Practical RL: Representation, Interaction, Synthesis, and Mortality (PRISM)

Peter Stone

Director, Learning Agents Research Group (LARG)  
Department of Computer Science  
The University of Texas at Austin

Joint work with members of LARG past and present
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Robotics
- Multiagent systems
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Robotics
- Multiagent systems
- Machine learning
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Robotics
- Multiagent systems
- Machine learning
  - Reinforcement Learning
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Robotics
- Multiagent systems
- Machine learning
  - Reinforcement Learning
Reinforcement Learning

Supervised learning mature [WEKA]
Reinforcement Learning

Supervised learning **mature** [WEKA]

For agents, reinforcement learning most appropriate
Reinforcement Learning

Supervised learning mature [WEKA]

For agents, reinforcement learning most appropriate

```
Agent
Policy π: S→A
```

```
Environment
```

state (s[t])

reward (r[t+1])

action (a[t])

```
4
3
1
?
```
Reinforcement Learning

Supervised learning mature [WEKA]

For agents, reinforcement learning most appropriate

Environment  \[ \text{Agent} \]

\[
\begin{align*}
\text{Policy } \pi : S &\rightarrow A \\
\text{state (s[t])} &\quad \text{reward (r[t+1])} \\
\text{action (a[t])}
\end{align*}
\]
Reinforcement Learning

Supervised learning mature [WEKA]

For agents, reinforcement learning most appropriate

- Foundational theoretical results
- Applications require innovations to scale up
RL Theory

Success story: **Q-learning** converges to $\pi^*$ [Watkins, 89]
Success story: **Q-learning** converges to $\pi^*$ [Watkins, 89]

- Table-based representation
- Visit every state infinitely often
Function Approximation

In practice, visiting every state impossible

\[ s[t-1] a[t-1] \]

\[ a[t] \]
Function Approximation

In practice, visiting every state impossible

Function approximation of value function

Theoretical guarantees harder to come by
Batch Methods

In practice, often experience is scarce
In practice, often experience is scarce

Batch Methods

Save transitions:

\[ s[t−1], a[t−1] \]

\[ <r[i], s[i], a[i]> \text{ for } i=0 \text{ to } t−1 \]
Applications: Towards a Useful Tool

- Backgammon [Tesauro, ’94]
- Helicopter control [Ng et al., ’03]
Applications: Towards a Useful Tool

- Backgammon [Tesauro, ’94]
- Helicopter control [Ng et al., ’03]

- Adaptive treatment of epilepsy [Pineau et al., ’08]

- Invasive species management, wildfire suppression [Dietterich et al., ’13]
Applications: Towards a Useful Tool

- Backgammon [Tesauro, ’94]
- Helicopter control [Ng et al., ’03]

- Adaptive treatment of epilepsy [Pineau et al., ’08]

- Invasive species management, wildfire suppression [Dietterich et al., ’13]
- Google Purchase of DeepMind, ’14
RL as a Tool

To learn

Finite MDP
RL as a Tool

To learn
Finite MDP
State Aliasing
Generalization

To apply
RL as a Tool

- To learn
  - Finite MDP
  - State Aliasing
  - Generalization

To apply

To learn
Rather than “Should RL work?” ...
Rather than “**Should** RL work?” . . .

. . . “**Does** RL work?”
Rather than “**Should** RL work?” . . .

. . . “**Does** RL work?”

– When not: “**How** can we make it work?”
Practical RL

- Representation
- Interaction
- Synthesis
- Mortality
**Practical RL**

- **Representation**
  - Selecting the Algorithm: *parameterized domains* [K.&S., MLJ 2011]
  - Adapting Representation: *NEAT+Q* [Whiteson & S., JMLR 2006]

- **Interaction**

- **Synthesis**

- **Mortality**
Practical RL

**Representation**
- Selecting the Algorithm: parameterized domains [K.&S., MLJ 2011]
- Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]

**Interaction**
- With adversaries: CMLES [Chakraborty & S., ICML 2010]
- With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
- With people: TAMER [Knox & S., AAMAS 2010]

**Synthesis**

**Mortality**
Practical RL

**Representation**
- Selecting the Algorithm: parameterized domains [K.&S., MLJ 2011]
- Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]

**Interaction**
- With adversaries: CMLES [Chakraborty & S., ICML 2010]
- With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
- With people: TAMER [Knox & S., AAMAS 2010]

**Synthesis**

**Mortality**
Practical RL

**Representation**
- Selecting the Algorithm: parameterized domains [K.
  & S., MLJ 2011]
- Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]

**Interaction**
- With adversaries: CMLES [Chakraborty & S., ICML 2010]
- With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
- With people: TAMER [Knox & S., AAMAS 2010]

**Synthesis**

**Mortality**
- Leverage the Past: Transfer Learning [Taylor, S., & Liu, JMLR 2007]
- Acknowledge a Finite Future: TEXPLORE [Hester & S., MLJ 2013]
Practical RL

**Representation**
- Selecting the Algorithm: parameterized domains [K.&S., MLJ 2011]
- Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]

**Interaction**
- With adversaries: CMLEs [Chakraborty & S., ICML 2010]
- With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
- With people: TAMER [Knox & S., AAMAS 2010]

**Synthesis**

**Mortality**
- Leverage the Past: Transfer Learning [Taylor, S., & Liu, JMLR 2007]
- Acknowledge a Finite Future: TEXPLORE [Hester & S., MLJ 2013]
Practical RL

**Representation**
- Selecting the Algorithm: parameterized domains [K.&S., MLJ 2011]
- Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]

**Interaction**
- With adversaries: CMLES [Chakraborty & S., ICML 2010]
- With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
- With people: TAMER [Knox & S., AAMAS 2010]

**Synthesis**

**Mortality**
- Leverage the Past: Transfer Learning [Taylor, S., & Liu, JMLR 2007]
- Acknowledge a Finite Future: TEXPLORE [Hester & S., MLJ 2013]
Practical RL

- **Representation**
  - Selecting the Algorithm: parameterized domains [K.&S., MLJ 2011]
  - Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]

- **Interaction**
  - With adversaries: CMLES [Chakraborty & S., ICML 2010]
  - With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
  - With people: TAMER [Knox & S., AAMAS 2010]

- **Synthesis**

- **Mortality**
  - Leverage the Past: Transfer Learning [Taylor, S., & Liu, JMLR 2007]
  - Acknowledge a Finite Future: TEXPLORE [Hester & S., MLJ 2013]
TEXPLORE: Real-Time Sample-Efficient Reinforcement Learning for Robots

Todd Hester and Peter Stone

Machine Learning, 2013
Reinforcement Learning

Model-free Methods

- Learn a value function directly from interaction with environment
- Can run in real-time, but not very sample efficient
Reinforcement Learning

Model-free Methods
- Learn a value function directly from interaction with environment
- Can run in real-time, but not very sample efficient

Model-based Methods
- Learn model of transition and reward dynamics
- Update value function using model (planning)
- Can update action-values without taking real actions in the world
Mortality

- Robot’s “lifetime” short compared to size of world
Mortality

- Robot’s “lifetime” short compared to size of world
- (Still need to act in real time)
Mortality

Robot’s “lifetime” short compared to size of world
(Still need to act in real time)

Problem: Cannot explore everywhere

Choose where not to explore
Idea: Learn multiple possible models and compare them
Only explore states that are both uncertain in model and promising for final policy
Most of state space is very predictable
But fuel stations have varying costs
Want to explore mainly fuel stations, and particularly ones on short path to goal
Fuel World Behavior

- Agent explores randomly at first
- Agent focuses its exploration on fuel stations near the shortest path to the goal, trying a different fuel station each episode.
- Agent finds near-optimal policies.
Vehicle upgraded to run **autonomously** by adding shift-by-wire, steering, and braking actuators.

- 10 second episodes (at 20 Hz: 200 samples / episode)
Velocity Control

- State:
  - Current Velocity
  - Desired Velocity
  - Accelerator Pedal Position
  - Brake Pedal Position

- Actions:
  - Do nothing
  - Increase/decrease brake position by 0.1
  - Increase/decrease accelerator position by 0.1

- Reward: $-10.0 \times$ velocity error (m/s)
Desiderata

1. Algorithm must learn in very few actions (be sample efficient)
2. Algorithm must act continually in real-time (while learning)
3. Algorithm must handle continuous state
4. Algorithm must handle delayed actions
Desiderata

1. Algorithm must learn in very few actions (be **sample efficient**)
2. Algorithm must act **continually** in real-time (while learning)
3. Algorithm must handle **continuous** state
4. Algorithm must handle **delayed** actions
Common Approaches

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Citation</th>
<th>Sample Efficient</th>
<th>Real Time</th>
<th>Continuous</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-Max</td>
<td>Brafman 2001</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Q-Learning</td>
<td>Watkins 1989</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>with F.A.</td>
<td>Sutton &amp; Barto 1998</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SARSA</td>
<td>Rummery &amp; Niranjan 1994</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GPRL</td>
<td>Deisenroth &amp; Rasmussen 2011</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>BOSS</td>
<td>Asmuth et al 2009</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Bayesian DP</td>
<td>Strens 2000</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MBBE</td>
<td>Dearden et al 1999</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MBS</td>
<td>Walsh et al 2009</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Dyna</td>
<td>Sutton 1990</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
## Common Approaches

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Citation</th>
<th>Sample</th>
<th>Real Time</th>
<th>Continuous</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-Max</td>
<td>Brafman 2001</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Q-Learning</td>
<td>Watkins 1989</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>with F.A.</td>
<td>Sutton &amp; Barto 1998</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SARSA</td>
<td>Rummery &amp; Niranjan 1994</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GPRL</td>
<td>Deisenroth &amp; Rasmussen 2011</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>BOSS</td>
<td>Asmuth et al 2009</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Bayesian DP</td>
<td>Strens 2000</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MBBE</td>
<td>Dearden et al 1999</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MBS</td>
<td>Walsh et al 2009</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Dyna</td>
<td>Sutton 1990</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
The TEXPLORE Algorithm

1. Limits exploration to be **sample efficient**
2. Selects actions continually in **real-time**
3. Handles **continuous** state
4. Handles actuator **delays**

Available publicly as a **ROS package**:

[www.ros.org/wiki/rl-texplore-ros-pkg](http://www.ros.org/wiki/rl-texplore-ros-pkg)
Challenge 1: Sample Efficiency
Challenge 1: Sample Efficiency

- Treat **model learning** as a supervised learning problem
  - **Input:** State and Action
  - **Output:** Distribution over next states and reward

```
+-------------------+
| x > 3             |
+-------------------+
| y < 7             |
+-------------------+           +-------------------+
|                  0 |           |                  2 |
+-------------------+           +-------------------+   +-------------------+   +-------------------+   +-------------------+   +-------------------+   +-------------------+   +-------------------+
                                           |                  2 |           |                  1 |
                                           +-------------------+           +-------------------+   +-------------------+   +-------------------+   +-------------------+   +-------------------+
```
**Challenge 1: Sample Efficiency**

- **Treat model learning** as a supervised learning problem
  - **Input:** State and Action
  - **Output:** Distribution over next states and reward

- **Factored model:** Learn a separate model to predict each next state feature and reward
Challenge 1: Sample Efficiency

- Treat **model learning** as a supervised learning problem
  - **Input:** State and Action
  - **Output:** Distribution over next states and reward
- **Factored** model: Learn a separate model to predict each next state feature and reward
- **Decision Trees:** Split state space into regions with similar dynamics
Random Forest Model

- **Average predictions** of $m$ different decision trees
- Each tree represents a **hypothesis** of the true dynamics of the domain
Random Forest Model

- **Average predictions** of $m$ different decision trees
- Each tree represents a **hypothesis** of the true dynamics of the domain
- Acting greedily w.r.t. the average model **balances** predictions of optimistic and pessimistic models
- **Limits** the agent’s exploration to state-actions that appear promising, while avoiding those which may have negative outcomes
Challenge 2: Real-Time Action Selection

- Model update can take too long
- Planning can take too long
Real-Time Model Based Architecture (RTMBA)

- Model learning and planning on parallel threads
Real-Time Model Based Architecture (RTMBA)

- Model learning and planning on parallel threads
- Action selection is not restricted by their computation time
Real-Time Model Based Architecture (RTMBA)

- Model learning and planning on parallel threads
- Action selection is not restricted by their computation time
- Use sample-based planning (anytime)
Real-Time Model Based Architecture (RTMBA)

- Model learning and planning on parallel threads
- Action selection is not restricted by their computation time
- Use sample-based planning (anytime)
- Mutex locks on shared data
Challenge 3: Continuous State

- Use regression trees to model continuous state
- Each tree has a linear regression model at its leaves
Challenge 3: Continuous State

- Use regression trees to model continuous state
- Each tree has a linear regression model at its leaves
- Discretize state space for value updates from UCT, but still plan over continuously valued states
Challenge 4: Actuator Delays

- Delays make domain non-Markov, but k-Markov
Challenge 4: Actuator Delays

- Delays make domain non-Markov, but k-Markov
- Provide model with previous $k$ actions (Similar to U-Tree [McCallum 1996])
- Trees can learn which delayed actions are relevant
Challenge 4: Actuator Delays

- Delays make domain non-Markov, but k-Markov
- Provide model with previous $k$ actions (Similar to U-Tree [McCallum 1996])
- Trees can learn which delayed actions are relevant
- UCT can plan over augmented state-action histories easily
Autonomous Vehicle

- Upgraded to run **autonomously** by adding shift-by-wire, steering, and braking actuators.
- Vehicle runs at 20 Hz.
- Agent **must** provide commands at this frequency.
Uses ROS [Quigley et al 2009]

http://www.ros.org/wiki/rl_msgs
Simulation Experiments

Exploration Approaches
- Epsilon-Greedy
- Boltzmann Exploration
- Use merged BOSS-like model
- Use random model each episode

Sample Efficient Methods
- BOSS [Asmuth et al 2009]
- Bayesian DP [Strens 2000]
- Gaussian Process RL [Deisenroth & Rasmussen 2011]
Simulation Experiments

Continuous Models
- Tabular Models
- Gaussian Process RL [Deisenroth & Rasmussen 2011]
- KWIK linear regression [Strehl & Littman 2007]

Real-Time Architectures
- Real Time Dynamic Dynamic Programming [Barto et al 1995]
- Dyna [Sutton 1990]
- Parallel Value Iteration

Actuator Delays
- Model Based Simulation [Walsh et al 2009]
Challenge 1: Sample Efficiency

Simulated Car Control Between Random Velocities

Average Reward vs. Episode Number

- TEXPLORE (Greedy)
- BOSS
- Bayesian DP
- GPRL
- R-Max
- Q-Learning with Tile-Coding
Challenge 2: Real-Time Action Selection

Simulated Car Control Between Random Velocities

Average Reward vs. Episode Number

- RTMBA (TEXPLORE)
- RTDP
- Parallel VI
- Value Iteration
- RT-Dyna
- Q-Learning Tile-Coding

Peter Stone (UT Austin)
Challenge 3: Modeling Continuous Domains

Model Accuracy on Next State

Model Accuracy on Reward
Challenge 4: Handling Delayed Actions

Simulated Car Control Between Random Velocities

Average Reward vs. Episode Number

TEXPLORE $k=0$
TEXPLORE $k=1$
TEXPLORE $k=2$
TEXPLORE $k=3$
MBS $k=1$
MBS $k=2$
MBS $k=3$
Tabular $k=2$

Peter Stone (UT Austin)
On the physical vehicle

But, does it work on the actual vehicle?
On the physical vehicle

Yes! It learns the task within 2 minutes of driving time
TEXPLORE Summary

- TEXPLORE can:
  1. Learn in few samples
  2. Act continually in real-time
  3. Learn in continuous domains
  4. Handle actuator delays

- TEXPLORE code has been released as a ROS package:
  www.ros.org/wiki/rl-texplore-ros-pkg
Practical RL

- **Representation**
  - Selecting the Algorithm: parameterized domains [K.&S., MLJ 2011]
  - Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]

- **Interaction**
  - With adversaries: CMSLES [Chakraborty & S., ICML 2010]
  - With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
  - With people: TAMER [Knox & S., AAMAS 2010]

- **Synthesis**

- **Mortality**
  - Leverage the Past: Transfer Learning [Taylor, S., & Liu, JMLR 2007]
  - Acknowledge a Finite Future: TEXPLORE [Hester & S., MLJ 2013]
UT Austin Villa 2014
RoboCup 3D Simulation League Champion via Overlapping Layered Learning

Patrick MacAlpine, Mike Depinet, and Peter Stone

AAAI, 2015
Layered Learning

- For domains too complex for tractably mapping state features $S \mapsto O$
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
Layered Learning

- For domains too complex for tractably mapping state features $S \mapsto O$
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: exploit data to train, adapt
Layered Learning

- For domains too complex for tractably mapping state features $S \rightarrow \text{outputs } O$
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: exploit data to train, adapt
- Synthesis: Learning in one layer feeds into next layer
Layered Learning

- For domains too complex for tractably mapping state features $S \rightarrow \text{outputs } O$
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: exploit data to train, adapt
- **Synthesis**: Learning in one layer feeds into next layer

```
High Level Goals
  Adversarial Behaviors
  Team Behaviors
  Multi-Agent Behaviors
  Individual Behaviors
    World State
      Environment
```

Peter Stone (UT Austin)
Layered Learning in Practice

First applied in simulated robot soccer [Stone & Veloso, ’97]

<table>
<thead>
<tr>
<th>Strategic Level</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>individual</td>
</tr>
<tr>
<td></td>
<td>ball interception</td>
</tr>
<tr>
<td>$L_2$</td>
<td>multiagent</td>
</tr>
<tr>
<td></td>
<td>pass evaluation</td>
</tr>
<tr>
<td>$L_3$</td>
<td>team</td>
</tr>
<tr>
<td></td>
<td>pass selection</td>
</tr>
</tbody>
</table>
Layered Learning in Practice

First applied in simulated robot soccer [Stone & Veloso, ’97]

<table>
<thead>
<tr>
<th>Strategic Level</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>individual ball interception</td>
</tr>
<tr>
<td>$L_2$</td>
<td>multiagent pass evaluation</td>
</tr>
<tr>
<td>$L_3$</td>
<td>team pass selection</td>
</tr>
</tbody>
</table>

Later applied on real robots [Stone, Kohl, & Fidelman, ’06]

<table>
<thead>
<tr>
<th>Strategic Level</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>individual fast walking</td>
</tr>
<tr>
<td>$L_2$</td>
<td>individual ball control</td>
</tr>
</tbody>
</table>
Layered Learning Paradigms

**DESCRIPTIONS:**

**Sequential Layered Learning:** Freeze parameters of layer after learning before learning of the next layer

**Concurrent Layered Learning:** Keep parameters of layer open during learning of the next layer
Layered Learning Paradigms

PROBLEMS:

**Sequential Layered Learning:** Can be too limiting in the joint layer policy search space

**Concurrent Layered Learning:** The increased dimensionality can make learning harder or intractible
Layered Learning Paradigms

**SOLUTION:**

**Overlapping Layered Learning:** Tradeoff between freezing or keeping open previous learned layers

Optimizes “seam” or overlap between behaviors: keeps some parts of previously learned layers open during subsequent learning
Overlapping Layered Learning

Sequential Layered Learning (SLL)

Concurrent Layered Learning (CLL)

Combining Independently Learned Behaviors (CILB)

Partial Concurrent Layered Learning (PCLL)

Previous Learned Layer Refinement (PLLH)
Overlapping Layered Learning

Combining Independently Learned Behaviors (CILB)

Partial Concurrent Layered Learning (PCLL)

Previous Learned Layer Refinement (PLLRL)

Combining Independently Learned Behaviors: Behaviors learned independently and then combined by relearning subset of behaviors’ parameters
Overlapping Layered Learning

Combining Independently Learned Behaviors (CILB): Behaviors learned independently and then combined by relearning subset of behaviors’ parameters

Partial Concurrent Layered Learning (PCLL): Part, but not all, of a previously learned layer’s behaviors are left open
Overlapping Layered Learning

Combining Independently Learned Behaviors: Behaviors learned independently and then combined by relearning subset of behaviors’ parameters

Partial Concurrent Layered Learning: Part, but not all, of a previously learned layer’s behaviors are left open

Previous Learned Layer Refinement: After a pair of layers is learned, part or all of the initial layer is unfrozen
RoboCup 3D Simulation Domain

- Teams of 11 vs 11 autonomous robots play soccer
- Realistic physics using Open Dynamics Engine (ODE)
- Simulated robots modeled after Aldebaran Nao robot
- Robot receives noisy visual information about environment
- Robots can communicate over limited bandwidth channel
RoboCup Champions 2011, 2012
RoboCup Champions 2011, 2012

Humanoid Walk Learning via Layered Learning and CMA-ES

- Parameterized double linear inverted pendulum model
RoboCup Champions 2011, 2012

Humanoid Walk Learning via Layered Learning and CMA-ES

- Parameterized double linear inverted pendulum model

- Stochastic, derivative-free, numerical optimization method
- Candidates sampled from multidimensional Gaussian

CMA-ES [Hansen, ’09]
RoboCup Champions 2011, 2012

Humanoid Walk Learning via Layered Learning and CMA-ES

- Parameterized **double linear inverted pendulum model**

  CMA-ES
  [Hansen, ’09]

- Stochastic, derivative-free, numerical optimization method
- Candidates sampled from **multidimensional Gaussian**
  - **Mean** maximizes likelihood of previous successes
  - **Covariance** update controls search step sizes
RoboCup Champions 2011, 2012

Humanoid Walk Learning via Layered Learning and CMA-ES

• Parameterized double linear inverted pendulum model

CMA-ES
[Hansen, ’09]

• Stochastic, derivative-free, numerical optimization method
• Candidates sampled from multidimensional Gaussian
  – Mean maximizes likelihood of previous successes
  – Covariance update controls search step sizes

Initial walk
3 layers

No layered learning
Final walk

2 layers
Champs*2

Peter Stone (UT Austin)
Learned Layers

- 19 learned behaviors for standing up, walking, and kicking
  - CILB, PCLL, PLLR
- Over 500 parameters optimized during the course of learning
  - frozen, passed, seeded
Dribbling and Kicking the Ball in the Goal

- Four different walk parameter sets
  - Target/sprint/position + approach ball to kick
- Learn **fixed kick**
- Combine **kick with walk**: combine independent layers (**CILB**)
  - Overlap kick parameters for positioning
- Final **walk and kick**
**Scoring on a Kickoff**

- **Kickoffs indirect** (2 players must touch to score)
- Learn **fixed kick**
- Learn **touch** behavior **interferes**
- Combine **kick with touch**
  - Relearn position patterns: combine independent layers (CILB)
  - Learn new timing parameter: partial concurrent (PCLL)
Impact of Overlapping Layered Learning

1000 games vs. top 3 teams from 2013
## Impact of Overlapping Layered Learning

1000 games vs. top 3 teams from 2013

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Full Team</th>
<th>No Kickoff</th>
<th>Dribble Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>apollo3d</td>
<td>2.703 (0.041)</td>
<td>2.062 (0.038)</td>
<td>1.861 (0.034)</td>
</tr>
<tr>
<td>UTAustinVilla2013</td>
<td>1.589 (0.036)</td>
<td>1.225 (0.033)</td>
<td>0.849 (0.025)</td>
</tr>
<tr>
<td>fcportugal3d</td>
<td>3.991 (0.051)</td>
<td>3.189 (0.048)</td>
<td>1.584 (0.030)</td>
</tr>
</tbody>
</table>

**No Kickoff**: On kickoff, kick ball deep into opponent’s end  
**Dribble Only**: No kicking
Repetition on Different Robot Types

Type 0: Standard Nao model
Type 1: Longer legs and arms
Type 2: Quicker moving legs
Type 3: Wider hips and longest legs and arms
Type 4: Added toes to foot
Repetition on Different Robot Types

Type 0: Standard Nao model
Type 1: Longer legs and arms
Type 2: Quicker moving legs
Type 3: Wider hips and longest legs and arms
Type 4: Added toes to foot

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Type 0</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>apollo3d</td>
<td>1.787</td>
<td>1.819</td>
<td>1.820</td>
<td>1.543</td>
<td>2.827</td>
</tr>
<tr>
<td>UTAustinVilla2013</td>
<td>0.992</td>
<td>0.892</td>
<td>1.276</td>
<td>0.573</td>
<td>1.141</td>
</tr>
<tr>
<td>fcportugal3d</td>
<td>2.423</td>
<td>3.025</td>
<td>3.275</td>
<td>2.678</td>
<td>4.033</td>
</tr>
</tbody>
</table>
Repetition on Different Robot Types

Type 0: Standard Nao model
Type 1: Longer legs and arms
Type 2: Quicker moving legs
Type 3: Wider hips and longest legs and arms
Type 4: Added toes to foot

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Type 0</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>apollo3d</td>
<td>1.787</td>
<td>1.819</td>
<td>1.820</td>
<td>1.543</td>
<td>2.827</td>
</tr>
<tr>
<td>UTAustinVilla2013</td>
<td>0.992</td>
<td>0.892</td>
<td>1.276</td>
<td>0.573</td>
<td>1.141</td>
</tr>
<tr>
<td>fcportugal3d</td>
<td>2.423</td>
<td>3.025</td>
<td>3.275</td>
<td>2.678</td>
<td>4.033</td>
</tr>
</tbody>
</table>

Computation per type

≈ 700k parameter sets evaluated
≈ 1.5 years compute time (≈ 50 hours on condor cluster)
RoboCup 2014

Won competition with **undefeated** record: outscored opps 52–0

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Avg. Goal Diff.</th>
<th>Record (W-L-T)</th>
<th>Goals (F/A)</th>
<th>KO Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>BahiaRT</td>
<td>2.075 (0.030)</td>
<td>990-0-10</td>
<td>2092/17</td>
<td>96.2</td>
</tr>
<tr>
<td>FCPortugal</td>
<td>2.642 (0.034)</td>
<td>986-0-14</td>
<td>2748/106</td>
<td>83.4</td>
</tr>
<tr>
<td>magmaOffenburg</td>
<td>2.855 (0.035)</td>
<td>990-0-10</td>
<td>2864/9</td>
<td>88.3</td>
</tr>
<tr>
<td>RoboCanes</td>
<td>3.081 (0.046)</td>
<td>974-0-26</td>
<td>3155/74</td>
<td>69.4</td>
</tr>
<tr>
<td>FUT-K</td>
<td>3.236 (0.039)</td>
<td>998-0-2</td>
<td>3240/4</td>
<td>96.3</td>
</tr>
<tr>
<td>SEU_Jolly</td>
<td>4.031 (0.062)</td>
<td>995-0-5</td>
<td>4034/3</td>
<td>87.6</td>
</tr>
<tr>
<td>KarachiKoalas</td>
<td>5.681 (0.046)</td>
<td>1000-0-0</td>
<td>5682/1</td>
<td>87.5</td>
</tr>
<tr>
<td>ODENS</td>
<td>7.933 (0.041)</td>
<td>1000-0-0</td>
<td>7933/0</td>
<td>92.1</td>
</tr>
<tr>
<td>HfutEngine</td>
<td>8.510 (0.050)</td>
<td>1000-0-0</td>
<td>8510/0</td>
<td>94.7</td>
</tr>
<tr>
<td>Mithras3D</td>
<td>8.897 (0.041)</td>
<td>1000-0-0</td>
<td>8897/0</td>
<td>90.4</td>
</tr>
<tr>
<td>L3M-SIM</td>
<td>9.304 (0.043)</td>
<td>1000-0-0</td>
<td>9304/0</td>
<td>93.7</td>
</tr>
</tbody>
</table>
RoboCup 2014

Won competition with **undefeated** record: outscored opps 52–0

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Avg. Goal Diff.</th>
<th>Record (W-L-T)</th>
<th>Goals (F/A)</th>
<th>KO Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>BahiaRT</td>
<td>2.075 (0.030)</td>
<td>990-0-10</td>
<td>2092/17</td>
<td>96.2</td>
</tr>
<tr>
<td>FCPortugal</td>
<td>2.642 (0.034)</td>
<td>986-0-14</td>
<td>2748/106</td>
<td>83.4</td>
</tr>
<tr>
<td>magmaOffenburg</td>
<td>2.855 (0.035)</td>
<td>990-0-10</td>
<td>2864/9</td>
<td>88.3</td>
</tr>
<tr>
<td>RoboCanes</td>
<td>3.081 (0.046)</td>
<td>974-0-26</td>
<td>3155/74</td>
<td>69.4</td>
</tr>
<tr>
<td>FUT-K</td>
<td>3.236 (0.039)</td>
<td>998-0-2</td>
<td>3240/4</td>
<td>96.3</td>
</tr>
<tr>
<td>SEU_Jolly</td>
<td>4.031 (0.062)</td>
<td>995-0-5</td>
<td>4034/3</td>
<td>87.6</td>
</tr>
<tr>
<td>KarachiKoalas</td>
<td>5.681 (0.046)</td>
<td>1000-0-0</td>
<td>5682/1</td>
<td>87.5</td>
</tr>
<tr>
<td>ODENS</td>
<td>7.933 (0.041)</td>
<td>1000-0-0</td>
<td>7933/0</td>
<td>92.1</td>
</tr>
<tr>
<td>HfutEngine</td>
<td>8.510 (0.050)</td>
<td>1000-0-0</td>
<td>8510/0</td>
<td>94.7</td>
</tr>
<tr>
<td>Mithras3D</td>
<td>8.897 (0.041)</td>
<td>1000-0-0</td>
<td>8897/0</td>
<td>90.4</td>
</tr>
<tr>
<td>L3M-SIM</td>
<td>9.304 (0.043)</td>
<td>1000-0-0</td>
<td>9304/0</td>
<td>93.7</td>
</tr>
</tbody>
</table>

- **After: 11,000 games:** won all by 67 (no losses)
RoboCup 2014

Won competition with undefeated record: outscored opps 52–0

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Avg. Goal Diff.</th>
<th>Record (W-L-T)</th>
<th>Goals (F/A)</th>
<th>KO Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>BahiaRT</td>
<td>2.075 (0.030)</td>
<td>990-0-10</td>
<td>2092/17</td>
<td>96.2</td>
</tr>
<tr>
<td>FCPortugal</td>
<td>2.642 (0.034)</td>
<td>986-0-14</td>
<td>2748/106</td>
<td>83.4</td>
</tr>
<tr>
<td>magmaOffenburg</td>
<td>2.855 (0.035)</td>
<td>990-0-10</td>
<td>2864/9</td>
<td>88.3</td>
</tr>
<tr>
<td>RoboCanes</td>
<td>3.081 (0.046)</td>
<td>974-0-26</td>
<td>3155/74</td>
<td>69.4</td>
</tr>
<tr>
<td>FUT-K</td>
<td>3.236 (0.039)</td>
<td>998-0-2</td>
<td>3240/4</td>
<td>96.3</td>
</tr>
<tr>
<td>SEU_Jolly</td>
<td>4.031 (0.062)</td>
<td>995-0-5</td>
<td>4034/3</td>
<td>87.6</td>
</tr>
<tr>
<td>KarachiKoalas</td>
<td>5.681 (0.046)</td>
<td>1000-0-0</td>
<td>5682/1</td>
<td>87.5</td>
</tr>
<tr>
<td>ODENS</td>
<td>7.933 (0.041)</td>
<td>1000-0-0</td>
<td>7933/0</td>
<td>92.1</td>
</tr>
<tr>
<td>HfutEngine</td>
<td>8.510 (0.050)</td>
<td>1000-0-0</td>
<td>8510/0</td>
<td>94.7</td>
</tr>
<tr>
<td>Mithras3D</td>
<td>8.897 (0.041)</td>
<td>1000-0-0</td>
<td>8897/0</td>
<td>90.4</td>
</tr>
<tr>
<td>L3M-SIM</td>
<td>9.304 (0.043)</td>
<td>1000-0-0</td>
<td>9304/0</td>
<td>93.7</td>
</tr>
</tbody>
</table>

- After: **11,000 games**: won all by 67 (no losses)
- **Highlights** from Final vs. RoboCanes (University of Miami)
RoboCup 2014

Won competition with **undefeated** record: outscored opps 52–0

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Avg. Goal Diff.</th>
<th>Record (W-L-T)</th>
<th>Goals (F/A)</th>
<th>KO Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>BahiaRT</td>
<td>2.075 (0.030)</td>
<td>990-0-10</td>
<td>2092/17</td>
<td>96.2</td>
</tr>
<tr>
<td>FCPortugal</td>
<td>2.642 (0.034)</td>
<td>986-0-14</td>
<td>2748/106</td>
<td>83.4</td>
</tr>
<tr>
<td>magmaOffenburg</td>
<td>2.855 (0.035)</td>
<td>990-0-10</td>
<td>2864/9</td>
<td>88.3</td>
</tr>
<tr>
<td>RoboCanes</td>
<td>3.081 (0.046)</td>
<td>974-0-26</td>
<td>3155/74</td>
<td>69.4</td>
</tr>
<tr>
<td>FUT-K</td>
<td>3.236 (0.039)</td>
<td>998-0-2</td>
<td>3240/4</td>
<td>96.3</td>
</tr>
<tr>
<td>SEU_Jolly</td>
<td>4.031 (0.062)</td>
<td>995-0-5</td>
<td>4034/3</td>
<td>87.6</td>
</tr>
<tr>
<td>KarachiKoalas</td>
<td>5.681 (0.046)</td>
<td>1000-0-0</td>
<td>5682/1</td>
<td>87.5</td>
</tr>
<tr>
<td>ODENS</td>
<td>7.933 (0.041)</td>
<td>1000-0-0</td>
<td>7933/0</td>
<td>92.1</td>
</tr>
<tr>
<td>HfutEngine</td>
<td>8.510 (0.050)</td>
<td>1000-0-0</td>
<td>8510/0</td>
<td>94.7</td>
</tr>
<tr>
<td>Mithras3D</td>
<td>8.897 (0.041)</td>
<td>1000-0-0</td>
<td>8897/0</td>
<td>90.4</td>
</tr>
<tr>
<td>L3M-SIM</td>
<td>9.304 (0.043)</td>
<td>1000-0-0</td>
<td>9304/0</td>
<td>93.7</td>
</tr>
</tbody>
</table>

- **After:** 11,000 games: won all by 67 (no losses)
- **Highlights** from Final vs. RoboCanes (University of Miami)
- More info: [www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/](http://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/)
Practical RL

- **Representation**
  - Selecting the Algorithm: parameterized domains [K. & S., MLJ 2011]
  - Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]

- **Interaction**
  - With adversaries: CMLEES [Chakraborty & S., ICML 2010]
  - With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
  - With people: TAMER [Knox & S., AAMAS 2010]

- **Synthesis**

- **Mortality**
  - Leverage the Past: Transfer Learning [Taylor, S., & Liu, JMLR 2007]
  - Acknowledge a Finite Future: TEXPLORE [Hester & S., MLJ 2013]
Making Friends on the Fly: Advances in Ad Hoc Teamwork

Samuel Barrett, Katie Genter, and Peter Stone

AAAI, 2015; AAMAS, 2015
Ad Hoc Teamwork [Stone et al., AIJ 2013]

- Only in control of a single agent or subset of agents
- Unknown teammates
- Shared goals
- No pre-coordination
Ad Hoc Teamwork [Stone et al., AIJ 2013]

- Only in control of a single agent or subset of agents
- Unknown teammates
- Shared goals
- No pre-coordination

Examples in humans:
- Pick up soccer
- Accident response
PLASTIC: Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation

- Learned Knowledge of Teammates
- Hand Coded Prior Knowledge

Ad Hoc Agent

Inference of Teammate Type → Updates → Teammate Knowledge

Action Selection

State → Reward

Environment

Team

Agent

Joint Action

Ad Hoc Agent’s Action

Teammates’ Actions
Testbed Domains

- Agent replaces single teammate in otherwise coherent team
- Adapts based on knowledge learned from previous teammates
Practical RL

- Representation
- Interaction
- Synthesis
- Mortality
Practical RL

**Representation**
- Selecting the Algorithm: parameterized domains [K.&S., MLJ 2011]
- Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]

**Interaction**
- With adversaries: CMLES [Chakraborty & S., ICML 2010]
- With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
- With people: TAMER [Knox & S., AAMAS 2010]

**Synthesis**

**Mortality**
- Leverage the Past: Transfer Learning [Taylor, S., & Liu, JMLR 2007]
- Acknowledge a Finite Future: TEXPLORE [Hester & S., MLJ 2013]
Practical RL

- **Representation**
  - Selecting the Algorithm: parameterized domains [K.&S., MLJ 2011]
  - Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]

- **Interaction**
  - With adversaries: CMLÉS [Chakraborty & S., ICML 2010]
  - With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
  - With people: TAMER [Knox & S., AAMAS 2010]

- **Synthesis**

- **Mortality**
  - Leverage the Past: Transfer Learning [Taylor, S., & Liu, JMLR 2007]
  - Acknowledge a Finite Future: TEXPLORE [Hester & S., MLJ 2013]