CSC2621 Topics in Robotics
Reinforcement Learning in Robotics

Week 1: Introduction & Logistics

Animesh Garg

Agenda

- Logistics
- Course Motivation
- Primer in RL
- Human learning and RL (sample paper presentation)
- Presentation Sign-ups

Course Logistics

- Professor Animesh Garg
- TA1: Dylan Turpin | TA2: TBD
- Contact us at through Quercus or email: garg@cs.toronto.edu
- For room information, office hours, etc, see website: https://pairlab.github.io/csc2621-w20/#

  Note: The logistics info on these slides is subject to change. The website will always contain the most up-to-date information, so please refer to it for all course logistics.

Learning Objectives

- Acquire familiarity with state of the art in RL
- Articulate limitations of current work, identify open frontiers, and scope research projects.
- Constructively critique research papers, and deliver a tutorial style presentation.
- Work on a research-based project, implement & evaluate experimental results, and discuss future work in a project paper.

Class Format

- In-Class Paper Presentation: 25%
- Take-Home Midterm: 15%
- Pop-quizzes & Class Participation: 10%
- Project: 50%

Class Format

- No standard lectures
- Discussion/Tutorial -based
- Students will present on readings

- 1 broad topic per class
- 1-2 overview reading on topic – Topic Tutorial
- 2-3 state-of-the-art paper on topic – Latest Results in Sub-Topic

Everyone is expected to have read the state-of-the-art reading before class. Encouraged but not required to read overview.

Class Format: Presentations

4 presentations per class in teams of 2 students per paper

Each student should expect to give a presentation in class.

Those presenting a reading are also the key "go-to" people for questions on that reading (on Quercus etc).

Survey presentation (40 minutes) on an important topic in RL

State of the art article presentation (30 minutes) related to the survey

Required to provide two exercise questions for the reading presented

Class Format: Presentations

The success of CSC 2621 depends on high quality presentations

To help facilitate this we

will provide presentation templates
will provide feedback the week before to go through your presentation
part of grade is based on your presentation at this point

Note: this effectively means slides are due a week in advance

Final presentation format dependent on class size (stay tuned)

Class Format: Presentations

The success of CSC 2621 depends on high quality presentations

To help facilitate this we

will provide presentation templates
will provide feedback the week before to go through your presentation
part of grade is based on your presentation at this point

Note: this effectively means slides are due a week in advance

Final presentation format dependent on class size (stay tuned)

Class Format: Presentations

We also ask presenters for 2 exercise questions related to the reading

- Used for helping students
- Practice and assess if understood some of key ideas in the reading
- Used to study for midterm

Questions should involve about 1-5 minutes of thought

Check with the TA about these questions (bring them to meeting)

Check if they are at the correct level or need further modification.

Should adhere to provided template

Class Format: Midterm

- 1 Take Home Midterm

24-hours to complete, should not take more than 90 mins in a single session. Allowed to consult books, notes, slides, but no discussion with anyone in the class or outside the class about the exam

If you have a clarification question, please contact the course staff through piazza with a private message.

To do well on the exam, you should attend class, read the paper readings, and complete and understand the practice exercises.

Class Format: Participation

- Participation in class is expected
  through preparation before coming to class,
  proactive discussion and questions
  peer-review of projects and paper presentations.

- Expect to have 2-4 pop quizzes through the term based on material covered in class up to that point including the expected reading of the day.

Class Format: 1 Course Project

Teams of 1-3 (ideally 3), but exceptions on a case-by-case basis.

The goal of the project is to instigate or continue to pursue a novel research effort in reinforcement learning.

The project provides an opportunity to

- synthesize related work,
- identify open gaps in the literature,
- define a feasible and new direction,
- make progress on this direction, and
- present your progress in a presentation and in a paper.

Agenda

- Logistics
- Course Motivation
- Primer in RL
- Human learning and RL (sample paper presentation)
- Presentation Sign-ups

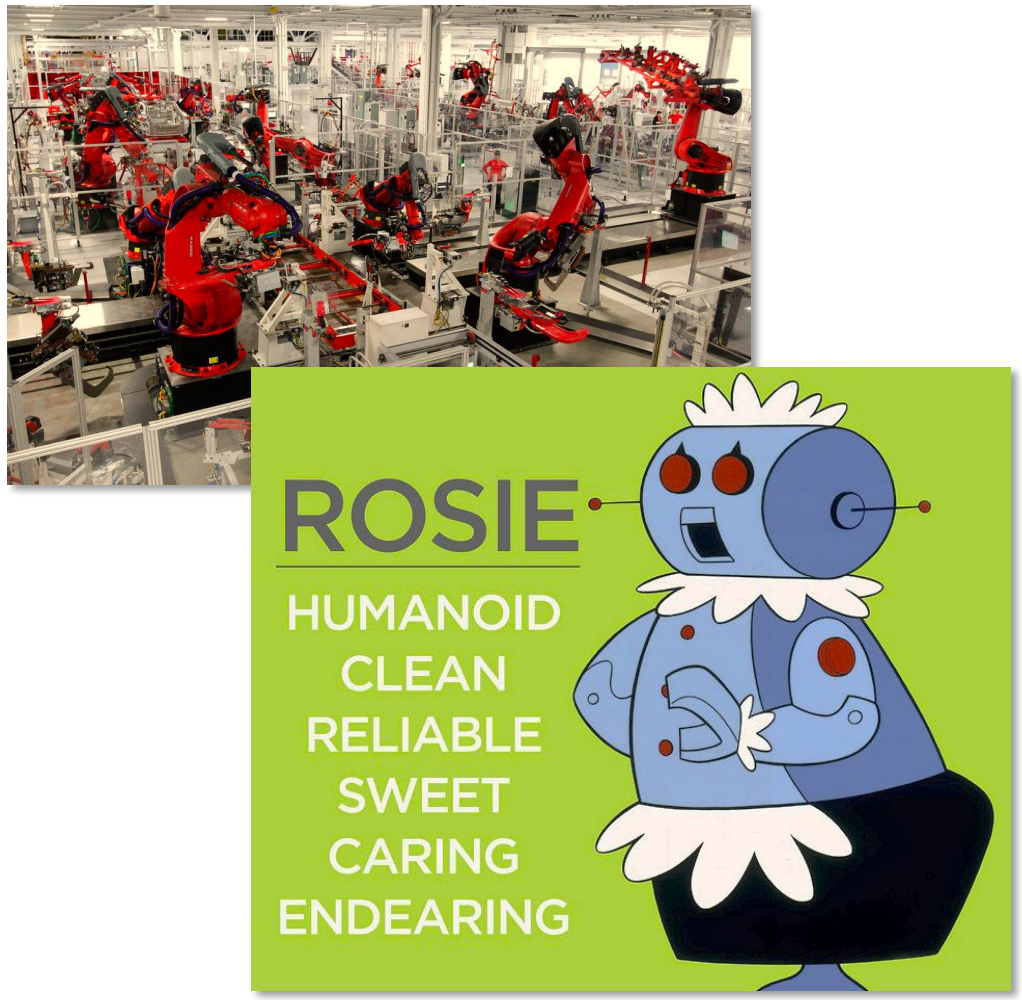
Learning Behaviors

What can we do now?

Sometimes automate some bounded tasks in static environments with pre-programmed behavior

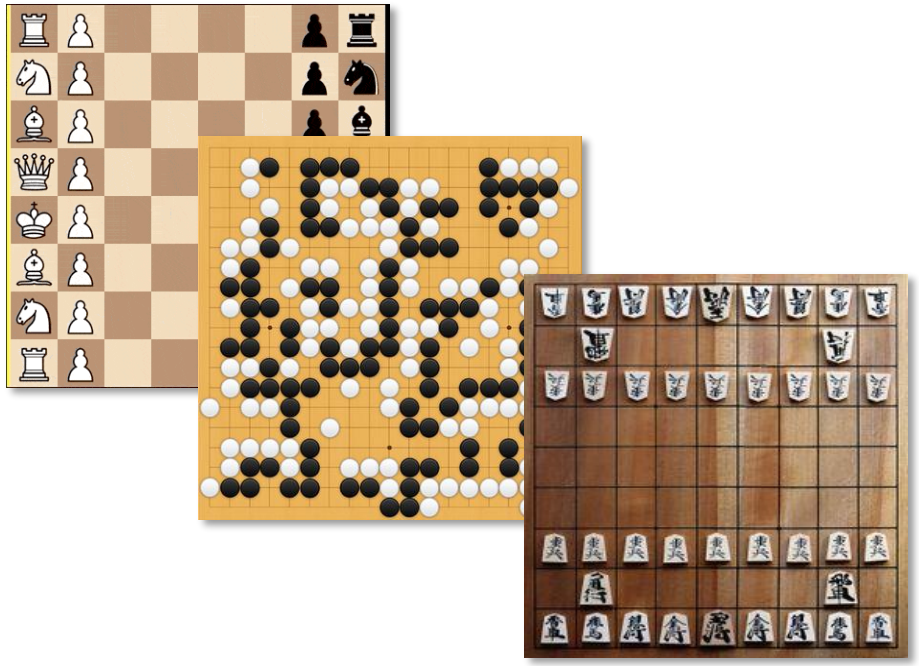
What do we want?

Autonomous agents in Physical world that interact to accomplish broad set of goals in Dynamic Environments.



Decision Making & Motor Control

Hard things are easy, it is the easy things that are ridiculously hard!
-- Moravec's Paradox



https://www.brainfacts.org/brain-anatomy-and-function/evolution/2015/daniel-wolpert-the-real-reason-for-brains

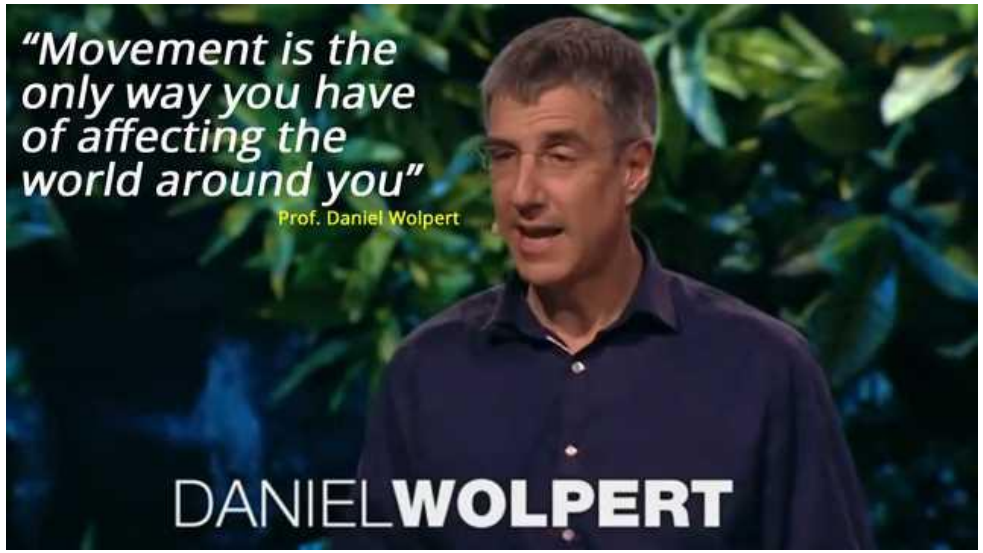
Decision Making & Motor Control

"The brain evolved, not to think or feel, but to control movement."

--Daniel Wolpert, Neuroscientist

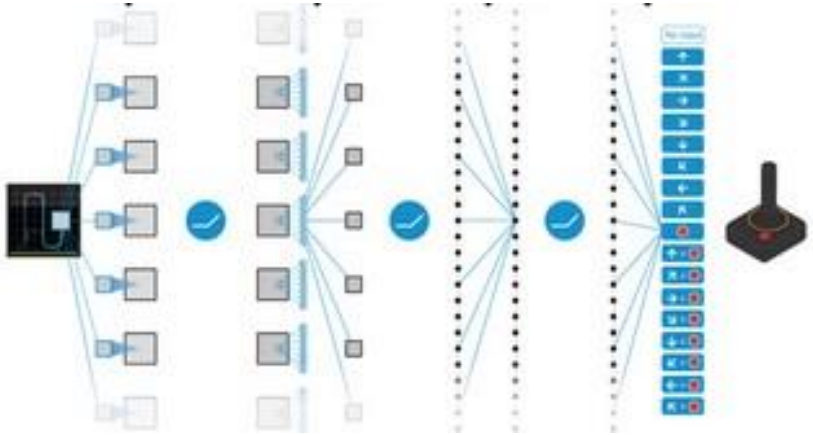
Sea Squirts

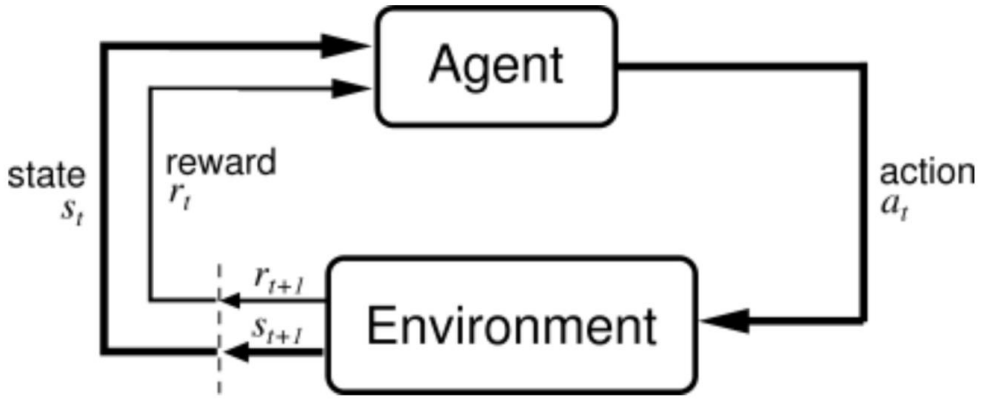
Digests brain after need for movement in life is complete

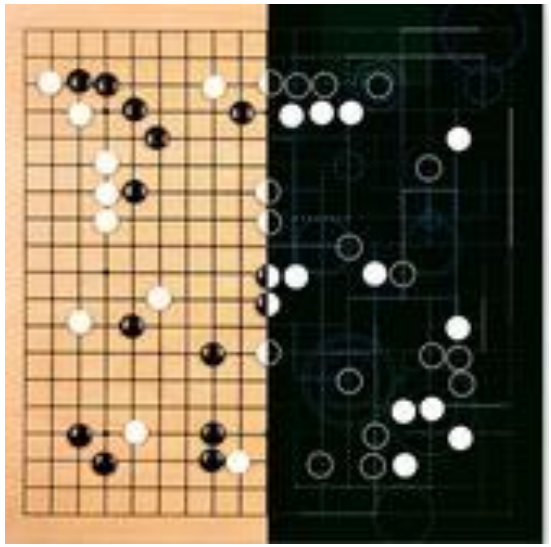


https://www.brainfacts.org/brain-anatomy-and-function/evolution/2015/daniel-wolpert-the-real-reason-for-brains

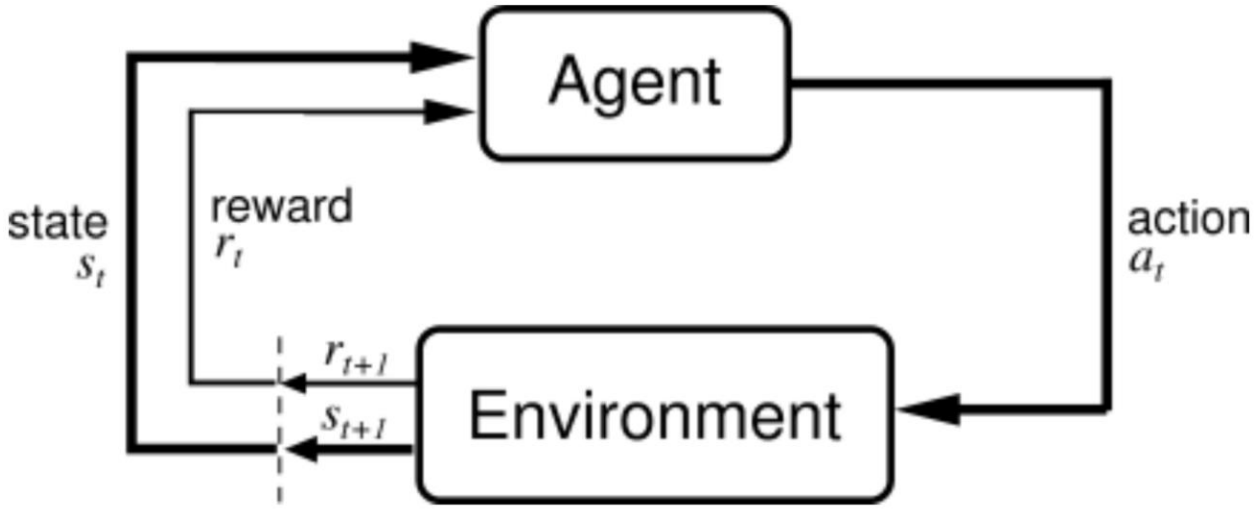
Reinforcement Learning





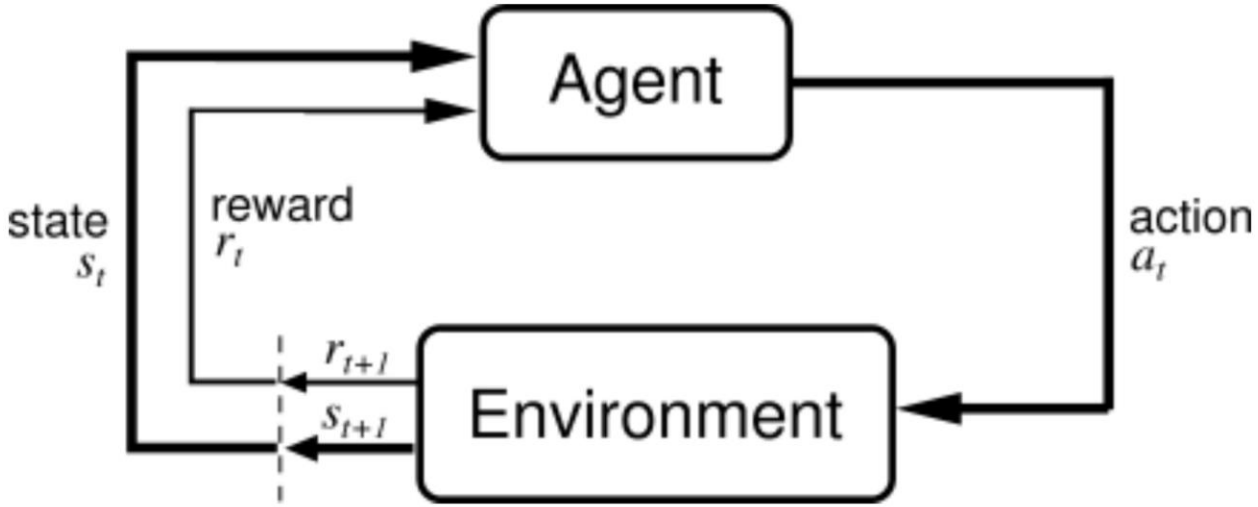


Reinforcement Learning



Provides a general-purpose framework to explain intelligent behavior in in simpler lifeforms and sometime humans, as well as a computational framework to solve problems of interest in Decision Making in AI.

Markov Decision Processes



$$\mathcal{M} = \langle S, A, P(\cdot,\cdot), R(\cdot,\cdot), T \rangle$$

State Space

Action Space

Transition Function

$Prob: S \times A \to S$

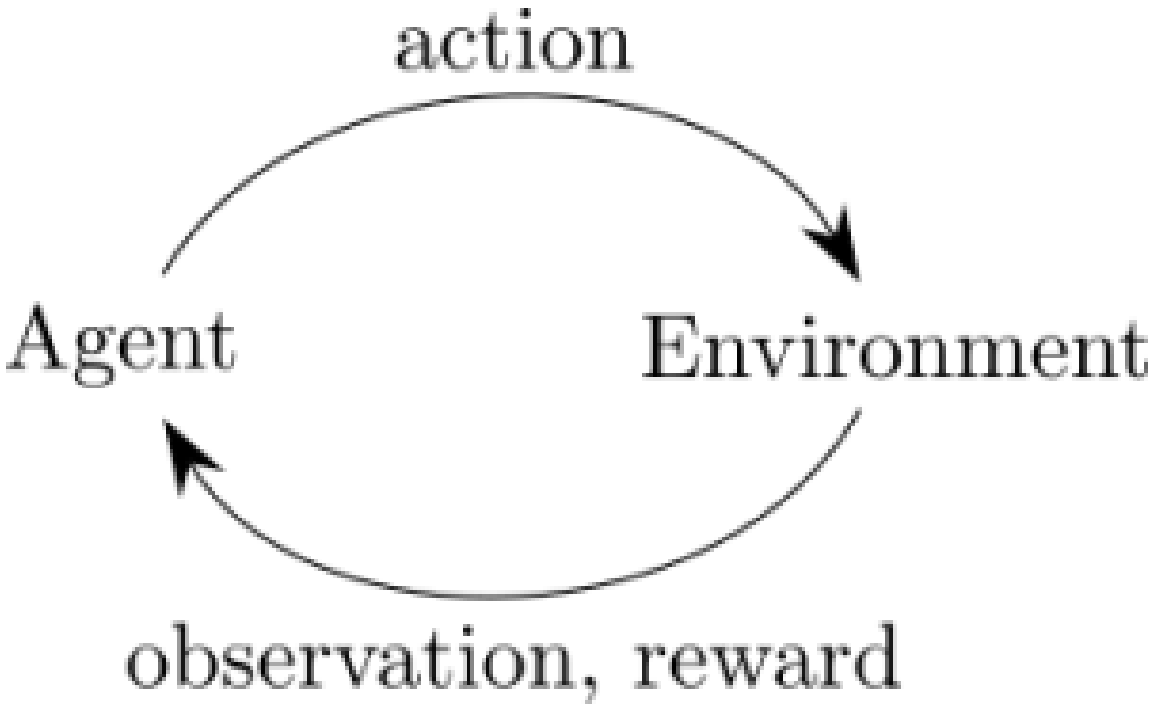
Reward Function

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

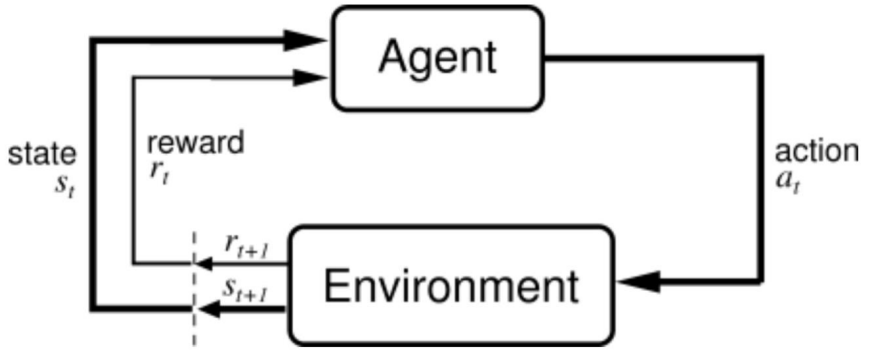
Time Horizon

What is RL: Reinforcement Learning

- At each step t the agent:
  - Executes actions A_t
  - Receive Obs O_t
  - Receive Reward R_t
- Environment
  - Receives actions A_t
  - Emits Obs O_{t+1}
  - Emits Scalar Reward R_{t+1}
- Time increments at Env. Update



Reinforcement Learning: MDP



$$\mathcal{M} = \langle S, A, P(\cdot,\cdot), R(\cdot,\cdot), T \rangle$$

State Space | Action Space | Transition Function | Reward Function | Time Horizon

$Prob: S \times A \to S$ $\quad$ $R: S \times A \to \mathbb{R}$

Goal: Find Optimal Policy: $\pi^*: S \to A$

Markov Decision Processes

- MDP: $\mathcal{M} = \langle S, A, P(\cdot,\cdot), R(\cdot,\cdot), T \rangle$
- Goal: Maximize Total Discounted Reward with discount factor γ
- Optimal Policy: π^*

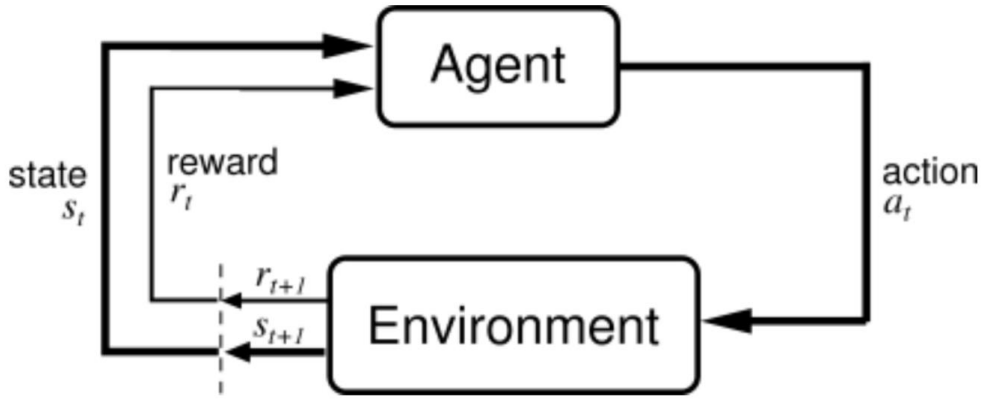
$$\pi^* = \underset{\pi}{arg\max}\, E_{s\sim\pi, p_0}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t) |\, \pi\right]$$

- Applications:
  Robotics, Control, Server Management, Drug Trials, Ad Serving

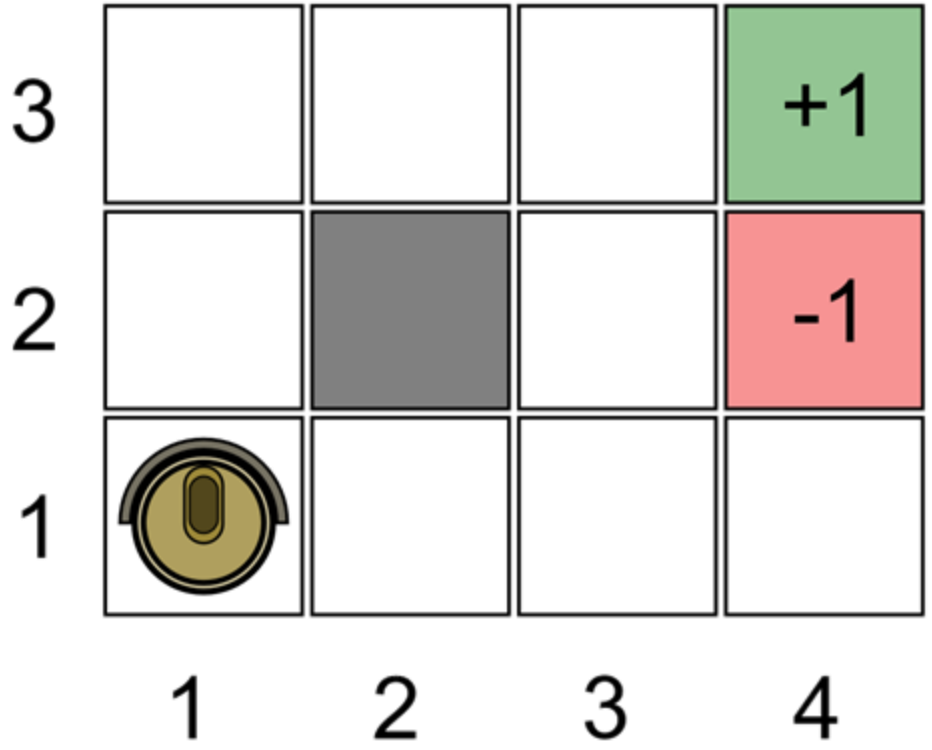
RL Applications

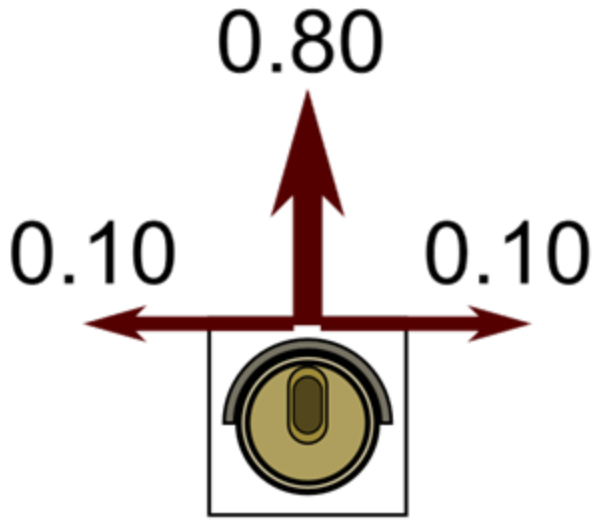
- Fly stunt maneuvers in a helicopter
- Defeat the world champion at Backgammon
- Manage an investment portfolio
- Control a power station
- Make a humanoid robot walk
- Play Atari games better than humans

….



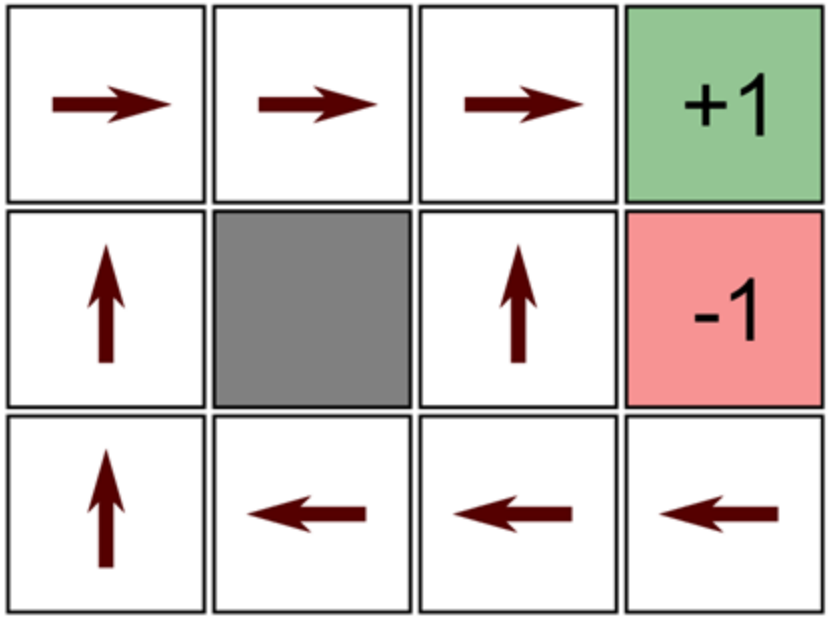
Example





$T(s, a, s')$

Example



optimal policy

0.812	0.868	0.918	+1
0.762		0.660	-1
0.705	0.655	0.611	0.388

utility values

Value Functions

- The *state-value function* V^π is defined as:

$$V^\pi(s) = E[r_0 + r_1 + r_2 + \ldots \mid s_0 = s]$$

Measures *expected future return, starting with state s*

- The *state-action value function* Q^π is defined as

$$Q^\pi(s, a) = E[r_0 + r_1 + r_2 + \ldots \mid s_0 = s, a_0 = a]$$

- The *advantage function* A^π is

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$$

Measures *how much better is action a than what the policy* π *would've done.*

Value of a Policy

$$V^\pi = \mathbb{E}_{s_0}[r_0 + \gamma V^\pi(s')]$$

Optimal Value Function

$$V^* = max_{s_0}[r_0 + \gamma V^*(s')]$$

Value Iteration

- $V^{\pi}(s) = \mathbb{E}[r_0 + \gamma r_1 + \ldots \mid s_0, a_0 = \pi(s_0)] = \mathbb{E}_{s_0}[r_0 + \gamma V^{\pi}(s')]$ Eval
- $V^{*}(s) = \mathbb{E}[r_0 + \gamma r_1 + \ldots \mid s_0, a_0 = \pi^{*}(s_0)] = max_{s_0}[r_0 + \gamma V^{*}(s')]$ Optimal

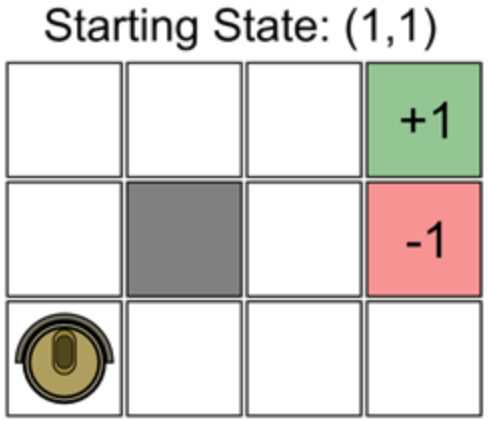
Algorithm 1 Value Iteration

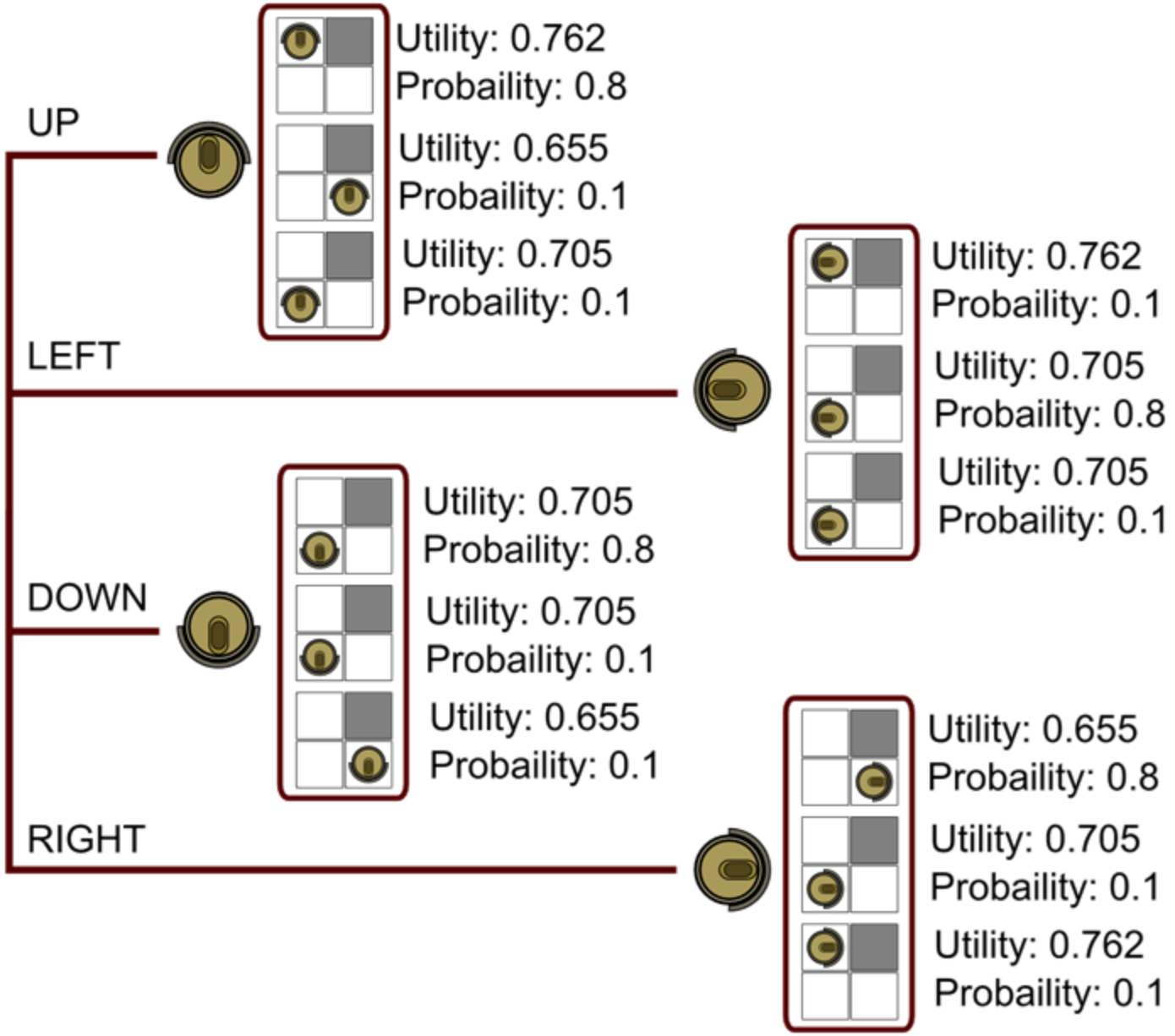
Initialize $V^{(0)}$.
for $n = 1,2,\ldots$ **do**
  for $s \in S$ **do**
    $V^{(n)}(s) = \max_a \sum_{s'} P(s,a,s')(R(s,a,s') + \gamma V^{(n-1)}(s'))$
  end for
  ▷ The above loop over s could be written as $V^{(n)} = TV^{(n-1)}$
end for

Once per n

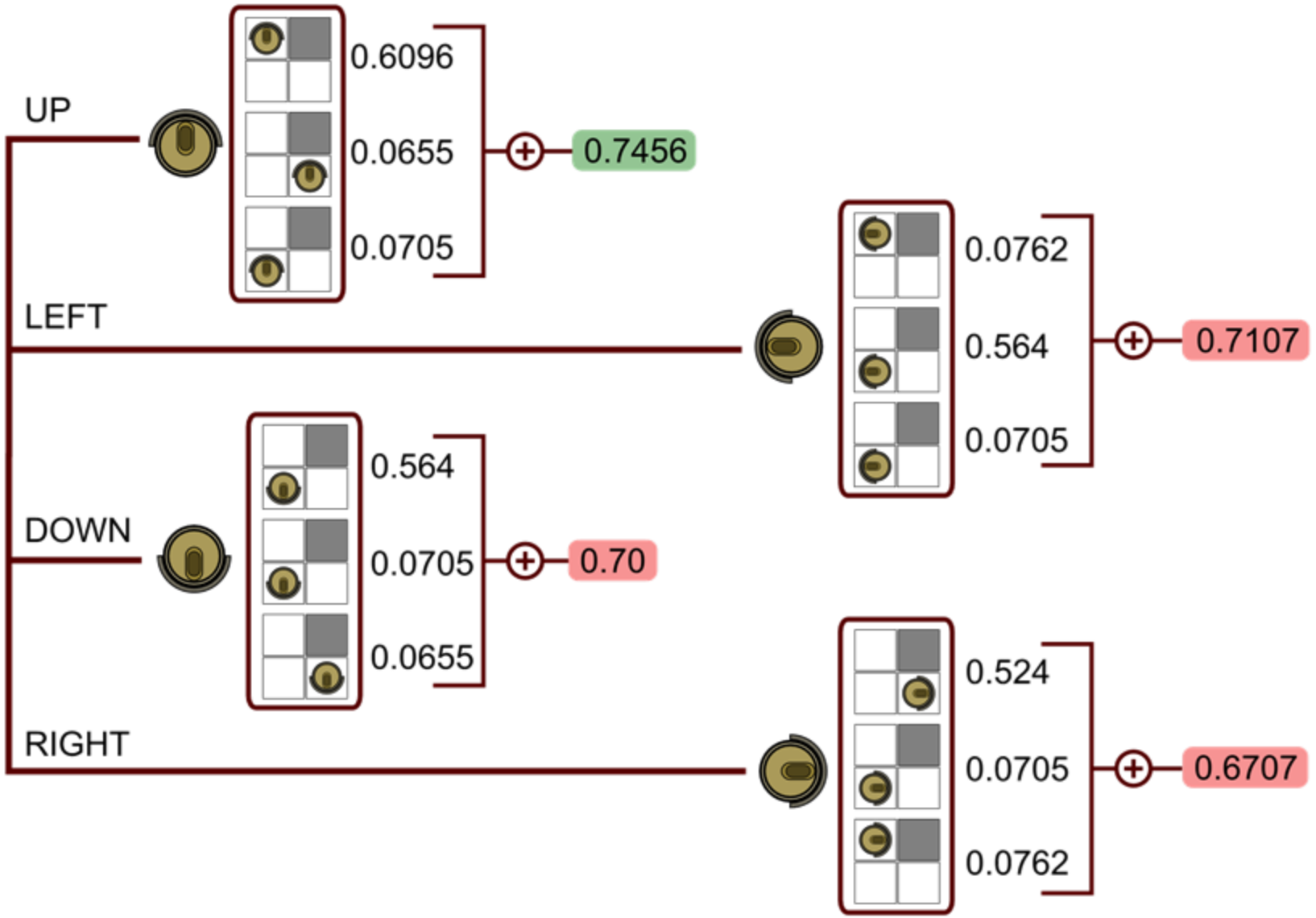
- $\pi^{*} = argmax_a V^{*}(s)$ Optimal

Example


Starting State: (1,1)
+1
-1


UP
Utility: 0.762
Probaility: 0.8
Utility: 0.655
Probaility: 0.1
Utility: 0.705
Probaility: 0.1
LEFT
Utility: 0.762
Probaility: 0.1
Utility: 0.705
Probaility: 0.8
Utility: 0.705
Probaility: 0.1
DOWN
Utility: 0.705
Probaility: 0.8
Utility: 0.705
Probaility: 0.1
Utility: 0.655
Probaility: 0.1
RIGHT
Utility: 0.655
Probaility: 0.8
Utility: 0.705
Probaility: 0.1
Utility: 0.762
Probaility: 0.1

Example


UP
0.6096
0.0655
0.0705
0.7456
LEFT
0.0762
0.564
0.0705
0.7107
DOWN
0.564
0.0705
0.0655
0.70
RIGHT
0.524
0.0705
0.0762
0.6707

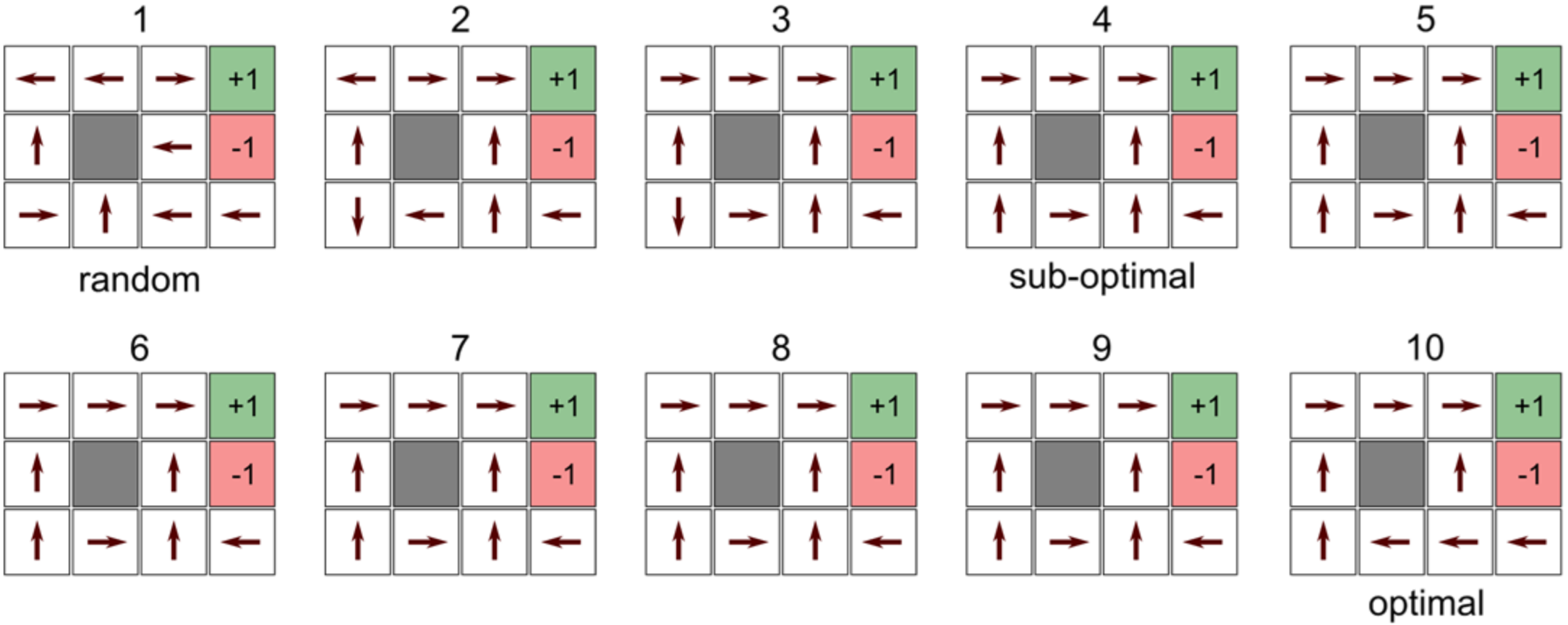
Policy Iteration

- $V^{\pi} = \mathbb{E}[r_0 + \gamma r_1 + \ldots \mid s_0, a_0 = \pi(s_0)] = \mathbb{E}_{s_0}[r_0 + \gamma V^{\pi}(s')]$ Eval
- $V^{*} = \mathbb{E}[r_0 + \gamma r_1 + \ldots \mid s_0, a_0 = \pi^{*}(s_0)] = max_{s_0}[r_0 + \gamma V^{*}(s')]$ Optimal

Algorithm 2 Policy Iteration

Initialize $\pi^{(0)}$.
for $n = 1,2,\ldots$ **do**
  $V^{(n-1)} = \text{Solve}[V = T^{\pi^{(n-1)}} V]$ } Once per n, but needs many iters
  for $s \in S$ **do**
    $\pi^{(n)}(s) = \text{argmax}_a \sum_{s'} P(s,a,s')[R(s,a,s') + \gamma V^{(n-1)}(s')]$
    $= \text{argmax}_a Q^{\pi^{(n-1)}}(s,a)$ } few updates
  end for
end for

Example – Iterates in Policy Iteration

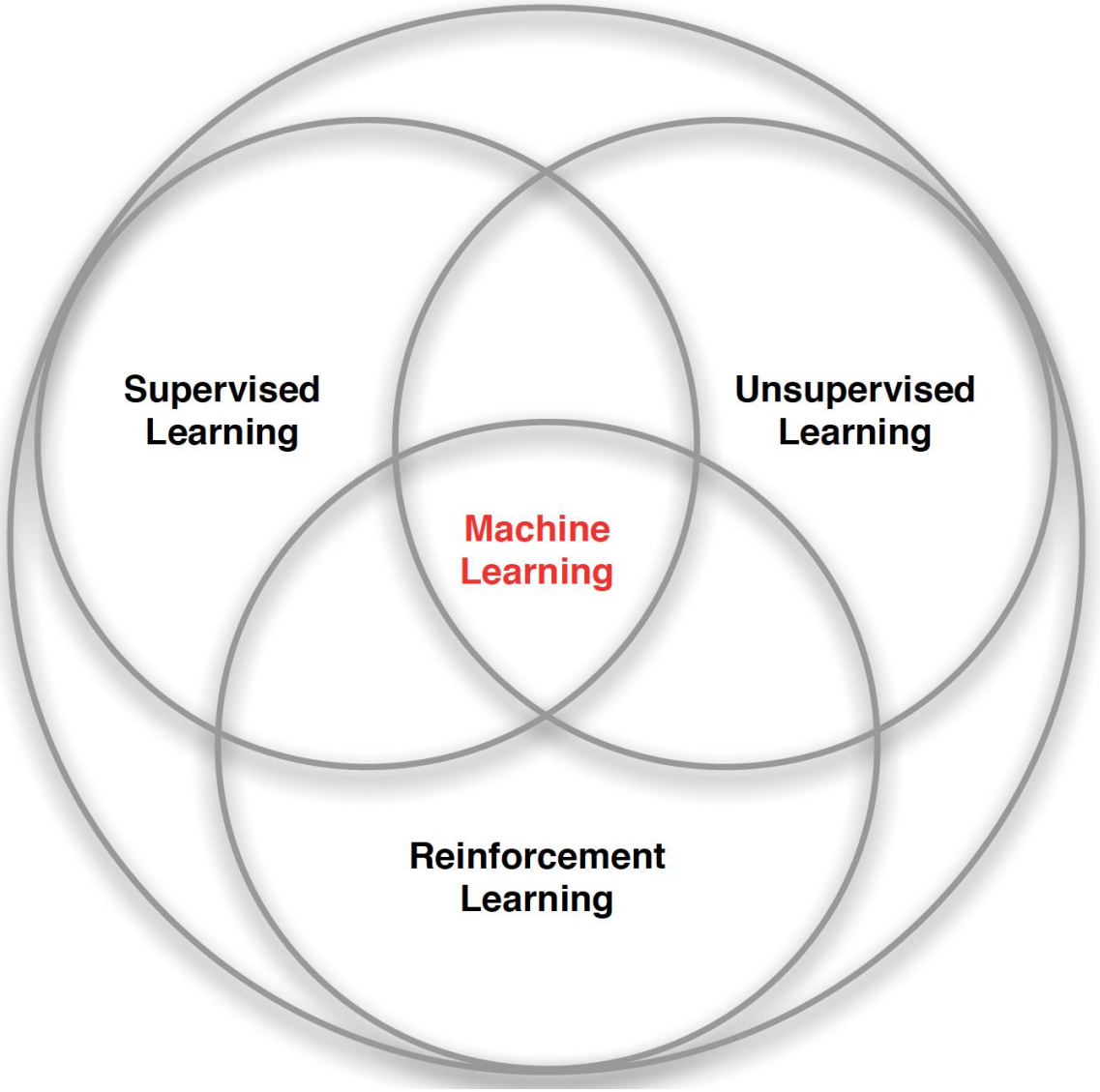


10 Updates in Policy Iteration, Same needs 16 Value Updates

What is not RL

- Supervised Learning
  - Train: x_i, y_i
  - Prediction: $\hat{y}_i = f(x_i)$
  - Loss $l(y_i, \hat{y}_i)$
- Contextual Bandits
  - Train: x_i
  - Prediction: $\hat{y}_i = f(x_i)$
  - Reward c_i

RL ≠ Supervised Learning, Bandits



Why is RL Different and Hard

- No Supervisor: Reward Signal
- Delayed Feedback: Credit Assignment is hard!
- Sequential Decision making: Time Matters
- Each Prediction affects Subsequent Examples: Data is not IID

How to identify an RL Problem

- Reward as an Oracle

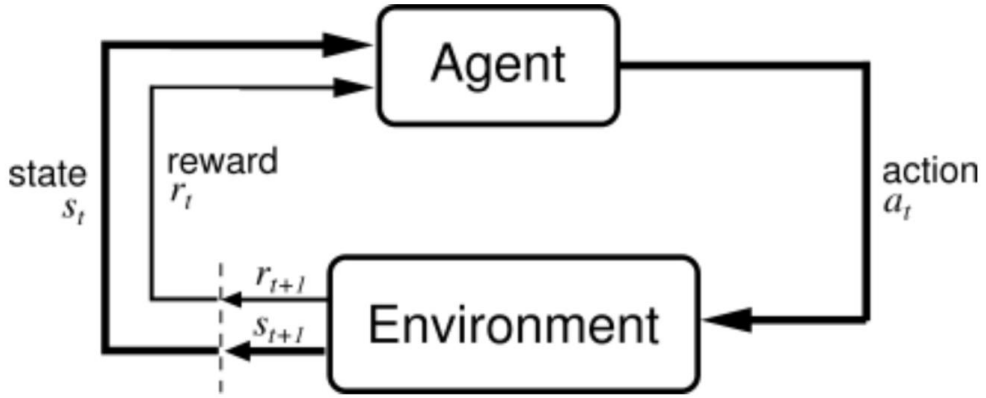
  Analytic function is not available

- State-ful

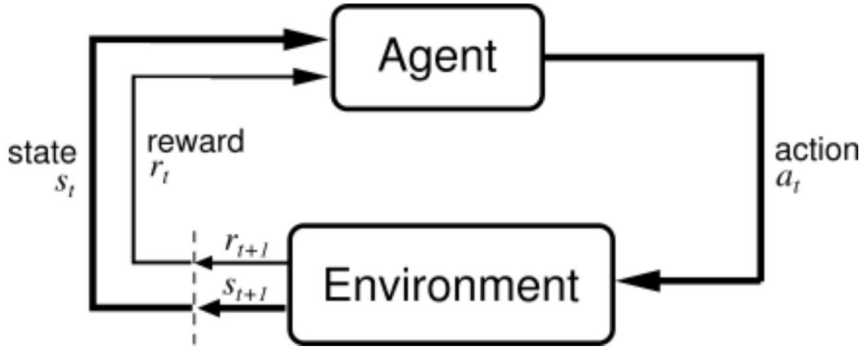
  The state evolves as a function of previous state action

RL Applications: Reward Model

- Fly stunt maneuvers in a helicopter
  - +ve reward for following desired trajectory
  - - ve reward for crashing
- Defeat the world champion at Backgammon
  - +/- ve reward for winning/losing a game
- Manage an investment portfolio
  - +ve reward for each $ in bank
- Control a power station
  - +ve reward for producing power
  - -ve reward for exceeding safety thresholds
- Make a humanoid robot walk
  - +ve reward for forward motion
  - -ve reward for falling over
- Play many dierent Atari games better than humans
  - +/- ve reward for increasing/decreasing score



Reinforcement Learning: MDP



$$\mathcal{M} = \langle S, A, P(\cdot,\cdot), R(\cdot,\cdot), T \rangle$$

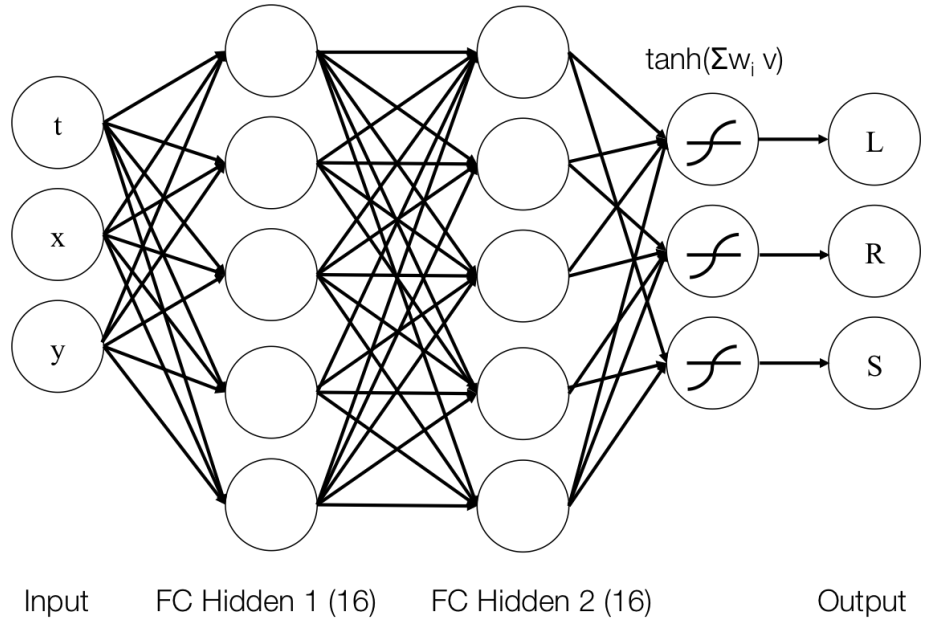
State Space | Action Space | Transition Function | Reward Function | Time Horizon

$Prob: S \times A \to S$ $\quad$ $R: S \times A \to \mathbb{R}$

Goal: Find Optimal Policy: $\pi^*: S \to A$

What is the Deep in Deep RL

- Value Function: Map state value to $\mathbb{R}$
- Policy: map input (say, image) to action
- Dynamics Model: Map $P(x_{t+1} \mid x_t, a_t)$



When is RL not a good idea?

- Which decision making problem either can't or shouldn't be formulated as RL

- The agent needs ability to try, and fail.
- Failure/Safety is a problem?
- What about very long horizon.
  Goal in Primary School – Win "Turing Award/Nobel Prize"

RL isn’t a Silver Bullet

- Derivative Free Optimization
  - Cross-Entropy Method
  - Evolutionary Methods
- Bandit Problems
  - Not State-ful
- Contextual Bandits
  - Special case with side information

Agenda

- Logistics
- Course Motivation
- Primer in RL
- Human learning and RL (sample paper presentation)
- Presentation Sign-ups

Human Learning in Atari*

Tsivdis, Pouncy, Xu, Tenenbaum, Gershman

Topic: Human Learning & RL

Presenter: Animesh Garg

with thanks to Sam Gershman sharing slides from RLDM 2017

*This presentation also serves as a worked example of type of expected presentation

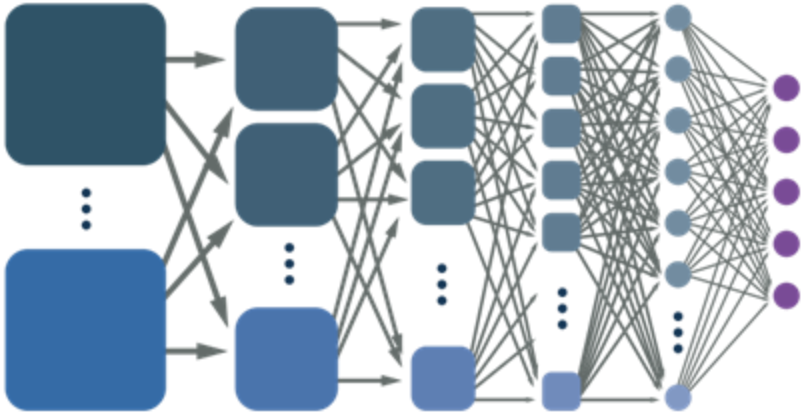
Motivation and Main Problem

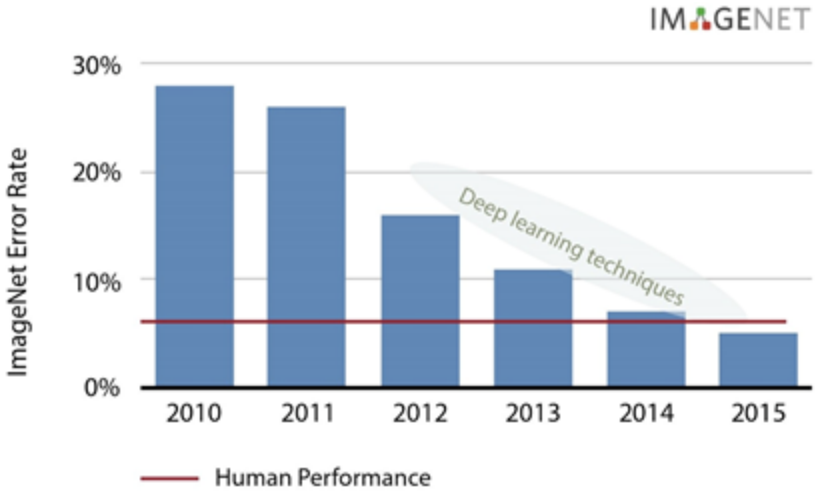
1-4 slides

Should capture

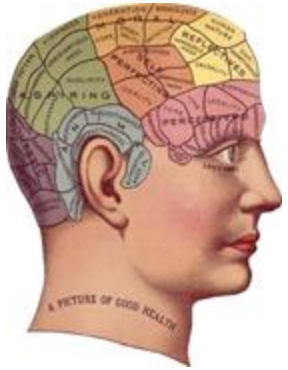
- High level description of problem being solved (can use videos, images, etc)
- Why is that problem important?
- Why is that problem hard?
- High level idea of why prior work didn't already solve this (Short description, later will go into details)

A Seductive Hypothesis



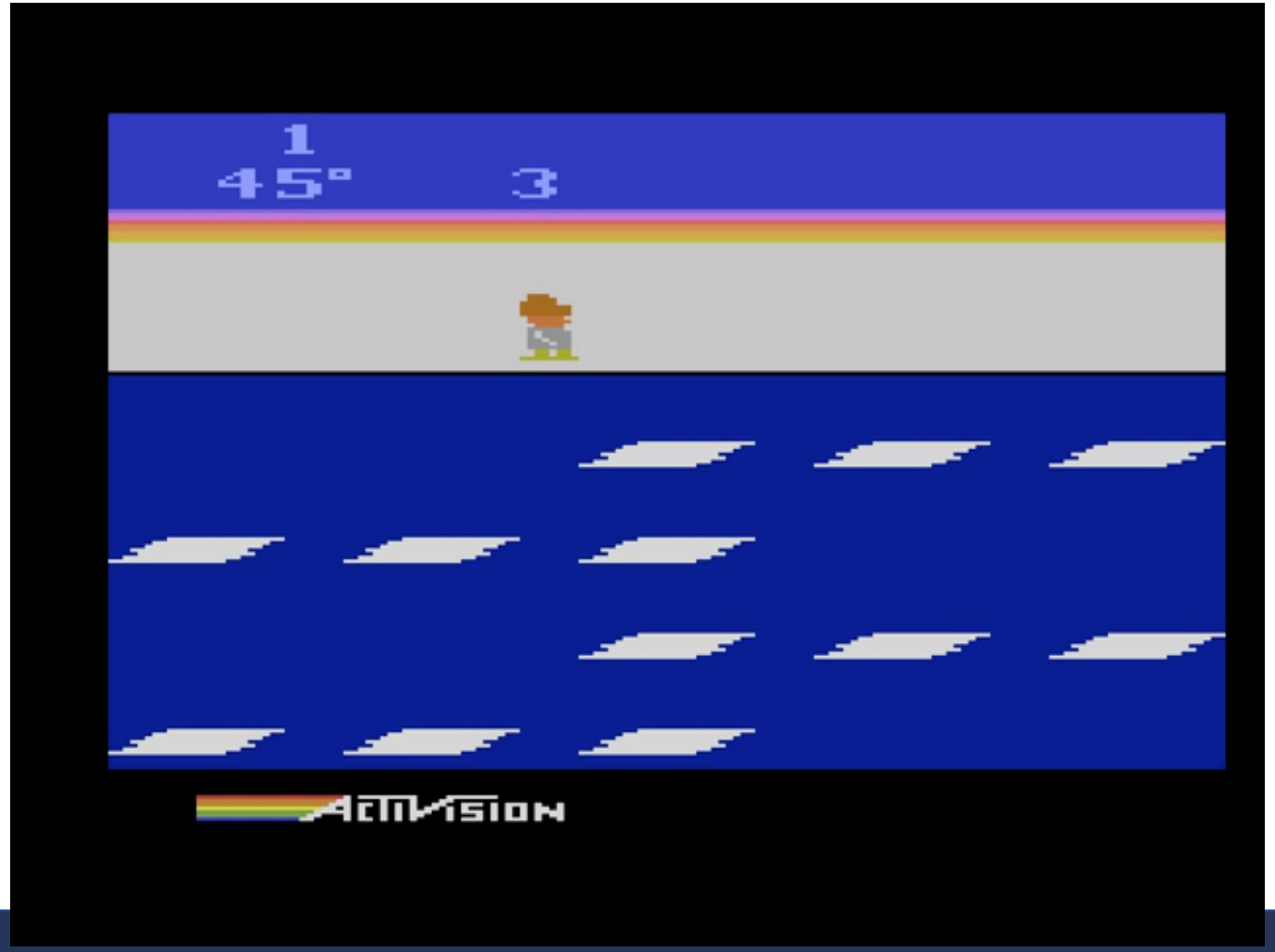


Brain-like computation + Human-level performance

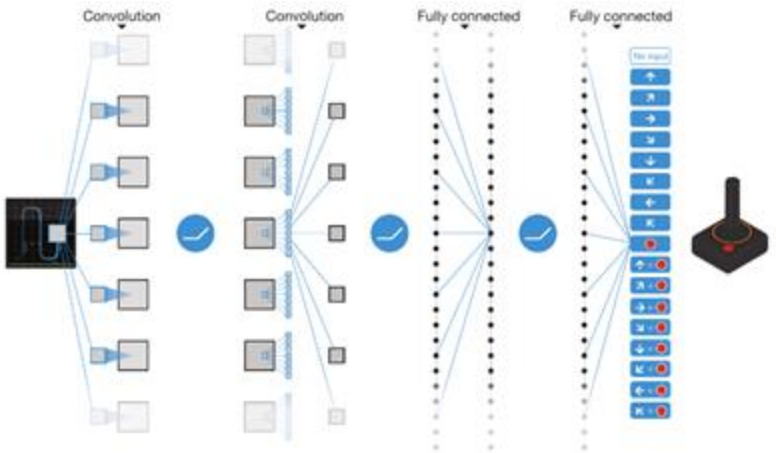


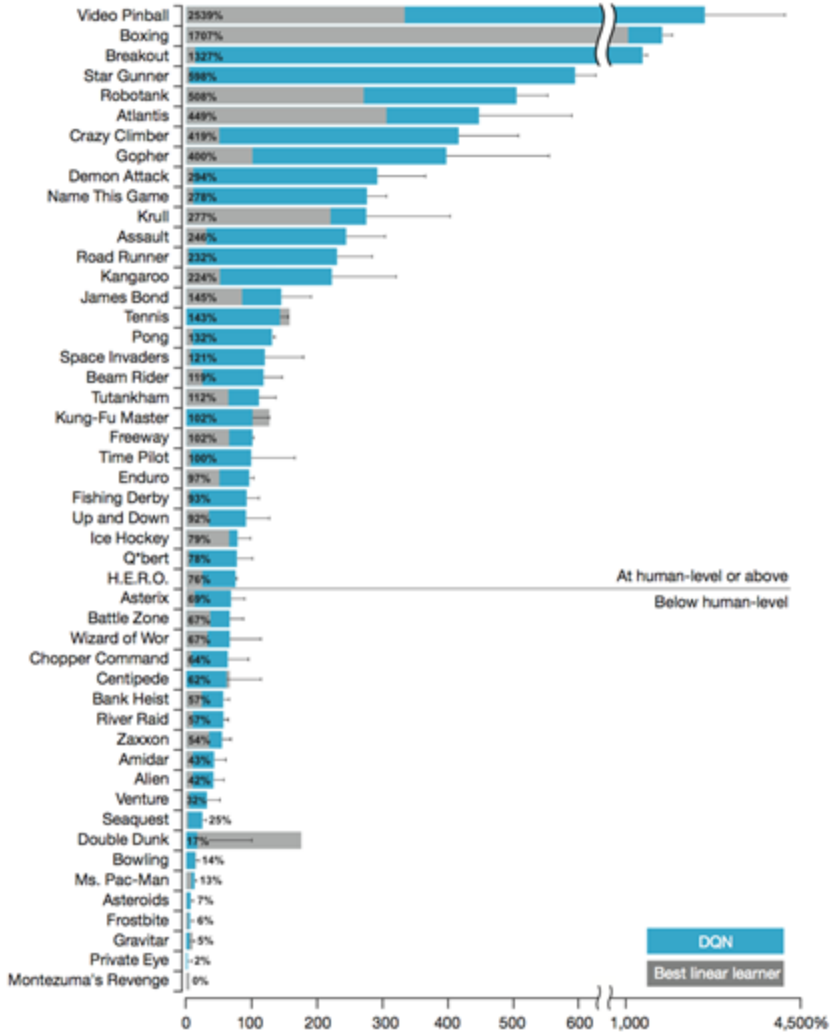
= Human intelligence?

Atari: a Good Testbed for Intelligent Behavior



Mastering Atari with deep Q-learning





Mnih et al. (2015)

Is this how humans learn?

Is this how humans learn?

Key properties of human intelligence:

1. Rapid learning from few examples.
2. Flexible generalization.

These properties are not yet fully captured by deep learning systems.

Contributions

Approximately one bullet, high level, for each of the following (the paper on 1 slide).

- Problem the reading is discussing
- Why is it important and hard
- What is the key limitation of prior work
- What is the key insight(s) (try to do in 1-3) of the proposed work
- What did they demonstrate by this insight? (tighter theoretical bounds, state of the art performance on X, etc)

Contributions

- **Problem:** Want to understand how people play Atari

Contributions

- **Problem:** Want to understand how people play Atari
- **Why is this problem important?**
  - Because Atari games seem like a good involve tasks with widely different visual aspects, dynamics and goals presented
  - Lots of success of deep RL agents but require a lot of training
  - Do people do this too? If not, what might we learn from them?

Contributions

- **Problem:** Want to understand how people play Atari
- **Why is this problem important?**
  - Because Atari games seem like a good involve tasks with widely different visual aspects, dynamics and goals presented
  - Lots of success of deep RL agents but require a lot of training
  - Do people do this too? If not, what might we learn from them?
- **Why is that problem hard?** Much unknown about human learning
- **Limitations of prior work**: Little work on human atari performance

Contributions

- **Problem:** Want to understand how people play Atari
- **Why is this problem important?**
  - Because Atari games seem like a good involve tasks with widely different visual aspects, dynamics and goals presented
  - Lots of success of deep RL agents but require a lot of training
  - Do people do this too? If not, what might we learn from them?
- **Why is that problem hard?** Much unknown about human learning
- **Limitations of prior work**: Little work on human atari performance
- **Key insight/approach**: Measure people's performance. Test idea that people are building models of object/relational structure
- **Revealed:** People learning much faster than Deep RL. Interventions suggest people can benefit from high level structure of domain models and use to speed learning.

General Background

1 or more slides

The background someone needs to understand this paper

That wasn’t just covered in the chapter/survey reading presented earlier in class during same lecture (if there was such a presentation)

Background: Prioritized Replay

Schaul, Quan, Antonoglou, Silver ICLR 2016

- Sample (s,a,r,s') tuple for update using priority
- Priority of a tuple is proportional to DQN error

$$p_i = \left| r + \gamma \max_{a'} Q(s', a', \mathbf{w}^-) - Q(s, a, w) \right|$$

- Update probability P(i) is proportional to DQN error $P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$
- α=0, uniform
- Update p_i every update
- Can yield substantial improvements in performance

Problem Setting

1 or more slides

Problem Setup, Definitions, Notation

Be precise-- should be as formal as in the paper

Approach / Algorithm / Methods (if relevant)

Likely >1 slide

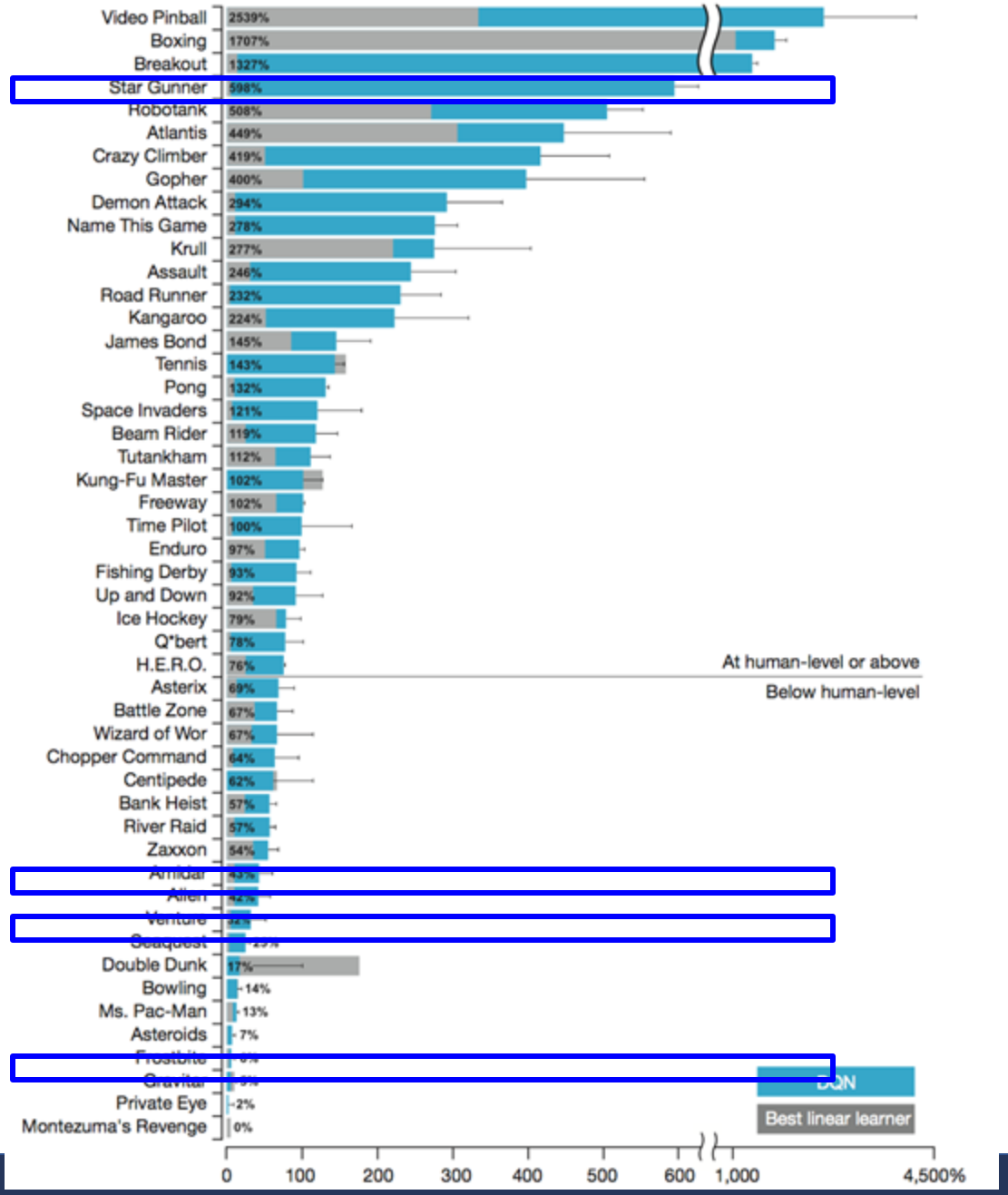
Describe algorithm or framework (pseudocode and flowcharts can help)

What is it trying to optimize?

Implementation details should be left out here, but may be discussed later if its relevant for limitations / experiments

Methods: Observation & Experiment

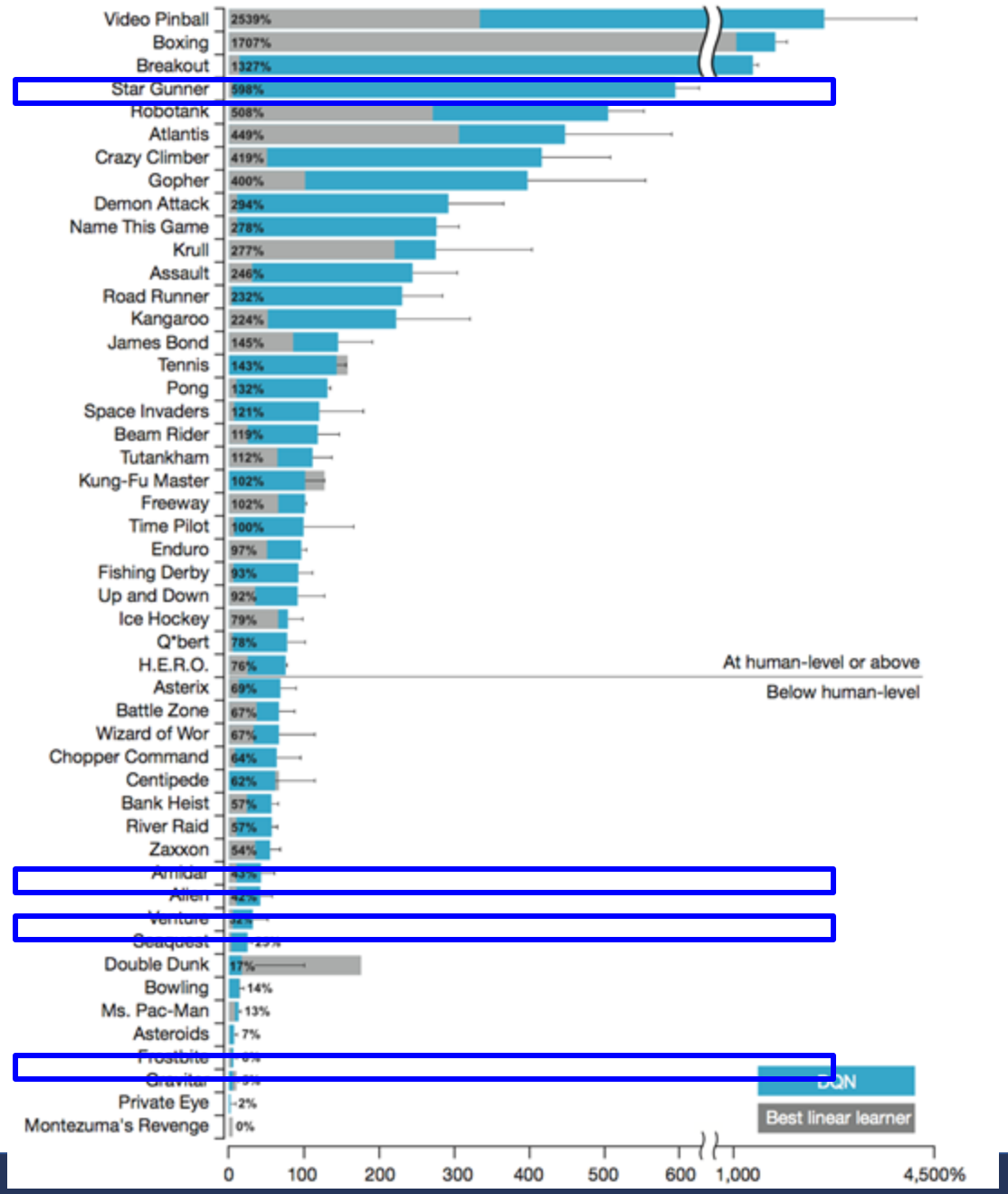
1. Human learning curves in 4 Atari games
2. How initial human performance is impacted by 3 interventions



Star Gunner

Amidar
Venture

Frostbite



Star Gunner

- 2 games where humans eventually outperform Deep RL
- 2 where Deep RL outperforms humans

Amidar
Venture

Frostbite

Human Learning in 4 Atari Games: Setting

- Amazon Mechanical Turk participants
  - Assigned to play a game said haven't played before
  - Play for at least 15 minutes
- Paid $2 and promised bonus up to $2 based on score
- Instructions
  - Could use arrow keys and space bar
  - Try to figure out how game worked to play well
- Subjects
  - 71 Frostbite
  - 18 Venture
  - 19 Amidar
  - 19 Stargunner

Human Learning in 4 Atari Games: Setting

Amazon Mechanical Turk participants

- Assigned to play a game said haven't played before
- Play for at least 15 minutes

All adults. What if we'd done this with children or teens?

- Paid $2 and promised bonus up to $2 based on score

Specifies the reward/incentive model for people

Instructions

- Could use arrow keys and space bar
- Try to figure out how game worked to play well

Is this telling people to build a model?

Subjects

- 71 Frostbite
- 18 Venture
- 19 Amidar
- 19 Stargunner

- Compared to Prioritized Replay Results (Schaul 2015)

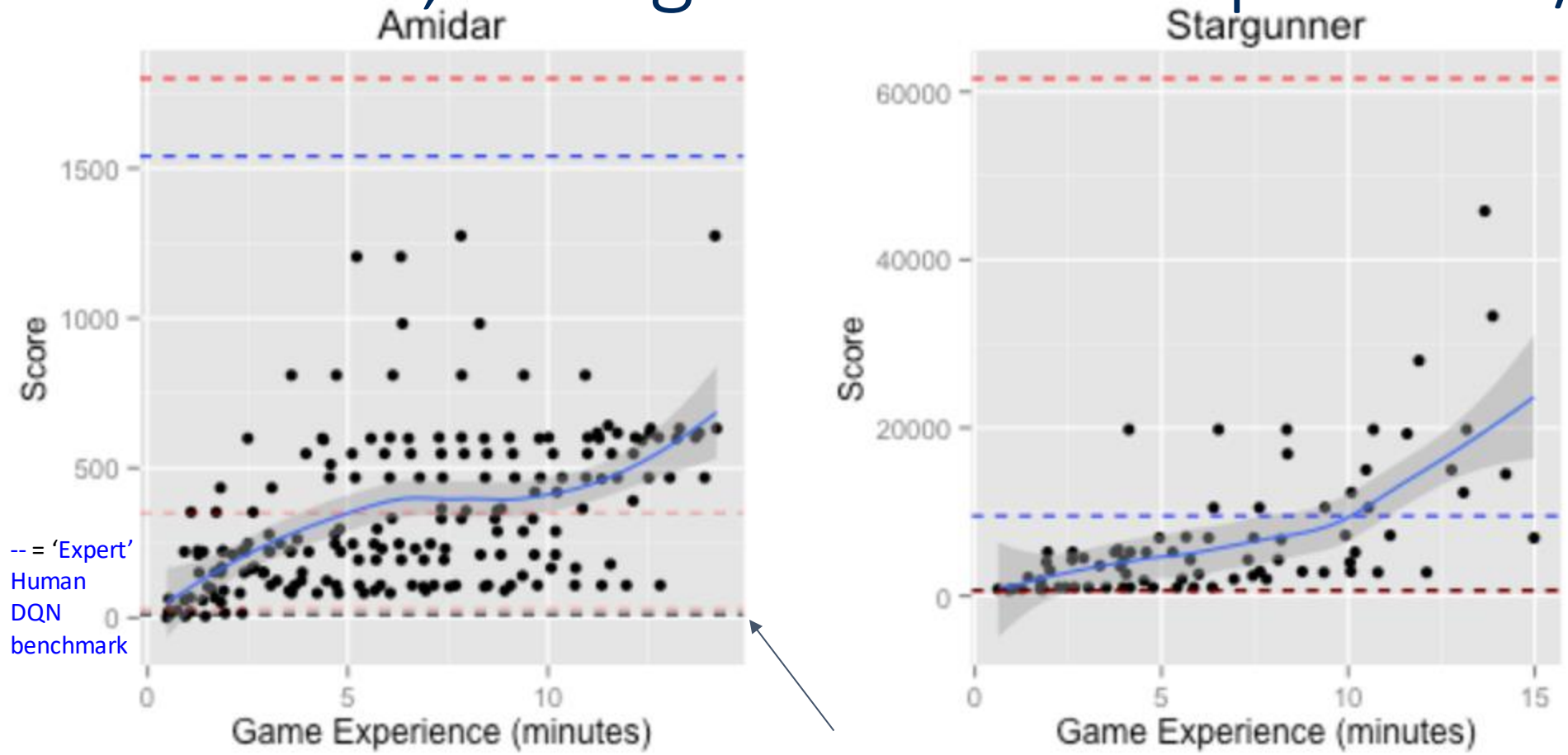
Experimental Results

>=1 slide

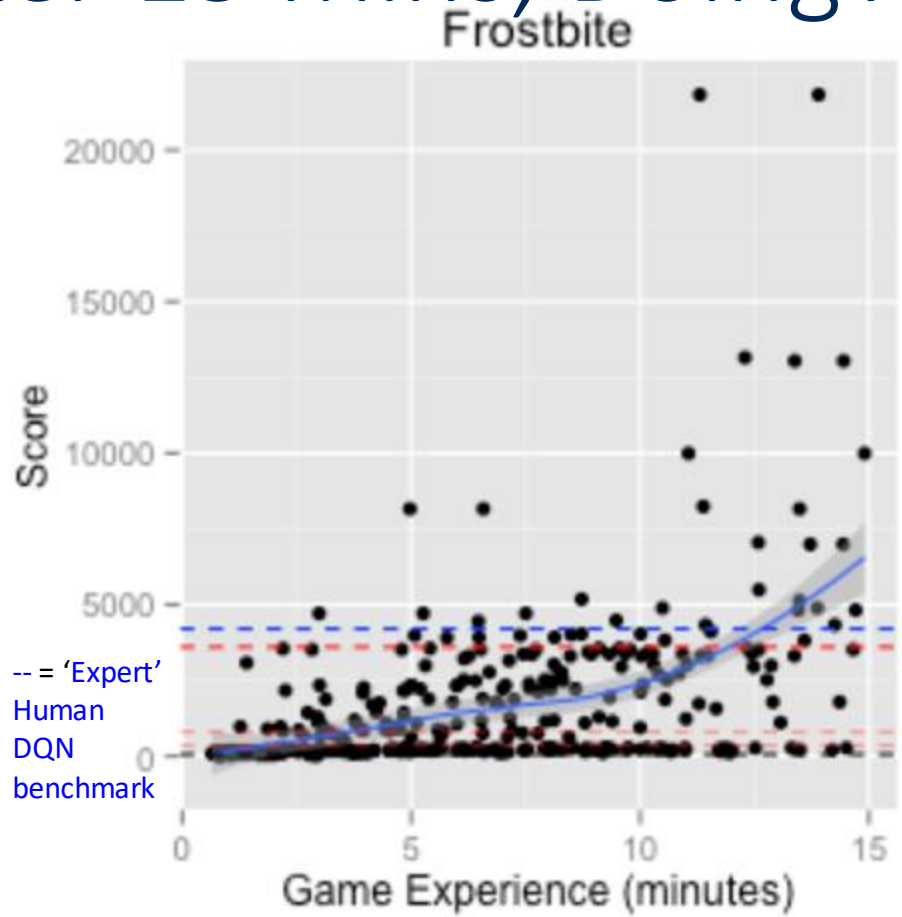
State results

Show figures / tables / plots

After 15 Mins, Doing As Well As Expert in 3/4

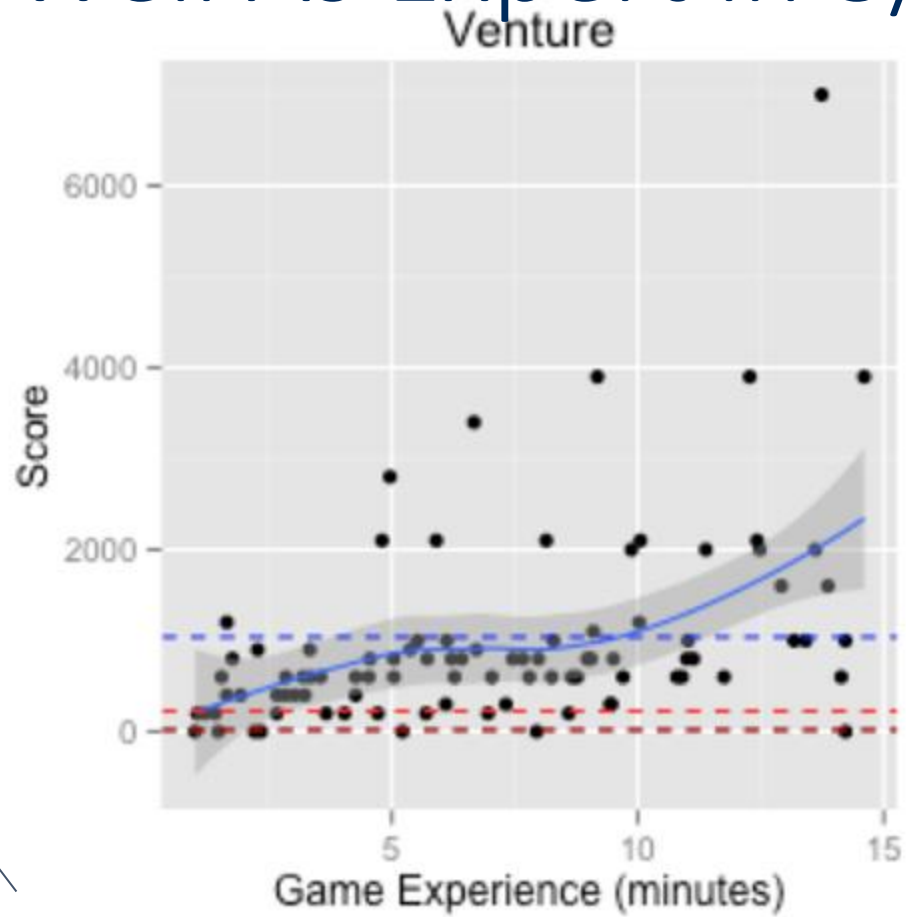


After 15 Mins, Doing As Well As Expert in 3/4



-- = 'Expert' Human DQN benchmark

-- = DQN after 46 / 115/ 920 hrs



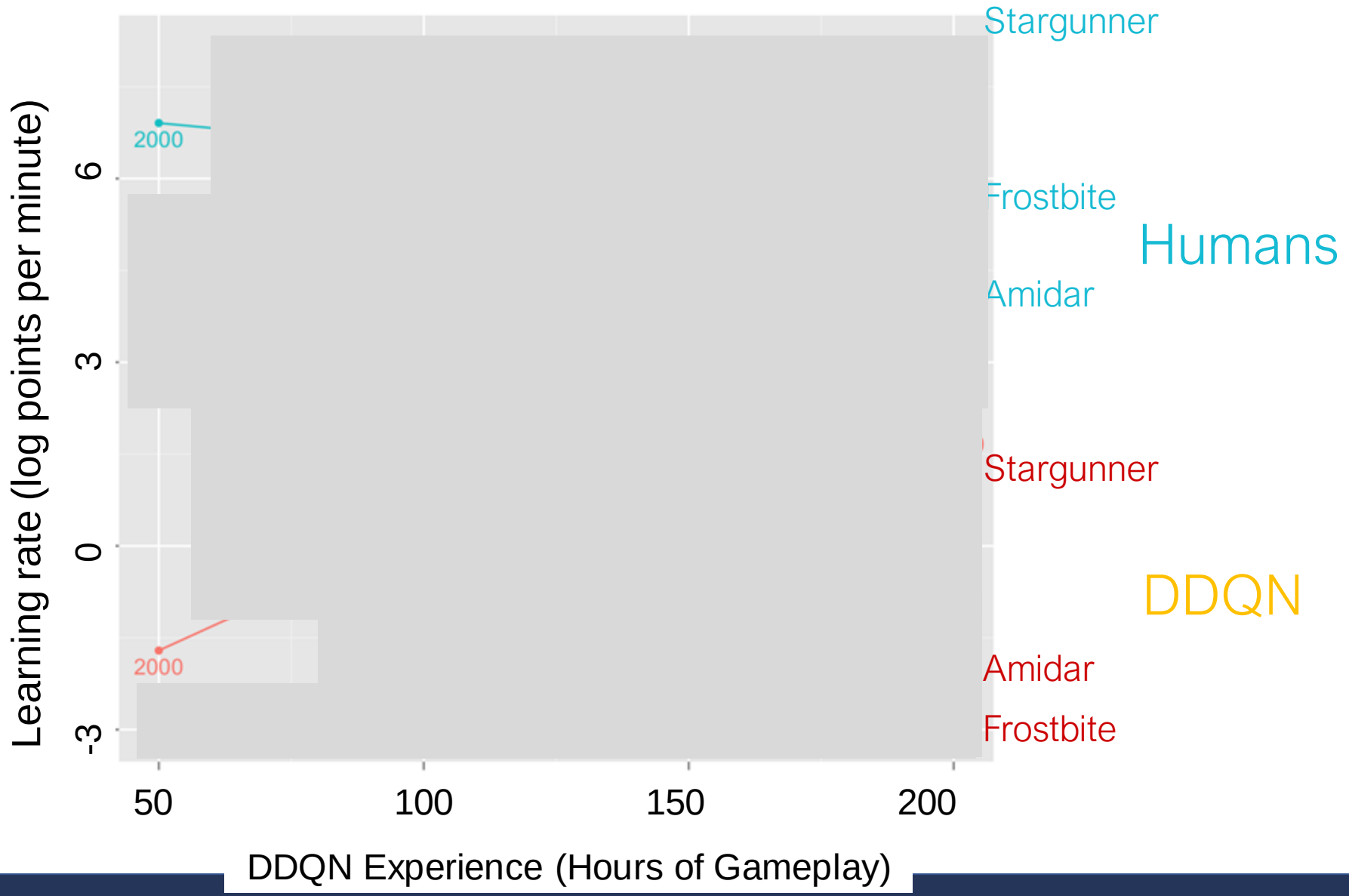
-- = Random play

Unfair Comparison

- Deep neural networks (at least in the way they're typically trained) must learn their entire visual system from scratch.
- Humans have their entire childhoods plus hundreds of thousands of years of evolution.
- Maybe deep neural networks learn like humans, but their learning curve is just shifted.

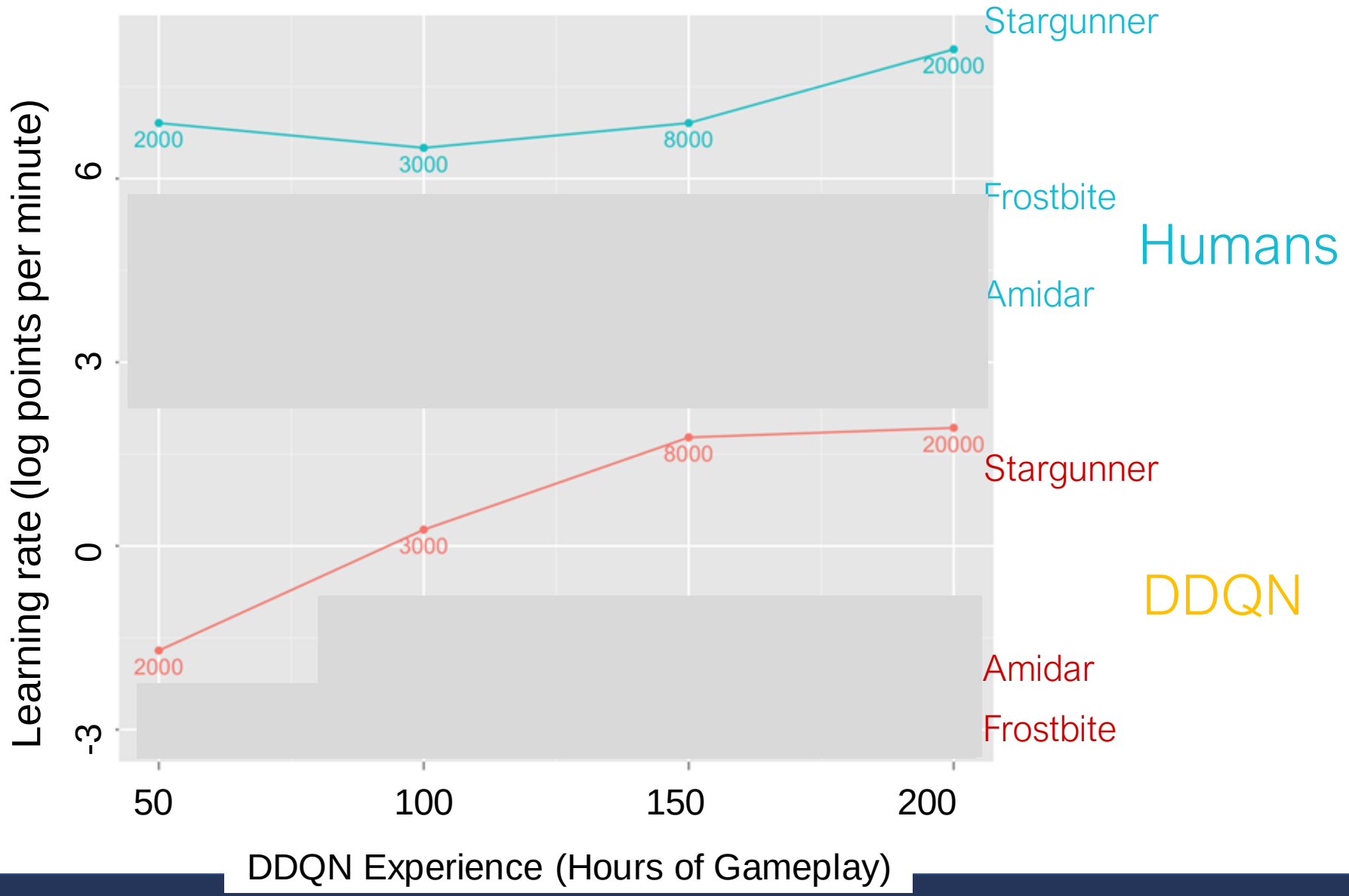
Learning rates matched for score level

Note: Y-axis is in Log!

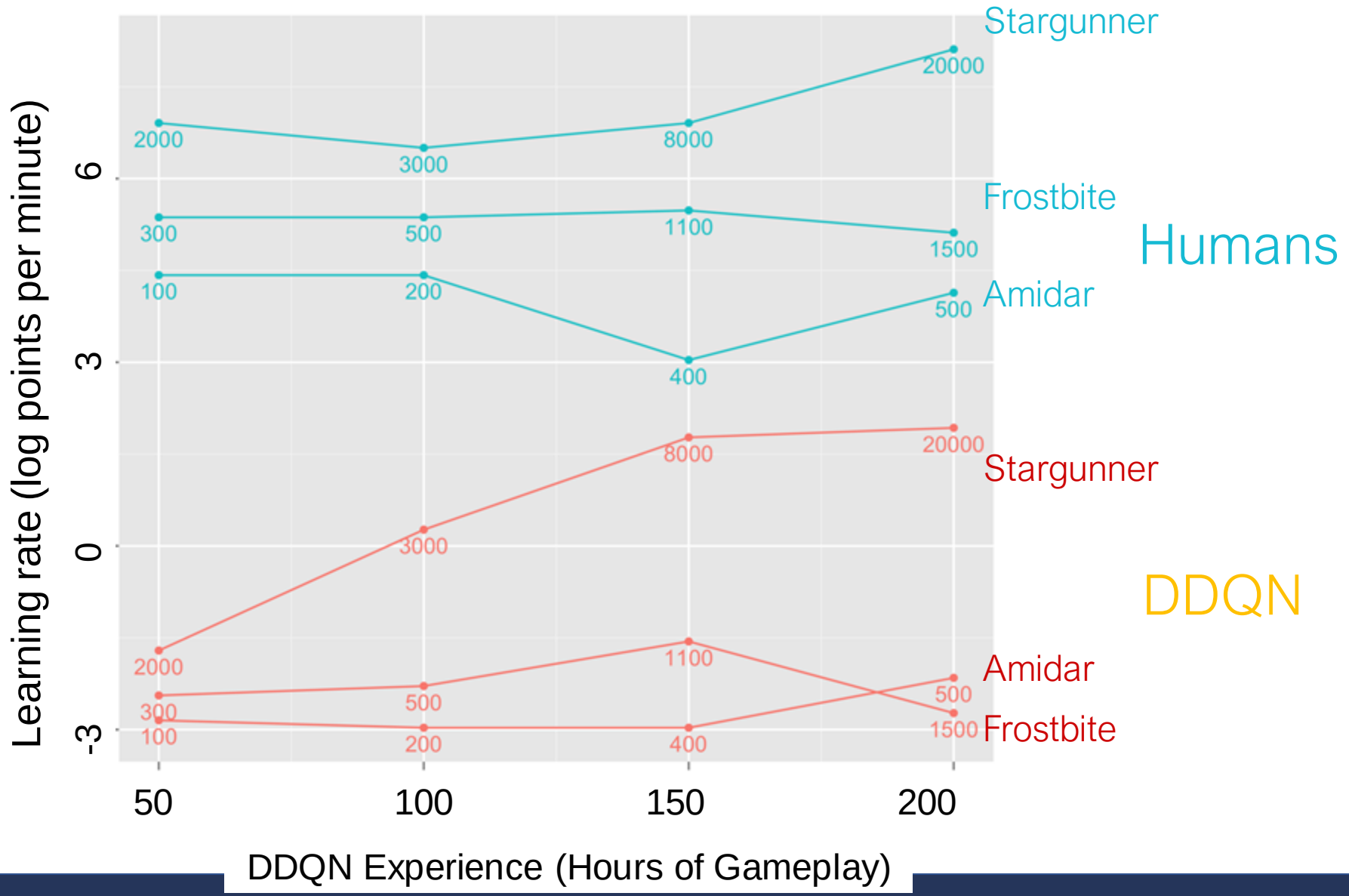


People are Learning Faster at Each Stage of Performance

Note: Y-axis is in Log!



People are Learning Faster at Each Stage of Performance
And This is True in Multiple Games

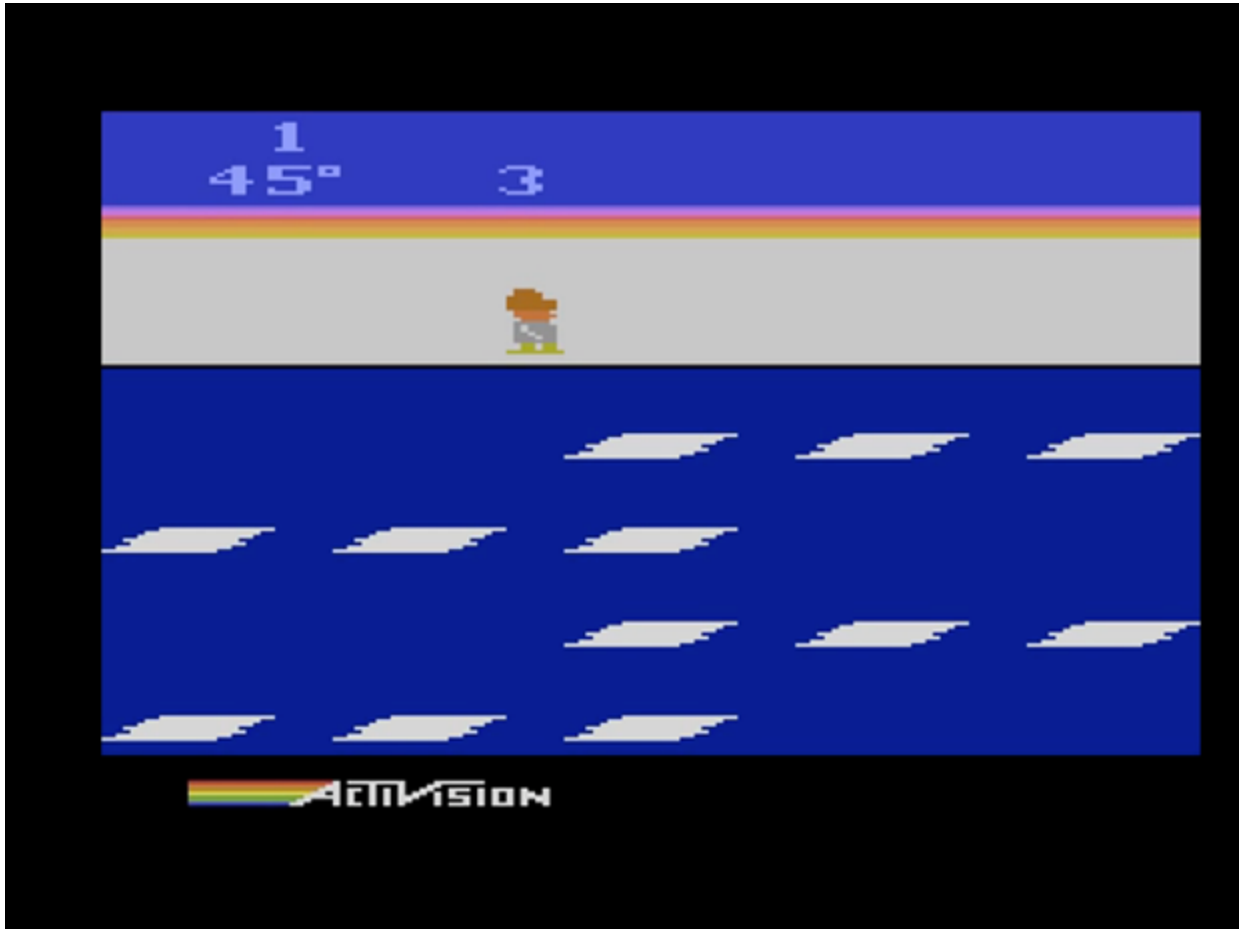


Methods: Observation & Experiment

1. Human learning curves in 4 Atari games
2. **How initial human performance in Frostbite is impacted by 3 interventions**

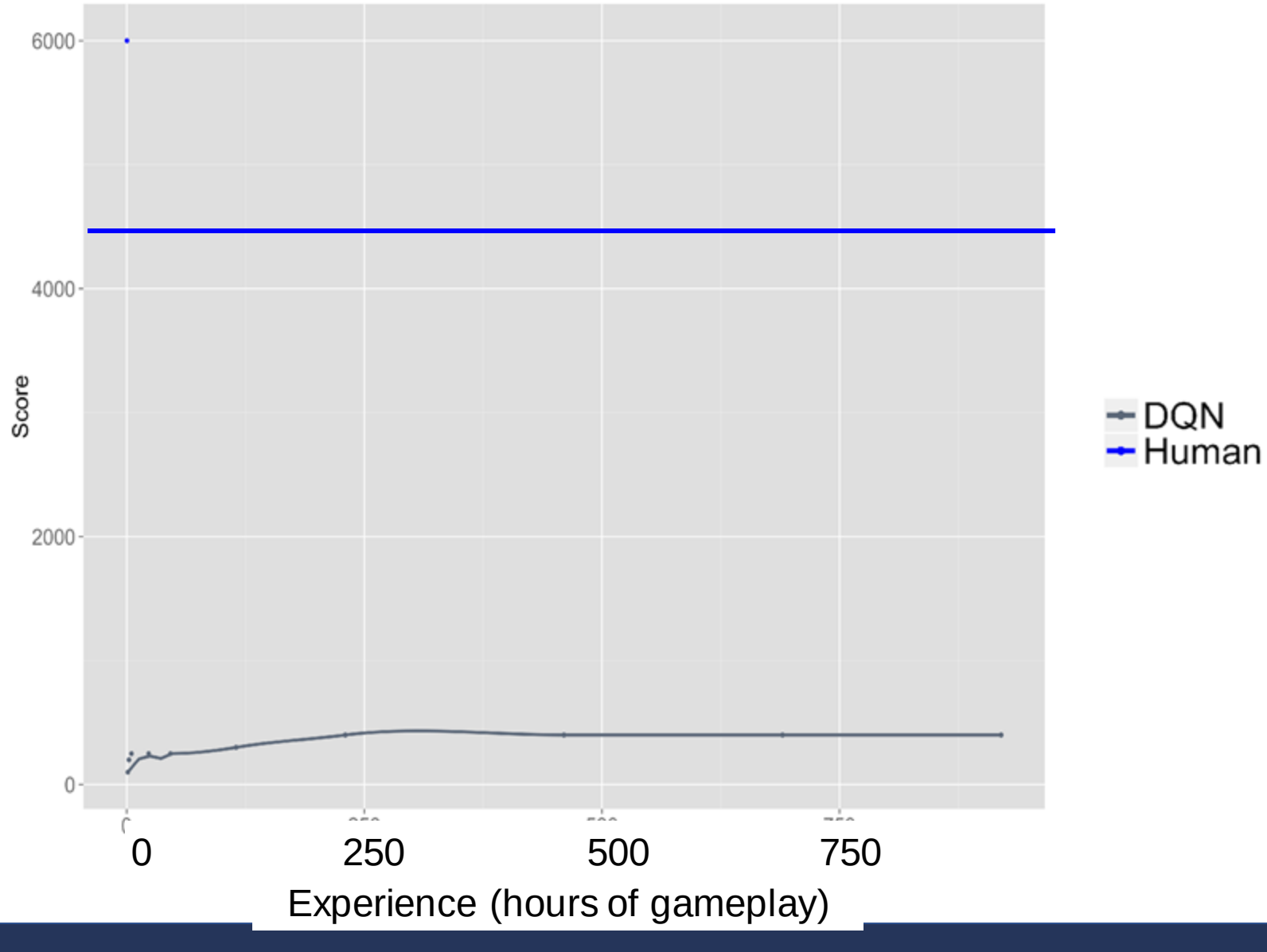
The “Frostbite challenge”

Why Frostbite? People do particularly well vs DDQN

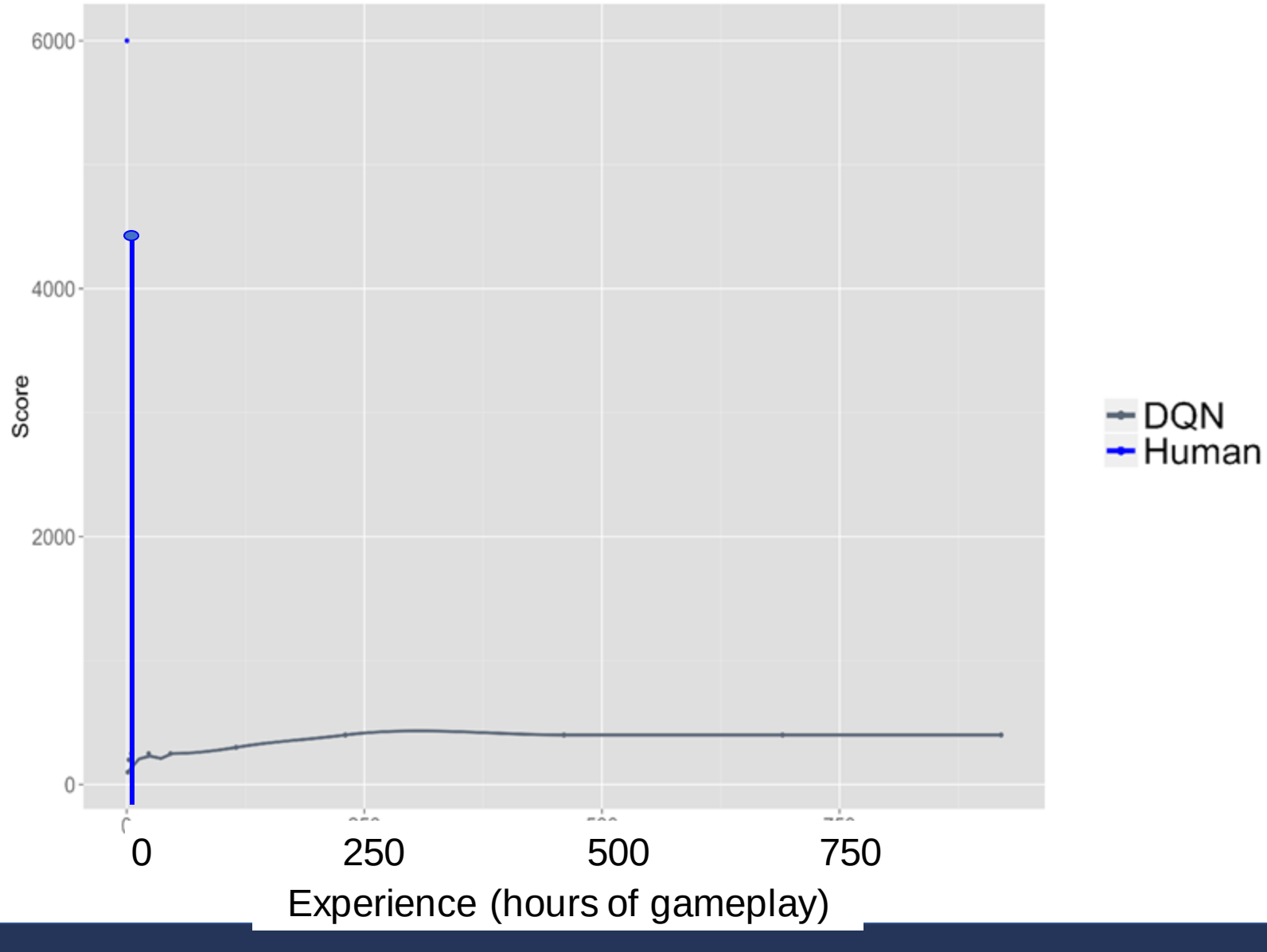


See Lake, Ullman, Tenenbaum & Gershman (forthcoming). Building machines that learn and think like people. *Behavioral and Brain Sciences.*

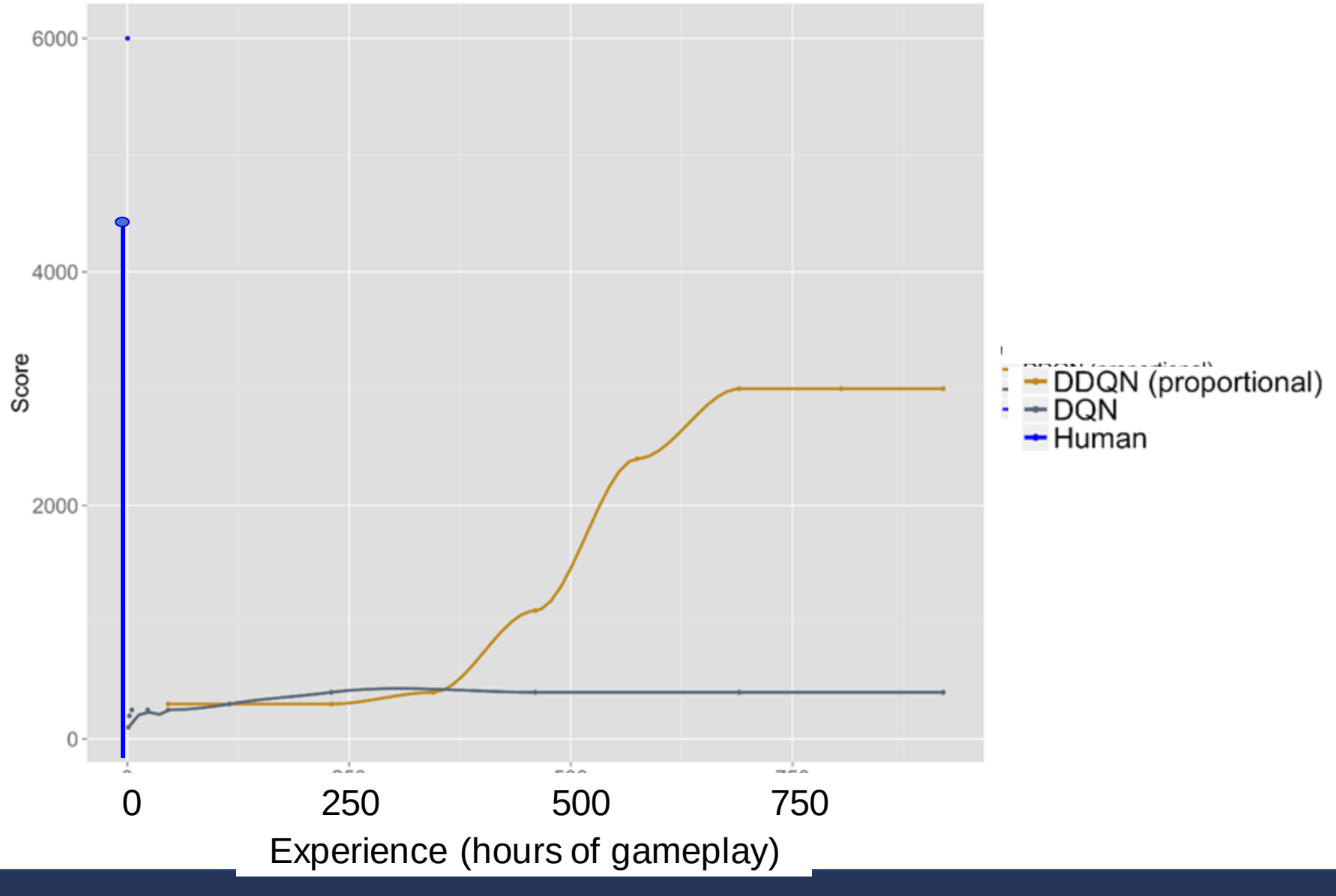
Frostbite


6000
4000
2000
0
Score
0
250
500
750
Experience (hours of gameplay)
DQN
Human

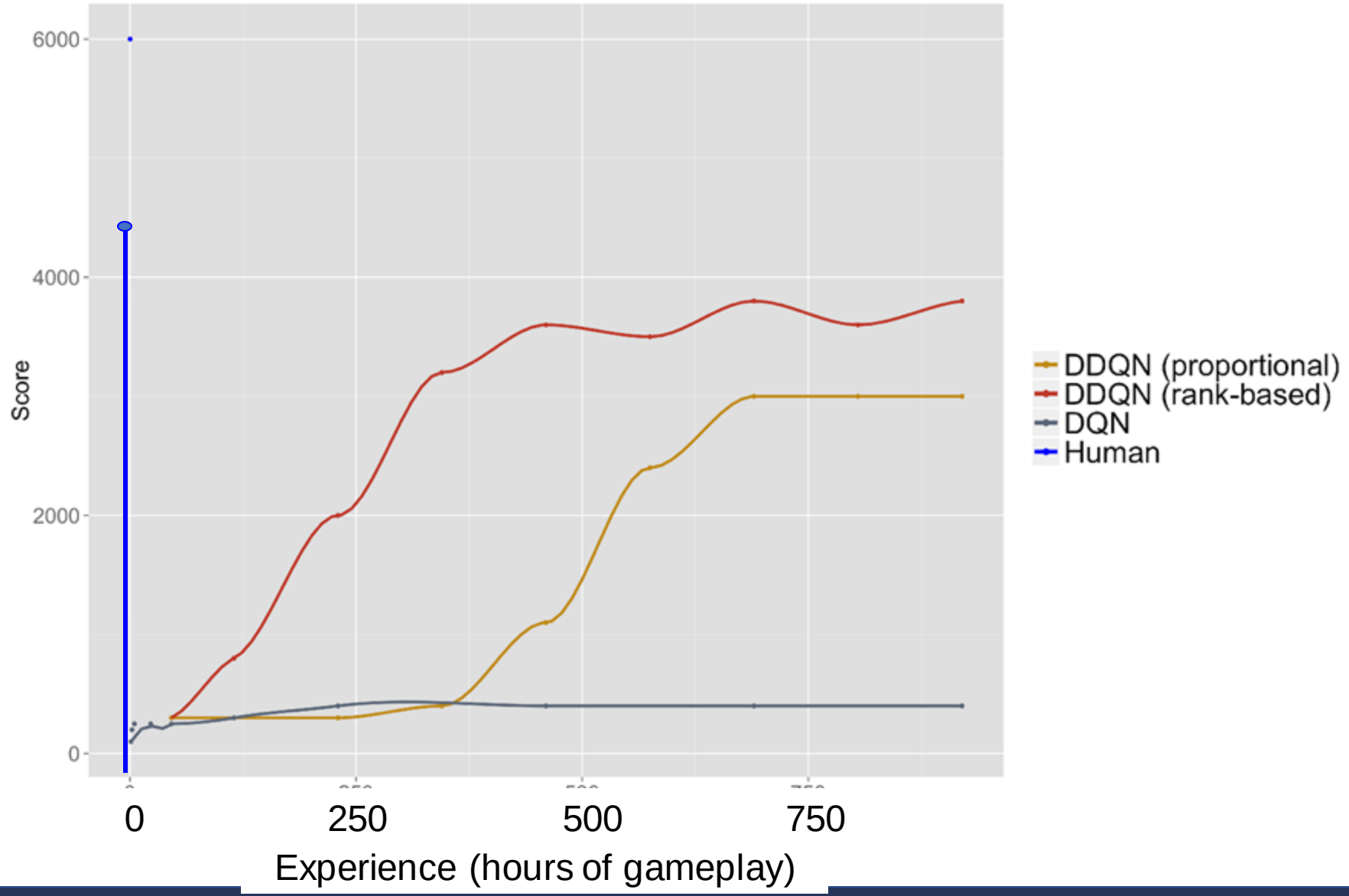
Frostbite


6000
4000
2000
0
Score
0
250
500
750
Experience (hours of gameplay)
DQN
Human

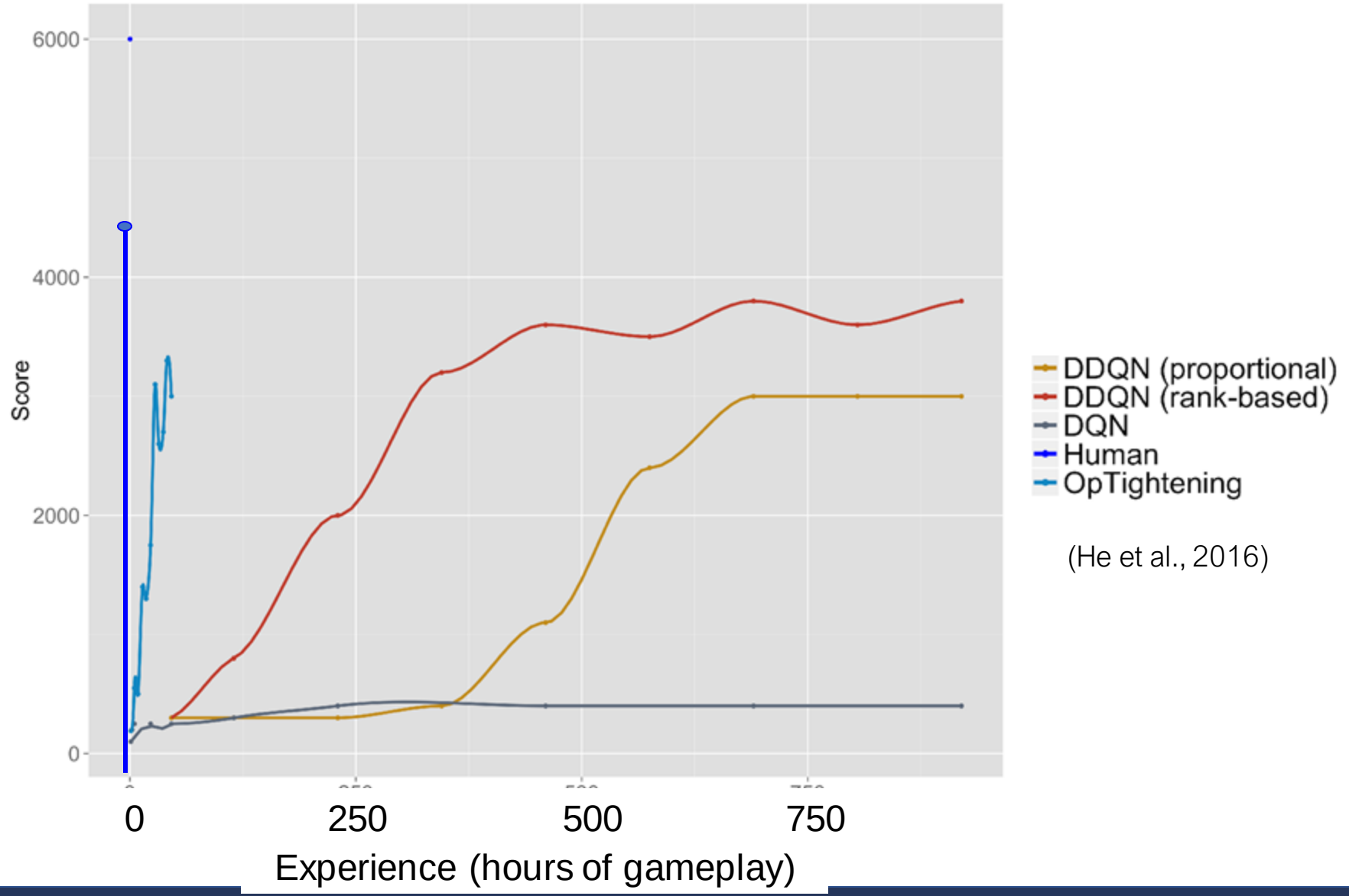
Frostbite


6000
4000
2000
0
Score
0
250
500
750
Experience (hours of gameplay)
DDQN (proportional)
DQN
Human

Frostbite

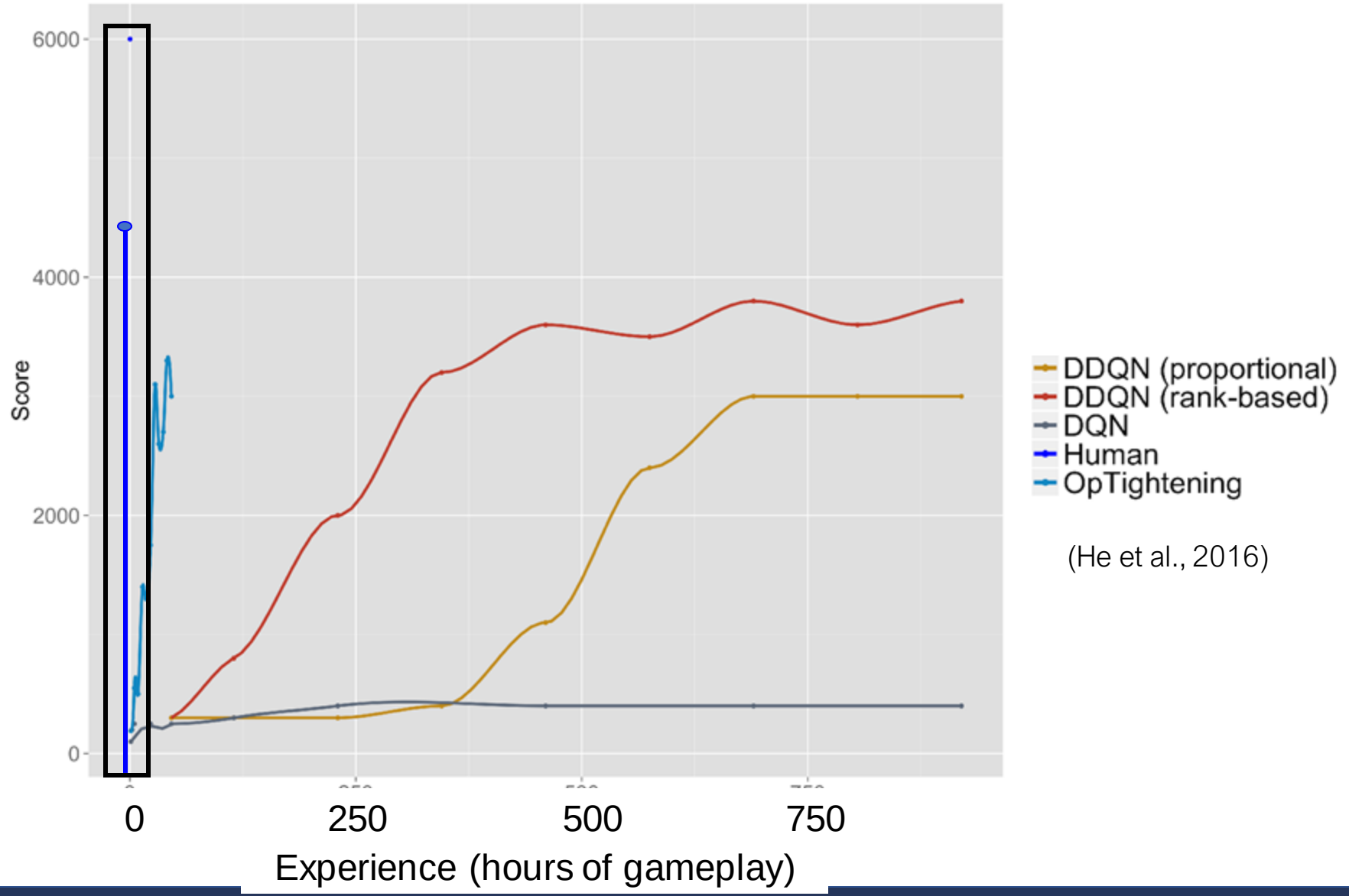

6000
4000
2000
0
Score
0
250
500
750
Experience (hours of gameplay)
DDQN (proportional)
DDQN (rank-based)
DQN
Human

Frostbite



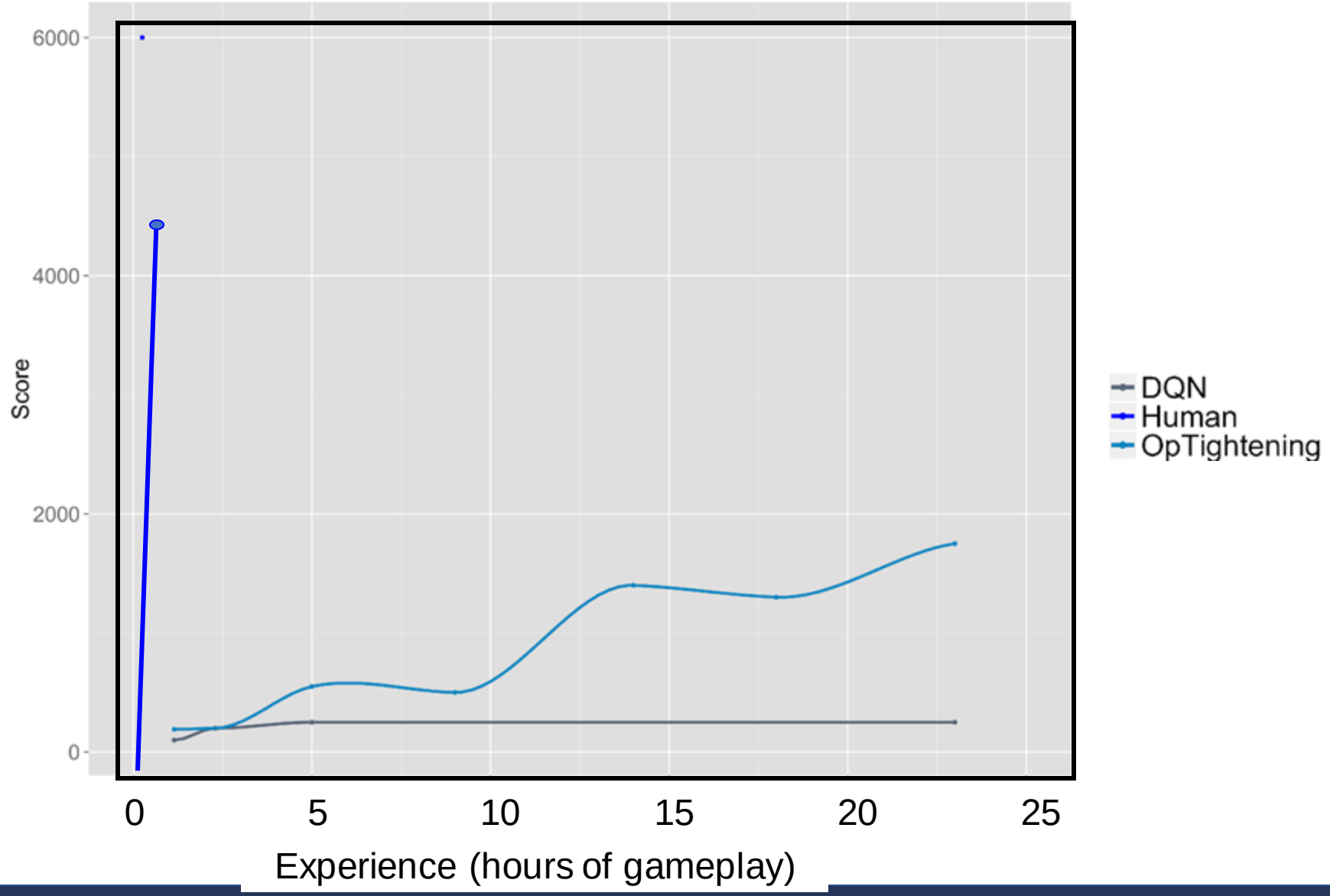
(He et al., 2016)

Frostbite



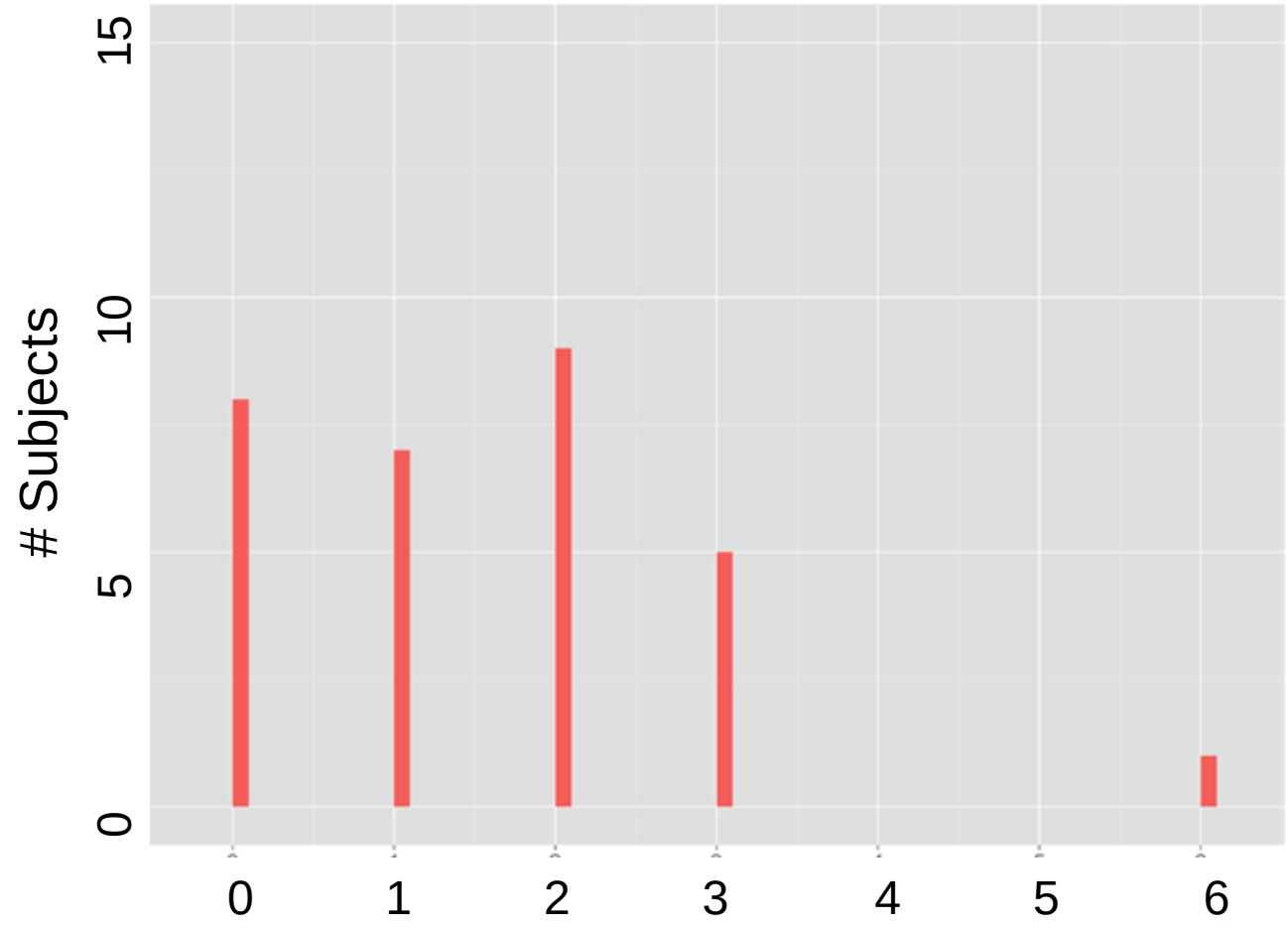
(He et al., 2016)

Frostbite


6000
4000
2000
0
Score
0
5
10
15
20
25
Experience (hours of gameplay)
DQN
Human
OpTightening

What drives such rapid learning?

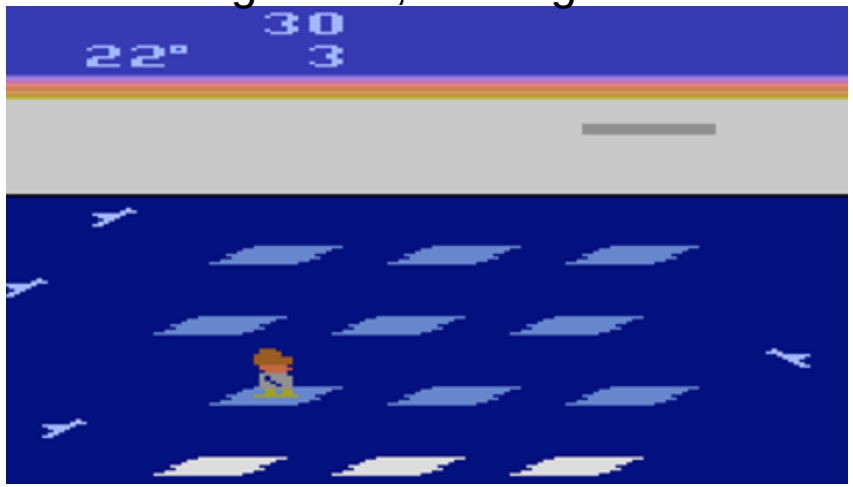
One-shot (or few-shot) learning about harmful actions and outcomes:



From the very beginning of play, people see objects, agents, physics. Actively explore possible object-relational goals, and soon come to multistep plans that exploit what they have learned.

A How to play Frostbite: Initial setup

B Visiting active, moving ice flows

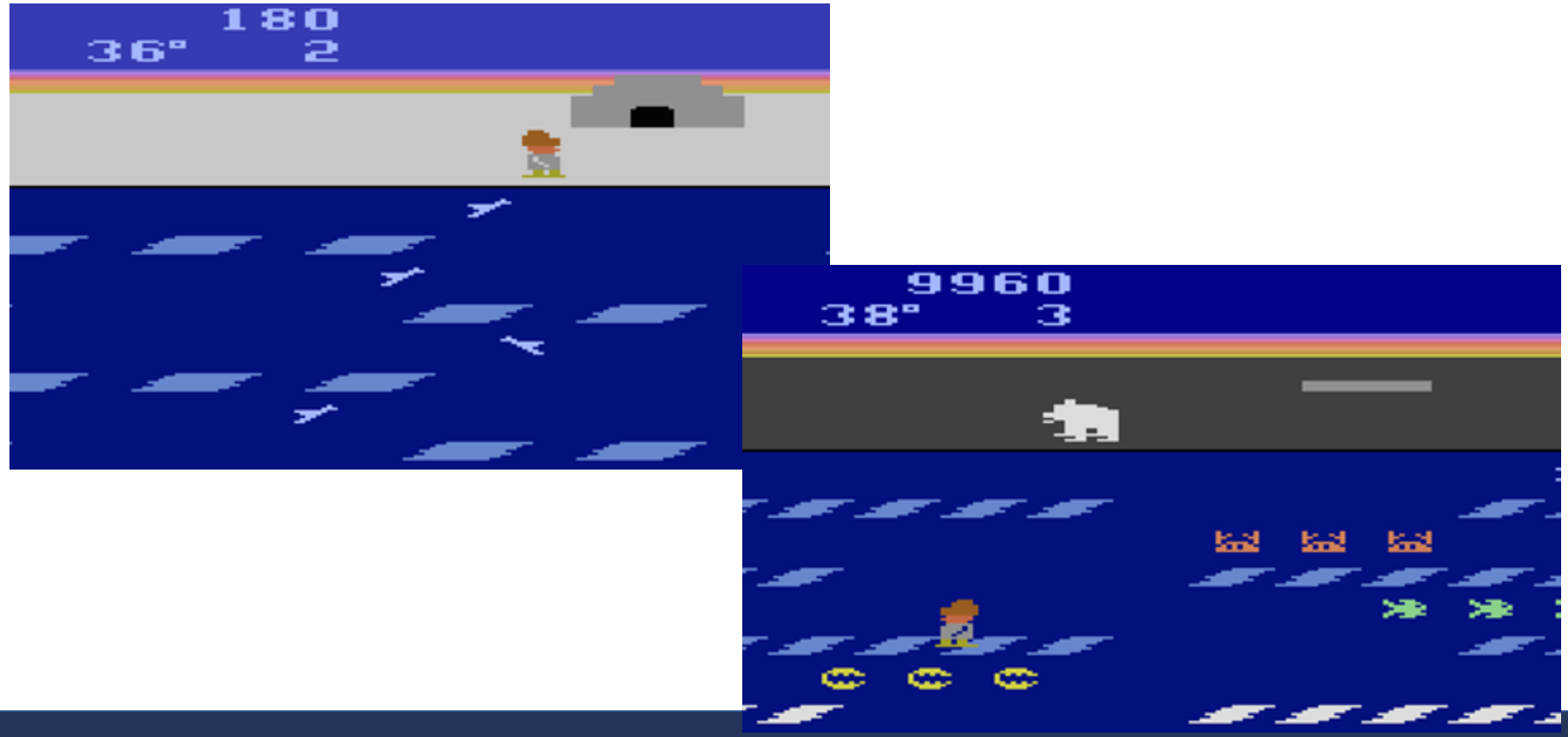


C Building the igloo

D Obstacles on later levels

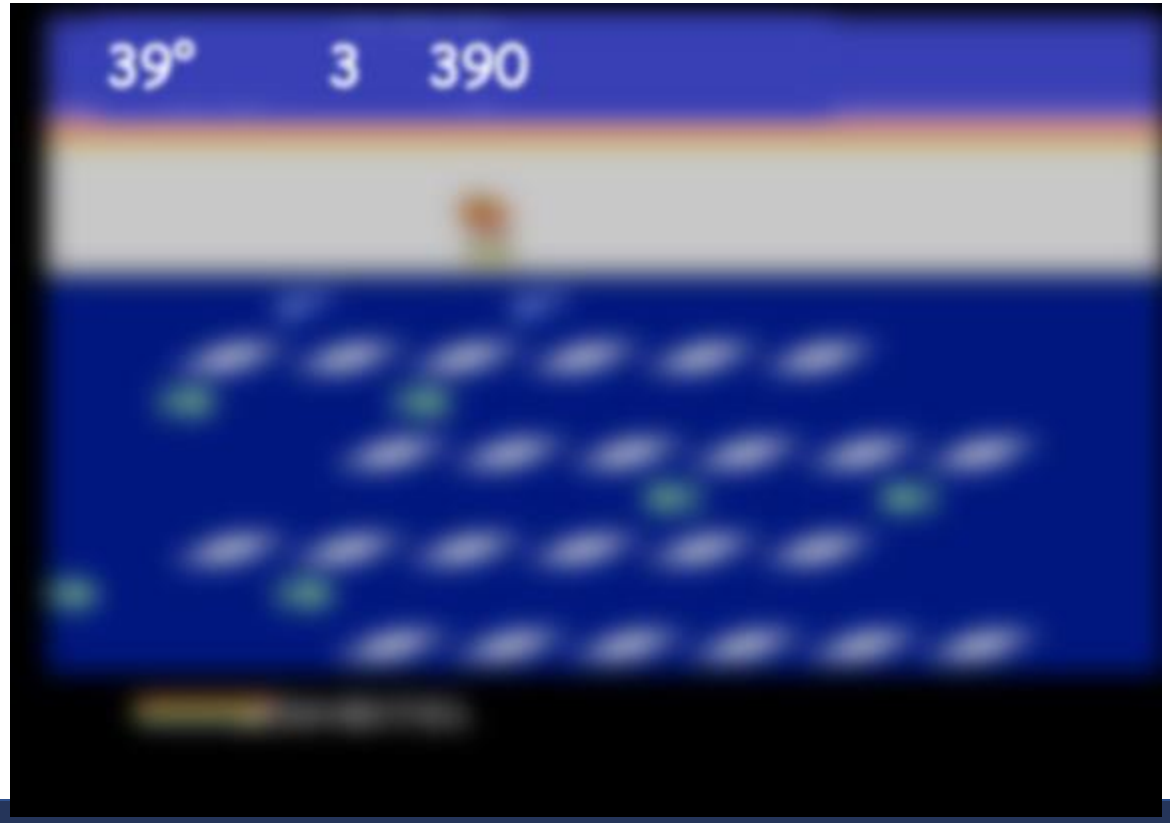
What drives such rapid learning?

To what extent is rapid learning dependent on prior knowledge about real-world objects, actions, and consequences?



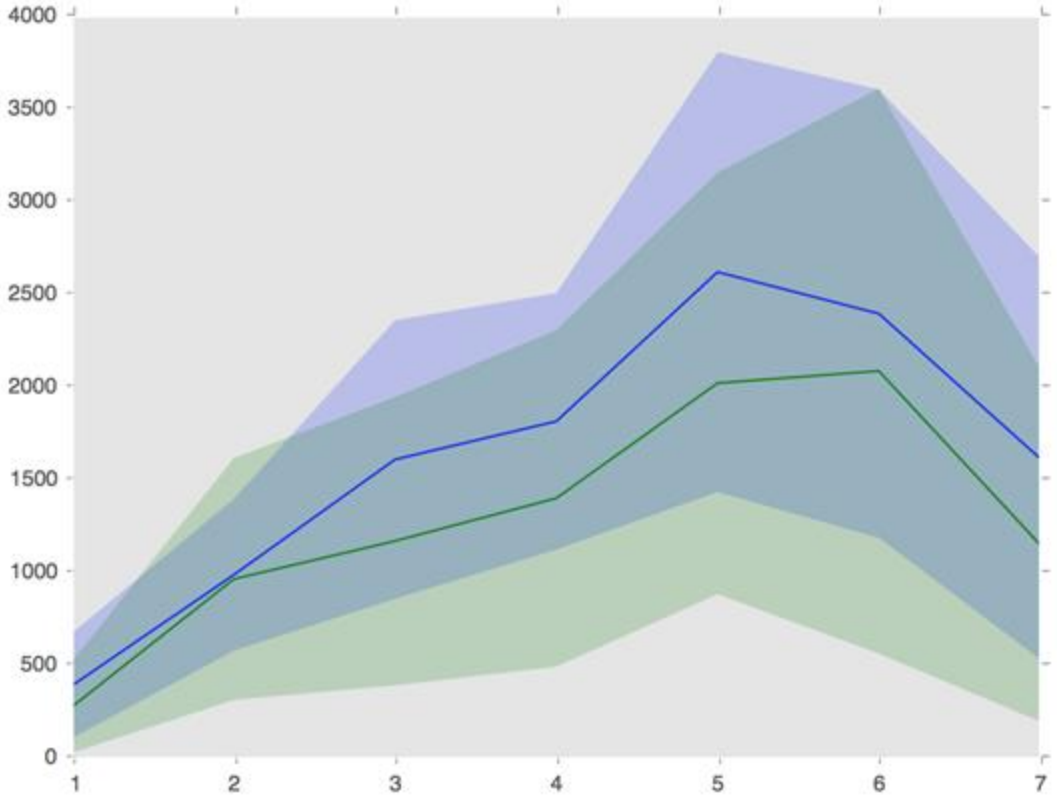
What drives such rapid learning?

To what extent is rapid learning dependent on prior knowledge about **real-world** objects, actions, and consequences?



What drives such rapid learning?

To what extent is rapid learning dependent on prior knowledge about real-world objects, actions, and consequences?



Blurred screen

Normal

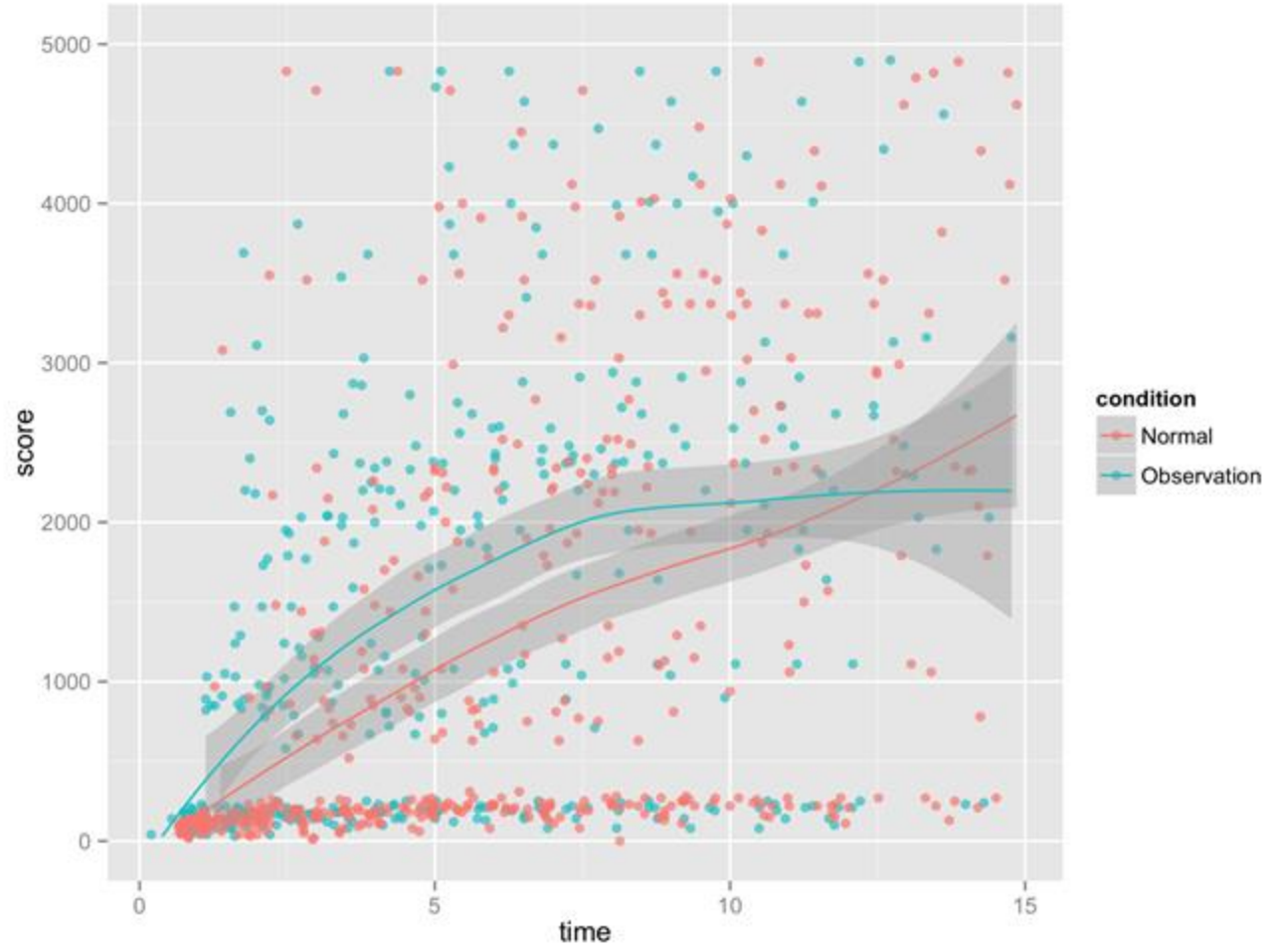
Being "object-oriented" in exploration matters, but prior world knowledge about specific object types doesn't so much!

What Drives Such Rapid Learning?

- Learning from demonstration & observation
- Popular idea in robotics
- Because of people!

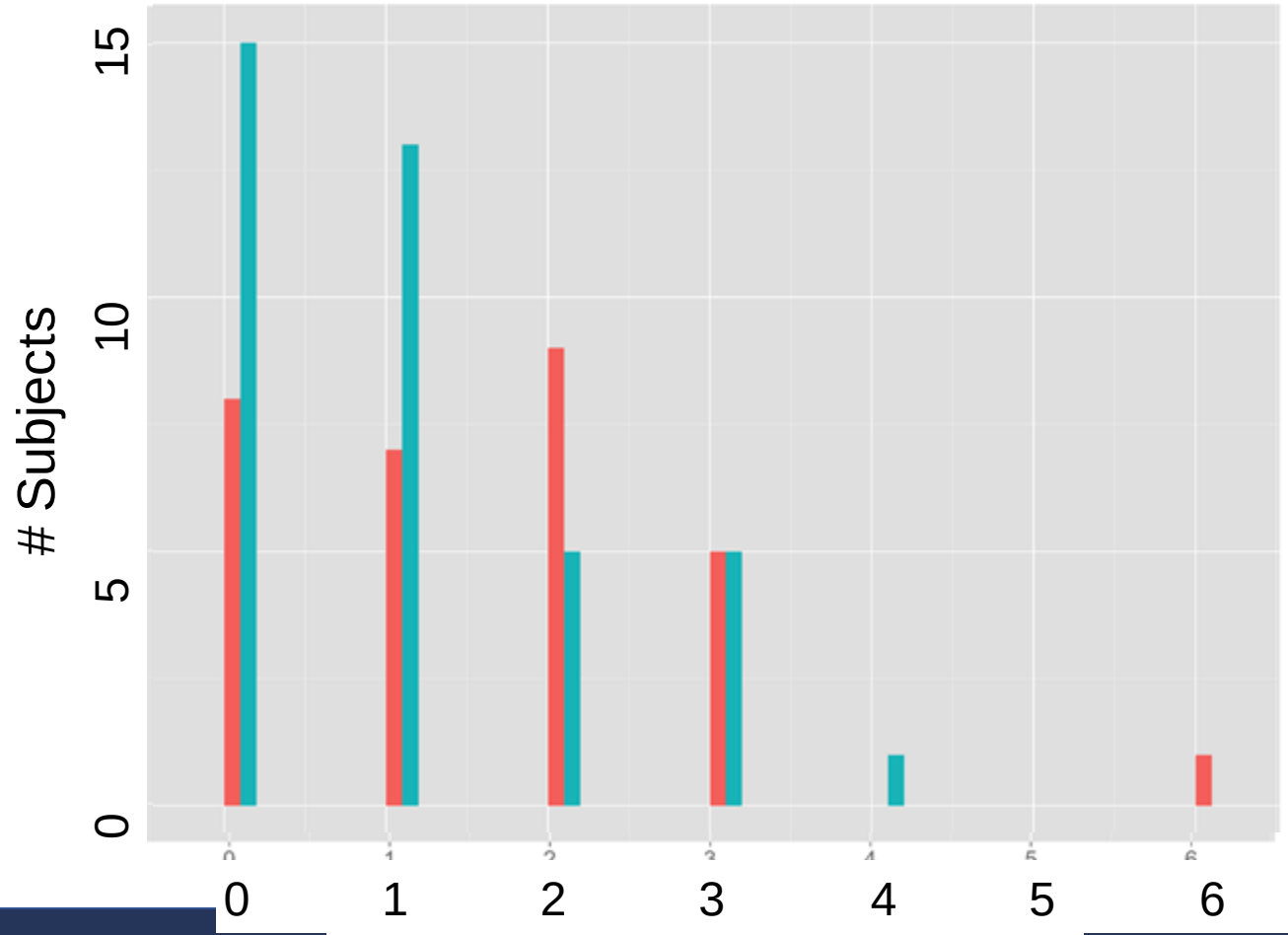
What drives such rapid learning?

People can learn even faster if they combine their own experience with just a little observation of others



What drives such rapid learning?

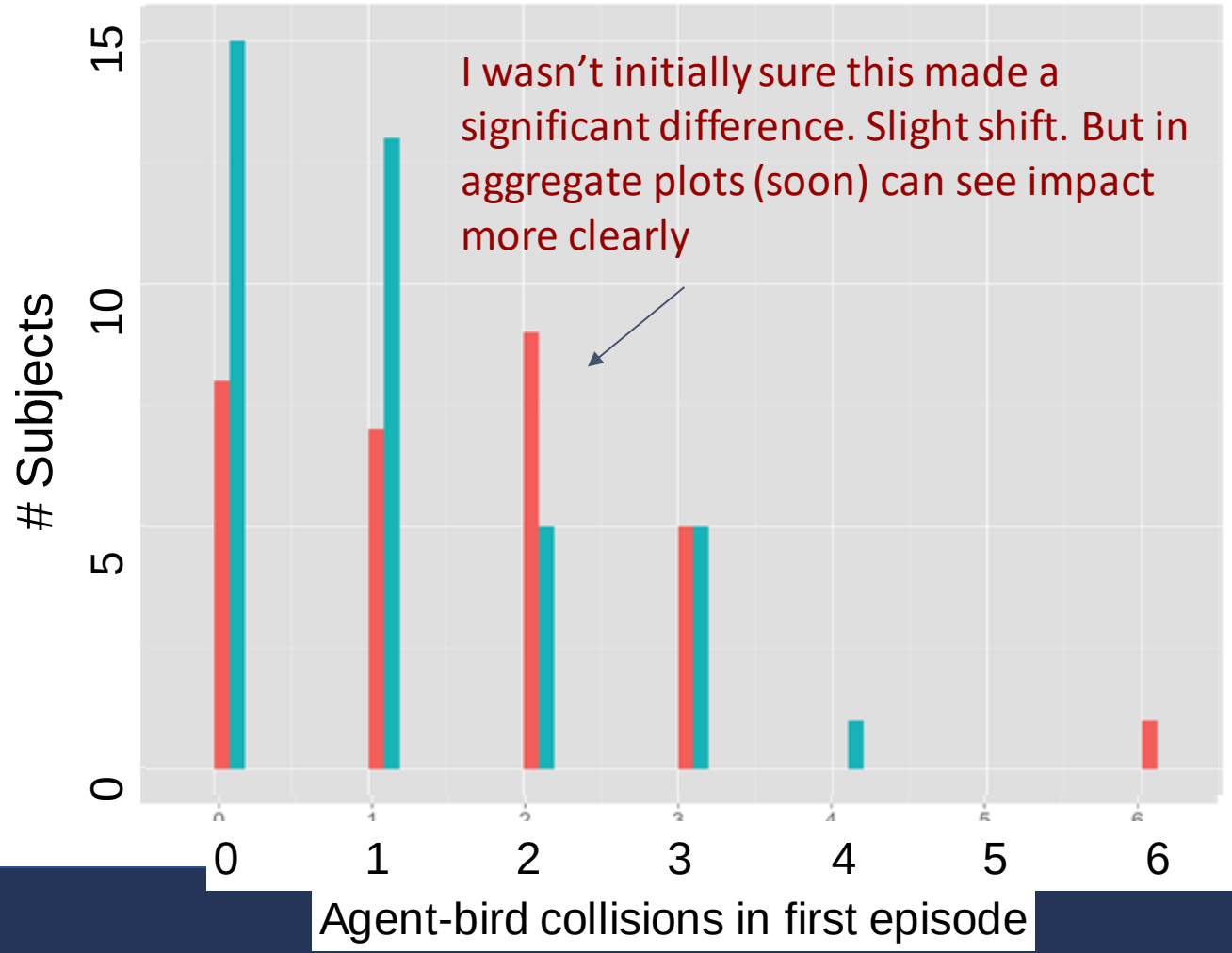
People can learn even faster if they combine their own experience with just a little help from others:



From one-shot learning to "zero-shot learning"

What drives such rapid learning?

People can learn even faster if they combine their own experience with just a little help from others:



From one-shot learning to "zero-shot learning"

What Drives Such Rapid Learning? Can We Support It?

- Hypothesis:
  - People are creating models of the world
  - Using these to plan behaviors
- If hypothesis is true
  - Speeding their learning of those models should improve performance
  - Therefore provide people with instruction manual
- Intervention
  - Had subjects read manual
  - Answered questionnaire about knowledge to ensure understood rules
  - Played for 15 minutes

FROSTBITE BASICS

The object of the game is to help Frostbite Bailey build igloos by jumping on floating blocks of ice. Be careful to avoid these deadly hazards: killer clams, snow geese, Alaskan king crab, grizzly polar bears and the rapidly dropping temperature.

To move Frostbite Bailey up, down, left or right, use the arrow keys. To reverse the direction of the ice floe you are standing on, press the spacebar. But remember, each time you do, your igloo will lose a block, unless it is completely built.

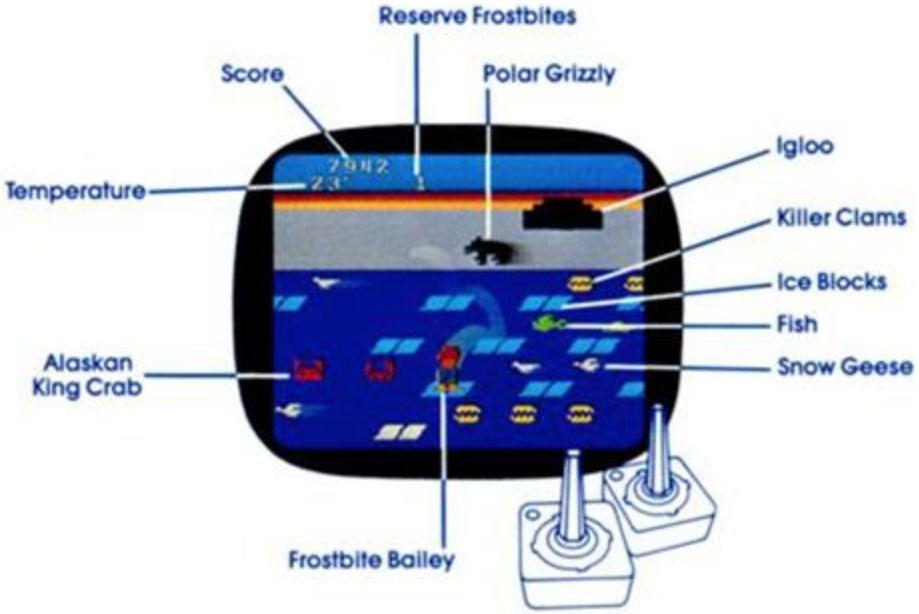
You begin the game with one active Frostbite Bailey and three on reserve. With each increase of 5,000 points, a bonus Frostbite is added to your reserves (up to a maximum of nine).

Frostbite gets lost each time he falls into the Arctic Sea, gets chased away by a Polar Grizzly or gets caught outside when the temperature drops to zero.

The game ends when your reserves have been exhausted and Frostbite is 'retired' from the construction business.

IGLOO CONSTRUCTION

Building codes. Each time Frostbite Bailey jumps onto a white ice floe, a "block" is added to the igloo. Once jumped upon, the white ice turns blue. It can still be jumped on, but won't add points to your score or blocks to your igloo. When all four rows are blue, they will turn white again. The igloo is complete when a door appears. Frostbite may then jump into it.



Work hazards. Avoid contact with Alaskan King Crabs, snow geese, and killer clams, as they will push Frostbite Bailey into the fatal Arctic Sea. The Polar Grizzlies come out of hibernation at level 4 and, upon contact, will chase Frostbite right off-screen.

No Overtime Allowed. Frostbite always starts working when it's 45 degrees outside. You'll notice this steadily falling temperature at the upper left corner of the screen. Frostbite must build and enter the igloo before the temperature drops to 0 degrees, or else he'll turn into blue ice!

SPECIAL FEATURES OF FROSTBITE

Fresh Fish swim by regularly. They are Frostbite Bailey's only food and, as such, are also additives to your score. Catch' em if you can.

FROSTBITE BASICS

Specifies reward structure

The object of the game is to help Frostbite Bailey build igloos by jumping on floating blocks of ice. Be careful to avoid these deadly hazards: killer clams, snow geese, Alaskan king crab, grizzly polar bears and the rapidly dropping temperature.

To move Frostbite Bailey up, down, left or right, use the arrow keys. To reverse the direction of the ice floe you are standing on, press the spacebar. But remember, each time you do, your igloo will lose a block, unless it is completely built.

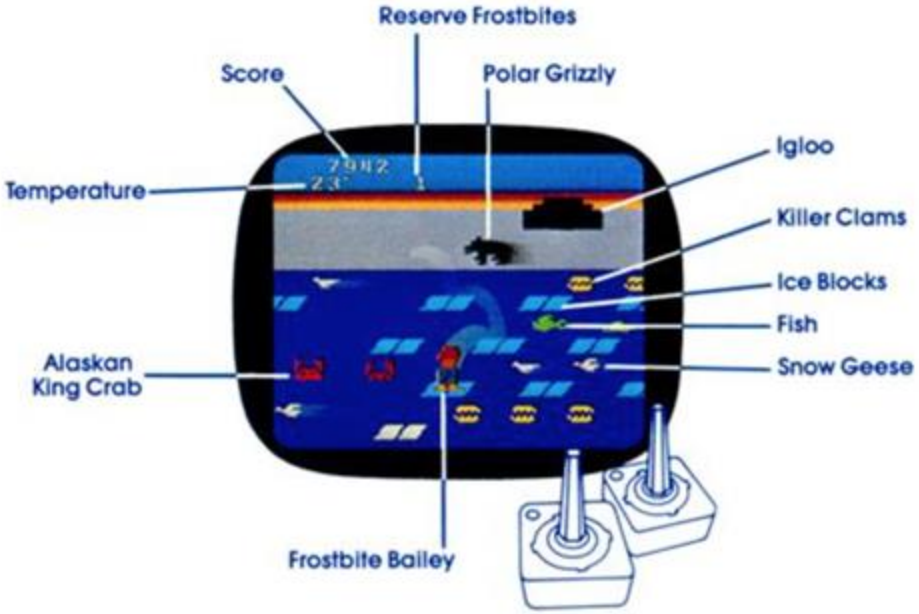
You begin the game with one active Frostbite Bailey and three on reserve. With each increase of 5,000 points, a bonus Frostbite is added to your reserves (up to a maximum of nine).

Frostbite gets lost each time he falls into the Arctic Sea, gets chased away by a Polar Grizzly or gets caught outside when the temperature drops to zero.

The game ends when your reserves have been exhausted and Frostbite is 'retired' from the construction business.

IGLOO CONSTRUCTION

Building codes. Each time Frostbite Bailey jumps onto a white ice floe, a "block" is added to the igloo. Once jumped upon, the white ice turns blue. It can still be jumped on, but won't add points to your score or blocks to your igloo. When all four rows are blue, they will turn white again. The igloo is complete when a door appears. Frostbite may then jump into it.



Work hazards. Avoid contact with Alaskan King Crabs, snow geese, and killer clams, as they will push Frostbite Bailey into the fatal Arctic Sea. The Polar Grizzlies come out of hibernation at level 4 and, upon contact, will chase Frostbite right off-screen.

No Overtime Allowed. Frostbite always starts working when it's 45 degrees outside. You'll notice this steadily falling temperature at the upper left corner of the screen. Frostbite must build and enter the igloo before the temperature drops to 0 degrees, or else he'll turn into blue ice!

SPECIAL FEATURES OF FROSTBITE

Fresh Fish swim by regularly. They are Frostbite Bailey's only food and, as such, are also additives to your score. Catch' em if you can.

FROSTBITE BASICS

The object of the game is to help Frostbite Bailey build igloos by jumping on floating blocks of ice. Be careful to avoid these deadly hazards: killer clams, snow geese, Alaskan king crab, grizzly polar bears and the rapidly dropping temperature.

To move Frostbite Bailey up, down, left or right, use the arrow keys. To reverse the direction of the ice floe you are standing on, press the spacebar. But remember, each time you do, your igloo will lose a block, unless it is completely built.

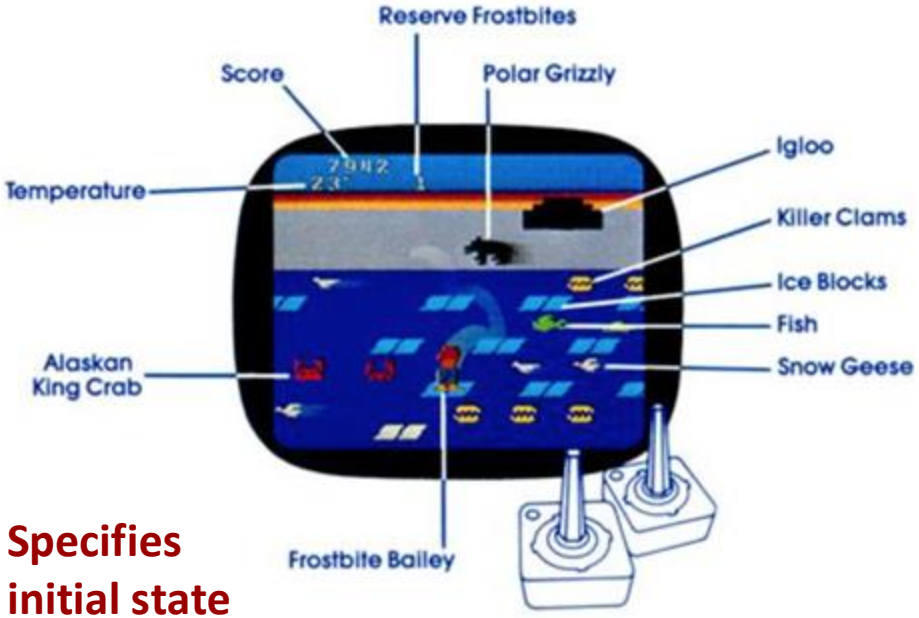
You begin the game with one active Frostbite Bailey and three on reserve. With each increase of 5,000 points, a bonus Frostbite is added to your reserves (up to a maximum of nine).

Frostbite gets lost each time he falls into the Arctic Sea, gets chased away by a Polar Grizzly or gets caught outside when the temperature drops to zero.

The game ends when your reserves have been exhausted and Frostbite is 'retired' from the construction business.

IGLOO CONSTRUCTION

Building codes. Each time Frostbite Bailey jumps onto a white ice floe, a "block" is added to the igloo. Once jumped upon, the white ice turns blue. It can still be jumped on, but won't add points to your score or blocks to your igloo. When all four rows are blue, they will turn white again. The igloo is complete when a door appears. Frostbite may then jump into it.



Specifies initial state

Work hazards. Avoid contact with Alaskan King Crabs, snow geese, and killer clams, as they will push Frostbite Bailey into the fatal Arctic Sea. The Polar Grizzlies come out of hibernation at level 4 and, upon contact, will chase Frostbite right off-screen.

No Overtime Allowed. Frostbite always starts working when it's 45 degrees outside. You'll notice this steadily falling temperature at the upper left corner of the screen. Frostbite must build and enter the igloo before the temperature drops to 0 degrees, or else he'll turn into blue ice!

SPECIAL FEATURES OF FROSTBITE

Fresh Fish swim by regularly. They are Frostbite Bailey's only food and, as such, are also additives to your score. Catch' em if you can.

FROSTBITE BASICS

The object of the game is to help Frostbite Bailey build igloos by jumping on floating blocks of ice. Be careful to avoid these deadly hazards: killer clams, snow geese, Alaskan king crab, grizzly polar bears and the rapidly dropping temperature.

To move Frostbite Bailey up, down, left or right, use the arrow keys. To reverse the direction of the ice floe you are standing on, press the spacebar. But remember, each time you do, your igloo will lose a block, unless it is completely built.

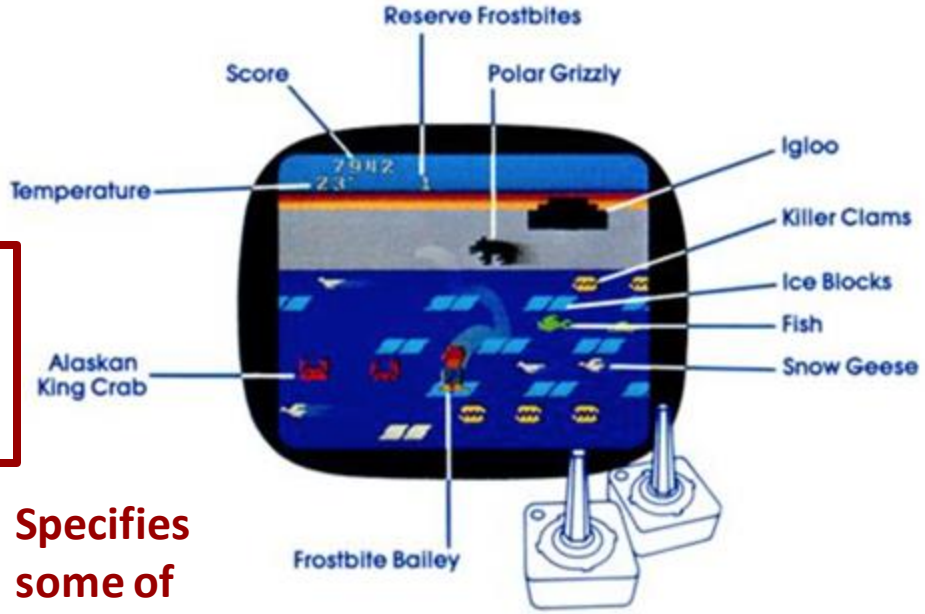
You begin the game with one active Frostbite Bailey and three on reserve. With each increase of 5,000 points, a bonus Frostbite is added to your reserves (up to a maximum of nine).

Frostbite gets lost each time he falls into the Arctic Sea, gets chased away by a Polar Grizzly or gets caught outside when the temperature drops to zero.

The game ends when your reserves have been exhausted and Frostbite is 'retired' from the construction business.

IGLOO CONSTRUCTION

Building codes. Each time Frostbite Bailey jumps onto a white ice floe, a "block" is added to the igloo. Once jumped upon, the white ice turns blue. It can still be jumped on, but won't add points to your score or blocks to your igloo. When all four rows are blue, they will turn white again. The igloo is complete when a door appears. Frostbite may then jump into it.



Specifies some of dynamics

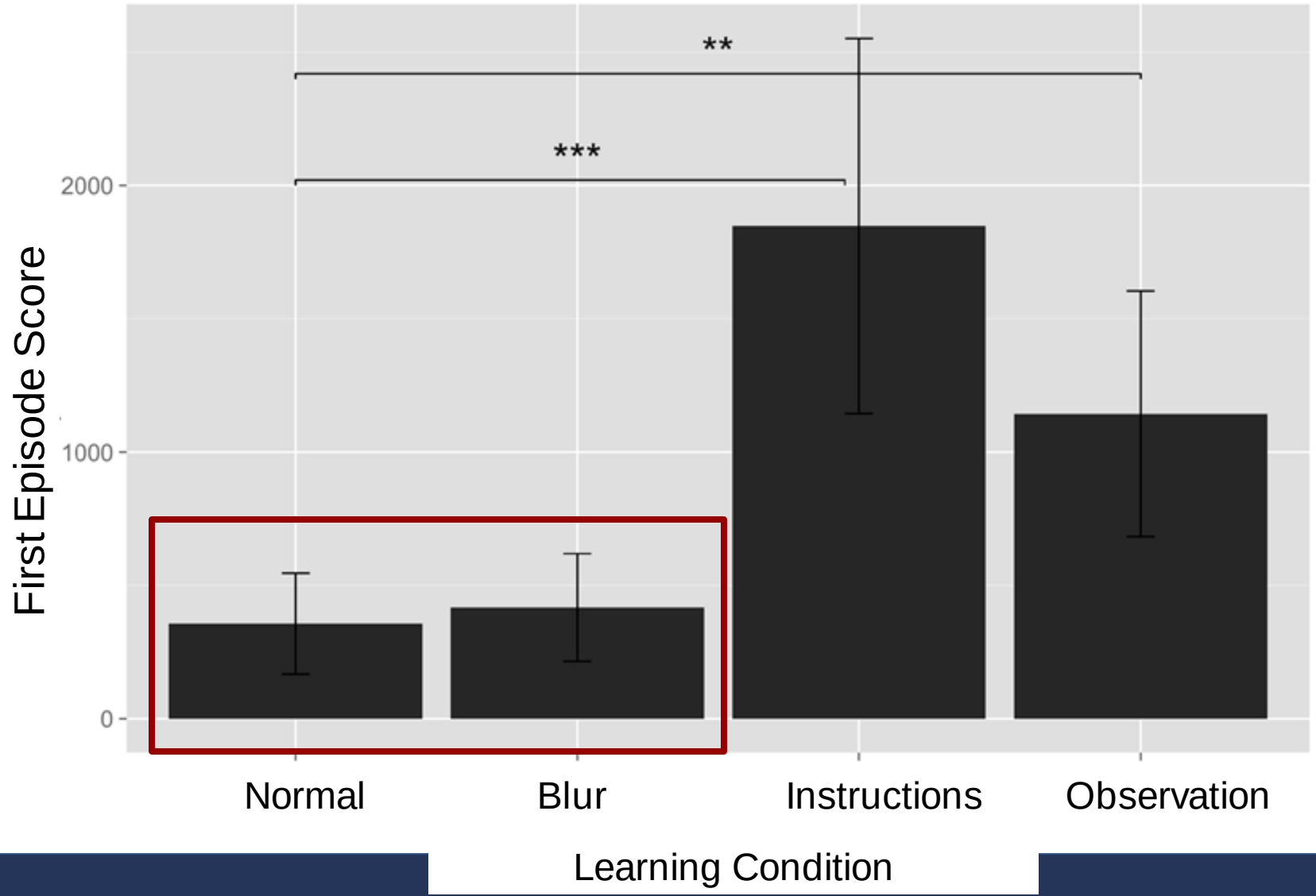
Work hazards. Avoid contact with Alaskan King Crabs, snow geese, and killer clams, as they will push Frostbite Bailey into the fatal Arctic Sea. The Polar Grizzlies come out of hibernation at level 4 and, upon contact, will chase Frostbite right off-screen.

No Overtime Allowed. Frostbite always starts working when it's 45 degrees outside. You'll notice this steadily falling temperature at the upper left corner of the screen. Frostbite must build and enter the igloo before the temperature drops to 0 degrees, or else he'll turn into blue ice!

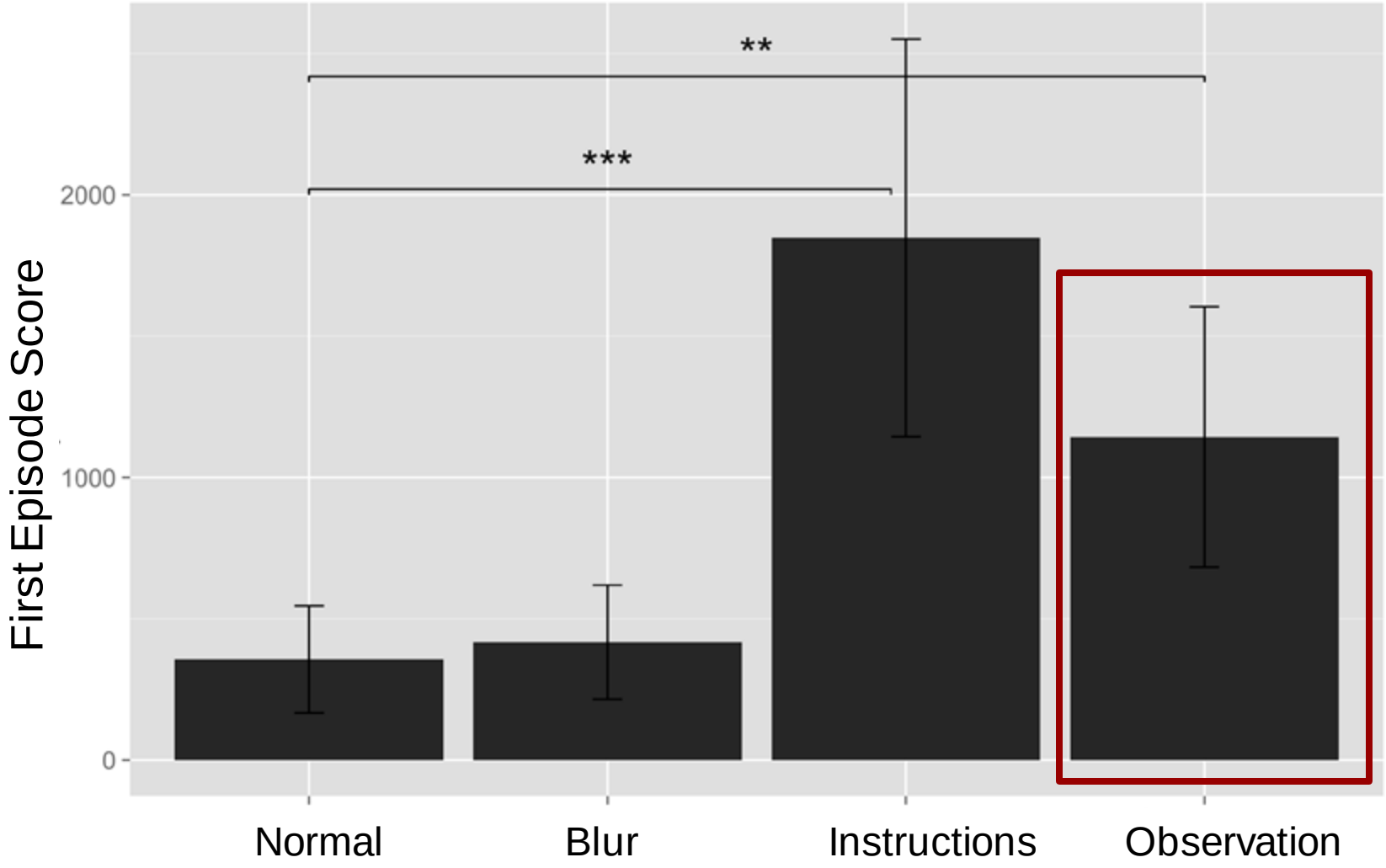
SPECIAL FEATURES OF FROSTBITE

Fresh Fish swim by regularly. They are Frostbite Bailey's only food and, as such, are also additives to your score. Catch' em if you can.

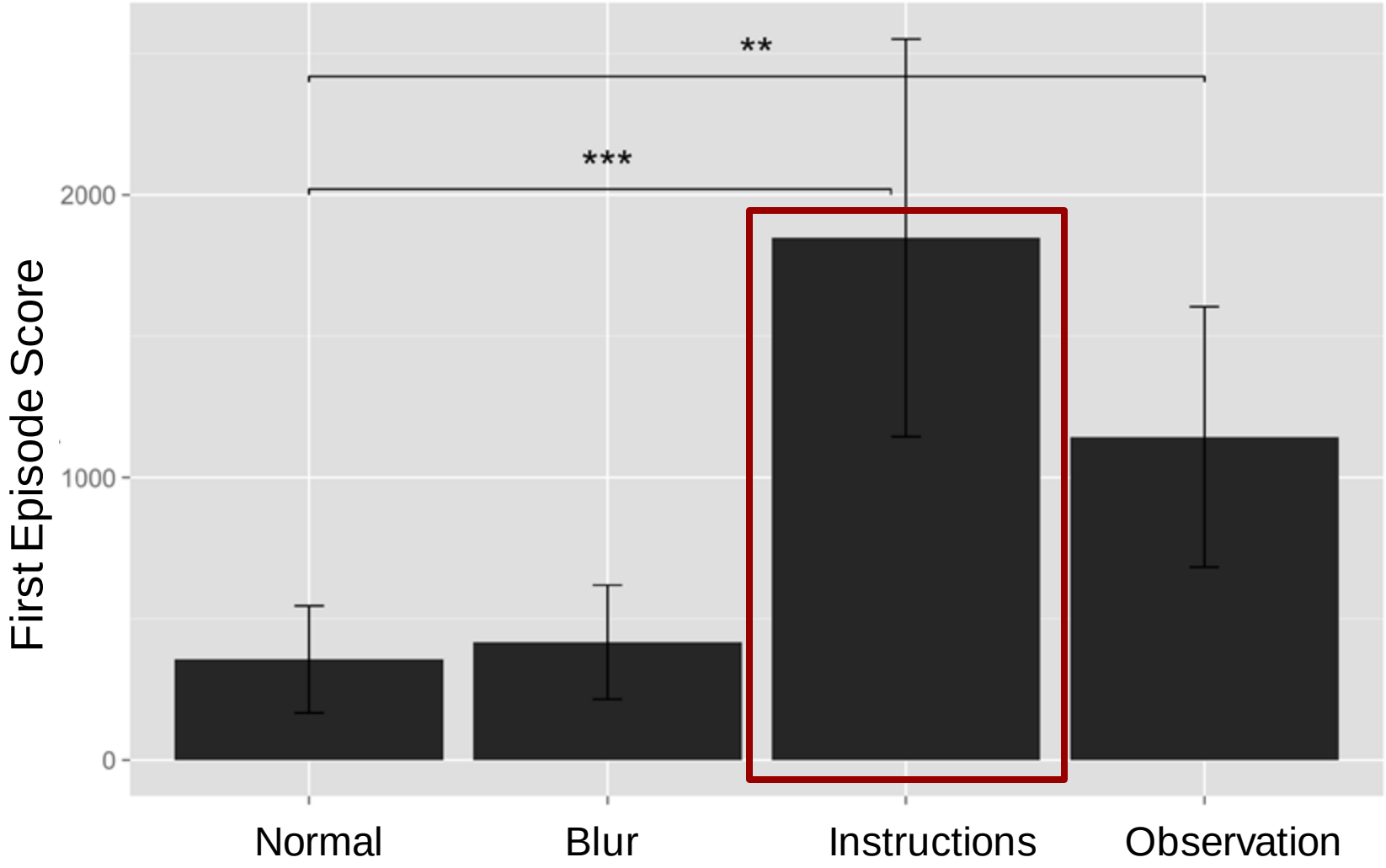
Humans aren’t relying on specific object knowledge



Watching Someone Else Who has Some Experience Significantly Improves Initial performance



Giving Information about the Dynamics & Reward Significantly Improves Initial Performance



Discussion of results

>=1 slide

What conclusions are drawn from the results?

Are the stated conclusions fully supported by the results and references? If so, why? (Recap the relevant supporting evidences from the given results + refs)

Discussion

- People learn and improve in several Atari tasks much faster than Deep RL
- Does not seem to be due to specific object prior information
  - E.g. about how birds fly
- But do seem to take advantage of relational / object oriented information about the dynamics and the reward
- People be building and testing models and theories using higher level representations

Critique / Limitations / Open Issues

1 or more slides: What are the key limitations of the proposed approach / ideas? (e.g. does it require strong assumptions that are unlikely to be practical? Computationally expensive? Require a lot of data? Find only local optima?)

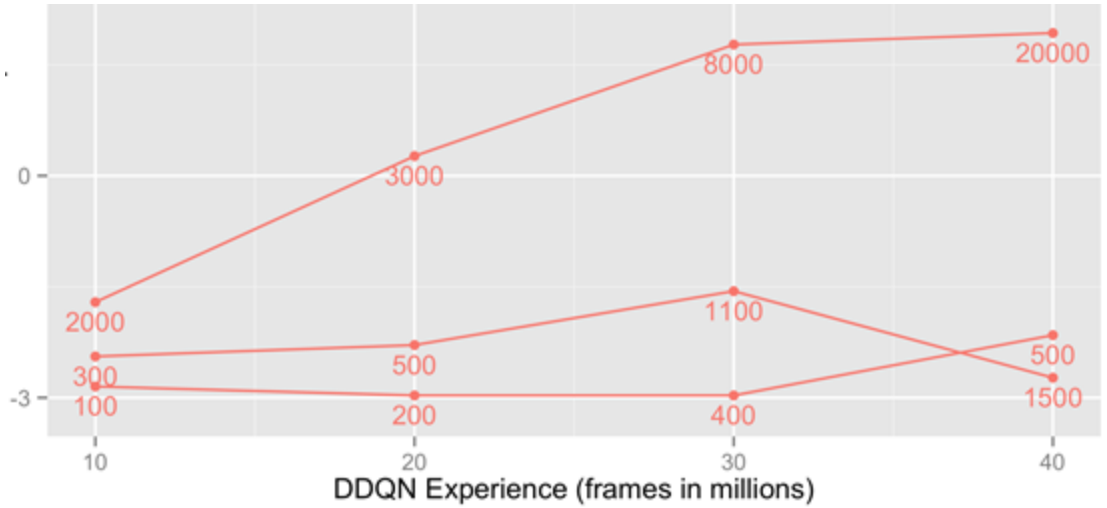
- If follow up work has addressed some of these limitations, include pointers to that. But don’t limit your discussion only to the problems / limitations that have already been addressed.

Critique / Limitations / Open Issues

- Teaching was better than observation
- Is this because people had to infer optimal policy?
- If we wrote down optimal policy (as a set of rules) and gave it to people
  - Would that be more effective than observation?
  - Would it be better than instruction?
- Broader question:
  - Is building a model better than policy search?
  - Is it that people can't do policy search in their head as well as build a model?
  - But machines don't have that constraint...

Critique / Limitations / Open Issues

- Many tasks require more than 15 minutes
- How do humans learn in these tasks? What is the rate of progress?
- DDQN improved its **rate** of learning over time
- Didn't see that with people in these tasks
- Why and when does this happen?



Contributions (Recap)

Approximately one bullet for each of the following (the paper on 1 slide)

- Problem the reading is discussing
- Why is it important and hard
- What is the key limitation of prior work
- What is the key insight(s) (try to do in 1-3) of the proposed work
- What did they demonstrate by this insight? (tighter theoretical bounds, state of the art performance on X, etc)

Contributions (Recap)

- **Problem:** Want to understand how people play Atari
- **Why is this problem important?**
  - Because Atari games seem like a good involve tasks with widely different visual aspects, dynamics and goals presented
  - Lots of success of deep RL agents but require a lot of training
  - Do people do this too? If not, what might we learn from them?
- **Why is that problem hard?** Much unknown about human learning
- **Limitations of prior work**: Little work on human atari performance
- **Key insight/approach**: Measure people's performance. Test idea that people are building models of object/relational structure
- **Revealed:** People learning much faster than Deep RL. Interventions suggest people can benefit from high level structure of domain models and use to speed learning.

Agenda

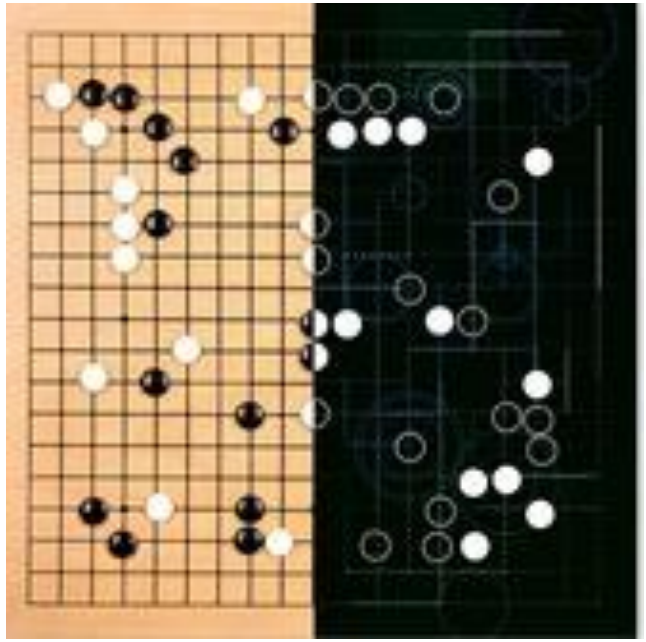
- Logistics
- Course Motivation
- Primer in RL
- Human learning and RL (sample paper presentation)
- Presentation Sign-ups

RL in Recent Memory

Atari

DQN (Mnih et al. 2013)
DAGGER (Guo et al, 2014)
Policy Gradients (Schulman et al 2015)
DDPG (Lillicrap et al. 2015)
A3C (Mnih et al. 2016)

Go

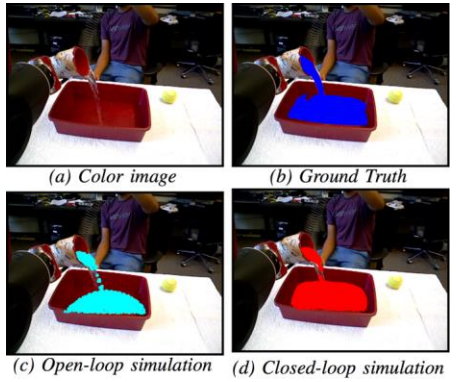


Policy Gradients
+
Monte Carlo Tree Search
(Silver et al. 2016)
…

Robotics

Levine et al. (2015)
Krishnan, G. et al (2016)
Rusu et al (2016)
Bojarski et al. (2016) nVidia
…

Success Stories for Learning in Robotics



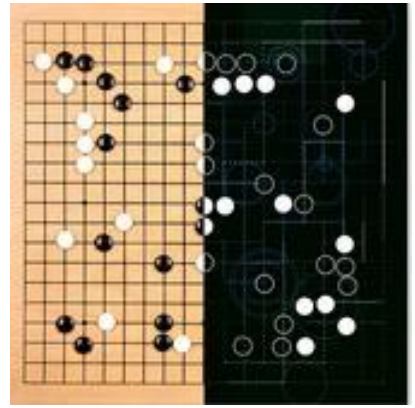
Mason & Salisbury 1985
Srinivasa et al 2010
Berenson 2013
Odhner1 et al 2014
Chavan-Dafle et al 2014
Yamaguchi, et. al, 2015
…

Li , Allen et al. 2015
Yahya et al, 2016
Schenck et al. 2017
Mar et al. 2017
Laskey et al 2017
Quispe et al 2018
…

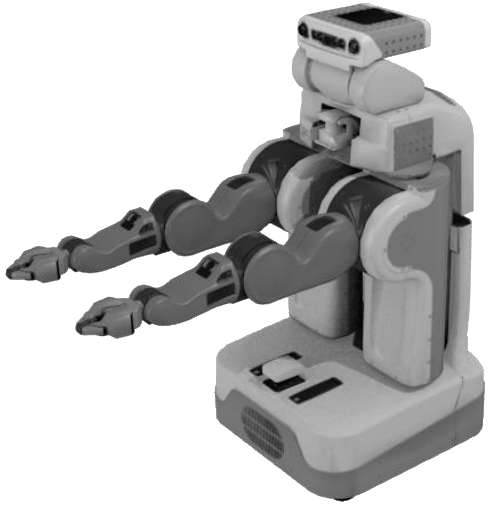
Mishra et al 1987
Ferrari & Canny, 1992
Ciocarlie & Allen, 2009
Dogar & Srinivasa, 2011
Rodriguez et al. 2012
Bohg et al 2014

Pinto & Gupta, 2016
Levine et al 2016
Mahler et al 2017
Jang et al 2017
Viereck et al 2017
…

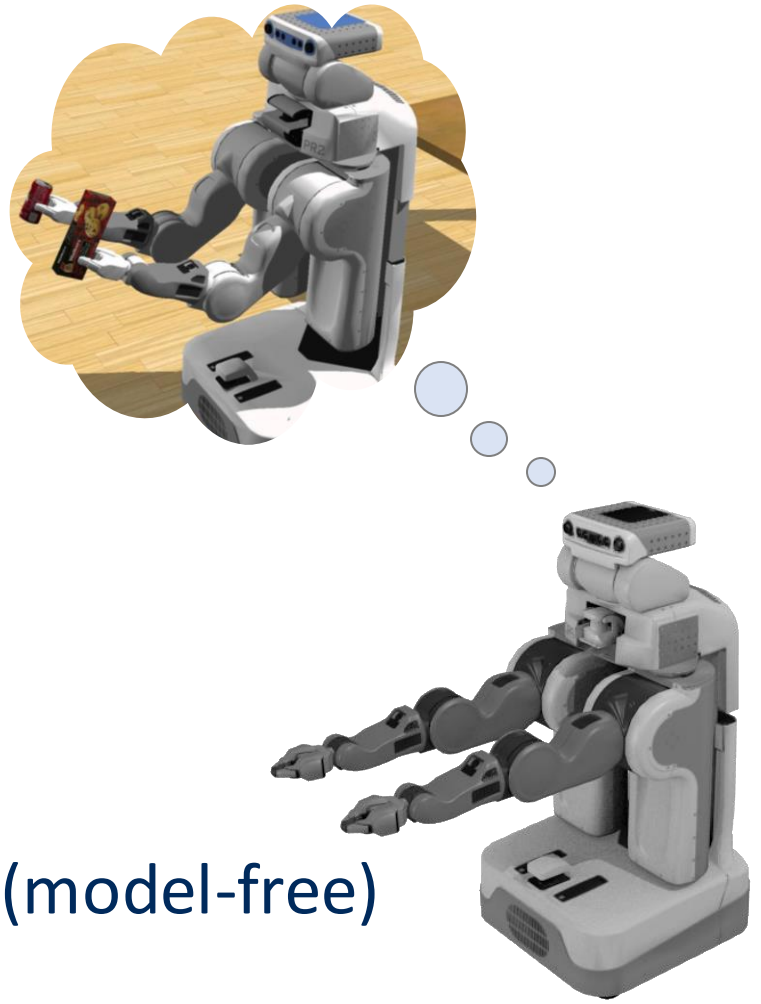
Going from Go to Robot/Control



- Known Environment vs Unstructured/Open World
- Need for Behavior Transfer
- Discrete vs Continuous States-Actions
- Single vs Variable Goals
- Reward Oracle vs Reward Inference



Other Open Problems



- Single algorithm for multiple tasks
- Learn new tasks very quickly
- Reuse past information about related problems
- Reward modelling in open environment
- How and what to build a model of?
- How much to rely on the model vs direct reflex (model-free)
- Learn without interaction if seen a lot of data

What this course plans to cover

- Imitation Learning: Supervised
- Policy Gradient Algorithms
- Actor-Critic Methods
- Value Based Methods
- Distributional RL
- Model-Based Methods
- Imitation Learning: Inverse RL
- Exploration Methods
- Bayesian RL
- Hierarchical RL

Let us help the Robots help us!

Animesh Garg

garg@cs.toronto.edu

@Animesh_Garg

Agenda

- Logistics
- Course Motivation
- Primer in RL
- Human learning and RL (sample paper presentation)
- Presentation Sign-ups