IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Lab-STICC

INTERACTIVE MACHINE LEARNING

LEARNING THROUGH INTERACTIONS WITH TUTORS AND THE ENVIRONMENT:

IMITATION AND REINFORCEMENT LEARNING

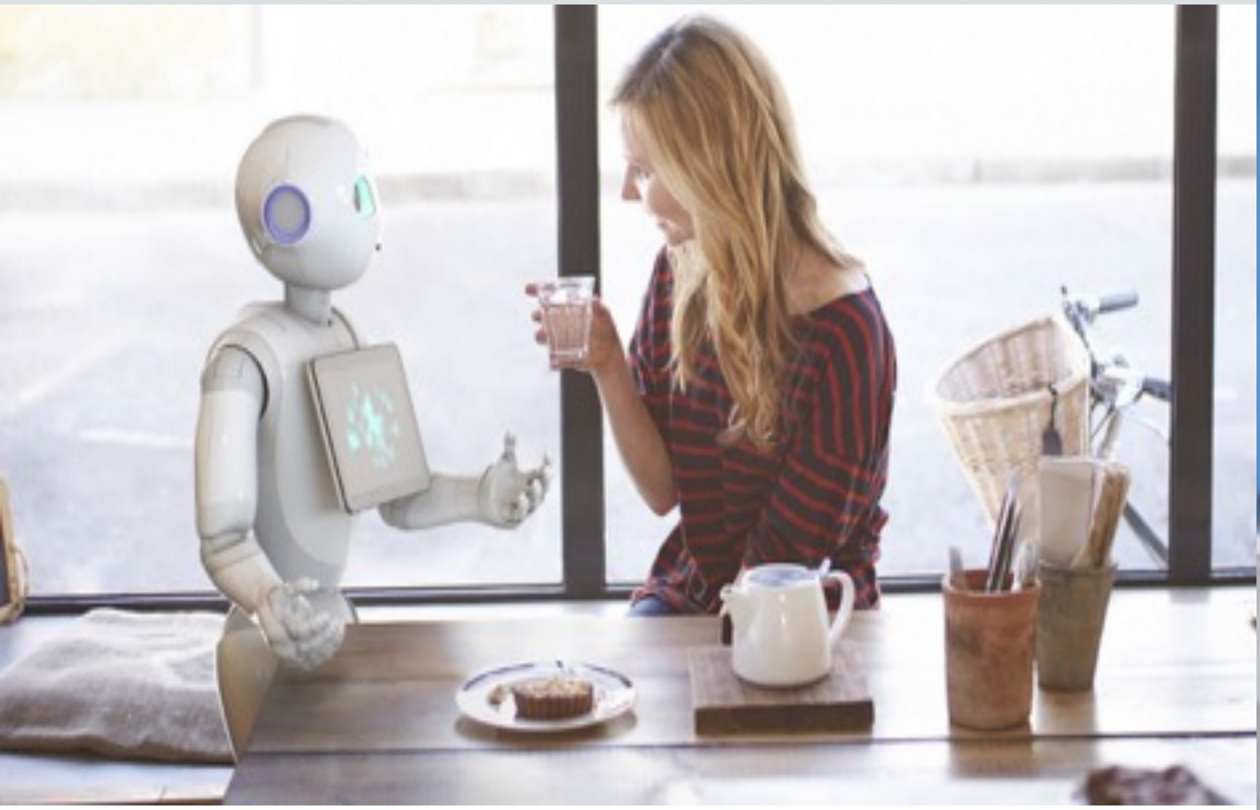
Nguyen Sao Mai
http://nguyensmai.free.fr

1. WHAT DOES INTERACTIVE LEARNING MEAN?

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

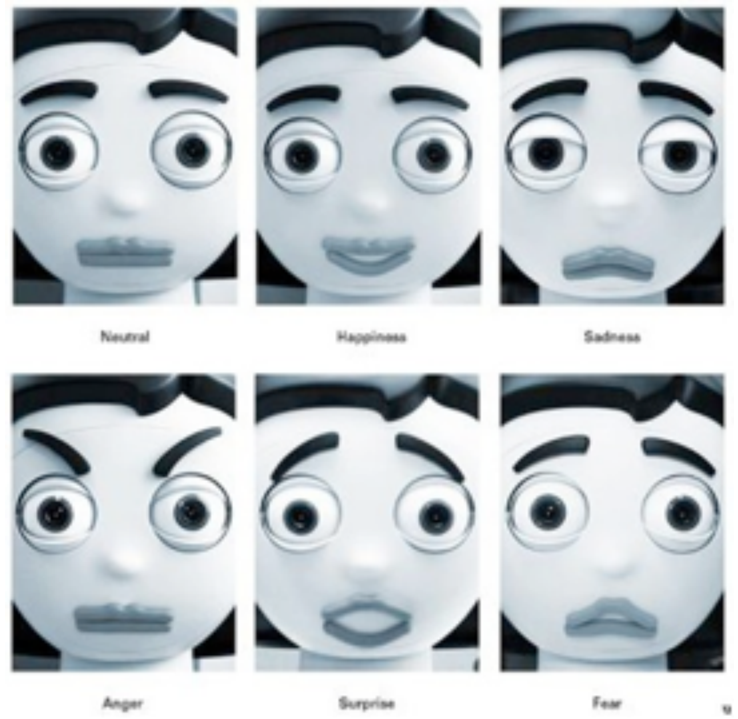
1. WHAT DOES INTERACTIVE LEARNING MEAN?

1.1. The Artificial Agent in Its Environment



1. WHAT DOES INTERACTIVE LEARNING MEAN?
1.2. Challenges

- **Vocal interaction**: speech recognition, speech generation (text to-speech)
- **Natural interaction** : multi-modal, non-verbal interaction, gesture, expressive emotion-based interaction
- **Socio-cognitive skills** : socially acceptable behaviours, turn-taking, coordination, theory of mind
- **Physical interaction** : touch (tactile sensors), grasping, manipulation



1. WHAT DOES INTERACTIVE LEARNING MEAN?

1.3. Theoretical approaches

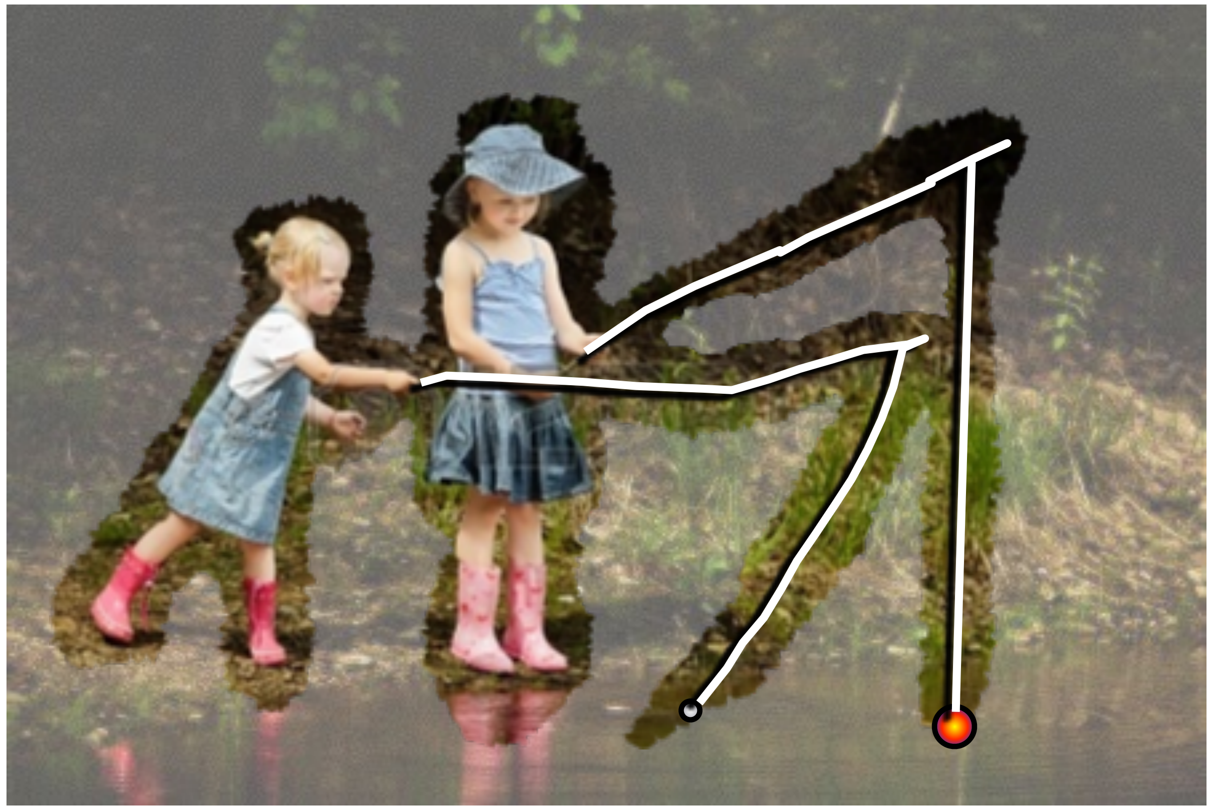
- **Embodiment** : the environment has a physical incarnation, the agent has a physical incarnation => its learning, capacities, behaviour depends on its physical body
- **Enactivism** : Learning of the agent in its environment
- **Life-long learning** : the environment and tasks can change
- **Developmental approaches** : there is an orderly way to learn multiple tasks, the learning is progressive and hierarchical -> Developmental psychology
- **Cognitive approaches** : inspired by cognitive science, neuroscience, neuronal computation models. Decomposes into a task into cognitive skills/ functions

2. INTERACTIONS WITH TUTORS:

IMITATION LEARNING OR PROGRAMMING BY DEMONSTRATION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

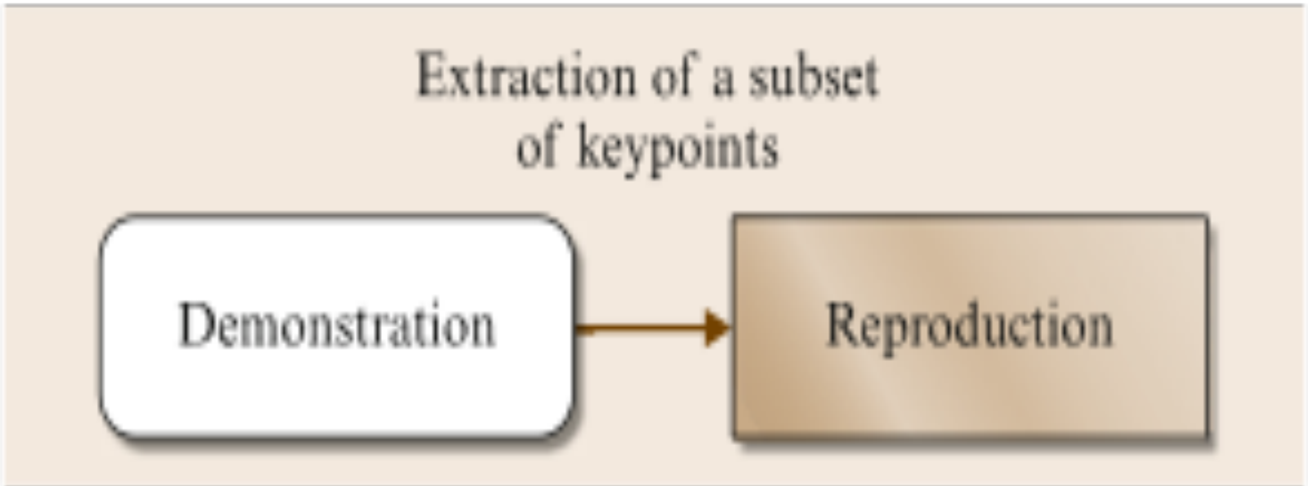
Mimicry : reproduce the movement



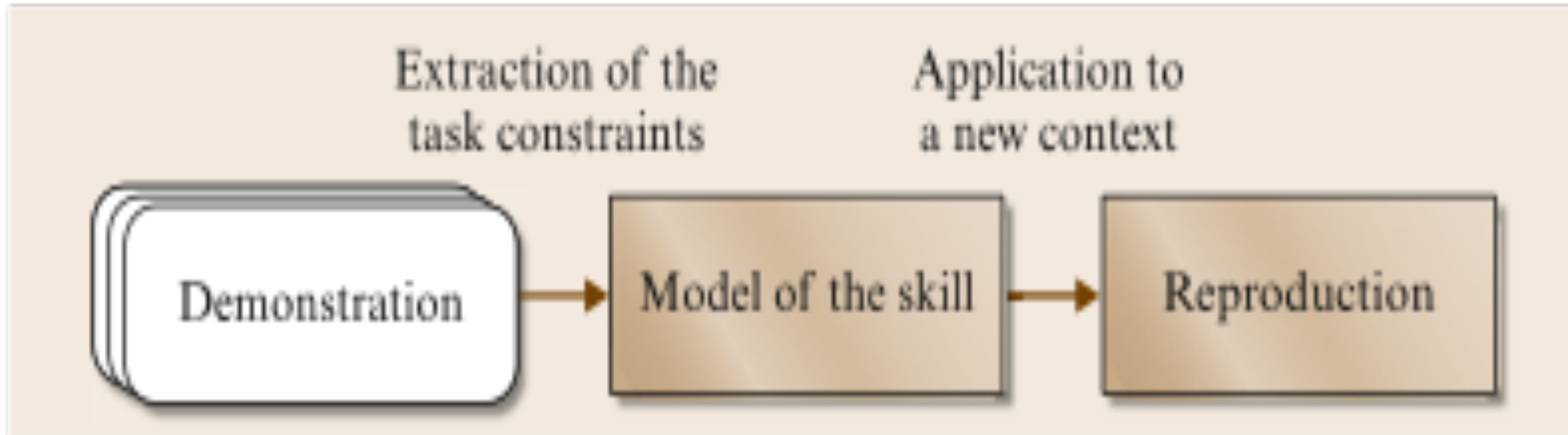
Emulation : reproduce the effects/outcomes

- An implicit, *natural* means of training a machine that would be **accessible to lay people**
- A powerful mechanism for **reducing the complexity of search** spaces for learning
- Studying and modeling the **coupling of perception and action**

Copying the demonstrated movements



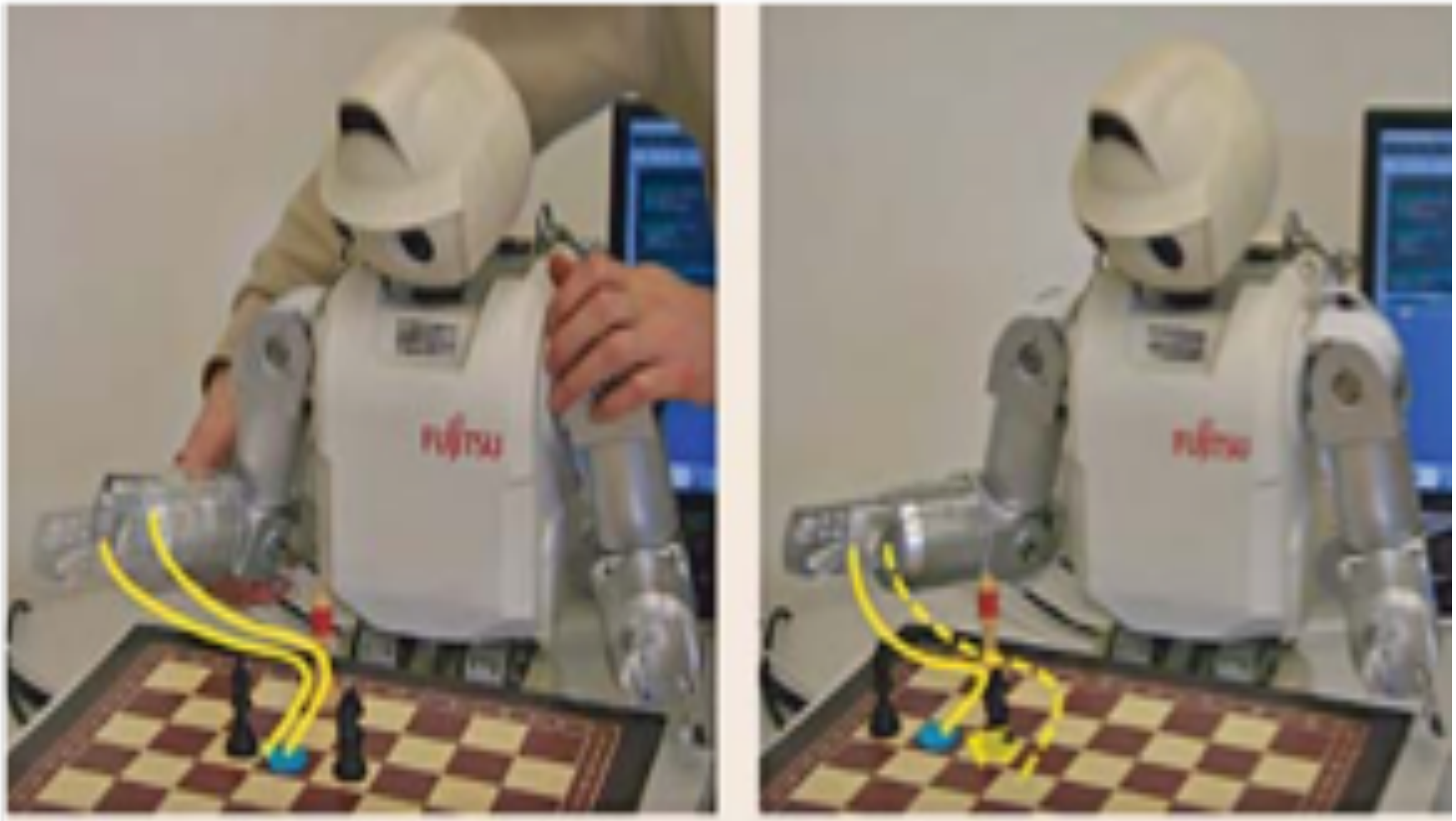
Generalize across sets of demonstrations.



- How to **generalize** a task
- How to **evaluate** a reproduction attempt
- How to better define the role of the **user** during learning

2. IMITATION LEARNING

2.2. Why imitation learning? What is imitation learning?



Observation of multiple demonstrations

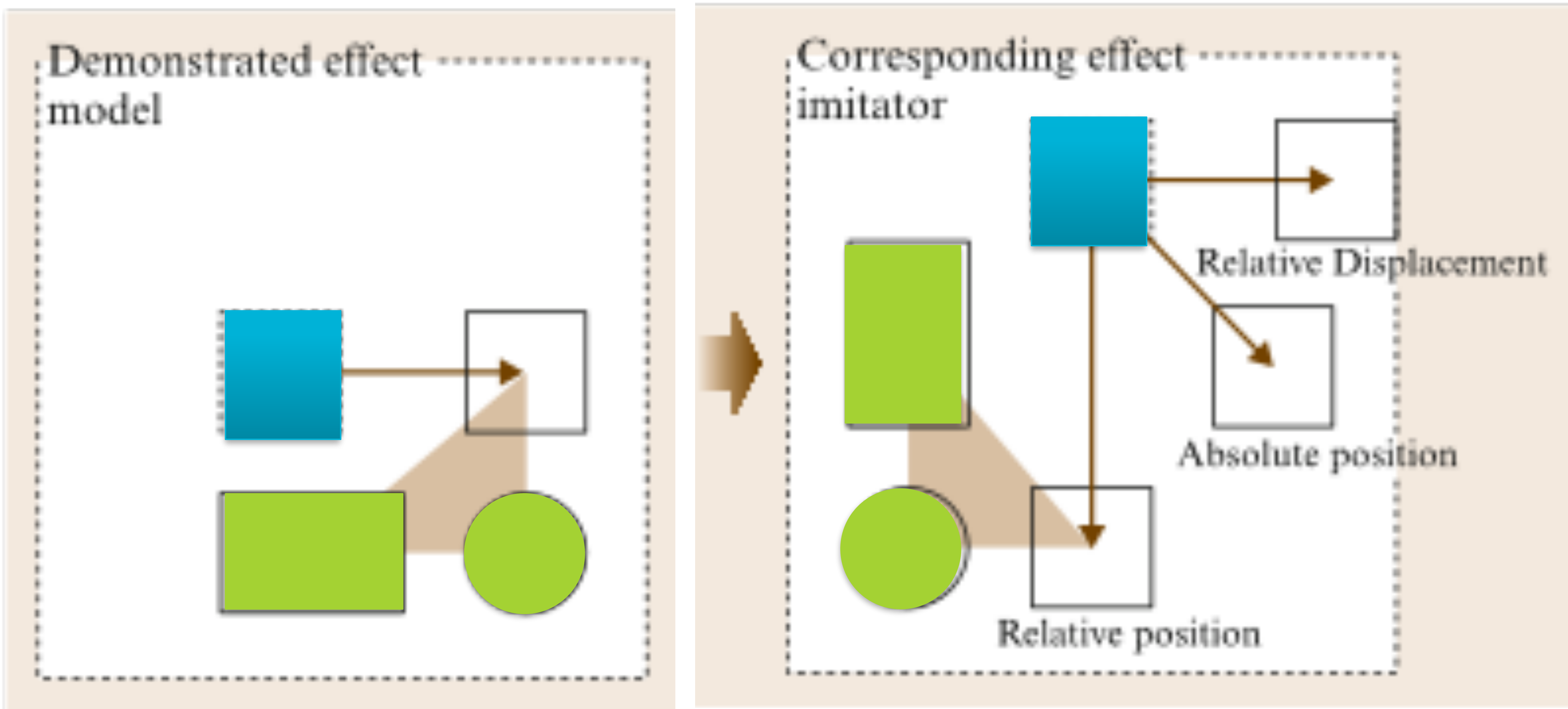
Reproduction of a generalized motion in a different situation

The different types of representation to encode a skill

- **a low-level representation** of the skill, taking the form of a non-linear mapping between sensory and motor information, which we will later refer to as *trajectories encoding*
- **high-level representation** of the skill that decomposes the skill in a sequence of action-perception units, which we will refer to as *symbolic encoding*

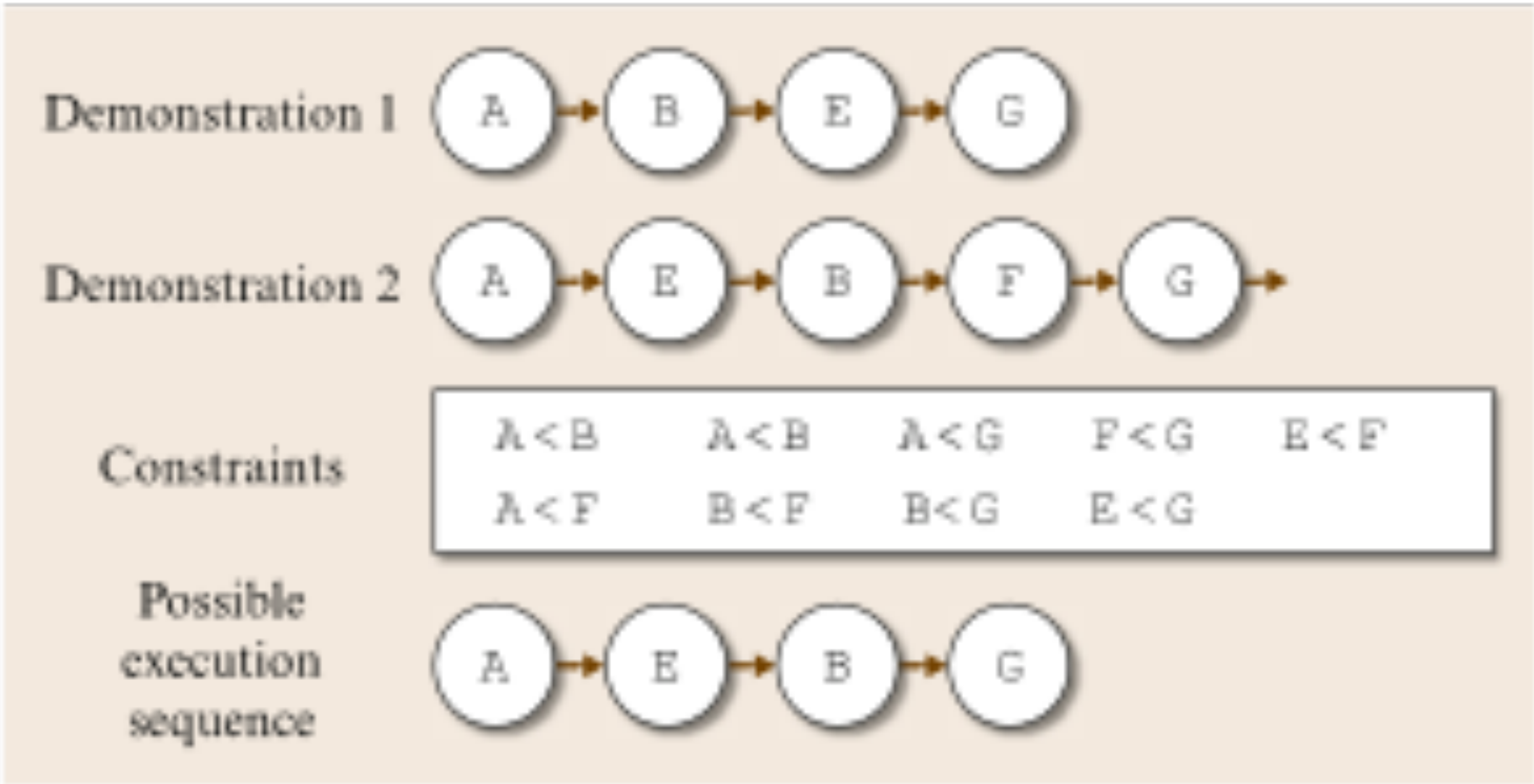
what to imitate, how to imitate, when to imitate and who to imitate : making no assumptions on the type of skills that may be transmitted

- **Metric of imitation performance:** extract the important features characterizing the skill
- An **optimal controller** to imitate by trying to **minimize this metric**



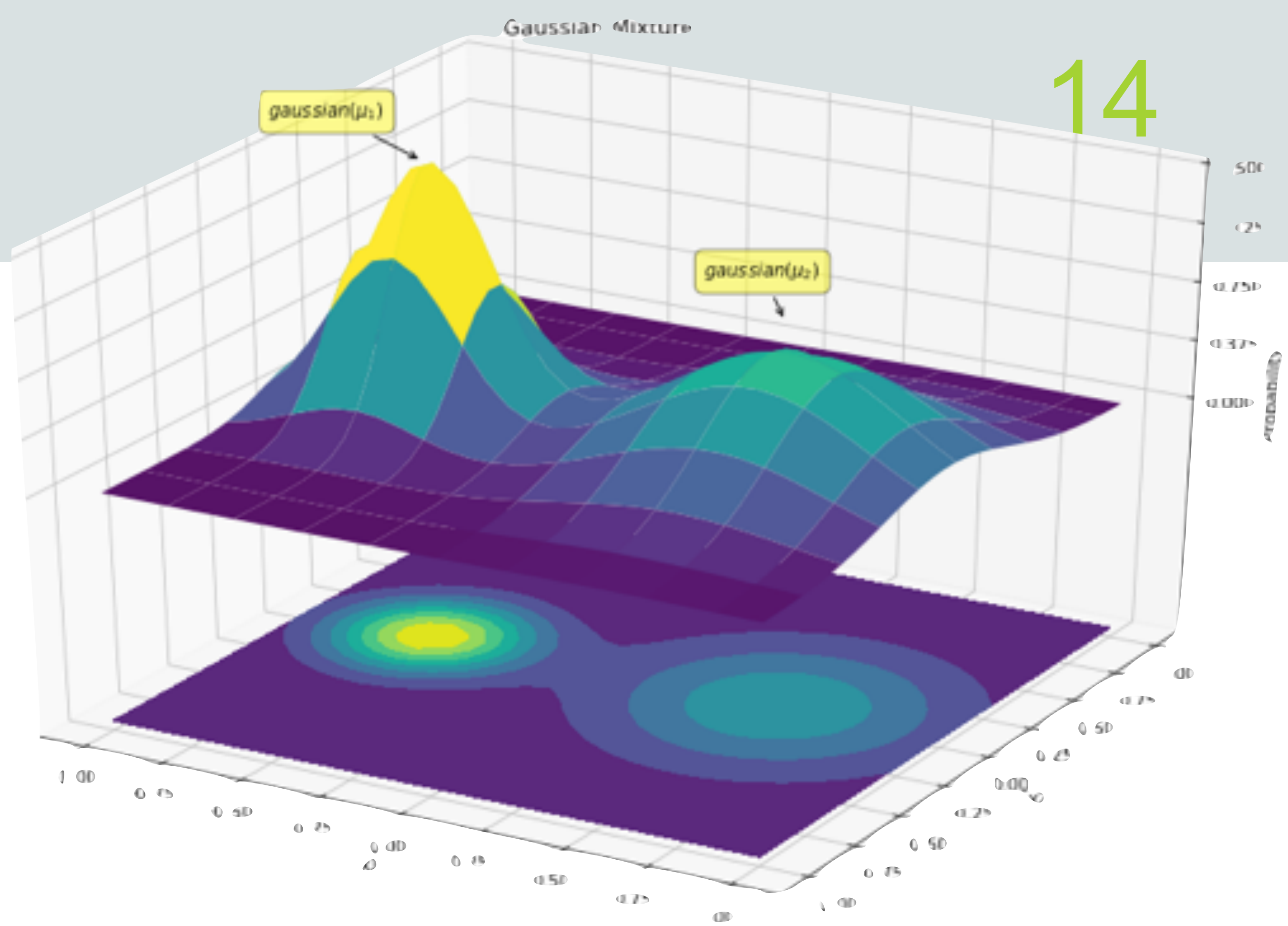
- **Segment and encode** the task according to sequences of *predefined* actions

- **Encoding and regenerating** (HMM)



2. IMITATION LEARNING

2.6. Gaussian Mixture Model and Regression



- We can model observed data X= (x,a) by a probabilistic density distribution P(X) = p(x,a)
  - Gaussian Mixture Models:

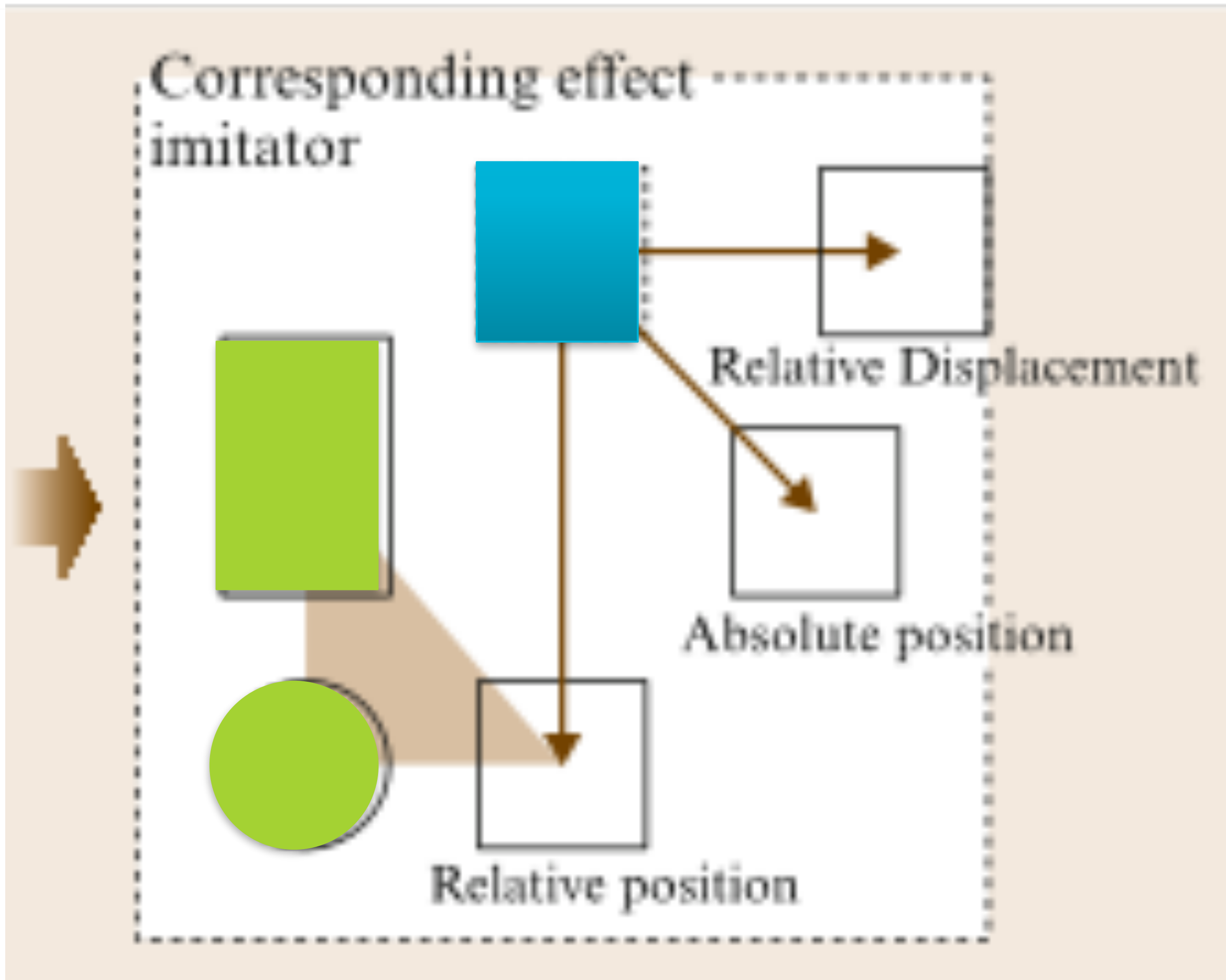
$$p(\boldsymbol{X}, \pi, \boldsymbol{\mu}, \Sigma) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\boldsymbol{X}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

  - Multivariate Gaussian

$$\mathcal{N}(\boldsymbol{X}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)$$

$\boldsymbol{\mu}$ is the mean
$\boldsymbol{\Sigma}$ is the covariance matrix

- We can infer the robotic command
  - v= argmax_v p(v|x)



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

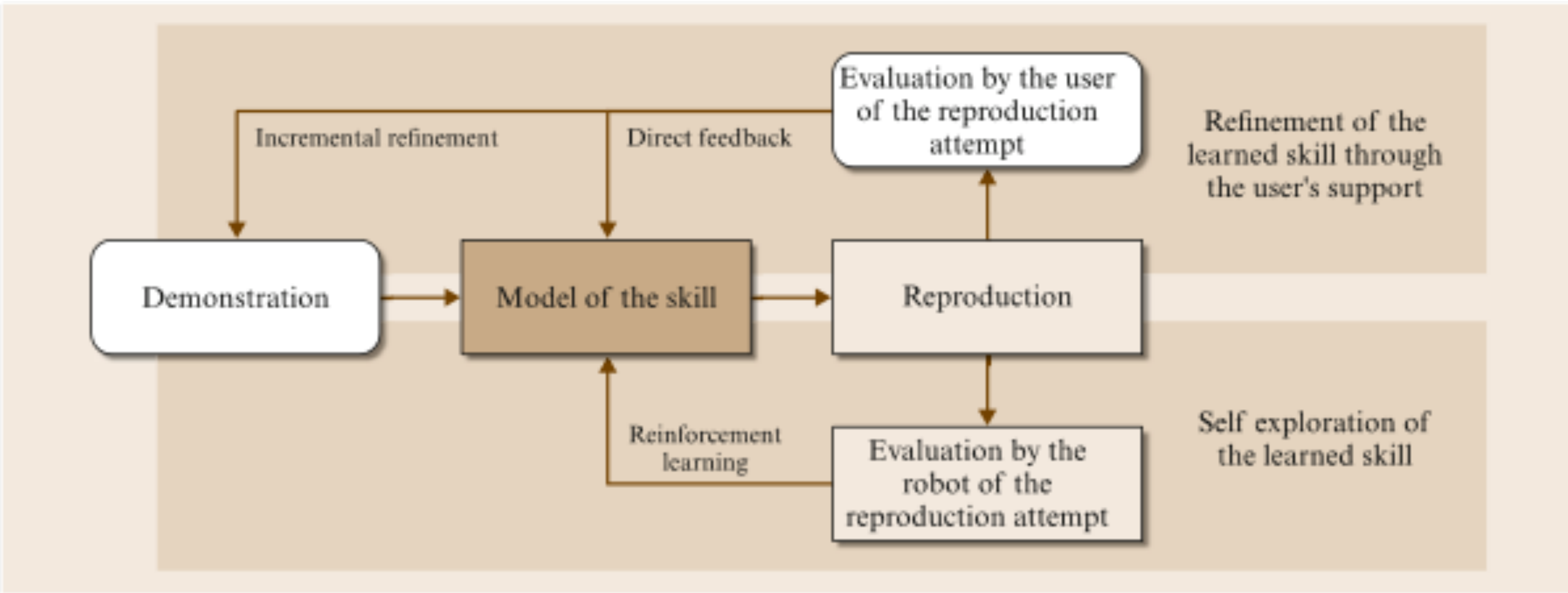
These early works highlighted the importance of providing a set of examples that the robot can use:

- by constraining the demonstrations to **modalities** that the robot can understand
- by providing a sufficient **number of examples** to achieve a desired generality.
- by providing **examples representative** enough of the all the situations
- By limiting the **correspondence problems**

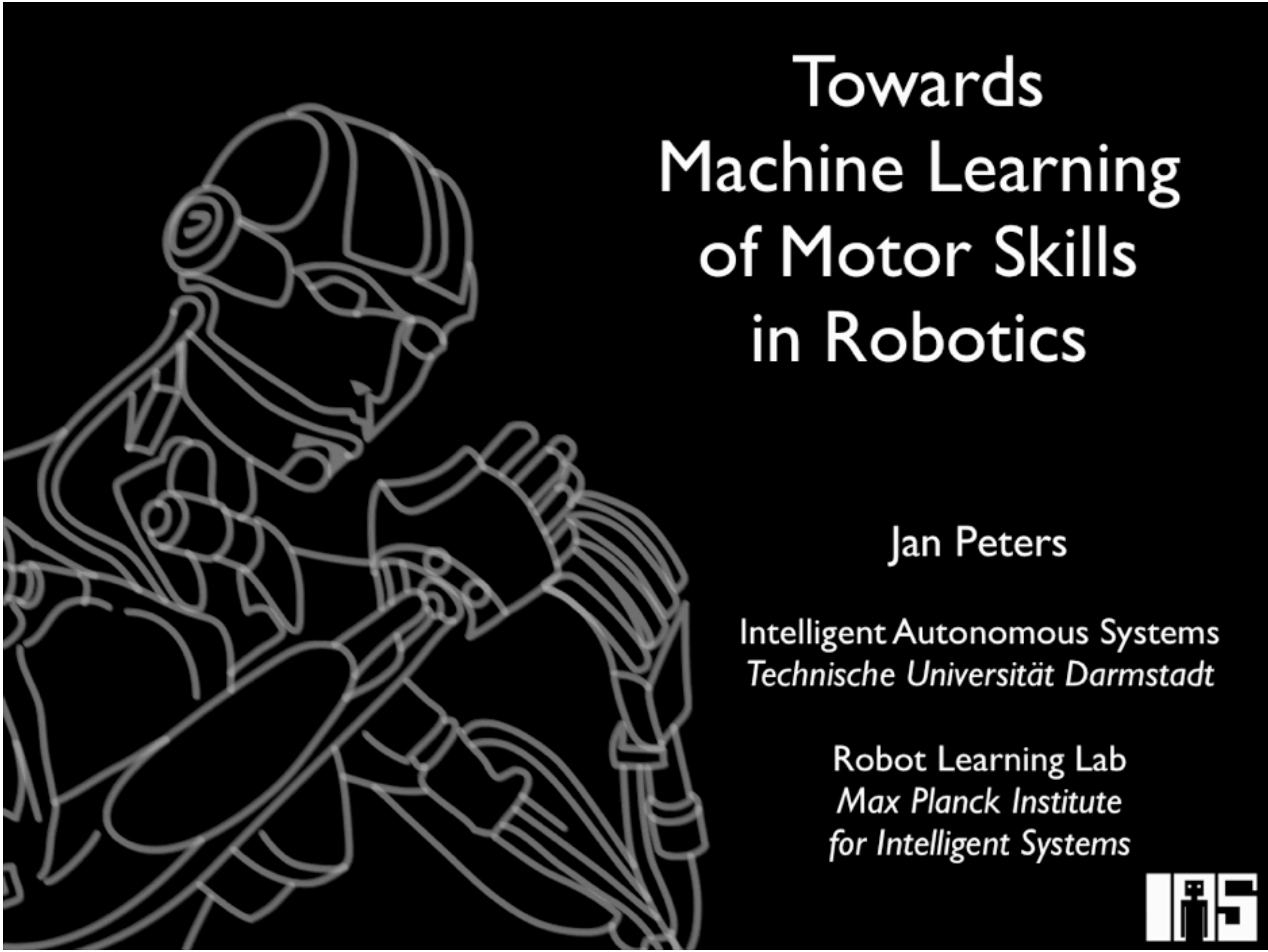
- give the **teacher an active role** during learning
- the interaction aspect of the **transfer process**

- *Social cues*
- Pointing and gazing
- Vocal speech recognition
- Prosody of the speech

PbD can be jointly used with other learning strategies to overcome some limitations of PbD



2.7. BEYOND IMITATION LEARNING



3:13

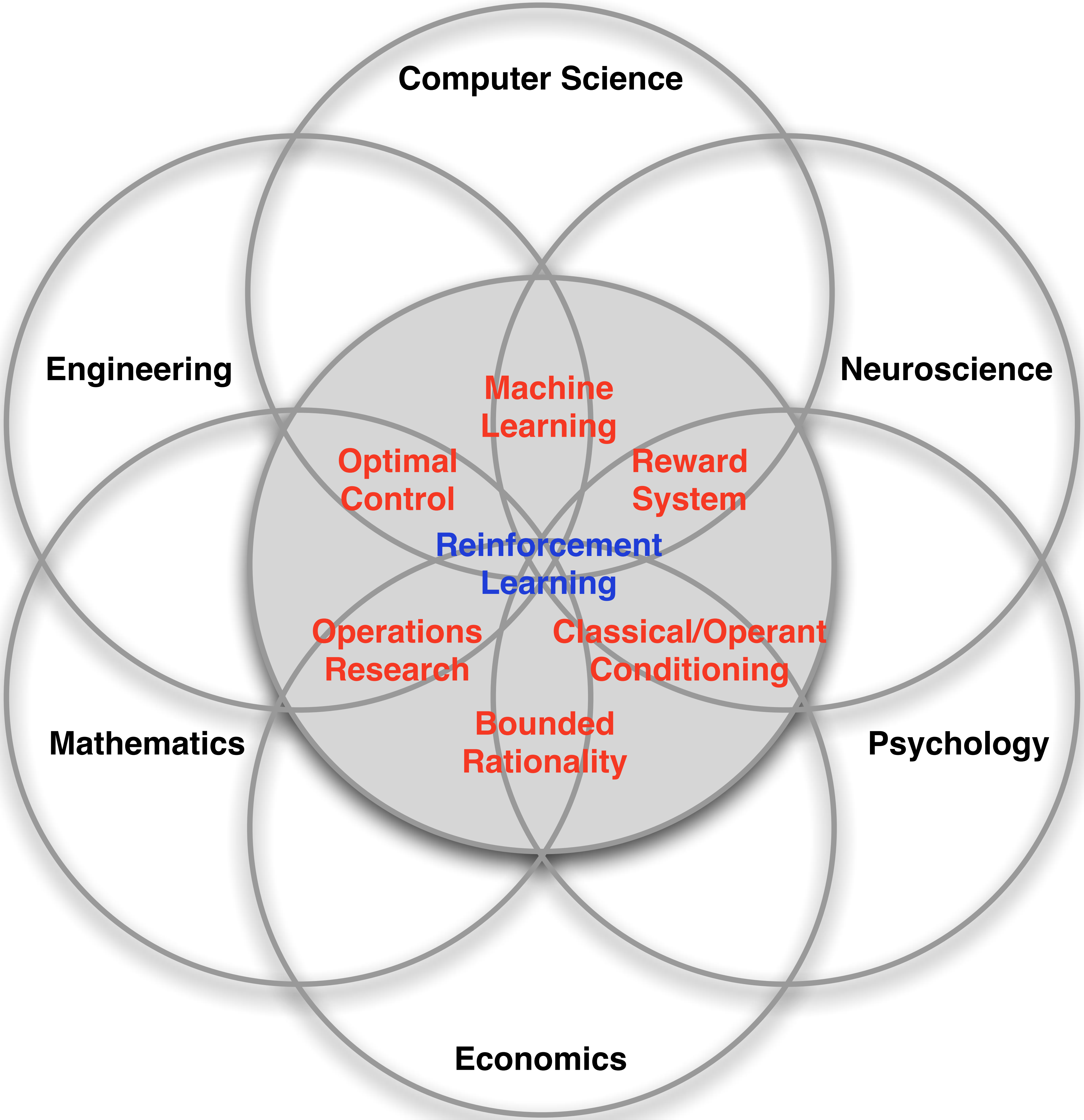
3. INTERACTION WITH THE ENVIRONMENT :

REINFORCEMENT LEARNING

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

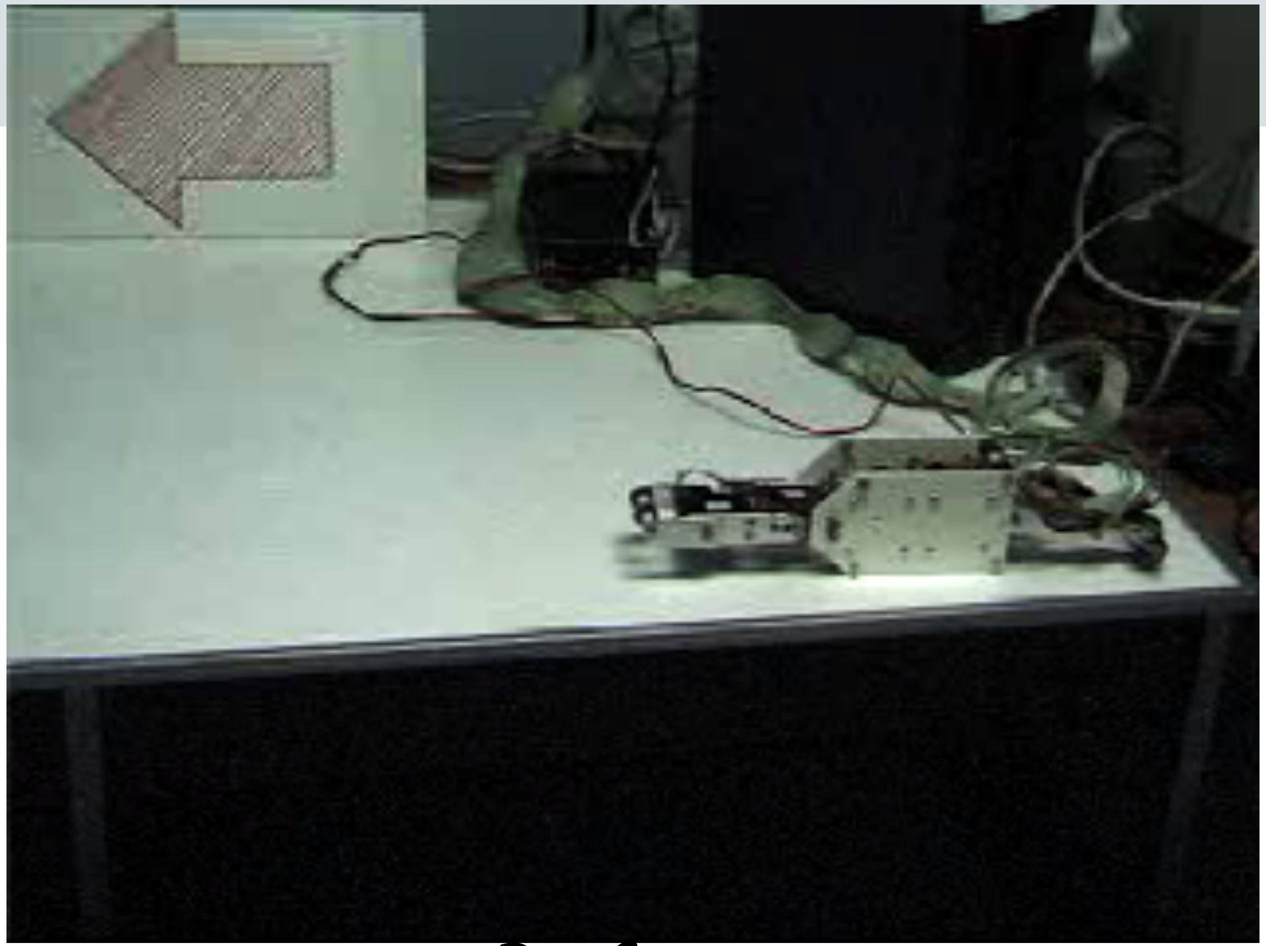
- Agent-oriented learning—learning by **interacting with an environment** to achieve a goal
  - more realistic and ambitious than other kinds of machine learning
- Learning by **trial and error**, with only delayed evaluative feedback (**reward**)
  - the kind of machine learning most like natural learning
  - learning that can tell for itself when it is right or wrong
- The beginnings of a **science of mind** that is neither natural science nor applications technology

3.1. What is reinforcement learning?

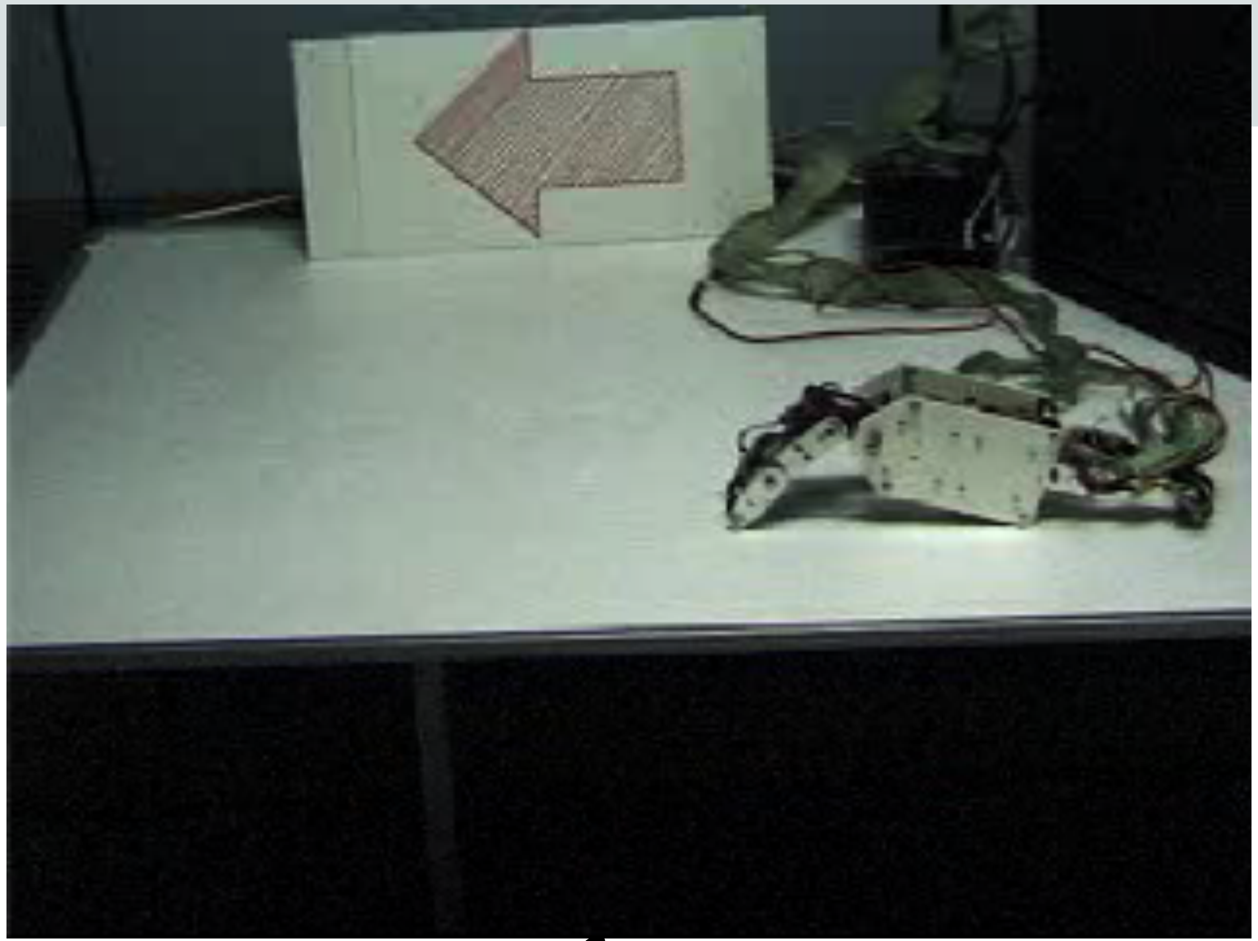


David Silver 2015

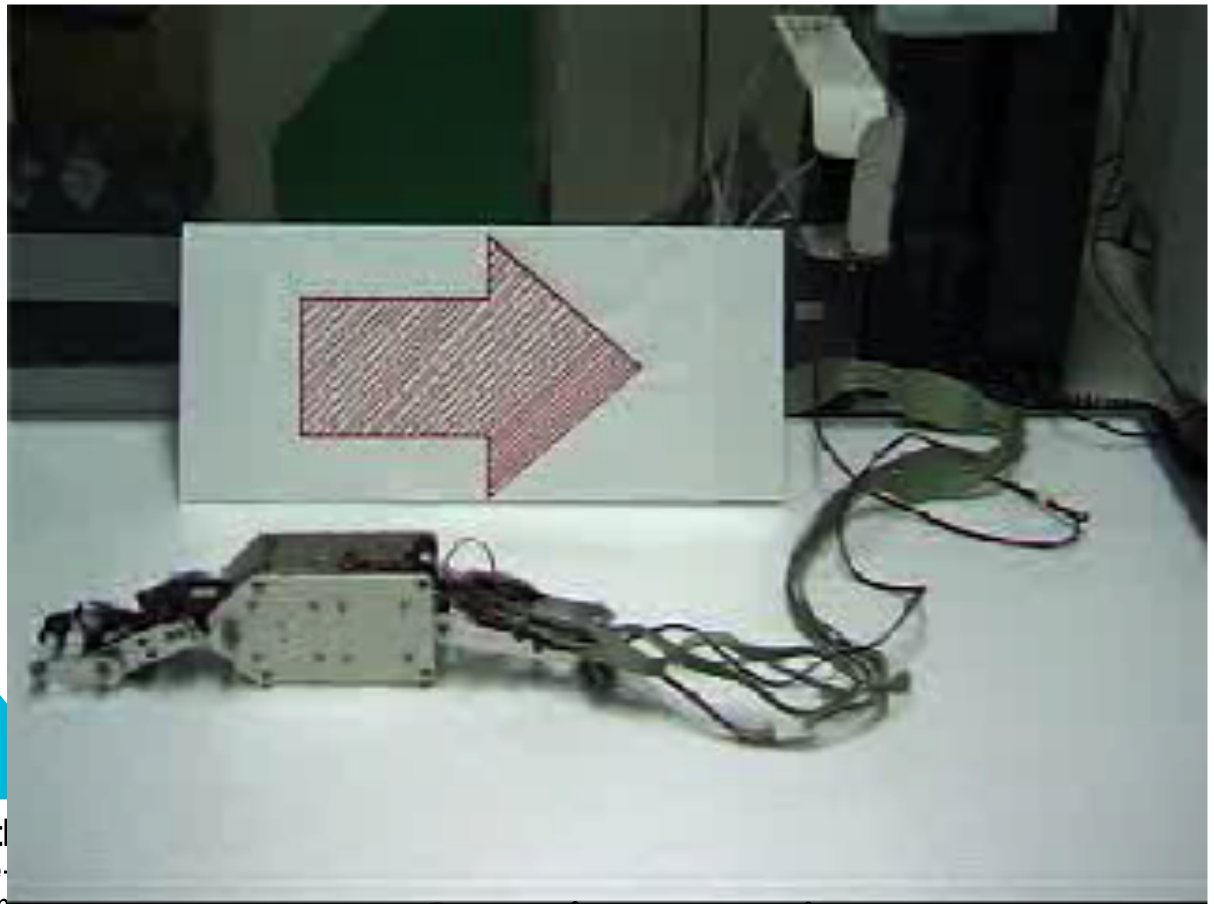
EXAMPLE: HAJIME KIMURA'S RL ROBOTS



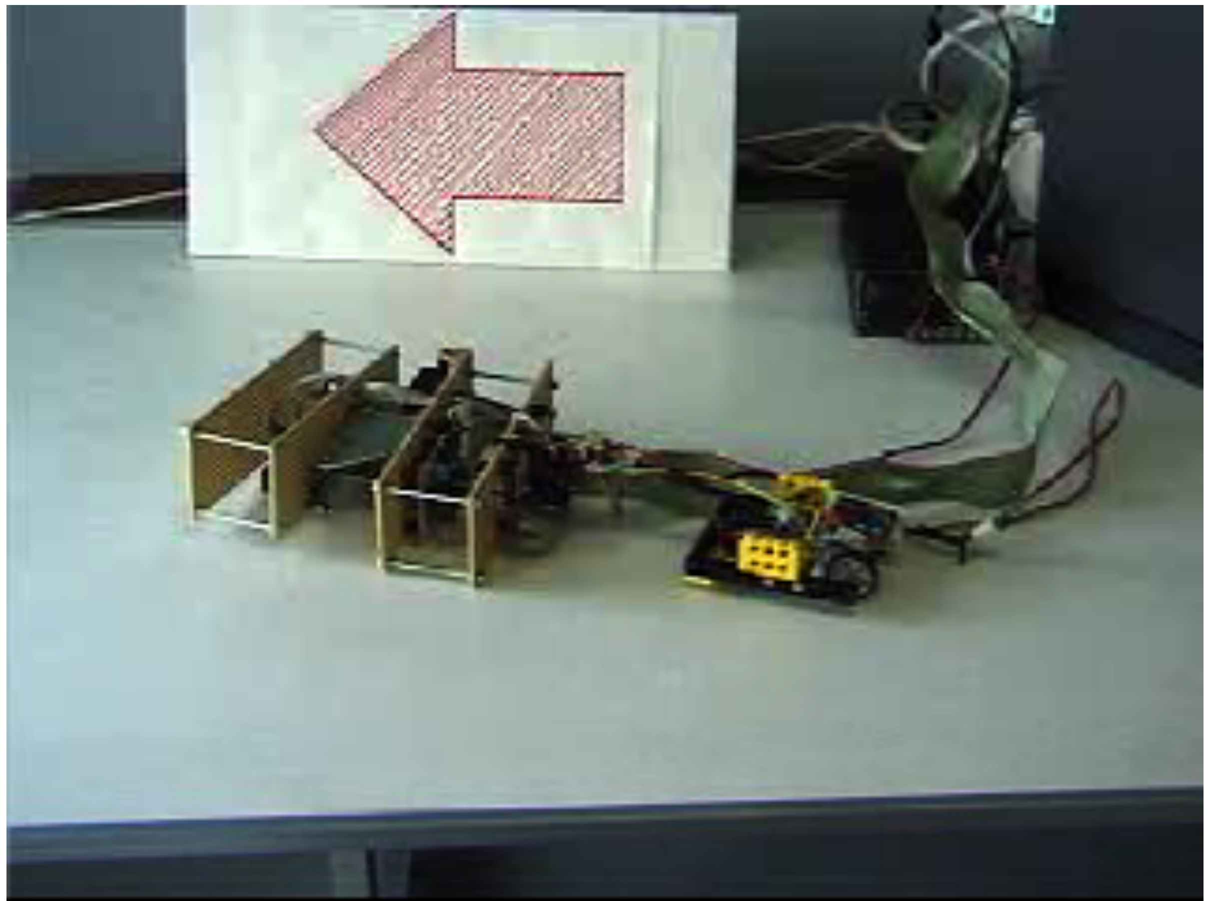
Before



After



Backward

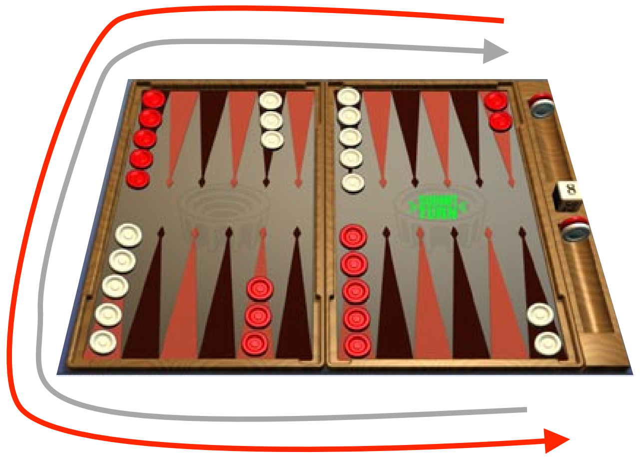


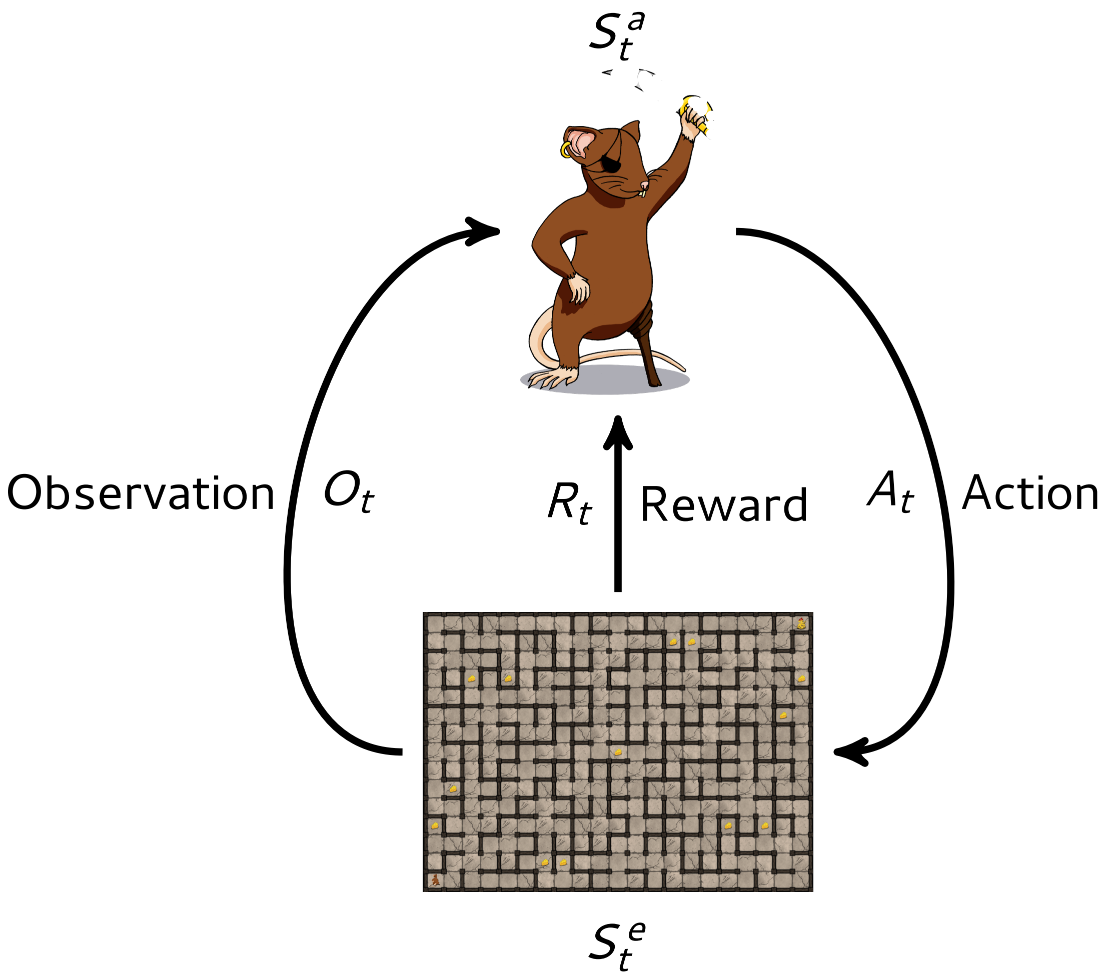
New Robot, Same algorithm

3. REINFORCEMENT LEARNING

3.2. Some Reinforcement Learning Successes

- ✤Learned the world's best player of Backgammon (Tesauro 1995)
- ✤Learned acrobatic helicopter autopilots (Ng, Abbeel, Coates et al 2006+)
- ✤Widely used in the placement and selection of advertisements and pages on the web (e.g., A-B tests)
- ✤Used to make strategic decisions in *Jeopardy!* (IBM's Watson 2011)
- ✤Achieved human-level performance on Atari games from pixel-level visual input, in conjunction with deep learning (Google Deepmind 2015)
- ✤Google Deepmind's AlphaGo defeats the world Go champion, vastly improving over all previous programs (2016)
- ✤In all these cases, performance was better than could be obtained by any other method, and was obtained without human instruction





- The agent …
  - performs action A_t
  - obtains an observation O_t
  - obtains reward R_t
- The environment …
  - receives action At
  - produces Ot
  - produces reward Rt

- Agent seeks to maximize its cumulative **reward** on the long run
- Agent learns a policy **mapping states to actions**
- Environment may be unknown, nonlinear, stochastic and complex and non-observable :
  - Full observability : $S_t^a = S_t^o = O_t$
  - Partial observability: s_t^a is estimated by the environment

- Policy π
  - A policy is the agent behavior
  - Map from state to action
  - Deterministic : $a = \pi(s)$
  - Stochastic : $\pi(a|s) = P[A_t = a|S_t = s]$
- Value Function V
  - Prediction of future reward
  - Evaluates the goodness of states
  - Action selection using the value function
  - $v_\pi(s)=\mathbb{E}(R_{t+1} + \gamma R_{t+2} + \ldots|S_t = s)$
- Q-Value Function Q
  - same as V but for each action : prediction of future reward
  - Evaluates the goodness of state-action pairs
  - Action selection using the value function
  - $Q_\pi(s,a)=\mathbb{E}(R_{t+1} + \gamma R_{t+2} + \ldots|S_t = s, a_t = a)$

- $V_\pi(s)=\mathbb{E}(R_{t+1} + \gamma R_{t+2} + \ldots | S_t = s)$

- Optimal solution
  - Policy π^*
  - $V^*(s) = \max_\pi V_\pi(s)$
  - $Q^*(s,a) = \max_\pi Q_\pi(s,a)$

- Bellmann equation: for s,a,r and next state s'
  - $V^*(s) = \max_a [R(s)+\gamma V^*(s') \mid s,a]$
  - $Q^*(s,a)=\mathbb{E} [R(s) + \gamma \max_{a'} Q(s',a') \mid s,a]$
  - Bellman optimality equation expresses the fact that the value of a state under an optimal policy must equal the expected return for the best action from that state

Input: the policy π to be evaluated

Algorithm parameter: step size $\alpha \in (0, 1]$

Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\textit{terminal}) = 0$

Loop for each episode:

  Initialize S

  Loop for each step of episode:

    $A \leftarrow$ action given by π for S

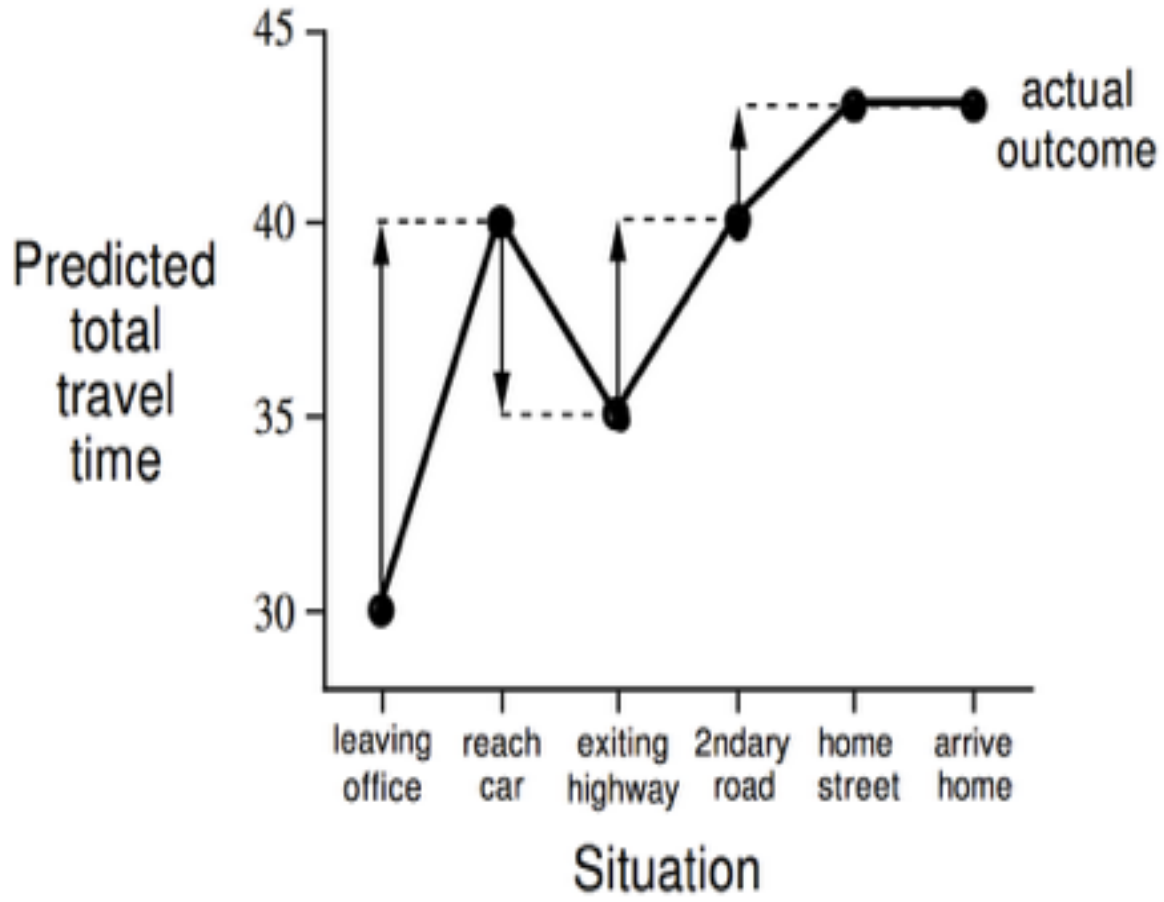
    Take action A, observe R, S'

    $V(S) \leftarrow V(S) + \alpha\big[\underbrace{R + \gamma V(S')}_{\textbf{target}\text{: an estimate of the return}} - V(S)\big]$

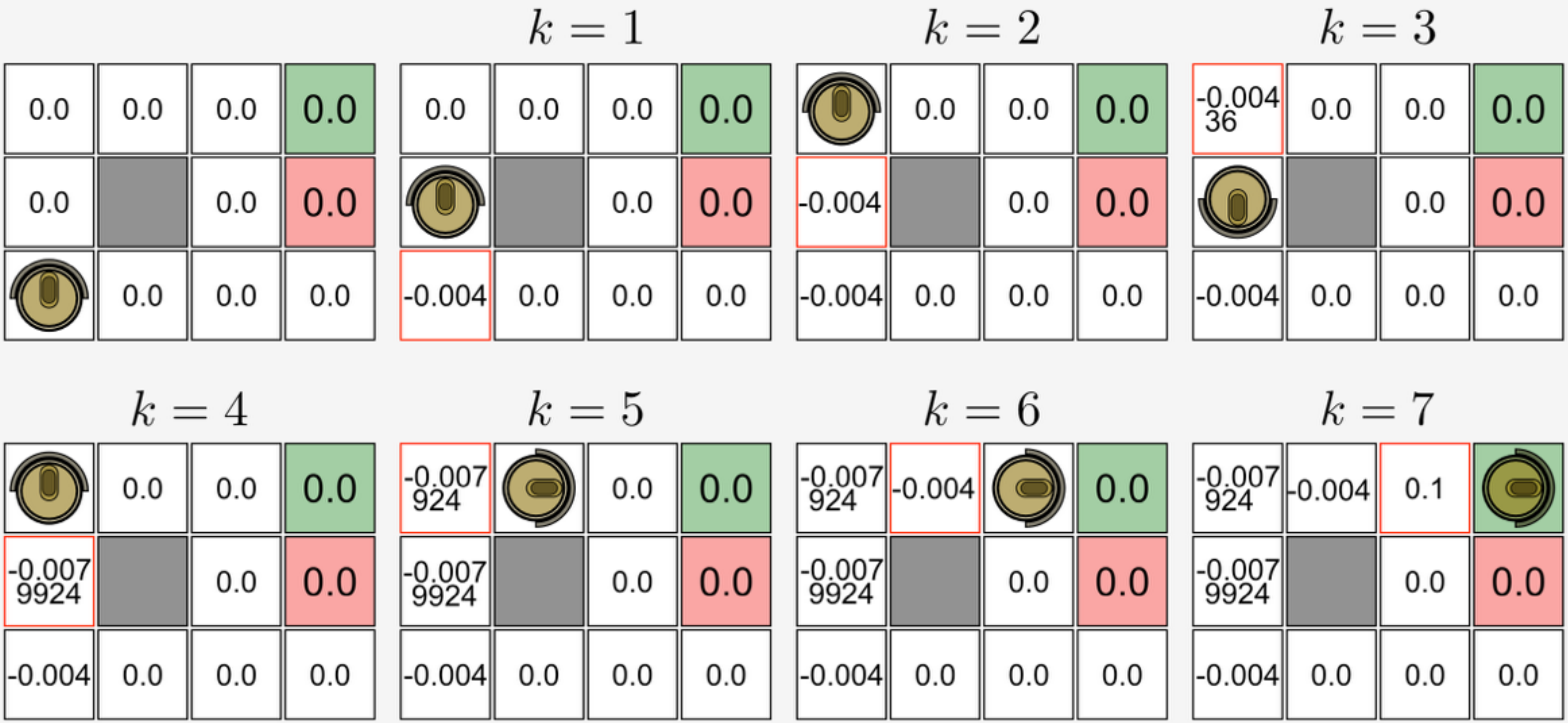
    $S \leftarrow S'$

  until S is terminal

State	*Elapsed Time (minutes)*	*Predicted Time to Go*	*Predicted Total Time*
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43

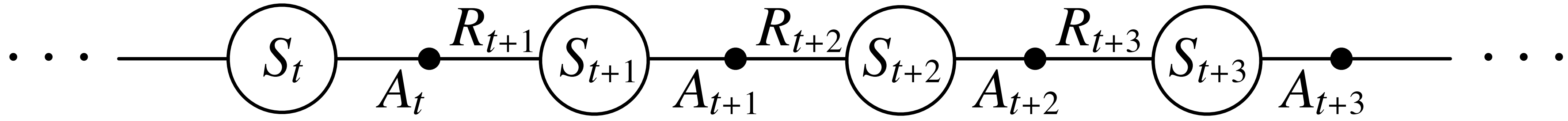


3.6. TD Prediction



3. REINFORCEMENT LEARNING

3.7. Sarsa: On-Policy TD Control



From state $\mathbf{s}_t$, choose action $\mathbf{a}_t$, observe $\mathbf{r}_{t+1}$, $\mathbf{s}_{t+1}$, choose $\mathbf{a}_{t+1}$

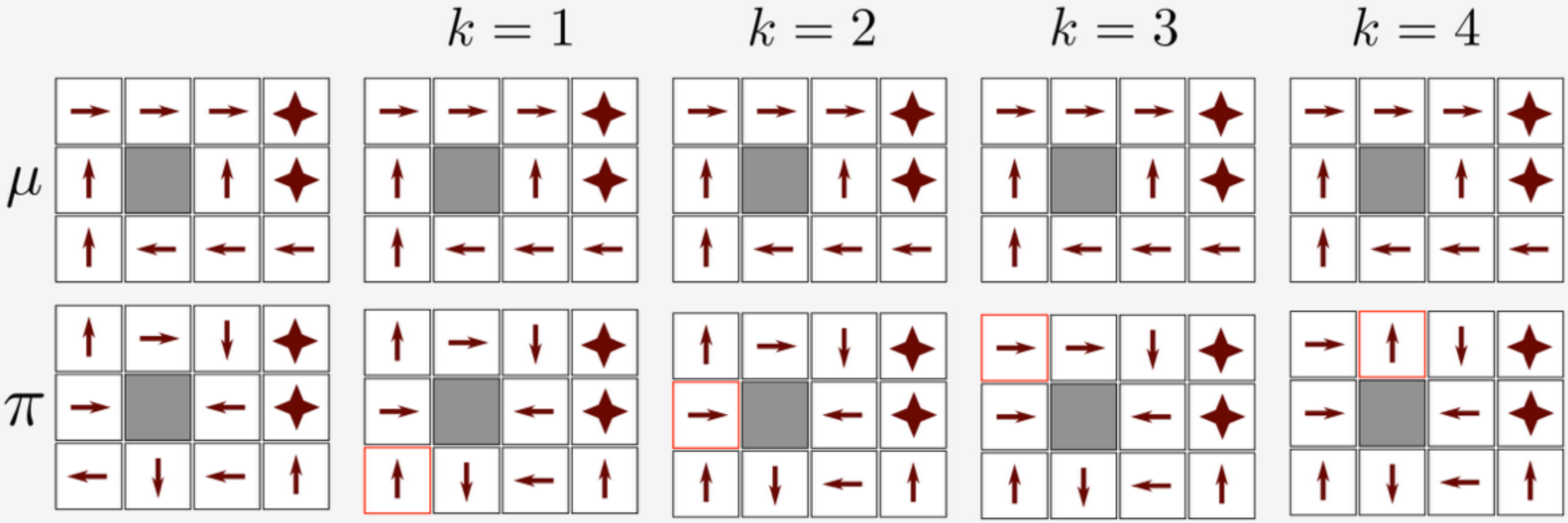
Update the state-action function Q(st,at) to update policy

Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize S
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Repeat (for each step of episode):
        Take action A, observe R, S'
        Choose A' from S' using policy derived from Q (e.g., ε-greedy)
        $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$
        $S \leftarrow S'$; $A \leftarrow A'$;
    until S is terminal

One-step Q-learning:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

Initialize $Q(s, a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize S
    Repeat (for each step of episode):
        Choose A from S using policy derived from Q (e.g., ε-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
        $S \leftarrow S'$;
    until S is terminal



$k = 1$
$k = 2$
$k = 3$
$k = 4$
μ
π

Bibliography

Imitation learning

- Aude Billard, Sylvain Calinon, Rüdiger Dillmann, Stefan Schaal, Ch 59 Robot Programming by Demonstration in : Siciliano, Bruno, and Oussama Khatib, eds. *Springer handbook of robotics*. Springer, 2016.
- S. Calinon, A. Billard: What is the Teacher's Role in Robot Programming by Demonstration? - Toward Benchmarks for Improved Learning, Interact. Stud. 8(3), 441–464 (2007), Special Issue on Psychological Benchmarks in Human-Robot Interaction
- S. Calinon, F. Guenter, A. Billard: On Learning Representing and Generalizing a Task in a Humanoid Robot, IEEE Trans. Syst. Man Cybernet. 37(2), 286– 298 (2007), Special issue on robot learning by observation, demonstration and imitation

Reinforcement learning

- R. S. Sutton and A. G. Barto. Reinforcement Learning: an introduction. MIT Press, 1998.
- https://mpatacchiola.github.io/blog/