# Speeding up a Reinforcement Learning

Tim Brys and Matthew E. Taylor

Vrije Universiteit Brussel

The Intelligent Robot Learning Laboratory

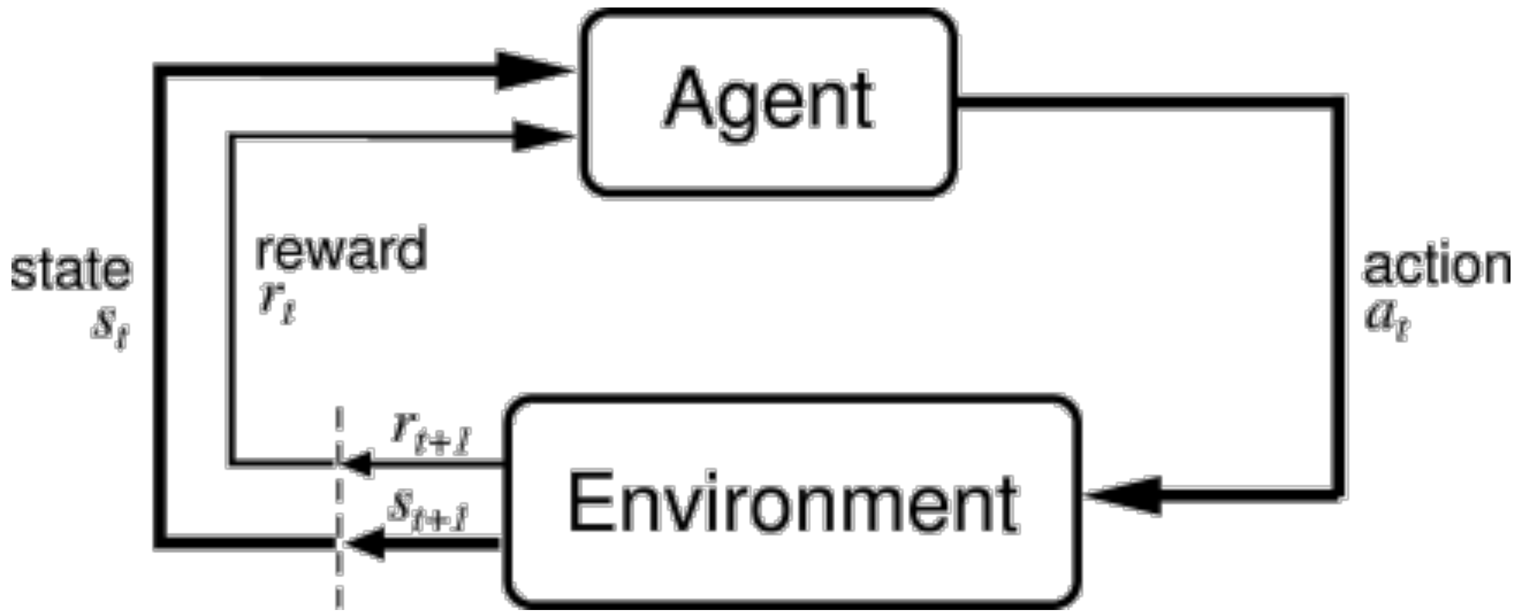
# The aim of the talk

Provide a non-exhaustive overview of techniques that can be used to help an RL agent learn faster

# Part I: Reinforcement Learning

- Learn from **interaction** with the environment
- Feedback is provided through a **reward signal**
  - Think of a dog trainer's cookies
- The agent should learn behaviour that results in the most **reward** collected

# Reinforcement Learning




- Markov Decision Process $\text{MDP M } \langle S, A, T, R \rangle$
    - State space $S$, Action space $A$
    - State transition probabilities $T : S \times A \times S \to \mathbb{R}$
    - A reward function $R : S \times A \times S \to \mathbb{R}$

# Reinforcement Learning

- Goal: learn a policy $\pi : S \times A \to \mathbb{R}$ that, given a state, assigns to each possible action a selection probability such that the expected, accumulated, discounted reward is maximised

$$J^{\pi} \equiv E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right]$$

- The value of an action in a certain state is expressed using the Q-function

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) | s_0 = s, a_0 = a\right\}$$

# RL Sample Complexity

- We want to learn such a policy with as little experiences (samples) in the environment as possible, since these may be costly
- Many RL techniques take a tabula rasa approach, resulting in fully random exploration initially
- Given an often sparse reward signal (e.g., only positive feedback at the goal), the more complex the task, the longer learning takes (more samples are needed)

# The solution

Bias the agent's otherwise purely random exploration using external/prior knowledge

# The solution

- Expert knowledge
    - Reward shaping
- Learning from demonstration
- Transfer learning
- Agents/Humans teaching Agents

# Part II: Expert knowledge

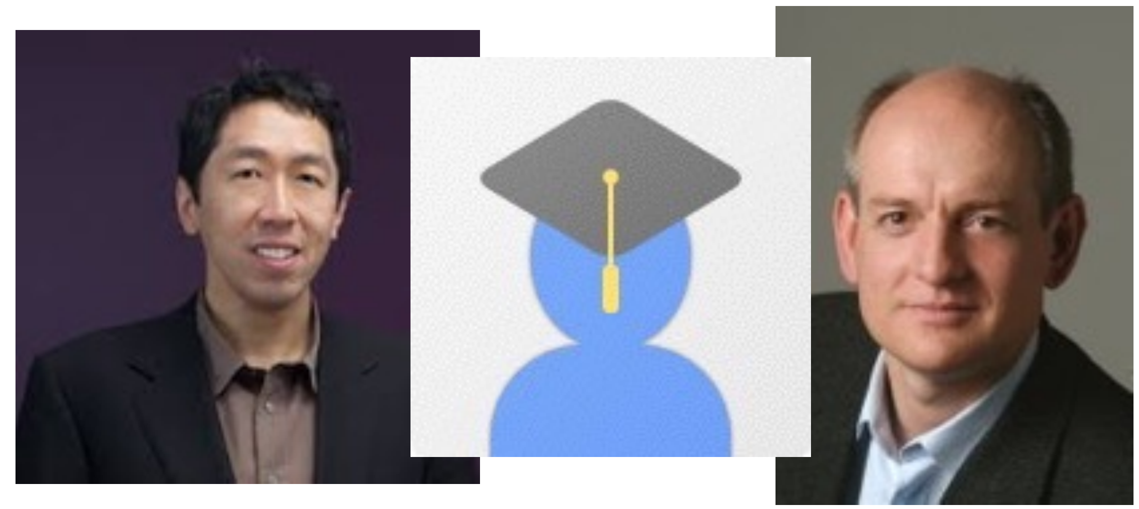
Rules of thumb derived from intuitions of a domain expert

# Part II: Expert knowledge


- Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. ICML
- Wiewiora, E., Cottrell, G., and Elkan, C. (2003). Principled methods for advising reinforcement learning agents. ICML.
- Harutyunyan, A., Devlin, S., Vrancx, P., & Nowé, A. (2015). Expressing Arbitrary Reward Functions as Potential-Based Advice. AAAI
- Brys, T., Harutyunyan, A., Vrancx, P., Taylor, M.E., & Nowé, A. (2014). Multi-Objectivization of Reinforcement Learning Problems by Reward Shaping. IJCNN
- Brys, T., Nowé, A., Kudenko, D., & Taylor, M.E. (2014). Combining Multiple Correlated Reward and Shaping Signals by Measuring Confidence. AAAI
- Harutyunyan, A., Brys, T., Vrancx, P., & Nowé, A. (2015). Multi-Scale Reward Shaping via an Off-Policy Ensemble. AAMAS
- Grześ, M., & Kudenko, D. (2010). Online learning of shaping rewards in reinforcement learning. Neural Networks, 23(4), 541-550.

# Reward Shaping

**Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. ICML**



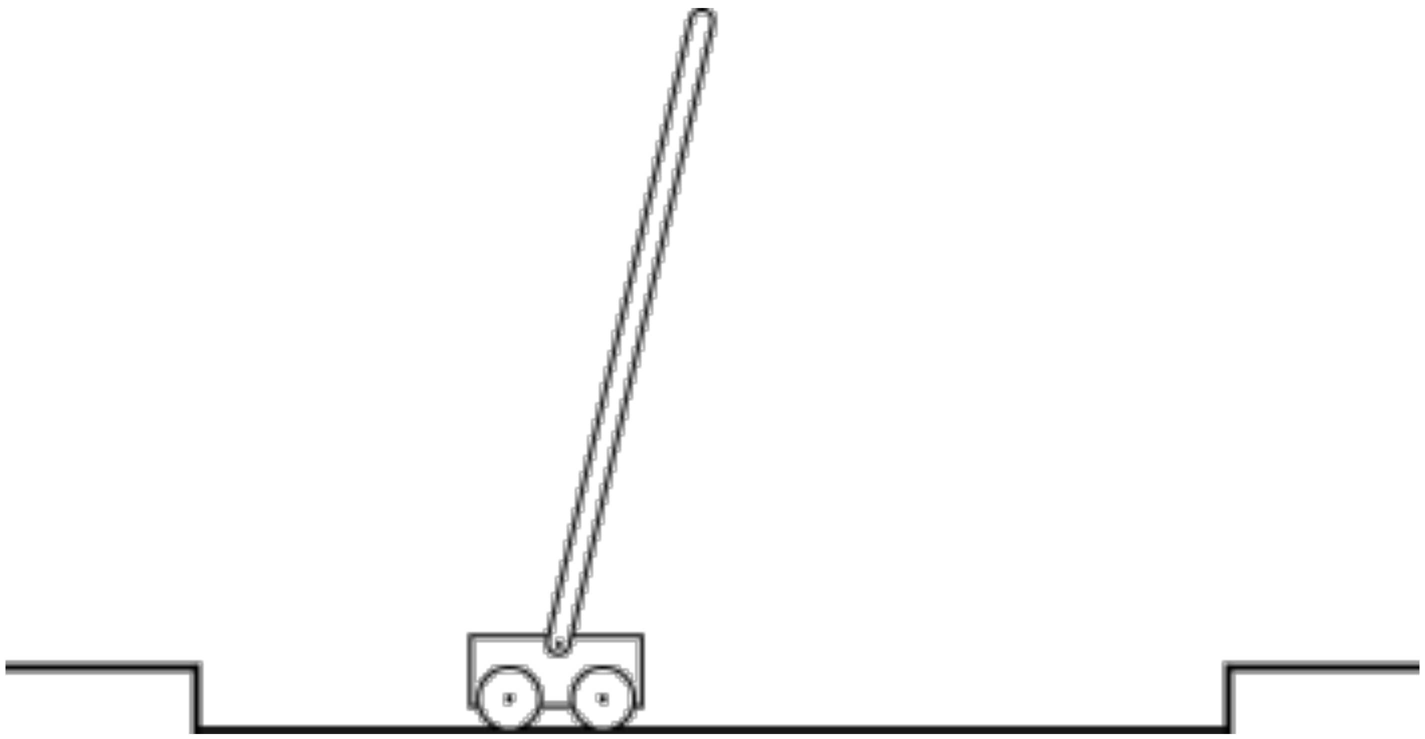
- Way to incorporate heuristic knowledge to speed up learning

$$R \rightarrow R + F$$

- If potential-based, guaranteed to preserve total order over solutions

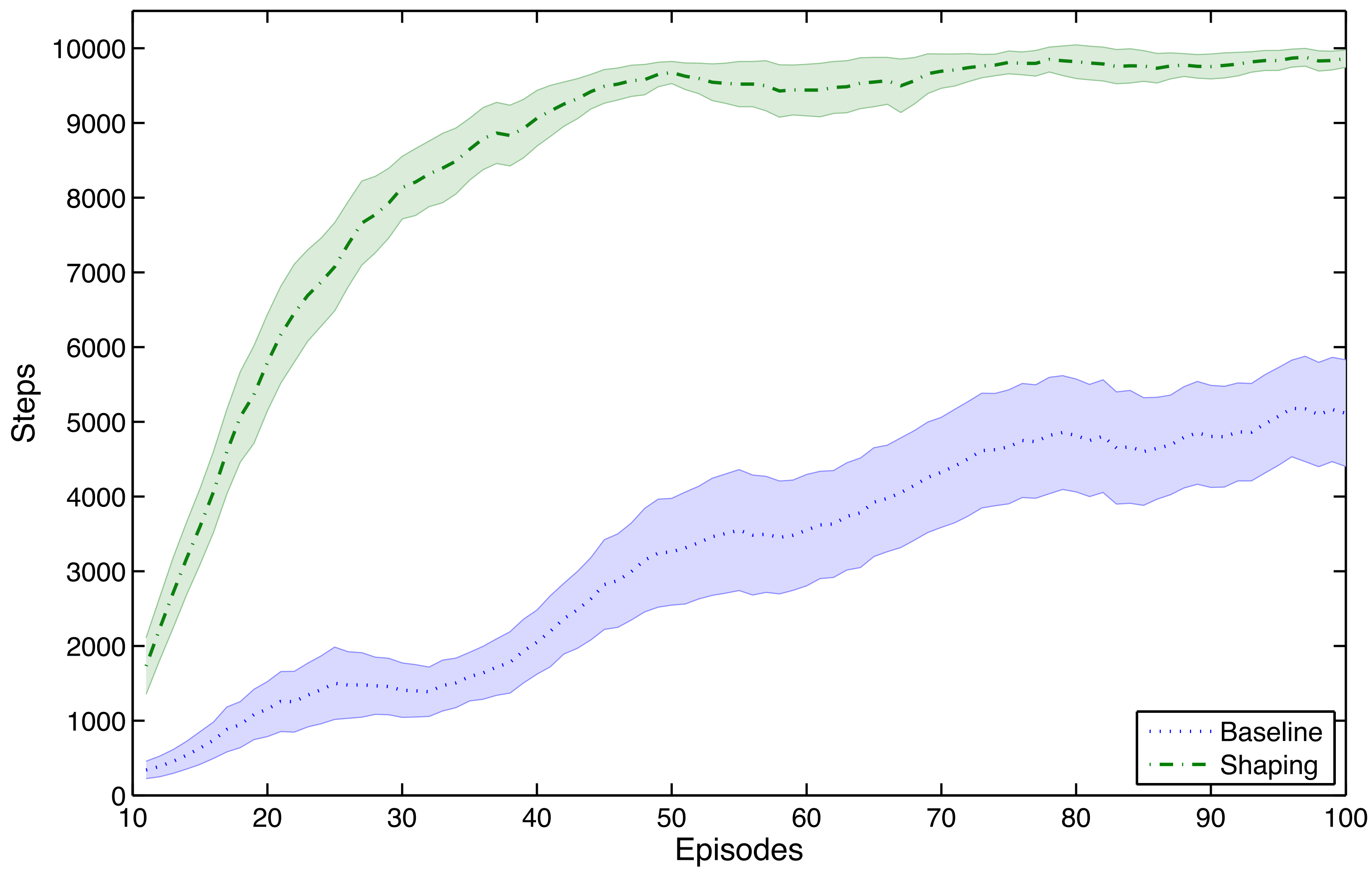
$$F(s, s') = \gamma \Phi(s') - \Phi(s)$$

# Cart Pole



$$S = \{x, \dot{x}, \theta, \dot{\theta}\}$$

# Shaping in Cart Pole




Shaping with the angle of the pole $\Phi(s) = -\theta^2$

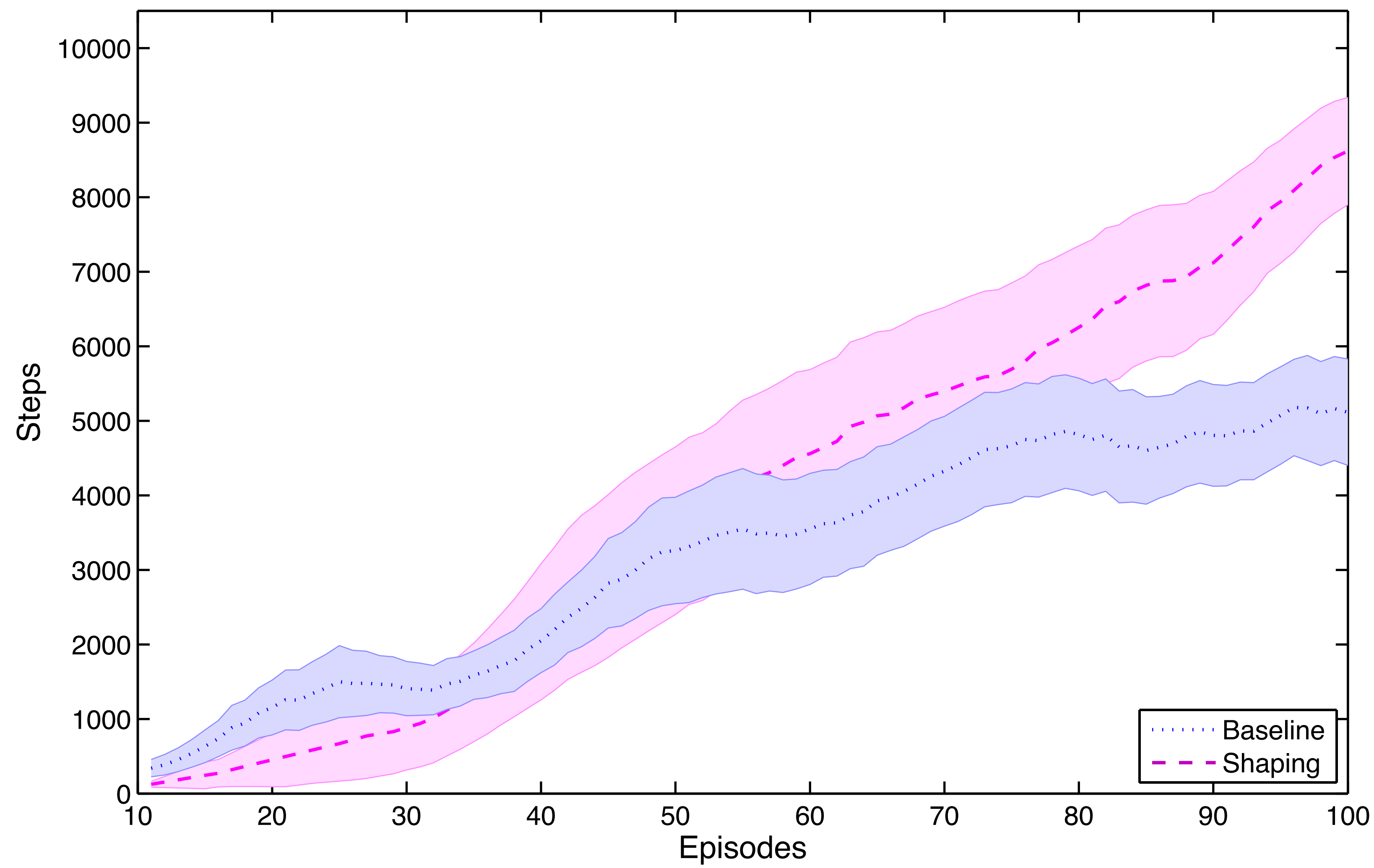
# Reward Shaping

**Wiewiora, E., Cottrell, G., and Elkan, C. (2003). Principled methods for advising reinforcement learning agents. ICML.**

- Shape over states and actions
- Encourage certain behaviour
- Also guaranteed to preserve total order over solutions

$$F(s, a, s', a') = \gamma \Phi(s', a') - \Phi(s, a)$$

# Shaping in Cart Pole




Potential is 1 for moves in the direction the pole is leaning in
0 otherwise

# Unexpected effects of shaping

- Assume $\Phi(s, a) = 1$ and zero elsewhere
- Then $\Phi(s', a') - \Phi(s, a) = -1$
- The desirable behaviour $(s, a)$ is effectively discouraged
- Setting potentials s.t. the desired effect is achieved is difficult

# Arbitrary Reward as Potential-Based Shaping

**Harutyunyan, A., Devlin, S., Vrancx, P., & Nowé, A. (2015). Expressing Arbitrary Reward Functions as Potential-Based Advice. AAAI**

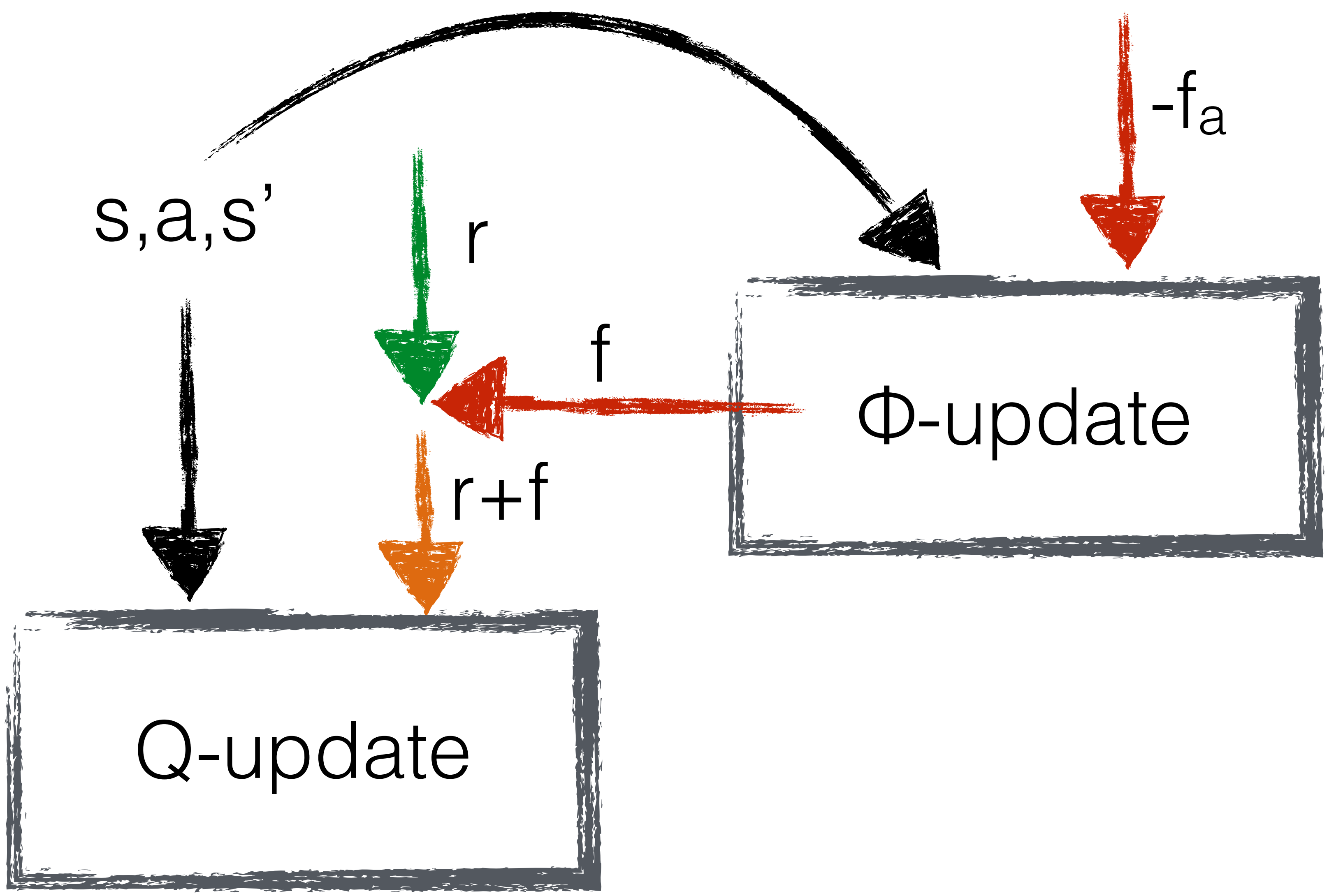



- Instead of defining a potential function $\Phi(s, a)$ , define a reward function $R^\dagger$, so that the actual shaping reward $F \approx R^\dagger$

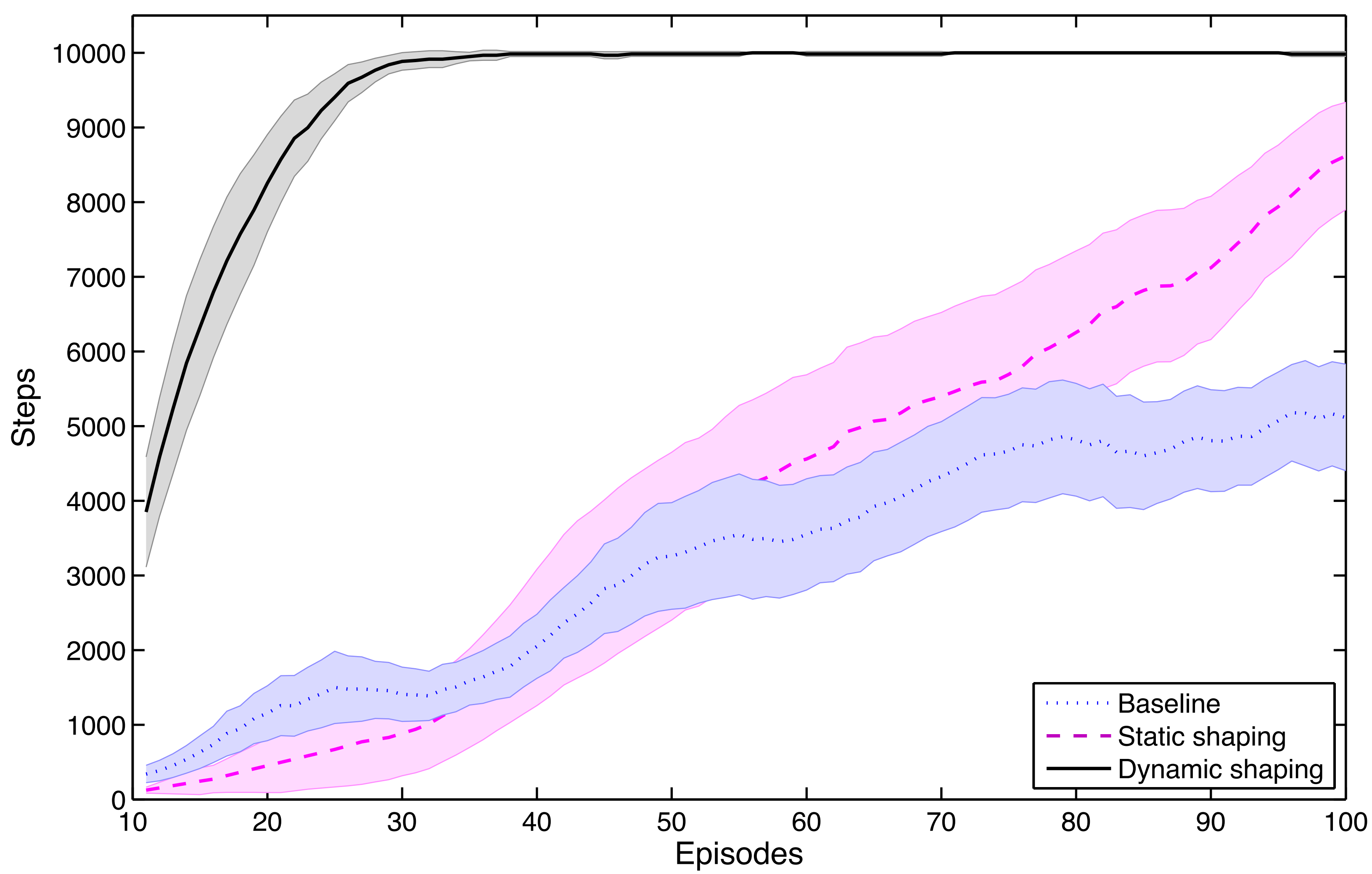
- Learn a second Q-function $Q^\dagger$ based on $R^\dagger$

- Use those Q-values to shape the main reward function $\Phi(s, a) = Q^\dagger(s, a)$

# Arbitrary Reward as Potential-Based Shaping



# Shaping in Cart Pole



# Shaping's hidden tuning problem

- In most papers, lots of pre-tuning
  - Which information to incorporate
  - Parameterization of the shaping (scaling)

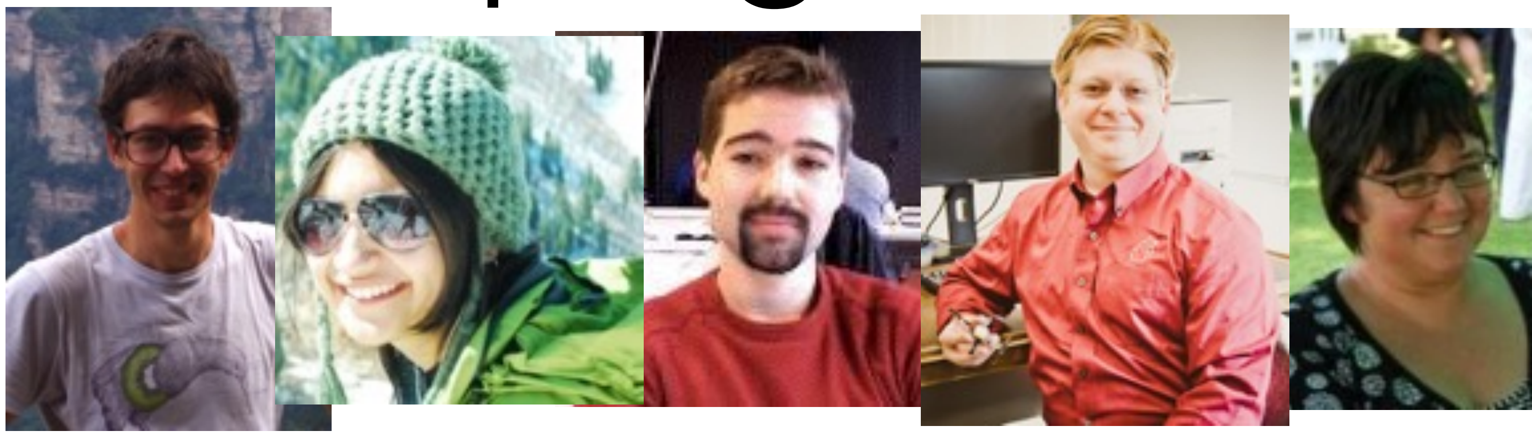
# Shaping's hidden tuning problem

- Instead of wasting a lot of samples during tuning to create a single best shaping, create lots of shapings based on different heuristics and differently parameterised

- Use them in an ensemble

# Multi-Objectivization by Reward Shaping

**Brys, T., Harutyunyan, A., Vrancx, P., Taylor, M.E., & Nowé, A. (2014). Multi-Objectivization of Reinforcement Learning Problems by Reward Shaping. IJCNN**



- Transform MDP into MOMDP

$$\text{MDP M } \langle S, A, T, R \rangle \rightarrow \text{ MOMDP M'} \langle S, A, T, \mathbf{R} \rangle$$

- Add different potential-based reward shaping to each copy of the original reward $\mathbf{R} = [R + F_0, R + F_1, \ldots, R + F_n]$

- We prove that this formulation yields a multi-objective problem with a total order over the solutions

# Ensembles in RL

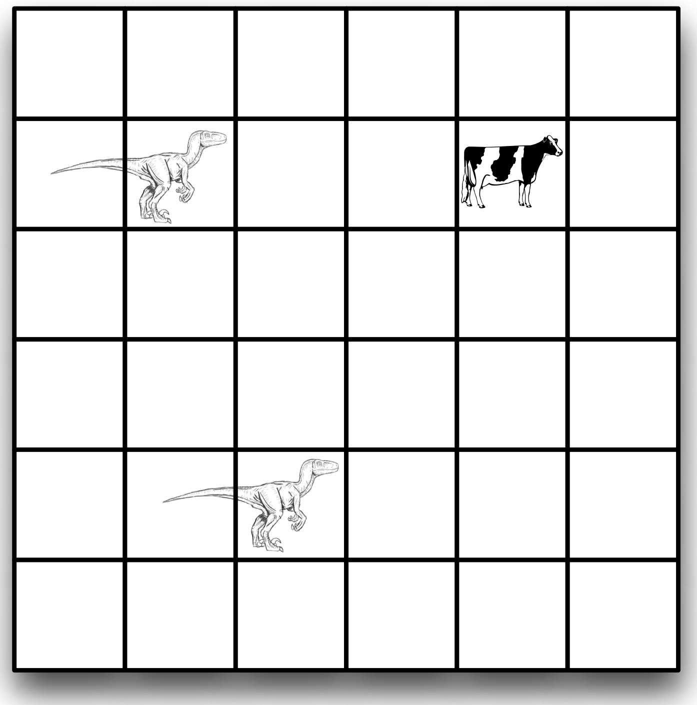
**Wiering, M. A., & van Hasselt, H. (2008). Ensemble algorithms in reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 38(4), 930-936.**

- Ensemble decision (for n decision makers):

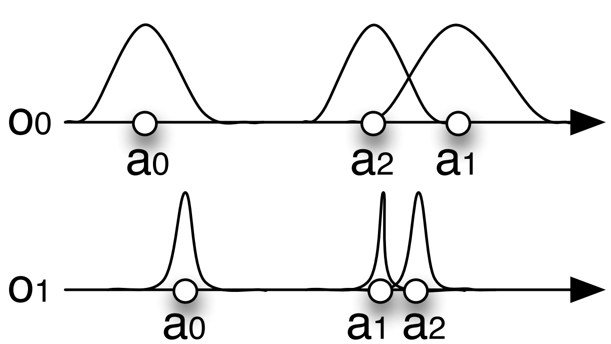
$$\arg\max_a \sum_i^{\mathrm{n}} w_i p_i(s,a)$$

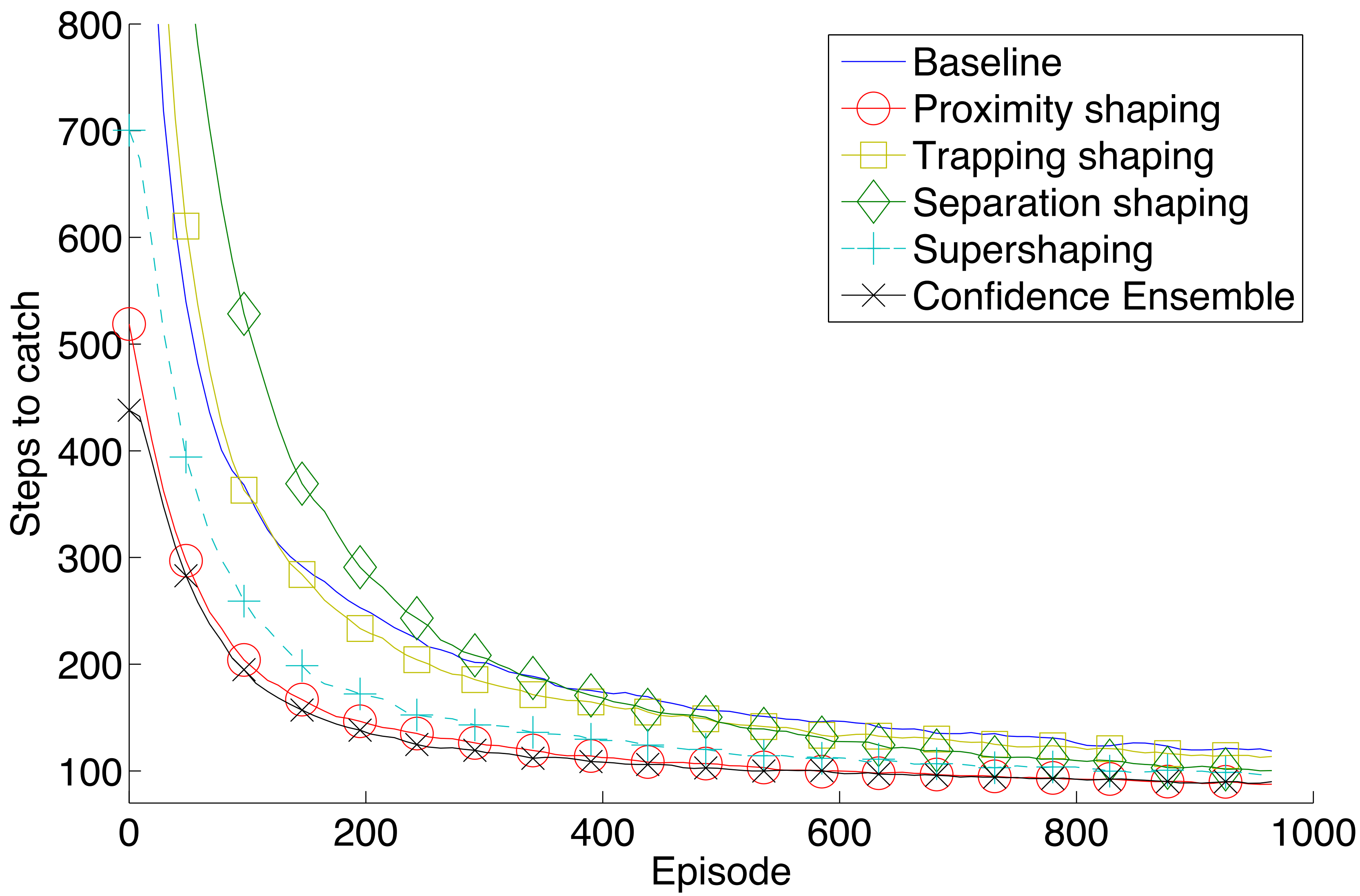
# Confidence Ensemble

Predator-Prey



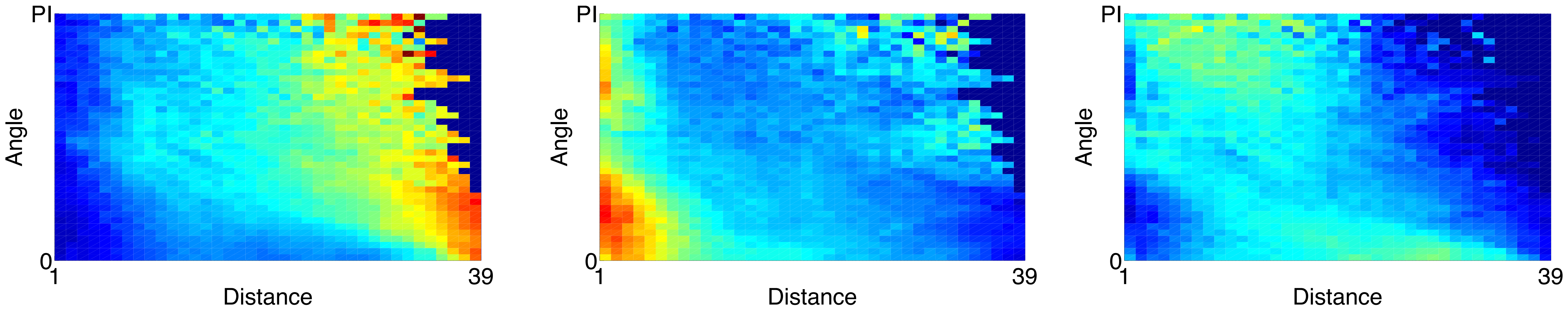
Confidence




**Brys, T., Nowé, A., Kudenko, D., & Taylor, M.E. (2014). Combining Multiple Correlated Reward and Shaping Signals by Measuring Confidence. AAAI**

# Shaping Selection in State Space




Proximity

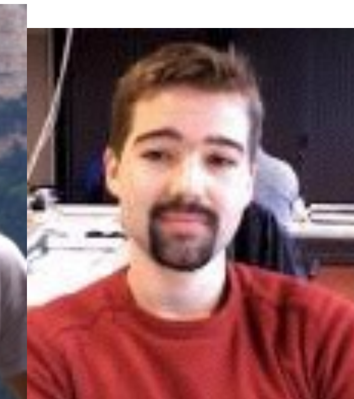
Trapping

Separation

# Choice of Heuristic and Scaling

**Harutyunyan, A., Brys, T., Vrancx, P., & Nowé, A. (2015). Multi-Scale Reward Shaping via an Off-Policy Ensemble. AAMAS**






- For each heuristic, include multiple differently scaled versions in the ensemble

# Learning the Shaping On-line

**Grześ, M., & Kudenko, D. (2010). Online learning of shaping rewards in reinforcement learning. Neural Networks, 23(4), 541-550.**

- Best shaping function is the value-function
- Learn in parallel on a fine- and coarse grained representation
- Shape the fine-grained values with the coarse grained ones

# Part III: Learning from Demonstration

Using (human) demonstrations of a task to learn a policy

# Part III: Learning from Demonstration

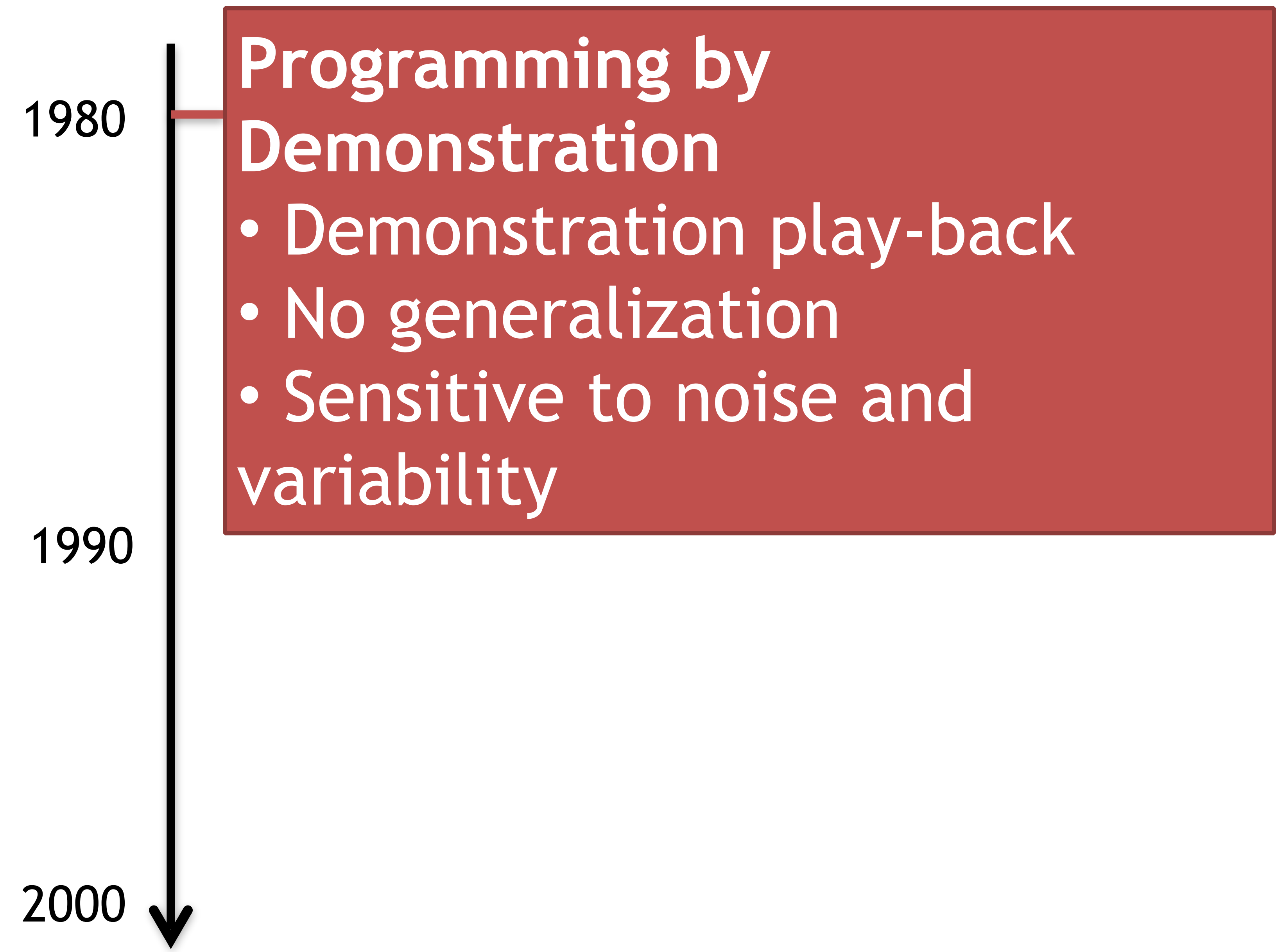
- Background: Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. Robotics and autonomous systems, 57(5), 469-483.

- Smart, W. D., & Kaelbling, L. P. (2002). Effective reinforcement learning for mobile robots. ICRA

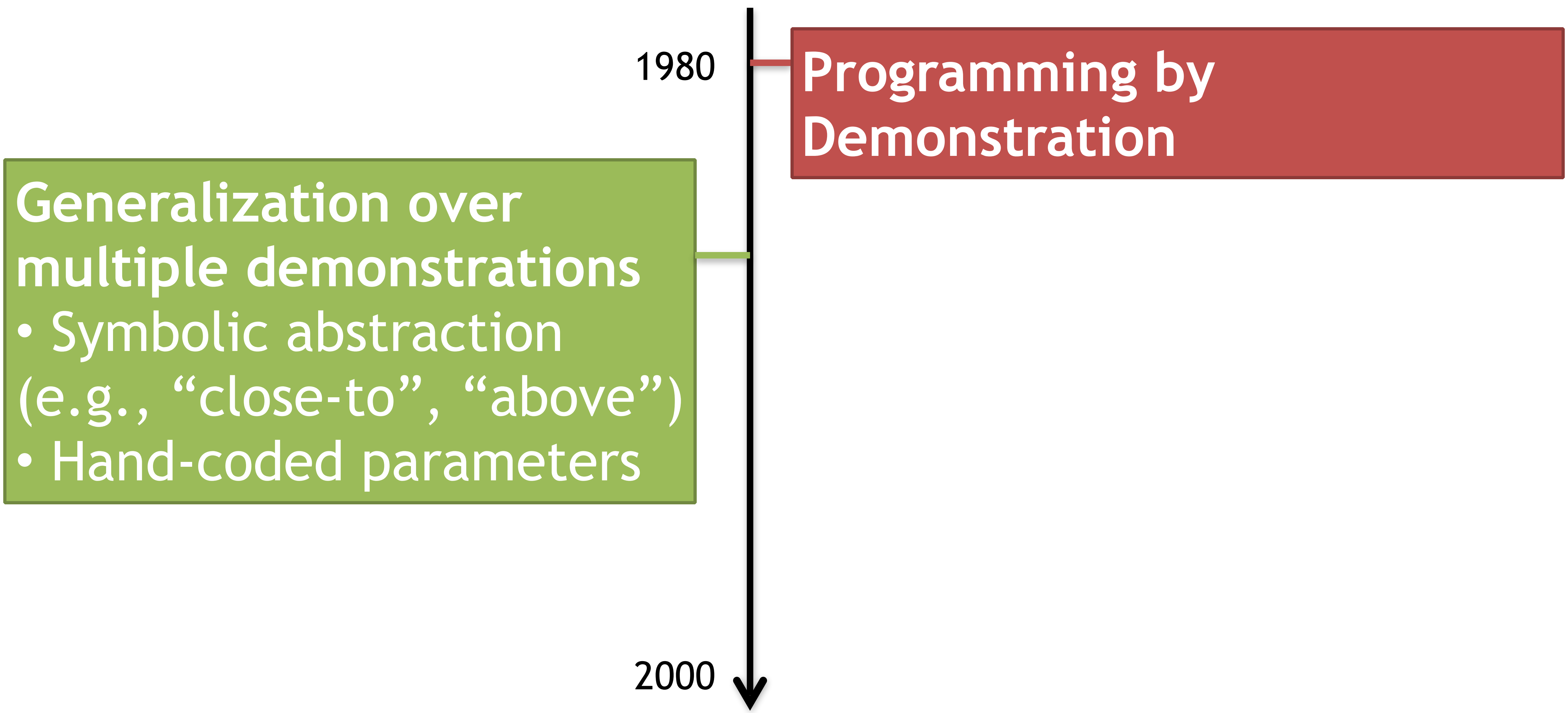
- Taylor, M. E., Suay, H. B., & Chernova, S. (2011). Integrating reinforcement learning with human demonstrations of varying ability. AAMAS

- Brys, T., Harutyunyan A., Suay, H. B., Chernova, S., Taylor, M. E. & Nowé, A, (2015). Reinforcement Learning from Demonstration through Shaping. IJCAI

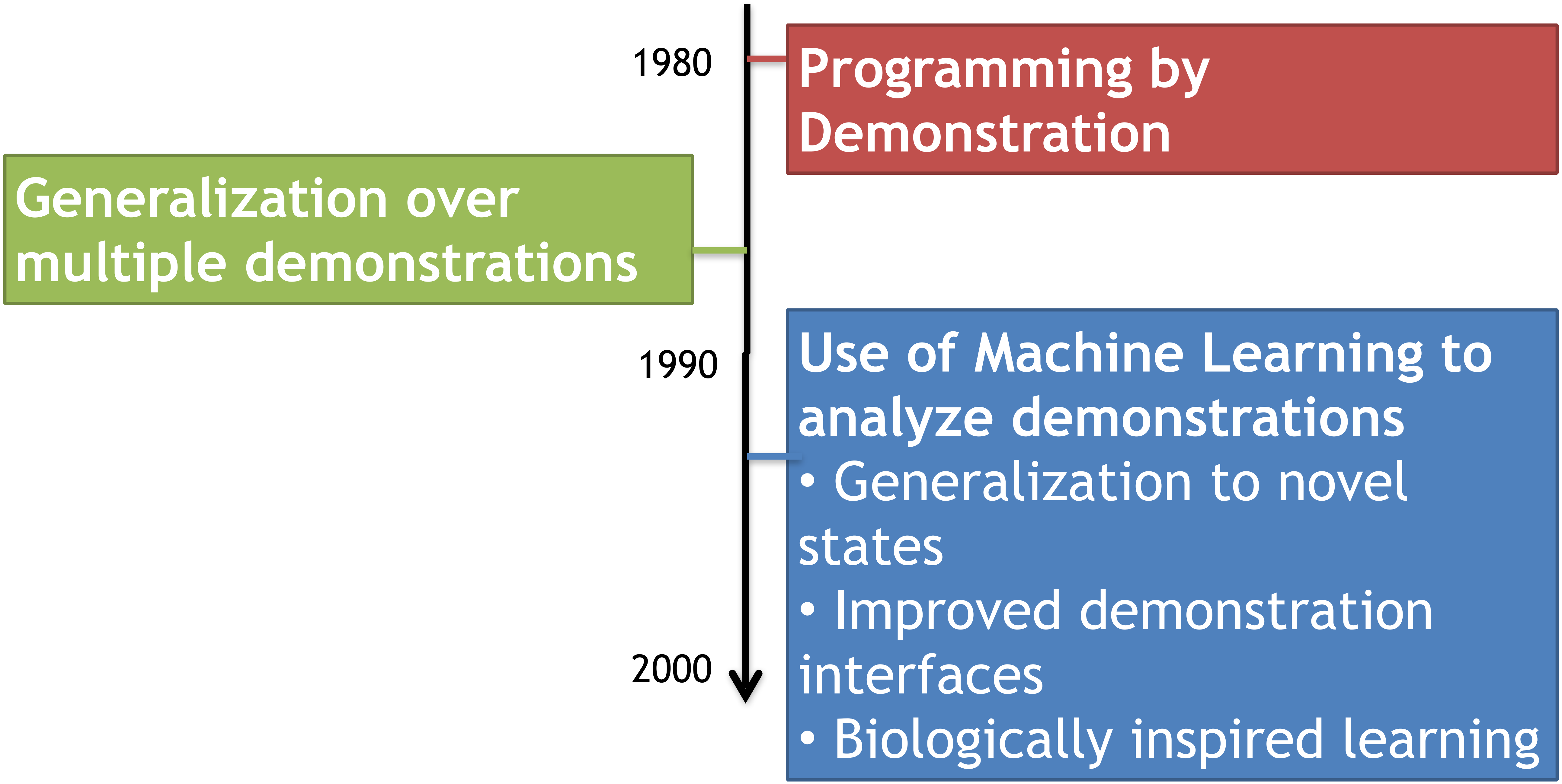
# Learning from Demonstration History



# Learning from Demonstration History



# Learning from Demonstration History



# Learning from Demonstration

**Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. Robotics and autonomous systems, 57(5), 469-483.**



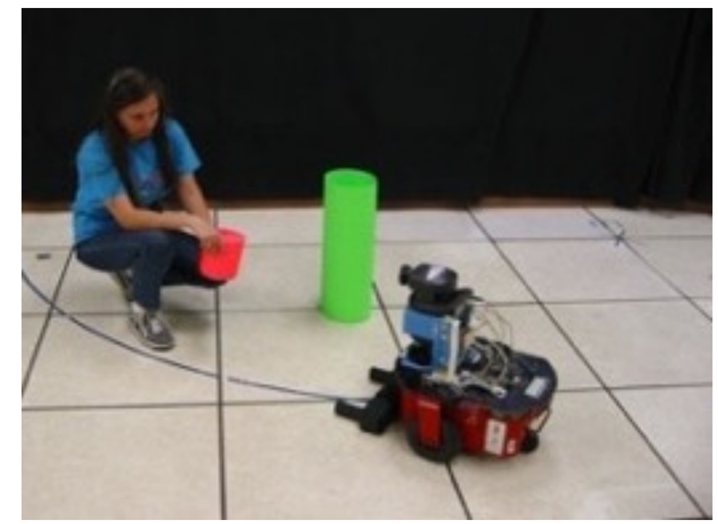

- Generate a policy solely based on demonstrations by abstracting and generalising them
- Demonstrations may
  - be suboptimal
  - not cover the whole state space

# Learning from Demonstration

*Lockerd & Breazeal*

*Grollman & Jenkins*

*Argall, Browning & Veloso*



*Nicolescu & Matarić*

# Reinforcement Learning from Demonstration

- Use demonstrations to speed up/kickstart a reinforcement learning process
- Relying on the ground truth (reward) for learning and using demonstrations as heuristic bias
- Advantages
    - Theoretical guarantees of RL
    - Suboptimality of demonstrations is less a problem
    - High sample complexity of RL is overcome

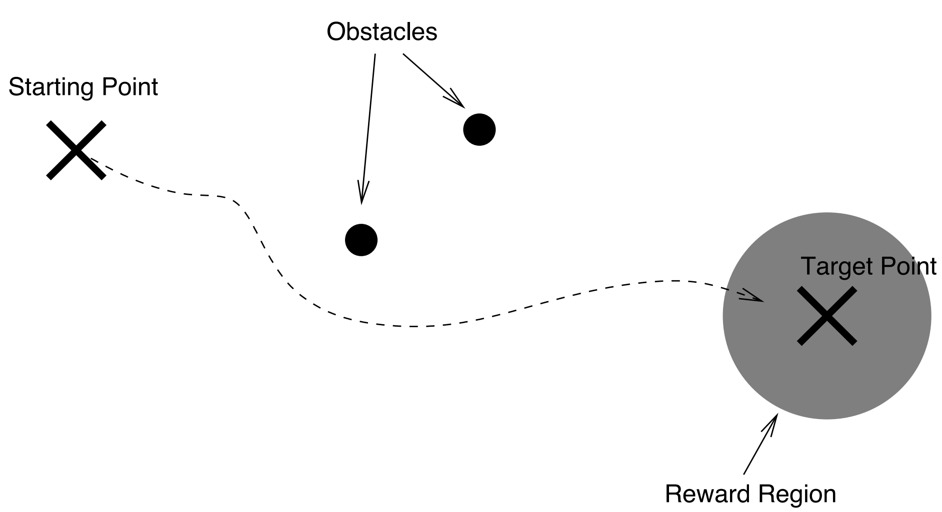
# Two-Stage RLfD

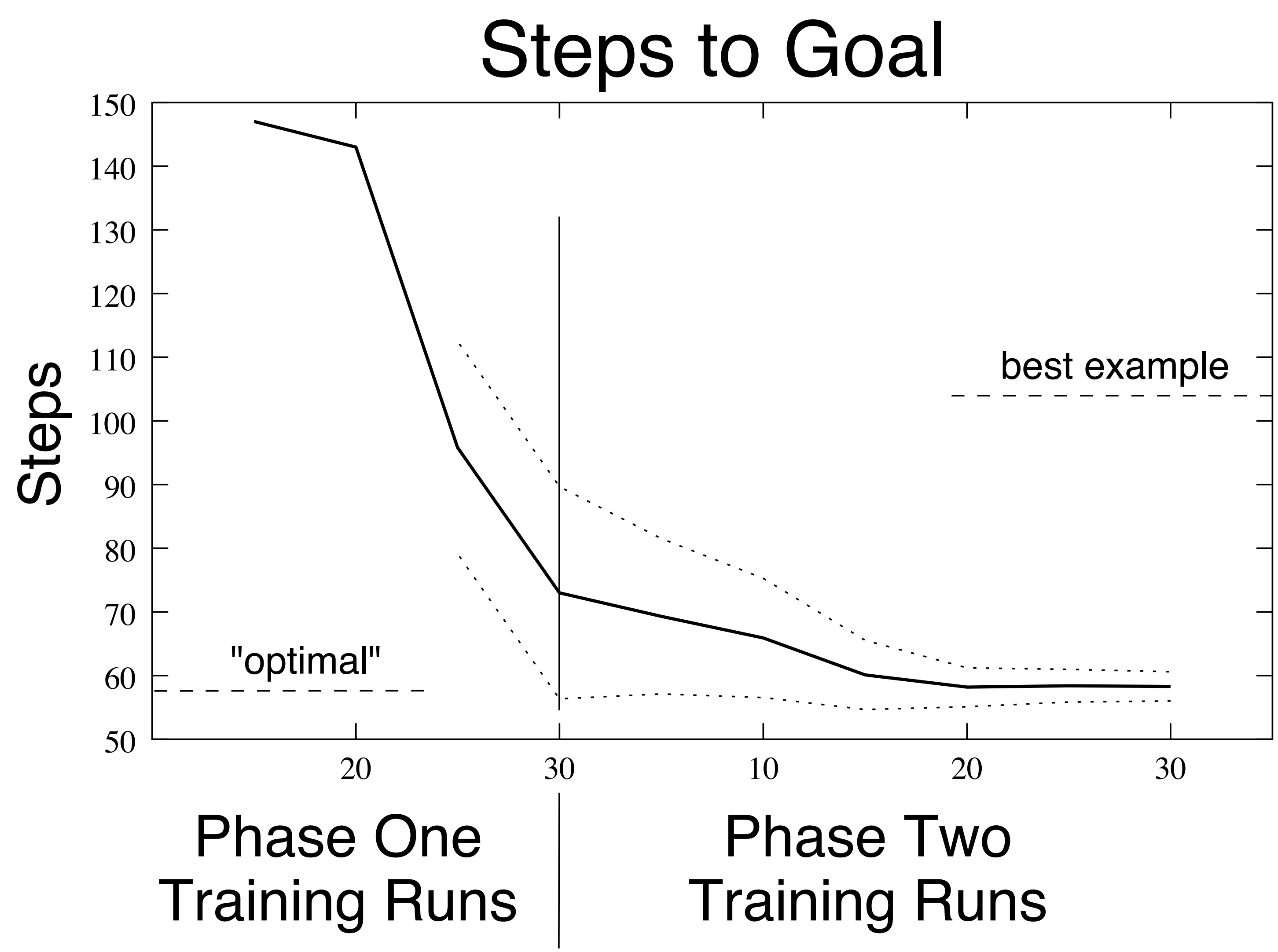

**Smart, W. D., & Kaelbling, L. P. (2002). Effective reinforcement learning for mobile robots. ICRA**

- 1st stage: robot passively watches human demonstrator and learns from observed (s,a,r,s')

- 2nd stage: robot actively controls the system and continues learning

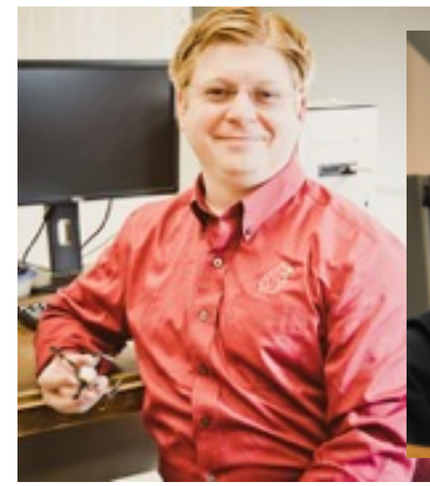
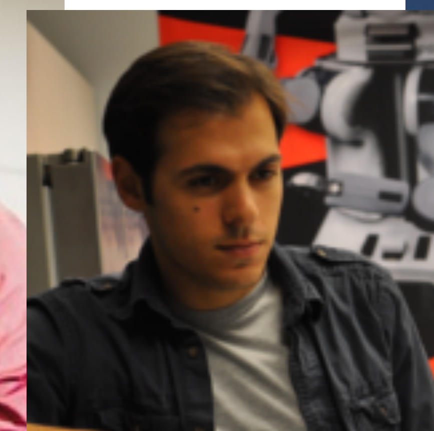
# Two-Stage RLfD



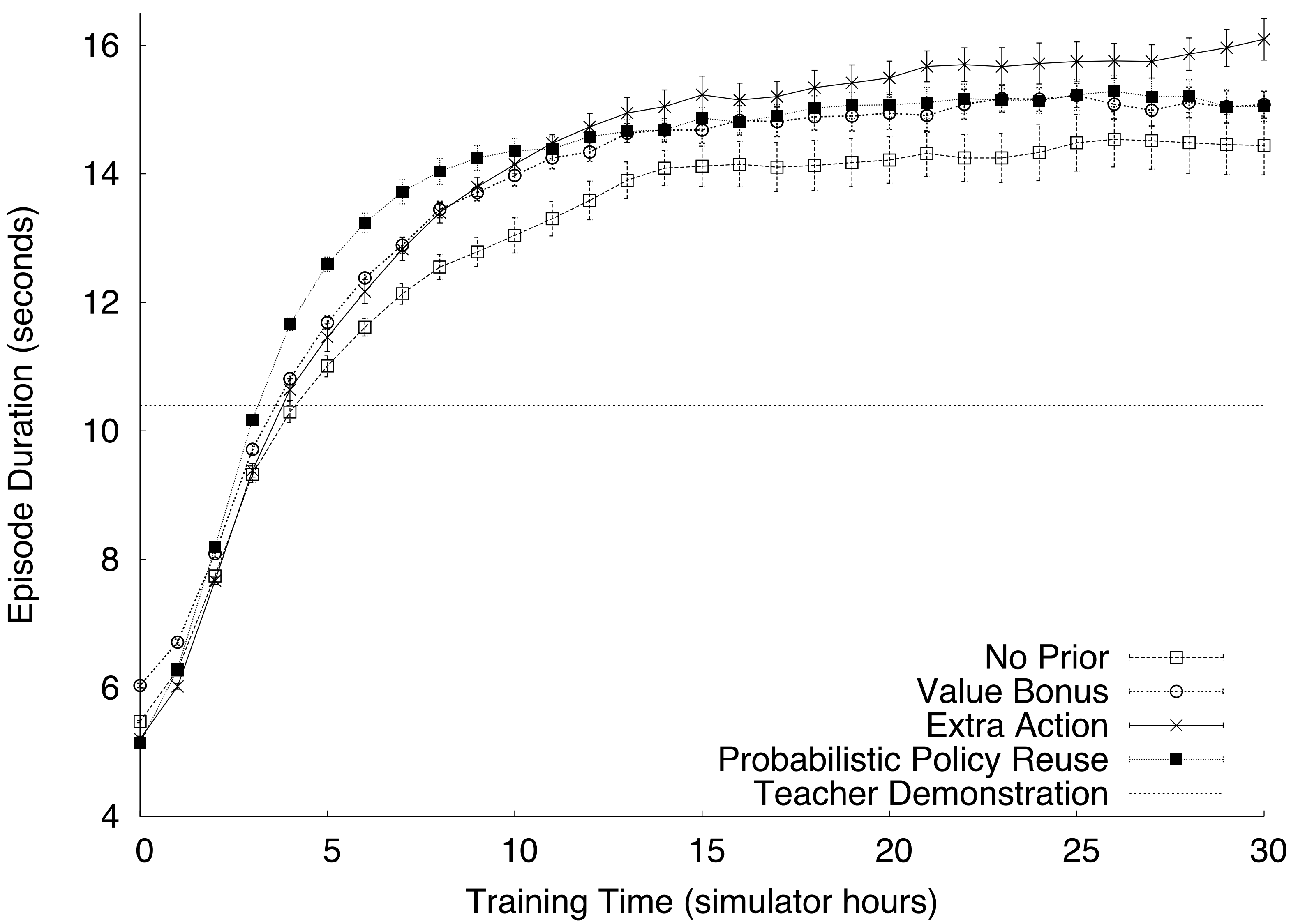
# HAT

**Taylor, M. E., Suay, H. B., & Chernova, S. (2011). Integrating reinforcement learning with human demonstrations of varying ability. AAMAS**


- Human-Agent Transfer
- Based on a set of demonstrations in a task, use a standard LfD technique to generate a policy for that task
- “Transfer” this policy to the RL agent, and let it use that policy to bias its learning

# HAT



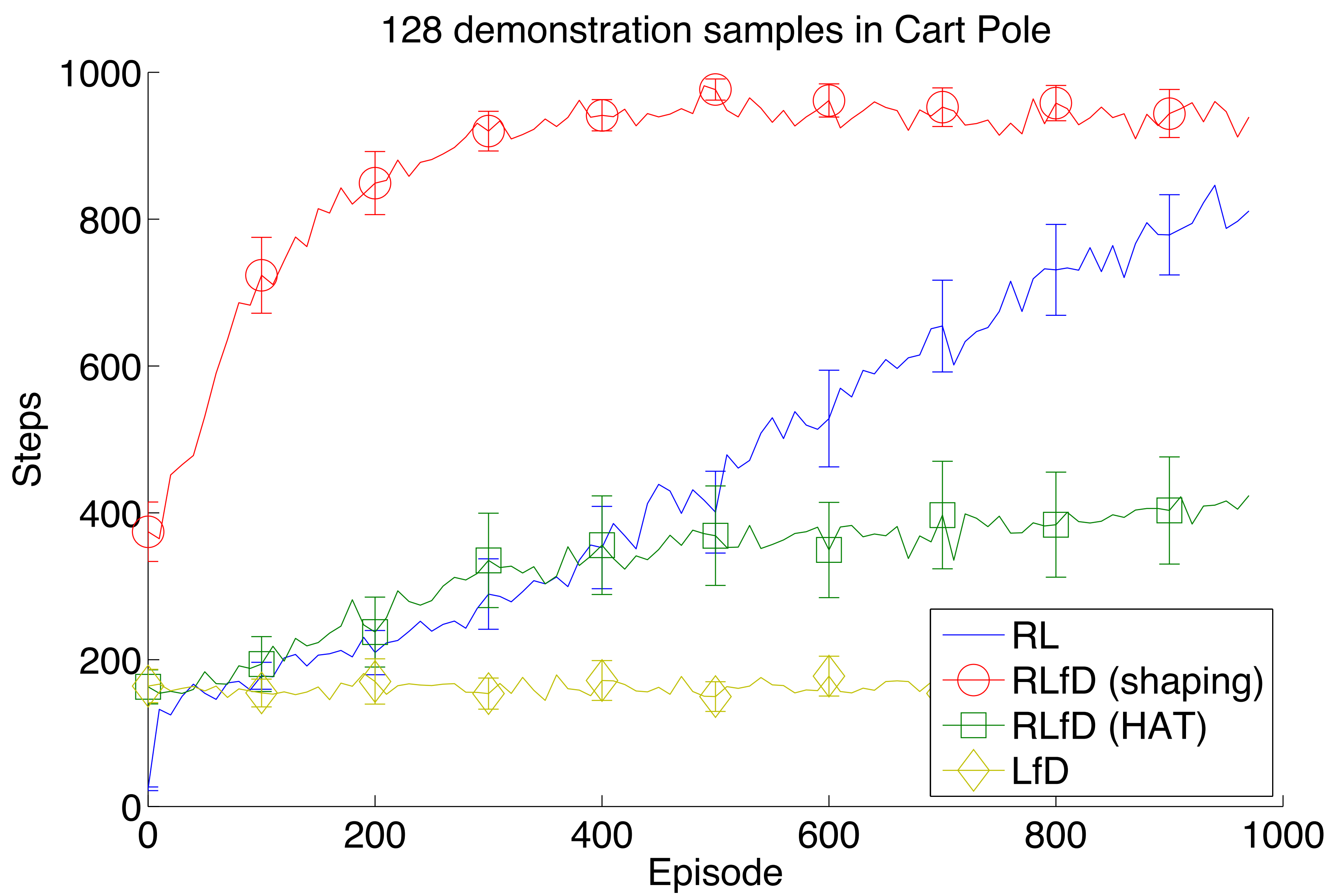
Episode Duration (seconds)
Training Time (simulator hours)
4
6
8
10
12
14
16
0
5
10
15
20
25
30
No Prior
Value Bonus
Extra Action
Probabilistic Policy Reuse
Teacher Demonstration

# RLfD through Shaping

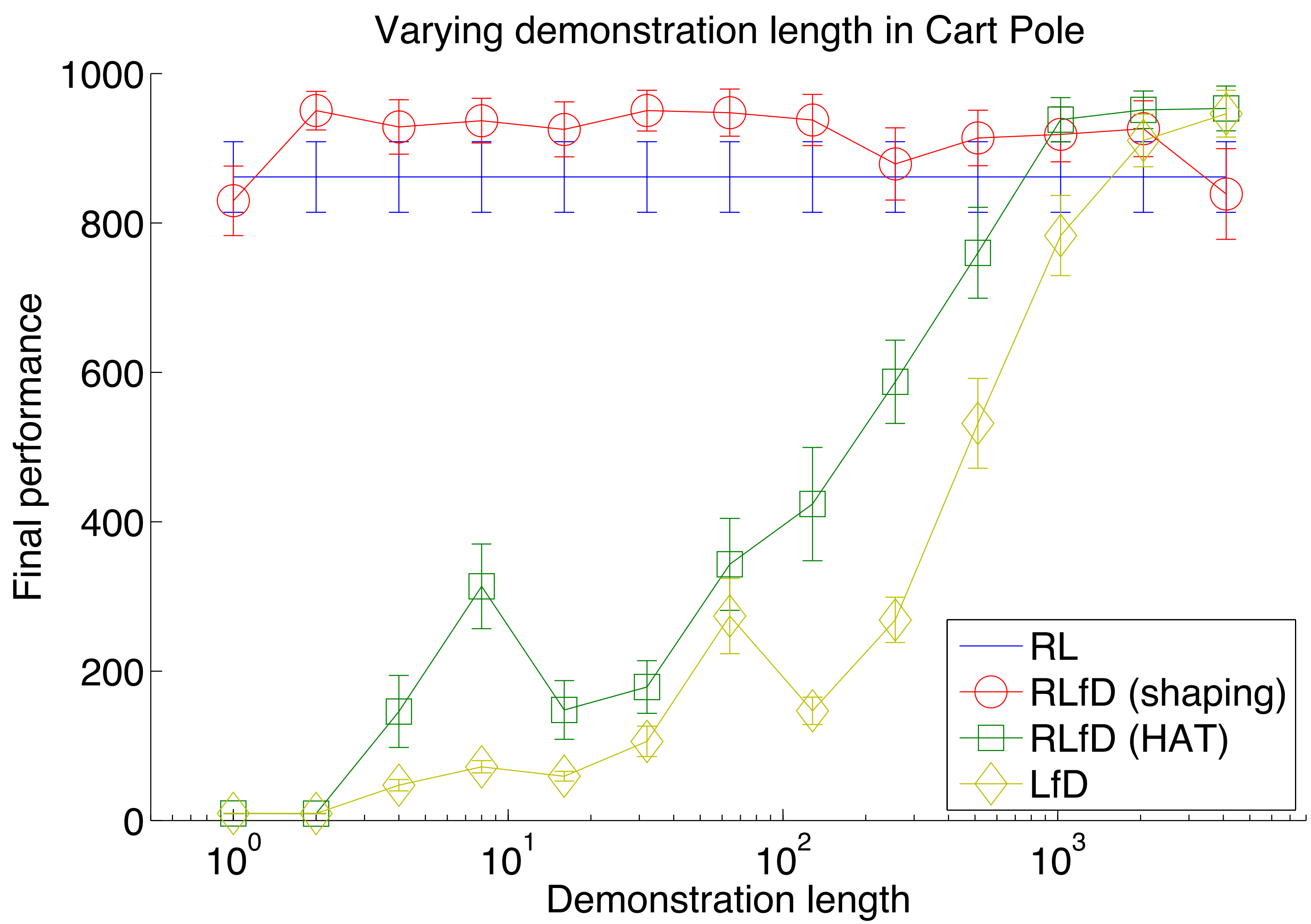
**Brys, T., Harutyunyan A., Suay, H. B., Chernova, S., Taylor, M. E. & Nowé, A, (2015). Reinforcement Learning from Demonstration through Shaping. IJCAI**

- Encode demonstrations as a reward shaping function
- Place a Gaussian everywhere a state-action pair has been demonstrated
- Potential is high when close by (in the state space) the same action has been demonstrated

# RLfD through Shaping



# RLfD through Shaping

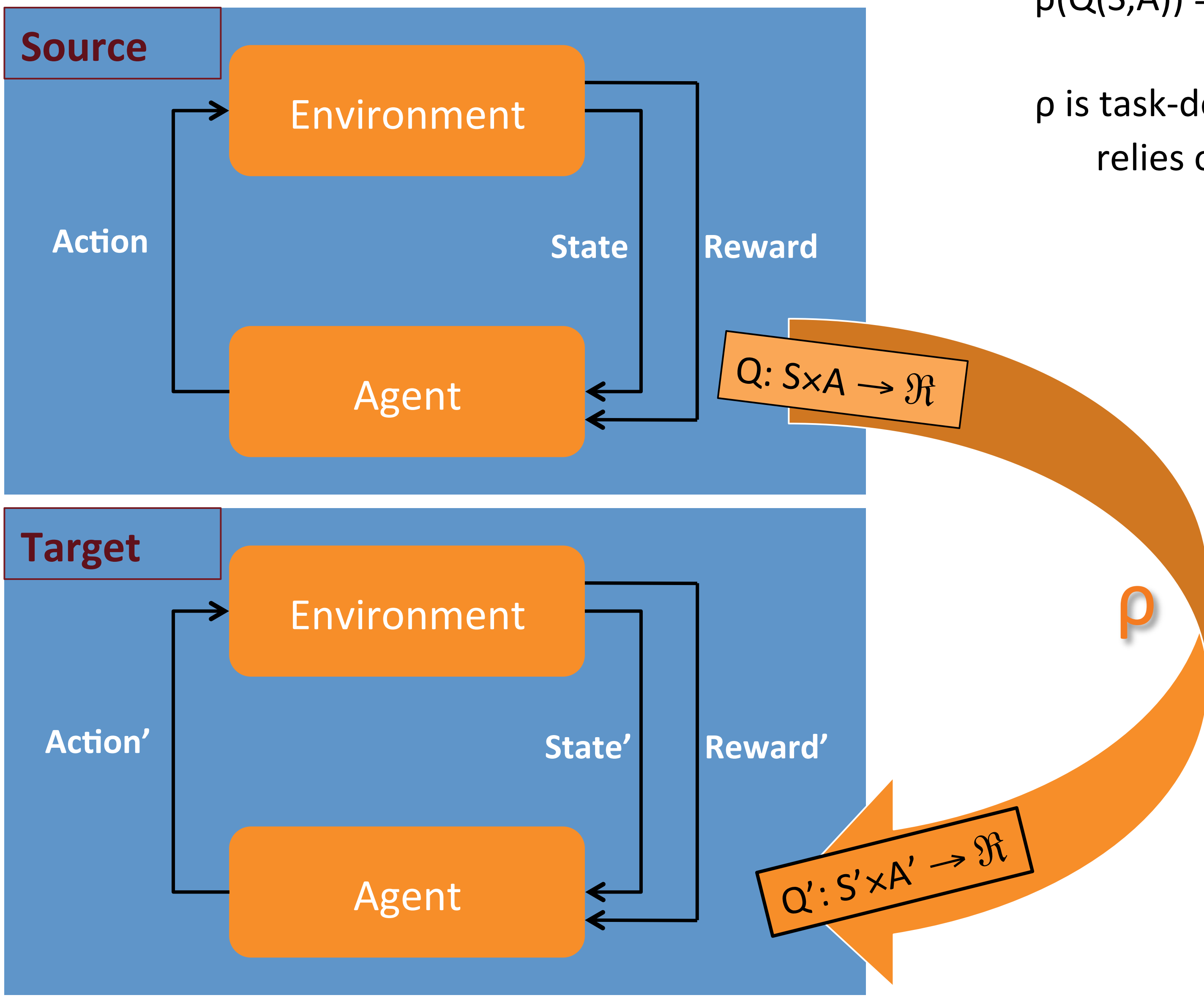


# Part IV: Transfer Learning

- Background: Matthew E. Taylor and Peter Stone. Transfer Learning for Reinforcement Learning Domains: A Survey. Journal of Machine Learning Research, 10(1):1633-1685, 2009

- Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. Online multi-task learning for policy gradient methods. ICML-14
- Anestis Fachantidis, Ioannis Partalas, Matthew E. Taylor. and Ioannis Vlahavas. Transfer learning with probabilistic mapping selection. Adaptive Behavior, 23(1): 3-19, 2015
- George Konidaris and Andrew Barto. Autonomous shaping: knowledge transfer in reinforcement learning. ICML-06
- Alessandro Lazaric , Marcello Restelli , Andrea Bonarini. Transfer of samples in batch reinforcement learning. ICML-08
- Paul Ruvolo and Eric Eaton. ELLA: an efficient lifelong learning algorithm. ICML-13
- Matthew E. Taylor, Nicholas K. Jong, and Peter Stone. Transferring instances for model-based reinforcement learning. ECML-08
- Matthew E. Taylor, Peter Stone, and Yaxin Liu. Transfer Learning via Inter-Task Mappings for Temporal Difference Learning. Journal of Machine Learning Research, 8(1):2125-2167, 2007

# Value Function Transfer




ρ(Q(S,A)) = Q'(S',A')

ρ is task-dependant:

relies on inter-task mappings

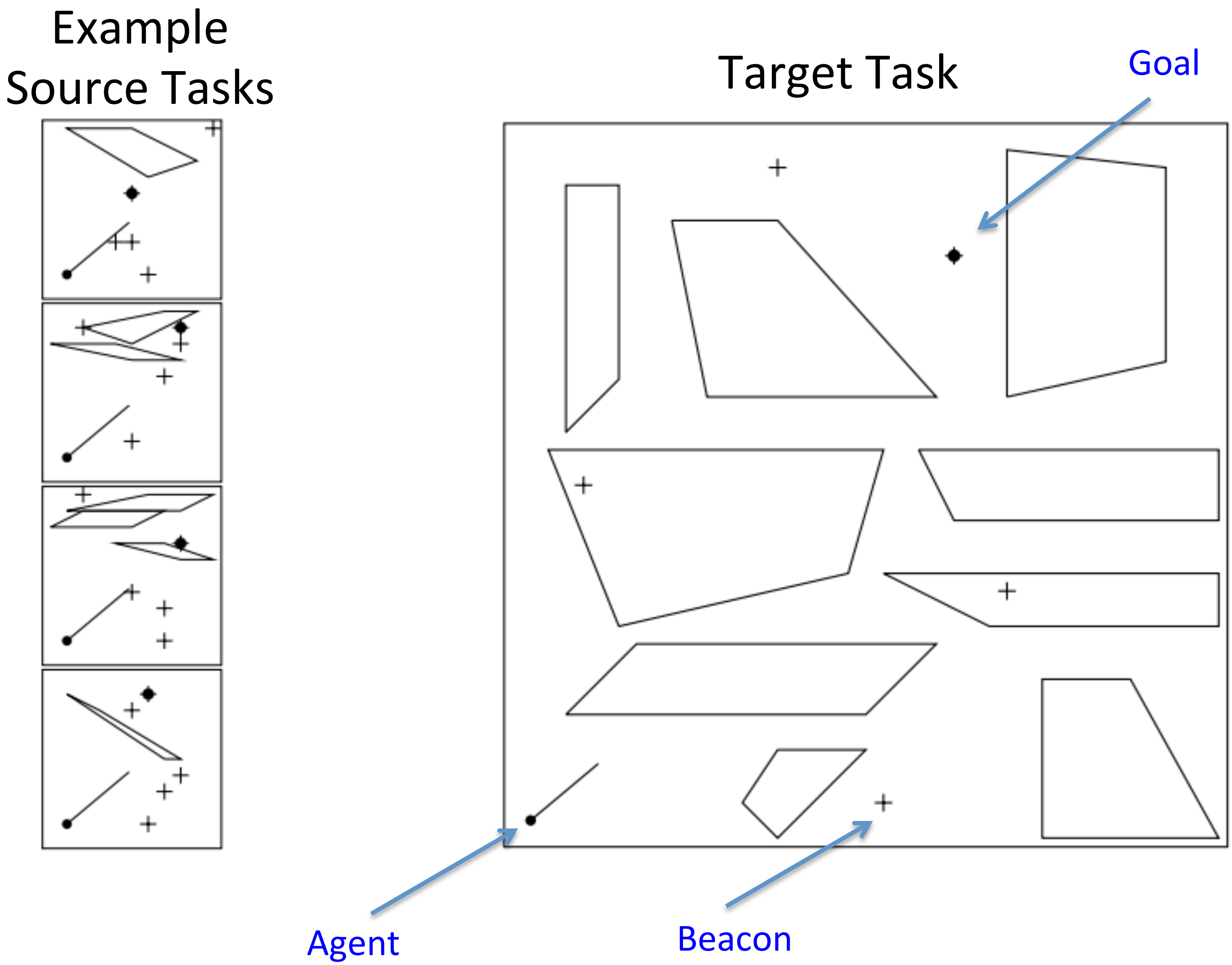
Taylor+, JMLR 2007

*Autonomous Shaping: Knowledge Transfer in Reinforcement Learning*, Konidaris & Barto, 2006

- Problem-Space: individual tasks
- Agent-Space: constant across tasks

- Example: heat sensor on robot, task = find heat source

- Shaping reward over states (e.g., V, not Q)




Beacons emits separate signals that drop off with square of Euclidean distance

# Transfer of Samples in Batch Reinforcement Learning, Lazaric+, 2008

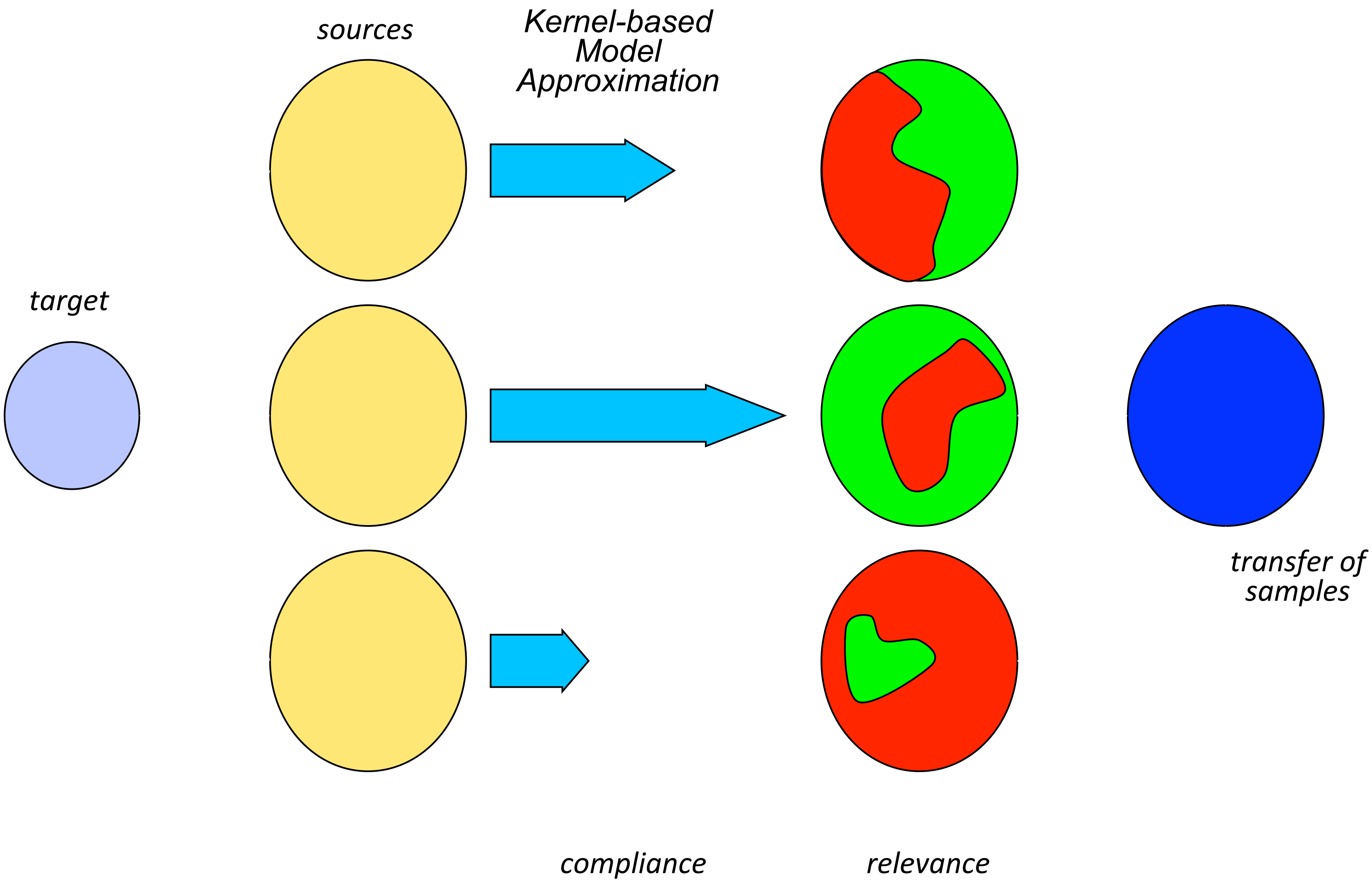
Multi-task setting

Instance-based method

Compliance: find most similar source task

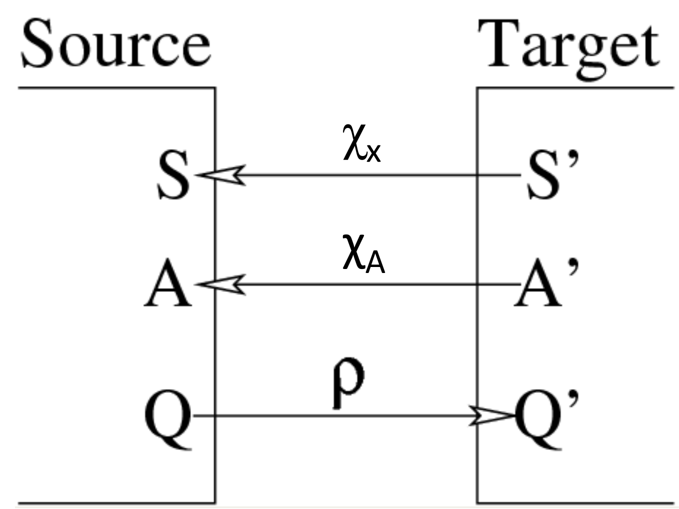
Relevance: find the most useful source task instances

- Ordered by similarity in afterstates
- "The assumption underlying the definition of relevance is that, whenever there is no evidence against the transfer of a sample, it is convenient to transfer it to the target task."


sources
Kernel-based
Model
Approximation
target
transfer of
samples
compliance
relevance

# Inter-Task Mappings

- $\chi_{X:}\ s_{target} \rightarrow s_{source}$
  - Given state / state variable in target task
  - Return corresponding state / state variable in source task
- $\chi_{A:}\ a_{target} \rightarrow a_{source}$
  - Similar, but for actions
- Intuitive mappings exist in some domains (Oracle)
- Used to construct ρ



# Transferring Instances for Model-Based REinforcement Learning, Taylor+, 2008

TIMBREL

*n.* An ancient percussion instrument similar to a tambourine

Leverages Fitted R-Max (Jong & Stone, 2007)

Instance-based method

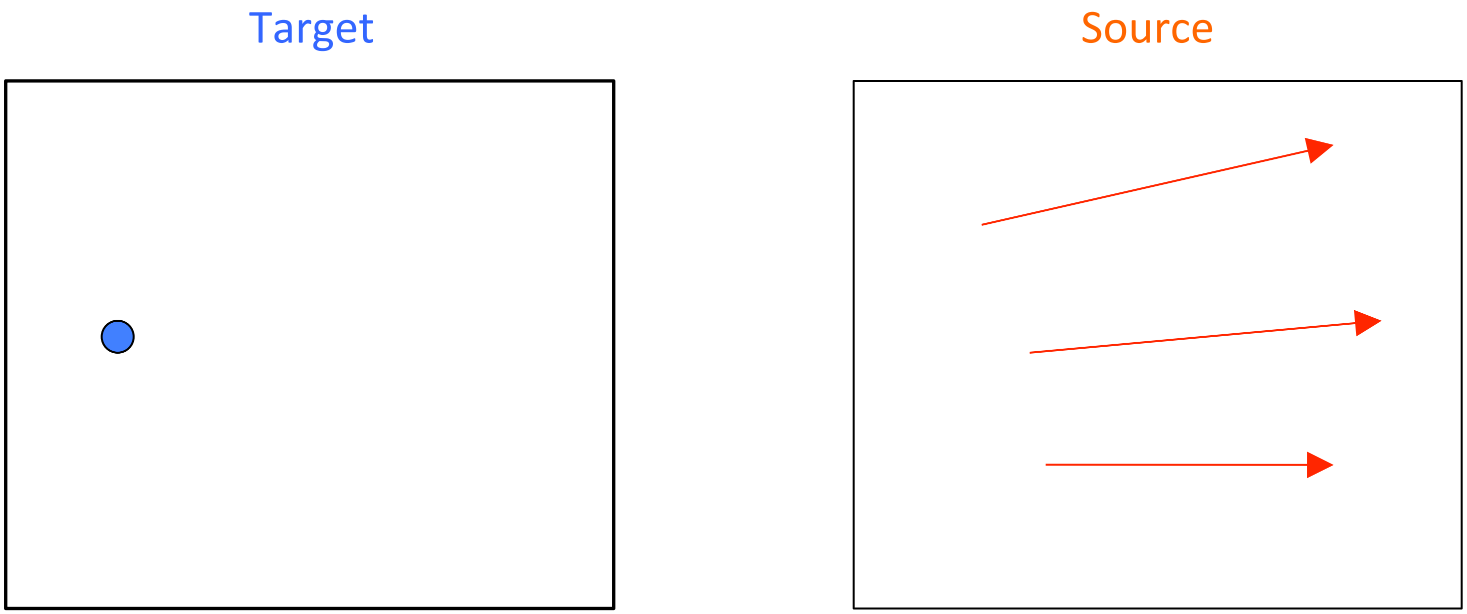
Assumes you know the (correct) inter-task mapping

# TIMBREL

if target task model (*T* or *R*) is poor

Use inter-task mapping to find closest source task instances most similar to $s_{target}$

Use transformed instances to estimate target task $T$ and $R$

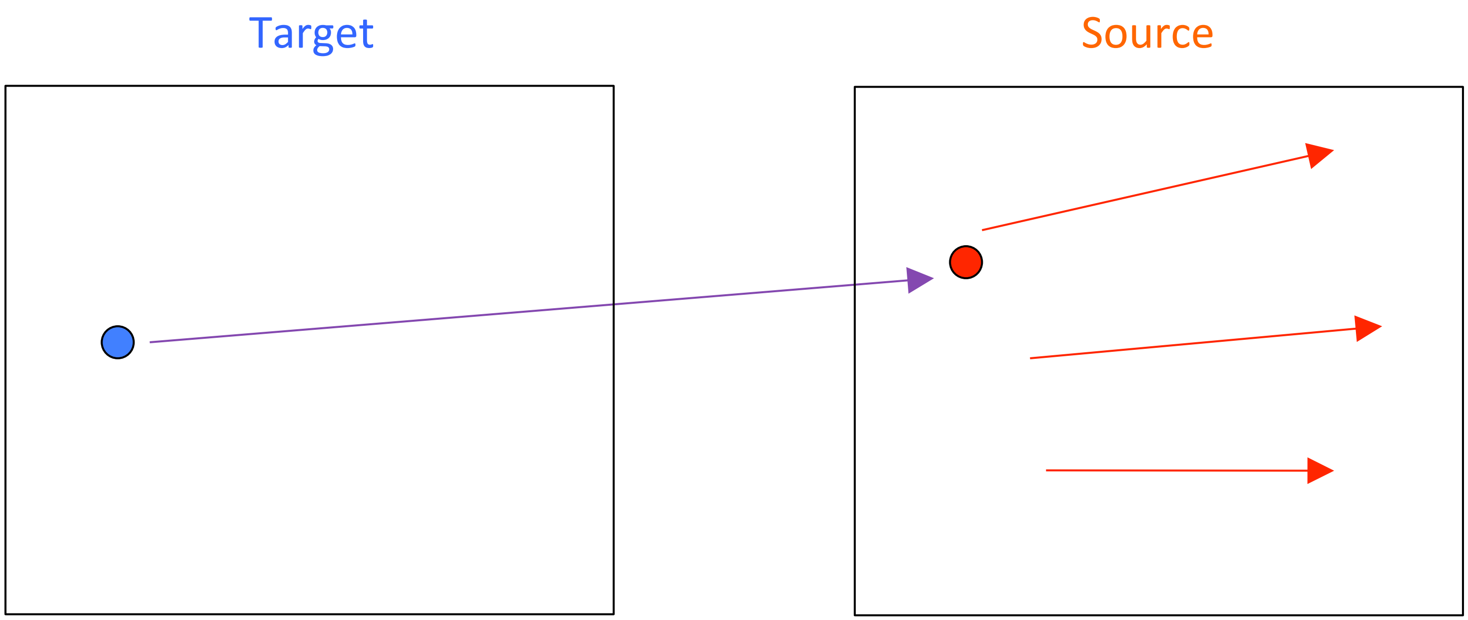


# TIMBREL

if target task model (*T* or *R*) is poor

Use inter-task mapping to find closest source task instances most similar to $s_{target}$

Use transformed instances to estimate target task *T* and *R*

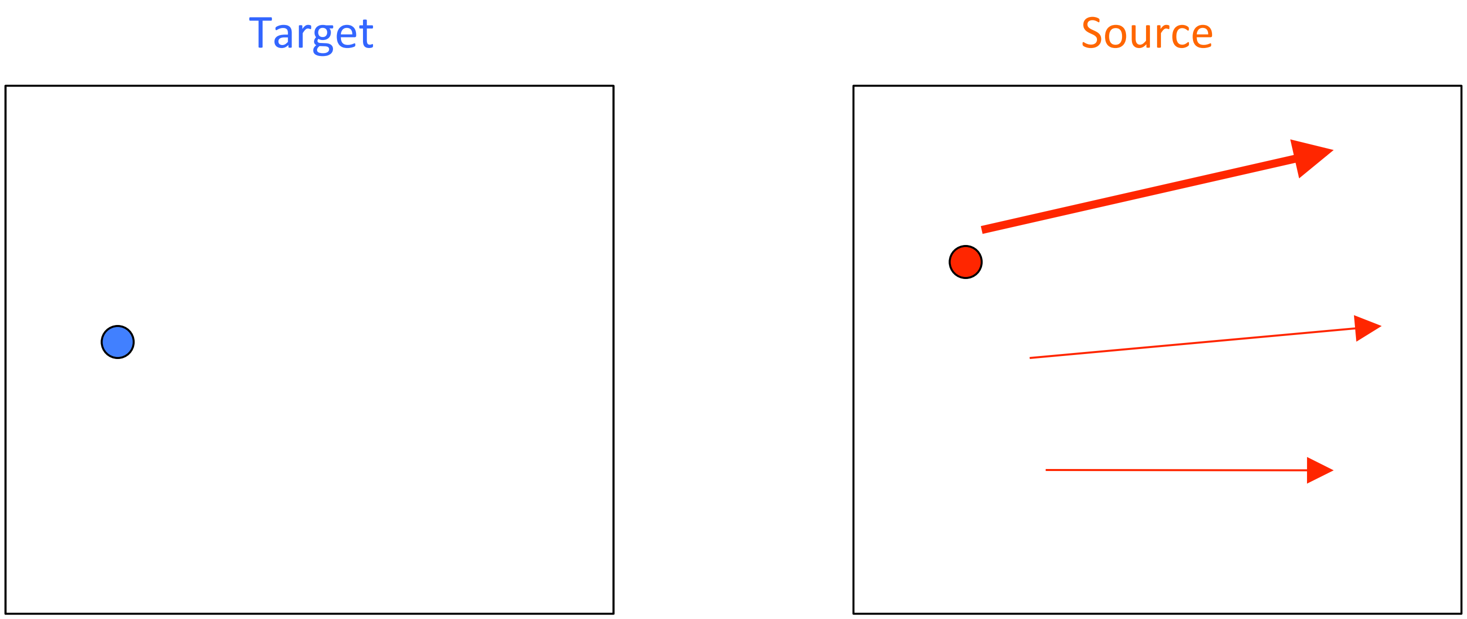


# TIMBREL

if target task model (*T* or *R*) is poor

Use inter-task mapping to find closest source task instances most similar to $s_{target}$

Use transformed instances to estimate target task *T* and *R*



# COMBREL

- Translate multiple mapping problem to multi-task transfer problem
  - Each inter-task mapping is a hypothesis
  - Consider multiple mappings to transform single source task to multiple *virtual* source tasks
  - Compliance!
- Automated method to select state and action mappings
  - Can be state-dependent (in target task)

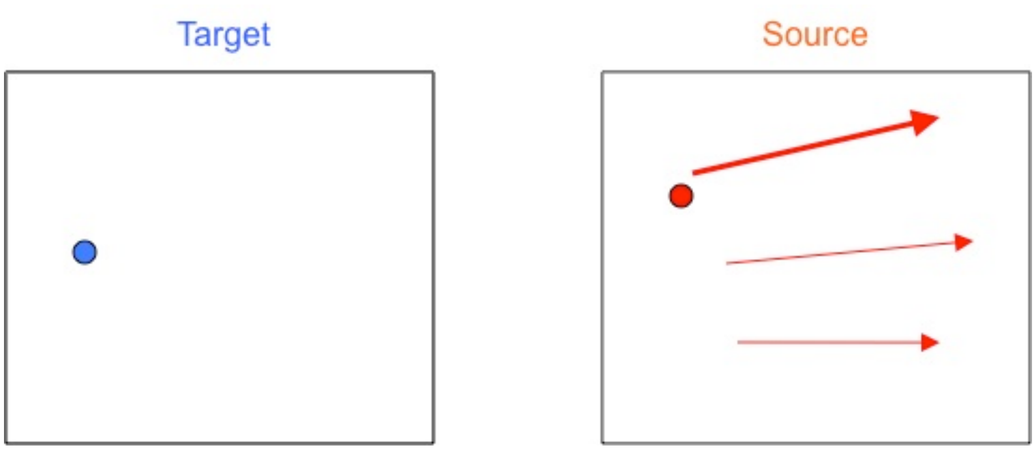
Transfer Learning with Probabilistic Mapping Selection, Fachantidis+, Adaptive Behavior, 2015

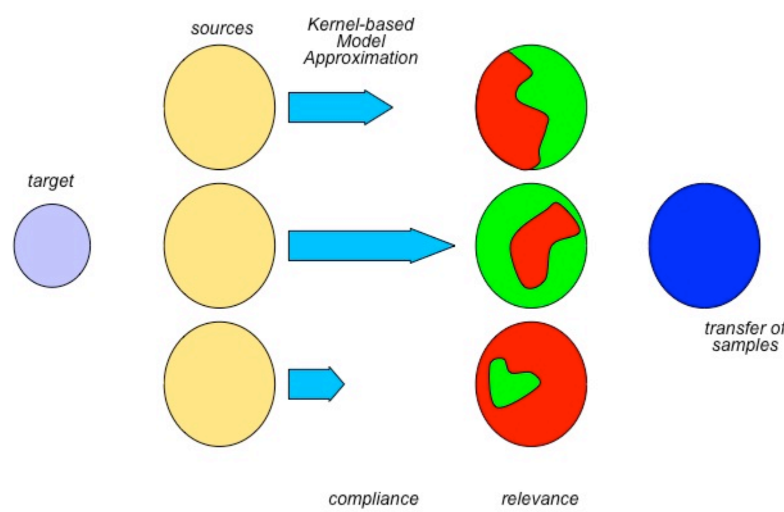
# COMBREL

## Compliance aware transfer for Model-Based REinforcement Learning

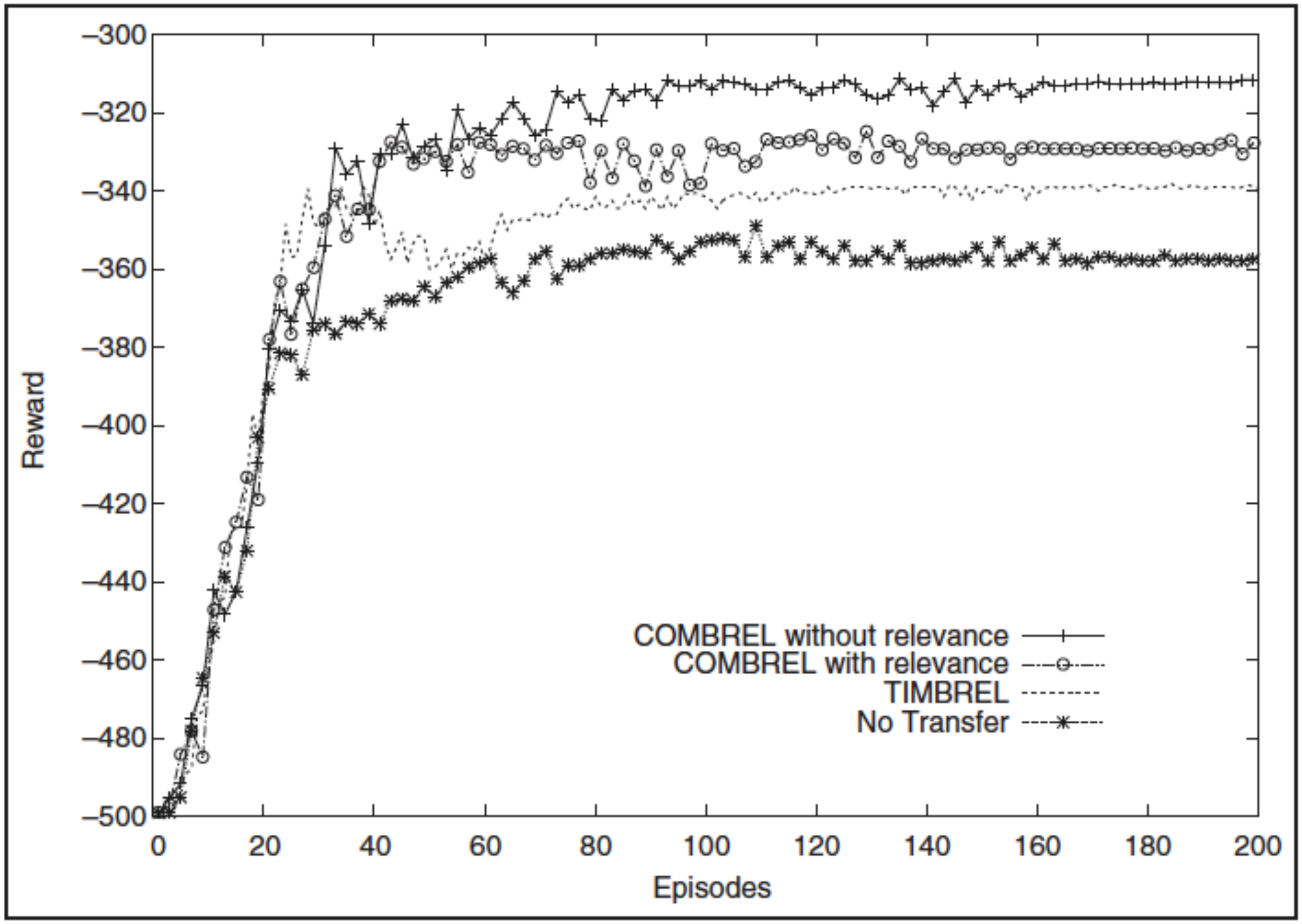
if target task model (*T* or *R*) is poor for current $s_{target}$, $a_{target}$

- Calc average compliance of k-nearest target task instances to each virtual source task
- Select most compliant source task
- if using relevance:
  - Compute relevance of each source task instance to $s_{target}$, $a_{target}$
  - Add most relevant to samples current model
- else
  - Use Euclidian distance to target task instance (TIMBREL method)



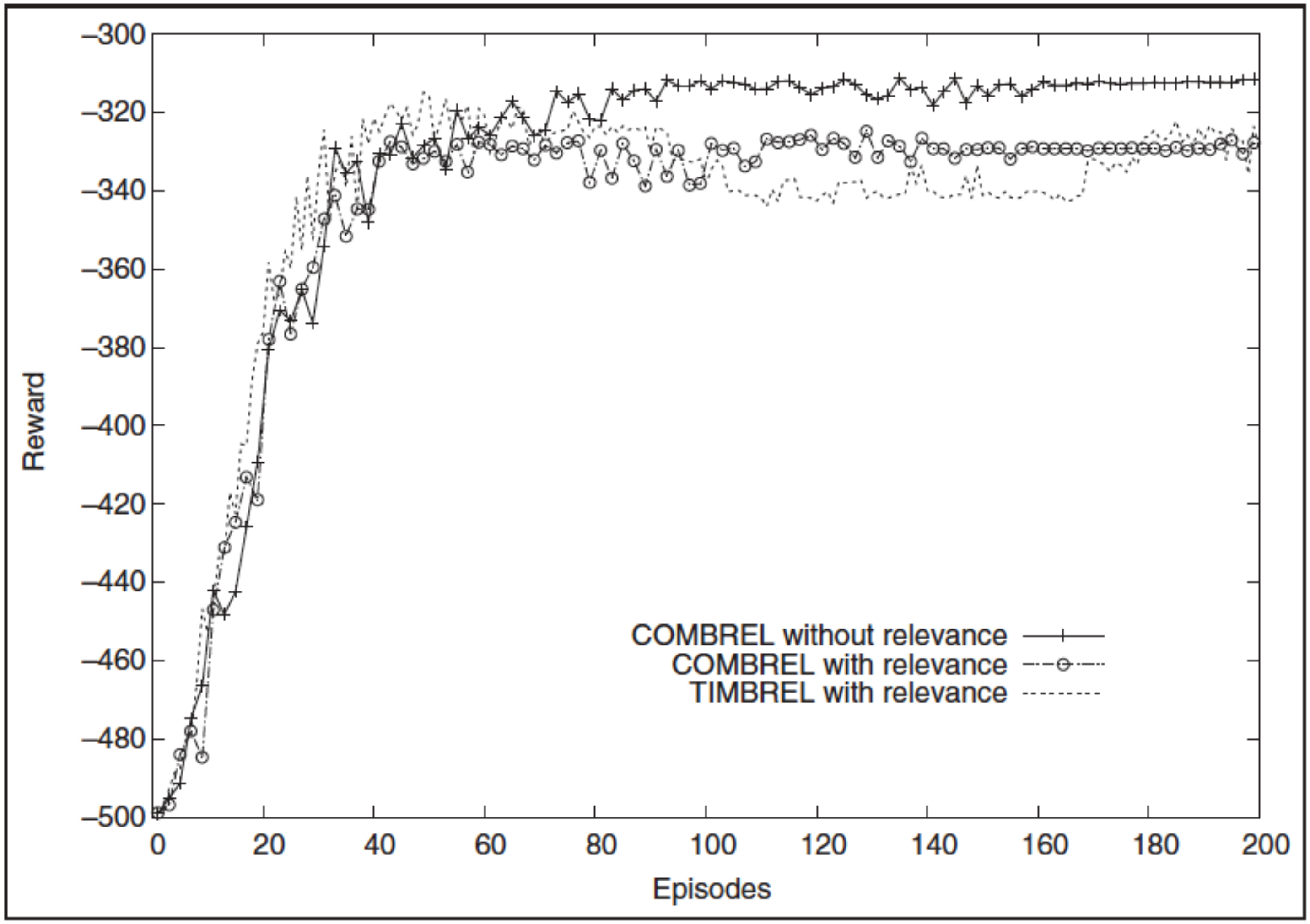



- 2D Mountain Car → 4D Mountain Car
- 1000 source task instances, 1960 mappings




Multiple mappings better than 1 'best' mapping

- 2D Mountain Car ➔ 4D Mountain Car
- Use 1960 mappings: create one instance “pool”

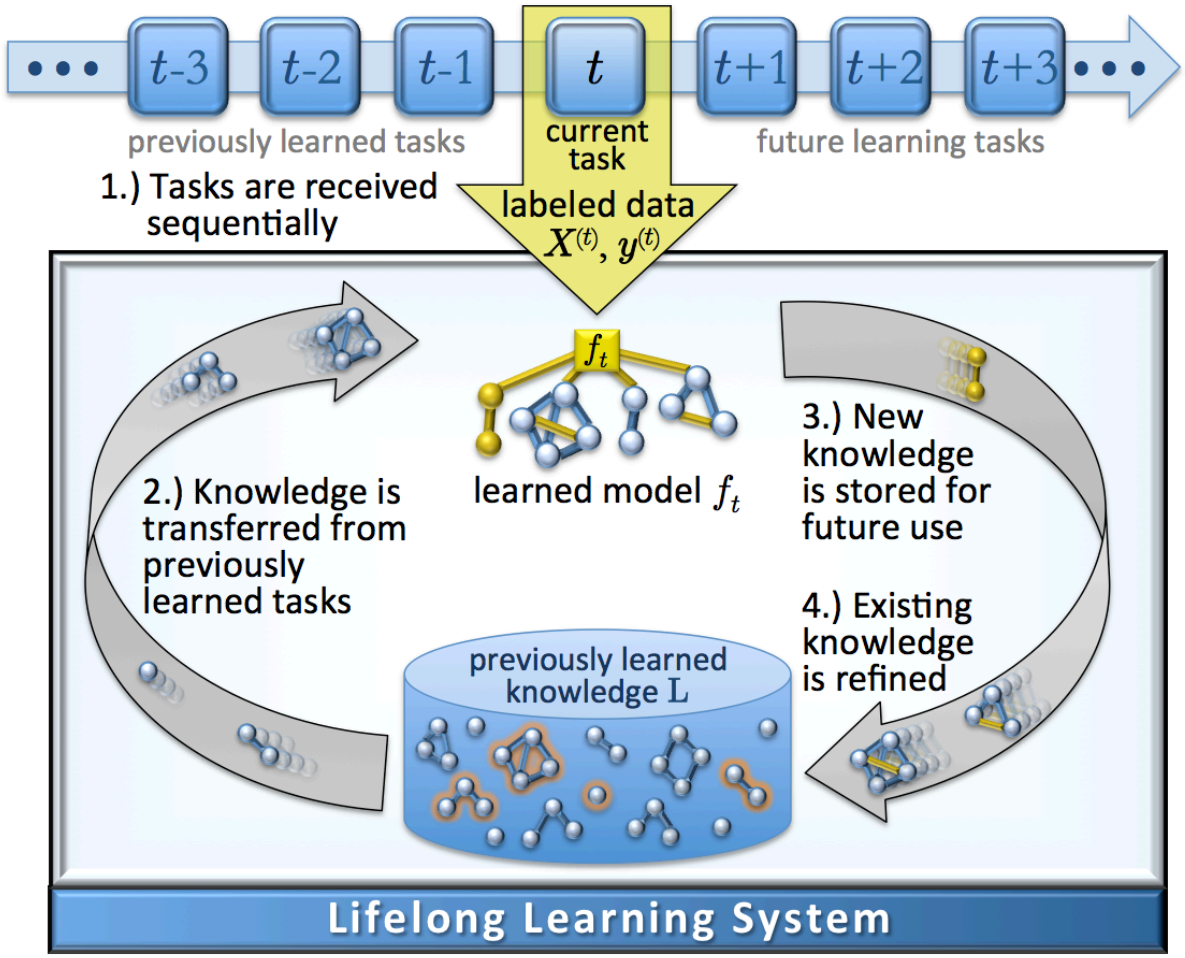



Compliance does improve performance

# iCub: Ball hitting task

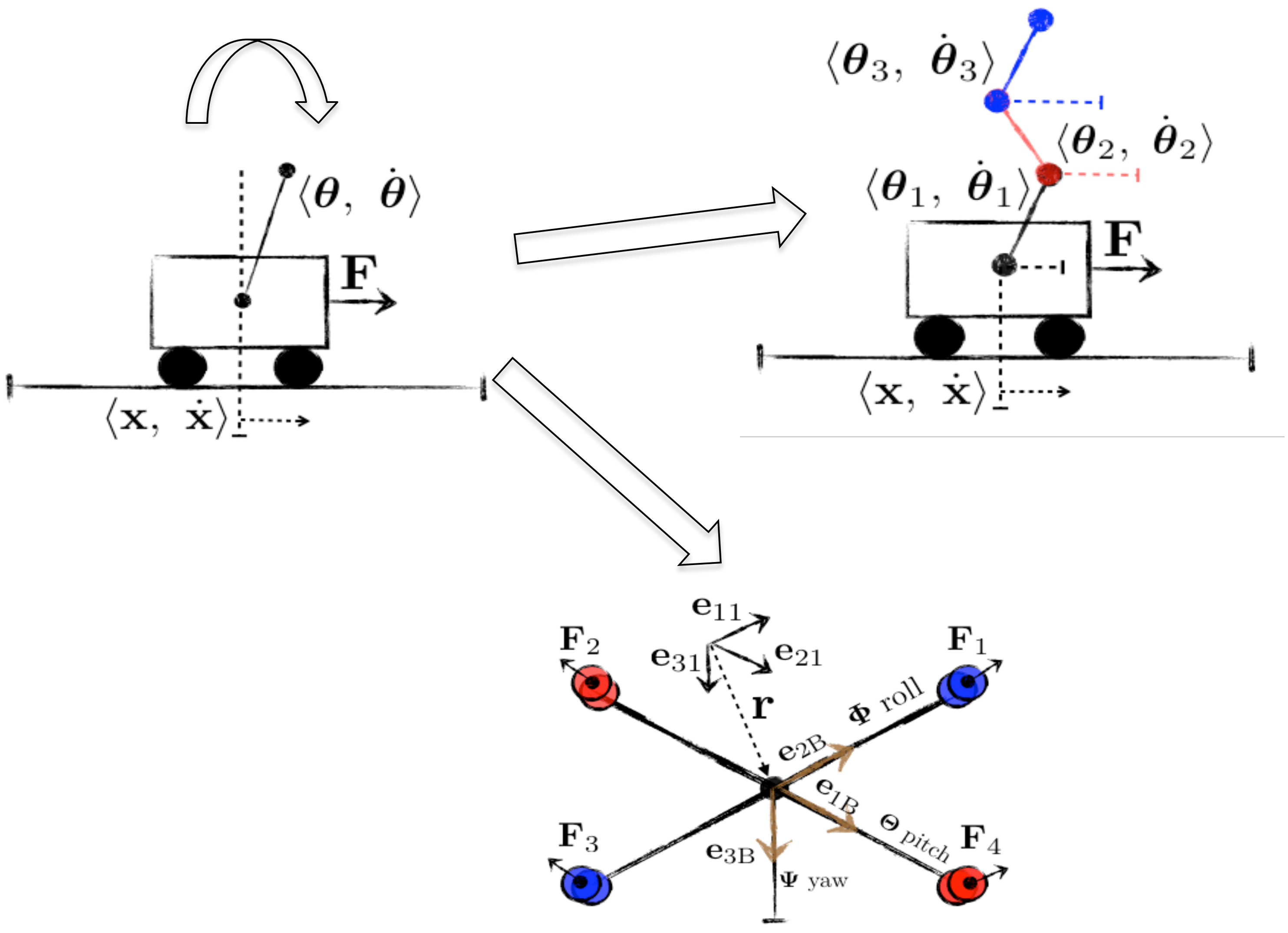
- 2 or 4 degrees of freedom
- 1152 mappings (24 state mappings, 48 action mappings)

# ELLA, Ruvolo & Eaton, 2013



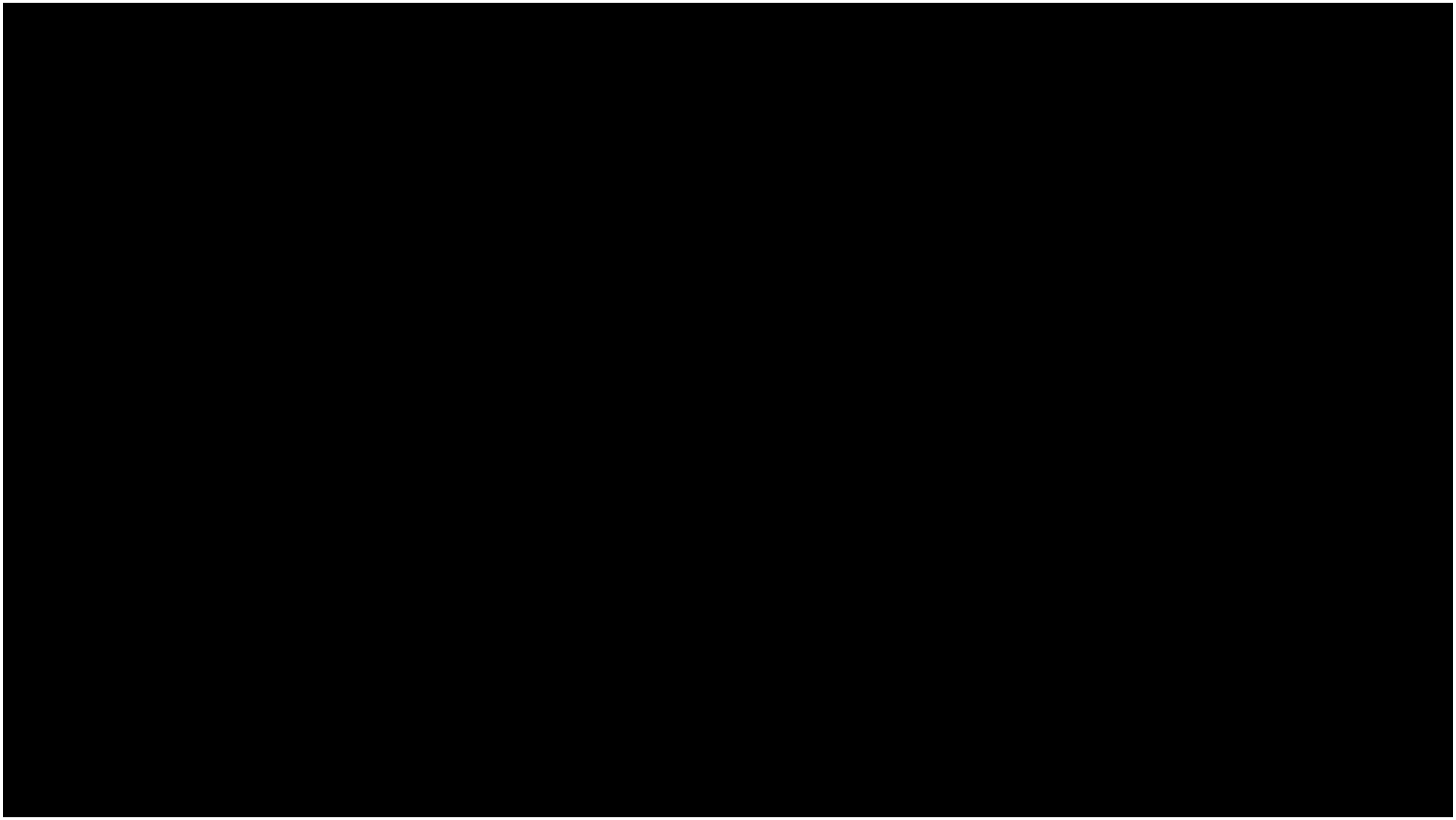

ELLA: Supervised learning, equivalent accuracy to batch multi-task learning, over 1,000x faster and can learn online

# PG-ELLA: Bou Aamar+, 2014



# Standard PG vs PG-ELLA: Cart-Pole

PG-ELLA Standard PG

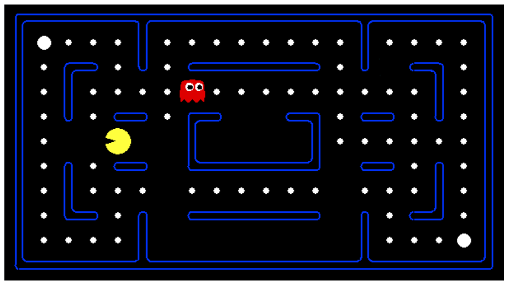


# Related Work at AAMAS-15

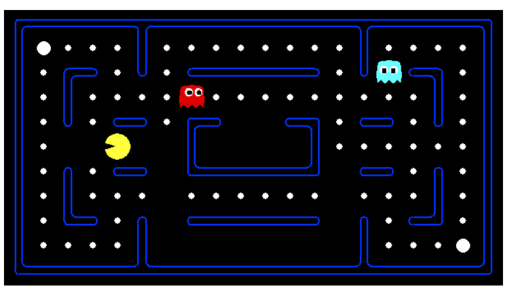
**Learning in Multi-agent Systems with Sparse Interactions by Knowledge Transfer and Game Abstraction**

Yujing Hu, Yang Gao, Bo An

**Question:** How to utilize agents' single-agent knowledge learnt before when they are learning in a MAS with sparse interactions?



**Single-agent knowledge??**



## Three Knowledge Transfer Mechanisms

**Value function transfer (VFT):**
Transferring agents' local value function directly since the interactions between agents are sparse

**Selective value function transfer (SVFT):**
1. Transferring value function only in states where agents can act independently
2. MDP similarity based on *Kantorovich metric* is defined to determine whether to transfer the value function in each state

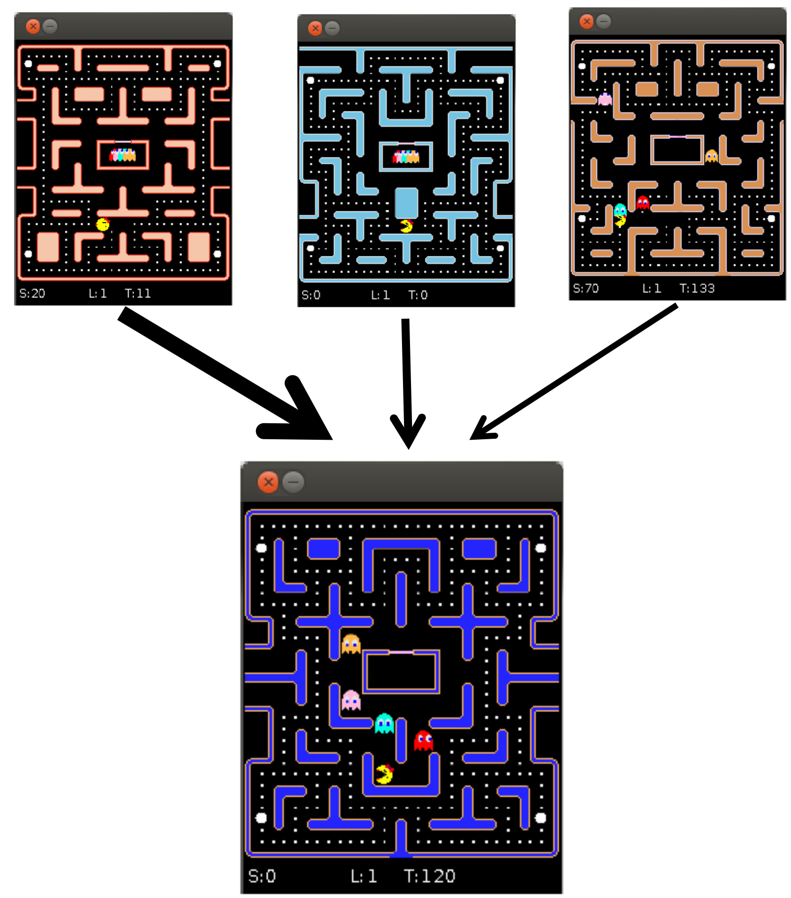
**Model transfer-based game abstraction (MTGA):**
1. Transferring reward and transition models
2. Reducing the joint state-action space of the learning algorithm based on MDP similarity

Learning II, G3, 11:00 – 12:30 on Thursday, 7th May, Üsküdar 1

# Learning Inter-Task Transferability in the Absence of Target Task Samples

**Jivko Sinapov, Sanmit Narvekar, Matteo Leonetti, Peter Stone**
**University of Texas at Austin**

- Can an agent learn to predict the benefit of transferring a policy from one task to another?
- Short answer: yes!
- Using the learned model, the agent was able to select good source task that improved learning on target tasks



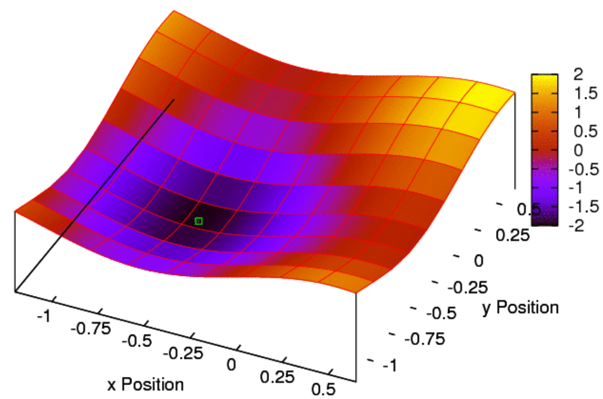

Learning II, G3, 11:00 – 12:30 on Thursday, 7th May, Üsküdar 1

# Policy Transfer using Reward Shaping

**Tim Brys, Anna Harutyunyan, Matthew E. Taylor, Ann Nowé**

Transfer policy from similar task

- RL, LfD, Human defined, …
- Black box: can only query π(s,a)
- Encode source as dynamic shaping reward
  - Strong theoretical guarantees




- More robust to suboptimal policies than state-of-the-art

- Mountain Car, Cart Pole, Mario


Learning I, B3, 11:00 – 12:30 on Wednesday, 6th May, Üsküdar 1

# Part V: Agents Teaching Agents

- Matthew E. Taylor, Nicholas Carboni, Anestis Fachantidis, Ioannis Vlahavas, and Lisa Torrey. Reinforcement learning agents providing advice in complex video games. Connection Science, 26(1):45-63, 2014.

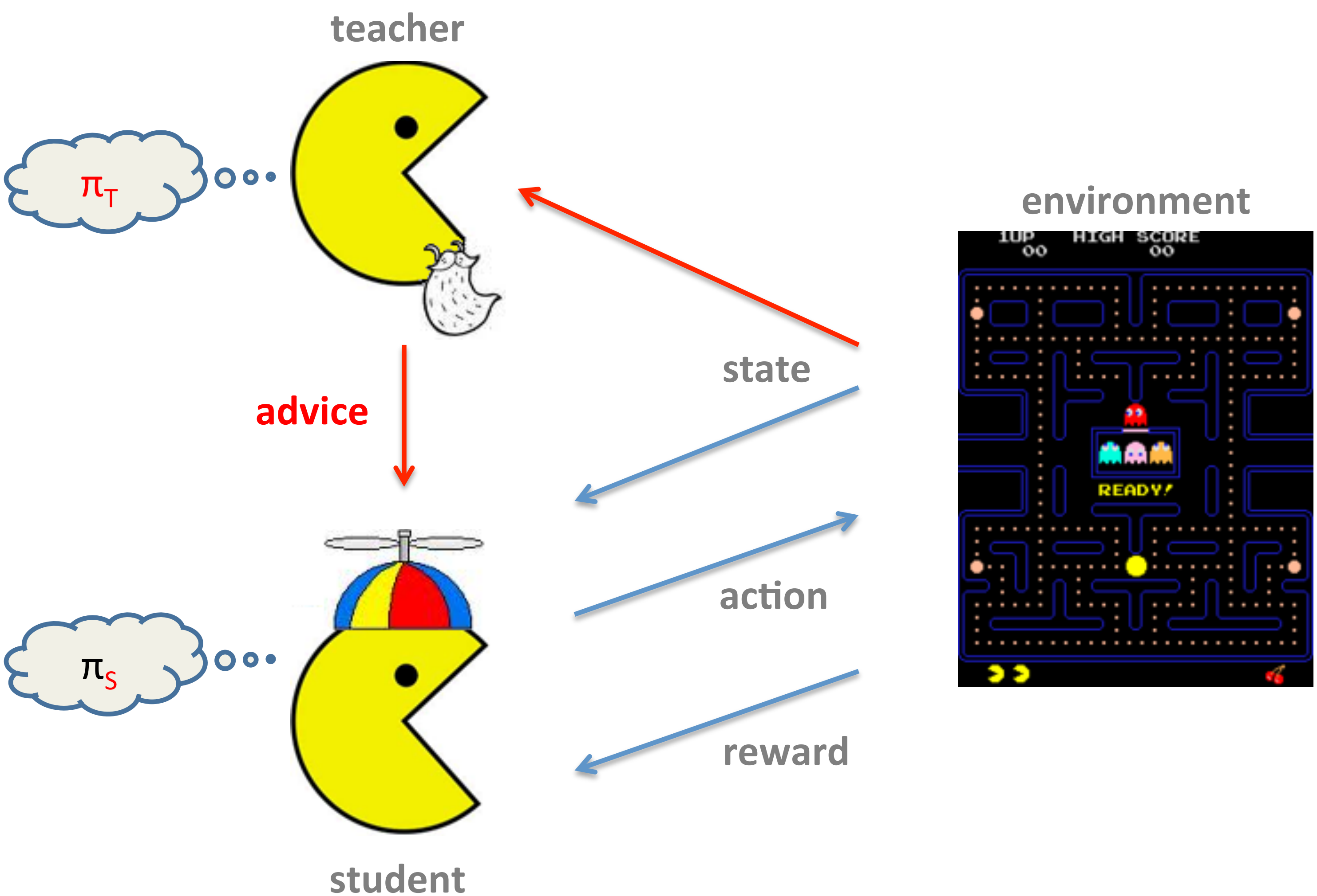
- Yusen Zhan, Anestis Fachantidis, Ioannis Vlahavas, and Matthew E. Taylor. Agents Teaching Humans in Reinforcement Learning Tasks. ALA (at AAMAS), 2014.

# Reinforcement Learning Agents Providing Advice in Complex Video Games

Taylor+, *Journal of Connection Science*, 2014

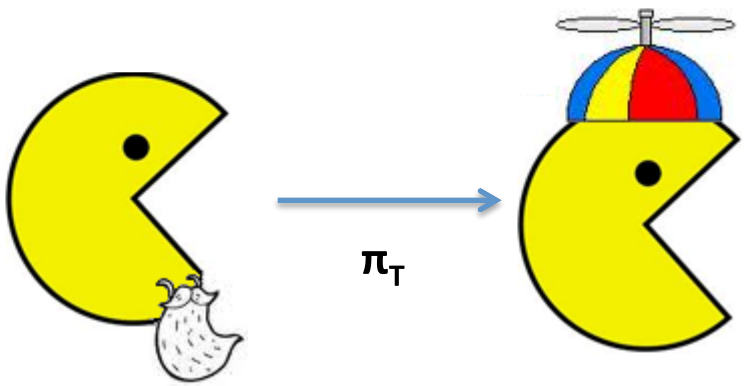
- Different state representation
- Different learning methods
- Only action advice
- Limited amounts of advice

# Reinforcement Learning + Teaching

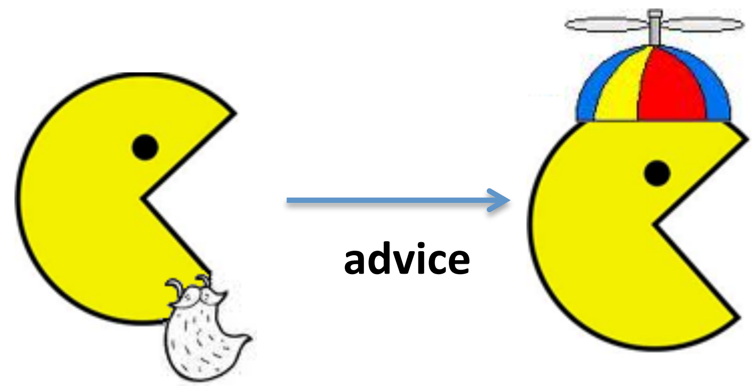


# Why Action Advice?

**Transfer learning**



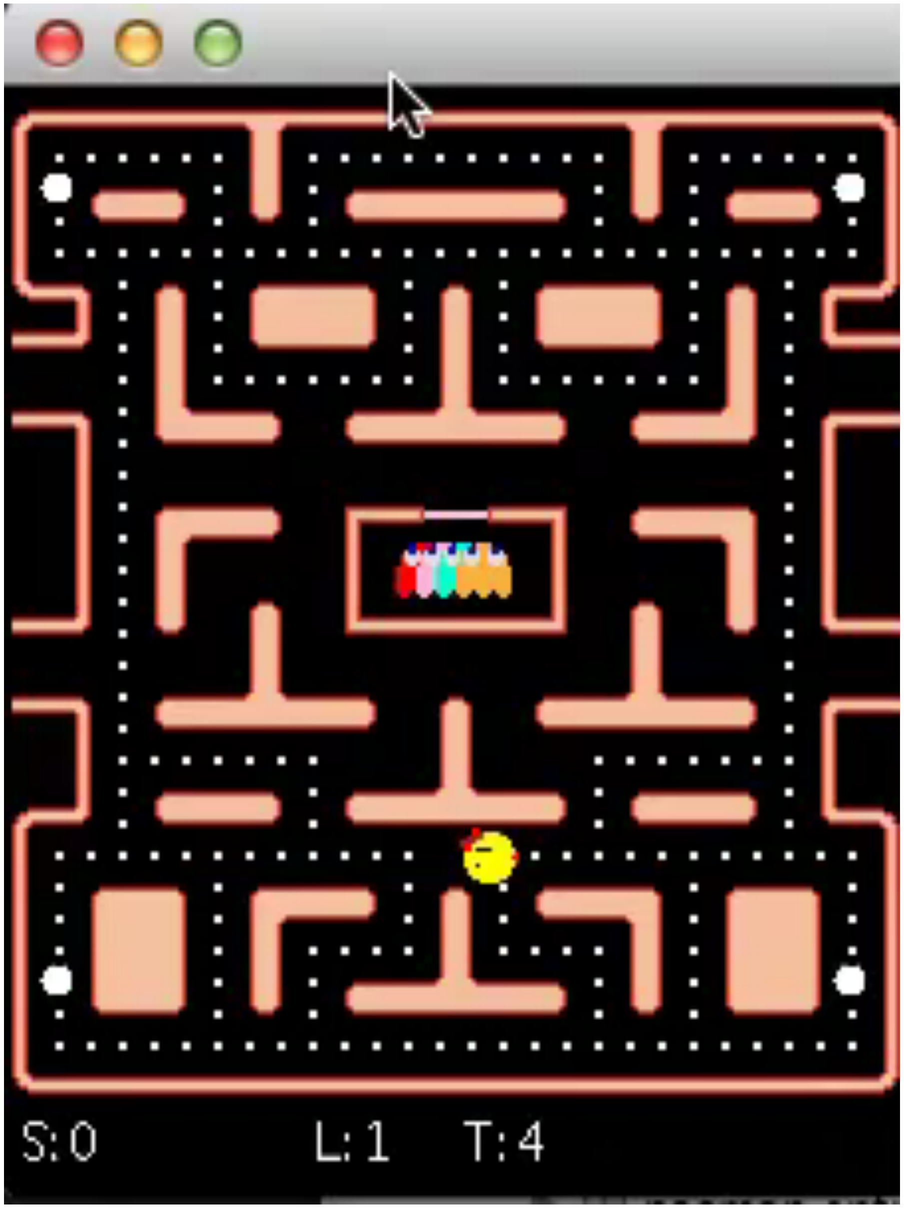

**Teaching via advice**




**Requirements**
- Direct access
- High similarity

**Requirements**
- Communication
- Minimal similarity

# Defining Advice Budget: Ms. Pac-Man




**Episode length**
Up to 2000 steps

**Training period**
500 episodes

**Advice budget**
1000 actions

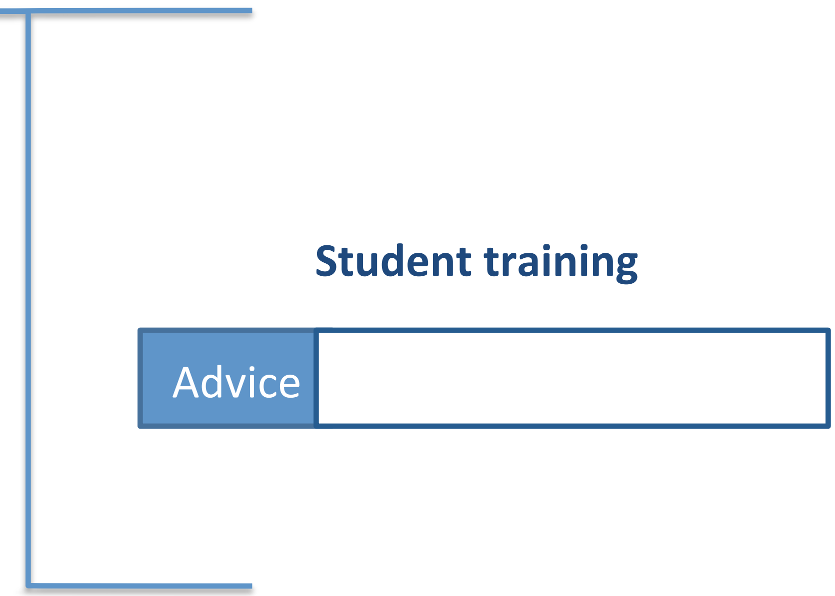
Main question:
How can the teacher spend its advice budget most effectively

# Proposed solutions

- Early advising
- Importance advising
- Mistake correcting
- Predictive advising

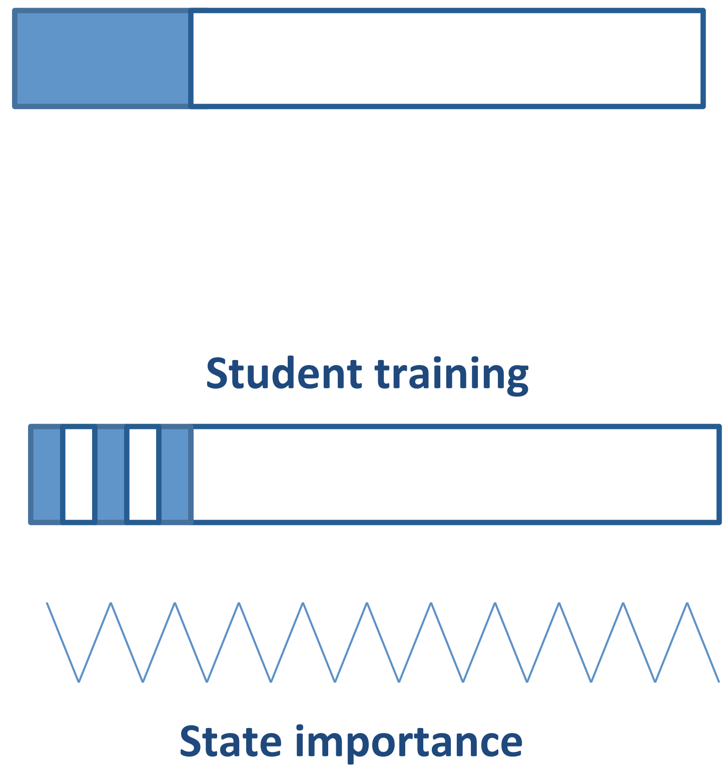
# Proposed solutions

- **Early advising**
- Importance advising
- Mistake correcting
- Predictive advising



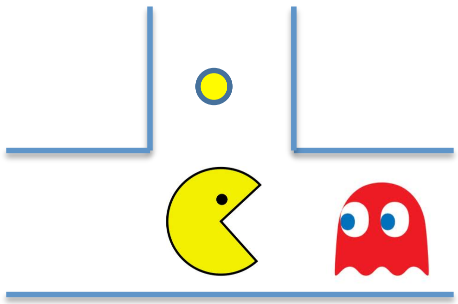
# Proposed solutions

- Early advising
- **Importance advising**
- Mistake correcting
- Predictive advising

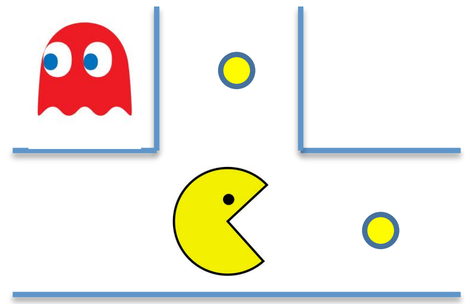


# State importance

High



Low



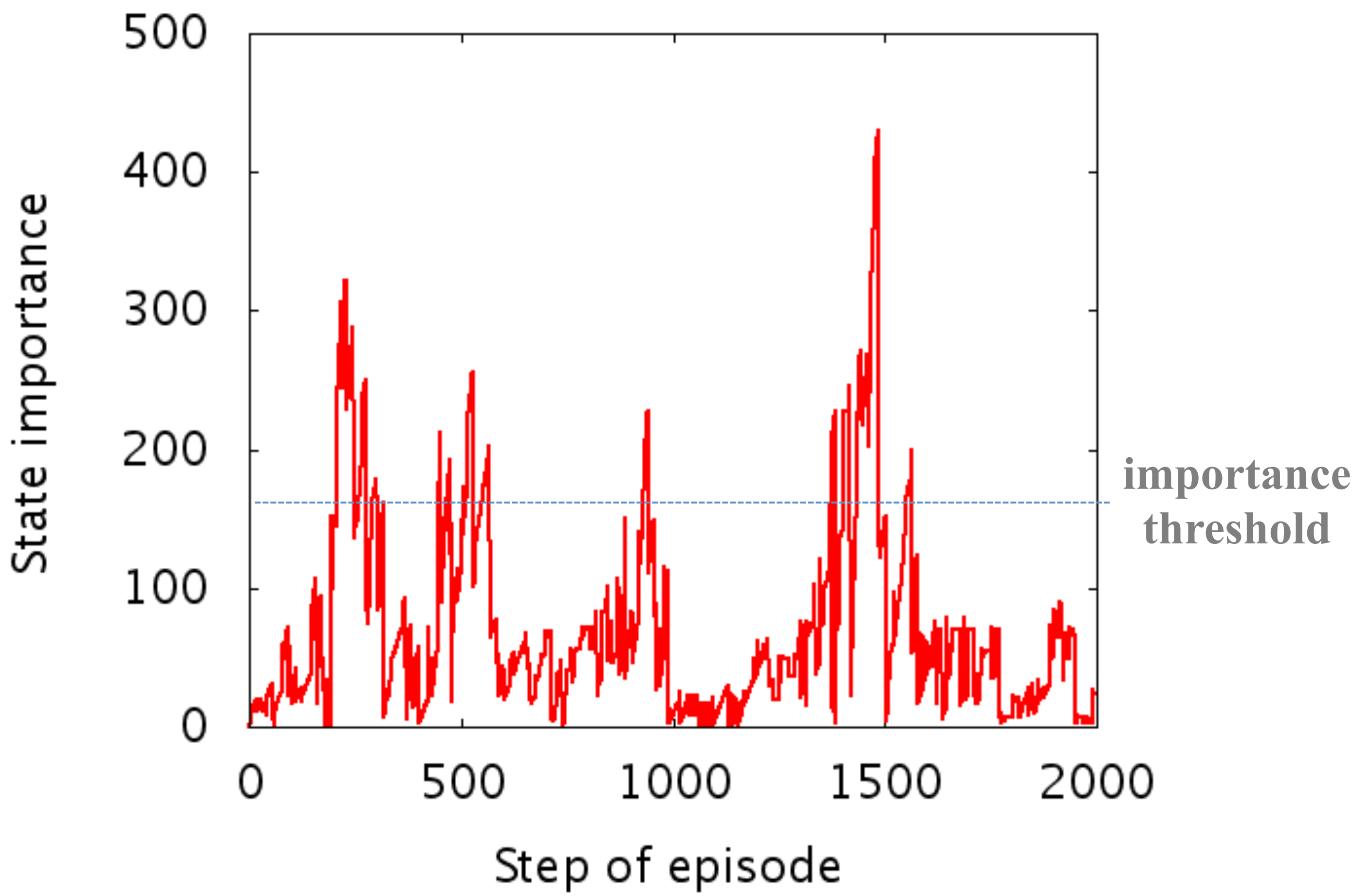
**Teacher knowledge**

$Q(s,a) \approx$ Return from taking action *a* in state *s*

**Importance metric**

$$I(s) = \max_a Q(s,a) - \min_a Q(s,a)$$

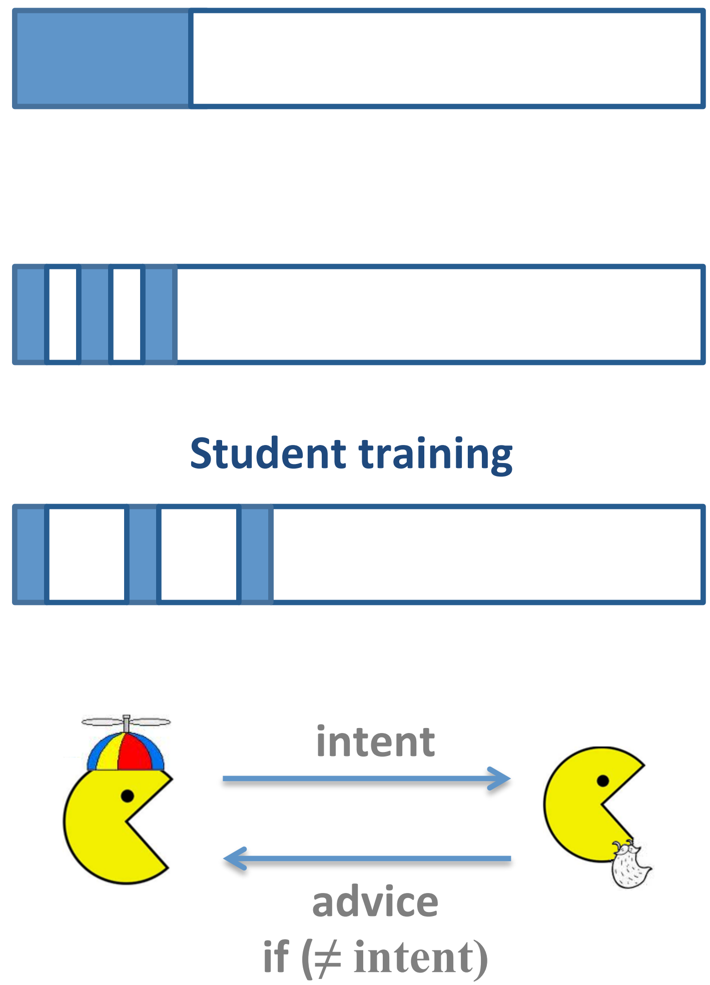
# In Pac-Man



500
400
300
200
100
0
State importance
0
500
1000
1500
2000
Step of episode
importance threshold

# Proposed solutions

- Early advising
- Importance advising
- **Mistake correcting**
- Predictive advising



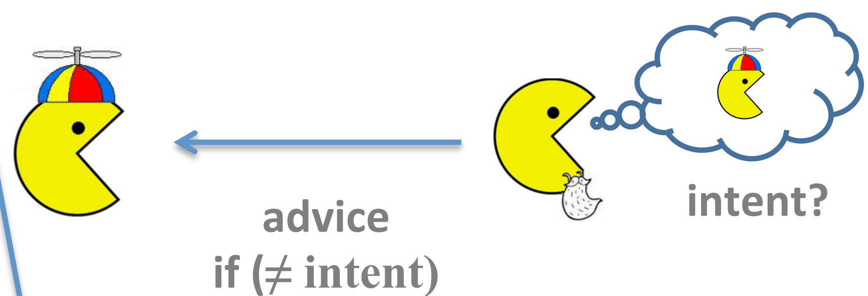
# Proposed solutions

- Early advising

- Importance advising

- Mistake correcting

- **Predictive advising**



# Proposed solutions

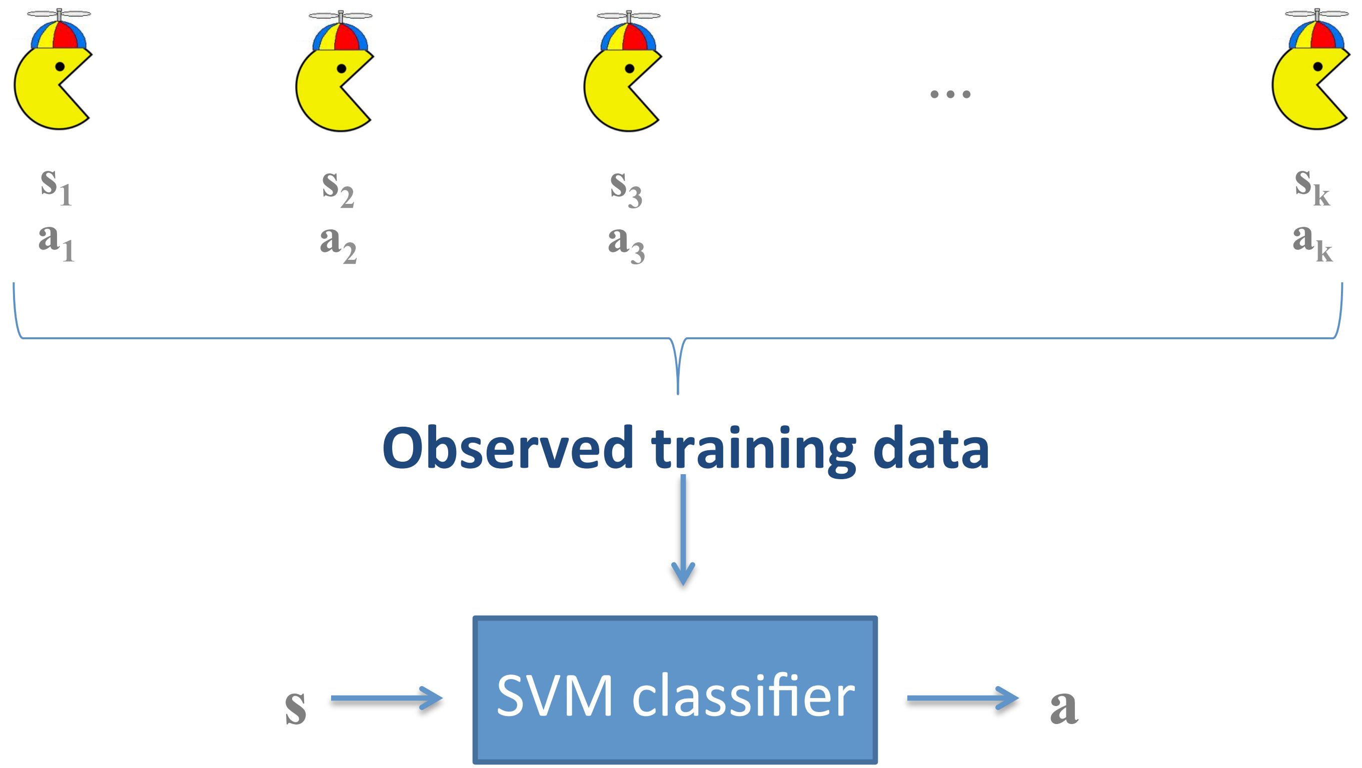
- Early advising

- Importance advising

- Mistake correcting

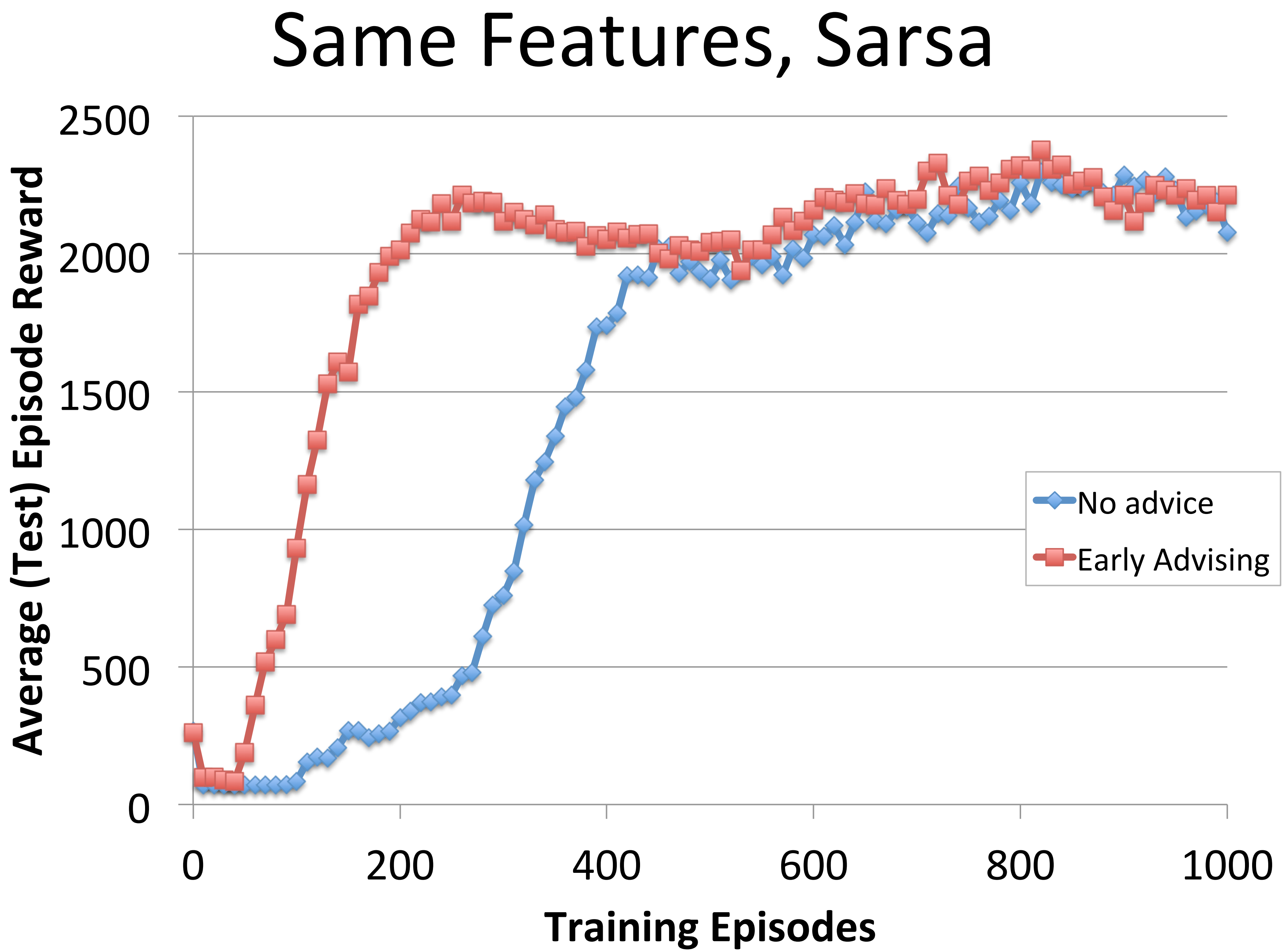
- Predictive advising

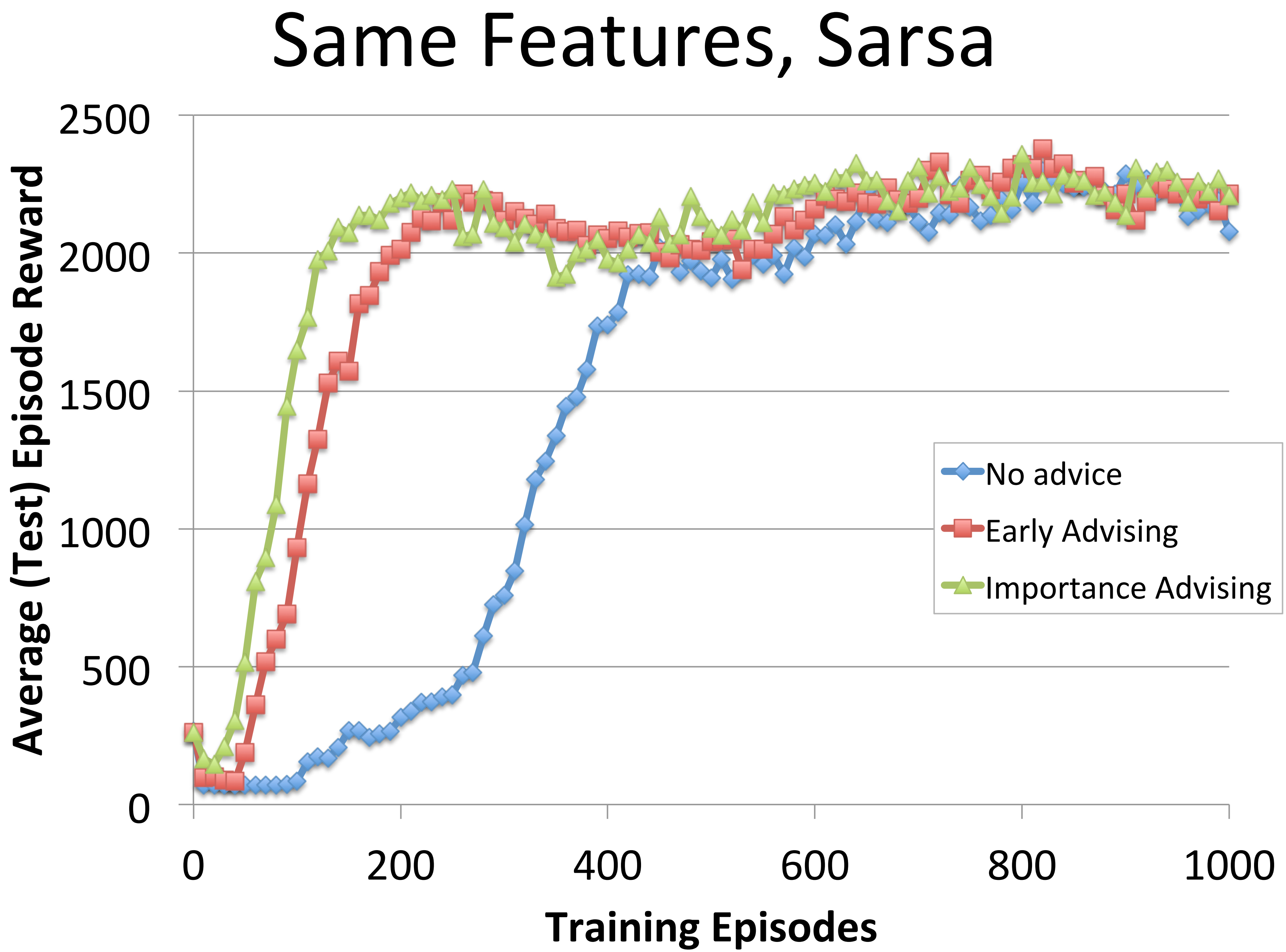
# Predicting intent



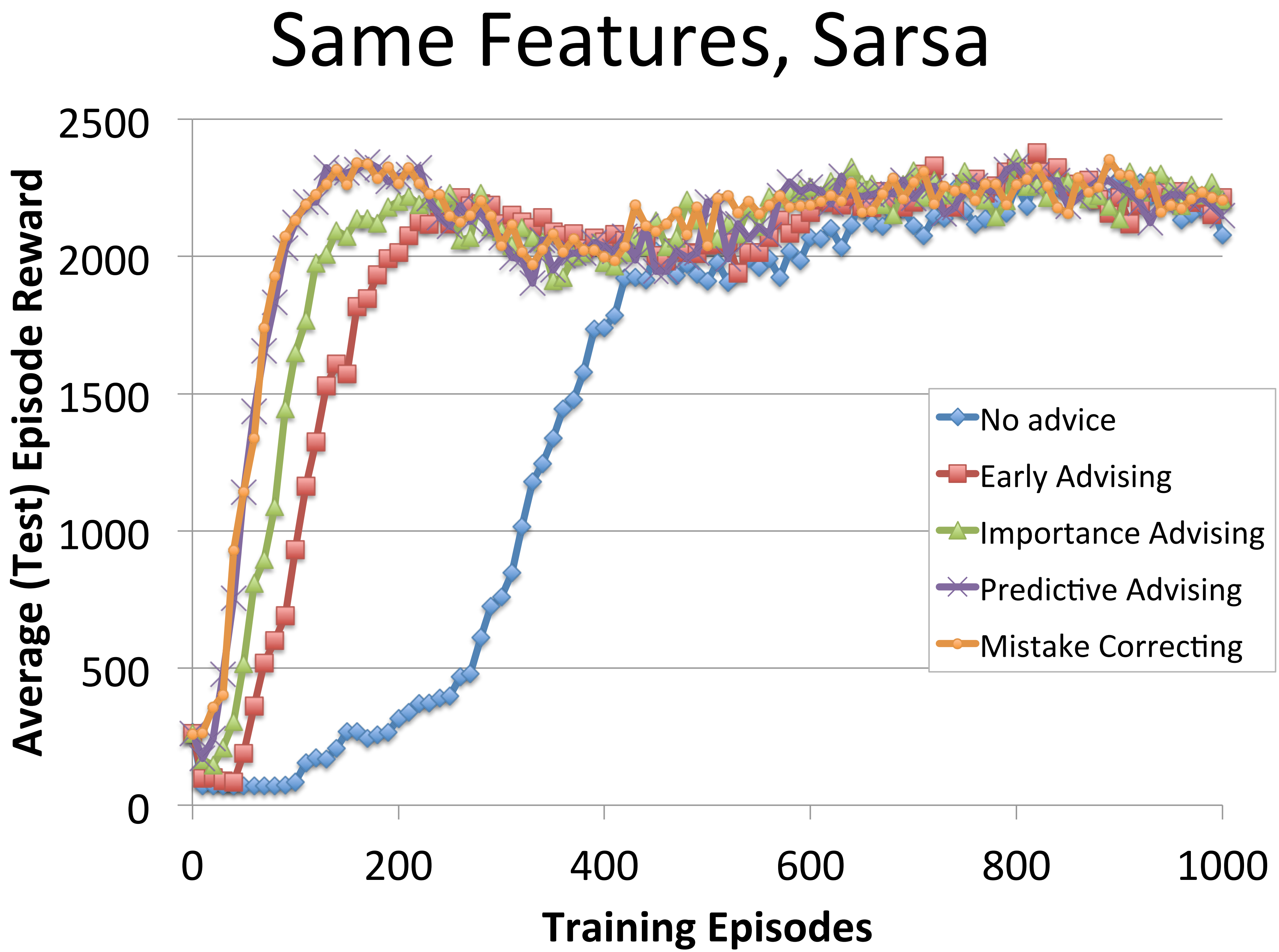
# Agent Variations

- Learning algorithms
  - Q-learning
  - SARSA
- Feature sets
  - Low-asymptote (initial state description)
  - High-asymptote (more useful features)


Same Features, Sarsa
Average (Test) Episode Reward
Training Episodes
0
500
1000
1500
2000
2500
0
200
400
600
800
1000
No advice
Early Advising



Same Features, Sarsa
Average (Test) Episode Reward
2500
2000
1500
1000
500
0
0
200
400
600
800
1000
Training Episodes
No advice
Early Advising
Importance Advising



Same Features, Sarsa
Average (Test) Episode Reward
Training Episodes
0
500
1000
1500
2000
2500
0
200
400
600
800
1000
No advice
Early Advising
Importance Advising
Predictive Advising
Mistake Correcting

- Current Work
  - Apply same techniques to teaching humans
  - Provide regret bounds depending on teacher's abilities

- Future Work
  - Multiple teachers
  - More differences between agents
  - When to ignore teacher
  - Definitions of state importance

# Agents Teaching Agents

- Transfer learning is great, if have full access to source agent

- Student learning can be improved with a small advice budget
- Advice has greater impact when spent on important states
- Advice has greater impact when spent on mistakes

- Teachers can improve student learning even when agents
  - have different learning algorithms
  - state representations
  - Can outperform teachers
- Mountain Car, Pac-Man, StarCraft

# Part VI: Humans Teaching Agents


- Gabriel V. de la Cruz Jr., Bei Peng, Walter S. Lasecki, and Matthew E. Taylor. Towards Integrating Real-Time Crowd Advice with Reinforcement Learning. IUI-15.
- W. Bradley Knox and Peter Stone. Reinforcement Learning from Simultaneous Human and MDP Reward. AAMAS-12.
- W. Bradley Knox and Peter Stone. Combining Manual Feedback with Subsequent MDP Reward Signals for Reinforcement Learning. AAMAS-10.
- W. Bradley Knox and Peter Stone. Interactively Shaping Agents via Human Reinforcement: The TAMER Framework. KCAP-09.
- W. Bradley Knox, Matthew Taylor, and Peter Stone. Understanding Human Teaching Modalities in Reinforcement Learning Environments: A Preliminary Report. ALIGHT workshop (at IJCAI-11).
- Robert Loftin, Bei Peng, James MacGlashan, Michael L. Littman, Matthew E. Taylor, Jeff Huang, and David L. Roberts. Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. Journal of Autonomous Agents and Multi-Agent Systems, pages 1-30, 2015.
- Robert Loftin, Bei Peng, James MacGlashan, Michael Littman, Matthew E. Taylor, David Roberts, and Jeff Huang. Learning Something from Nothing: Leveraging Implicit Human Feedback Strategies. RO-MAN-14.
- James Macglashan, Michael L. Littman, Robert Loftin, Bei Peng, David Roberts, and Matthew E. Taylor. Training an Agent to Ground Commands with Reward and Punishment. AAAI-14.

# Learning from feedback (interactive shaping)

Knox+, 2008-2013

TAMER

Key insight: trainer evaluates behavior using a model of its long-term quality

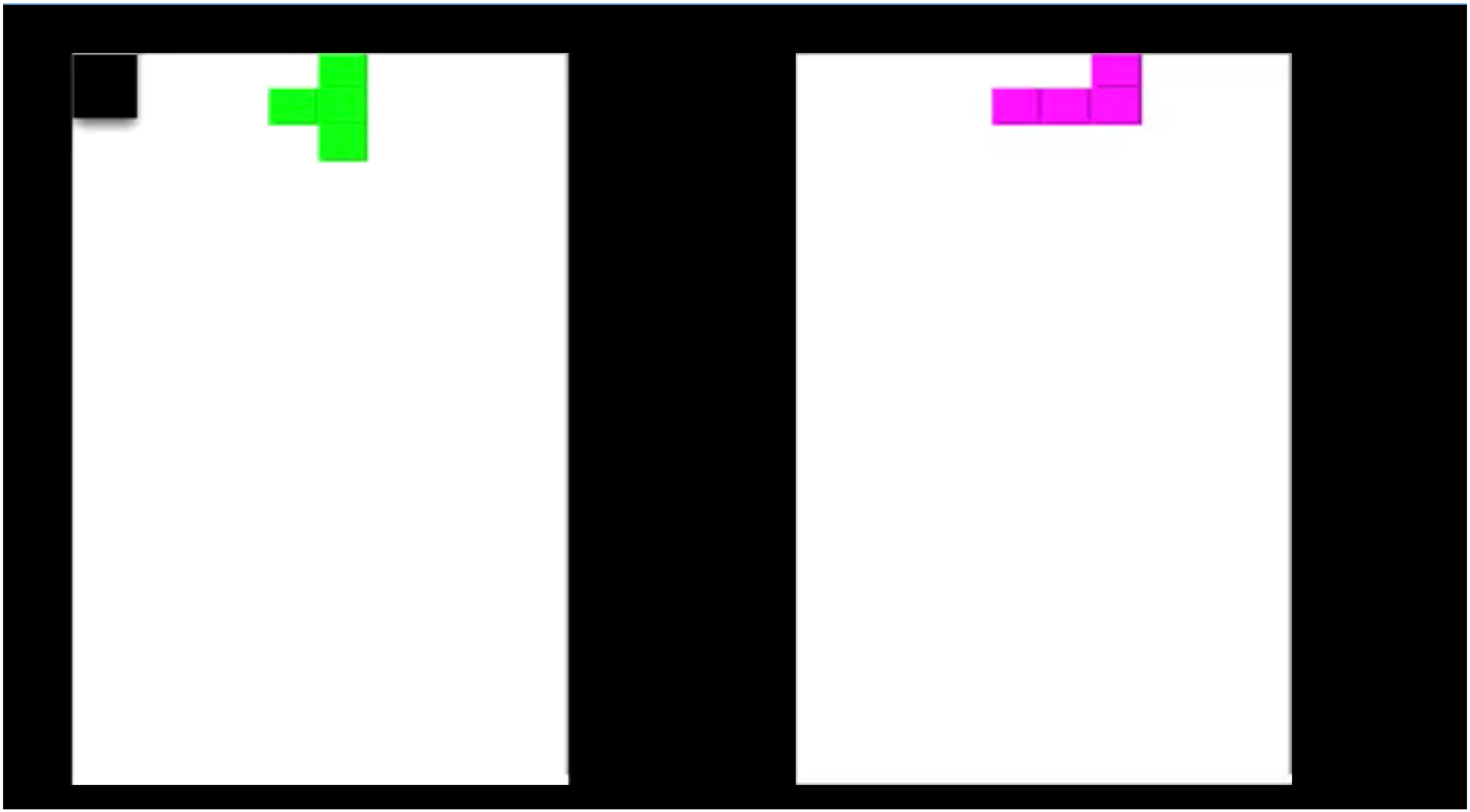
Learn a model of human reinforcement

$$H : S \times A \to \mathbb{R}$$

Directly exploit the model to determine action
Also, can combine with MDP's reward

# Tetris

During Training After 2 games of training



# a priori comparison

## Demonstration more specifically points to the correct action

**Interface**

- LfD interface may be familiar to video game players
- LfF interface is simpler and task-independent


**Task expertise**

- LfF - easier to judge than to control
- Easier for human to increase  expertise while training with LfD

**Cognitive load**

- Less for LfF

# Bayesian Inference Approach

- Here, feedback is categorical
- Use Bayesian approach
  - Find *maximum a posteriori* (MAP) estimate of target behavior

- *Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning*, Loftin+, JAAMAS-15

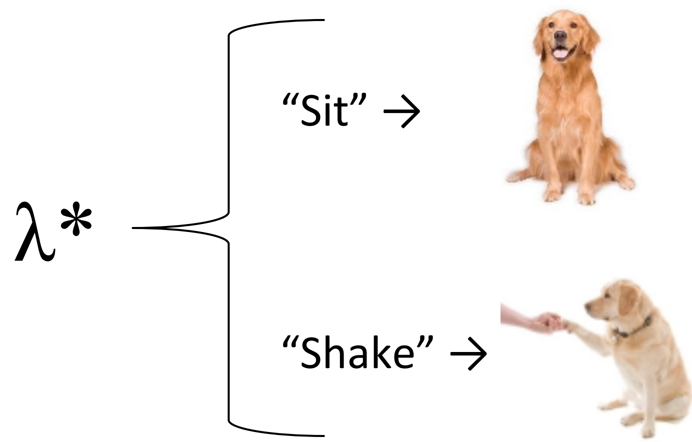
# Goal

- Human can give positive or negative feedback
- Agent tries to learn policy $\lambda^*$
- Maps observations to actions

- For now: think contextual bandit

# Example: Dog Training

- Teach dog to sit & shake




- Mapping from observations to actions
- Feedback: {Bad Dog, Good Boy}

# History in Dog Training

Feedback history *h*

- Observation: “sit”, Action:  Feedback: “Bad Dog”
- Observation: “sit”, Action:  Feedback: “Good Boy”
- …

Really make sense to assign numeric rewards to these?

# Bayesian Framework

- Trainer desires policy $\lambda^*$
- $h_t$ is the training history at time $t$
- Find MAP hypothesis of $\lambda^*$:

$$\operatorname*{argmax}_{\lambda} p(\lambda^* = \lambda | h_t) = \operatorname*{argmax}_{\lambda} \underbrace{p(h_t | \lambda^* = \lambda)}_{\text{Model of training process}} \underbrace{p(\lambda^* = \lambda)}_{\text{Prior distribution over policies}}$$

# Assumed trainer behavior

- Decide if action is correct
  - Does $\lambda^*(o)=a$ ? Trainer makes an error with $p(\varepsilon)$
- Decide if should give feedback
  - $\mu^+$, $\mu^-$ are probabilities of neutral feedback
  - If thinks correct, give positive feedback with $p(1- \mu^+)$
  - If thinks incorrect, give negative feedback with $p(1- \mu^-)$
- Could depend on trainer

# Feedback Probabilities

Probability of feedback $l_t$ at time $t$ is:

$$p(l_t = l^+ | o_t, a_t, \lambda^*) = \begin{cases} (1-\epsilon)(1-\mu^+) & , \lambda^*(o_t) = a_t \\ \epsilon(1-\mu^+) & , \lambda^*(o_t) \neq a_t \end{cases}$$

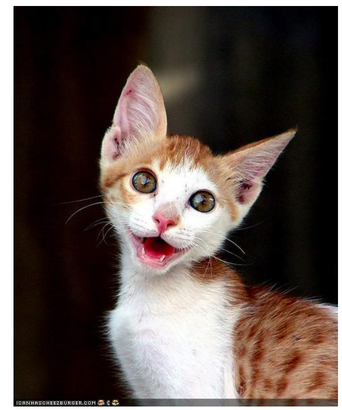
$$p(l_t = l^0 | o_t, a_t, \lambda^*) = \begin{cases} (1-\epsilon)\mu^+ + \epsilon\mu^- & , \lambda^*(o_t) = a_t \\ \epsilon\mu^+ + (1-\epsilon)\mu^- & , \lambda^*(o_t) \neq a_t \end{cases}$$

$$p(l_t = l^- | o_t, a_t, \lambda^*) = \begin{cases} \epsilon(1-\mu^-) & , \lambda^*(o_t) = a_t \\ (1-\epsilon)(1-\mu^-) & , \lambda^*(o_t) \neq a_t . \end{cases}$$

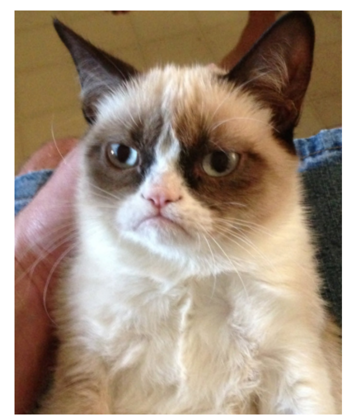
# Inferring Neutral

- Try to learn $\mu^+$ and $\mu^-$
- Don't assume they're equal

- Many trainers don't use punishment
  - Neutral feedback could be punishment



- Some don't use reward
  - Neutral feedback could be reward



# EM step

$$\lambda_{i+1} = \underset{\lambda \in P}{\operatorname{argmax}} \int_0^1 \int_0^1 p(\mu^+, \mu^- | h, \lambda_i) \ln p(h, \mu^+, \mu^- | \lambda) d\mu^+ d\mu^-$$

- Where $\lambda_i$ is *i*th estimate of maximum likelihood hypothesis
- Can simplify this (eventually) to:

$$\lambda_{i+1}(o) = \underset{a \in A}{\operatorname{argmax}}(\alpha(p_{o,a} - n_{o,a}) + \beta u_{o,a})$$

- $\alpha$ has to do with the value of neutral feedback (relative to $|\beta|$)
- $\beta$ is negative when neutral implies punishment and positive when implies reward

# User Study

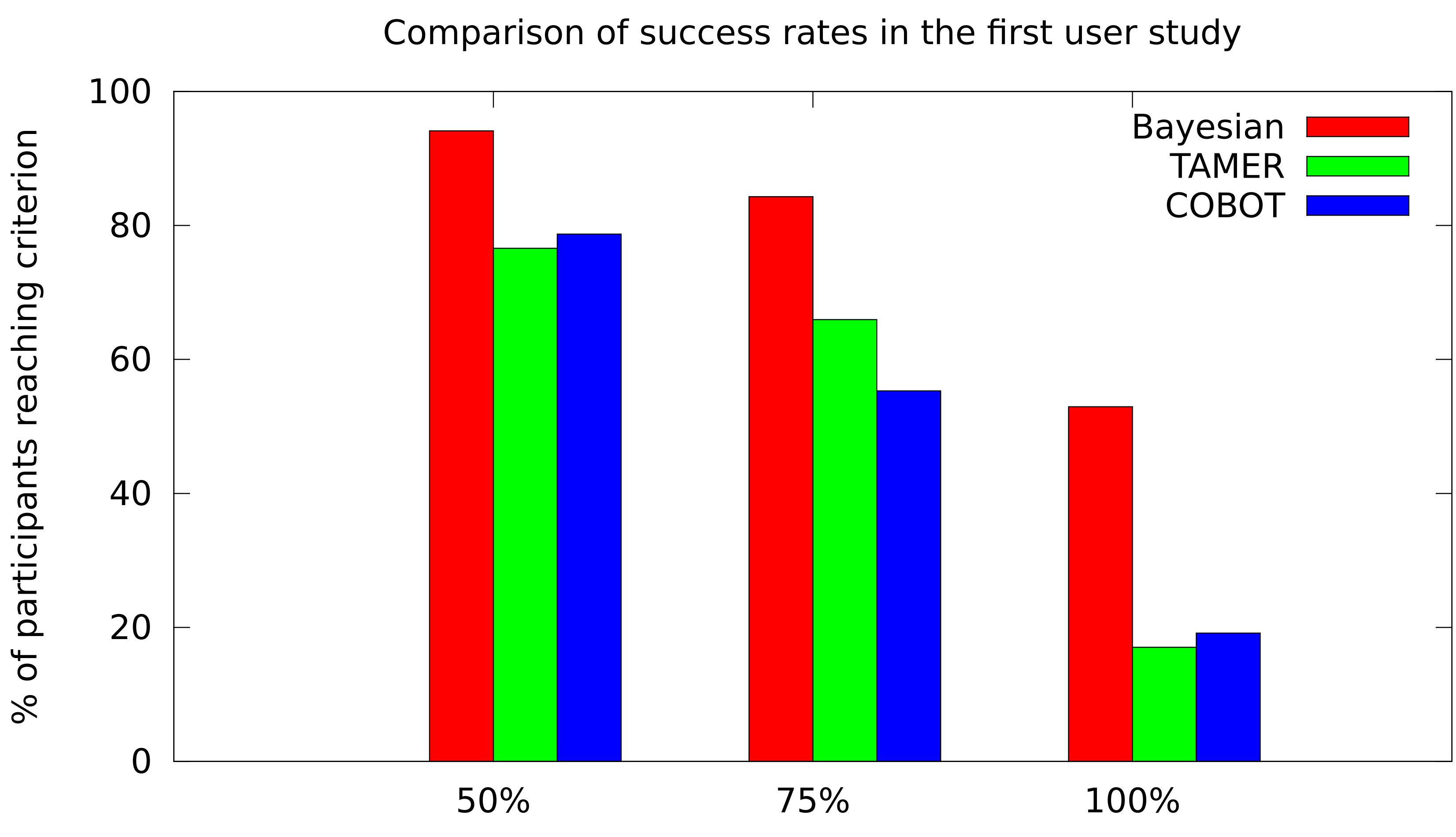
BEGIN TRAINING

LEARNING COMPLETE

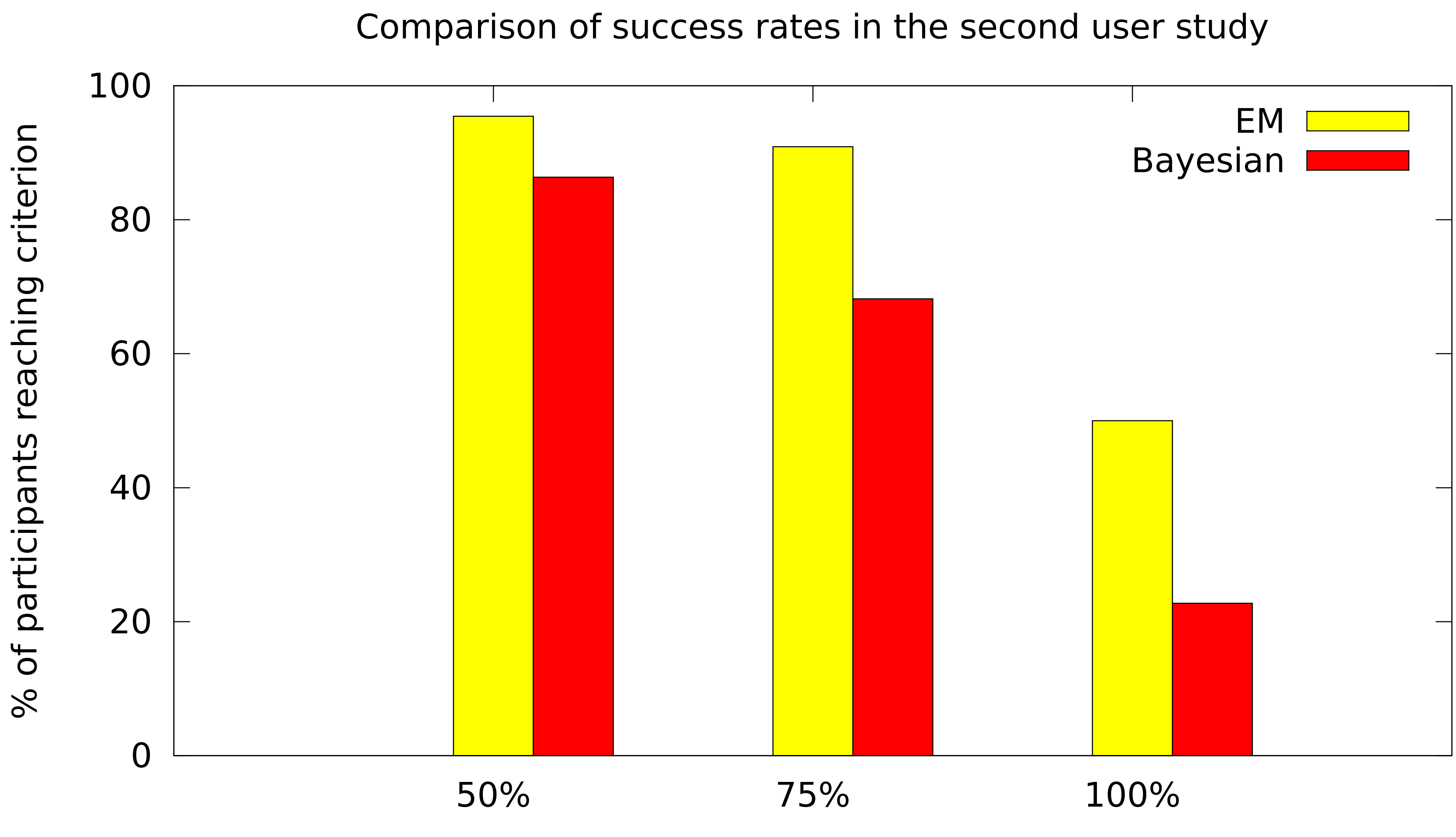
Once a rat reaches the corn field, it will disappear

# Comparisons

- Sim-TAMER
  - Numerical reward function
  - Zero ignored
  - No delay assumed
- Sim-COBOT
  - Similar to Sim-TAMER
  - Doesn’t ignore zero rewards




Categorical Feedback outperforms Numeric Feedback




Leveraging Neutral Improves Performance

# Mechanical Turk Studies

For our third study we posted three Human Intelligence Tasks to Amazon Mechanical Turk.

The Dog/Rat sprites, and three other sprite pairs (right) were used.

A total 211 users participated in the Mechanical Turk studies.

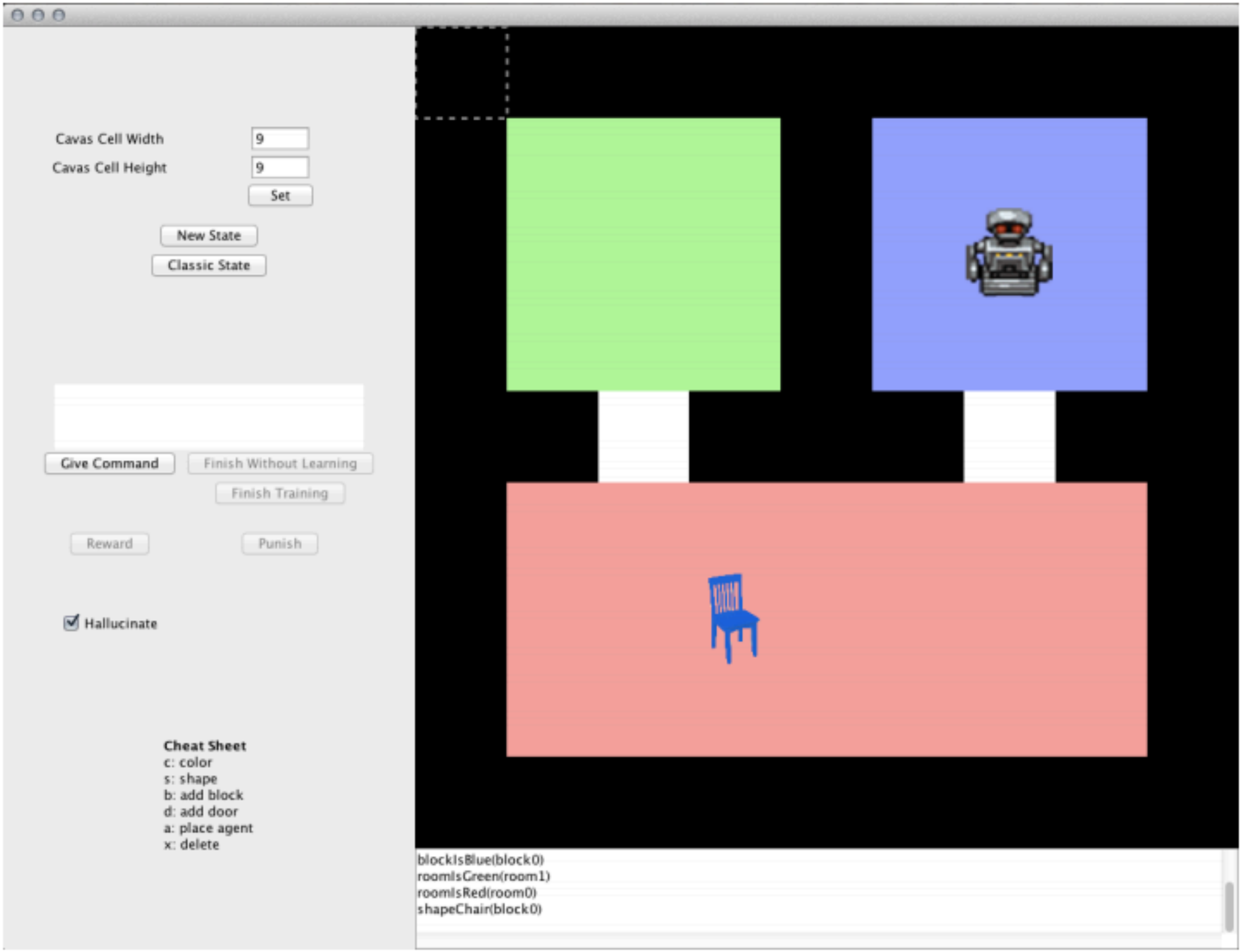
Users were paid $0.75 for participating, with a $0.25bonus for training performance.

Alternative Sprites:

# Effects of Agent Appearance

Distribution of strategies used in the Mechanical Turk study when training agents appearing as a dog, robot, snake or arrow.

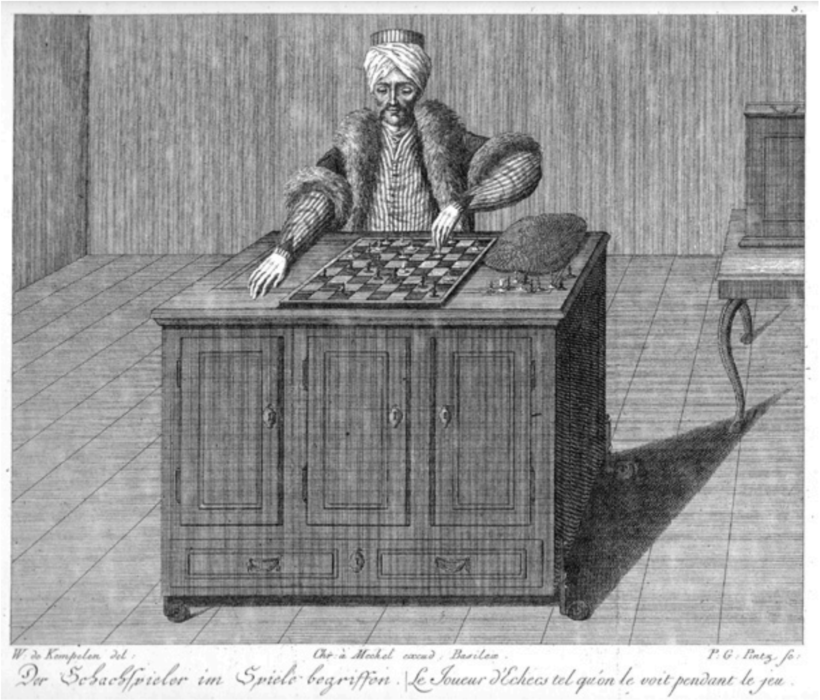
| Agent Sprite | Target Sprite | R+/P+ | R+/P− | R−/P+ | R−/P− |
|---|---|---|---|---|---|
| dog | rat | 151(85%) | 25(14%) | 1(.5%) | 1(.5%) |
| robot | battery | 188(88%) | 21(10%) | 0(0%) | 4(2%) |
| snake | bird | 64(84%) | 7(9%) | 2(3%) | 3(4%) |
| arrow | box | 43(83%) | 6(11%) | 1(2%) | 2(4%) |


Cavas Cell Width
9
Cavas Cell Height
9
Set
New State
Classic State
Give Command
Finish Without Learning
Finish Training
Reward
Punish
Hallucinate
Cheat Sheet
c: color
s: shape
b: add block
d: add door
a: place agent
x: delete
blockIsBlue(block0)
roomIsGreen(room1)
roomIsRed(room0)
shapeChair(block0)

- Current Work
  - Sequential tasks
  - Simultaneously learning language model

- Future Work
  - How do people want to teach?
  - How do people sequence tasks?
  - Automated training sequences?

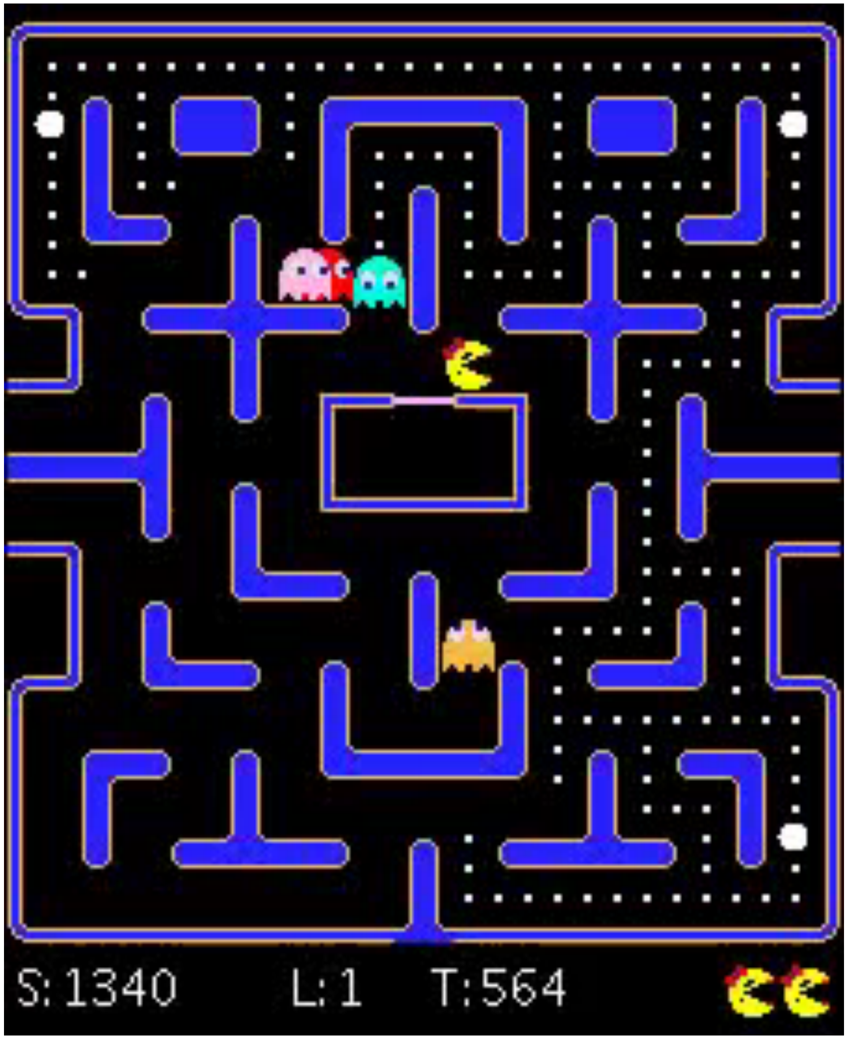
# RL + Crowdsourcing

Unlikely to be experts
May not take task seriously
May intentionally act poorly

*Towards Integrating Real-Time Crowd Advice with Reinforcement Learning*, de la Cruz+, IUI-15

# Crowd can identify "forced errors"




4 Distinct Experiments

| | Mistake Identification | Action Suggestion |
|---|---|---|
| Review | | |
| Real-time | | |

Current work: Leveraging Crowd Advice

- Reward Shaping (e.g., Brys+, AAAI-15, AAMAS-15, IJCAI-15)
- Learning from domonstration ideas (e.g., HAT)
- Bias action selection

Future Work

- Collecting the Crowd's Advice
  - Real-time System
  - Cyclic review system
  - Integrating multiple responses
  - Weigh by workers competence
- Generalize to other domains?
- Physical robots?

- LfD is great if have expert and lots of time
  - How to improve autonomously on few demonstrations?
- What about teaching like dog?
- Task sequencing?
- Leveraging crowd?

# Conclusions

- RL is awesome
- Faster RL is awesomer
- What other ways are there to bias agents and their exploration?

*irll.eecs.wsu.edu*

*eecs.wsu.edu/~taylorm*

Tim Brys and Matthew E. Taylor