Learning from Instruction without Shared Meanings

Active Learning from Uncalibrated Brain Signals

Manuel Lopes

How to program and instruct robots in an intuitive way?

- Learning from Demonstration
- Verbal commands
- Gestures
- Specialized Interfaces
- Remote control
- …

Learning from Demonstration

Pros

- Natural/intuitive (is it?), in most cases the demonstror is an expert in the system
- Facilitates social acceptance

Cons

- Requires an expert with knowledge about the task and the learning system
- Long and Costly Demonstrations
- No Feedback on the Learning Process (on most methods)
- Common interface for all users
- Lack of personalization
- Need for calibration in many cases

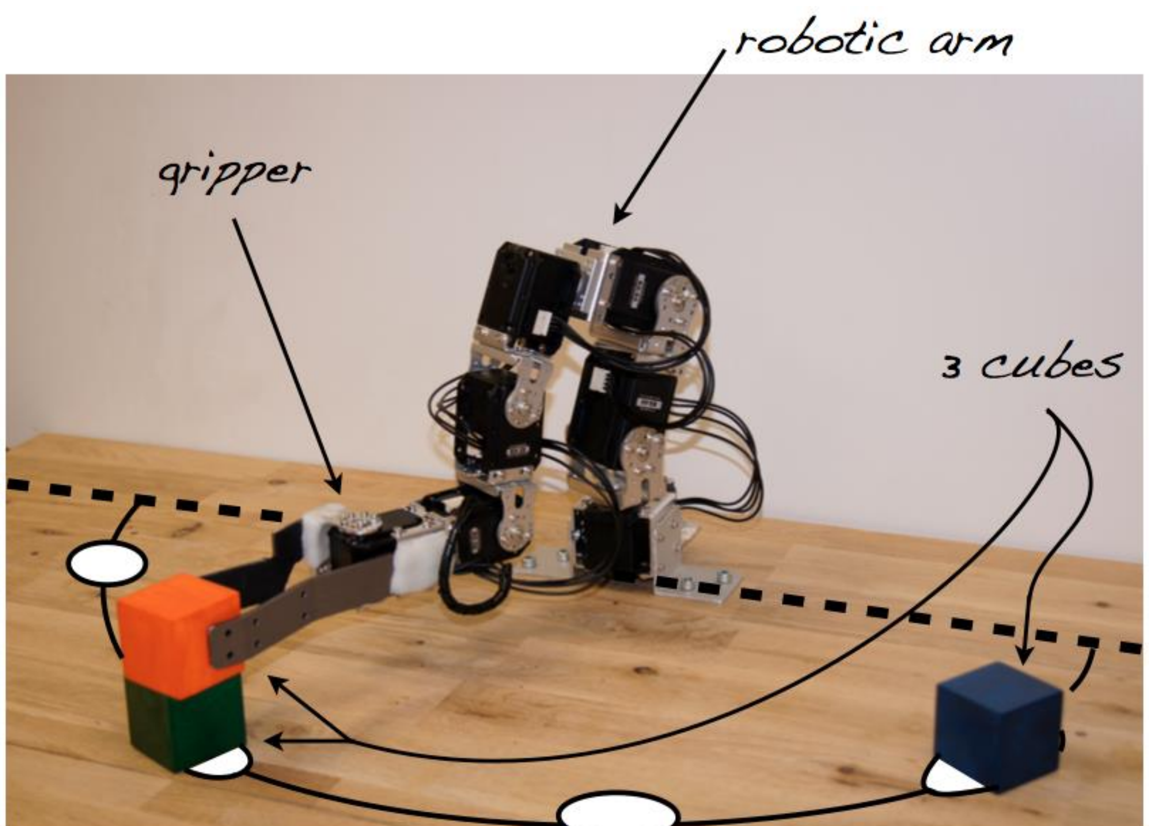
How to improve learning from a user

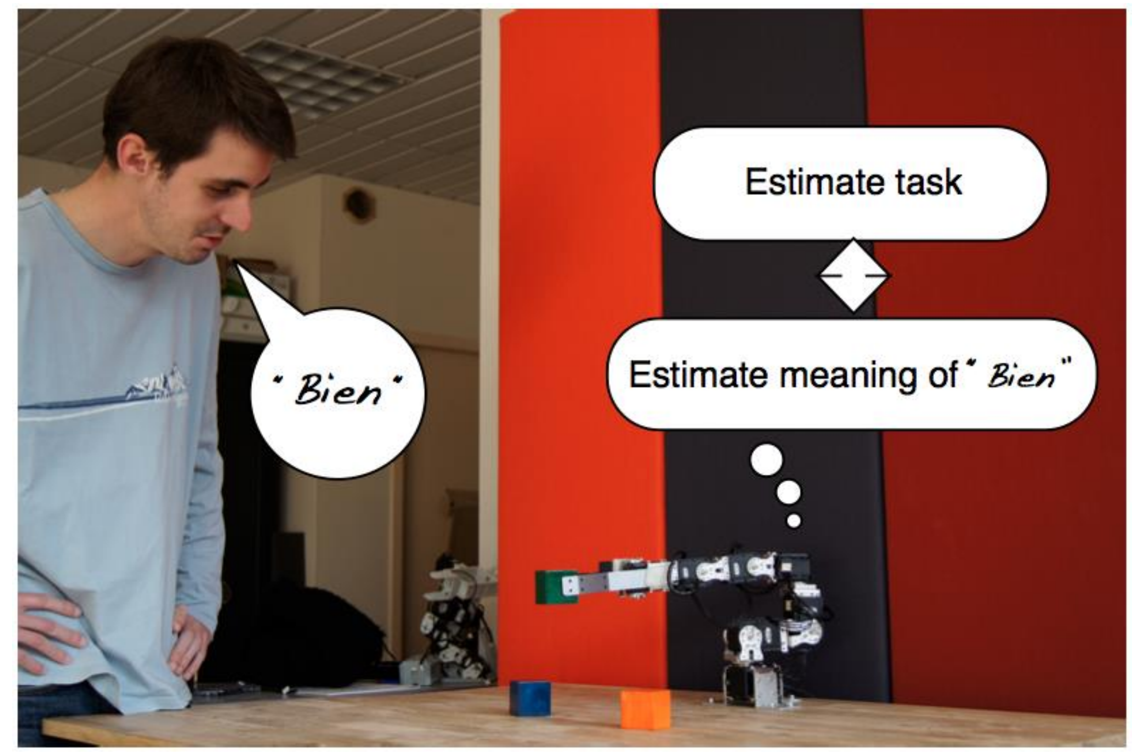
- Uncertainty Modelling to Evaluate the Quality of Learning
- Allow Active Requests from the User
- Adapt to the User Preferred Way of Interaction

Example Scenario
Control a robot with verbal commands

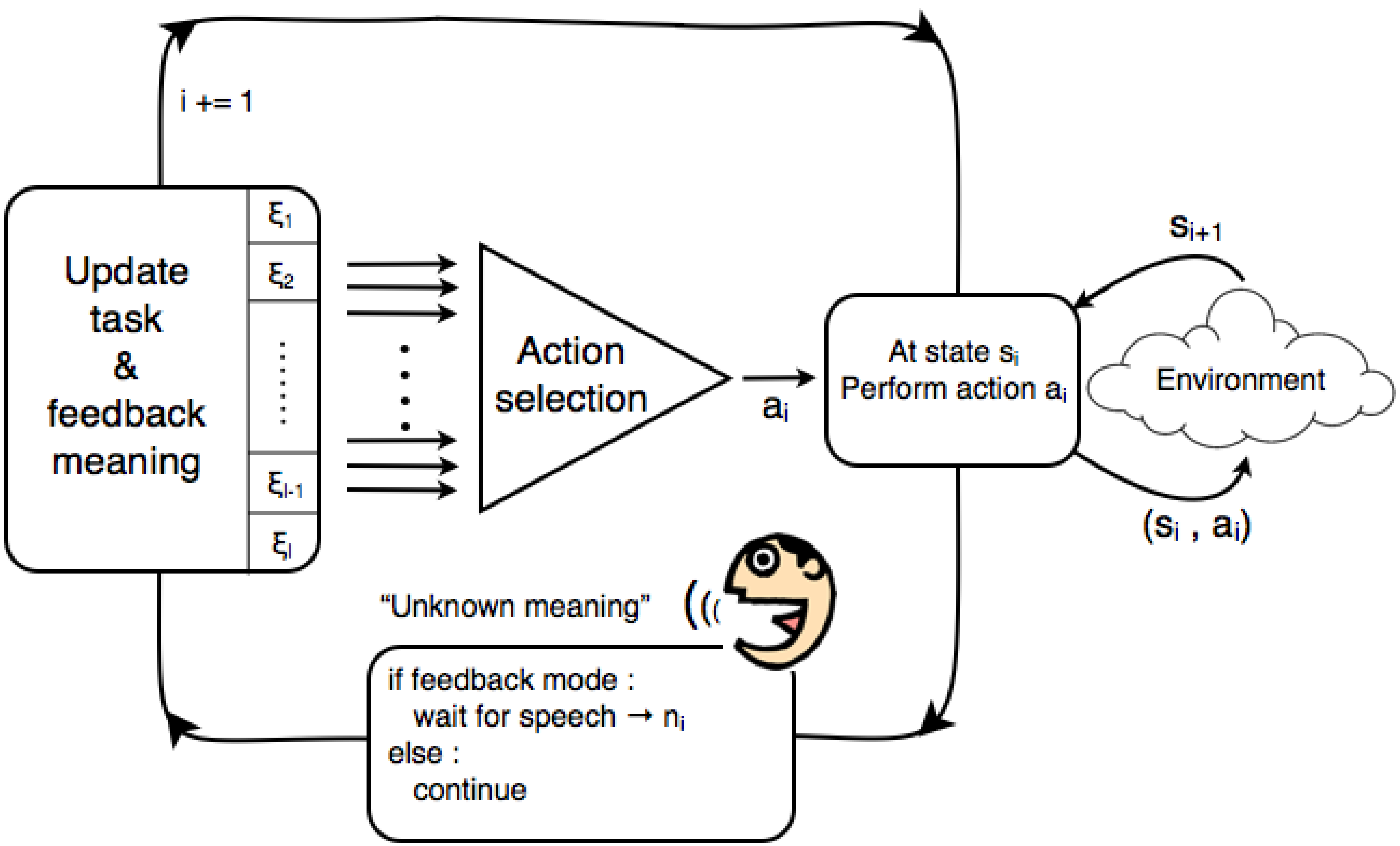
Need for:

- a dedicated speech recognition system
- pre-defined states, commands, actions



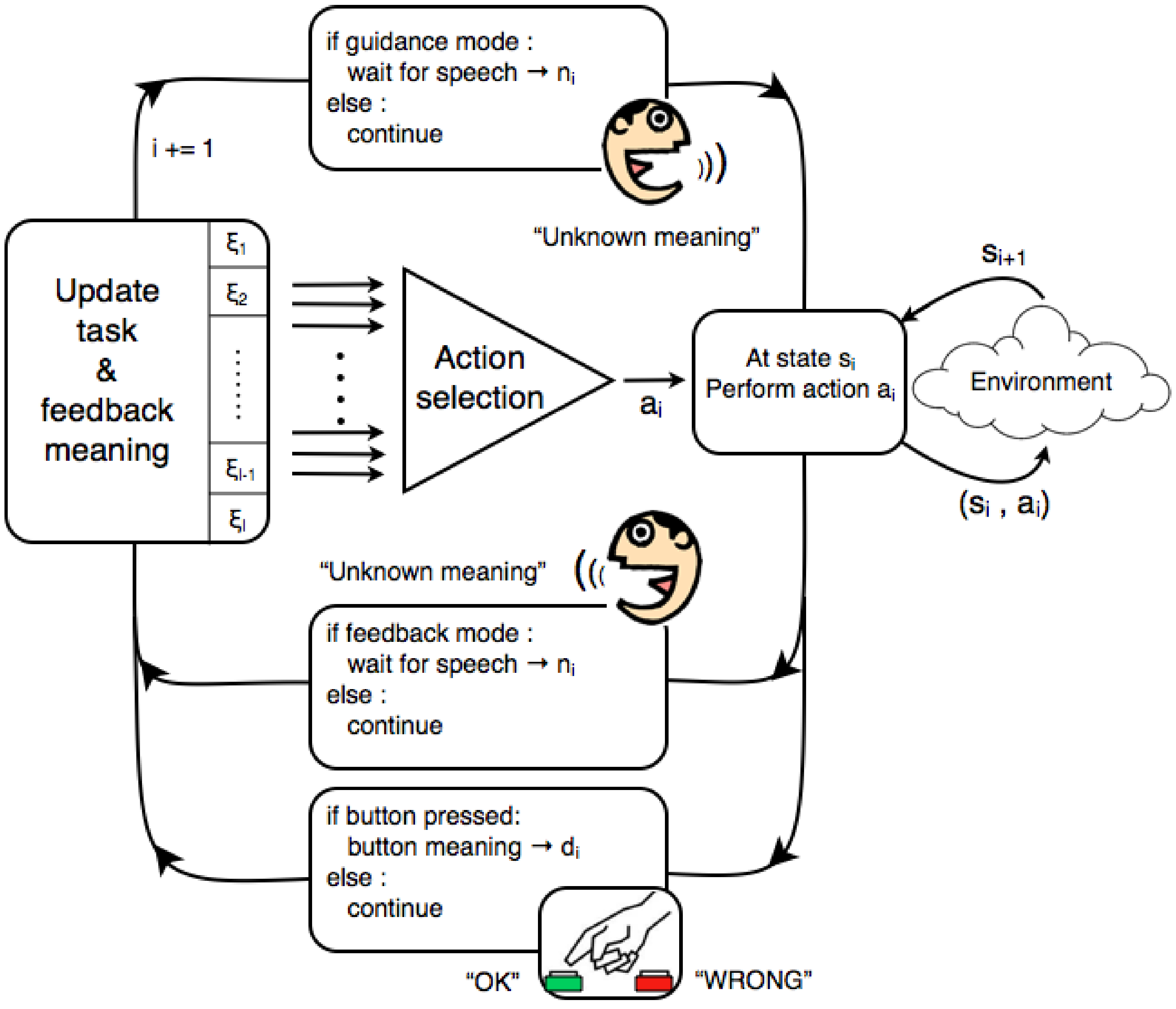


What if the interaction commands are unknown?

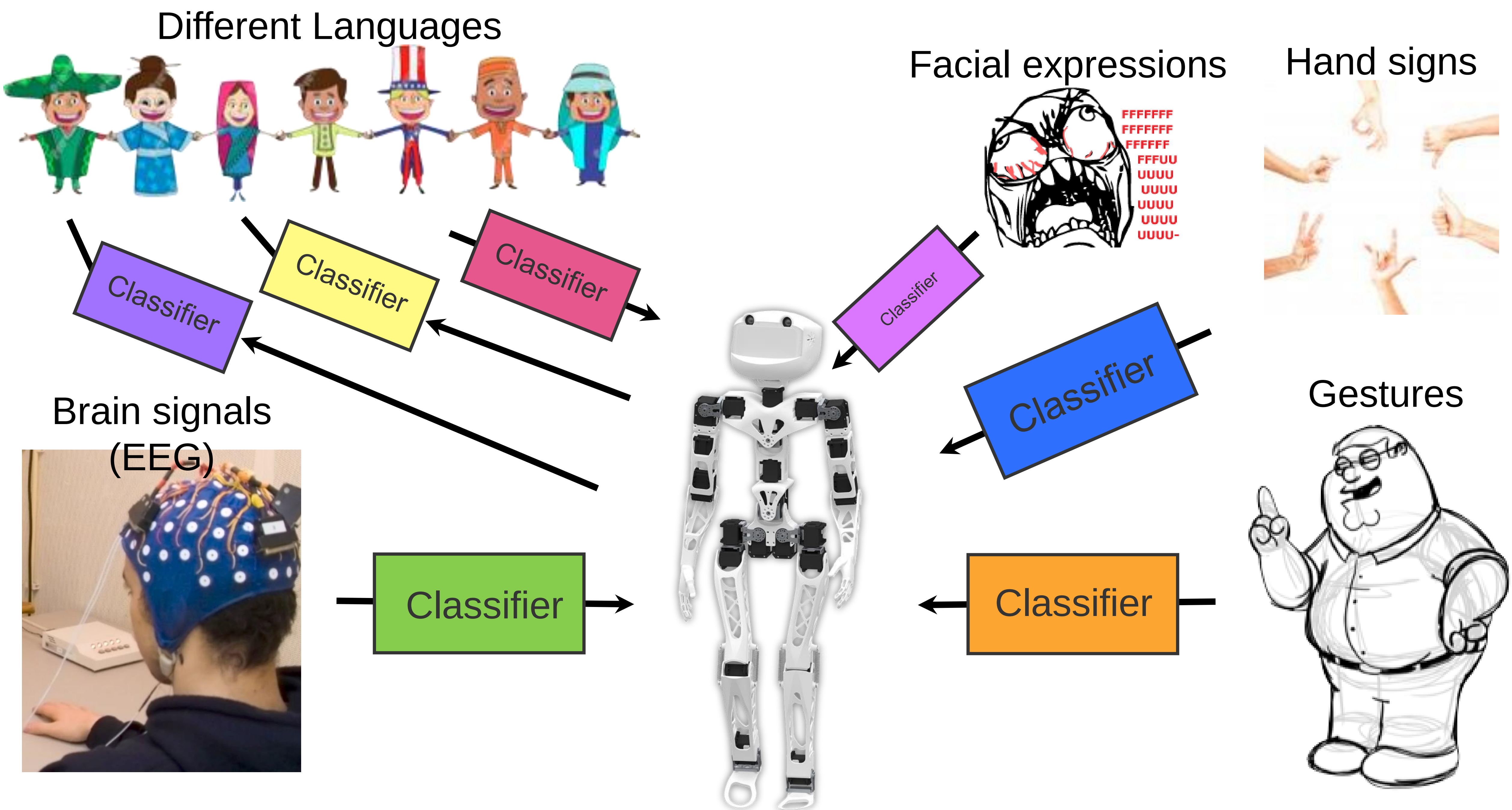


More realistic case

Combine known/calibrated interaction commands with new, user-defined, interaction commands

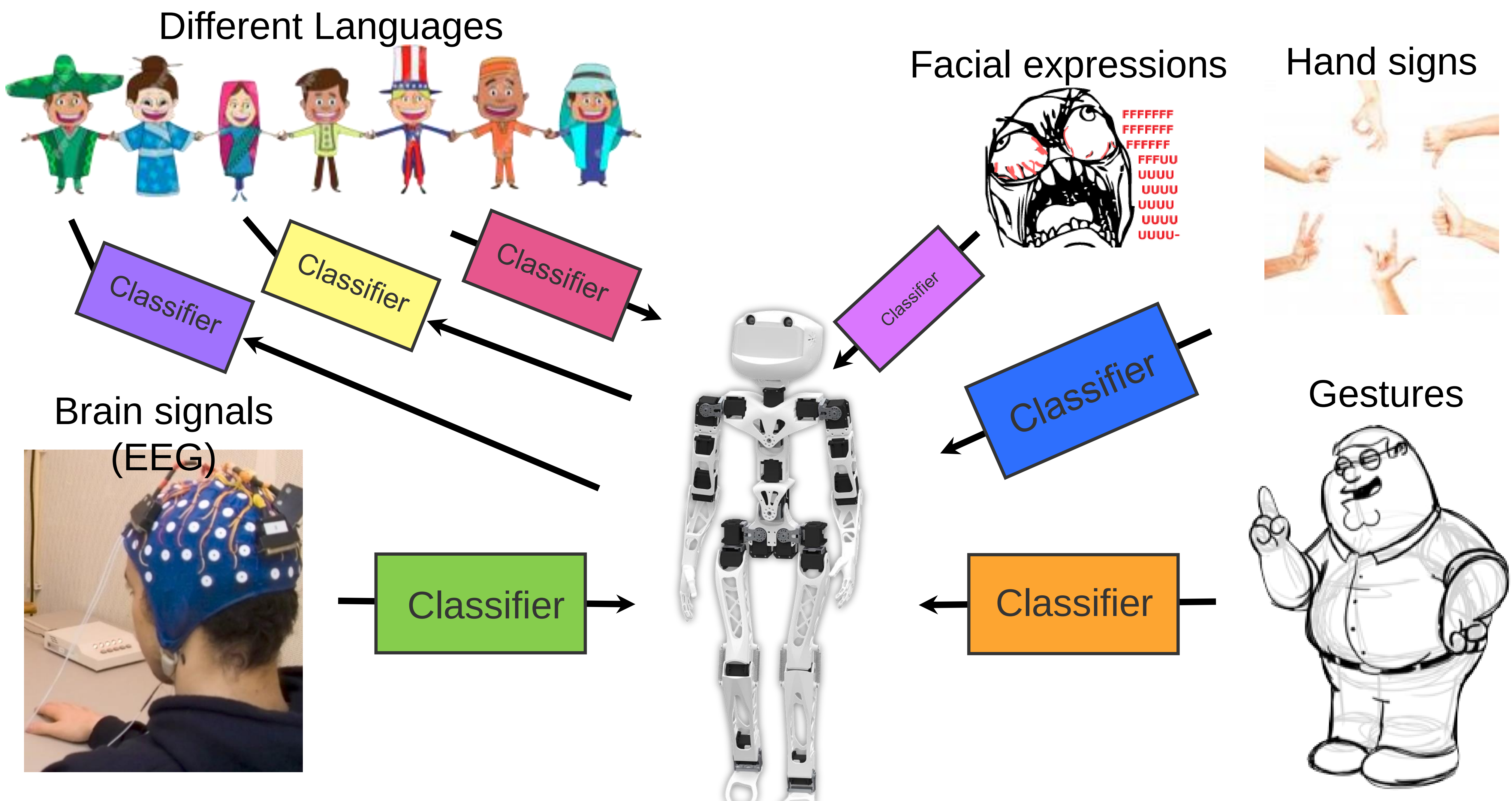


Everyone has their own preferences, skills, and limitations.



Requires a specific calibration for each user and modality

Different people, with their own preferences, skills, and limitations.



Can we adapt <u>automatically and online</u> to each user’s own preferred teaching signals?

Learning Symbols for Human-Robot Collaboration

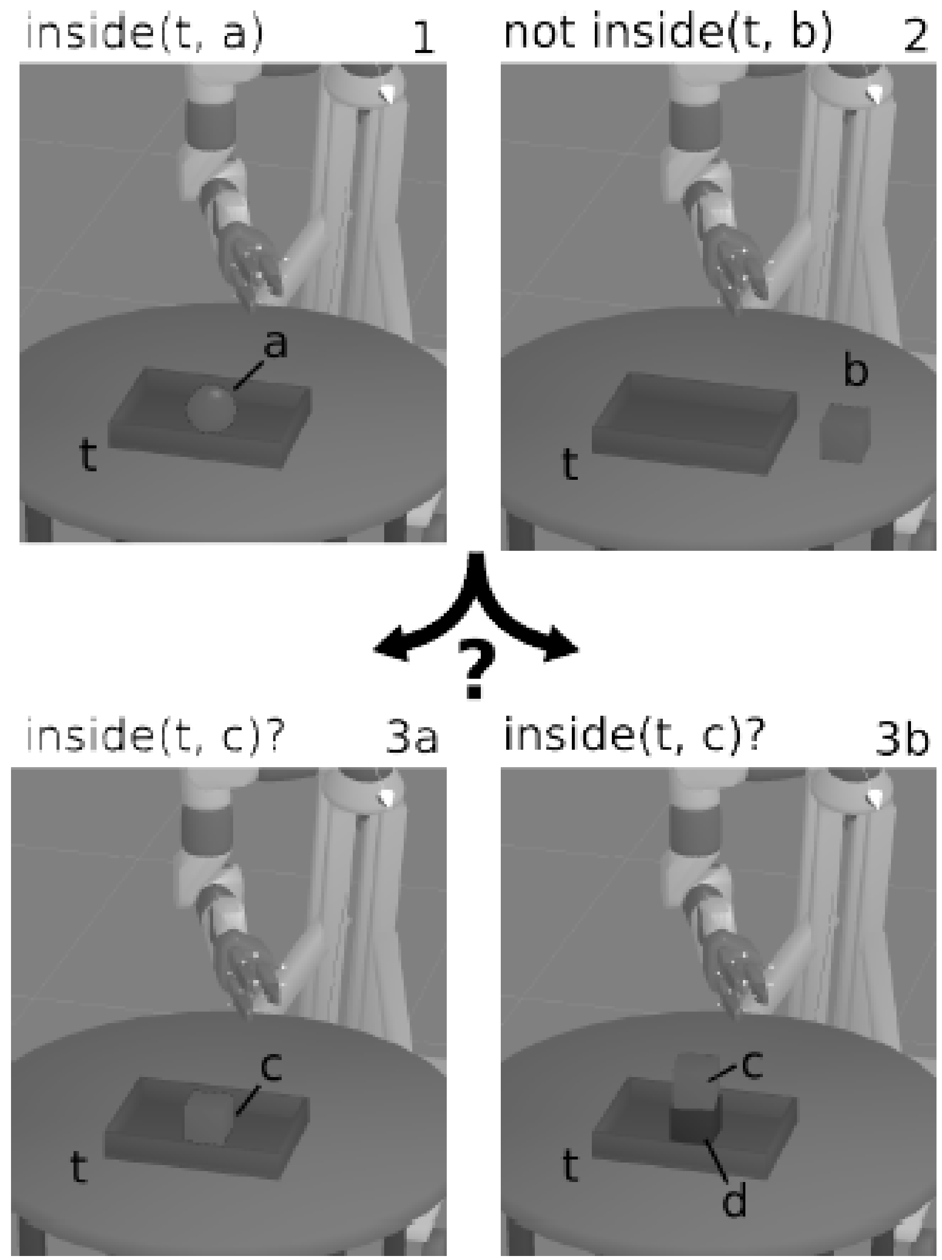
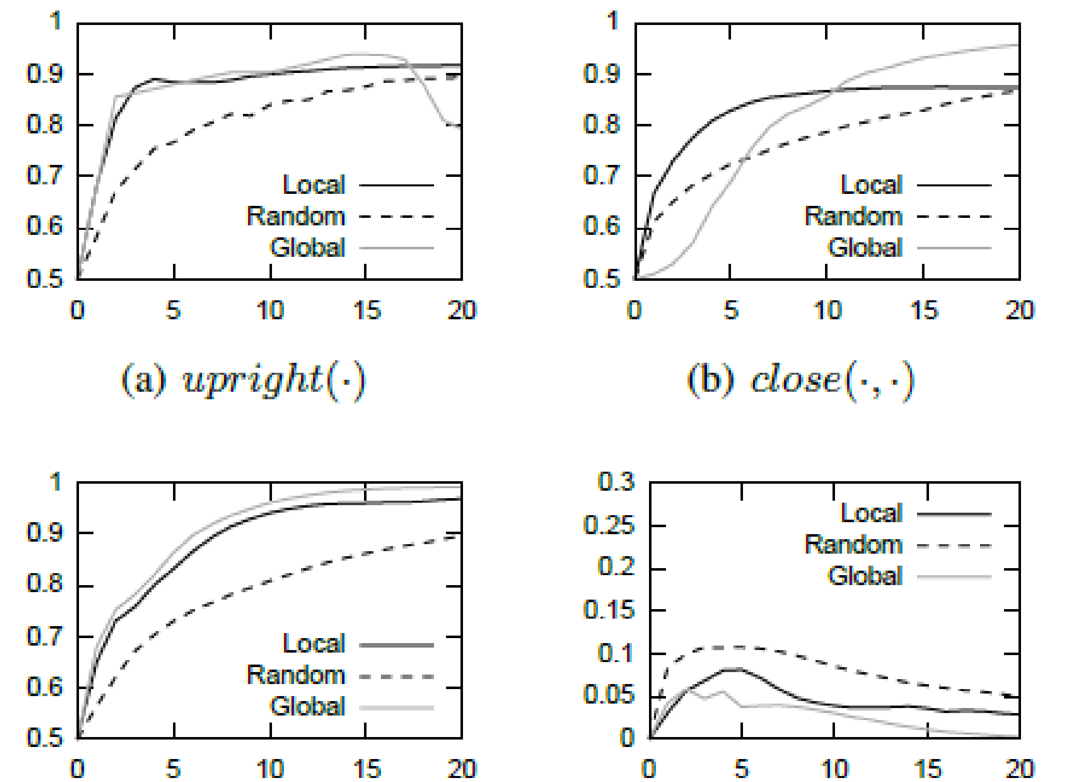


Figure 1: In active learning of grounded relational symbols, the robot generates situations in which it is uncertain about the symbol grounding. After having seen the examples in (1) and (2), the robot can decide whether it wants to see (3a) or (3b). An actively learning robot takes its current knowledge into account and prefers to see the more novel (3b).

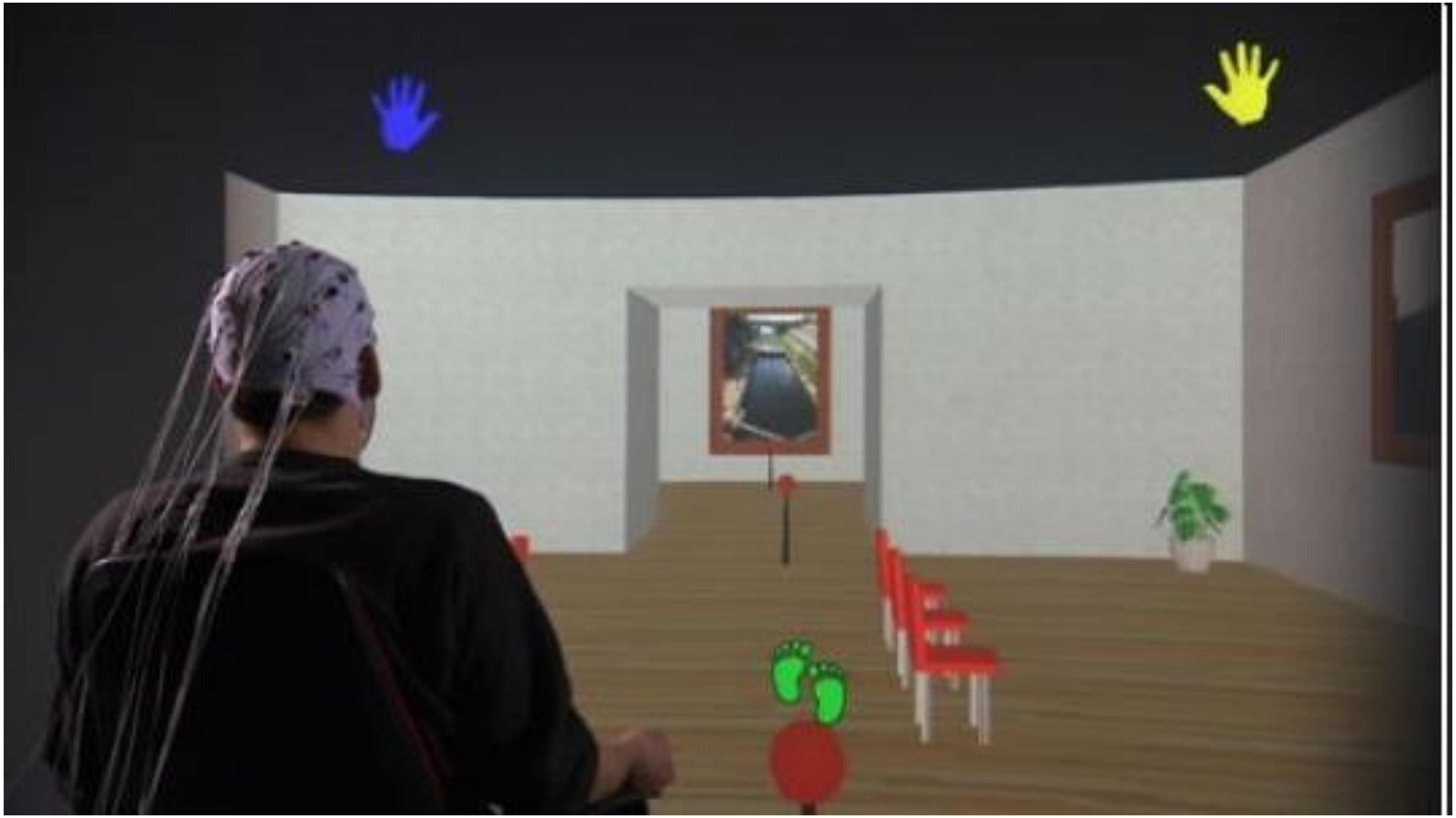


(a) $upright(\cdot)$ (b) $close(\cdot,\cdot)$ (c) $on(\cdot,\cdot)$ (d) Standard deviation of *on* learner

Kulick, J., Toussaint, M., Lang, T., and Lopes, M. (2013). Active learning for teaching a robot grounded relational symbols. In IJCAI.

A more challenging scenario

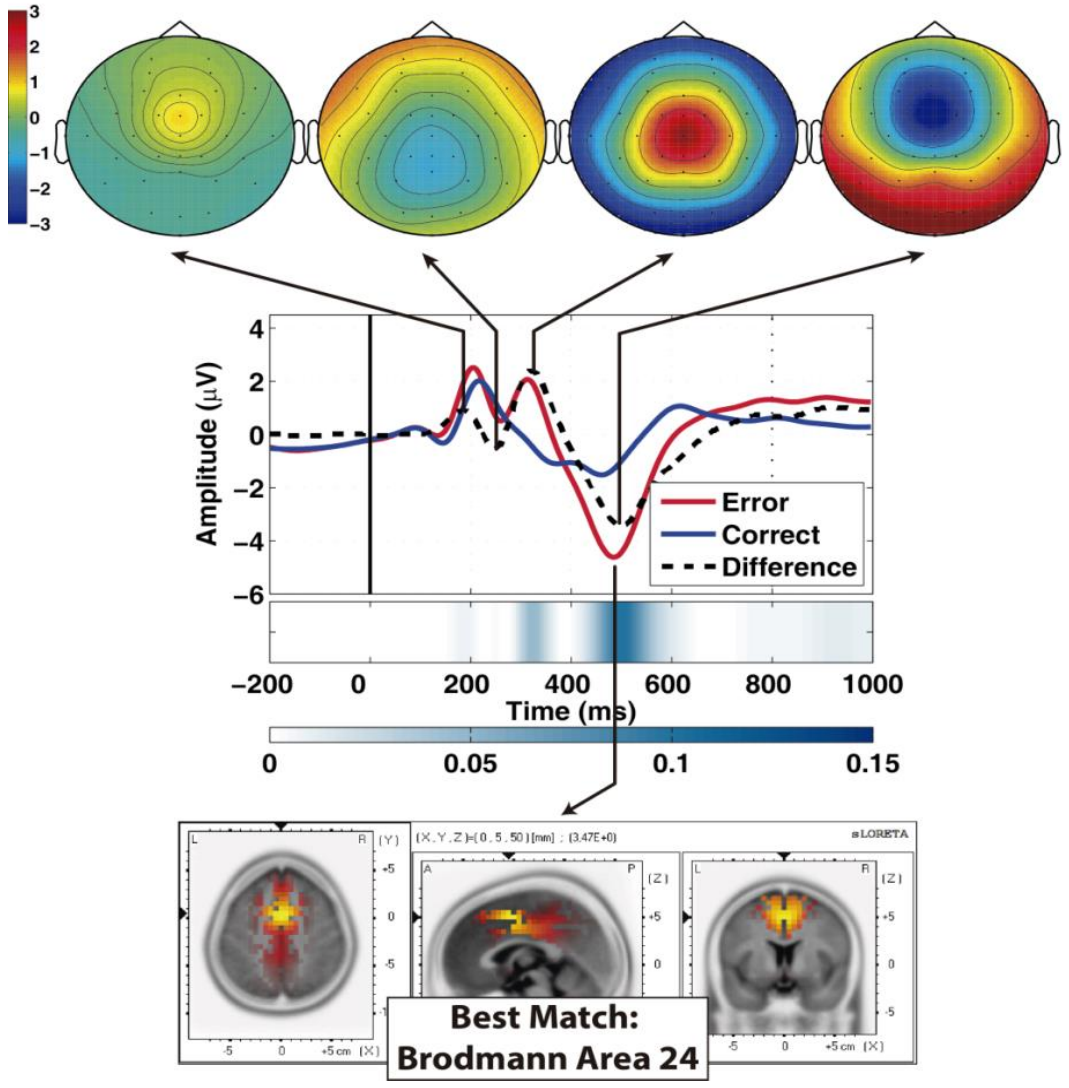
- Control Based on Brain Signals



Error Potentials (ErrP)

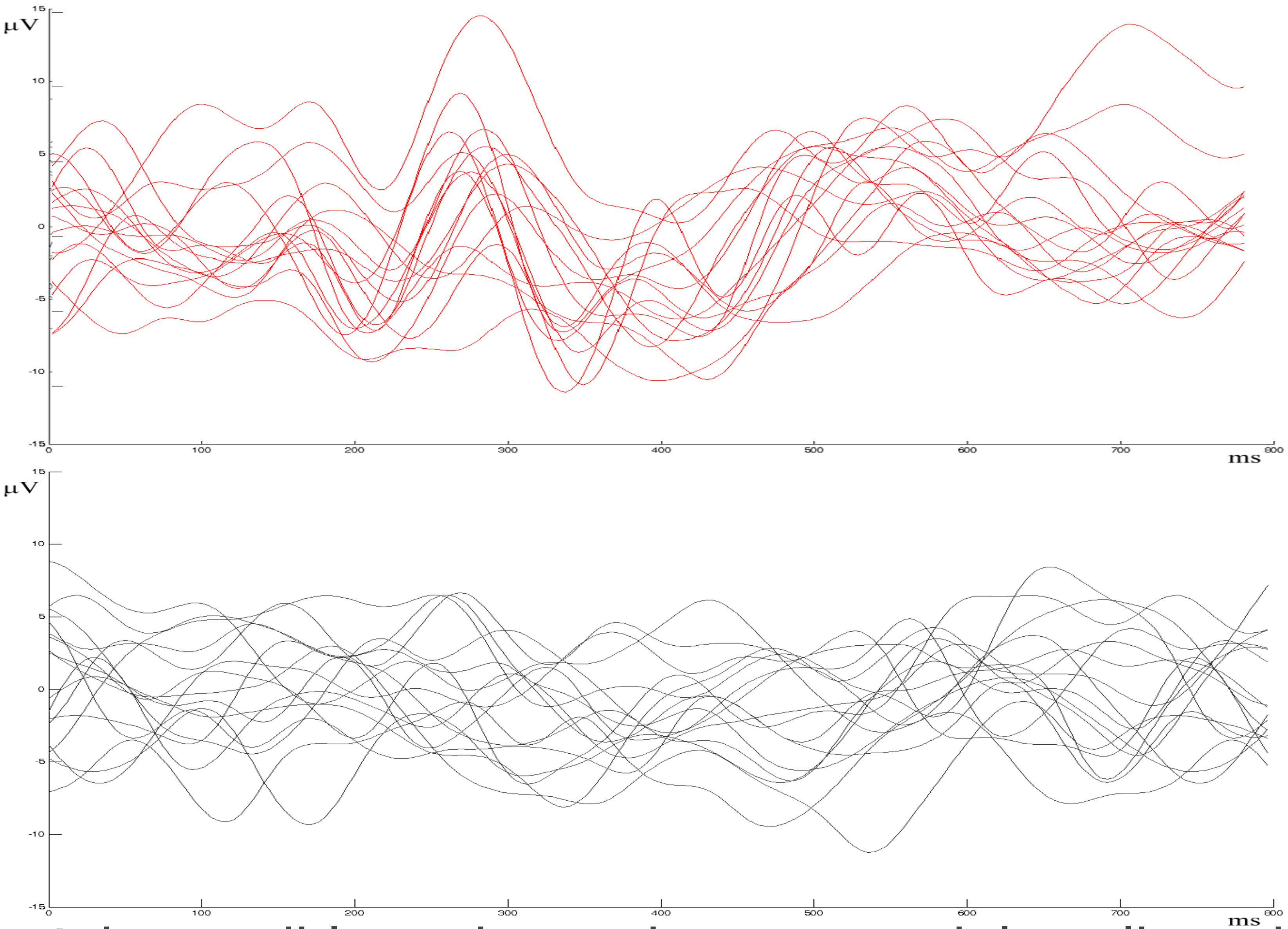
1 + 1 = 2

1 + 1 = 7



- The error potentials (ErrPs) are event-related potentials (ERPs), that occur after the observation of erroneous events.
- Negative deflection (N2, P3, N4)

Introduction: ErrPs in Single Trial



ERROR

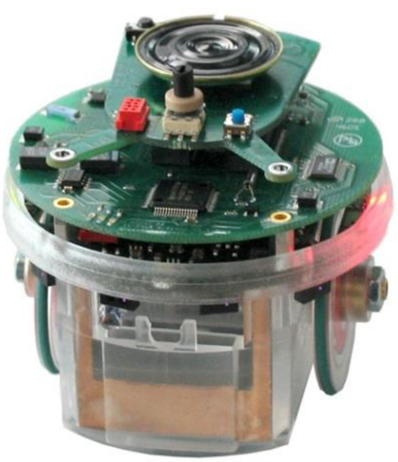
CORRECT

- It is possible to detect these potentials online with accuracies over 70% **[Ferrez08, Chavarriaga10, Iturrate 2010]**

- Applications: Learning **[Chavarriaga10],** Control of devices **[Iturrate13],** Adaptation of classifier **[Blankertz03, Blumberg11, Llera11, Sanchez13]**

Brain Control based on ErrP

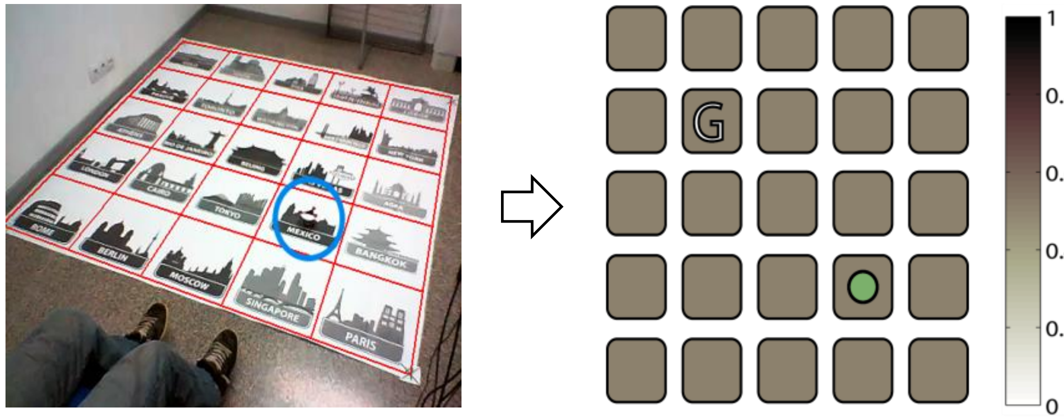
- **Goal reaching task with a real robot (ePuck)**
  - Non-holonomic actions: **turn then advance**
- **32 EEG channels + 6 EOG channels**
- **Continuous error potential detection**
  - stop when an error is detected



Control using error potentials

- **Inverse reinforcement learning**
- **Exploit the task constraints**
  - Finite set of possible goal locations
  - Precompute each optimal policy
- **Continuous updating**
  - Execute action untill error detected
- **Recursive Bayesian filtering**

Targets are still discrete: 5x5 grid



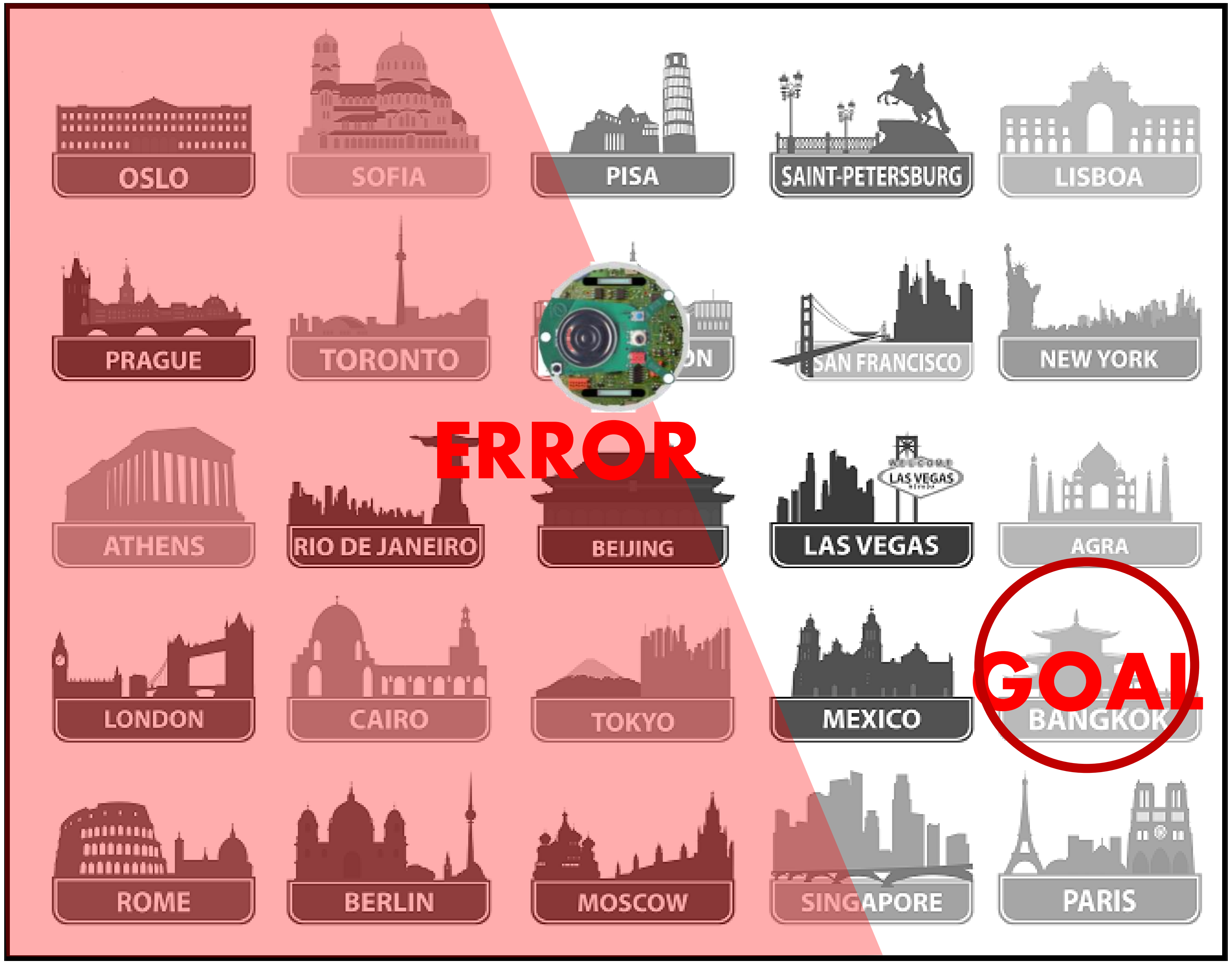
$$p(\pi^*{}_i|(a,s,x)_{1..t}) \propto p\left(a_t\middle|\pi^*{}_i,(\mathrm{s},\mathrm{x})_t\right)\cdot p(\pi^*{}_i|(a,s,x)_{1..t-1})$$

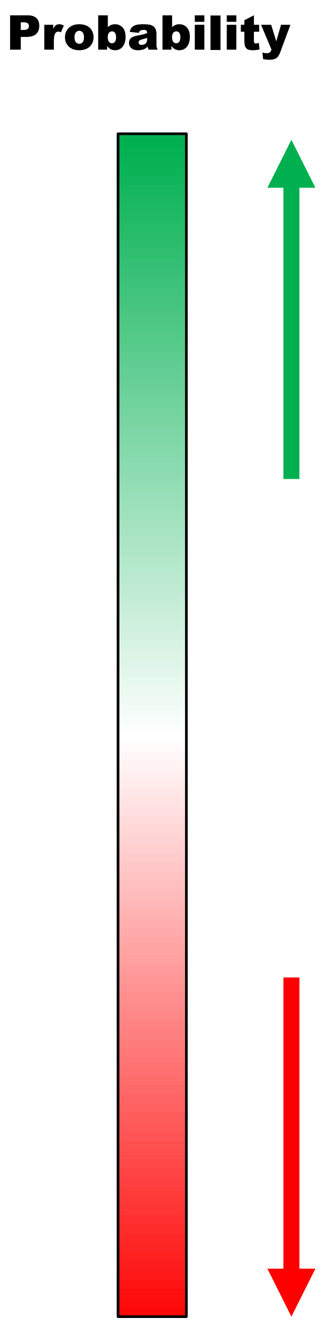
POSTERIOR LIKELIHOOD PRIOR

s: state
a: action
x: eeg
π: policy

Iturrate, Montesano, Minguez, EMBC 2013

Likelihood: Turn



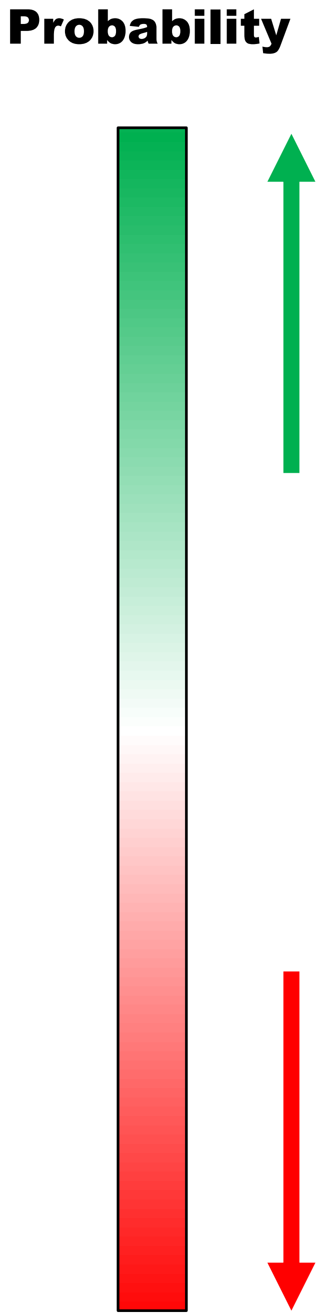


$$p(\mathbf{a}_t|\pi_i^*, (\mathbf{s},\mathbf{x})_t) = \begin{cases} k_n & \text{if } (p(c_t = 1|\mathbf{x}_t) \geq T_e) \wedge (\theta_t - \theta_{t-1} > 0) \wedge (\theta_i - \theta_t \in (\;\;0, \pi\,]), \\ k_n & \text{if } (p(c_t = 1|\mathbf{x}_t) \geq T_e) \wedge (\theta_t - \theta_{t-1} < 0) \wedge (\theta_i - \theta_t \in (-\pi,\; 0\,]), \\ 1 & \text{otherwise} \end{cases}$$

$k_n = 0.2$

$Te = 0.8$

Likelihood: Advance



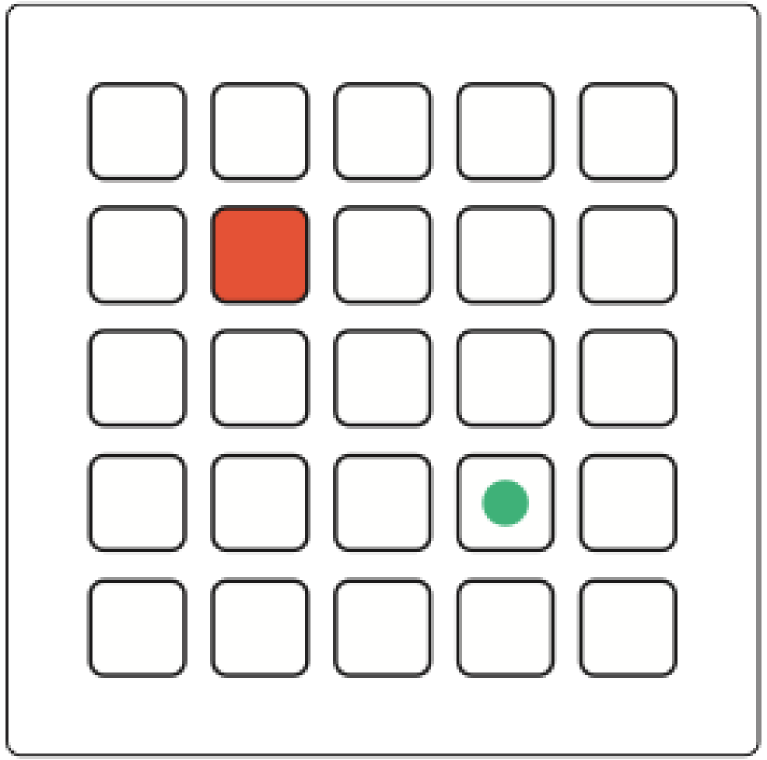
$$p(\mathbf{a}_t|\pi_i^*, (\mathbf{s}, \mathbf{x})_t) = \begin{cases} 1 + k_p \cdot \mathcal{N}(\theta_t - \theta_i \; ; \; 0, \sigma) & \text{if } (p(c_t = 1|\mathbf{x}_t) < T_e), \\ 1 - k_n \cdot \mathcal{N}(\theta_t - \theta_i \; ; \; 0, \sigma) & \text{if } (p(c_t = 1|\mathbf{x}_t) \geq T_e) \end{cases}$$

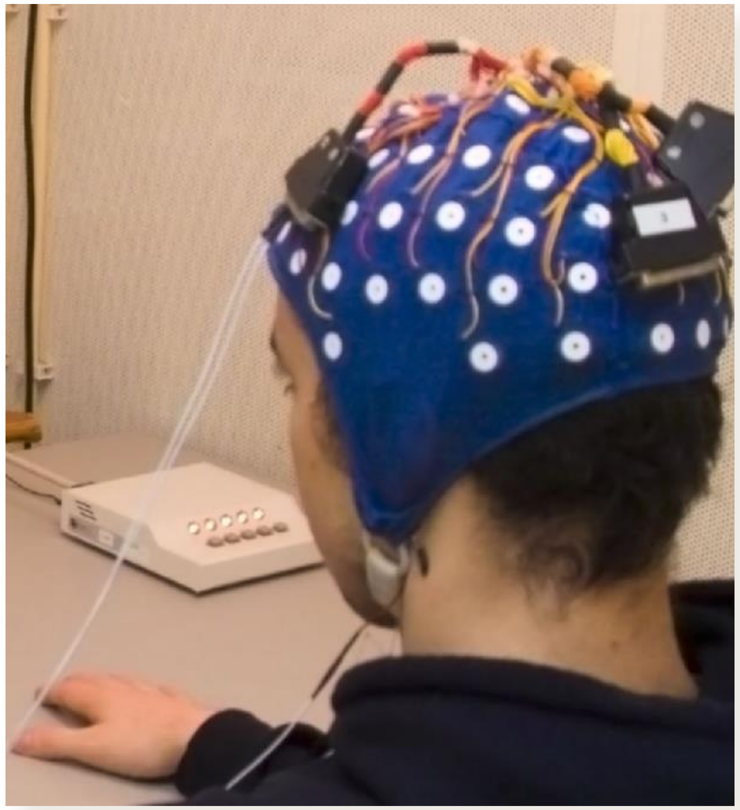
$k_p = 0.01$
$k_n = 0.7$
$Te = 0.8$

Preliminary Results

Robot goes from Mexico to Pisa

Experimental setup





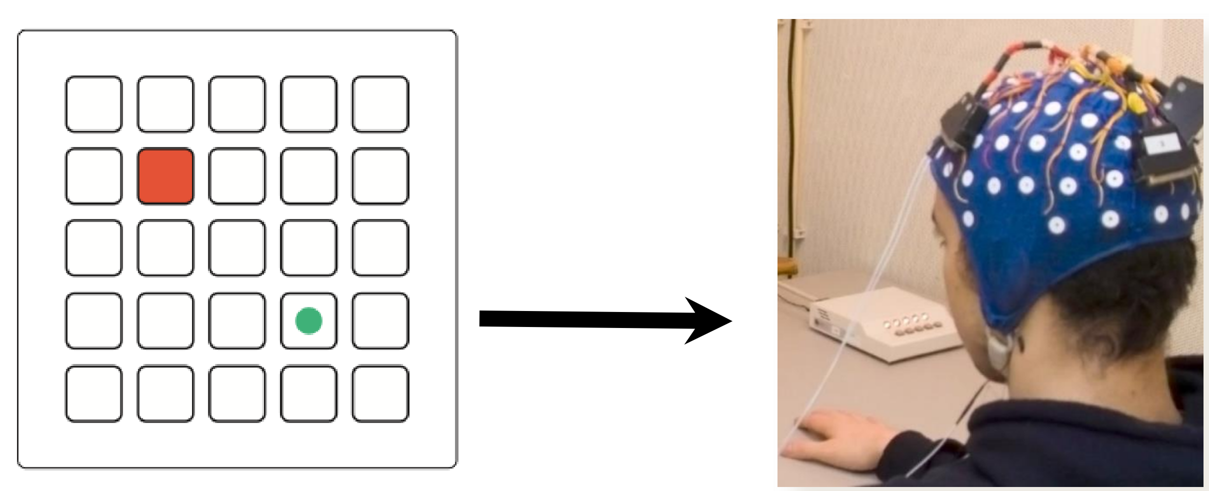
Iturrate, I., L. Montesano, and J. Minguez. "Task-dependent signal variations in EEG error-related potentials for brain–computer interfaces." Journal of neural engineering 10.2 (2013): 026024.

Calibration

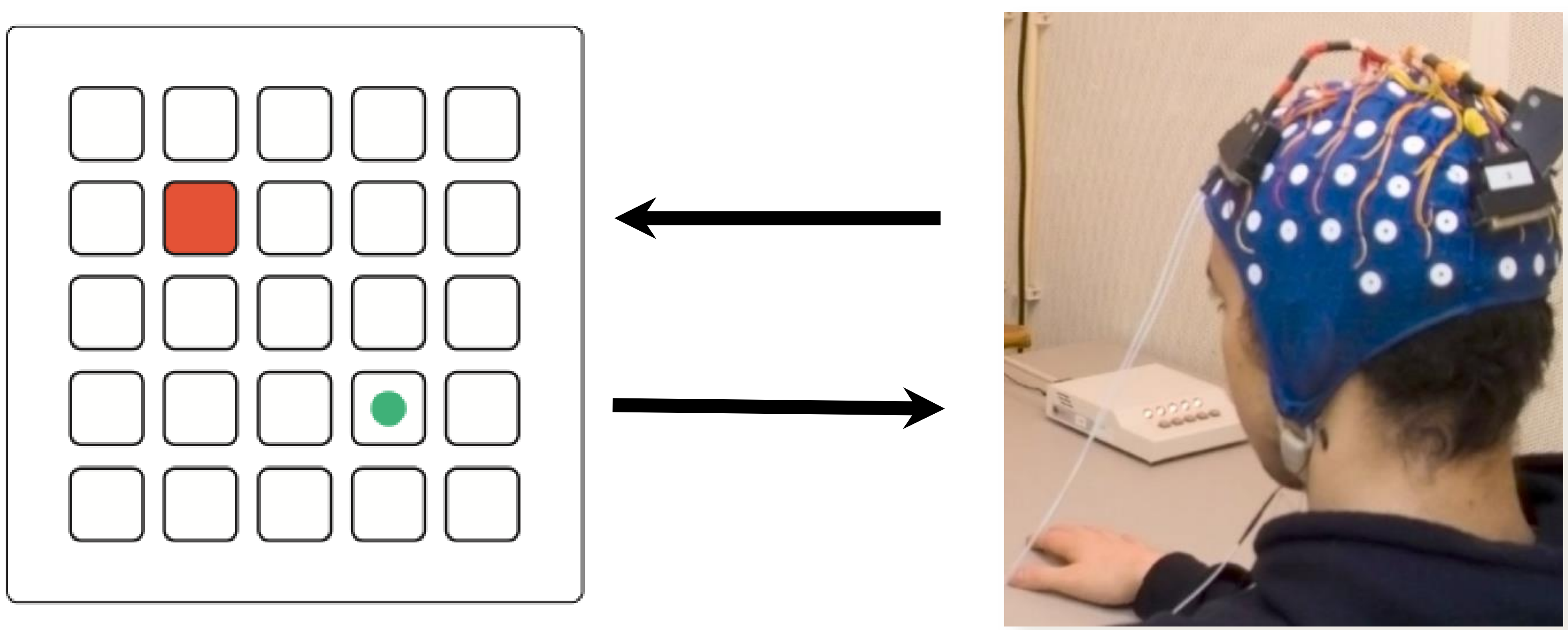
- The user is instructed to move the cursor to a target (red)
- The cursor moves and the brain activity is recorded
- By comparing the signals with the signals expected due to the task we can learn a classifier

Problems:

- Signals change:
  - with the task
  - with time
- Difficult to know when the activity is well detected, or when the calibration can finish



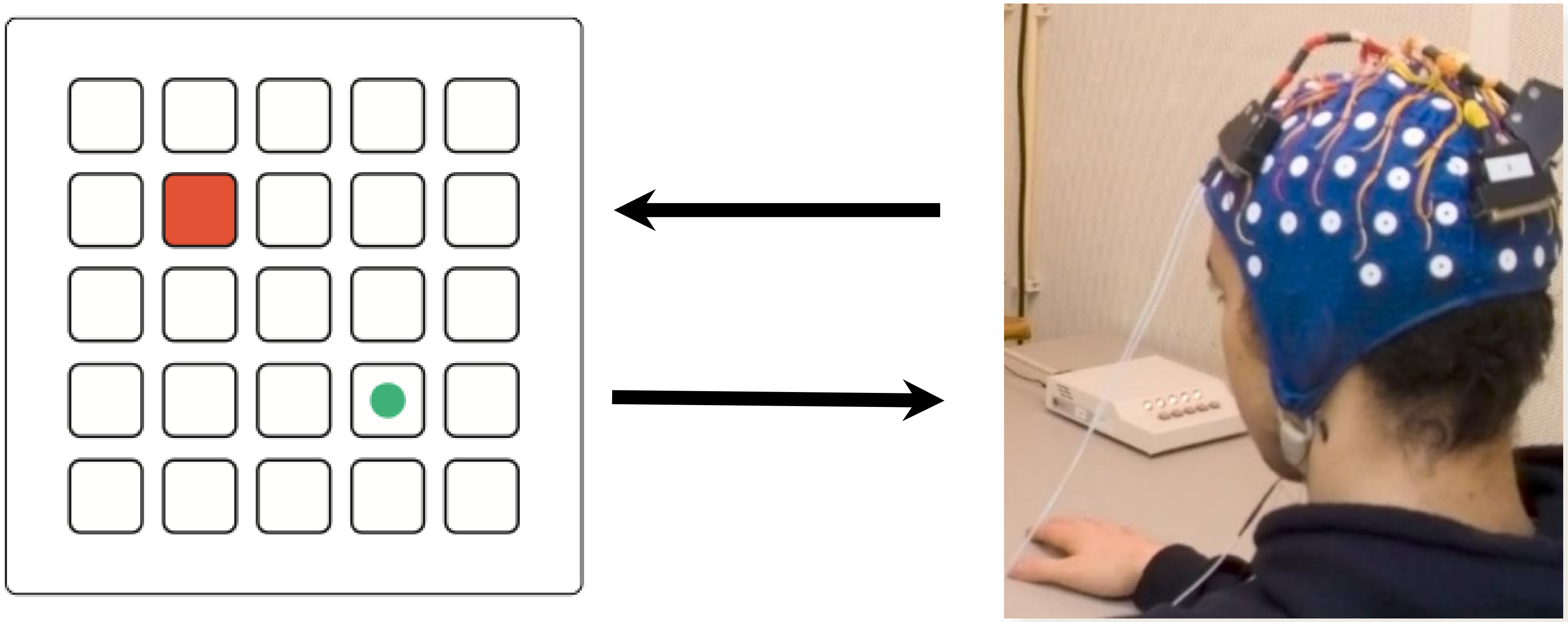
Is it possible to simultaneously do execution and calibration



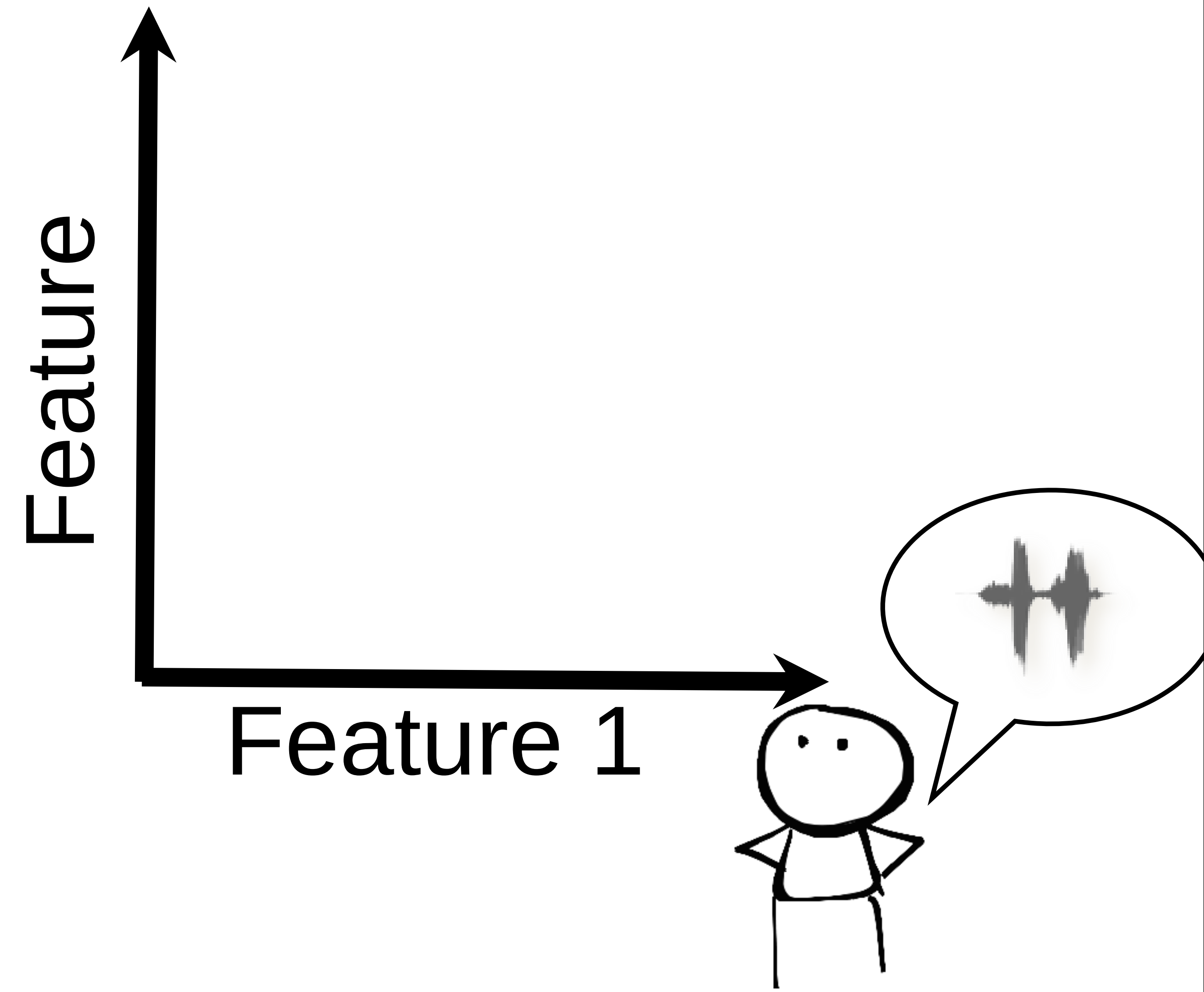
Is it possible to simultaneously do execution and calibration

Assumptions:

For the correct task, the classifier will have the best classification rate.



Toy example



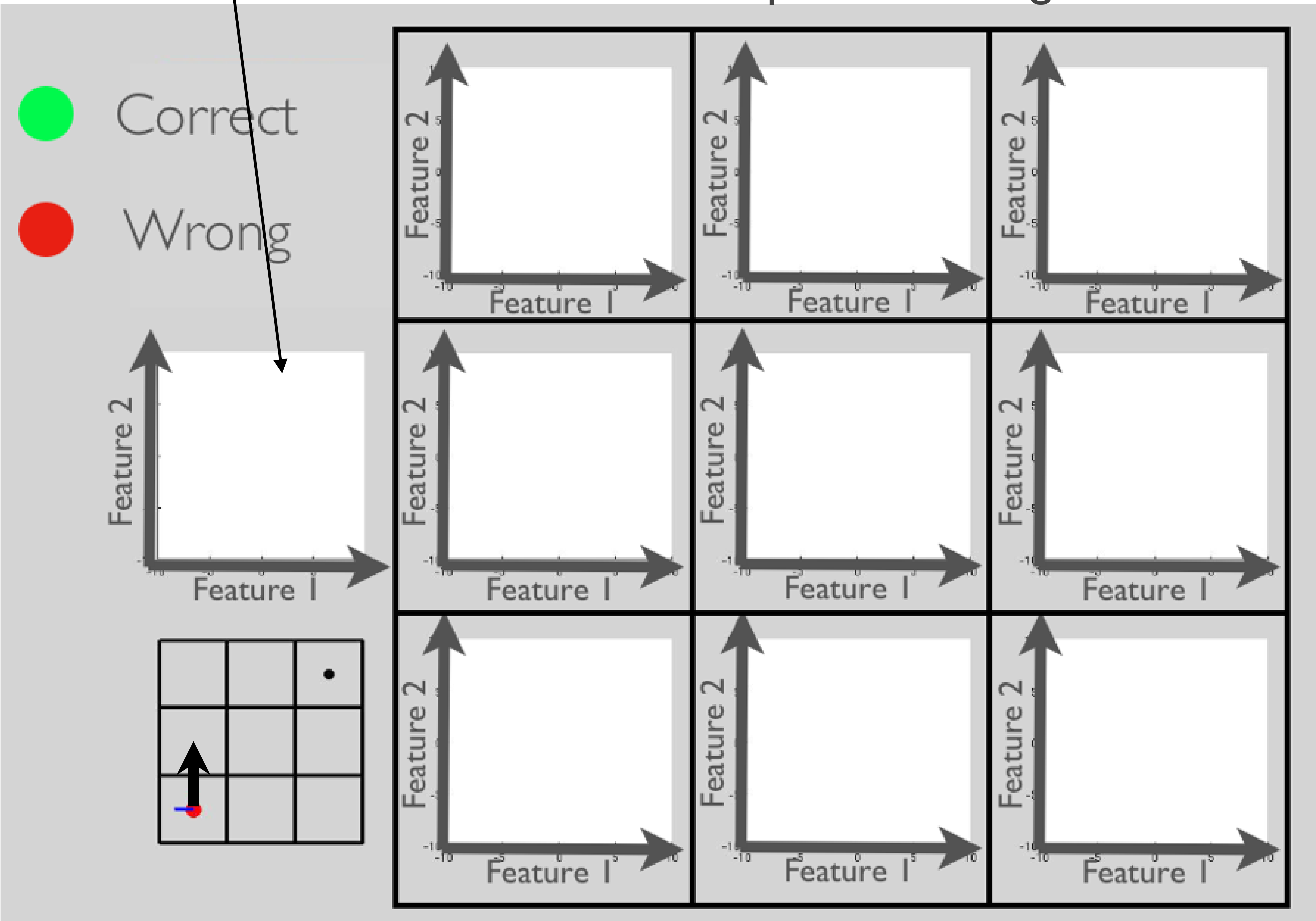
Sequential task

{state, action, instruction} interaction loop

Instructions are feedback on the robot' actions

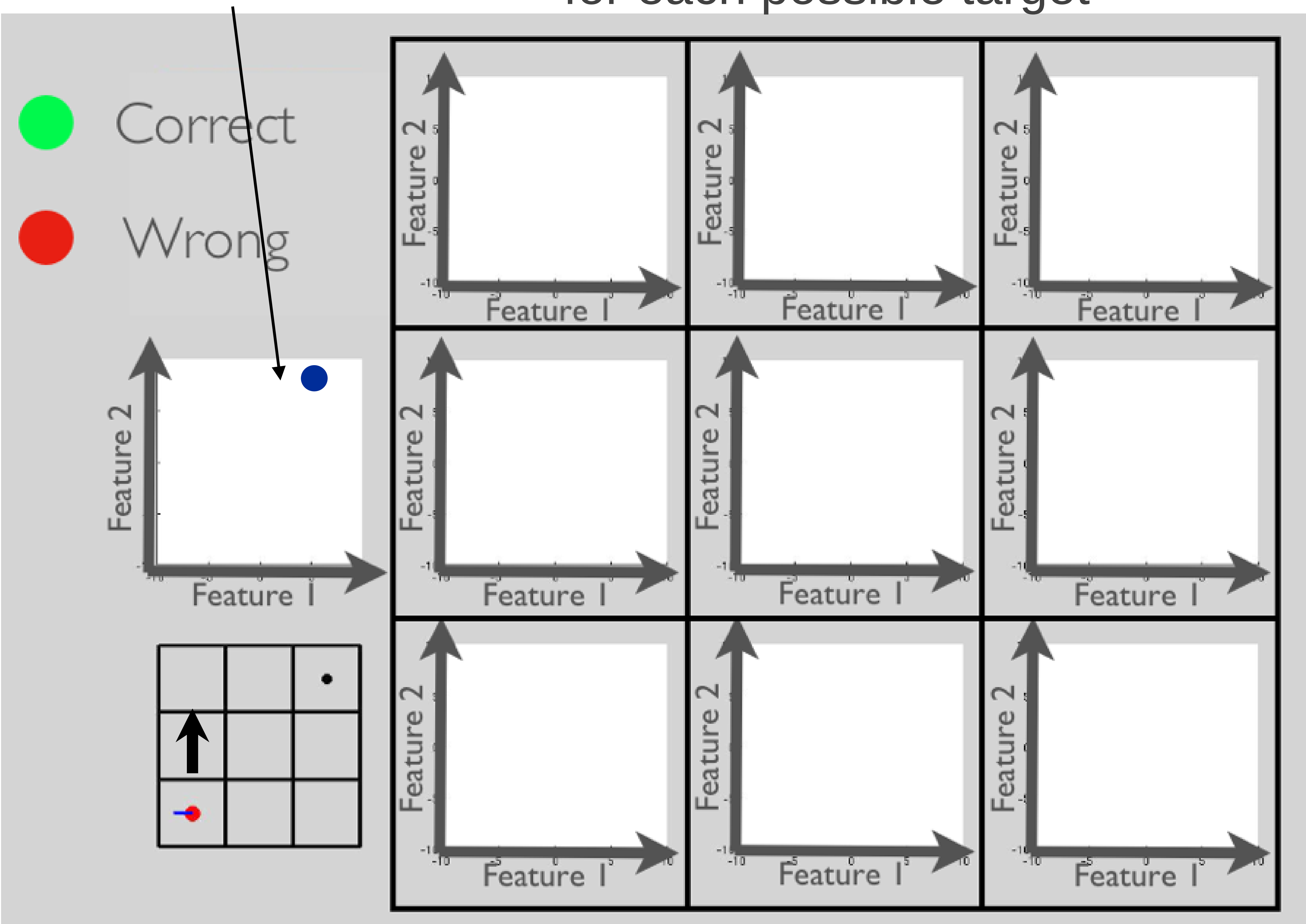
Observations

Interpretation (colors) for each possible target



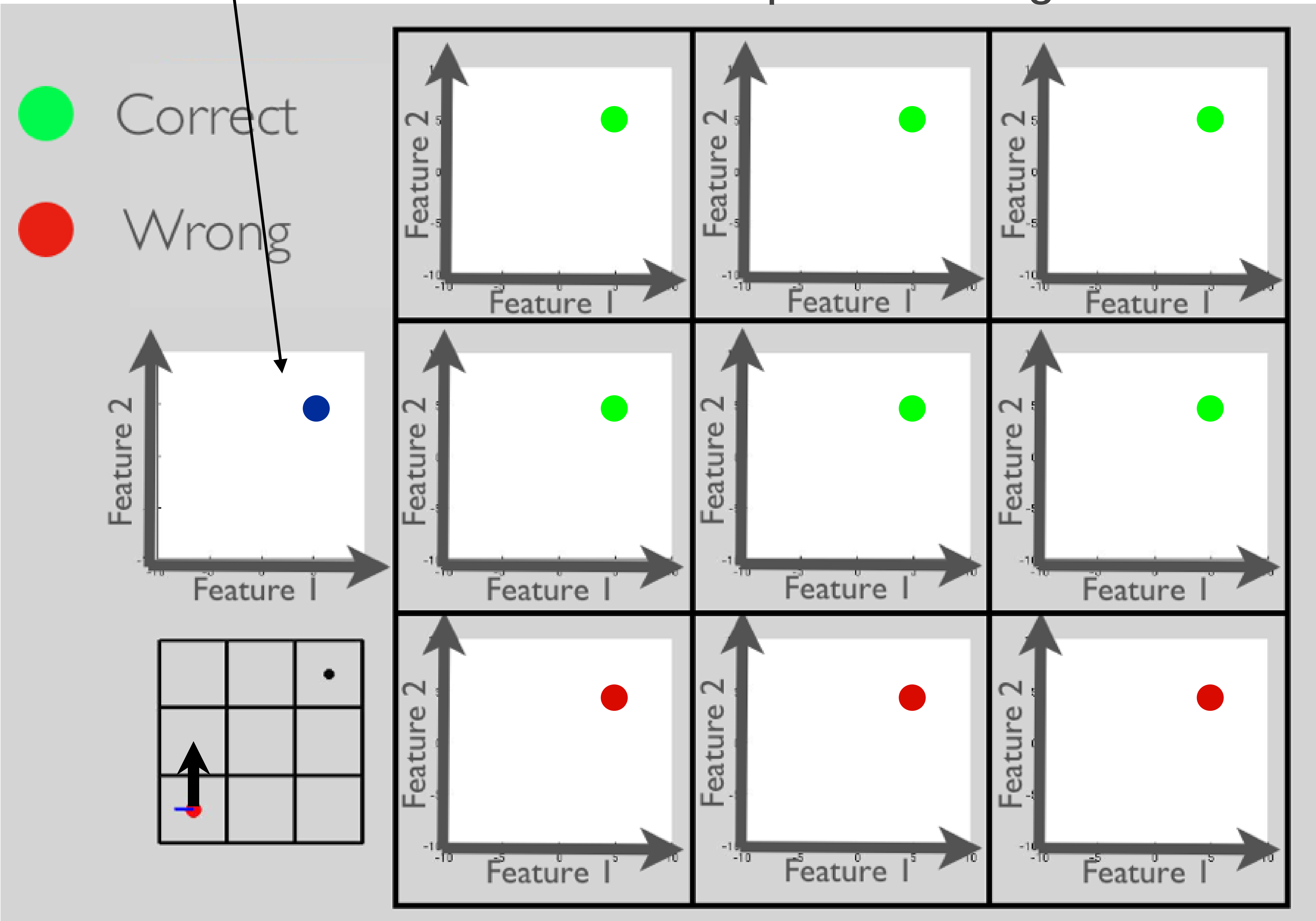
Interpretation (colors)
for each possible target

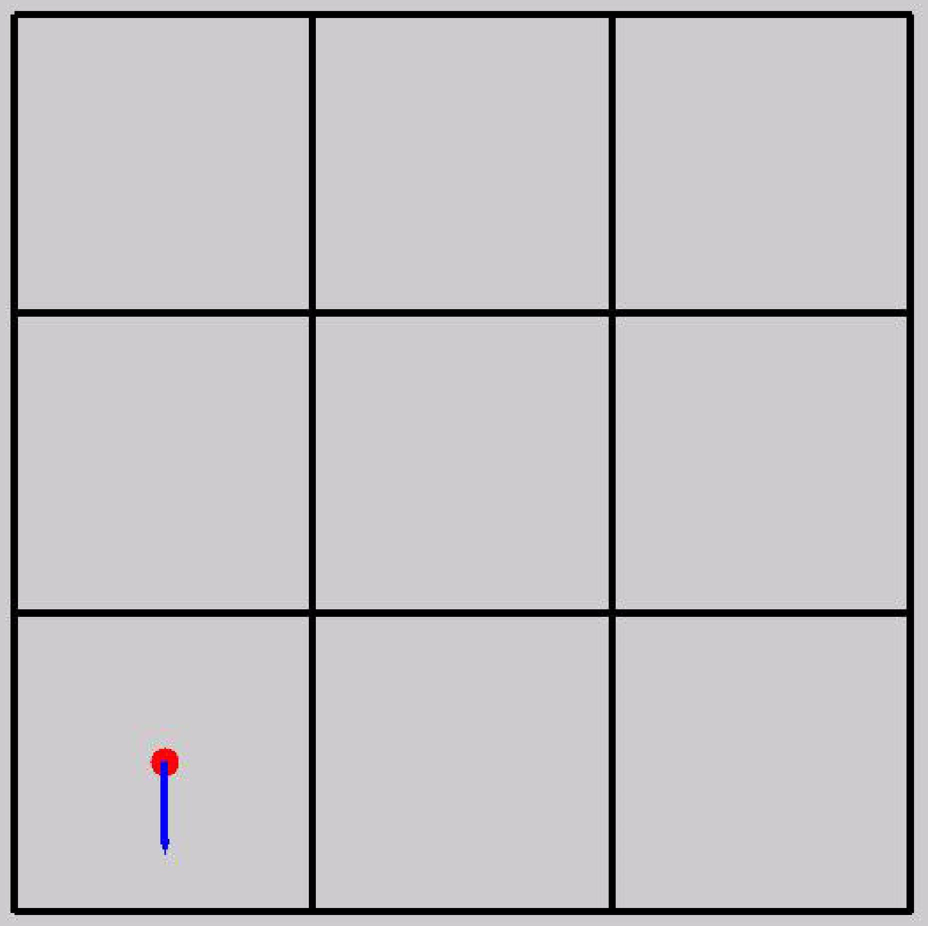
Observations

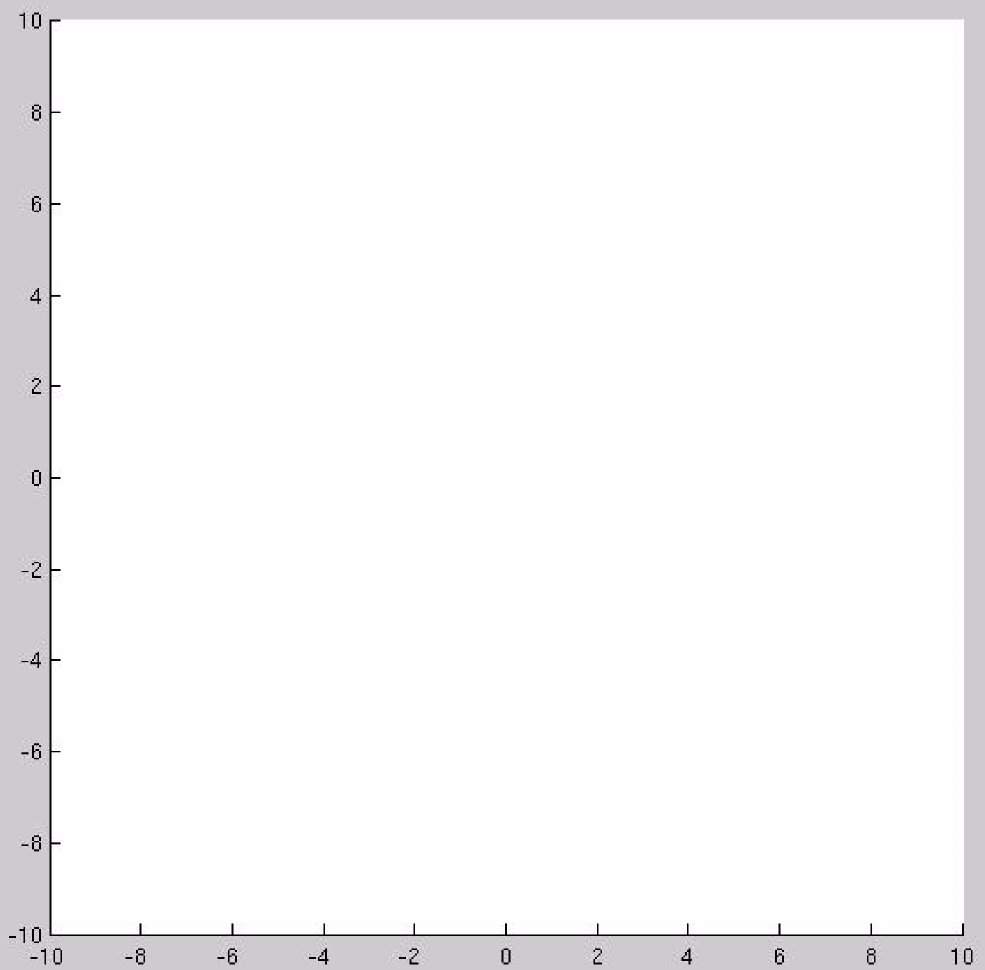


Observations

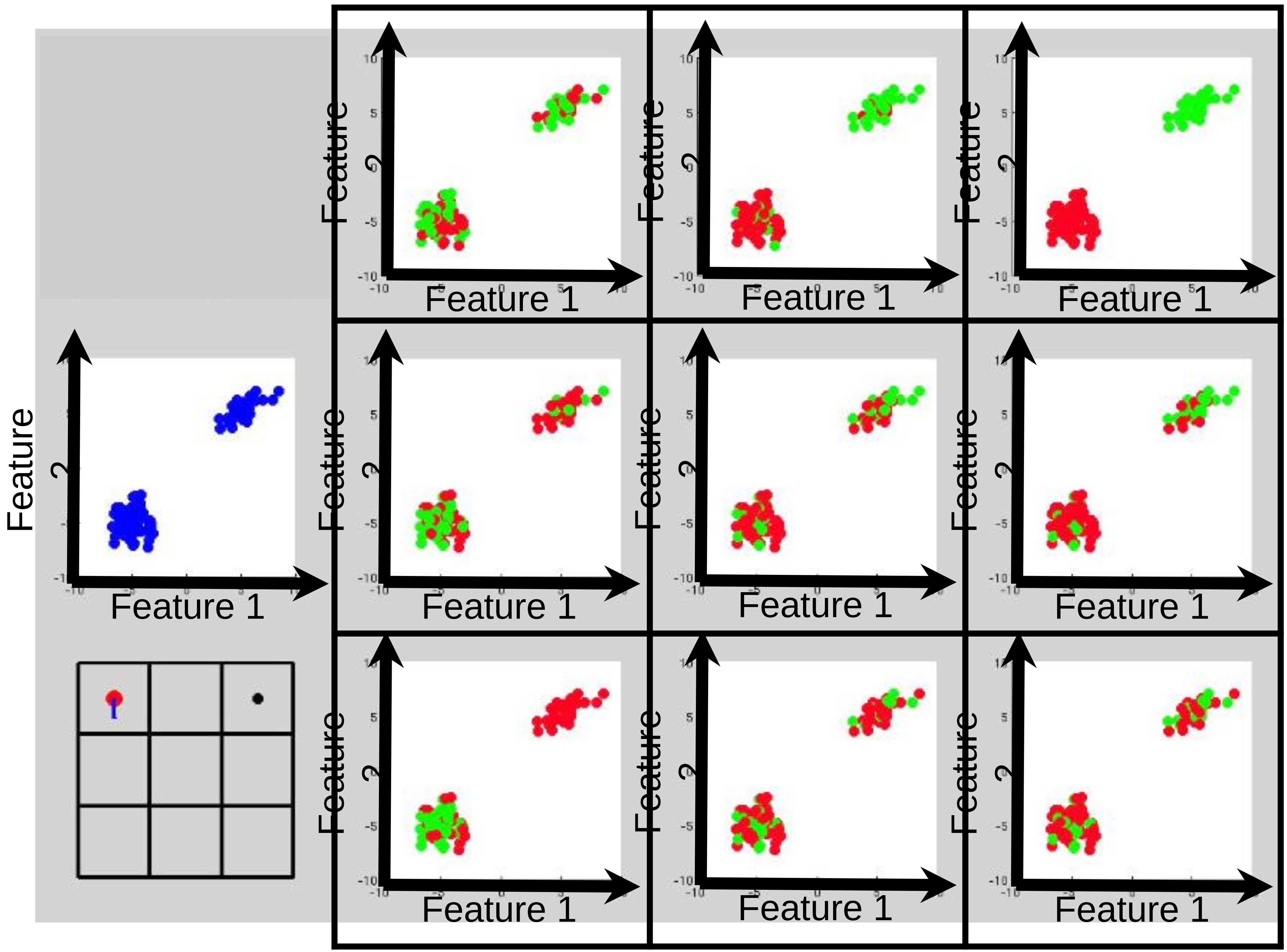
Interpretation (colors)
for each possible target







10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10


Feature 2
Feature 1

Simultaneously Execution and Calibration

Algorithm:

1. Set of possible tasks, ξ_k
2. Execute action a
3. Read signal s
4. For each ξ_k
   1. Compute expected classification $l(s, \xi_k)$
   2. Add to dataset D_k
   3. Fit classifier to D_k
   4. Compute likelihood(ξ_k)
5. Goto 2

Algorithm

Algorithm 1 EM for learning Signals

Require: Data $\{(x_i, a_i, n_i),\ i = 1, \dots, m\}$
Require: Task ξ

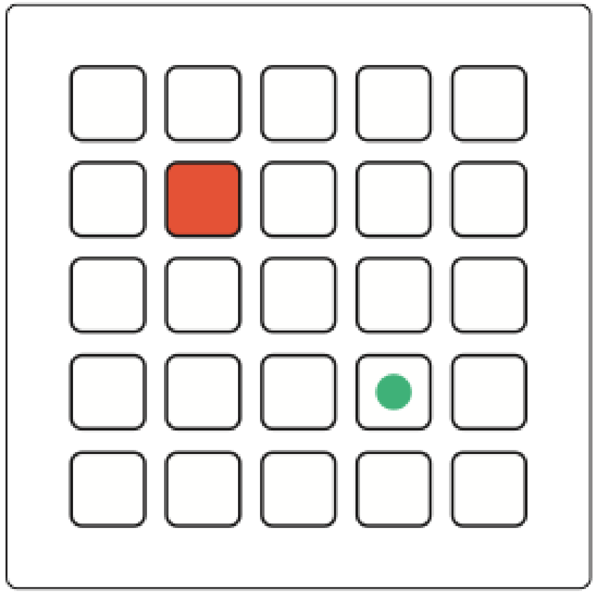
while true do
    E-Step
    $F(\theta|\theta^t) = \sum_{ij} \left(\log p(n_i|z_i, \theta) + \log z_i^{\xi} \right) w_{ij}$
    $w_{ij} = p(n_i|z_i, \theta^t) p(z_i|s_i, a_i, \xi)$
    M-Step
    $\theta^{t+1} = \arg\max_{\theta,\xi} F(\theta|\theta^t)$
end while

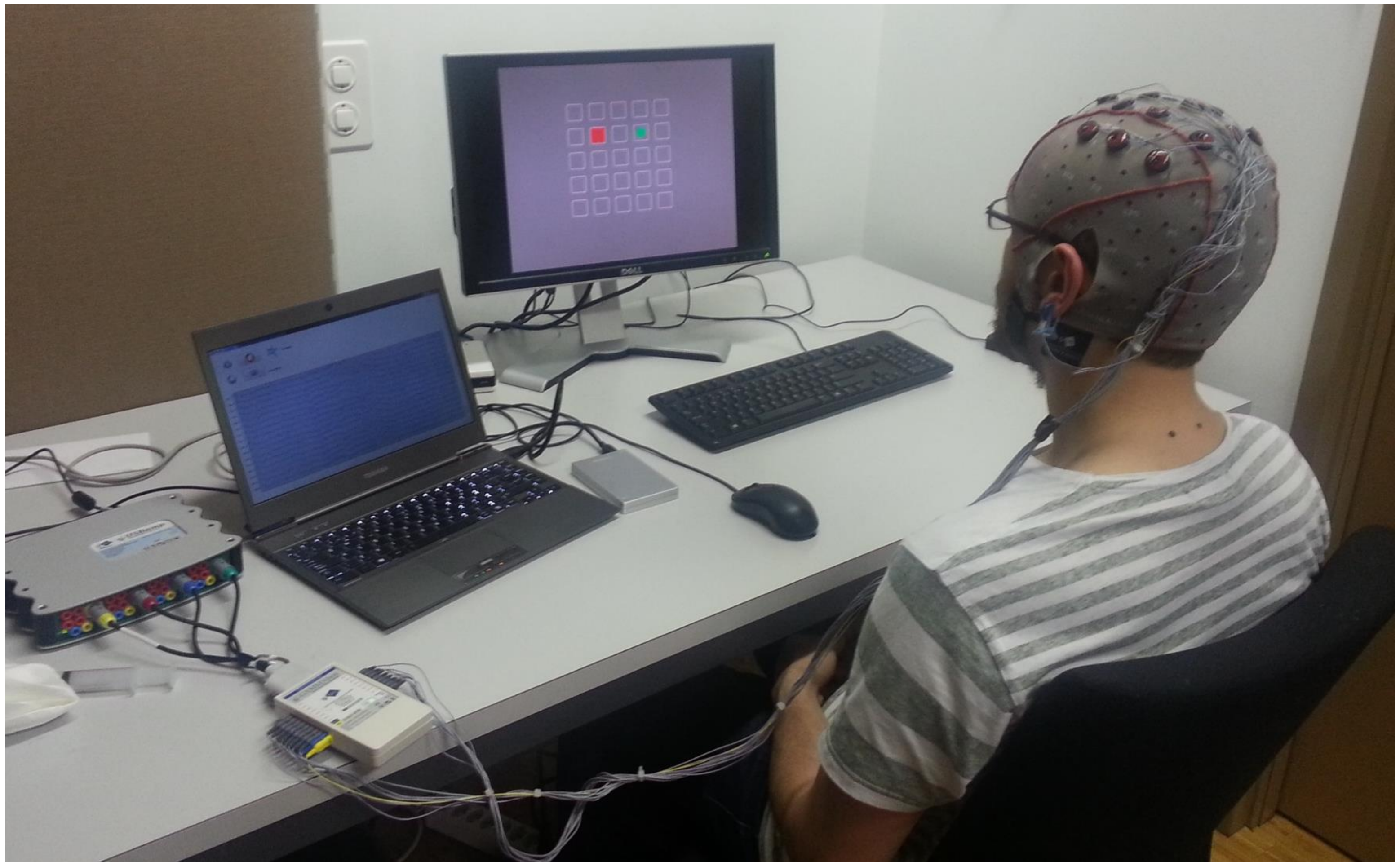
Algorithm 2 Learning Simultaneously Tasks and Feedback Signals

Require: Set of m possible actions A
Require: Set of n possible states X

Sample l different tasks $\xi_1, \dots, \xi_l$
$x_1 \leftarrow x_0$
$i = 1$
while true do
    Choose and apply action a_i
    Observe next state y_i and user feedback n_i
    for all $k = 1, \dots, l$ **do**
        From Algorithm 1 find:
        $\theta_k = \arg\max_{\theta} F(\theta|\theta^0, \xi_k)$
        $q_k(\xi_k) = \mathcal{L}(\theta_k)$
    end for
    Resample $\xi_k, k = 1, \dots, l$ according to $q_k(\xi_k)$
    $x_{i+1} \leftarrow y_i$
    $i \leftarrow i + 1$
end while
return $q_k(\xi_k), \xi_k\ ,\ k = 1, \dots, l$

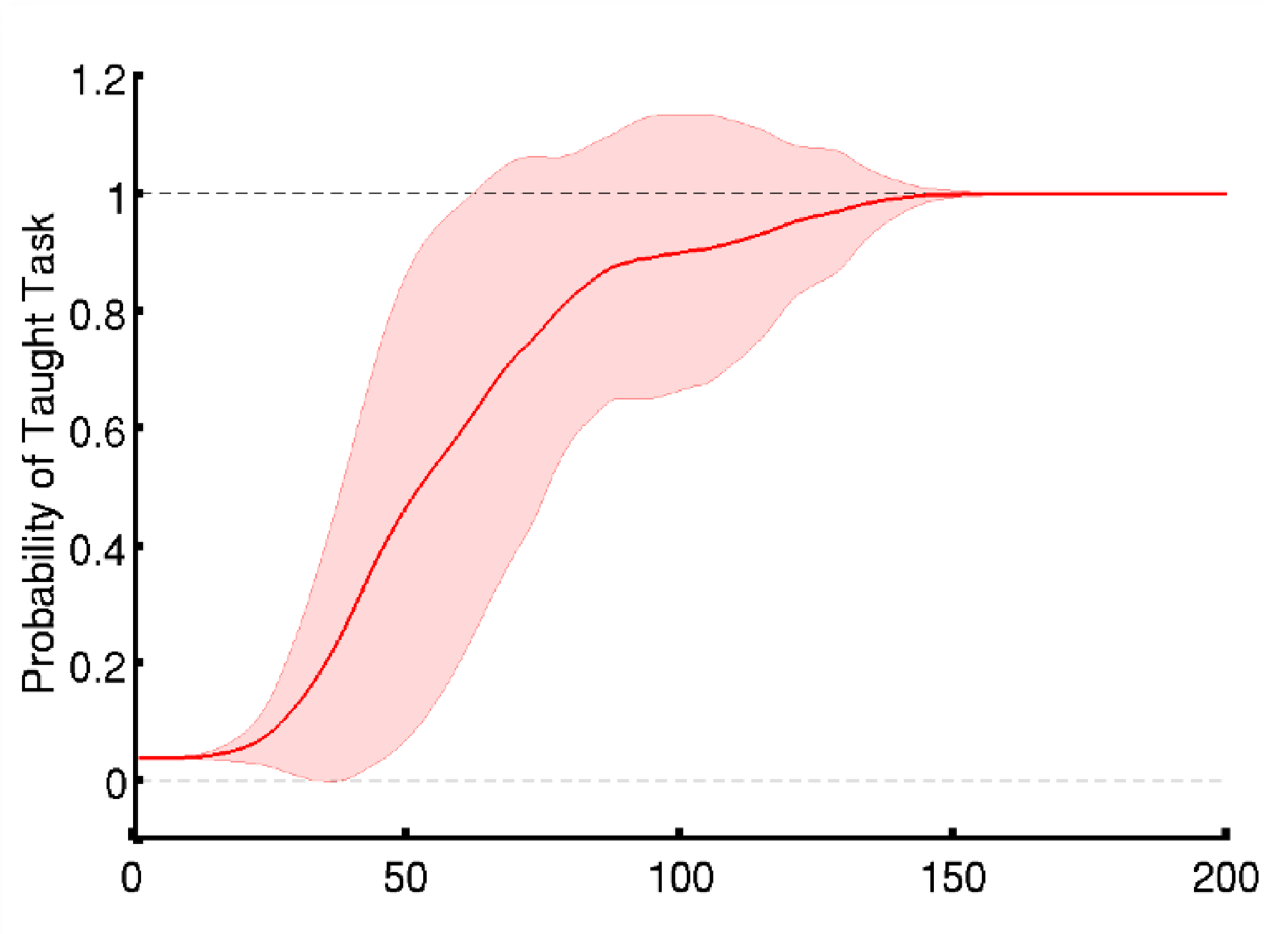
Experimental setup

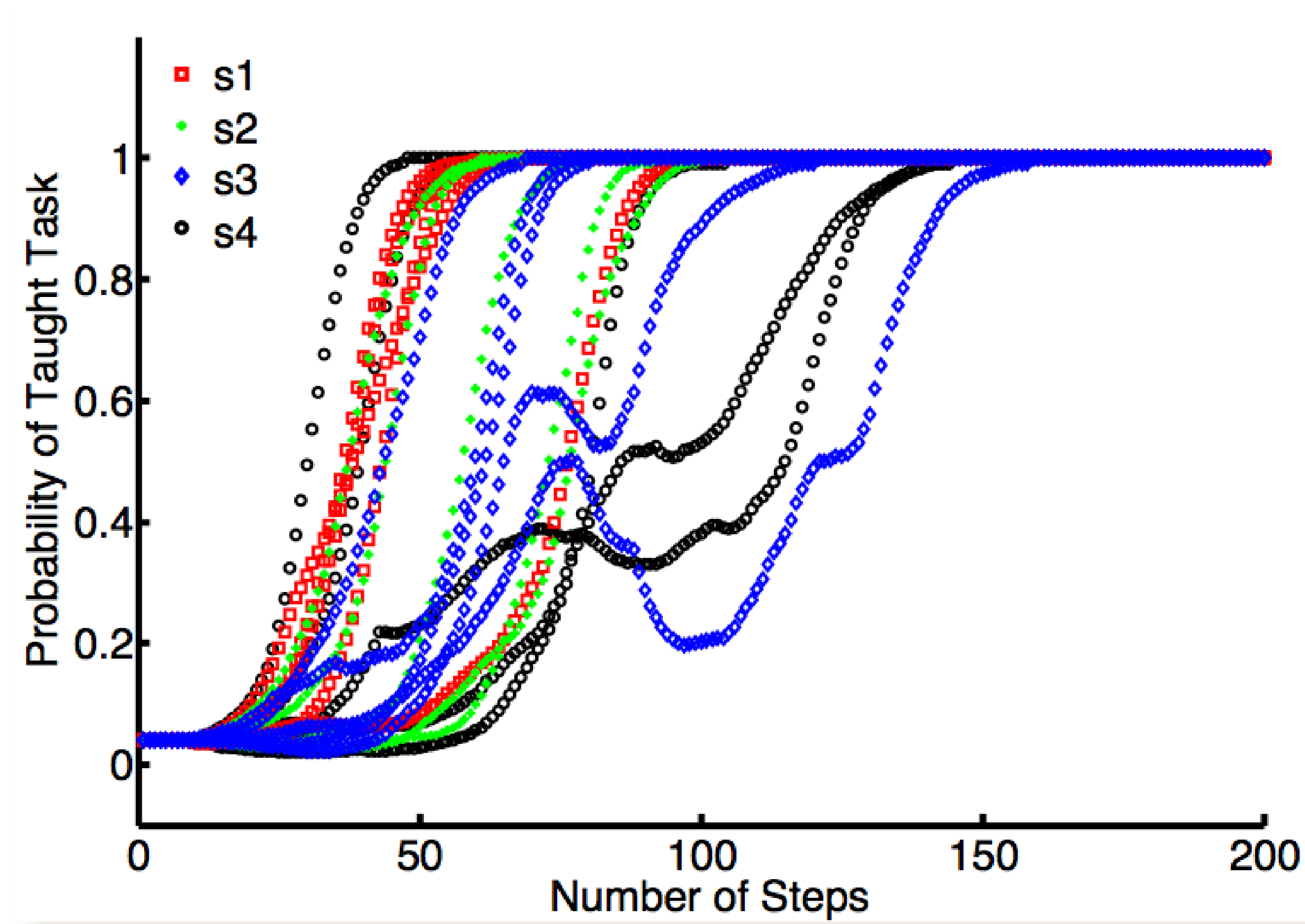


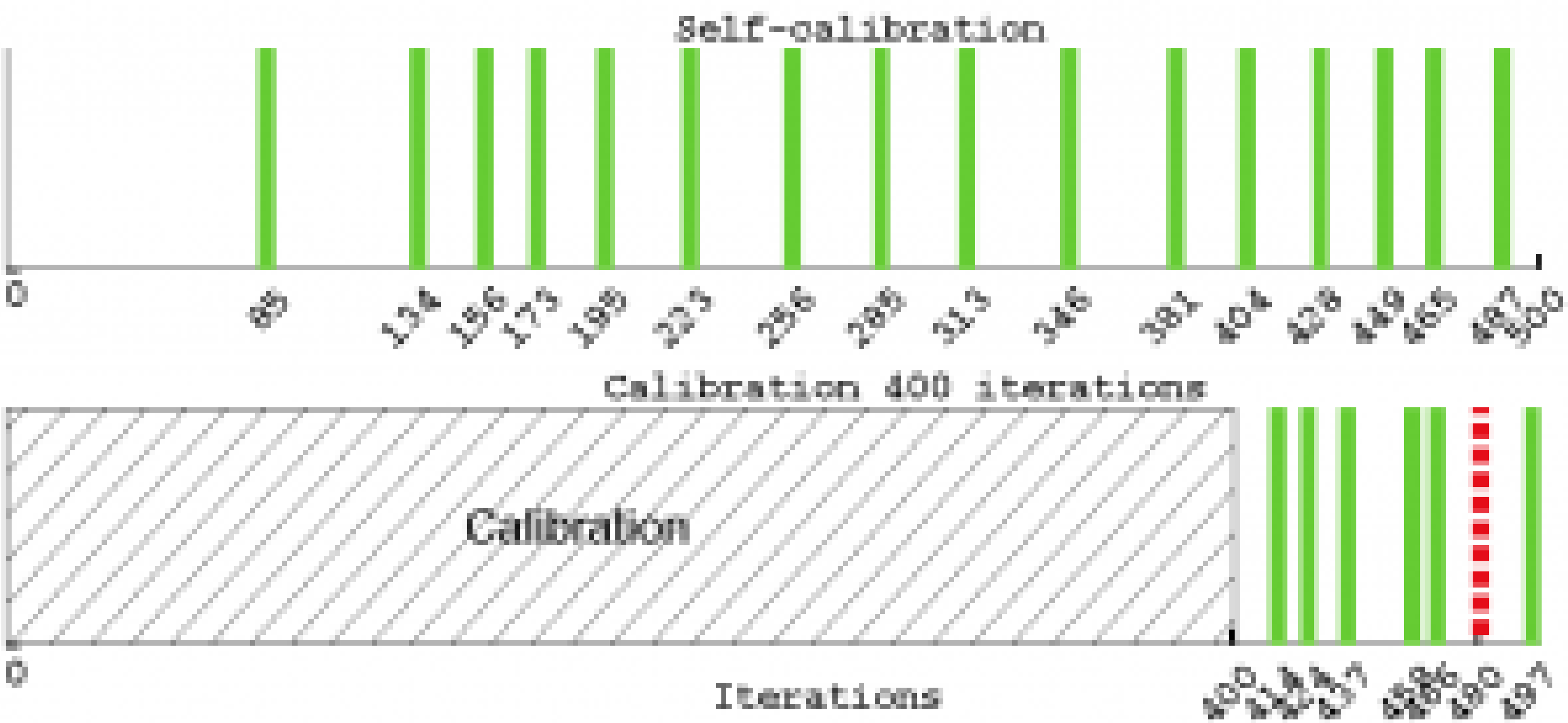


- 34 features, high amount of noise
- 25 possible targets (5x5 grid world)

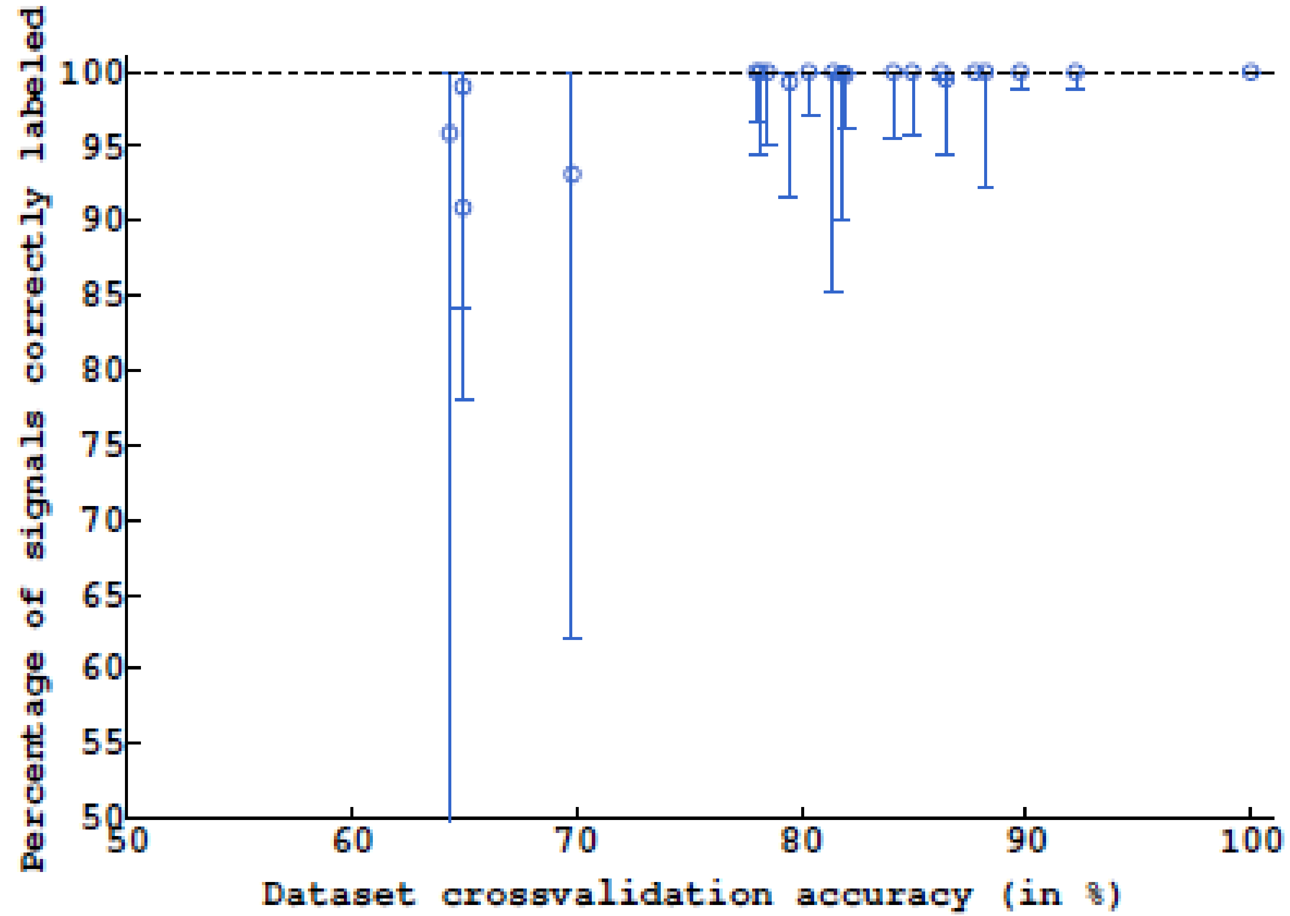
User Studies

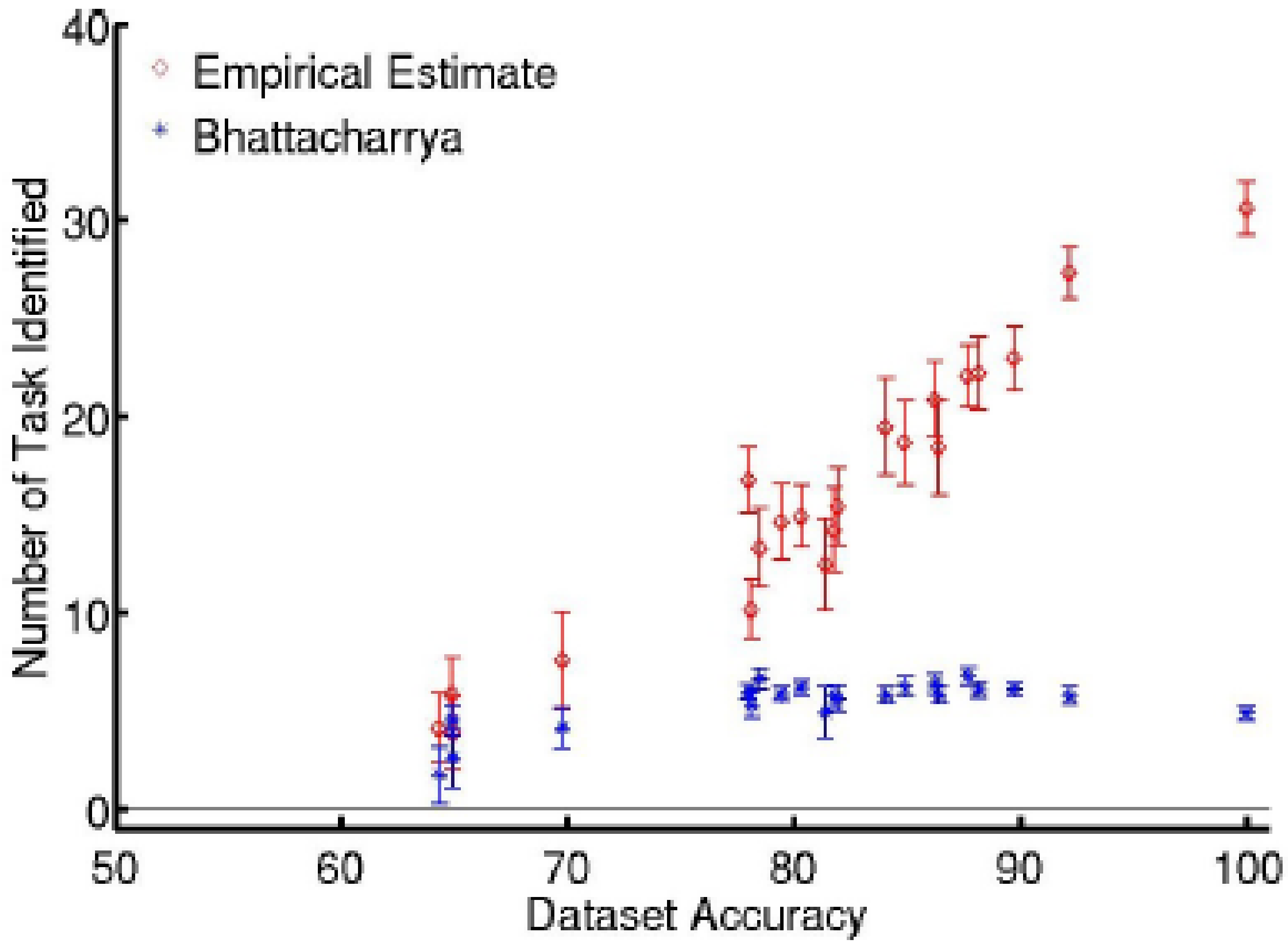


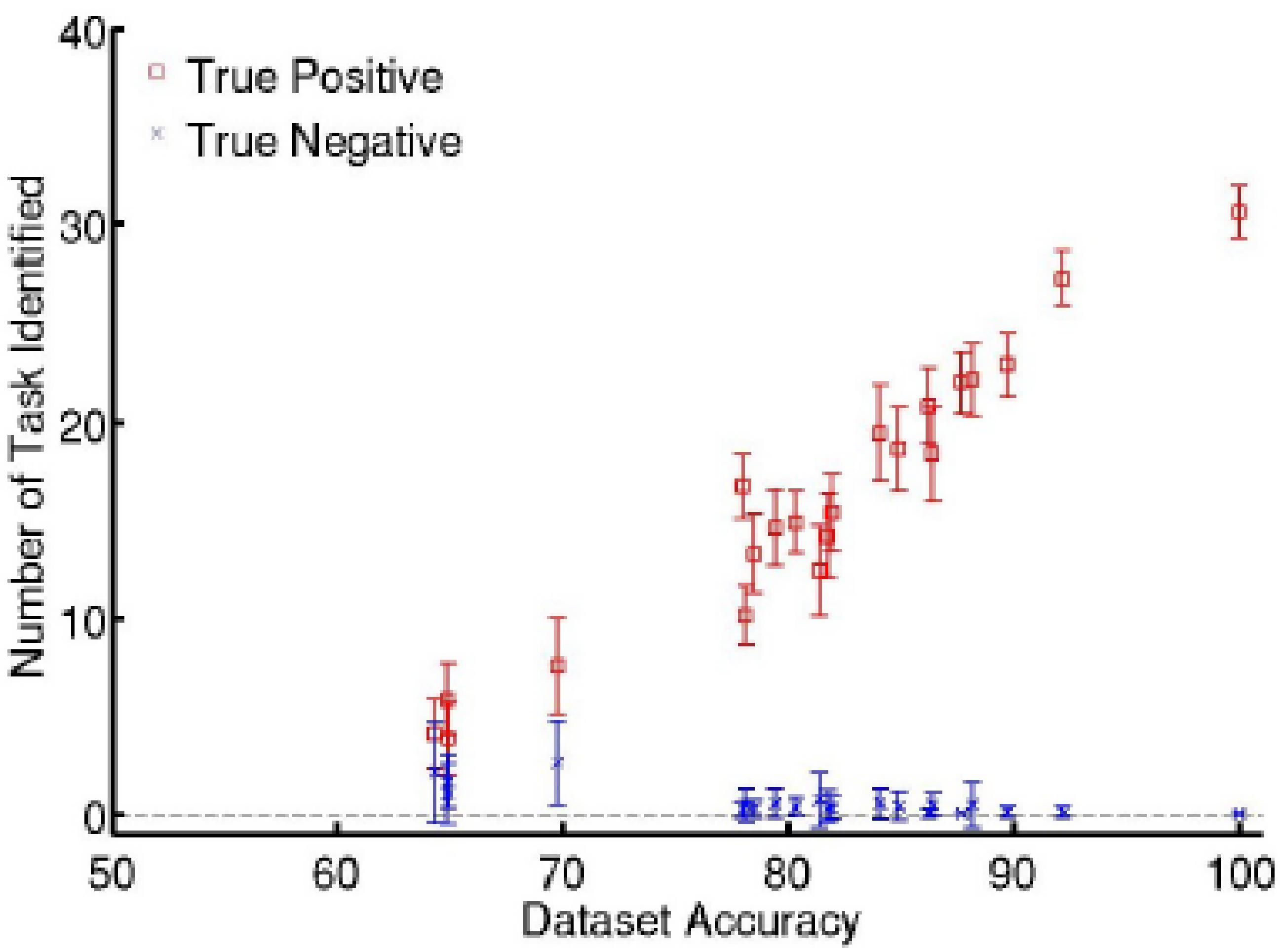




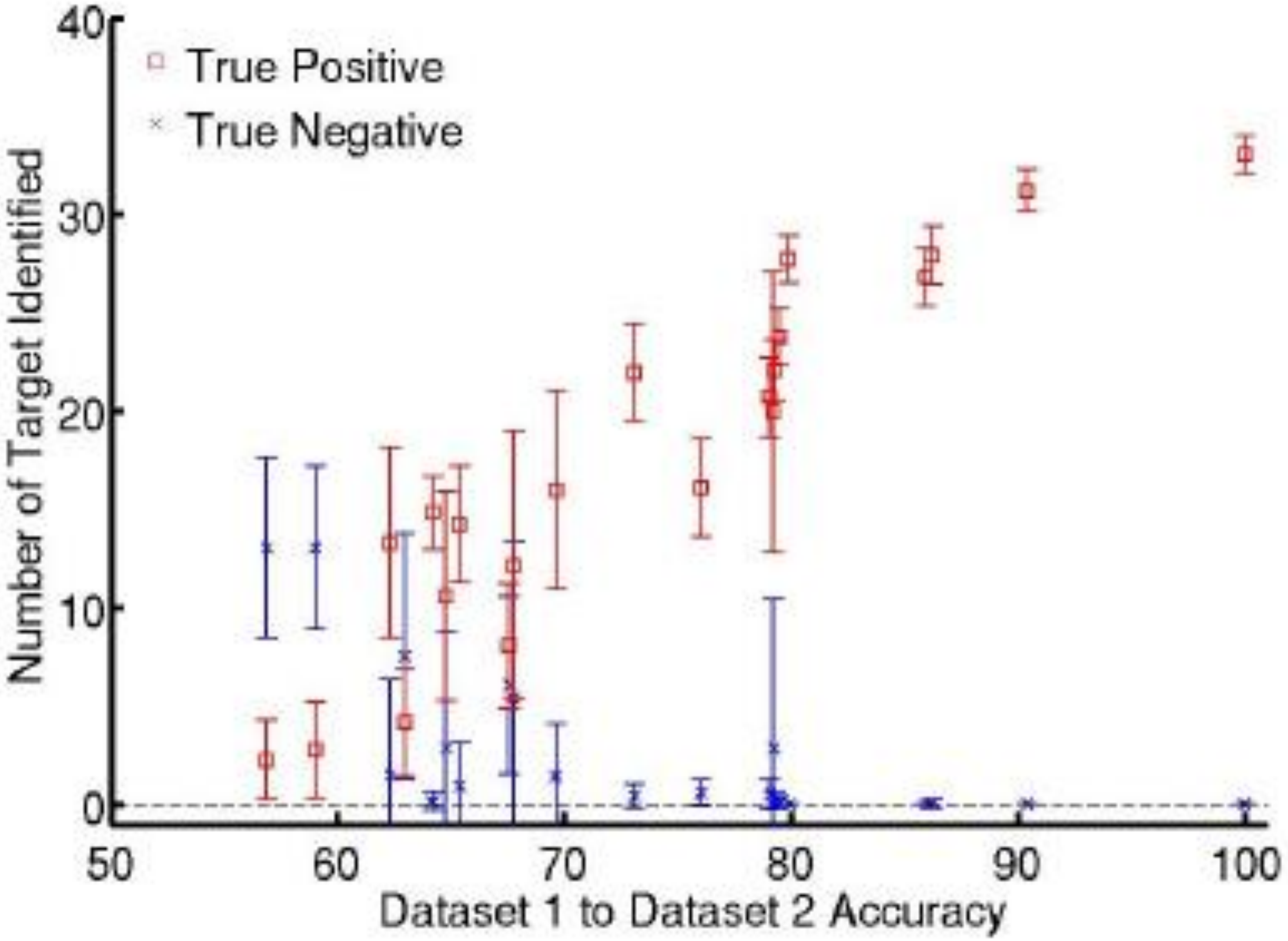
How does it compare in relation to the maximum expected calibration?







Transfer of information between two related tasks



How to choose the actions?

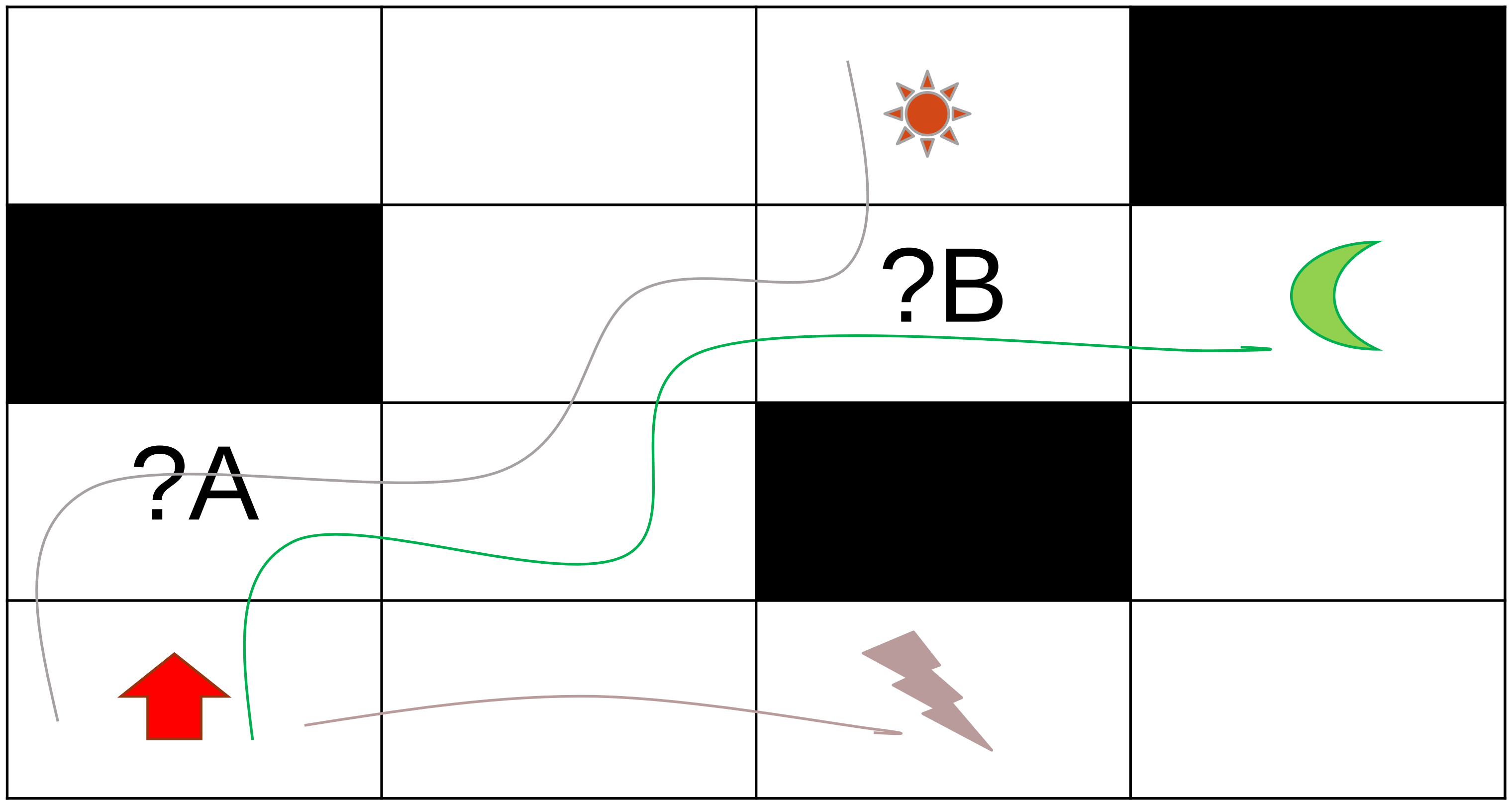
Up-to-now agents actions are random

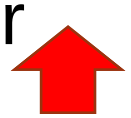
- Not time efficient
- Produces too-many errors, brain might start considering errors as the expected behavior

Active learning

- Agent decides actions that minimize uncertainty

Reducing the uncertainty. How to choose actions?



- Possible goals: 
- After  what are the possible goals?
- What should you ask the teacher? The correct action in A or in B?

Reducing the uncertainty. How to choose actions?

- Planning can consider the uncertainty in the meaning(classifier), in the task, or in the expected signals.

Likelihood

$$P(D_n|\xi,\theta) \approx \prod_{i=1}^{..} p(e_i|\xi,\theta_{-i},s_i,a_i) \quad (4)$$

$$= \prod_{i=0}^{N} \sum_{l_c} \sum_{l} p(e_i|\theta_{-i},l_c)p(l_c|\theta_{-i},l)p(l|\xi,s_i,a_i) \quad (5)$$

Uncertainty of (s,a)

$$U(s,a|e) = weightedVariance(J^{\xi}(s,a,e),W^{\xi})$$

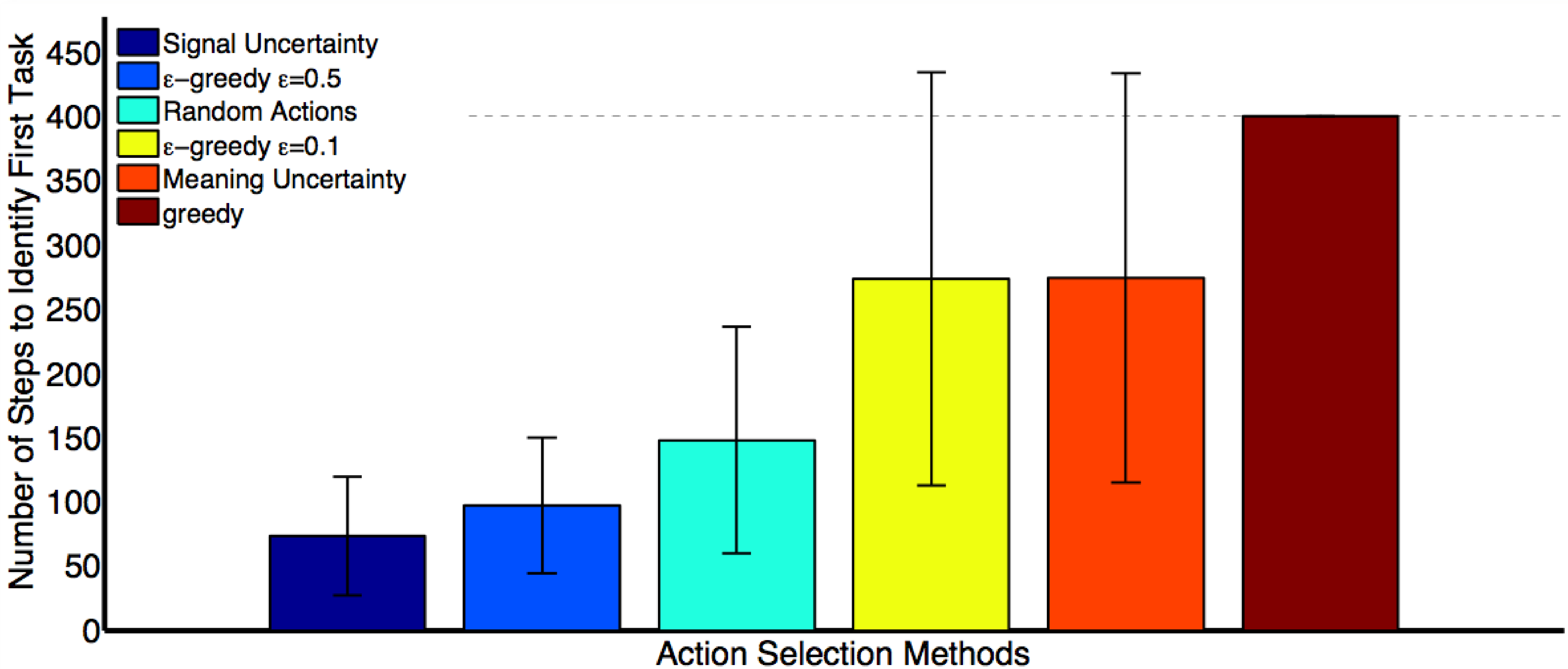
Uncertainty

$$U(s,a) \approx \sum_{e} U(s,a|e)p(e)$$

Reducing the uncertainty. How to choose actions?

- The uncertainty $U(s,a) \approx \sum_e U(s,a|e)p(e)$ is used as an exploration bonus.
- The agent moves to maximize the expected long term cumulative sum of U(s,a)

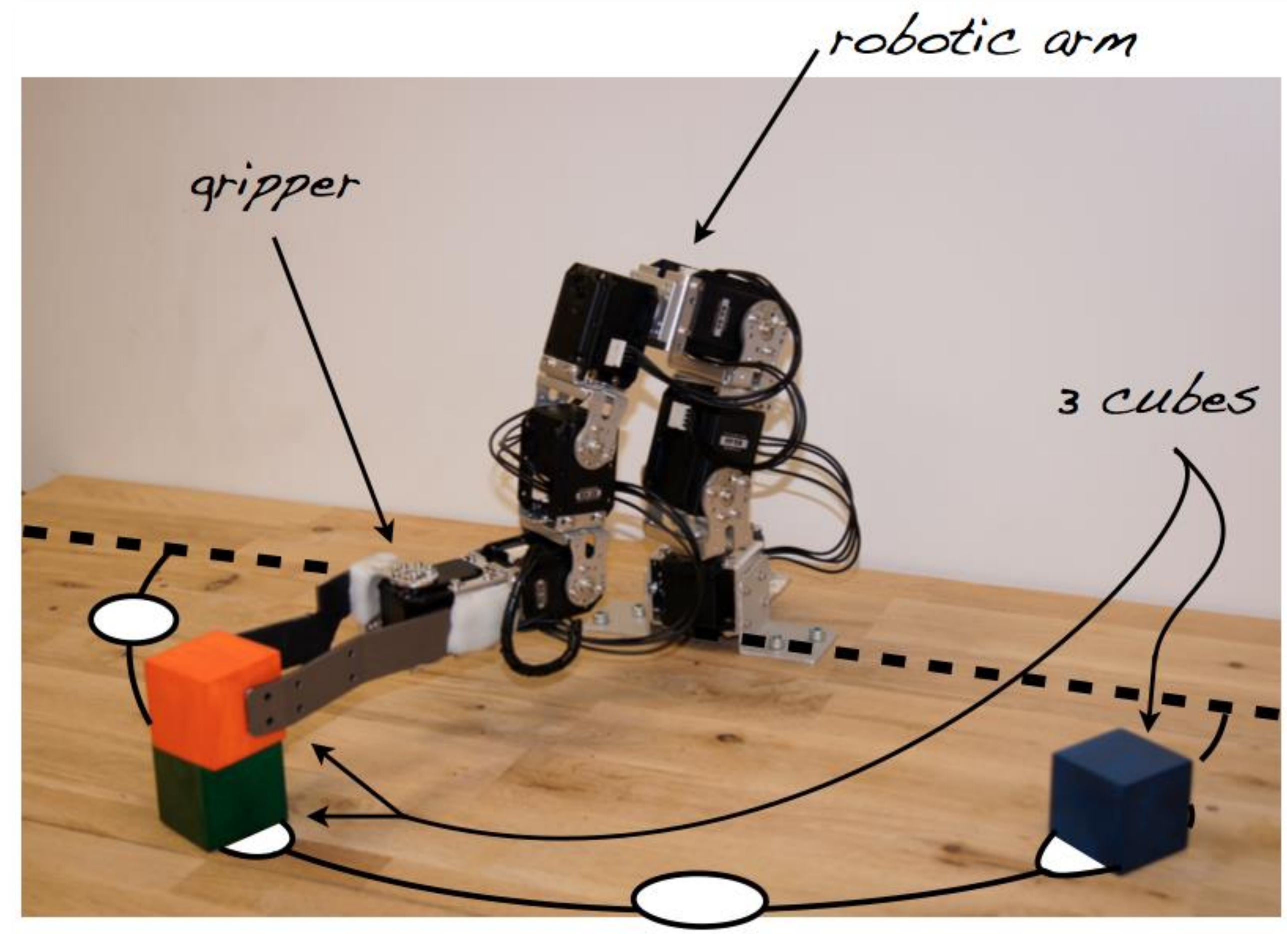
Planning



Planning can consider the uncertainty in the meaning(classifier), in the task, or in the expected signals.

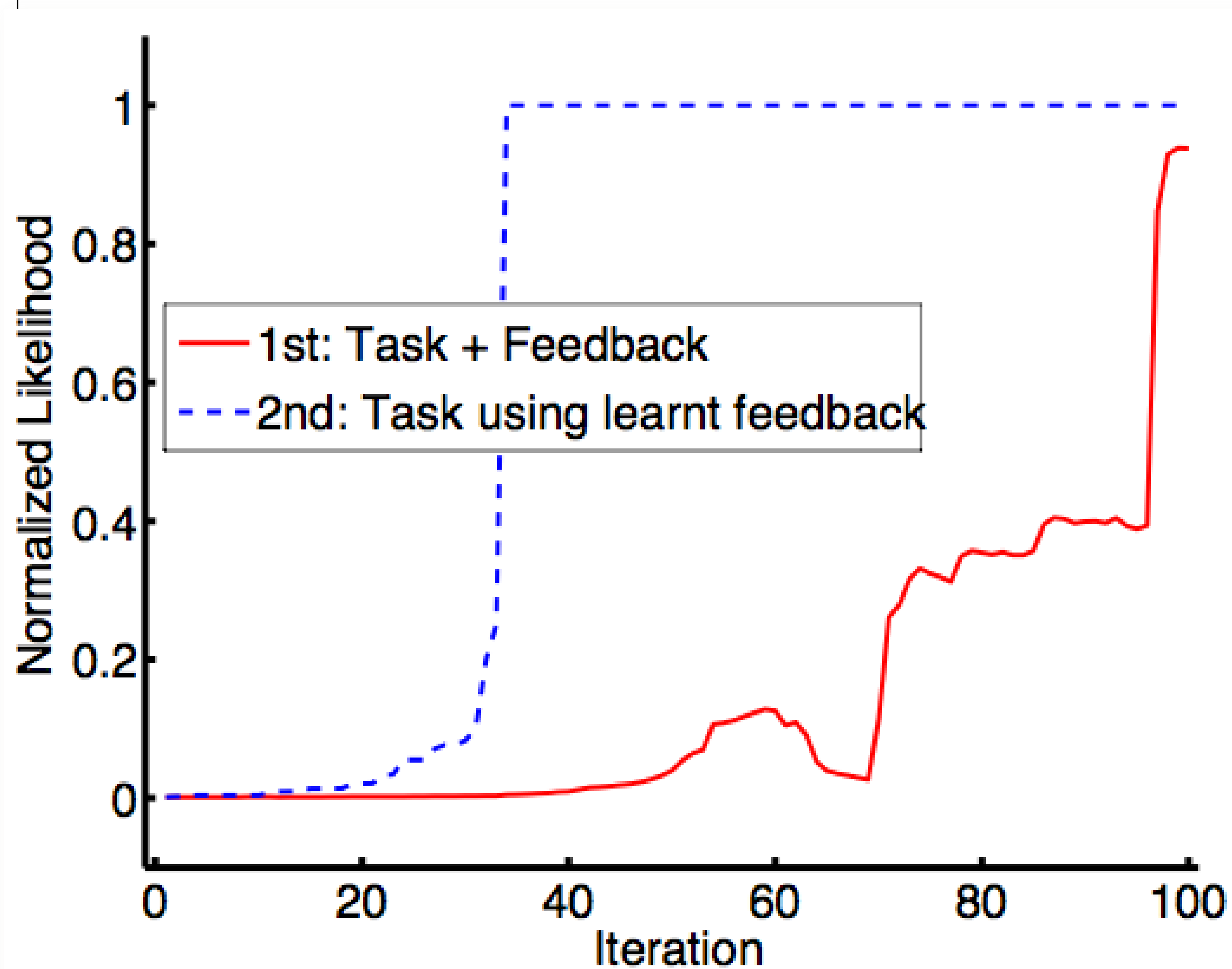
MDP : 624 states, 4 actions
(left,right,grasp,release)

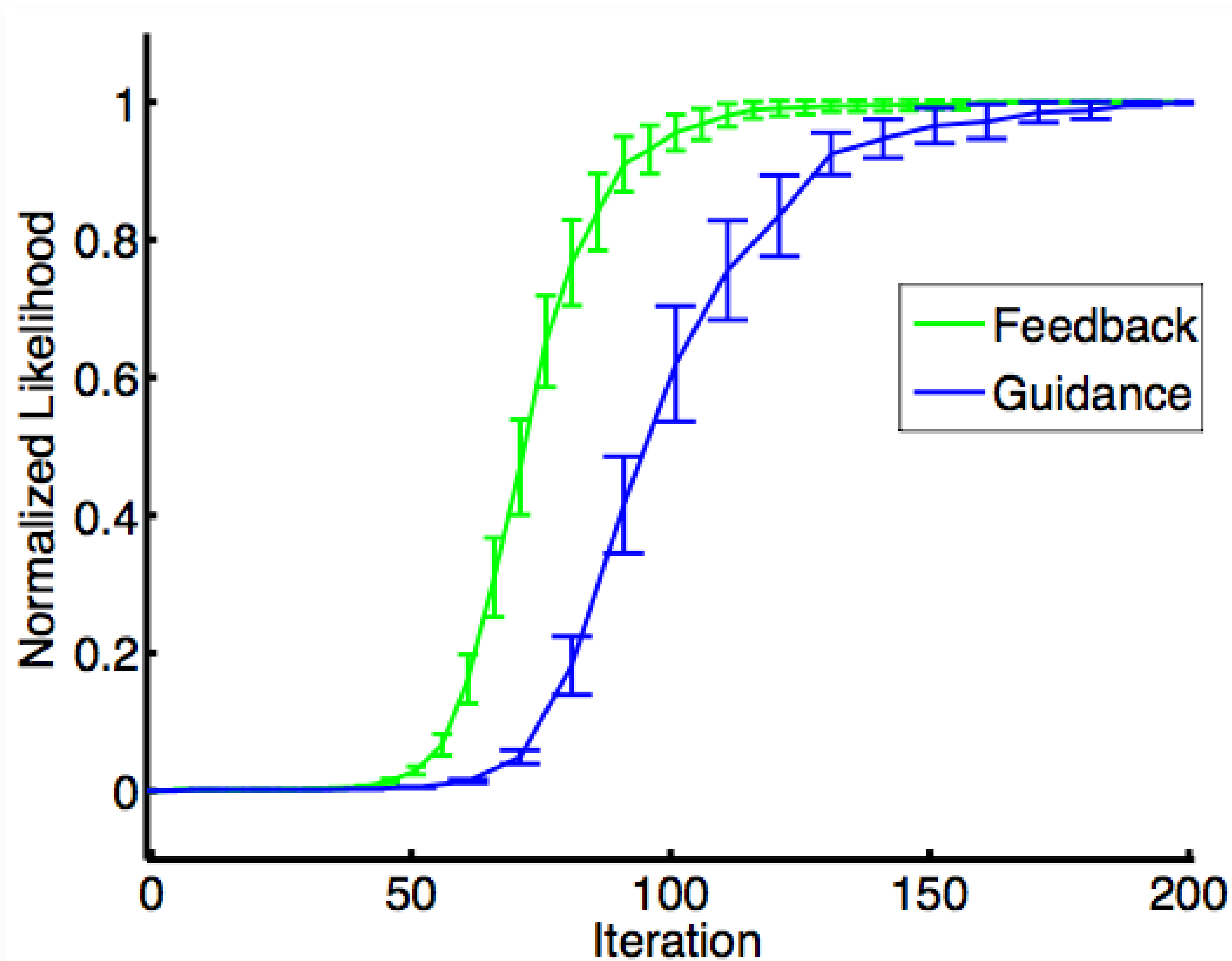
Task hypothesis :
Reach one of the 624 possible configuration



Feedback signals :
Spoken words mapped to a 20 dimensional feature space

Noise :
1- Words never spoken the same way
2- Teachers make mistakes





Using ε-greedy with ε = 0.1

Conclusions

- Yes, it is possible to cold-start a BCI system and simultaneously calibrate and control the system
- For the equivalent calibration time, the system executes the task several times, and achieves a similar calibration rate

Future work

- Reduce the synchronous aspect of the protocol