Source Task Creation for Curriculum Learning

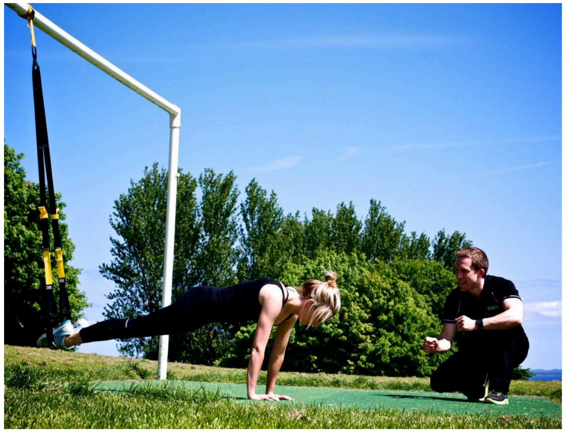
Sanmit Narvekar, Jivko Sinapov, Matteo Leonetti, and Peter Stone
Department of Computer Science
University of Texas at Austin
{sanmit, jsinapov, matteo, pstone} @cs.utexas.edu

Learning Agents Research Group
The University of Texas at Austin

Introduction

- Curricula widespread in human learning
  - Education, sports, games…

- Why curricula?
  - Target task too hard to make progress
  - Faster to learn and combine skills from easier tasks



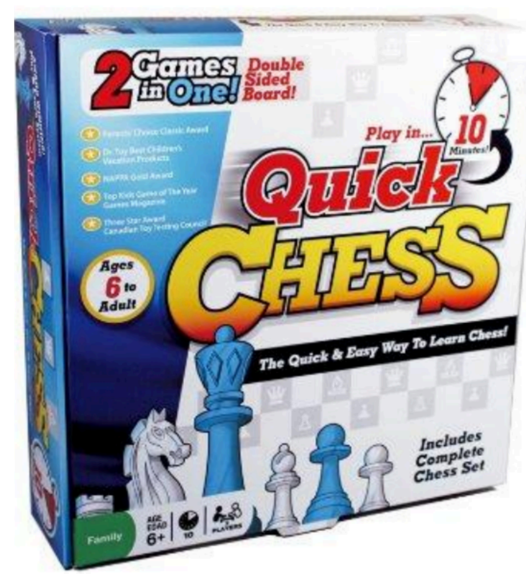
A good curriculum:

- Breaks down the task
- Lets the agent learn on its own
- Adjusts to the progress of the agent

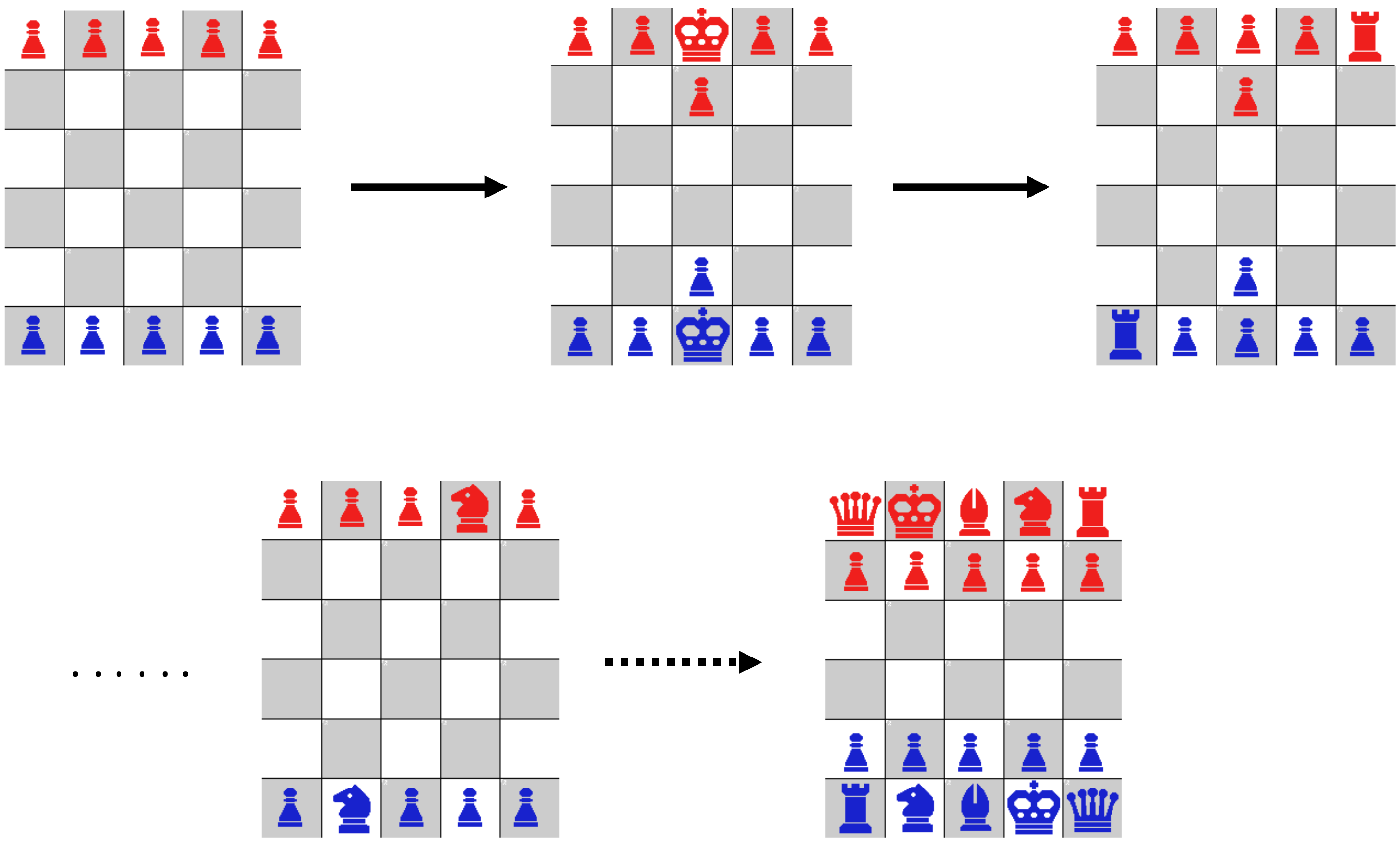
Example: Quick Chess

- Quickly learn the fundamentals of chess

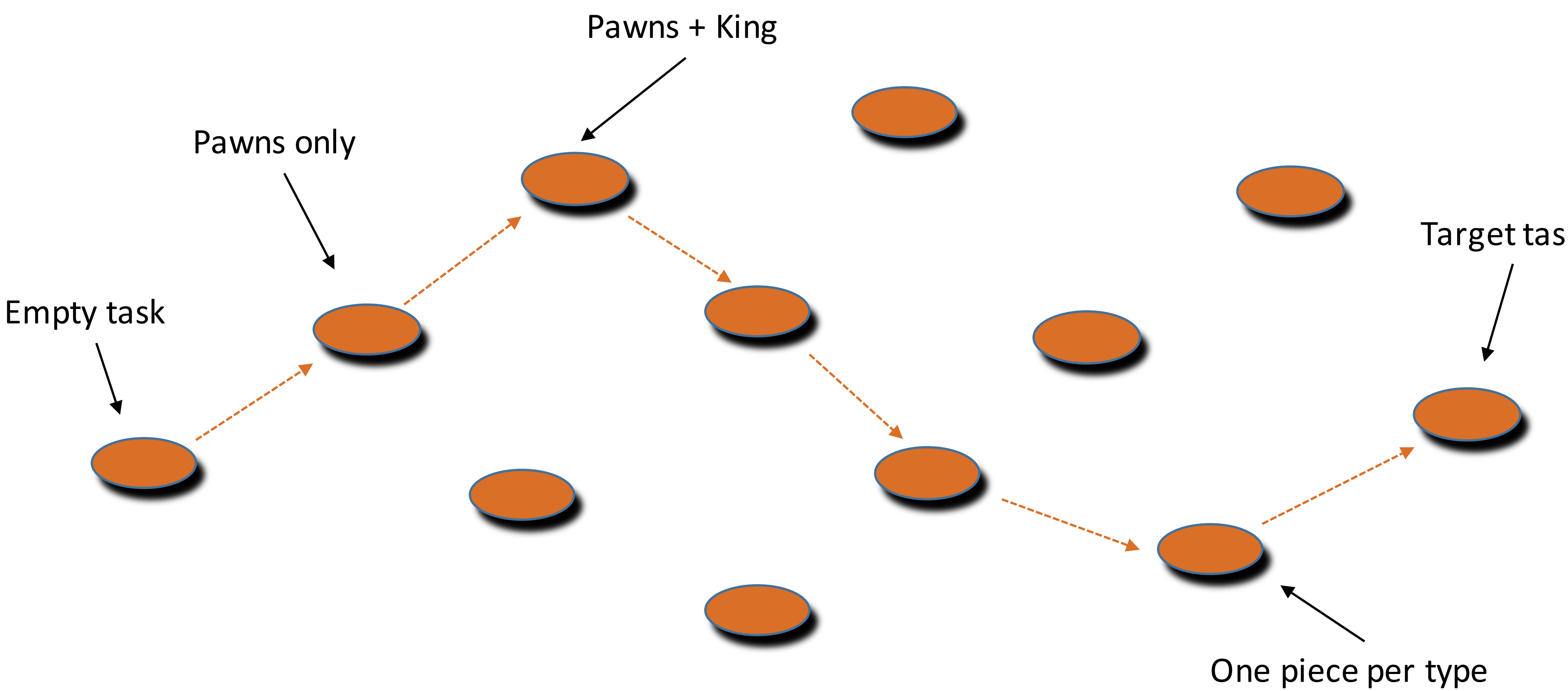
- 5 x 6 board
- Fewer pieces per type
- No castling
- No en-passant



Example: Quick Chess

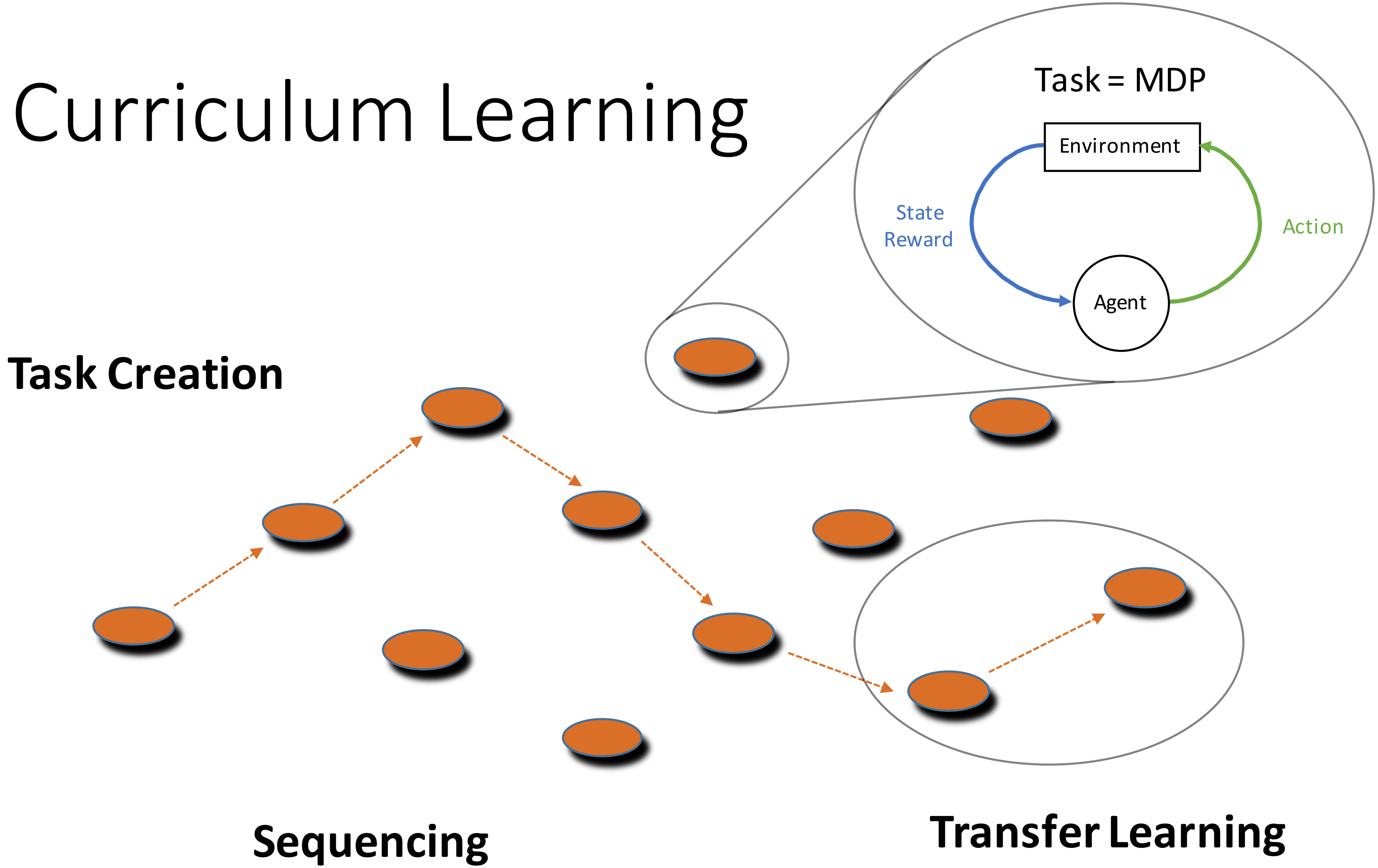


Task Space



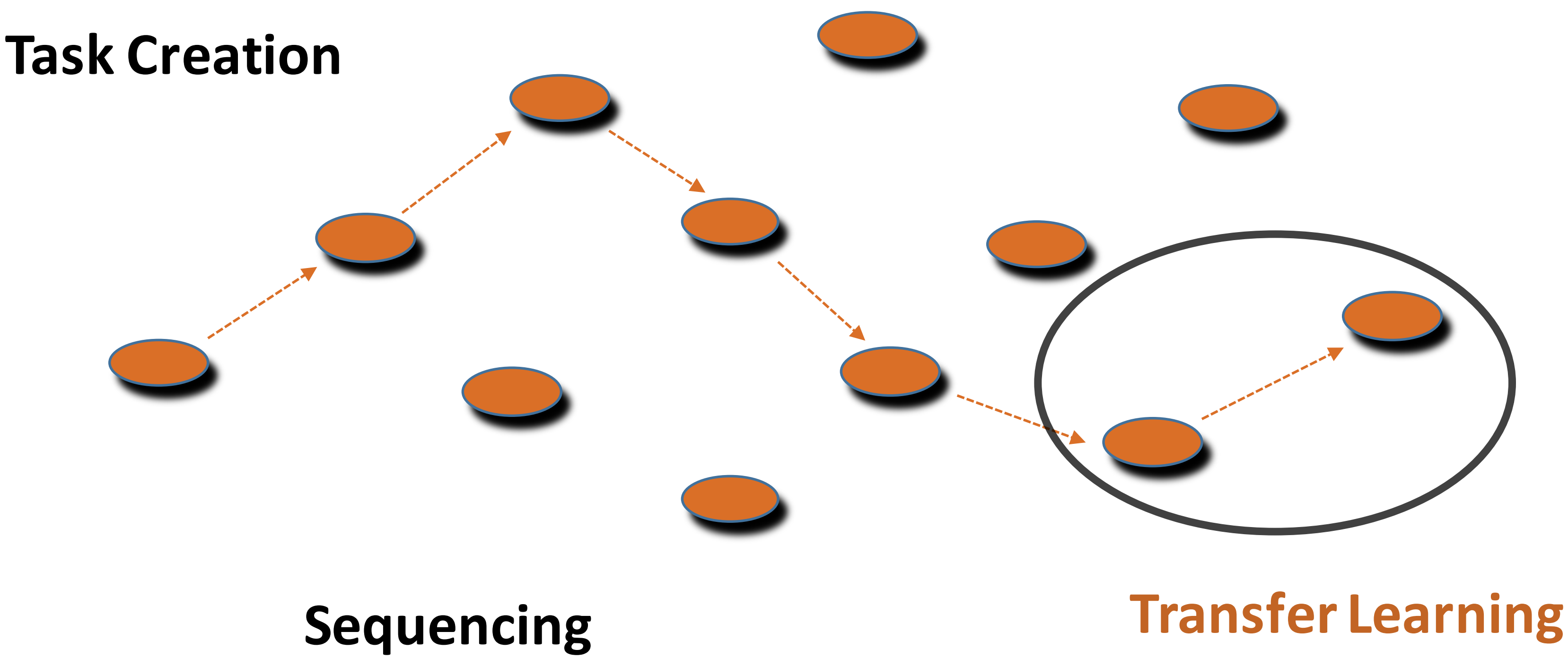
- Quick Chess is a curriculum designed for people
- We want to do something similar for autonomous agents

Curriculum Learning



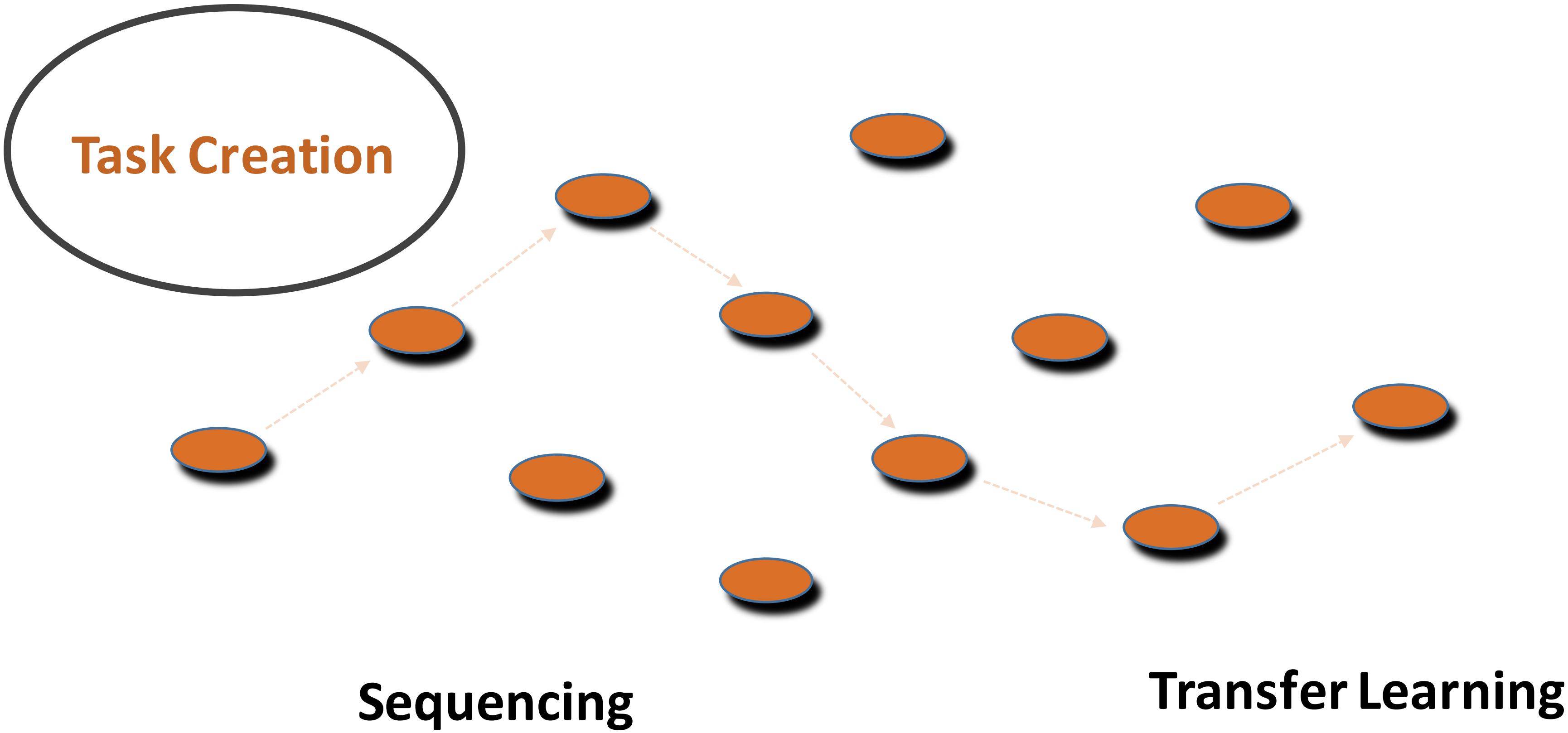
- Curriculum learning is a complex problem that ties task creation, sequencing, and transfer learning

Transfer Learning



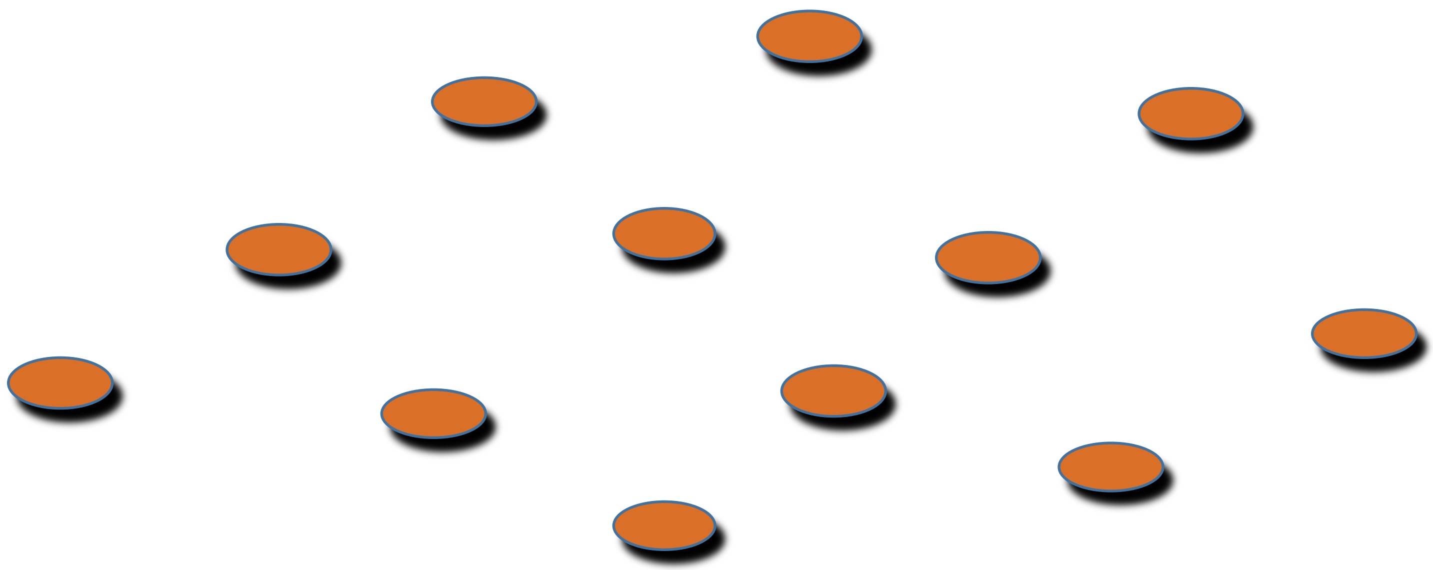
- Well studied problem [Taylor 2009, Lazaric 2011]
- Given a source and target task, how to transfer knowledge
  - We transfer value functions

Task Creation



- This talk will focus on task creation
- Automatic sequencing is an important direction for future work
- Show we can create a useful space of tasks to compose a curriculum

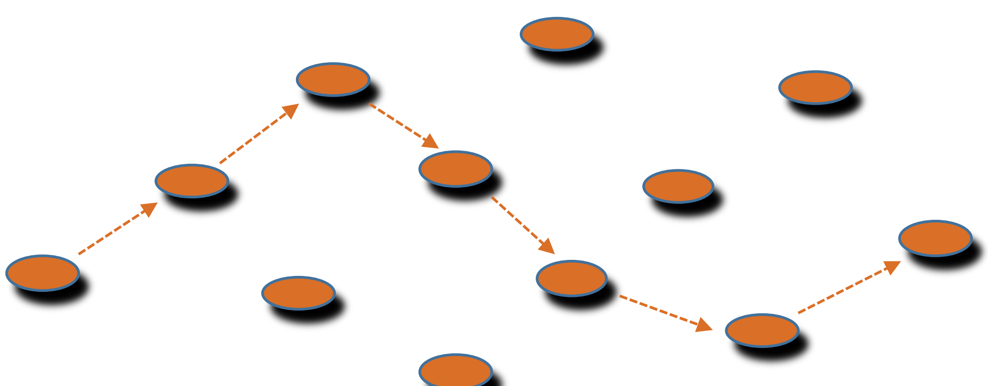
Task Creation



Formalism for Task Creation

- Key Idea: create tasks using both domain knowledge and by observing the agent's performance on a task
- We propose a formalism for task creation
- Consists of a set of heuristic functions $f : M_t \times X \mapsto M_s$

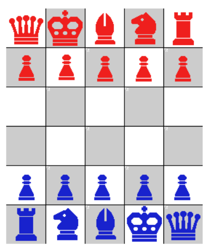
  that create a source task M_s given a target task M_t and (s,a,s',r) trajectory tuples X from M_t
- Formalism is domain-independent (applicable to many domains)

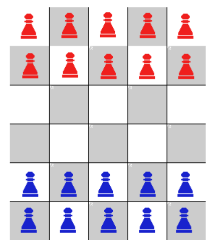


Formalism for Task Creation

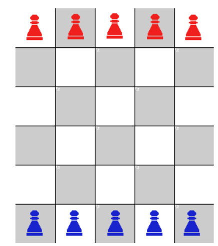
- Each function alters different parts of the MDP *M* to create source tasks

State/Action Space





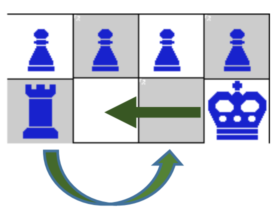
$$M = (S, A, P, R, S_0, S_f)$$

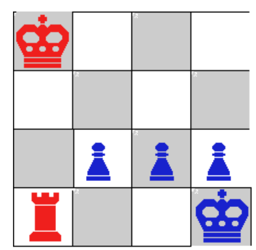


Rewards

Reward for promotion

Transitions





Initial/Terminal State Distributions

Heuristic Functions

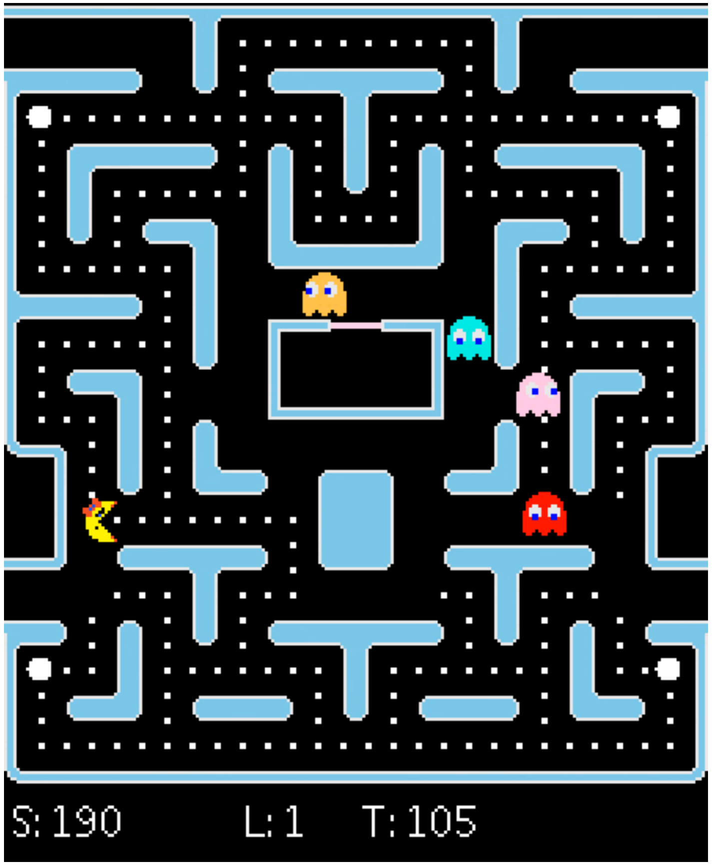
1. Task Simplification
2. Promising Initializations
3. Mistake Learning
4. Action Simplification
5. Option-based Subgoals
6. Task-based Subgoals
7. Composite Subtasks

Uses knowledge of domain

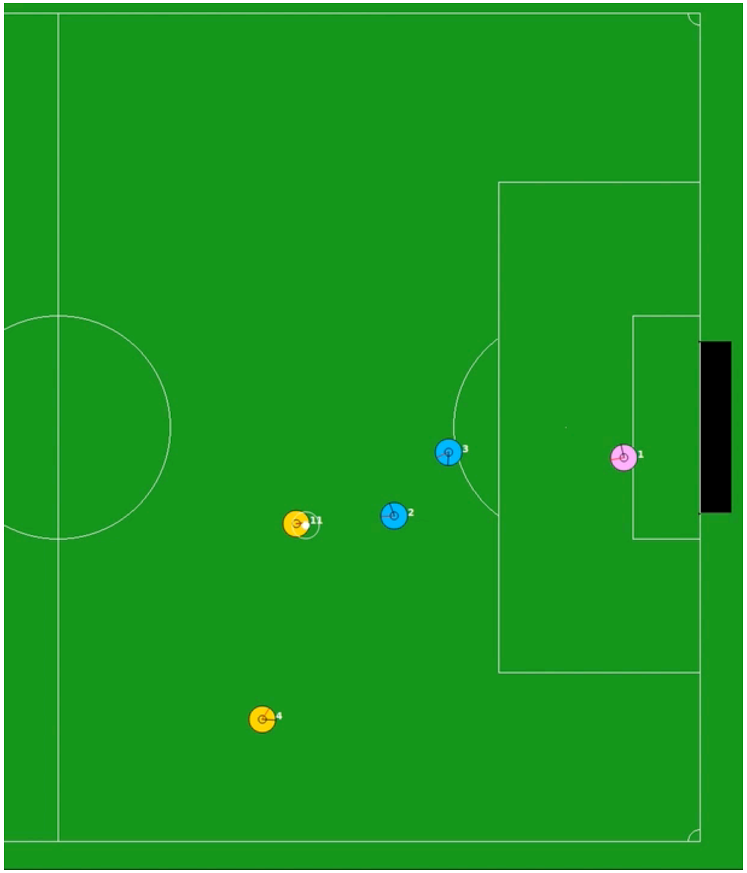
Observes the agent

Experimental Domains

Ms. Pac-Man

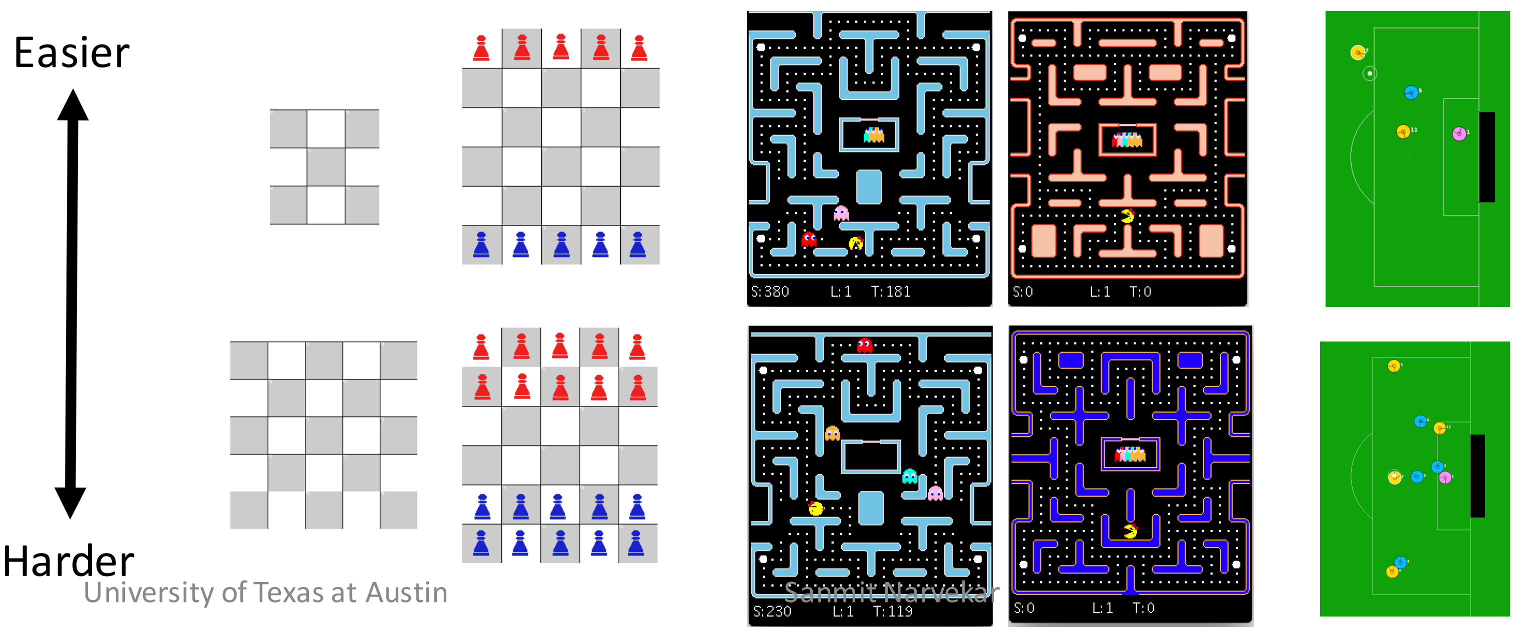


Half Field Offense



Task Simplification

- Use knowledge of the domain encoded in degrees of freedom F to simplify the task
  - F = [F_1, F_2, … F_n] vector of features that parameterize the domain
- Assumes ordering over each F_i corresponding to task complexity
- Reduces the complexity of one degree of freedom at a time



Promising Initializations

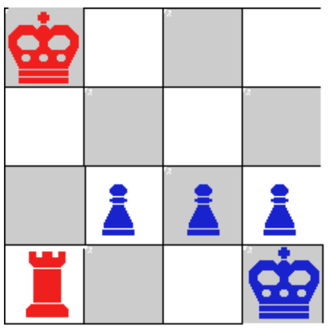
- Positive outcomes can be rare at onset of learning
- Explores regions of state space near positive outcomes/rewards

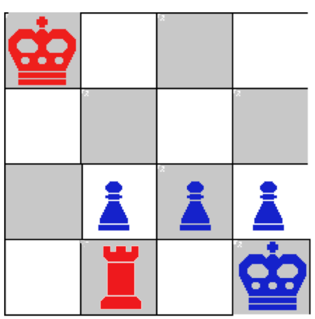
$$\textsc{PromisingInitializations}(M, X, C, \delta, \rho)$$

- $C(s_1, s_2)$: distance measure quantifying state proximity
- δ : threshold on distance
- ρ : percentile threshold on which states/rewards in X are positive outcomes
- Returns MDP that initializes start state distribution to these states

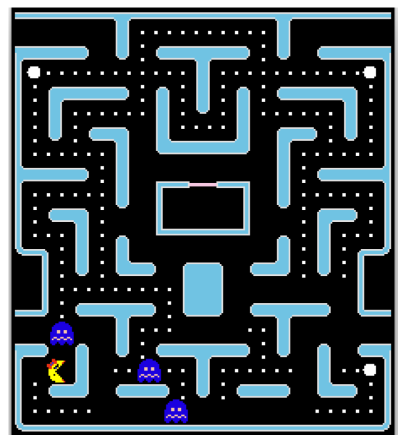
Promising Initializations

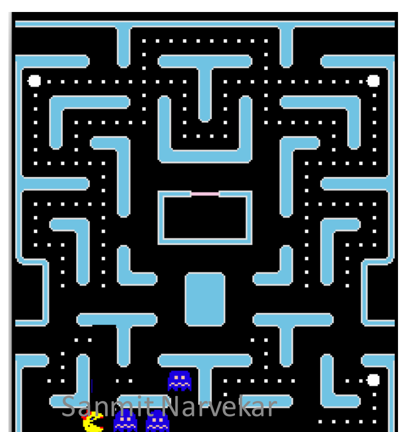
Number of “moves” away



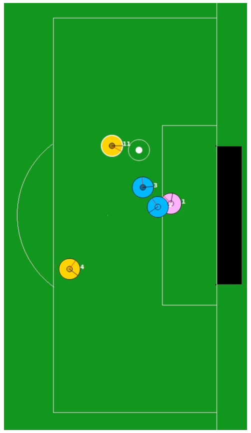


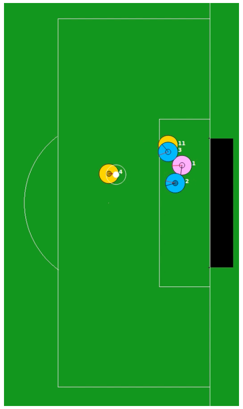
Number of steps away





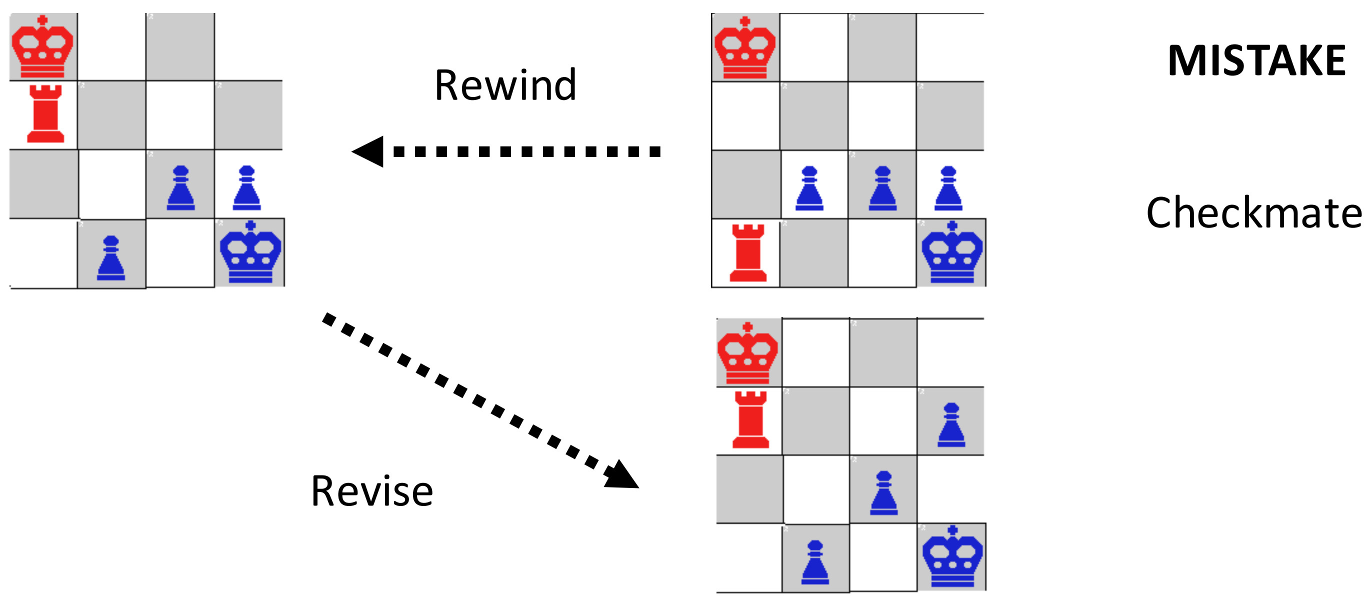
Euclidean Distance





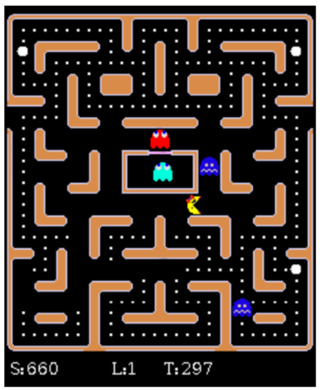
Mistake Learning

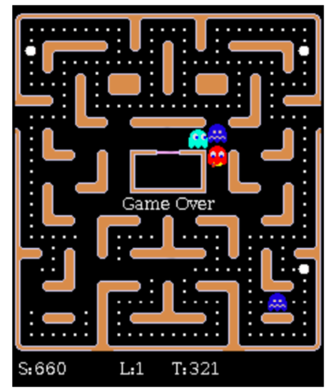
- Create subtasks to avoid or correct mistakes
  - Specified by the domain
  - Eg. Termination in non-goal state
- Rewind the episode epsilon steps back, and learn a revised policy from there



Mistake Learning

MISTAKES

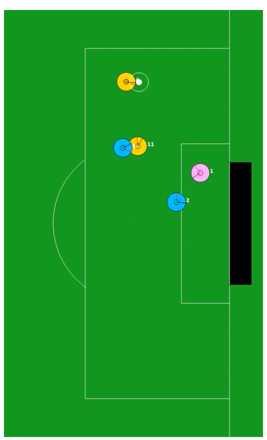


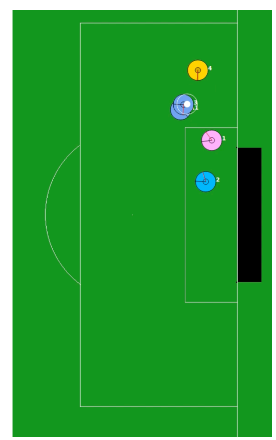


Getting eaten by ghost

Not eating edible ghost

How far back to rewind?



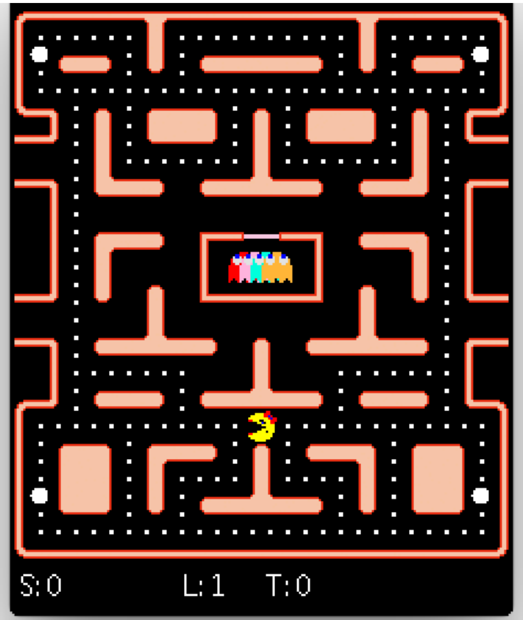


Failing to score

Losing possession

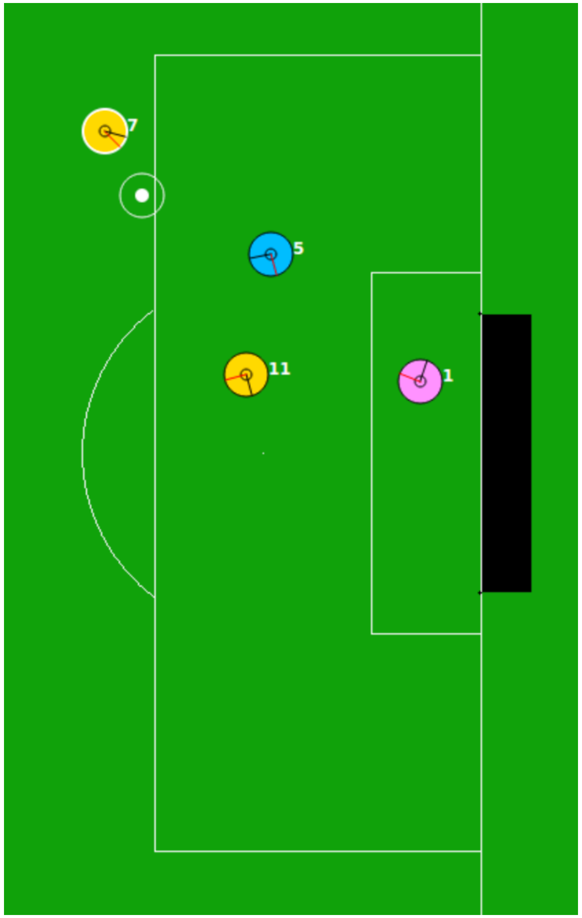
Results

Ms. Pac-Man

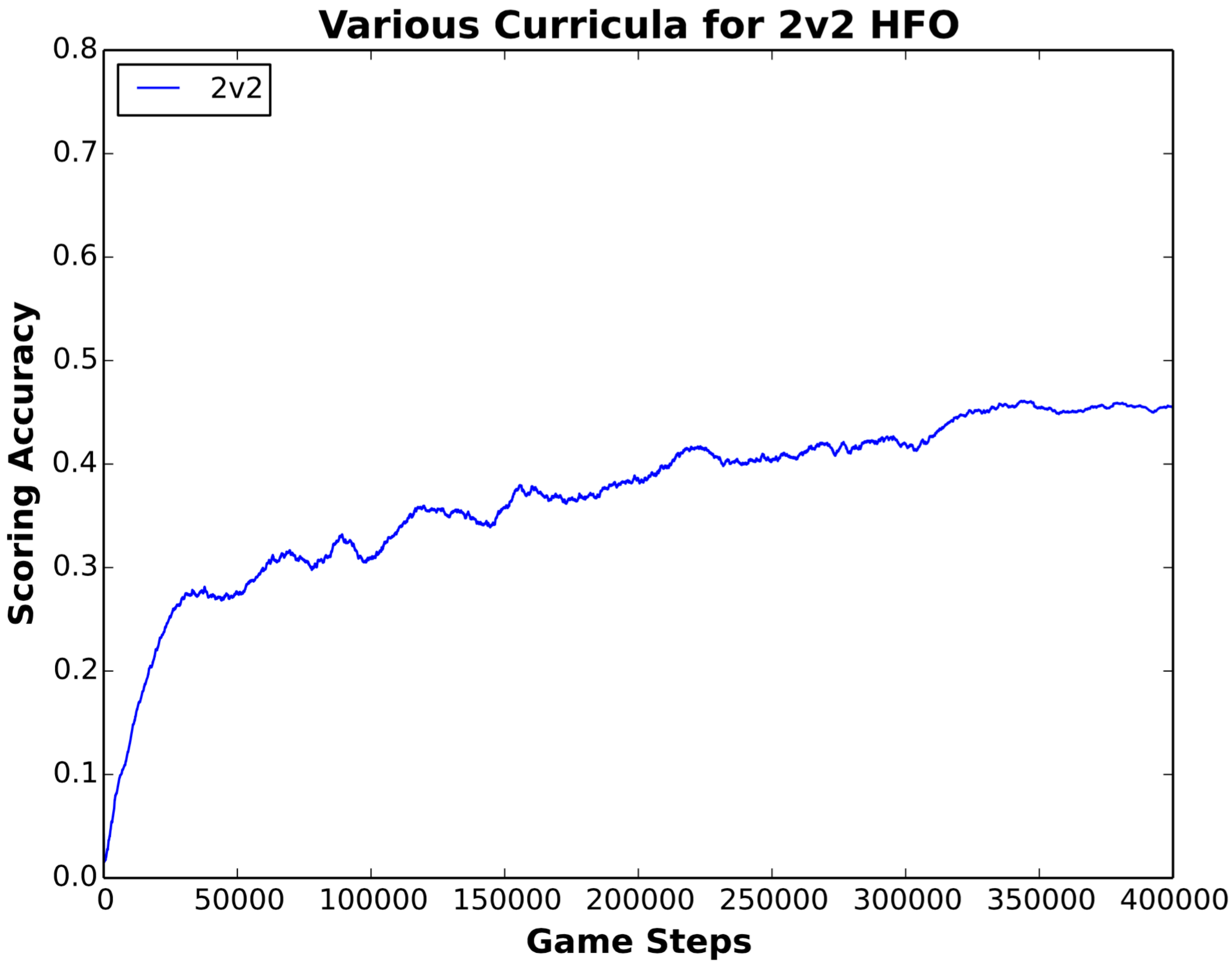


(results in paper)

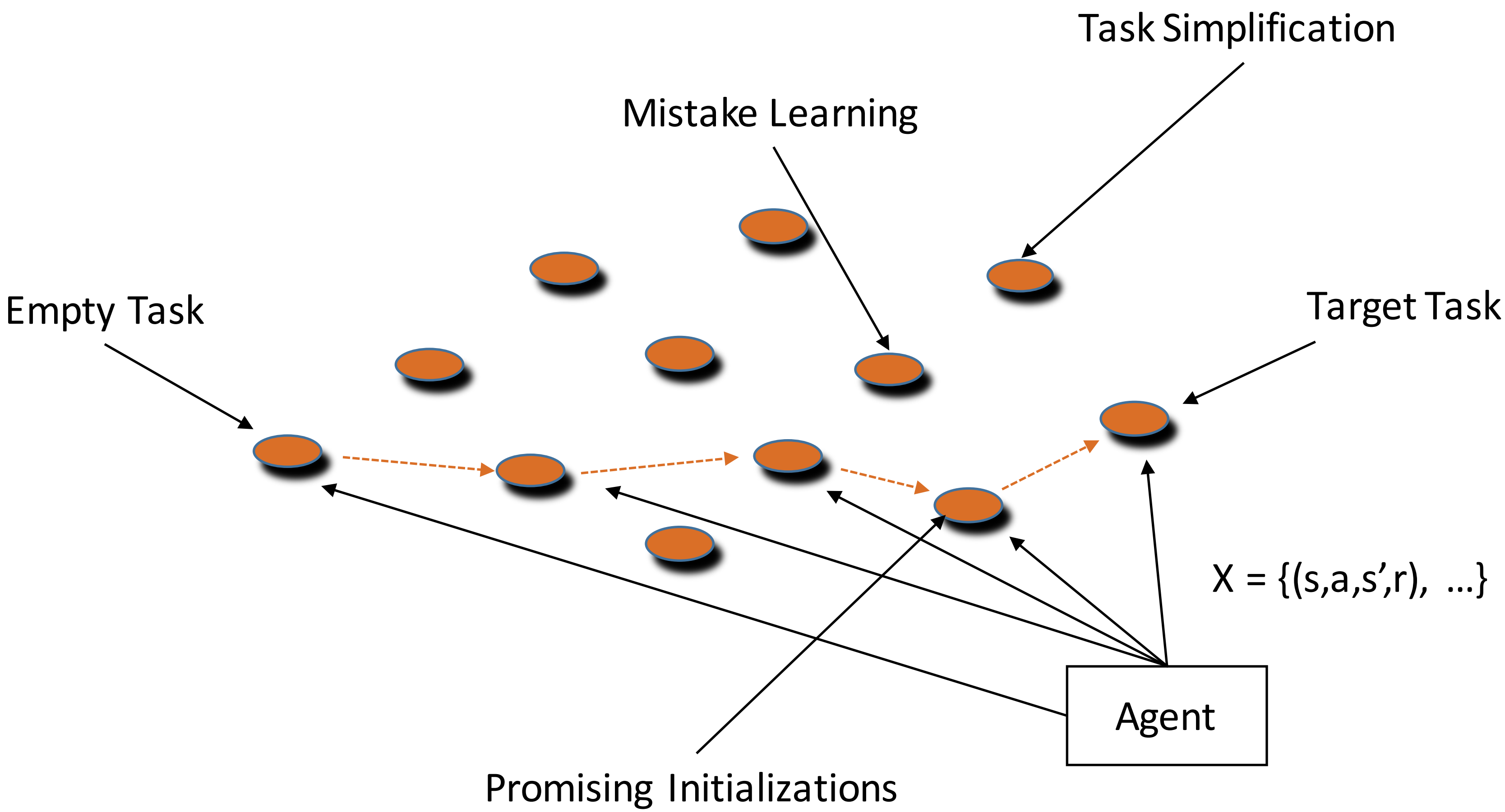
2v2 Half Field Offense



2v2 HFO Baseline



Curriculum Generation

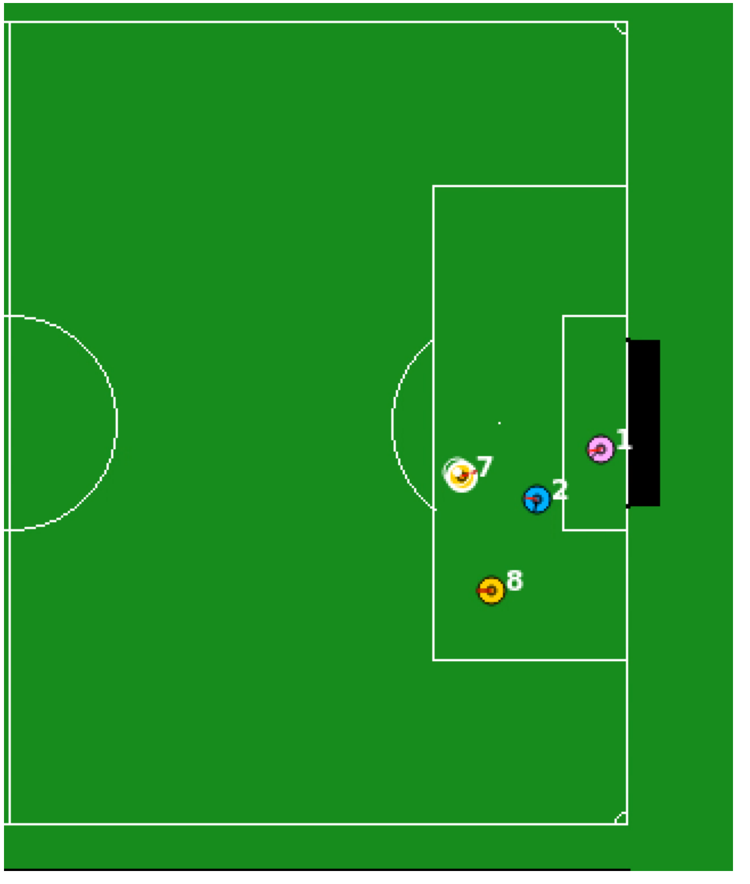


Shoot Task

- Initially, goal scoring episodes are rare
- We observe a few successful goals
- Use PromisingInitializations to target exploration in this region

$$M_{shoot} = \textsc{PromisingInitializations}(M_{2v2}, X_{2v2}, C, \delta, \rho)$$

- Agents learn to shoot on goal

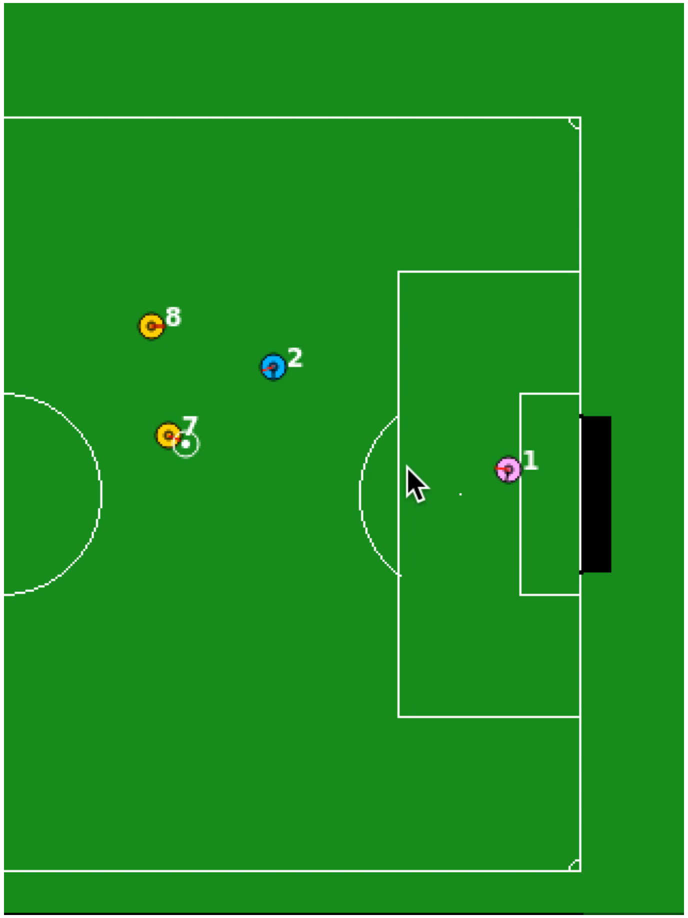


Dribble Task

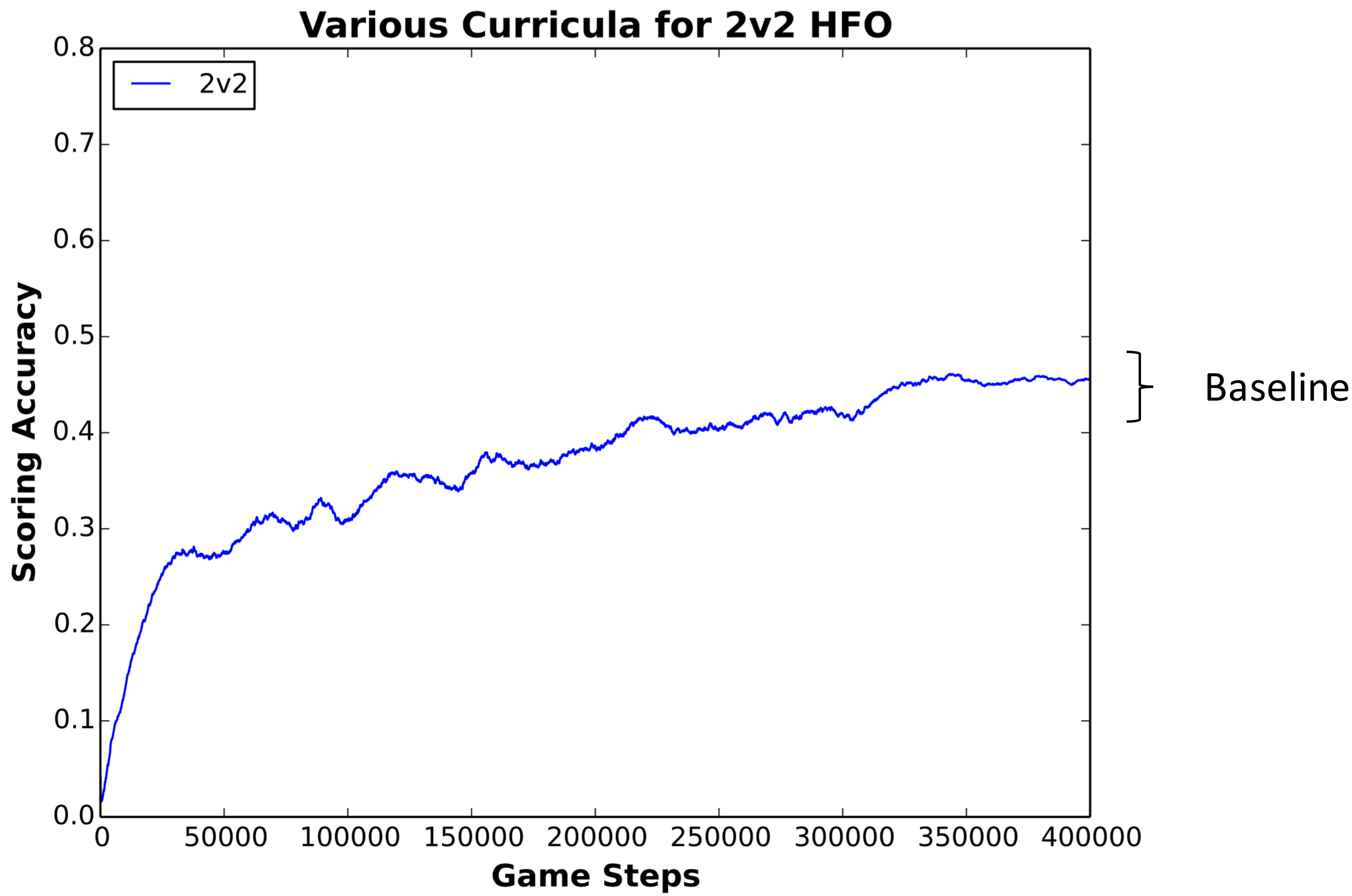
- Agent takes too many shots from far away

- Skill needed: move the ball up the field while maintaining possession, until a shot is likely to score

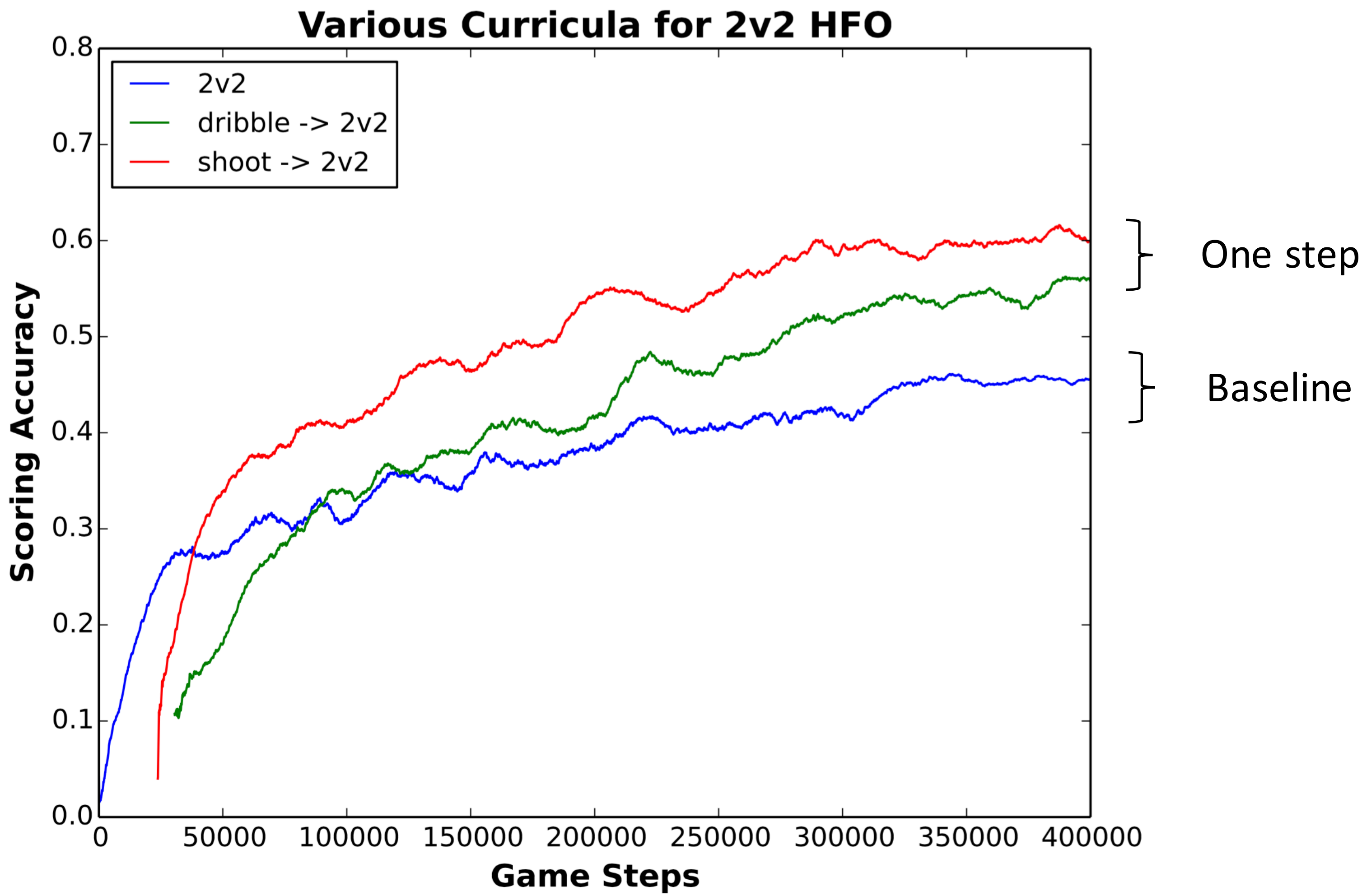
$$
\begin{aligned}
M_1 &= \textsc{LinkSubTask}(M_{2v2}, M_{shoot}, V_{shoot}) \\
M_{dribble} &= \textsc{ActionSimplification}(M_1, X_{2v2}, \alpha)
\end{aligned}
$$



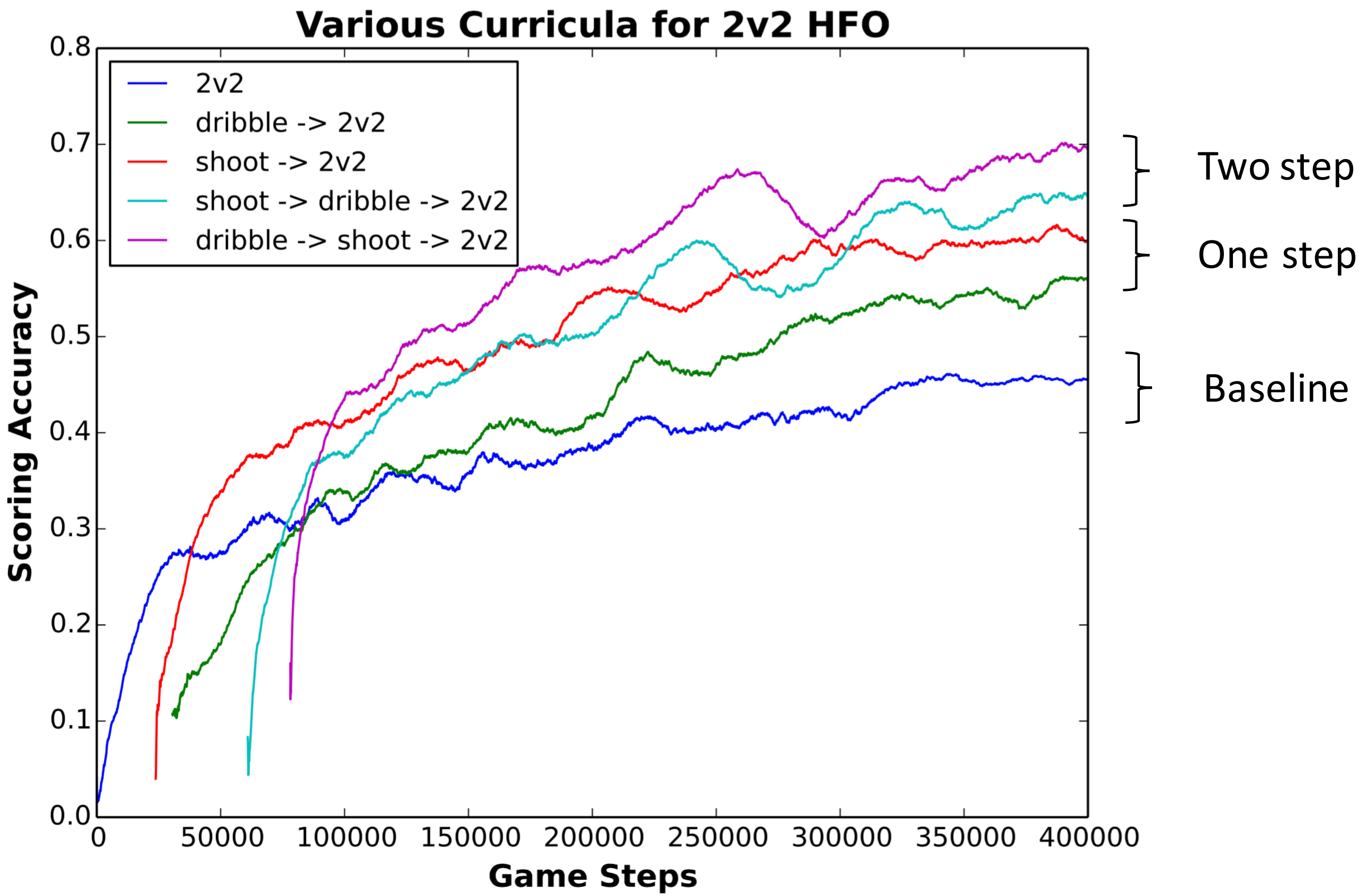
2v2 HFO Results



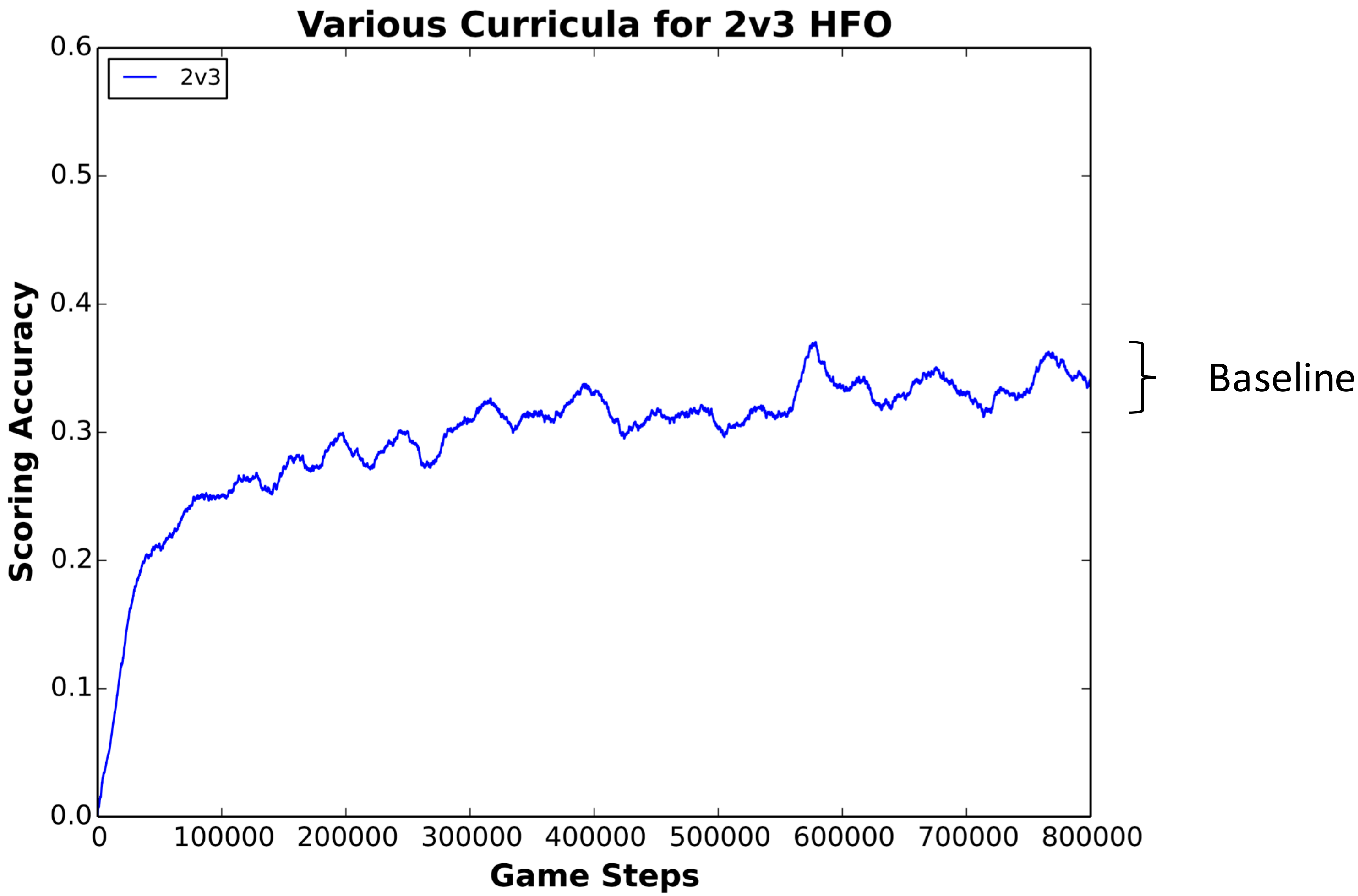
2v2 HFO Results



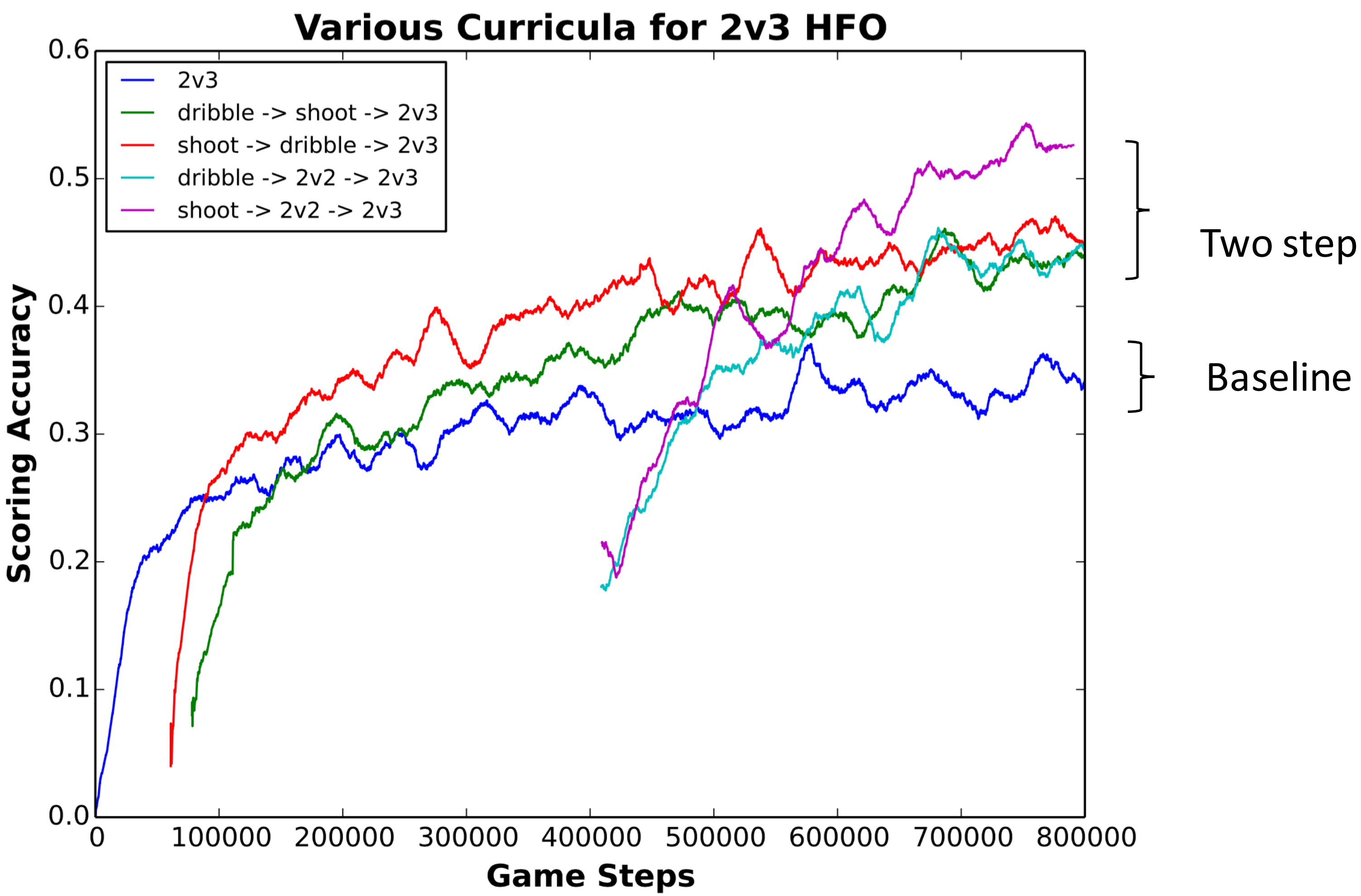
2v2 HFO Results



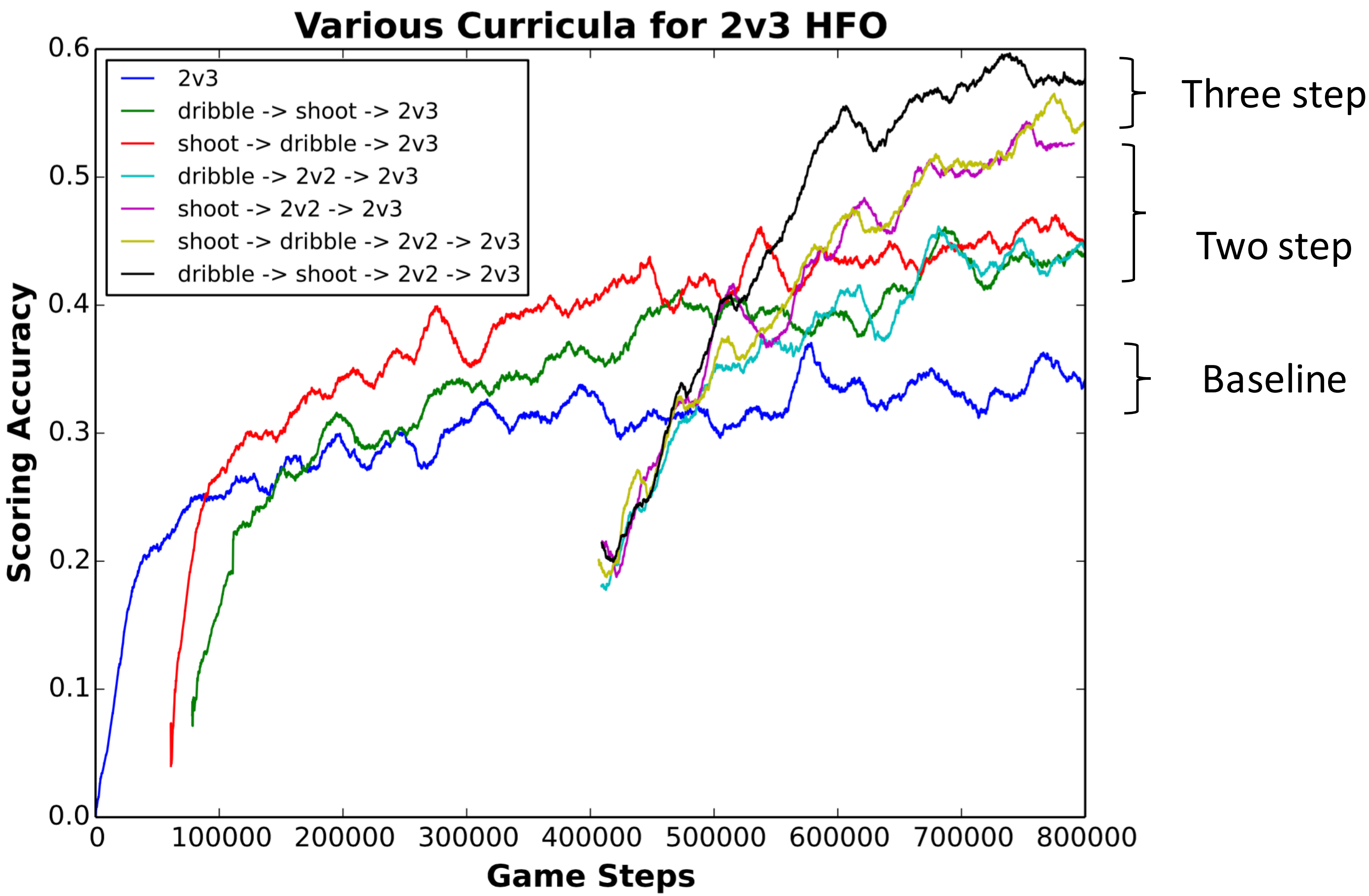
2v3 HFO Results



2v3 HFO Results



2v3 HFO Results



Experimental Recap

- Tasks created by our formalism can be used as source tasks in a curriculum
- Learning via a curriculum can improve learning speed or performance

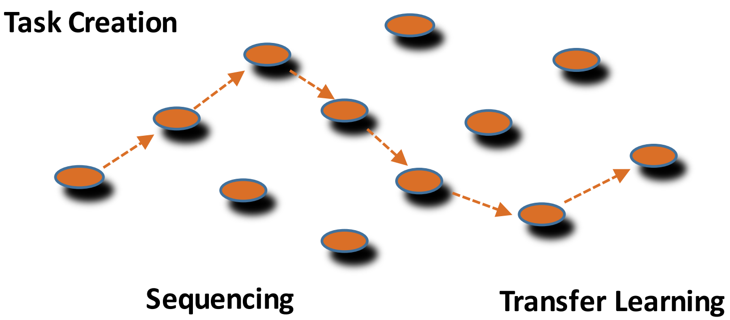
Related Work

- Curriculum learning in supervised learning [Bengio et al. 2009]
- Multi-task reinforcement learning [Wilson et al. 2007]
- Lifelong reinforcement learning [Ammar et al. 2014]
- Learning task transferability [Sinapov et al. 2015]

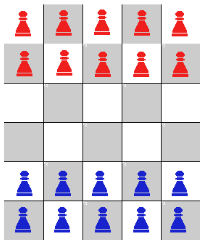
Key Differences

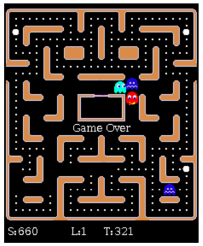
- Source tasks created solely to improve performance on target
- Focus on task generation, not selection
- Agent-tailored source tasks based on agent performance

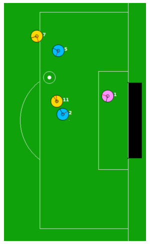
Summary



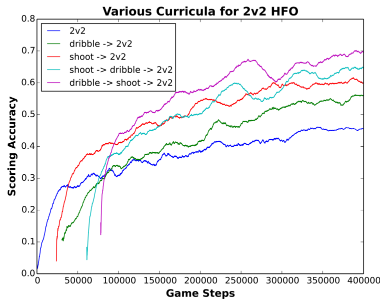
- Presented curriculum learning in the context of reinforcement learning

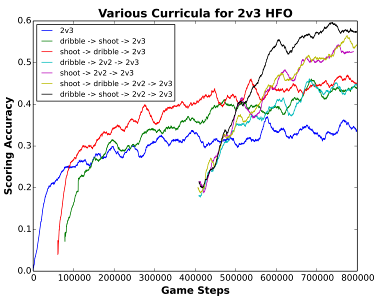






- Defined a domain-independent formalism to create source tasks, tailored to the performance of the agent





- Empirically demonstrated using a curriculum can improve learning speed or performance