Social Learning and Imitation for Teamwork

Manuel Lopes

INRIA

Bordeaux Sud-Ouest

manuel.lopes@inria.fr

flowers.inria.fr/mlopes

Outline

- Interactive Learning
  - Ambiguous Protocols
  - Ambiguous Signals
  - Active Learning

- Inverse Reinforcement Learning for Team Coordination
  - IRL in distributed multi-agent scenarios

Learning from Demonstration

Pros

- Natural/intuitive (is it?)
- 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)

What is the best strategy to learn/teach?

Considering teaching how to play tennis.

Information provided:

- Rules of the game
  $R(x)$
- Strategies or verbal instructions of how to behave
  $V(x)>V(y)$
- Demonstrations (demonstration of a particular hit)
  $\pi(x)=a$

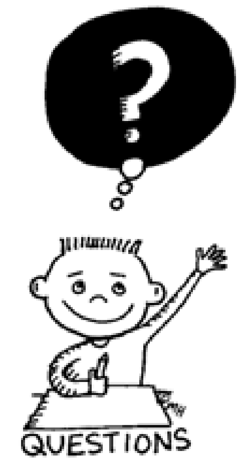
How to improve learning from demonstration?

- Combine:
  - demonstrations to initialize
  - self-experiment to correct modeling errors
- Feedback corrections
- Instructions
- More data
- …

How to improve learning/teaching?

Learner

– Active Learning
– Combine with Self-Experimentation

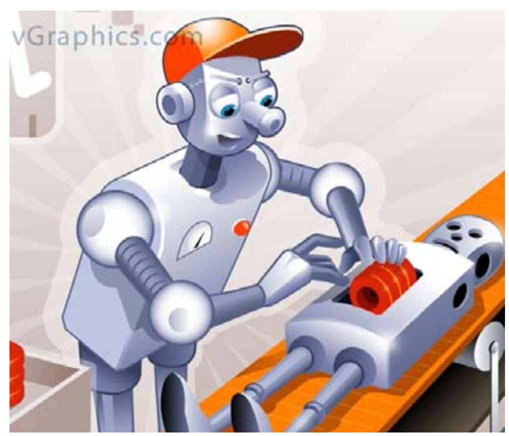


Teacher

– Better Strategies
– Extra Cues

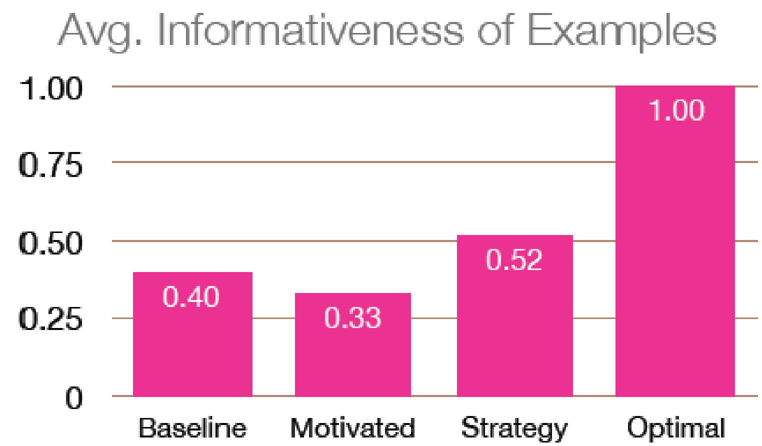
How are demonstrations provided?

- Remote control (direct control)
  - Exoskeleton, joystick, Wiimote,…
- Unobtrusive
  - Acquired with vision, 3d-cameras from someone's execution
- Remote instruction (indirect control)
  - Verbal commands, gestures, …



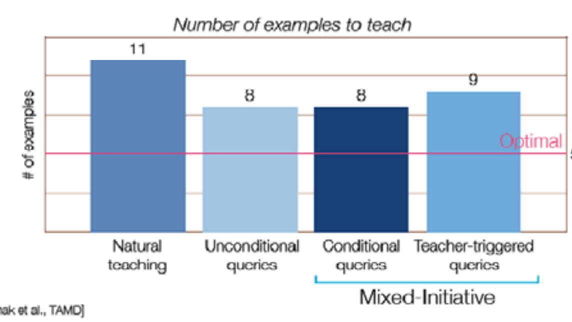
Behavior of Humans

- People want to direct the agent's attention to guide exploration
- People have a positive bias in their rewarding behavior, suggesting both instrumental and motivational intents with their communication channel.
- People adapt their teaching strategy as they develop a mental model of how the agent learns.
- **People are not optimal, even when they try to be so**



Mixed Initiative Active Learning

As good as traditional AL (unconditional queries).



[Cakmak et al., TAMD]

20

Cakmak, Thomaz

Interactive Learning Approaches

Active Learner

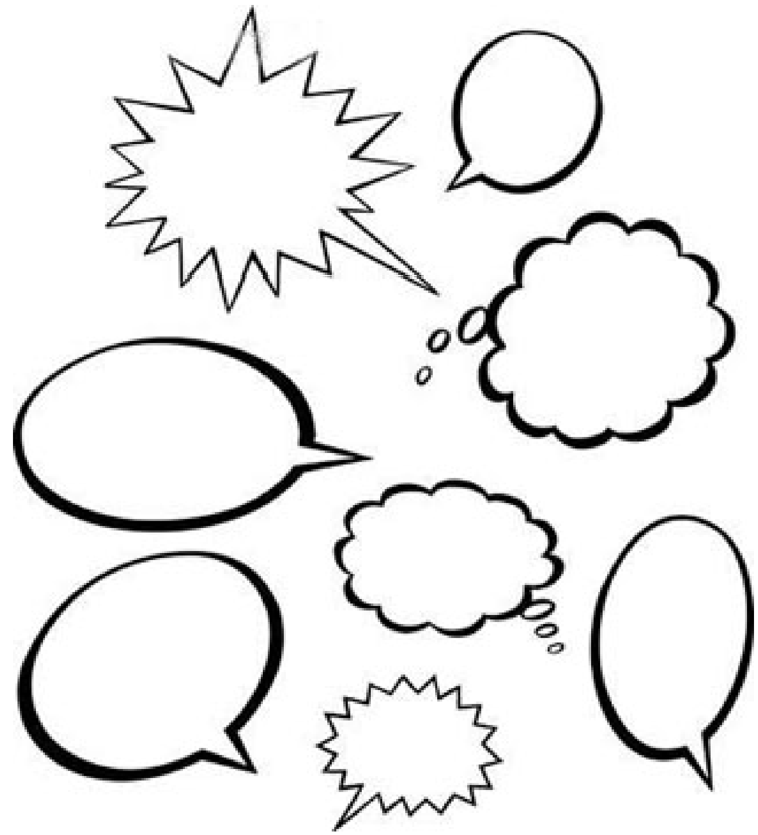
- Decide what to ask (*Lopes*)
- Ask when when Uncertain/Risk (*Chernova*, *Roy*, …)
- Decide when to ask (*Cakmak*)
- …

Improved Teacher

- Dogged Learning (*Grollman*)
- User Preferences (*Mason*)
- Extra Cues (*Thomaz, Knox, Judah*)
- User Queries the Learner (*Cakmak*)
- Tactile Guidance (*Billard*)
- …

Learning under a weakly specified protocol

- People do not follow protocols rigidly
- Some of the provided cues depart from their mathematical meaning, e.g. extra utterances, gestures, guidance, motivation
- Can we exploit those extra cues?
- If robots adapt to the user, will training be easier?



Different Feedback Structures

User can provide direct feedback:

- Reward
  - Quantitative evaluation
- Corrections
  - Yes/No classifications of behavior
- Actions

User can provide extra signals:

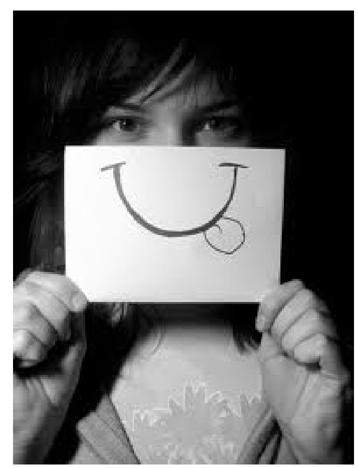
- Reward of exploratory actions
- Reward of getting closer to target

Unknown/Ambiguous Feedback

Unknown feedback signals:

- Gestures
- Prosody
- Word synonyms
- …





Goal / Contribution

Learn simultaneously:

- Task

  reward function

- Interaction Protocol

  what information is the user providing

- Meaning of extra signals

  what is the meaning of novel signals, e.g. prosody, unknown works,…

Simultaneous Acquisition of Task and Feedback Models, Manuel Lopes, Thomas Cederborg and Pierre-Yves Oudeyer. *IEEE - International Conference on Development and Learning (ICDL)*, Germany, 2011.

Markov decision process

Set of possible states of the world and actions:

$X = \{1, ..., |X|\}$ $A = \{1, ..., |A|\}$

- State evolves according to

  $P[X_{t+1} = y \mid X_t = x, A_t = a] = \mathbf{P}_a(x, y)$

- Reward r defines the task of the agent

- A policy defines how to choose actions

  $P[A_t = a \mid X_t = x] = \pi(x, a)$

- Determine the policy that maximizes the total (expected) reward:

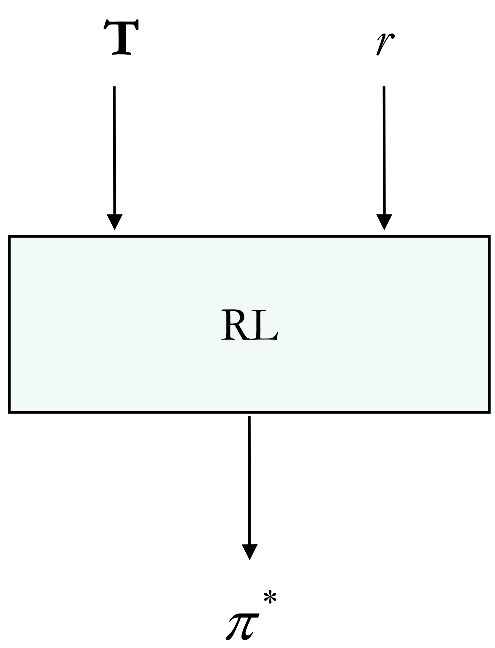
  $V(x) = E_\pi[\sum_t g^t r_t \mid X_0 = x]$

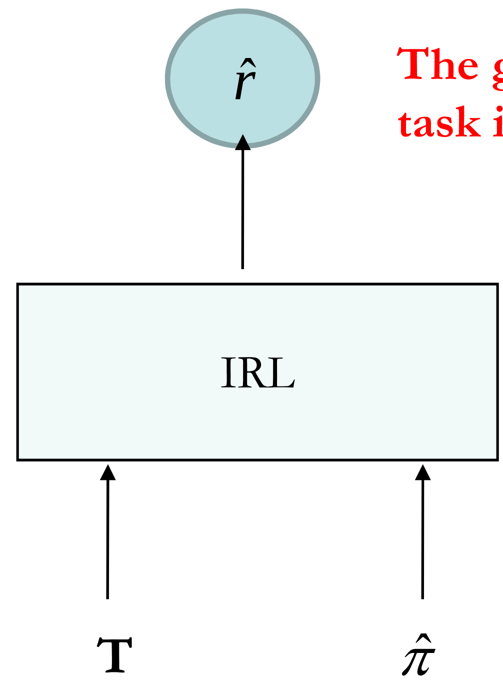
- Optimal policy can be computed using DP:

  $V^*(x) = r(x) + g \max_a E_a[V^*(y)]$

  $Q^*(x, a) = r(x) + \gamma E_a[V^*(y)]$

Inverse Reinforcement Learning





The goal of the task is unknown

From world model and reward
Find optimal policy

From samples of the policy and world model
Estimate reward

Ng et al, ICML00; Abbeel et al ICML04; Neu et al, UAI07; Ramachandran et al IJCAI 07; Lopes et al IROS07

Probabilistic View of IRL

- Suppose now that agent is given a demonstration:
  D = $\{(x_1, a_1), \ldots, (x_n, a_n)\}$
- The teacher is not perfect (sometimes makes mistakes)

$$\pi'(x, a) = \frac{e^{\eta Q^*(x,a)}}{\sum_b e^{\eta Q^*(x,b)}}$$

- Likelihood of observed demo: $L(D) = \prod_i \pi'(x_i, a_i)$

- Prior distribution $P[r]$
- Likelyhood of demo,

$$L(D) = \prod_i \pi_r(x_i, a_i)$$

- Posterior over rewards:

$$P[r \mid D] \propto P[r]\, P[D \mid r]$$

- MC-based methods to sample $P[r \mid D]$

Ramachandran

Bayesian inverse reinforcement learning

Algorithm $\texttt{PolicyWalk}$(Distribution P, MDP M, Step Size δ)

1. Pick a random reward vector $\boldsymbol{R} \in \mathbb{R}^{|S|}/\delta$.
2. $\pi := \texttt{PolicyIteration}(M, \boldsymbol{R})$
3. Repeat
   (a) Pick a reward vector $\tilde{\boldsymbol{R}}$ uniformly at random from the neighbours of $\boldsymbol{R}$ in $\mathbb{R}^{|S|}/\delta$.
   (b) Compute $Q^{\pi}(s, a, \tilde{\boldsymbol{R}})$ for all $(s, a) \in S, A$.
   (c) If $\exists(s, a) \in (S, A)$, $Q^{\pi}(s, \pi(s), \tilde{\boldsymbol{R}}) < Q^{\pi}(s, a, \tilde{\boldsymbol{R}})$
      i. $\tilde{\pi} := \texttt{PolicyIteration}(M, \tilde{\boldsymbol{R}}, \pi)$
      ii. Set $\boldsymbol{R} := \tilde{\boldsymbol{R}}$ and $\pi := \tilde{\pi}$ with probability $\min\{1, \frac{P(\tilde{\boldsymbol{R}}, \tilde{\pi})}{P(\boldsymbol{R}, \pi)}\}$

      Else
      i. Set $\boldsymbol{R} := \tilde{\boldsymbol{R}}$ with probability $\min\{1, \frac{P(\tilde{\boldsymbol{R}}, \pi)}{P(\boldsymbol{R}, \pi)}\}$
4. Return $\boldsymbol{R}$

Gradient-based IRL

- Idea: Compute the maximum-likelihood estimate for r given the demonstration D
- We use a gradient ascent algorithm:

$$r_{t+1} = r_t + \nabla_r L(\mathrm{D})$$

- Upon convergence, the obtained reward maximizes the likelihood of the demonstration

Policy Loss (*Neu et al.*), Maximum likelihood (*Lopes et al.*)

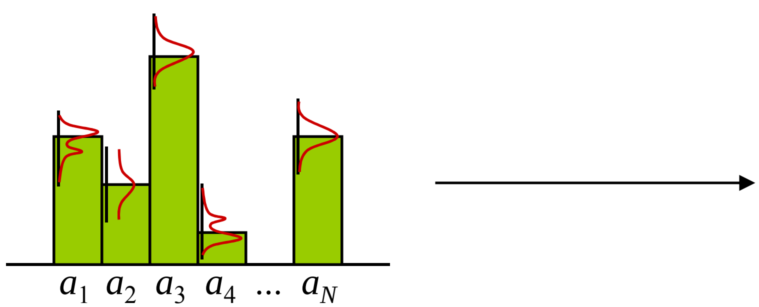
The Selection Criterion

- Distribution $P[r \mid \mathbb{D}]$ induces a distribution on Π
- Use MC to approximate $P[r \mid \mathbb{D}]$
- For each (x, a), $P[r \mid \mathbb{D}]$ induces a distribution on $\pi(x, a)$:

$$\mu_{xa}(p) = P[\pi(x, a) = p \mid \mathbb{D}]$$

- Compute per state average entropy:

$$H(x) = {}^{1}/_{|A|} \sum_a H(\mu_{xa})$$



Compute entropy $H(\mu_{xa})$

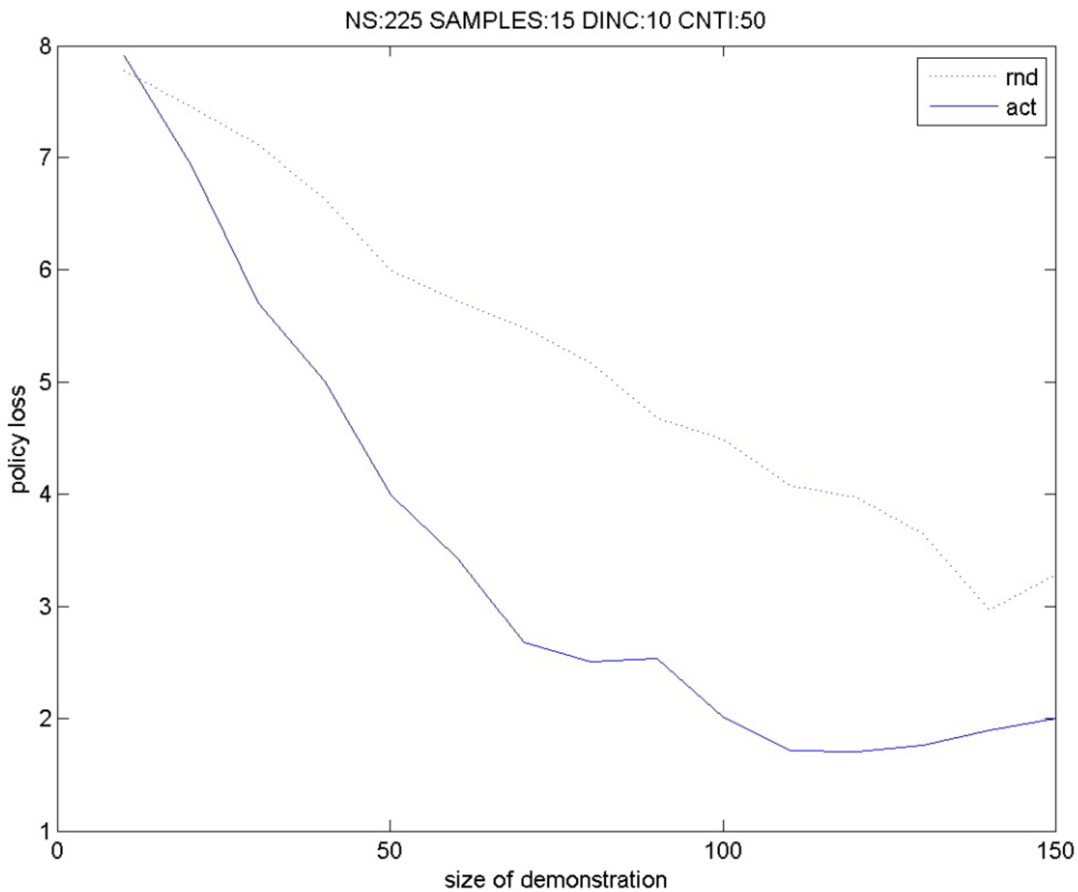
Active IRL

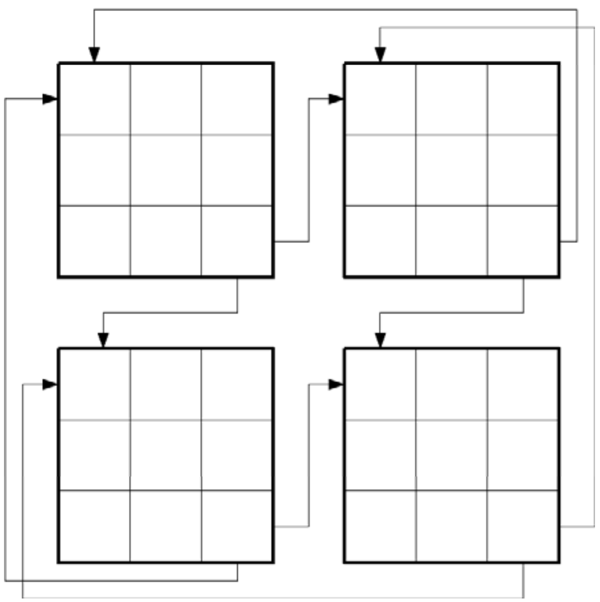
Require: Initial demonstration D

1. Estimate P[π | D] using MC
   maybe only around the ML estimate
2. **for all** $x \in X$
3. Compute $H(x)$
4. **endfor**
5. Query action for $x^* = \text{argmax}_x H(x)$
6. Add new sample to D

Active Learning for Reward Estimation in Inverse Reinforcement Learning, Manuel Lopes, Francisco Melo and Luis Montesano. *European Conference on Machine Learning (ECML/PKDD),* Bled, Slovenia, 2009.

Results III. General Grid World



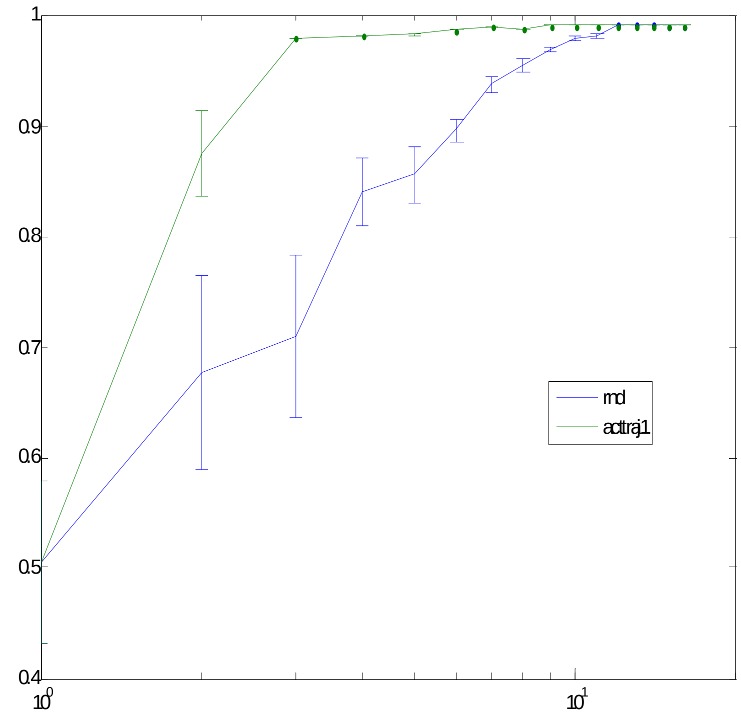


- General grid world ($M \times M$ grid), >200 states
- Four actions available (N, S, E, W)
- Parameterized reward (goal state)

Active IRL, sample trajectories

Require: Initial demonstration D

1. Estimate P[π | D] using MC
2. **for all** $x \in X$
3. Compute $H(x)$
4. **endfor**
5. Solve MDP with R=H(x)
6. Query trajectory following optimal policy
7. Add new trajectory to D



Unknown/Ambiguous Feedback

Unknown feedback protocol

The information provided by the demonstration **has not** a predefined semantics

Meanings of the user signals

- **Binary Reward**
- **Action**

Feedback Profiles

FEEDBACK PROFILES. POSSIBLE FEEDBACK INSTRUCTIONS GIVEN BY THE USER WHEN THE ROBOT DOES THE **CORRECT** OR **WRONG** ACTION ARE: THE ACTION NAME, NOTHING, CORRECT OR WRONG. EIGHT FEEDBACK PROFILES WERE CONSIDERED.

Action \ Feedback	1	2	3	4	5	6	7	8
Correct	A	A	A	∅	∅	O	O	O
Wrong	A	∅	W	A	W	A	∅	W

Demonstration

Ambiguous

Ambiguous

Binary Reward

Combination of Profiles

Each different teacher will be modeled has a convex combination of these profiles. For the teacher model we will consider a set of parameters M that describe the mixture of profiles in Table I. As an example, consider $M = [0\ 0.8\ 0\ 0\ 0\ 0.2\ 0\ 0]$, the statistical model for the feedback is as follows:

$$\text{if A is optimal} \quad \begin{cases} p(F = A|A, M) & = 0.8 \\ p(F = O|A, M) & = 0.2 \end{cases}$$

$$\text{if A is non-optimal} \quad \begin{cases} p(F = \emptyset|A, M) & = 0.8 \\ p(F = A|A, M) & = 0.2 \end{cases}$$

Acquisition of Task and Feedback Model

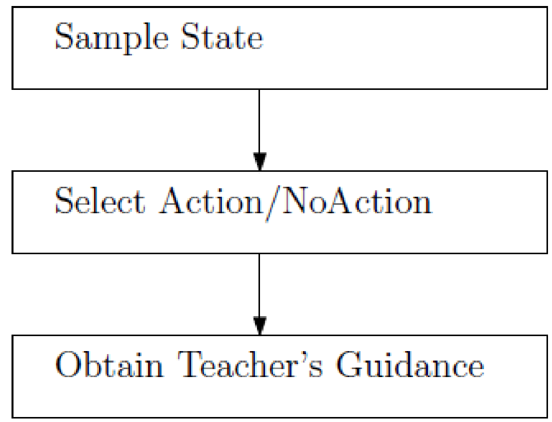


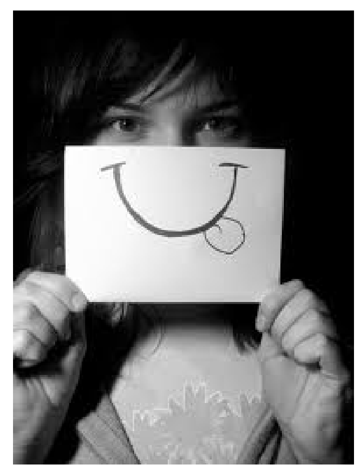
Fig. 1. Learning protocol. The robot experiments an action at a given state and based on that a user provides a guidance signal that consists on unknown combinations of confirmation/correction signals, or directly policy information (e.g. “yes”, “no”, “up” or “down”).

$$
\begin{aligned}
& p(R_{t+1}, M_{t+1} | A_{0:t}, F_{0:t}) \\
& \quad \propto \quad p(F_t | A_t, R_t, M_t) p(R_t, M_t | A_t) \\
& \quad \propto \quad p(F_t | A_t, R_t, M_t) p(A_t | M_t, R_t) p(R_t, M_t) \\
& \quad = \quad p(F_t | A_t, R_t, M_t) p(A_t | R_t) p(R_t, M_t)
\end{aligned}
$$

Unknown/Ambiguous Feedback

Unknown feedback signals:

- Gestures
- Prosody
- Word synonyms
- …



Feedback meaning of user signals

User might use different words to provide feedback

- Ok, correct, good, nice, …
- Wrong, error, no no, …
- Up, Go, Forward

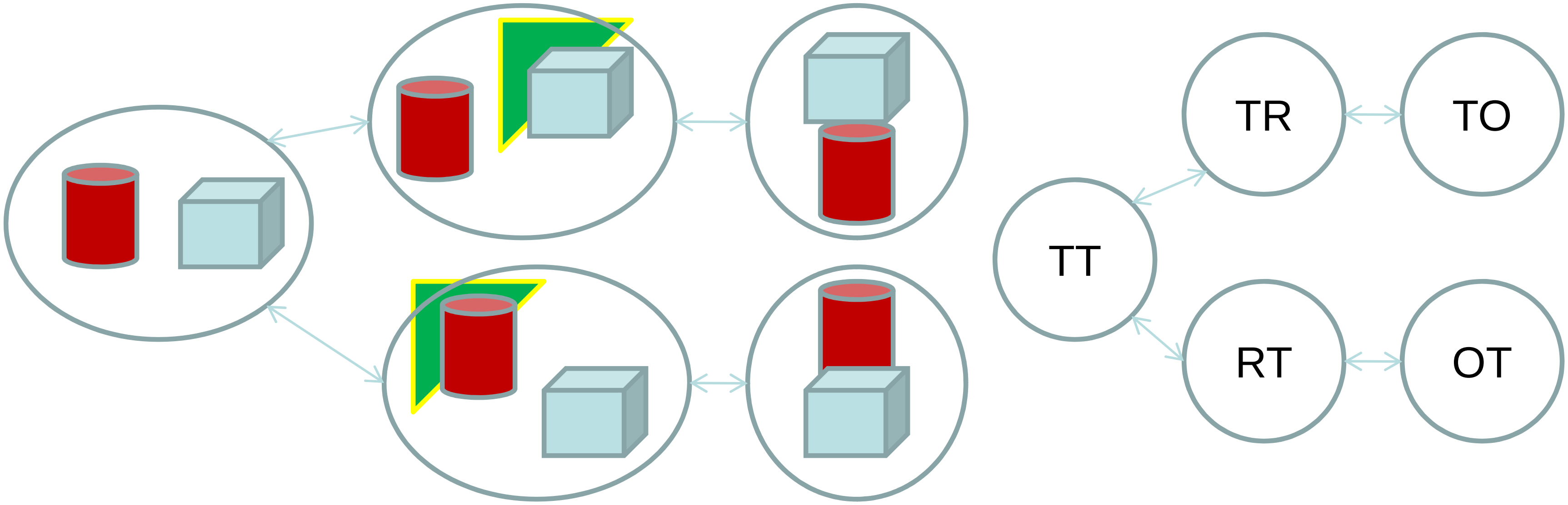
An intuitive interface should allow the interaction to be as free as possible

Even if the user does not follow a strict vocabulary, can the robot still make use of such extra signals?

Learn the meaning of new vocabulary

Feedback

	Signs	Meanings
Known	up	$\uparrow$
	down	$\downarrow$
	left	$\leftarrow$
	right	$\rightarrow$
	$\emptyset$	CORRECT/WRONG
	ok	CORRECT
	error	WRONG
Unknown	good	?
	bad	?
	$\vdots$	?



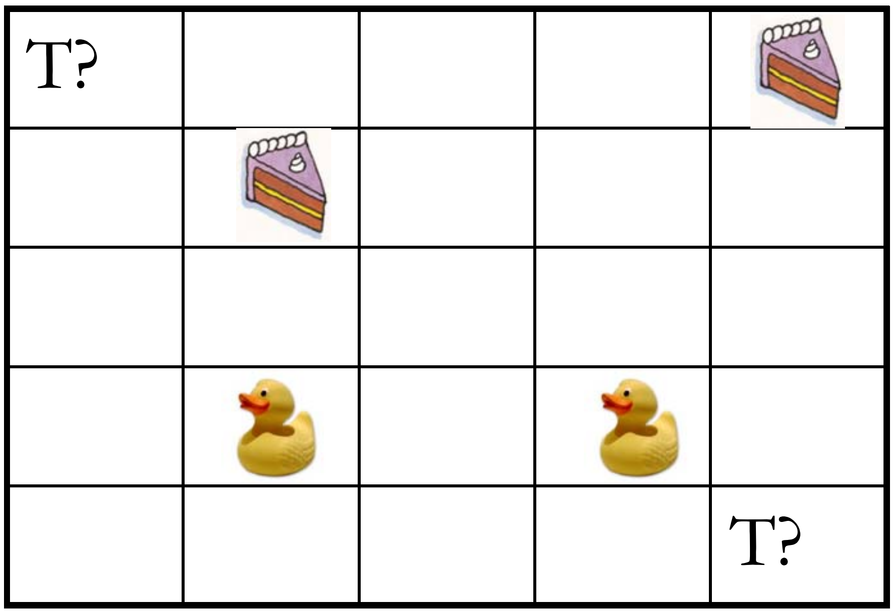
Init State	Action	Next State	Feedback		F1 (_/A)		F2 (A/_)	
					OT	TO	OT	TO
TT	Grasp1	RT	_		+	-	-	+
RT	Grasp2	RT	*RelOnObj*		++	-+	--	+-
RT	RelOnObj	OT	_		+++	-+-	--+	+-+
TT	Grasp2	TR	*AgarraVer*	Assuming (F1,OT) *AgarraVer* means Grasp1				

ALGORITHM FOR THE JOINT ESTIMATION OF THE TASK, FEEDBACK AND GUIDANCE MODELS. IT COMBINES THREE PARTICLE FILTERS TO APPROXIMATE THE POSTERIOR DISTRIBUTION OF THE THREE VARIABLES.

- Select number of samples n_r, n_g and n_m
- Sample n_r reward vectors
- Sample n_g guidance parameters
- Sample n_m meanings tables

1) Sample state x
2) Choose and execute action a
3) Observe guidance g
4) Sample feedback from f_t $p(f|g_t)$
5) Find best feedback parameters $M = argmax_i w_f^{(i)}$
6) $w_r^{(i)} \leftarrow p(f_t|A_t, R_t^i, M)p(A_t|R_t)w_r^{(i)}$
7) Resample reward particles
8) Find best reward parameters $r^* = argmax_i w_r^{(i)}$
9) $w_f^{(i)} \leftarrow p(f_t|A_t, r^*, M_t)p(A_t|r^*)w_f^{(i)}$
10) Resample feedback model
11) $w_g^{(i)} \leftarrow \sum_i p(g_t|f_t)w_g^{(i)}$
12) Resample guidance model
13) goto 1

Scenario



Actions:

Up, Down, Left, Right, Pick, Release

Task consist in finding:

what object to pick and

where to take it

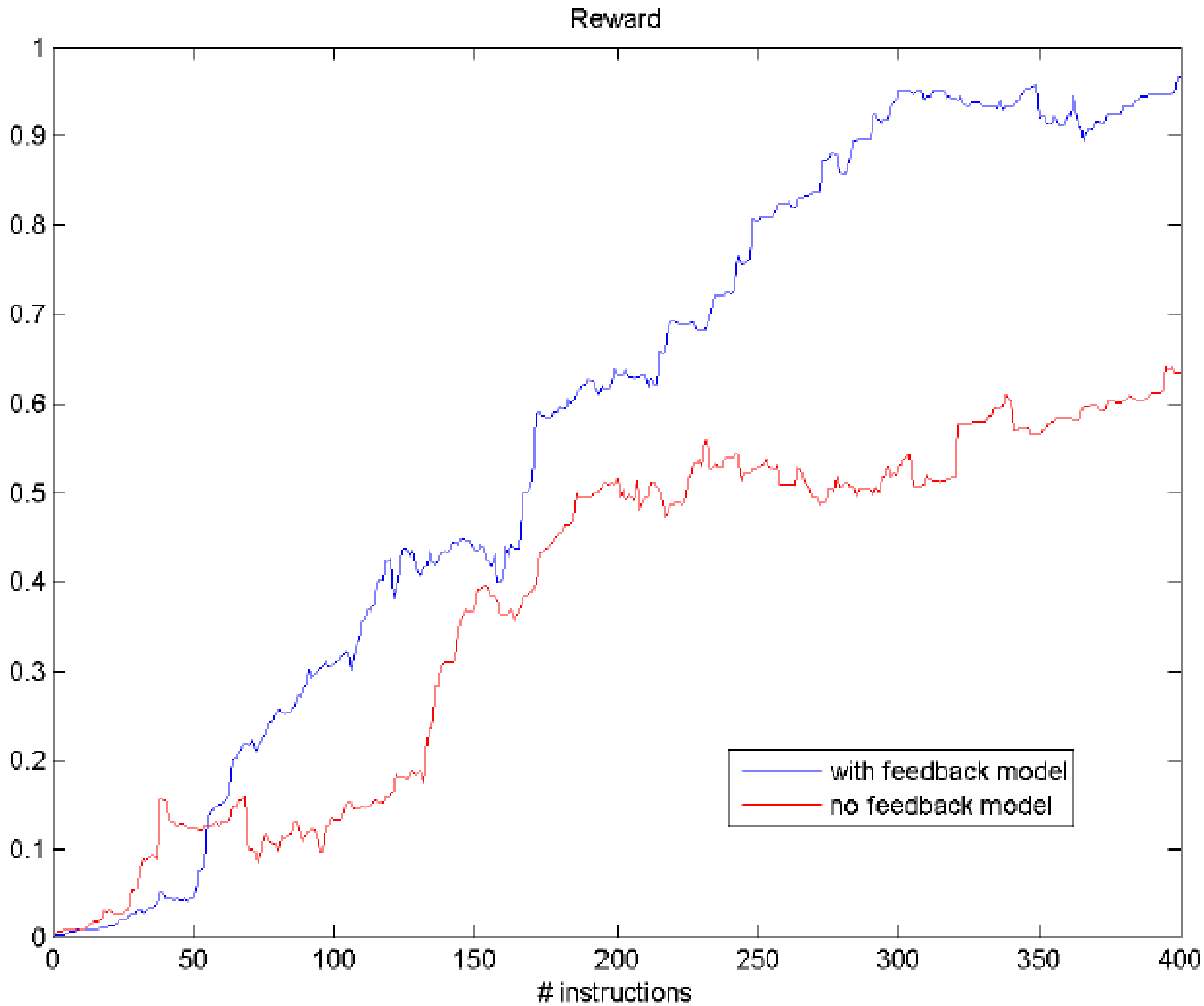
Robot tries an action, including none

User provides feedback

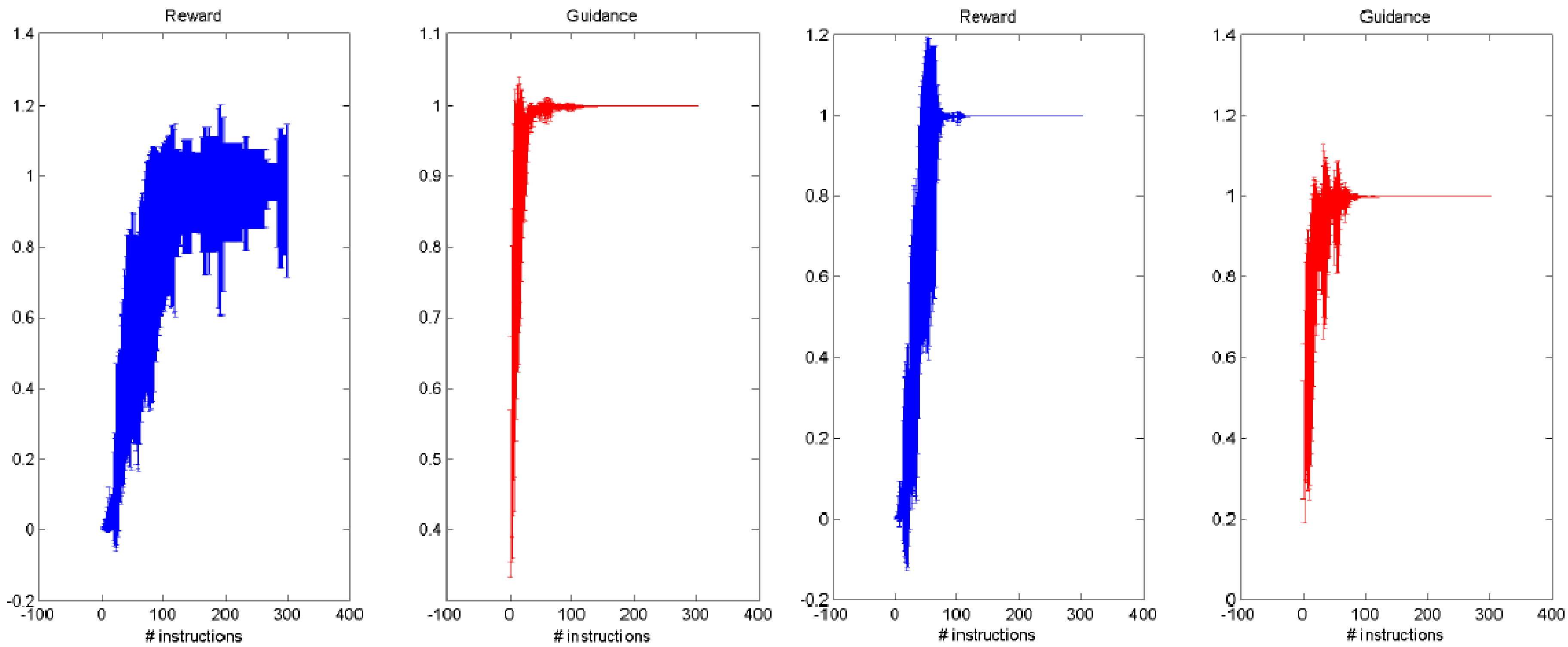
8 known symbols, 8 unknown ones

Robot must learn the task goal, how the user provides feedback and some unknown signs

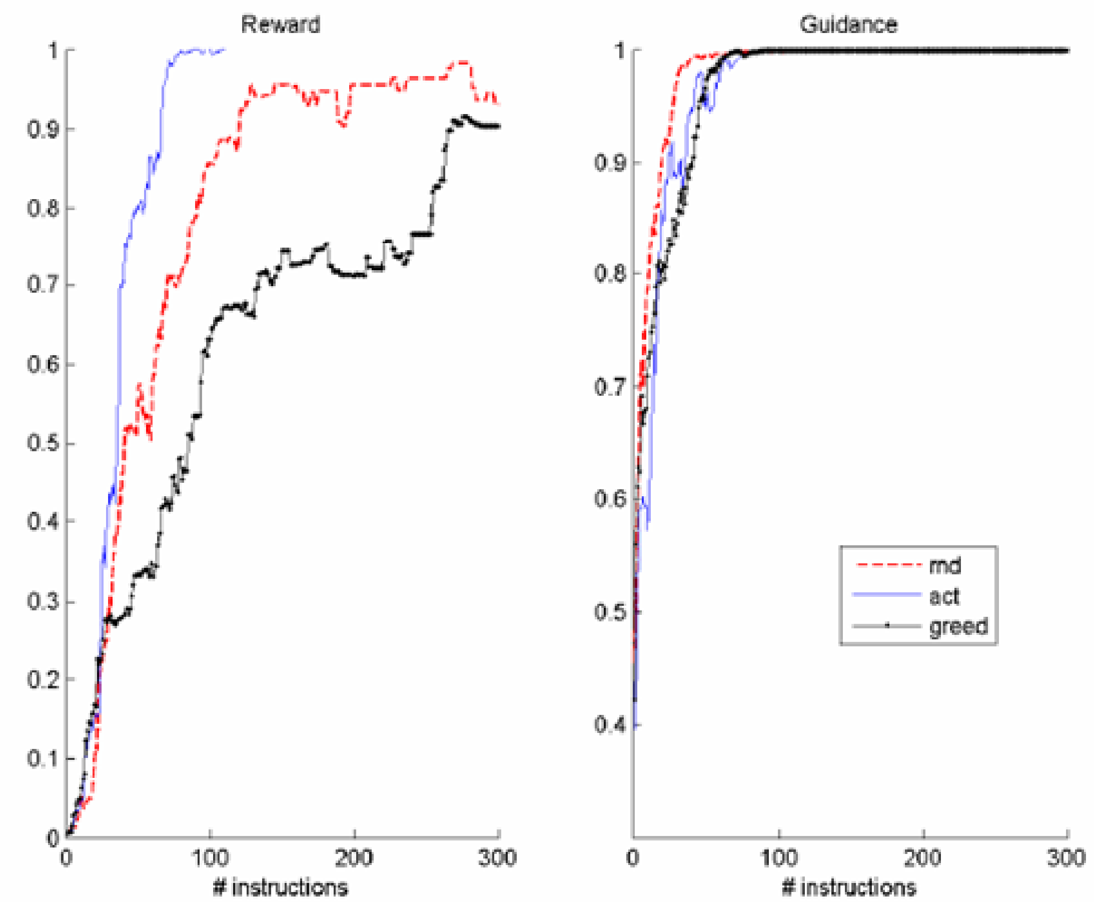
Protocol Uncertainty



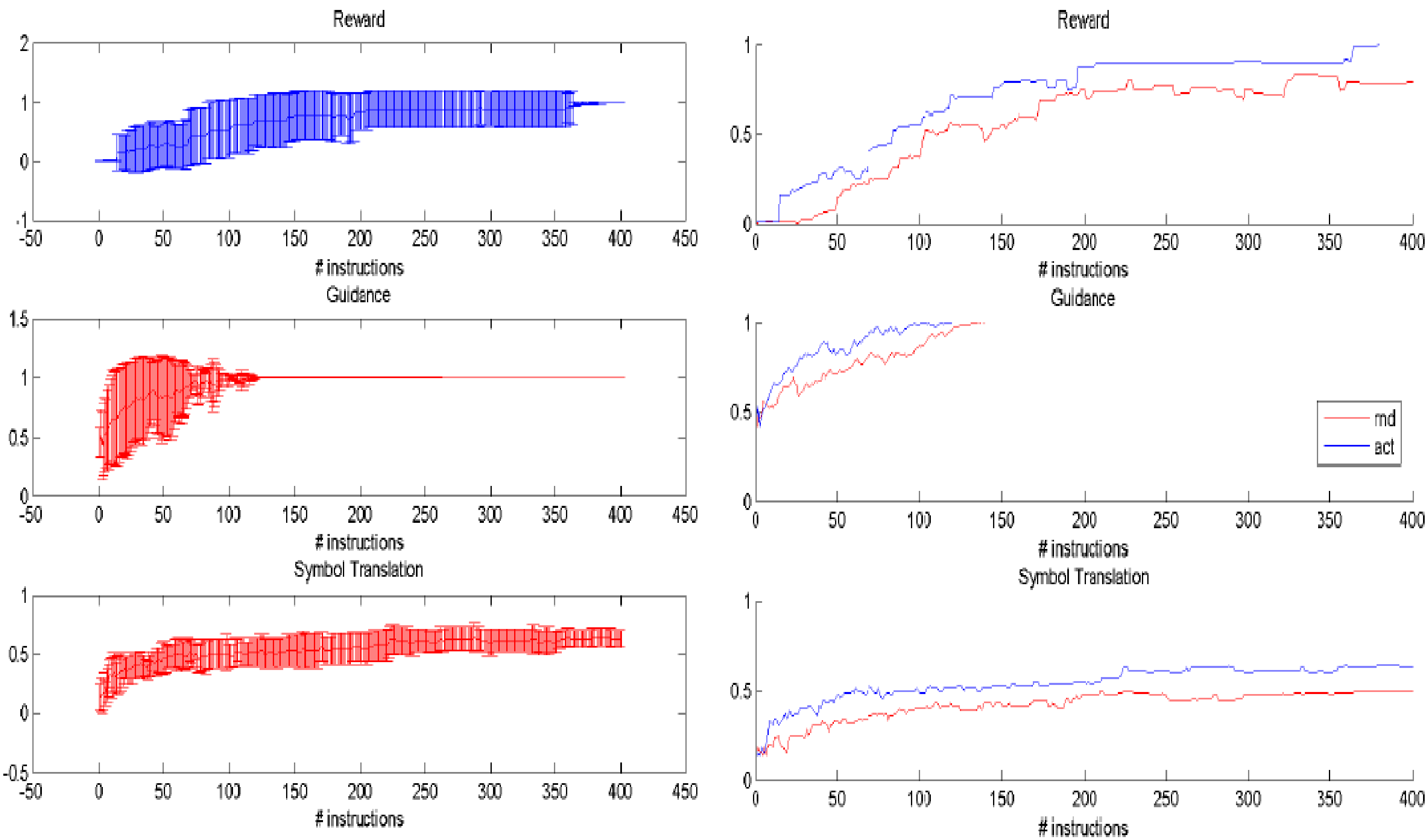
Unknown Task and Feedback



Query Strategies



Unknown Task/Feedback/Utterances



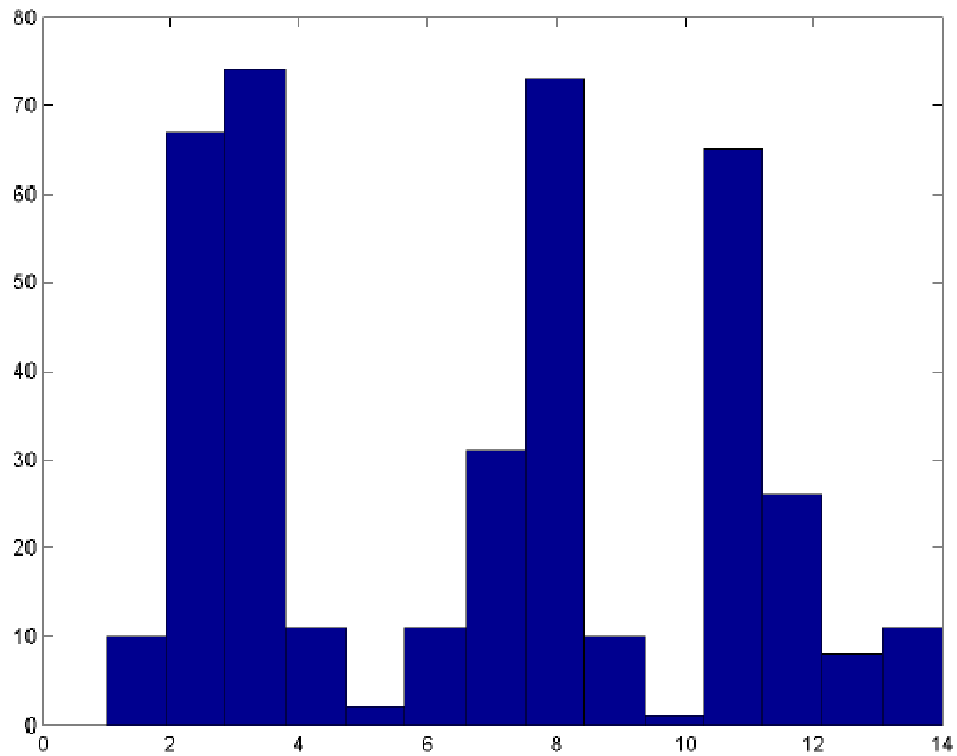
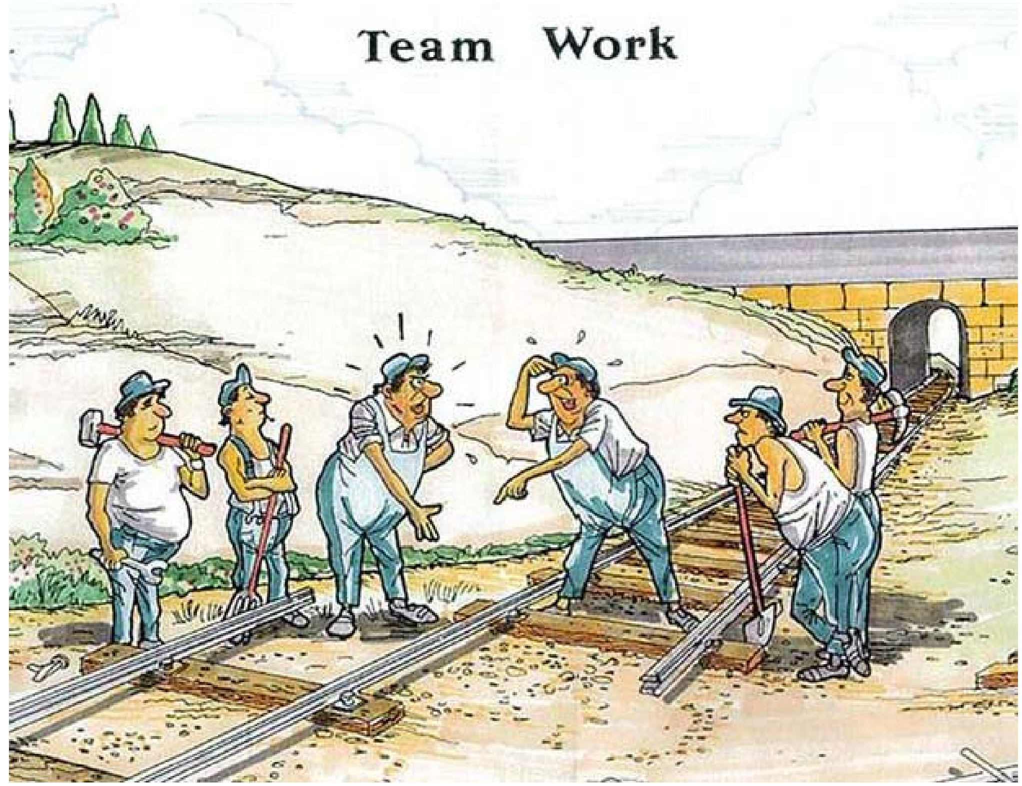


Fig. 6. Histogram of observed guidance symbols

Outline

- *Interactive Learning*
  - *Ambiguous Protocols*
  - *Ambiguous Signals*
  - *Active Learning*

- Inverse Reinforcement Learning for Team Coordination
  - IRL in distributed multi-agent scenarios

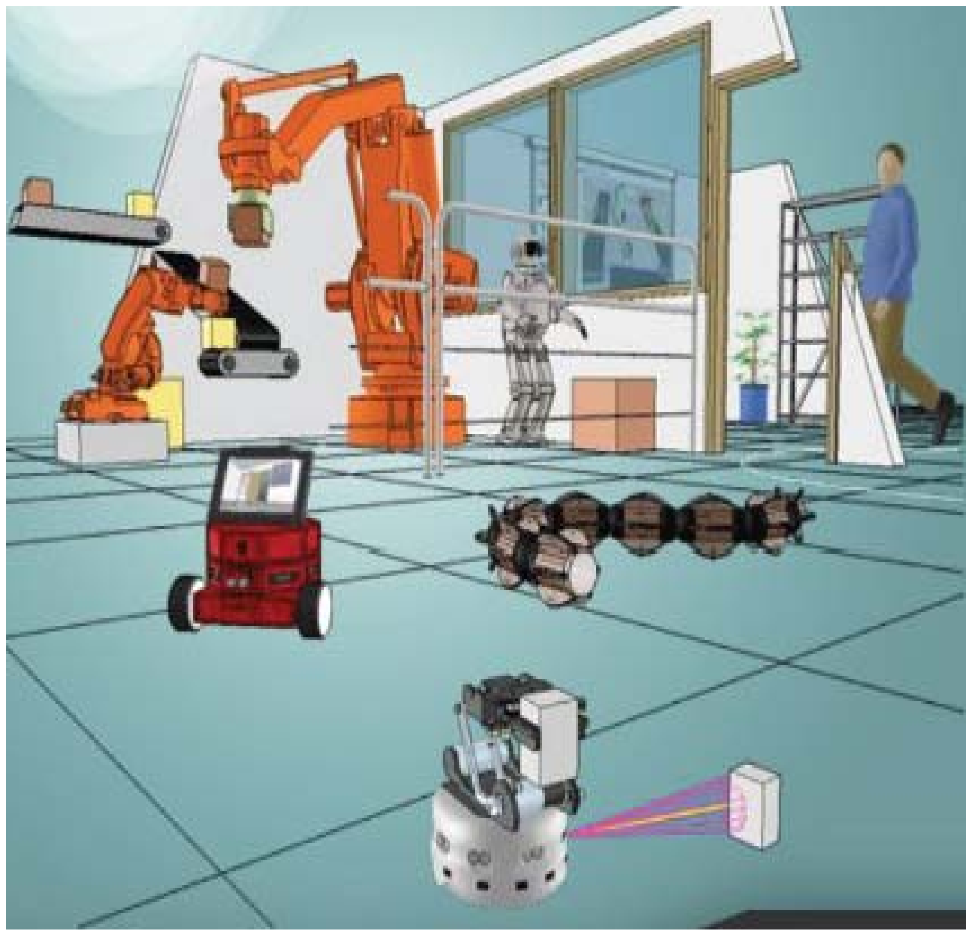


Coordinated Inverse Reinforcement Learning

Coordinated Inverse Reinforcement Learning
Manuel Lopes, Jonathan Sprauler. (under review)

Motivation

- Efficient Human-Robot Collaboration
- Creation of Adhoc teams [Barrett et al., 2011]



Previous Works

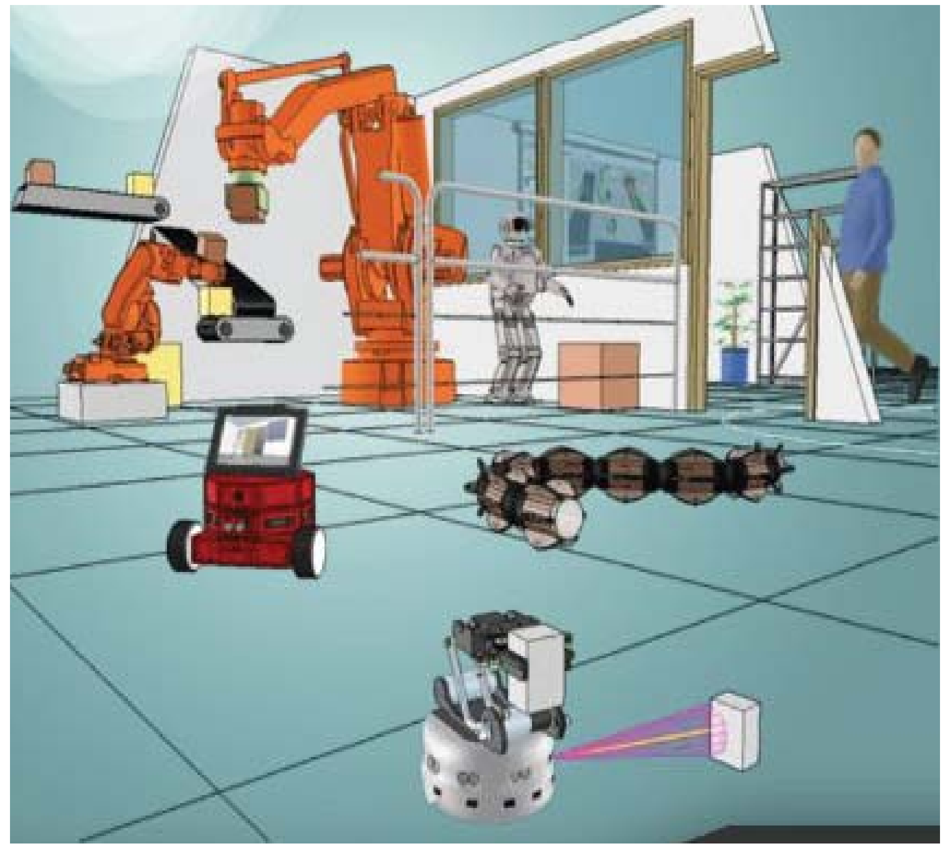
- Multiple mentors, single learner
  - used to improve a model-based reinforcement learning [Price and Boutilier, 1999, 2003].
  - [Shon et al., 2007], which mentor to ask for information as they might not be always helpful, using side payments
  - [Babes et al., 2011], multi demonstrator with different tasks
- [Chernova and Veloso, 2008], a single user teaches a team of robots in a loosely-coordinated task. The user teaches when to ask for further information.
- Truly cooperative task [Martins and Demiris, 2010b] studies the role of communication between mentors.
- **[Natarajan et al., 2010] IRL in MAS**. A central controller and separate tasks.
- **[Waugh et al., 2011] IRL in matrix games**.

How to learn a (*distributed*) team behavior from demonstration?

Difficult correspondence problems:

- All the ones from single-agent
- Heterogeneous or Homogeneous agents?
- Same number?
- What is the minimum required?
- …

Strongly vs Weakly Connected



Problems

- Number of agents might change from demonstration to learning
- Who corresponds to whom?
- Is the communication observed? Used for learning?

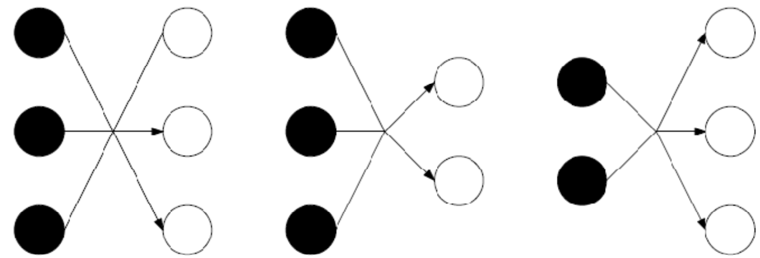


Figure 2: Different number of demonstrator and imitators

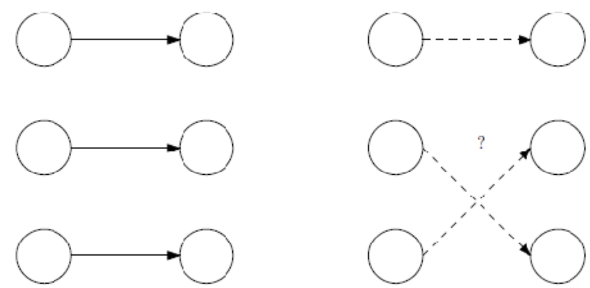


Figure 1: Equal number of imitators and learner

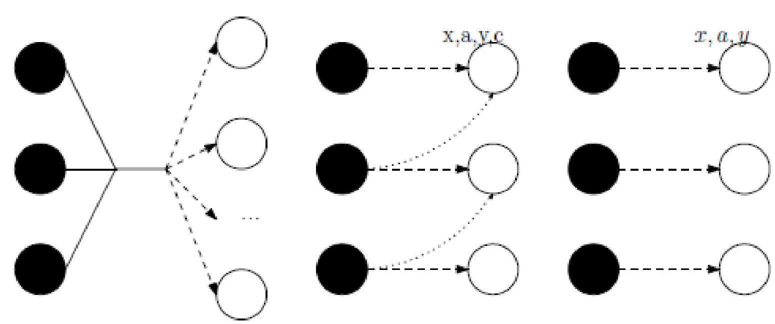
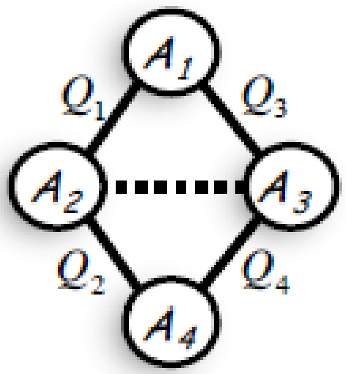


Figure 3: Learning styles

Coordinated RL [Guestrin et al., 2002]

For each agent the Q function does not depend on all the states and all actions (factored MDP)



Coordination graph for a 4-agent

$$Q = \sum_j Q_j$$

$$Observable[Q_j] = \{X_i \in \mathbf{X} \mid X_i \in \text{Scope}[Q_j]\};$$

$$Relevant[Q_j] = \{A_i \in \mathbf{A} \mid A_i \in \text{Scope}[Q_j]\}.$$

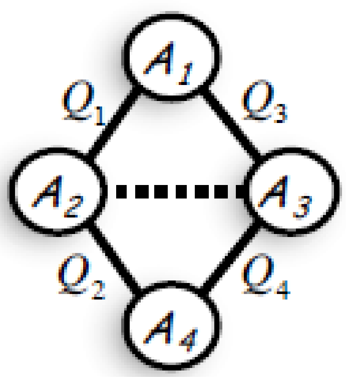
Alternatives: [Clouse, 1996] [Littman, 2001][Lauer and Riedmiller, 2000] [Wang and Sandholm, 2003]

Factored Q-Functions

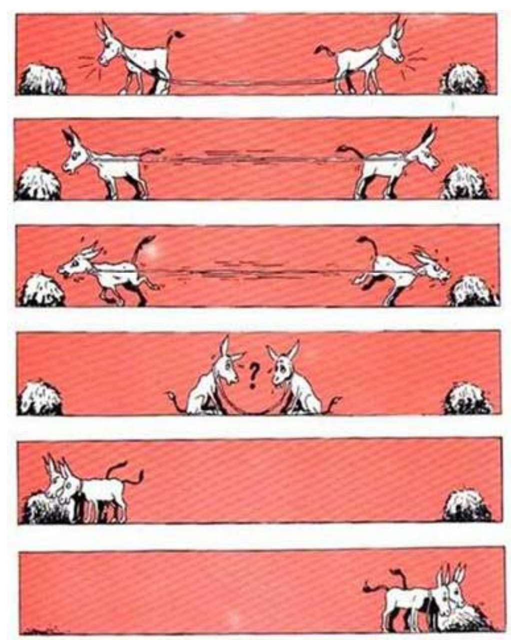
$$Q = Q_1(a_1, a_2) + Q_2(a_2, a_4) + Q_3(a_1, a_3) + Q_4(a_3, a_4)$$

$$\max_{a_1,a_2,a_3,a_4} Q_1(a_1, a_2)+Q_2(a_2, a_4)+Q_3(a_1, a_3)+Q_4(a_3, a_4).$$

$$\max_{a_1,a_2,a_3} Q_1(a_1, a_2)+Q_3(a_1, a_3)+\max_{a_4}[Q_2(a_2, a_4)+Q_4(a_3, a_4)].$$



Coordination graph for a 4-agent problem.



Factored Gradient IRL - Likelihood

$$\begin{aligned} l(x,a) &= \frac{e^{\beta Q^T(x,a)}}{\sum_b e^{\beta Q^T(x,b)}} \\ &= \frac{e^{\beta \sum_i Q^i(x_i,a_i)}}{\sum_b e^{\beta \sum_i Q^i(x_i,b_i)}} \\ &\approx \prod_i \frac{e^{\beta Q^i(x,a)}}{\sum_b e^{\beta Q^i(x,b)}} \end{aligned}$$

Factored Gradient IRL - Gradient

$$
\begin{aligned}
\mathcal{L} &= \prod_i l(x_i, a_i) \\
log\mathcal{L} &= \sum_i log(l(x_i, a_i)) \\
\frac{log\mathcal{L}}{dR} &= \frac{\sum_i log(\pi(x_i, a_i))}{dR} \\
&= \frac{de^{\beta Q^i(x,a)}}{dR} - \frac{dlog \sum_b e^{\beta Q^i(x,b)}}{dR} \\
&= \beta e^{\beta Q^i} \frac{dQ^i_a}{dR} - \frac{\sum_b \beta e^{\beta Q^i} \frac{dQ^i_b}{dR}}{\sum_b e^{\beta Q^i(x,b)}} \\
&= \beta e^{\beta Q^i} \frac{dQ^i_a}{dR} - \sum_b l(x_i, b_i) \frac{dQ^i_b}{dR}
\end{aligned}
$$

Coordinated Inverse Reinforcement Learning
Manuel Lopes, Jonathan Sprauler. (under review)

Scenario

Flat Model

State space:

$(N+M-1+P)^M \times N^P$

State-Action combinations:

$(N+M-1+P)^M \times N^P \times A^P$

With Factorization

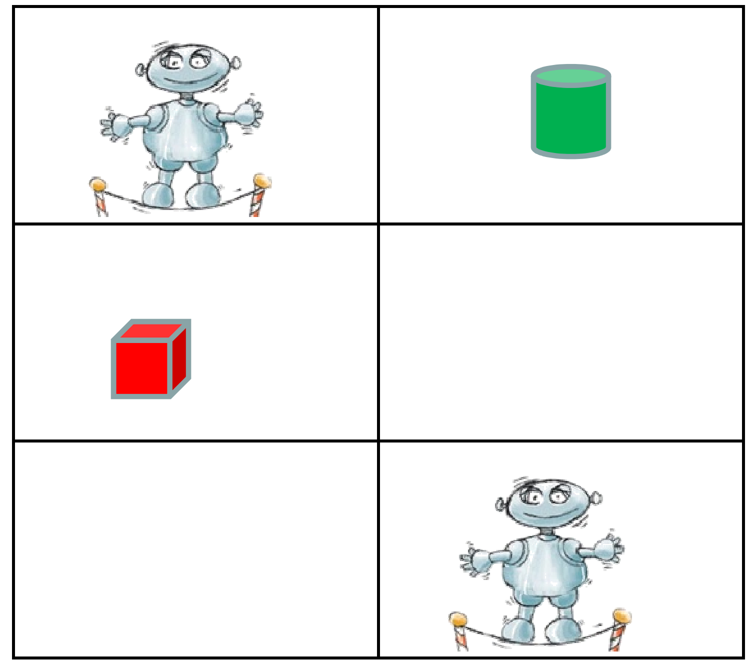
Robots do not interact directly

$(N+M-1+P)^M \times N^1$ per agent

Objects do not interact

$(N+P) \times M^1 \times N^P$ per agent

Robots do not interact directly and
Objects do not interact
$(N+P) \times M \times N$



M objects = 2
P robots = 3
N locations = 6
A actions = 8

Results (preliminary)

- Full Independent Learning
  - A reward function is learned per agent, learning is made independent (a GradIRL per agent)

- Simultaneous Learning
  - The reward function is the same for all agents, ignoring the other agent (simultaneous GradIRLs)

- Coordinated Learning
  - A single reward function, learned using the coordinated gradient IRL

Results (preliminary)

Full Independent and Simultaneous

- Learned policy only works if the other team members follow same policy
- Very little generalization to non-demonstrated states

Coordinated Learning

- Learned policy more efficient than demonstration
- Learned policy generalizes to more non-demonstrated states
- Possibility of changing number of agents
- …

	simil X0, NI=2	Diff X0, NI=2	simil X0, NI=1	diff X0, NI=1
Ind	1	2, ∞	∞	∞
CoordIRL	0.9	1.2	2	2

Conclusions/Future

- Experimental results show active sampling in IRL can help decrease number of demonstrated samples
- Prior knowledge (about reward parameterization) impacts usefulness of active IRL, Experimental results indicate that active is not worse than random
- It can even work with weakly specified protocols
- We can learn the task, the feedback and (some) guidance symbols simultaneously

- Coordination graph and Factorization are known
- All scope variables are observable

Future

- More General Feedback/Guidance Models
- Include More Sources of Information, e.g. Speech prosody
- Learn factored model / coordination structure