loose”; “Current red win”; “Current red lose”}

We defined 20 hidden states:

S = {“Previous choice” x “previous reward” x “current correct answer” x “current mode of interaction” ; “Current black win”; “Current black loose”; “Current red win”; “Current red lose”}

The finite set of action is A = {“Choose red”; “Choose black”} bringing the agent from the 16 first possible hidden states to their corresponding 4 last hidden states which are {“Current black win”; “Current black loose”; “Current red win”; “Current red lose”}.

Log prior preferences over the observed states are C = [0 ; 1 ; 0 ; 1 ; 2 ; -1 ; 2 ; -1] meaning that the agent prefers to observe, in decreasing order, a current winning, then a previous winning, a previous defeat and finally a current defeat.

For each trial, prior beliefs about hidden states are equally spread between the four hidden states composed by {“Previous choice” x “previous reward”} leaving unknown the “current mode of interaction” and the “current good answer”.

To allow the agent to learn about the hidden state “current mode of interaction” we added concentration parameters about observation. Concentration parameters are prior about what hidden states lead to what observations, and can be viewed as the number of hidden states’ occurrences encountered in the past. We arbitrarily set this number to 2 for being in “cooperative” mode, when observing a previous winning, and 1 for being in the hidden state “competitive”. Inversely for a previous defeat, we set this parameter at 1 for being in the “cooperative” hidden state and 2 for being in the “competitive” state.

Hierarchical Gaussian Filter

The Hierarchical Gaussian Filter was constructed to model the agent’s learning under a volatile environment (Mathys et al., 2011, 2014). We are interested in a binary state x_1^t of the environment at time *t* (for convenience we will often omit the time index *k*) :

$x_1^k \in \{\text{“Red } card \text{ is the good answer”}\ ;\ \text{“Black } card \text{ is the good answer”}\}$ is causing sensory input u. Thus, we assume the following form of likelihood function:

$$p(u|x_1) = (u)^{x_1}(1-u)^{1-x_1} \qquad \text{Eq 34}$$

Because x_1 is binary, it could be described by a single real number, x_2, the state at the next level of hierarchy. We then define the conditional prior density to map x_2 to the probability x_1 as a Bernnouilli law of parameter $s(x_2)$:

$$p(x_1|x_2) = s(x_2)^{x_1}\big(1-s(x_2)\big)^{1-x_1} \qquad \text{Eq 35}$$

Where $s(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. This prior density gives us $x_1 = 1$ and $x_1 = 0$ equally probable for $x_2 = 0$ and for $x_2 \to +\infty$ or $-\infty$ we have respectively $x_1 \to 1$ or 0. Thus, x_2 is an unbounded parameter of the probability that $x_1 = 1$. In our example, a higher x_2 corresponds to a strong tendency for the red target to be the good answer. The only hypothesis on x_2 is that it evolves with time as a Gaussian random walk.

$$p\left(x_2^{(k)}|x_2^{(k-1)}, x_3^{(t)}\right) = N\left(x_2^{(k)}; x_2^{(k-1)}, \exp\left(\kappa x_3^{(k)} + \omega\right)\right) \qquad \text{Eq 36}$$

Where ω and κ are two free parameters corresponding to the dispersion of the random walk. $x_3^{(k)}$ represents the log-volatility of the environment, meaning the tendency of the red target to be the good answer, and follows a Gaussian random walk. ω represents the volatility independent of the state x_3. Then we can apply the same approach to x_3 as we do to x_2, and so forth, to add as many levels as desired. Here we stop at the fourth level introducing a new free parameter representing the volatility of x_3 :

$$p\left(x_3^{(k)}|x_3^{(k-1)}, \vartheta\right) = N\left(x_3^{(k)}; x_3^{(k-1)}, \vartheta\right) \qquad \text{Eq 37}$$

Then, using the Variational inversion method explained in (Mathys et al., 2011), we can inverse the model to update hidden states given a regular sensory entry $u^{(k)}$. The approximation of the inversion assumes Gaussian posteriors at all levels, with means μ_i and precision (inverse of variance) π_i :

$$x_i^{(k)}|\, u^{(1)}, \dots, u^{(k)}, \chi \sim N\left(\mu_i^{(k)}, \left(\pi_i^{(k)}\right)^{-1}\right) \qquad \text{Eq 38}$$

with χ the set of all free parameters. Thus parameters μ_i and π_i are the sufficient statistic, to be updated after each input u as follows:

$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{1}{2}\kappa_{i-1} v_{i-1}^{(k)} \frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} \qquad \text{Eq 39}$$

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \frac{1}{2}\left(\kappa_{i-1} v_{i-1}^{(k)} \hat{\pi}_{i-1}^{(k)}\right)^2 \left(1 + \left(1 - \frac{1}{v_{i-1}^{(k)} \pi_{i-1}^{(k-1)}}\right)\delta_{i-1}^{(k)}\right) \qquad \text{Eq 40}$$

With

$$v_i^{(k)} = \begin{cases} t^{(k)} \exp\left(\kappa_i \mu_{i+1}^{(k-1)} + \omega_i\right), \ i = 1, \dots, n-1 \\ t^{(k)}, \ i = n \end{cases} \quad \text{Eq 41}$$

$$\hat{\mu}_i^{(k)} = \mu_i^{(k-1)} \ \ by\ definition$$

$$\hat{\pi}_i^{(k)} = \frac{1}{\sigma_i^{(k-1)} + v_i^{(k)}} \ \ by\ definition$$

$$\delta_i^{(k)} = \frac{\sigma_i^{(k)} + \left(\mu_i^{(k)} - \hat{\mu}_i^{(k)}\right)^2}{\sigma_i^{(k-1)} + v_i^{(k)}} by\ definition$$

Bayesian sequence learner

The n-BSL (Bayesian sequence learner) is a model which tracks probabilities of a certain outcome "a" given the previous *n* outcomes as a Gaussian function:

$$P(a = \text{"Red target is the good answer"} | \ S_i \ ) = \ N(\mu_i, \sigma_i)$$

With S_i the sequence of the *n* last outcomes "a". For each observation at time t, the update is a Laplace-Kalman rule:

$$\sigma_i^{t+1} = \frac{1}{\frac{1}{\sigma_i^t + \Omega} + s(\mu_i^t) * \left(1 - s(\mu_i^t)\right)} \quad \text{Eq 42}$$

$$\mu_i^{t+1} = \mu_i^t + \left(\sigma_i^{t+1} + \Omega\right) * \left(a^t - s\left(\mu_i^t\right)\right) \quad \text{Eq 43}$$

With $a^t = 1$ if "Red target is the good answer" and $a^t = 0$ if "Black target is the good answer", $s(x) = \frac{1}{1+e^{-x}}$, and Ω is a free parameter representing prior volatility.

Win-Stay / Lose-Switch model

This model reproduces a heuristic behavior, precisely "I keep the same option if I just won, I switch if I just lost". To implement that, we use two pseudo Q-values, $V^{stay} = 1$ for the action of stay and $V^{switch} = -1$ for the action of switch. Then we use the sigmoid function to compute the probability of choosing the same option as the previous trial:

$$p^{stay} = s\left(V^{stay} - V^{switch}\right) \quad \text{Eq 44}$$