Speeding up a Reinforcement Learning

Tim Brys and Matthew E. Taylor
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 \( MDP \) \( M \langle S, A, T, R \rangle \)
  - State space \( S \), Action space \( A \)
  - State transition probabilities \( T : S \times A \times S \rightarrow \mathbb{R} \)
  - A reward function \( R : S \times A \times S \rightarrow \mathbb{R} \)
Reinforcement Learning

- Goal: learn a policy \( \pi : S \times A \rightarrow \mathbb{R} \) 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

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

Reward Shaping


• 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


- 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


- 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

\[ \Phi - \text{update} \]

- Simple to implement
- Cheap to maintain

Algorithm

\[ s, a, s' \]

\[ r \]

\[ r + f \]

\[ \Phi \text{-update} \]

\[ -f_a \]

\[ Q \text{-update} \]
Shaping in Cart Pole

![Graph showing comparison between Baseline, Static shaping, and Dynamic shaping in Cart Pole. The x-axis represents Episodes ranging from 0 to 100, and the y-axis represents Steps ranging from 0 to 10,000. The Baseline curve is represented by a dotted line, Static shaping by a dashed line, and Dynamic shaping by a solid line. The graph illustrates the performance improvement of Dynamic shaping over the other two methods.]
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 of Reinforcement Learning Problems by Reward Shaping


- Transform MDP into MOMDP

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

- Add different potential-based reward shaping to each copy of the original reward

\[ R = [R + F_0, R + F_1, \ldots, R + F_n] \]

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


- Ensemble decision (for n decision makers):

\[
\arg\max_a \sum_{i}^{n} w_i p_i(s, a)
\]
Confidence Ensemble

Shaping Selection in State Space

Proximity  Trapping  Separation
Choice of Heuristic and Scaling


- For each heuristic, include multiple differently scaled versions in the ensemble
Learning the Shaping On-line


- 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


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


Learning from Demonstration History

Programming by Demonstration
• Demonstration play-back
• No generalization
• Sensitive to noise and variability
Learning from Demonstration History

Generalization over multiple demonstrations
- Symbolic abstraction (e.g., “close-to”, “above”)
- Hand-coded parameters
Learning from Demonstration History

- Programming by Demonstration
  - Generalization over multiple demonstrations
- Use of Machine Learning to analyze demonstrations
  - Generalization to novel states
  - Improved demonstration interfaces
  - Biologically inspired learning

Timeline:
- 1980: Programming by Demonstration
- 1990: Generalization over multiple demonstrations
- 2000: Use of Machine Learning to analyze demonstrations
Learning from Demonstration


- 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 RLfdD

Steps to Goal

Phase One Training Runs | Phase Two Training Runs

"optimal"

best example
HAT


• 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
and outperform the human teacher. The performance of the different methods improved after each episode; all four of the RL-based learning methods improve the average duration of the teacher’s demonstration.

Figure 3 shows the average duration of the teacher's demonstration over 10 independent trials. Using 20 episodes of transferring the learner running time of roughly 2.5 hours) to ensure convergence, learning algorithms were executed for 30 simulator hours (presenting results of the best performance achieved by the learner. This value is computed by using a sliding window over the past episodes to account for the high degree of noise in the Keepaway domain.

The best performance is measured as the average episode reward (corresponding to the total reward accumulated by an agent (i.e., the area under the learning curve) may also be improved. This metric measures the average episode reward at training hour, as defined in the keepaway task [23]:

\[
\text{Average Reward} = \frac{1}{T} \sum_{t=1}^{T} R_t
\]

where the experiment lasts one integral hour of training:

\[
T = \frac{1}{10} \times 60 \times 60 \times 24 = 36000
\]

This section presents results showing that HAT can effectively use human demonstration to bootstrap RL in Keepaway agents. Throughout this paper, t-tests are used to calculate significance, while in Section 5.1 will explore how performance changes with different types or amounts of demonstration, while Section 5.2 discusses transfer rules (No Prior), using the Value Bonus, using the Extra Action, and using Probabilistic Policy Reuse.}

### Table 1: Comparison of HAT and Human Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Total Reward</th>
<th>Final Reward</th>
<th>Jumpstart</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Prior</td>
<td>14.3</td>
<td>16.0</td>
<td>N/A</td>
</tr>
<tr>
<td>Value Bonus</td>
<td>15.1</td>
<td>15.1</td>
<td>0.57</td>
</tr>
<tr>
<td>Extra Action</td>
<td>14.8</td>
<td>15.0</td>
<td>0.42</td>
</tr>
<tr>
<td>Probabilistic Policy Reuse</td>
<td>16.0</td>
<td>16.0</td>
<td>0.36</td>
</tr>
<tr>
<td>Teacher Demonstration</td>
<td>16.0</td>
<td>16.0</td>
<td>0.36</td>
</tr>
</tbody>
</table>

The table shows the total reward, final reward, and jumpstart for different methods compared to human results. Statistical significance is indicated by the t-test results.
RLfD through Shaping


• 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

128 demonstration samples in Cart Pole

Steps

Episode

RL
RLfD (shaping)
RLfD (HAT)
LfD
RLfD through Shaping

Varying demonstration length in Cart Pole

Final performance

Demonstration length

RL
RLfD (shaping)
RLfD (HAT)
LfD
Part IV: Transfer Learning


• Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. Online multi-task learning for policy gradient methods. ICML-14


• 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

Value Function Transfer

\[ \rho(Q(S,A)) = Q'(S',A') \]

\( \rho \) is task-dependant:
relies on inter-task mappings

Source

Target

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 emit 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.”
target

sources

Kernel-based Model Approximation

compliance

relevance

transfer of samples
Inter-Task Mappings

- $\chi_x: s_{\text{target}} \rightarrow s_{\text{source}}$
  - Given state / state variable in target task
  - Return corresponding state / state variable in source task

- $\chi_A: a_{\text{target}} \rightarrow a_{\text{source}}$
  - Similar, but for actions

- **Intuitive** mappings exist in some domains (Oracle)

- Used to construct $\rho$

![Diagram showing inter-task mappings between source and target domains](image)
Transferring Instances for Model-Based Reinforcement Learning, Taylor+., 2008

TIMBREL

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

Instance-based method

Assumes you know the (correct) inter-task mapping

n. An ancient percussion instrument similar to a tambourine
TIMBREL

if target task model \((T \text{ or } R)\) 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 \(R\)
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)

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
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?

### 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 $\pi(s,a)$
• 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


Reinforcement Learning Agents Providing Advice in Complex Video Games


• Different state representation
• Different learning methods
• Only action advice
• Limited amounts of advice
Reinforcement Learning + Teaching

\[ \pi_T \quad \cdots \quad \text{teacher} \]

\[ \pi_S \quad \cdots \quad \text{student} \]

\[ \text{advice} \quad \cdots \quad \text{state} \]

\[ \text{environment} \]

\[ \text{state} \quad \cdots \quad \text{action} \]

\[ \text{action} \quad \cdots \quad \text{reward} \]
Why Action Advice?

Transfer learning

\[ \pi_T \]

Requirements
- Direct access
- High similarity

Teaching via advice

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

Teacher knowledge

\[ Q(s, a) \approx \text{Return from taking action } a \text{ in state } s \]

Importance metric

\[ I(s) = \max_a Q(s, a) - \min_a Q(s, a) \]
In Pac-Man

![Graph showing state importance over steps of episode with a dotted line indicating the importance threshold.](image)
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

Observed training data

\[ s_1 \quad a_1 \]
\[ s_2 \quad a_2 \]
\[ s_3 \quad a_3 \]
\[ \ldots \]
\[ s_k \quad a_k \]

\[ s \quad \text{SVM classifier} \quad a \]
Agent Variations

• Learning algorithms
  – Q-learning
  – SARSA

• Feature sets
  – Low-asymptote (initial state description)
  – High-asymptote (more useful features)
Same Features, Sarsa

![Graph showing average test episode reward over training episodes with two lines: one for No advice and another for Early Advising.]
Same Features, Sarsa

![Graph showing the comparison of average test episode reward with training episodes for Sarsa with different advising methods. The x-axis represents training episodes ranging from 0 to 1000, and the y-axis represents average test episode reward ranging from 0 to 2500. Three lines are plotted: blue for No advice, red for Early Advising, and green for Importance Advising. The graph shows how the advising methods affect the convergence and performance of the algorithm.]
Same Features, Sarsa

Average (Test) Episode Reward vs. Training Episodes

- 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

- W. Bradley Knox and Peter Stone. Reinforcement Learning from Simultaneous Human and MDP Reward. AAMAS-12.
Learning from feedback (interactive shaping)

Knox+, 2008-2013

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

Learn a model of human reinforcement

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

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

During Training  After 2 games of training

During Training: A game of training with a green piece on the left side.

After 2 games of training: A game of training with a pink piece on the right side.
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

\[ \lambda^* \]

  “Sit” →

  “Shake” →

• Mapping from observations to actions
• Feedback: \{Bad Dog, Good Boy\}
History in Dog Training

Feedback history $h$

- ...

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 $\lambda^*$:

$$\arg\max_{\lambda} p(\lambda^* = \lambda | h_t) = \arg\max_{\lambda} p(h_t | \lambda^* = \lambda)p(\lambda^* = \lambda)$$

Model of training process       Prior distribution over policies
Assumed trainer behavior

• Decide if action is correct
  – Does $\lambda^*(o) = a$? Trainer makes an error with $p(\epsilon)$

• 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

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

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

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

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

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 $0.25 bonus for training performance.
Effects of Agent Appearance

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

<table>
<thead>
<tr>
<th>Agent Sprite</th>
<th>Target Sprite</th>
<th>R+/P+</th>
<th>R+/P−</th>
<th>R−/P+</th>
<th>R−/P−</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog robot</td>
<td>rat</td>
<td>151(85%)</td>
<td>25(14%)</td>
<td>1(.5%)</td>
<td>1(.5%)</td>
</tr>
<tr>
<td></td>
<td>battery</td>
<td>188(88%)</td>
<td>21(10%)</td>
<td>0(0%)</td>
<td>4(2%)</td>
</tr>
<tr>
<td>snake arrow</td>
<td>bird</td>
<td>64(84%)</td>
<td>7(9%)</td>
<td>2(3%)</td>
<td>3(4%)</td>
</tr>
<tr>
<td></td>
<td>box</td>
<td>43(83%)</td>
<td>6(11%)</td>
<td>1(2%)</td>
<td>2(4%)</td>
</tr>
</tbody>
</table>
• 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”
Current work: Leveraging Crowd Advice
- Reward Shaping (e.g., Brys+, AAAI-15, AAMAS-15, IJCAI-15)
- Learning from demonstration 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