CS 8803
Deep Reinforcement Learning

Lec 1: Introduction & Logistics

Fall 2024

Animesh Garg

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

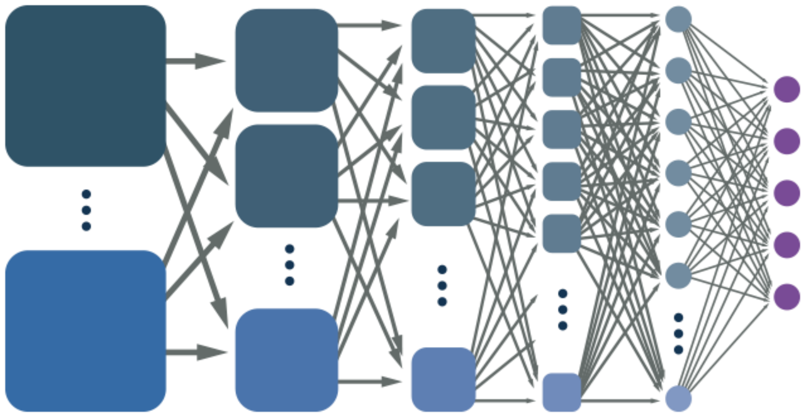
Motivation and Main Problem

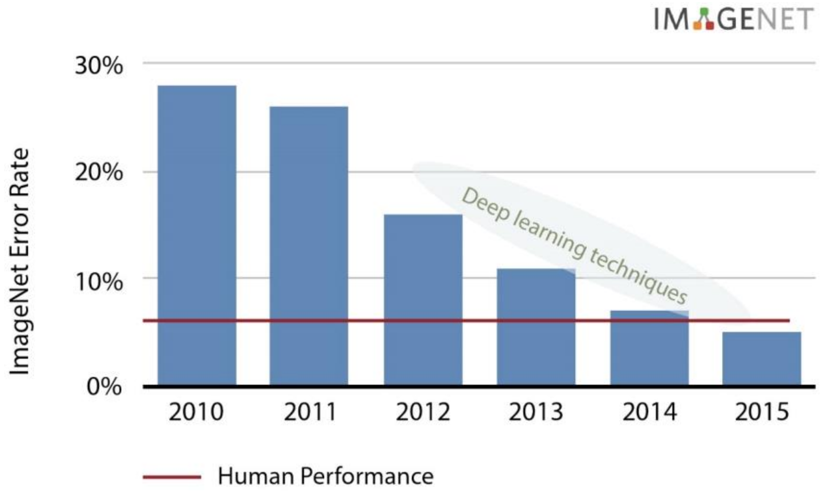
1-4 slides

Should capture

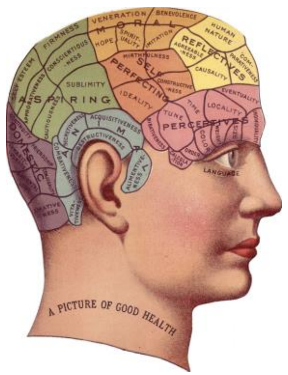
- High level description of problem being solved (can use videos, images, etc)
- Why is that problem important?
- Why is that problem hard?
- High level idea of why prior work didn’t already solve this (Short description, later will go into details)

A Seductive Hypothesis



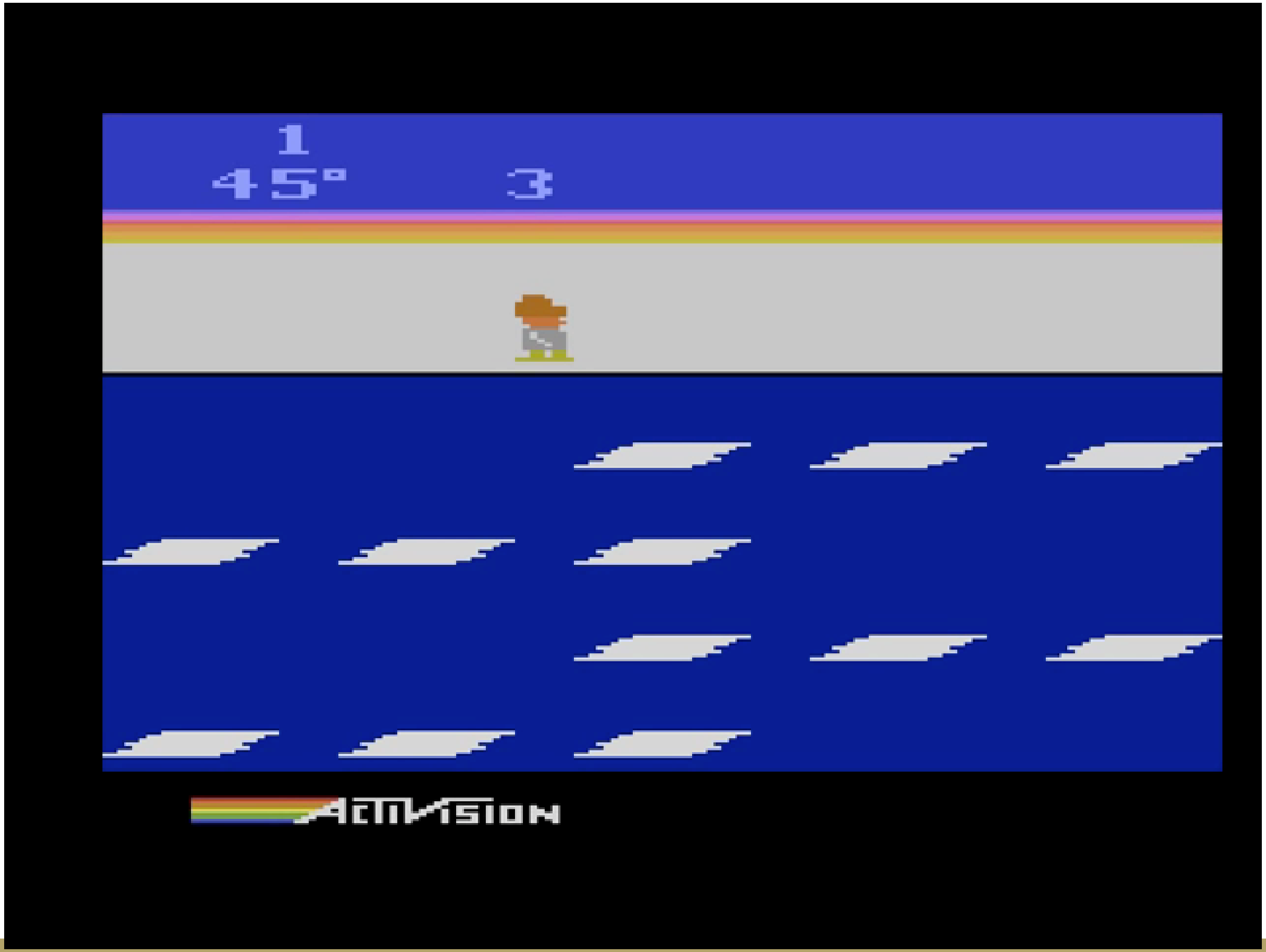


Brain-like computation + Human-level performance

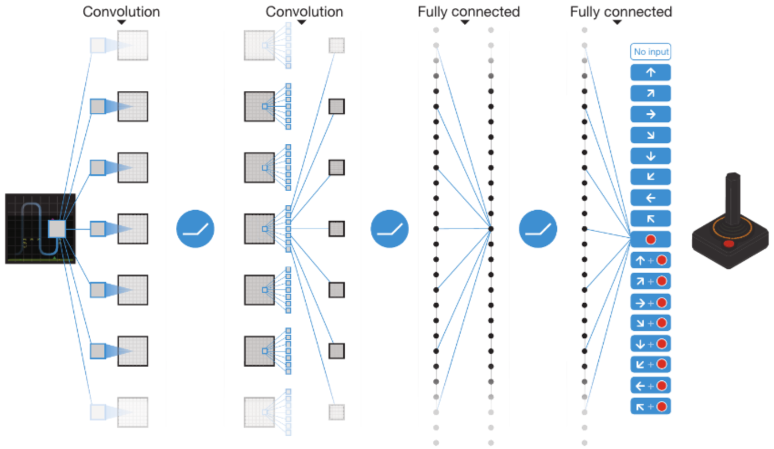


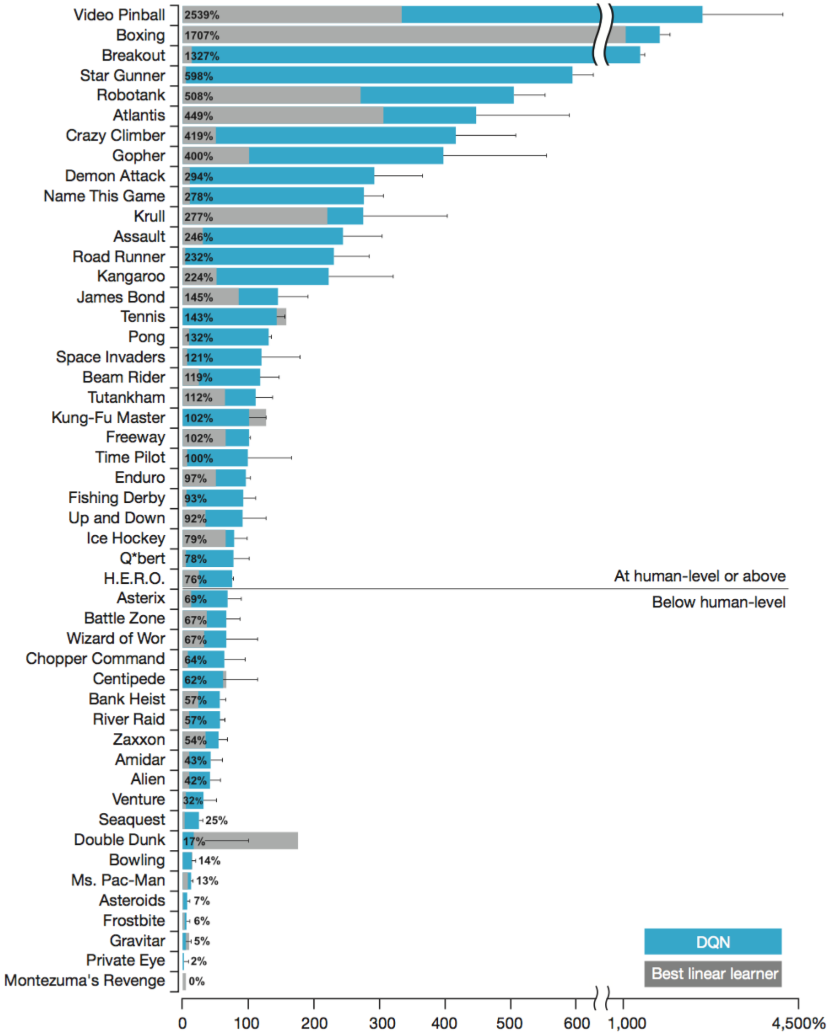
= Human intelligence?

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

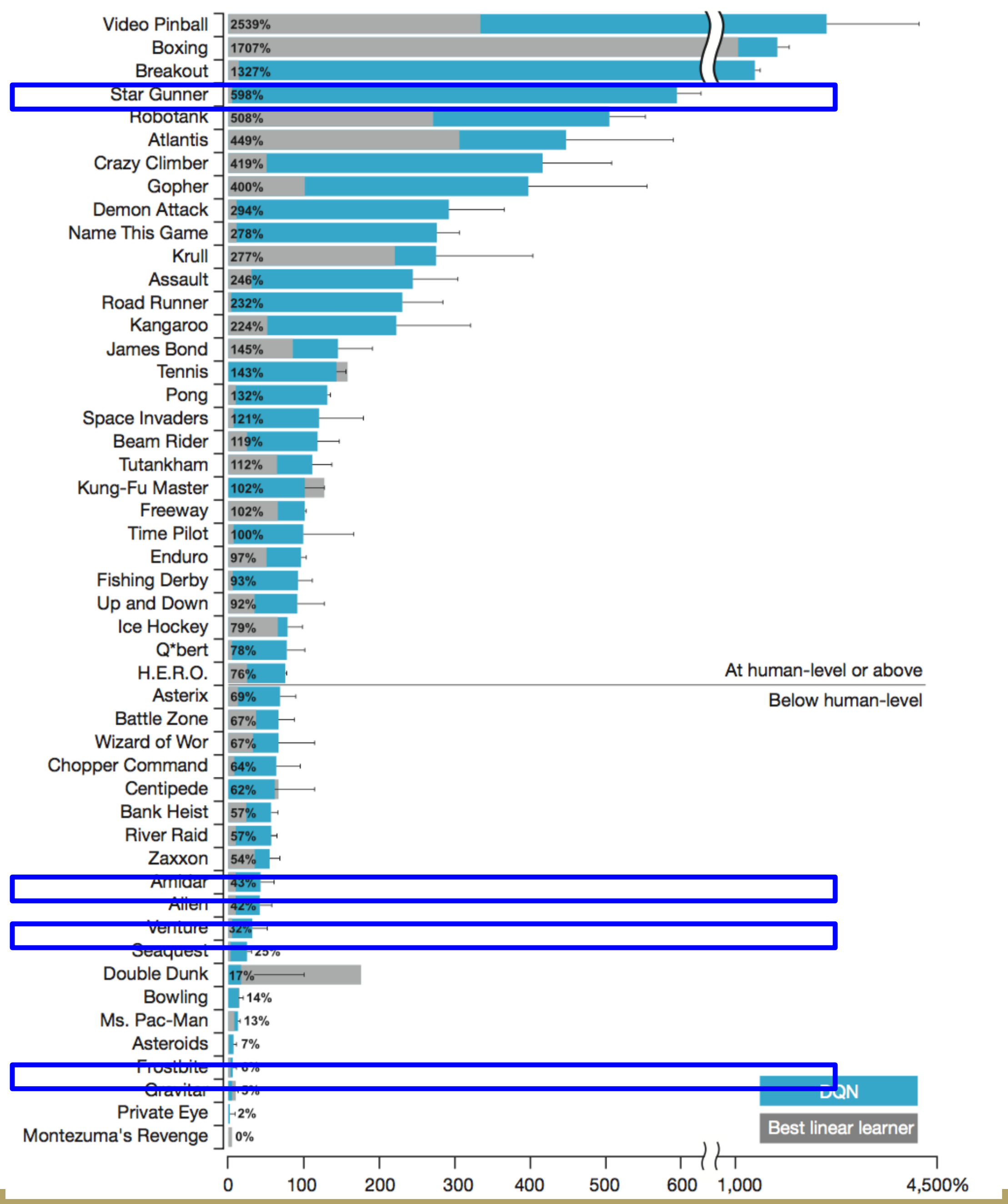
Describe algorithm or framework (pseudocode and flowcharts can help)

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

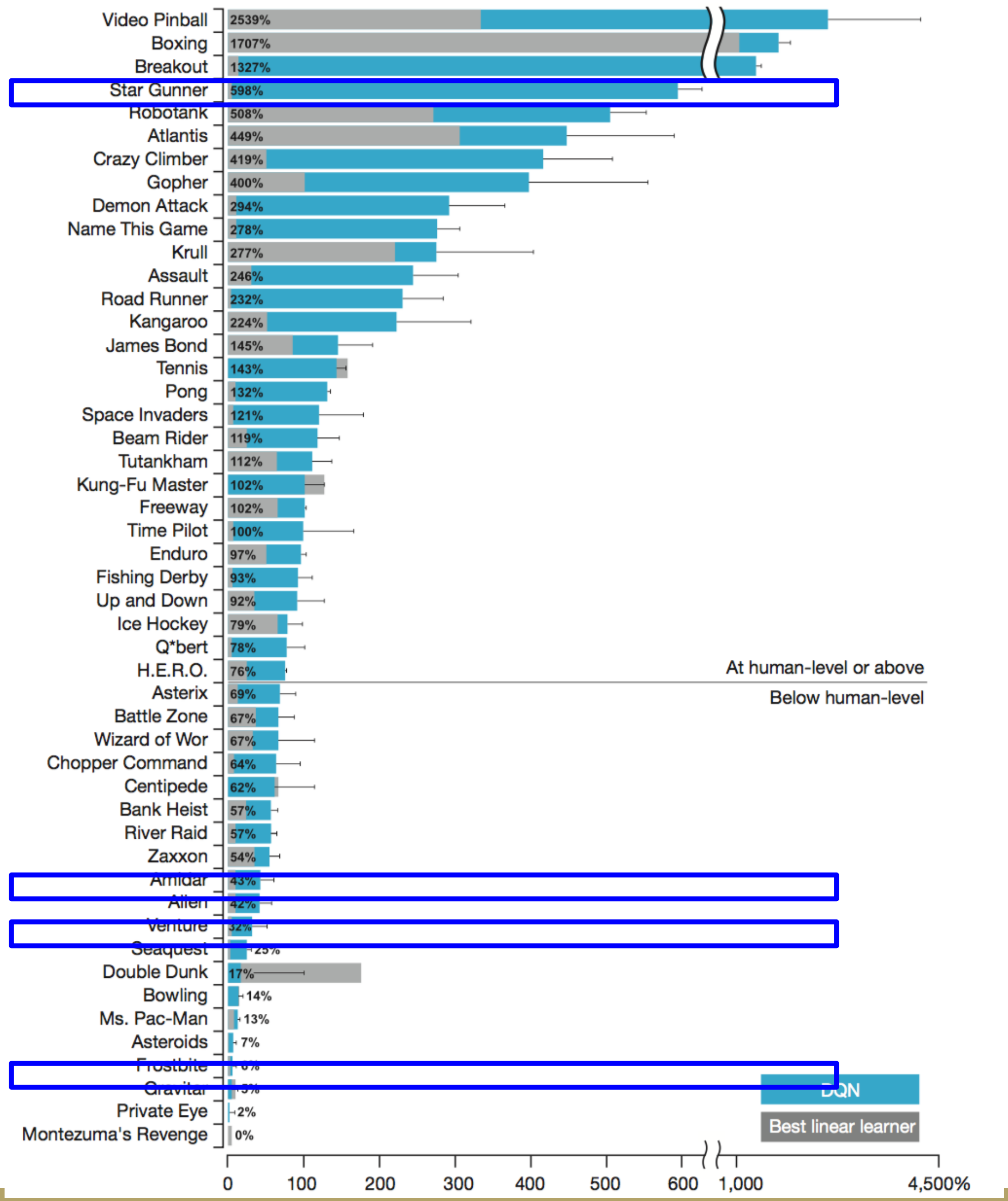


Star Gunner

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)

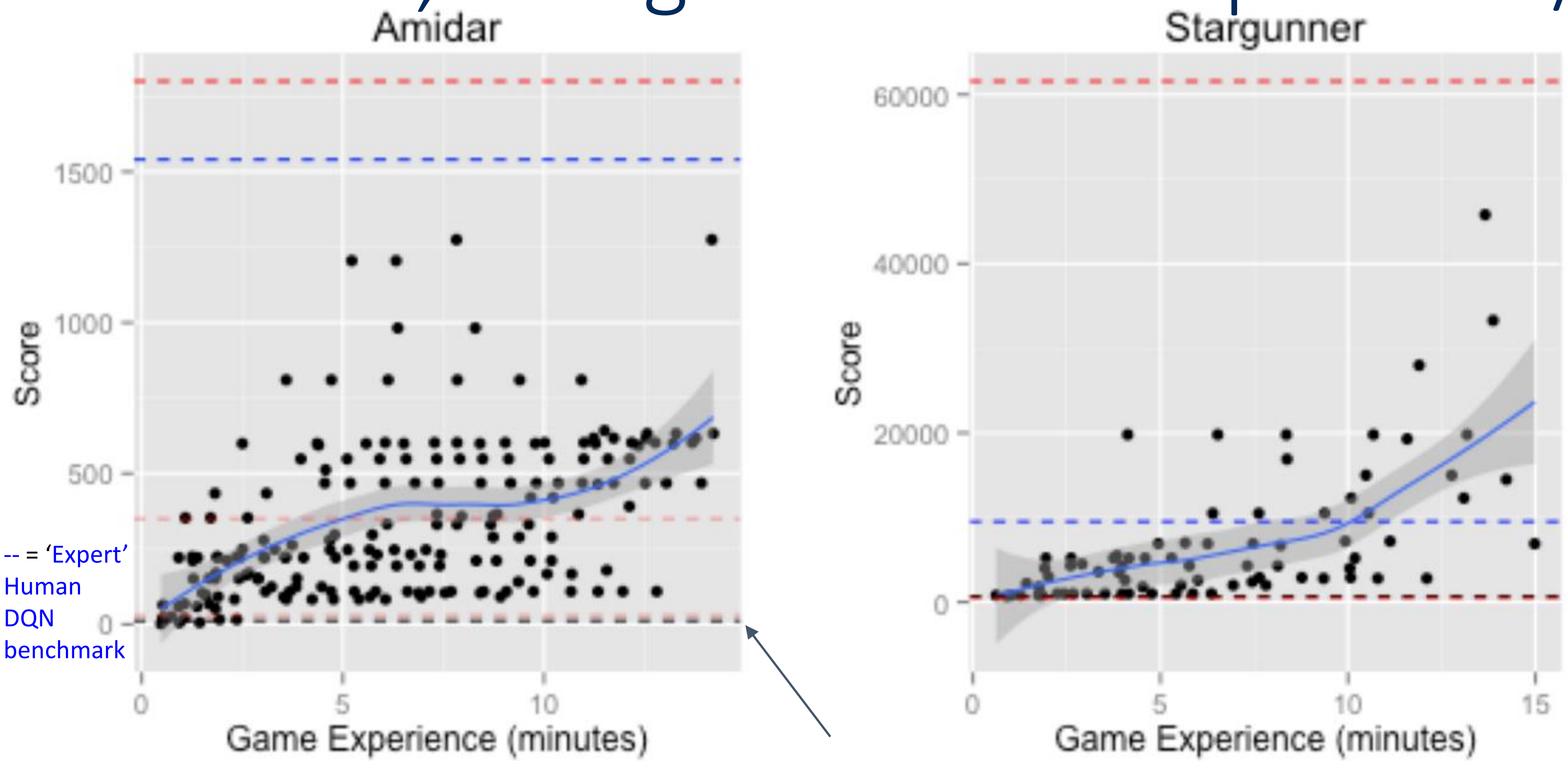
Experimental Results

>=1 slide

State results

Show figures / tables / plots

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

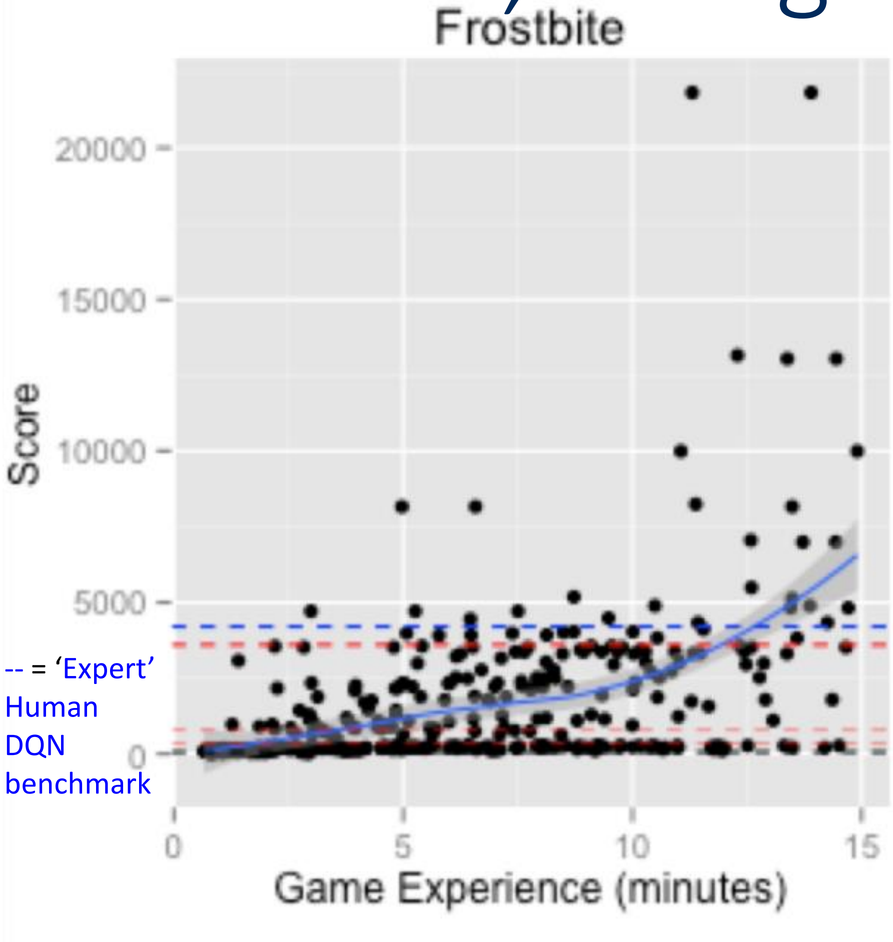


-- = ‘Expert’ Human DQN benchmark

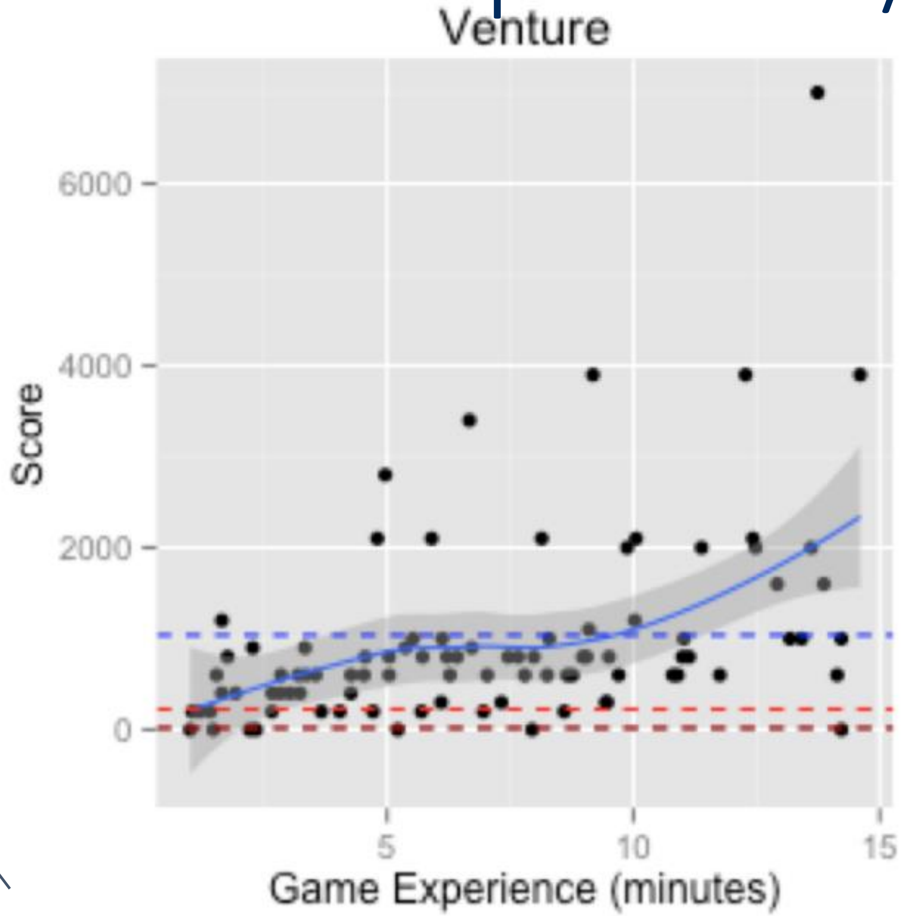
-- = DQN after 46 / 115/ 920 hrs

-- = Random play

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



-- = ‘Expert’
Human
DQN
benchmark



-- = Random play

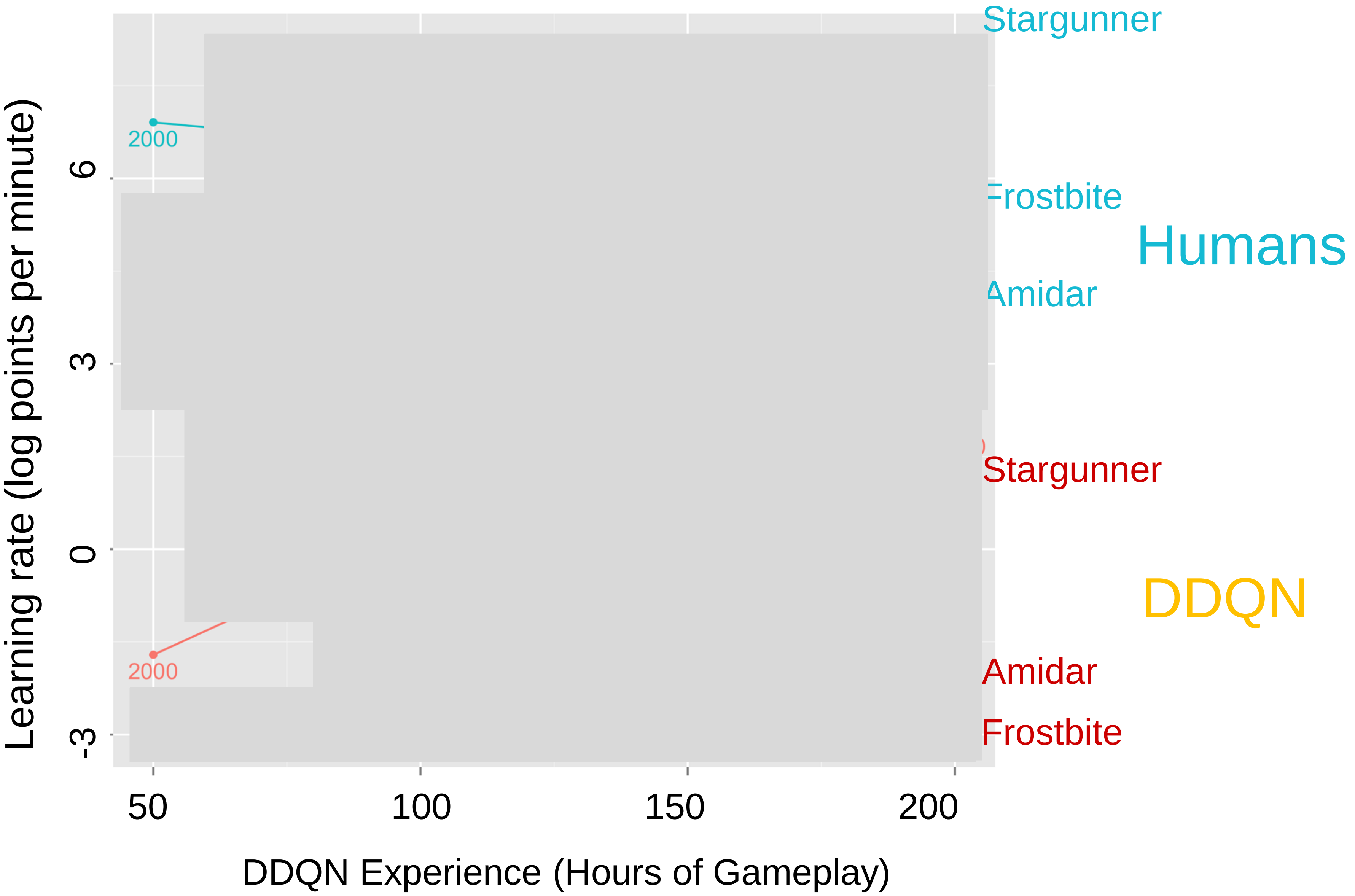
-- = DQN after 46 / 115/ 920 hrs

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.

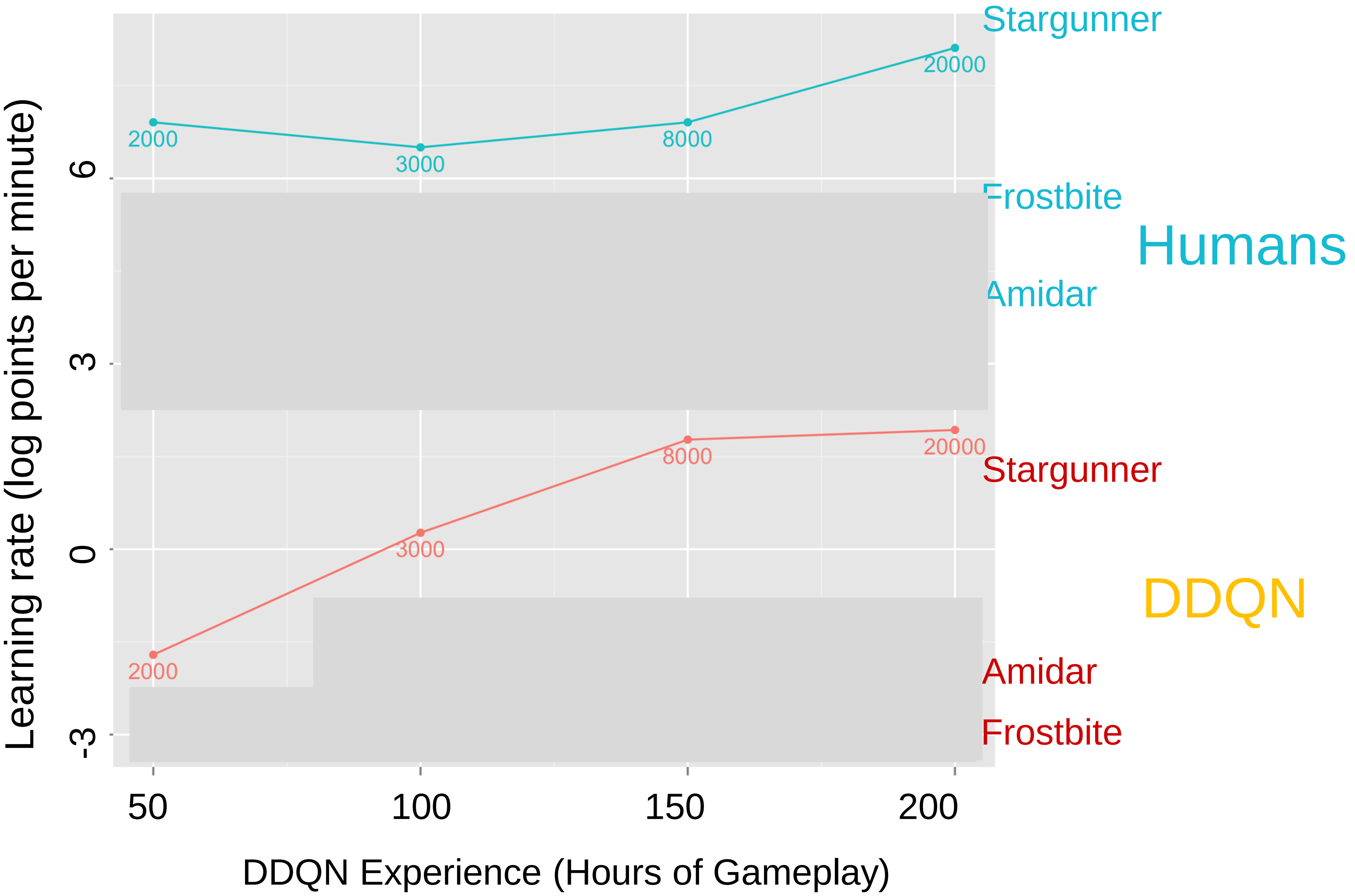
Learning rates matched for score level

Note: Y-axis is in Log!

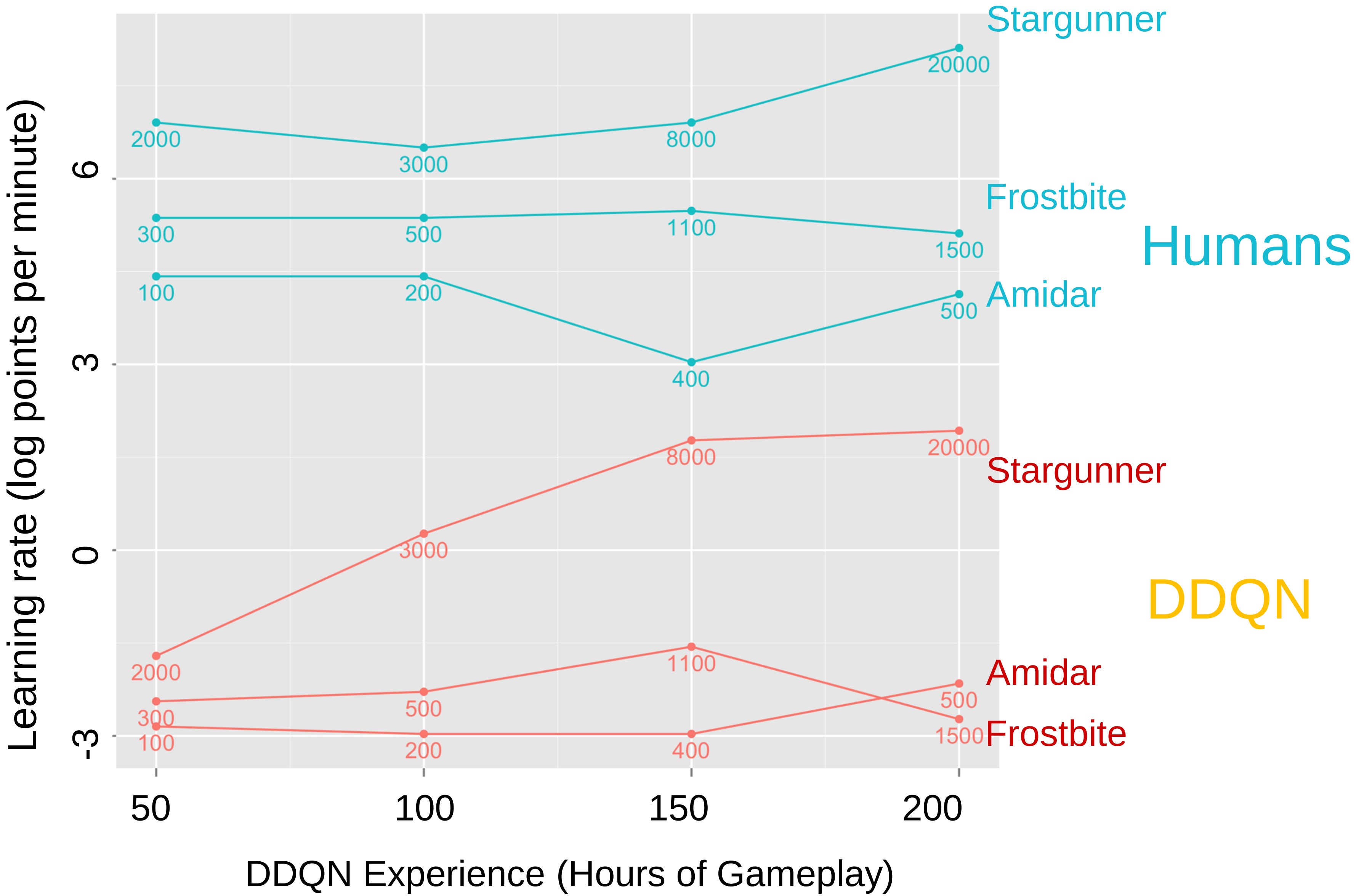


People are Learning Faster at Each Stage of Performance

Note: Y-axis is in Log!



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

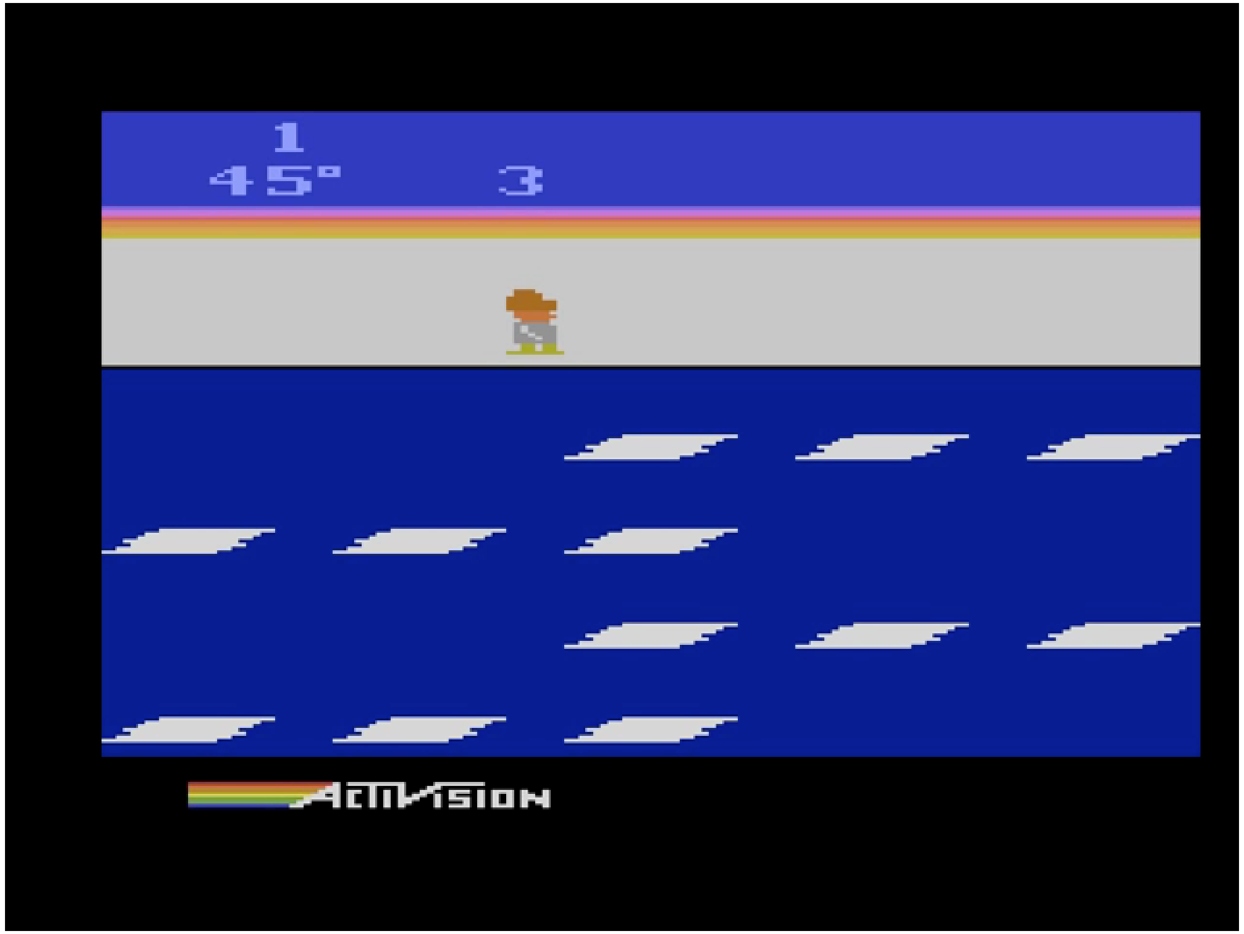


Methods: Observation & Experiment

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

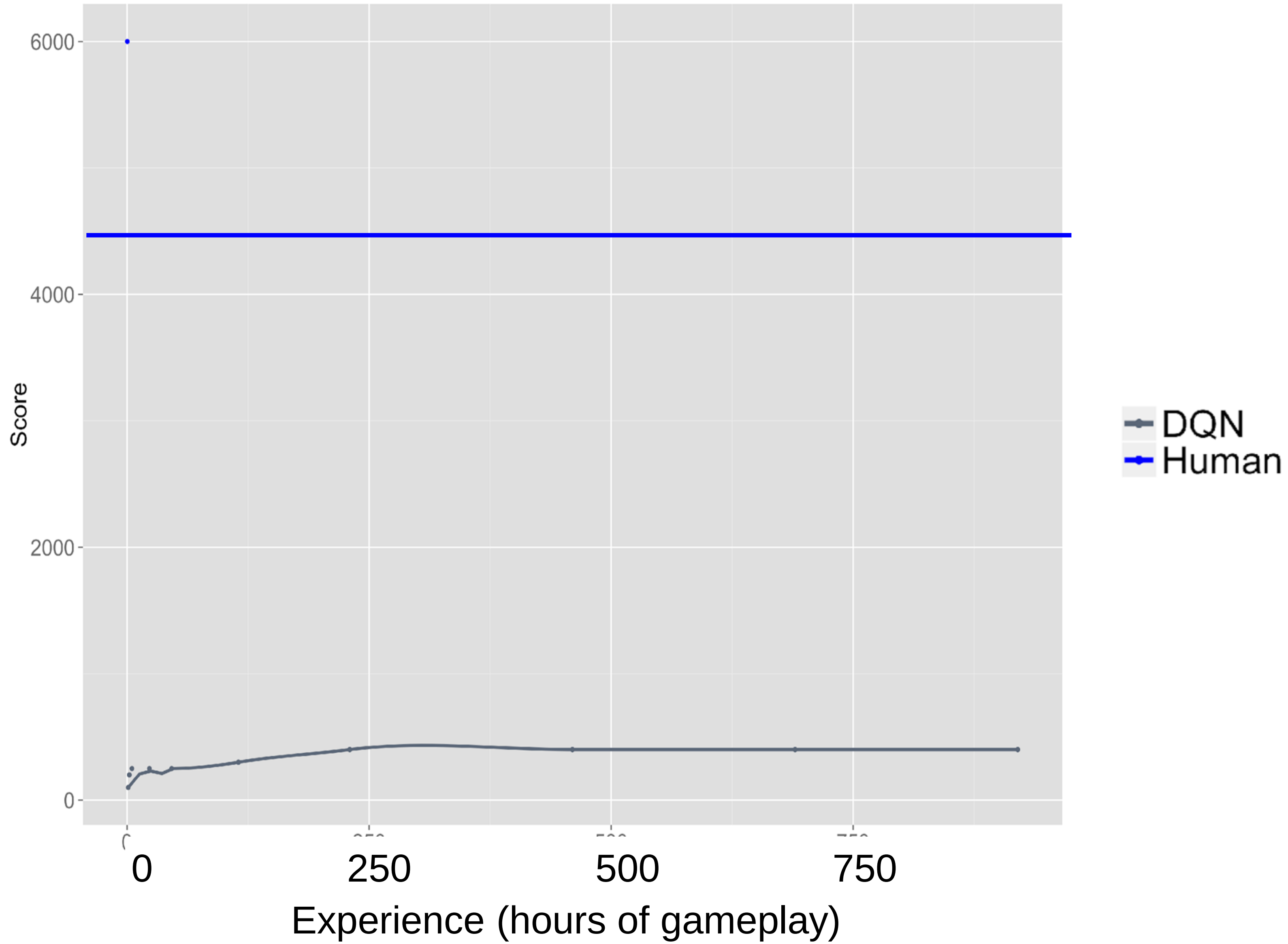
The “Frostbite challenge”

Why Frostbite? People do particularly well vs DDQN

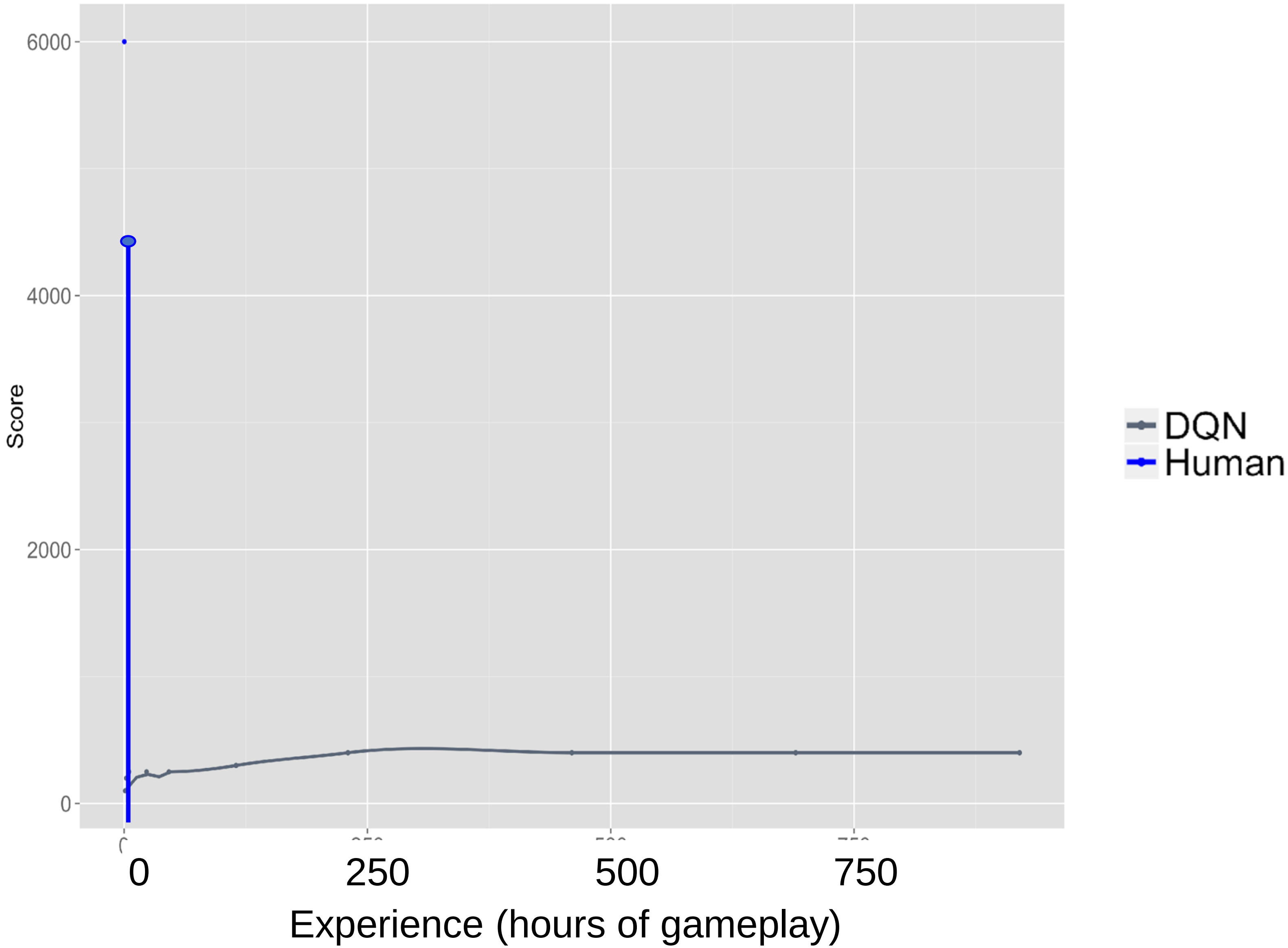


See Lake, Ullman, Tenenbaum & Gershman (forthcoming). Building machines that learn and think like people. *Behavioral and Brain Sciences*.

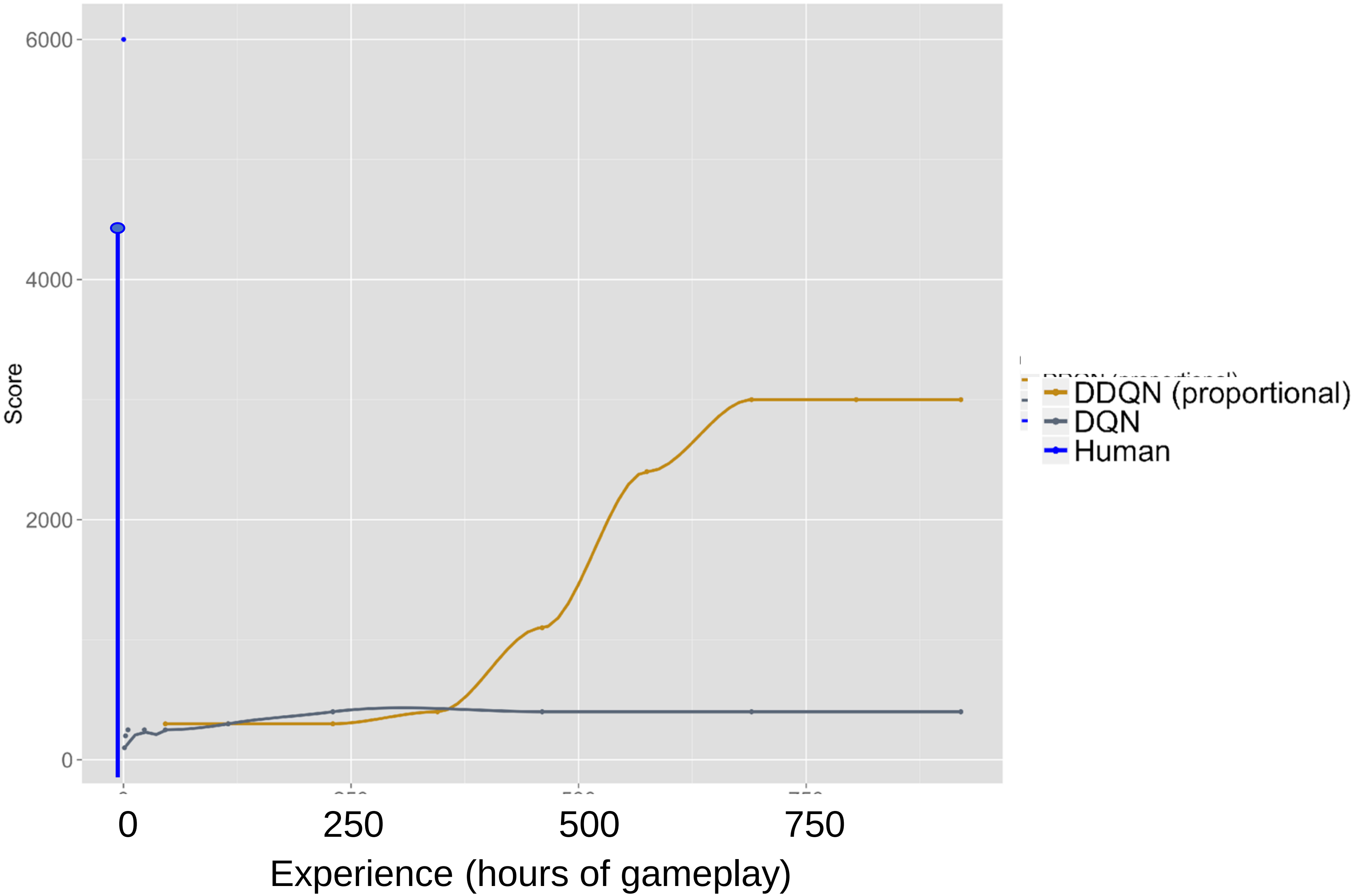
Frostbite


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

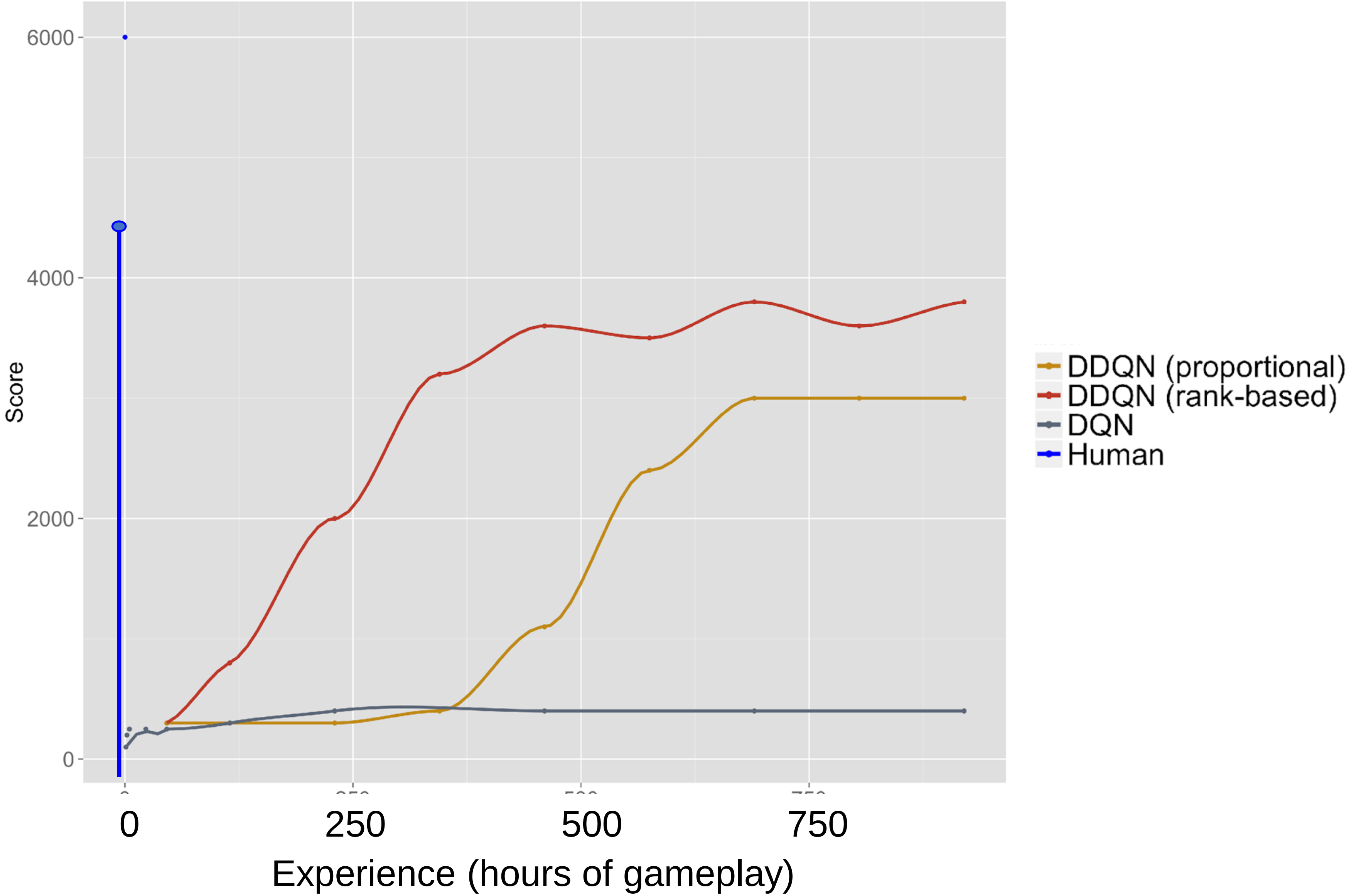
Frostbite


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

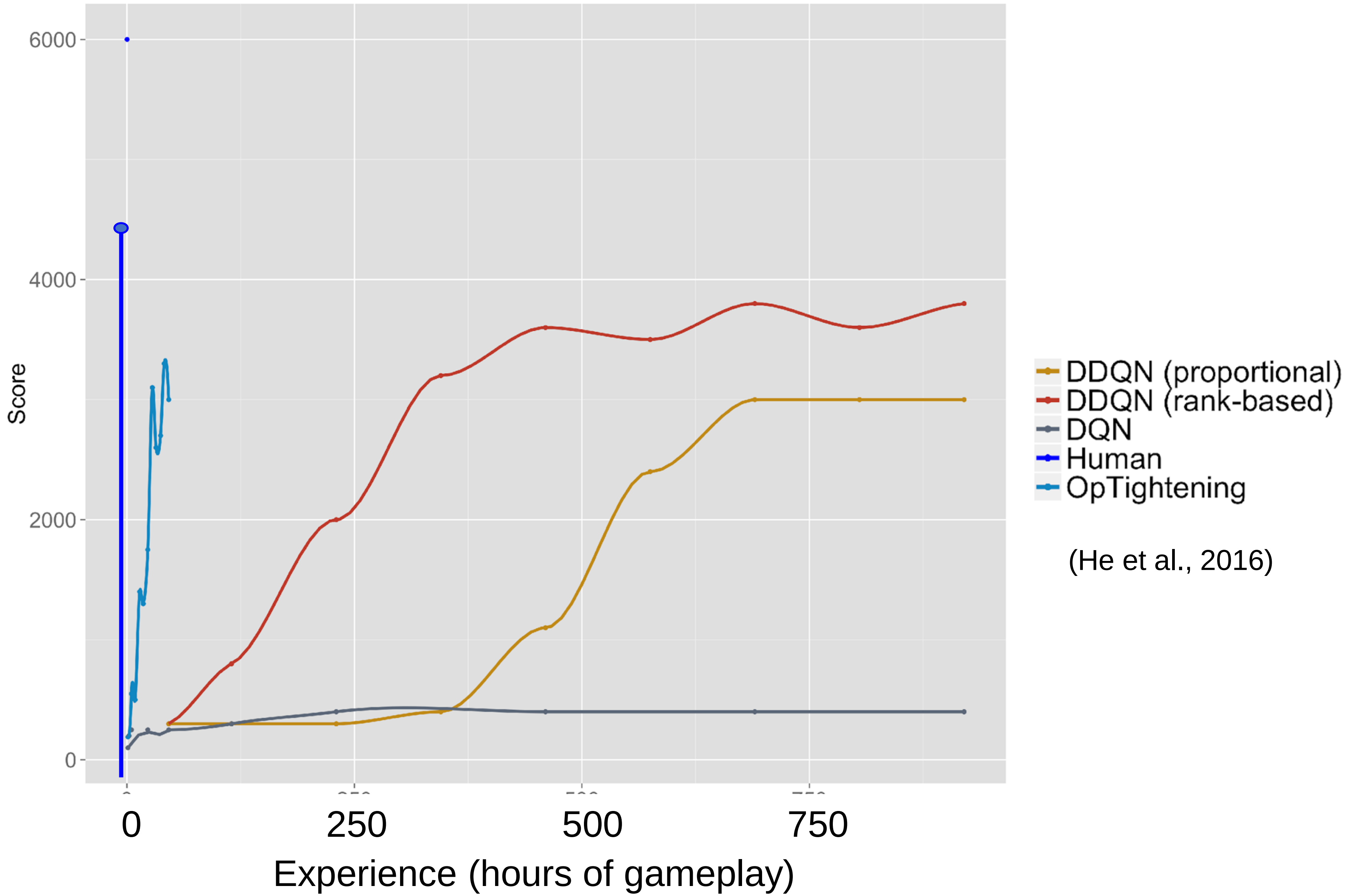
Frostbite


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

Frostbite

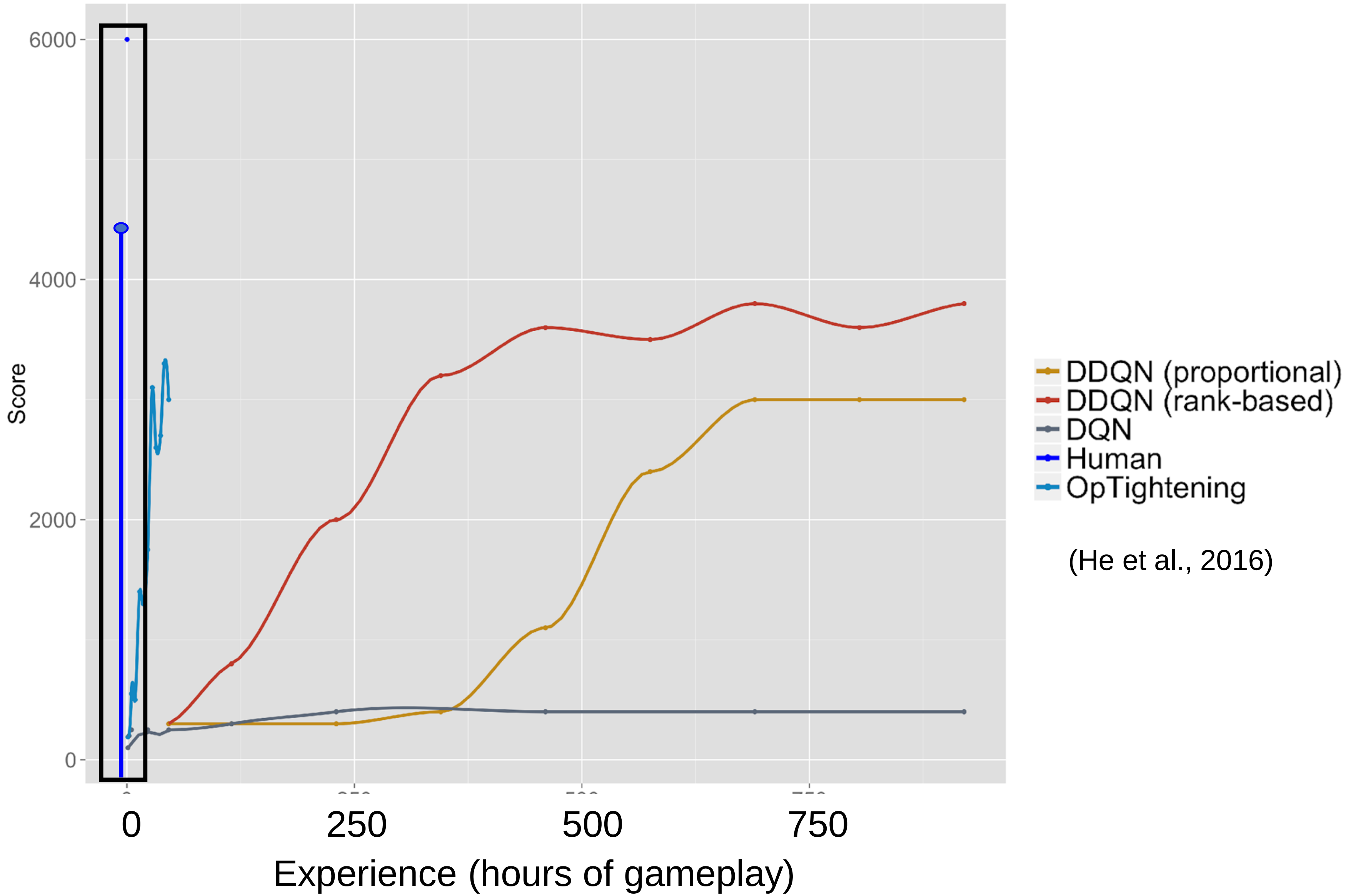

Score
6000
4000
2000
0
0
250
500
750
Experience (hours of gameplay)
DDQN (proportional)
DDQN (rank-based)
DQN
Human

Frostbite



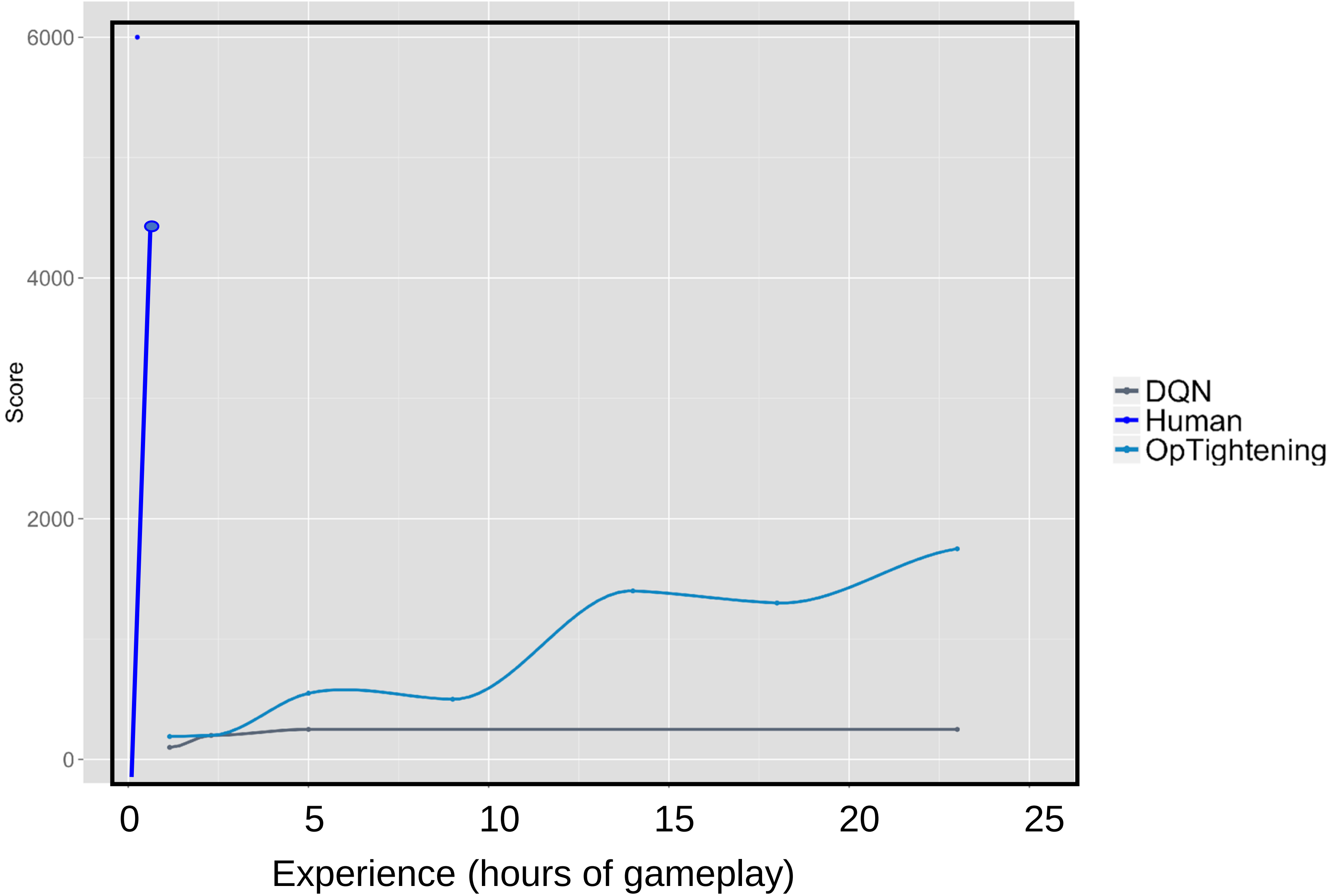
(He et al., 2016)

Frostbite



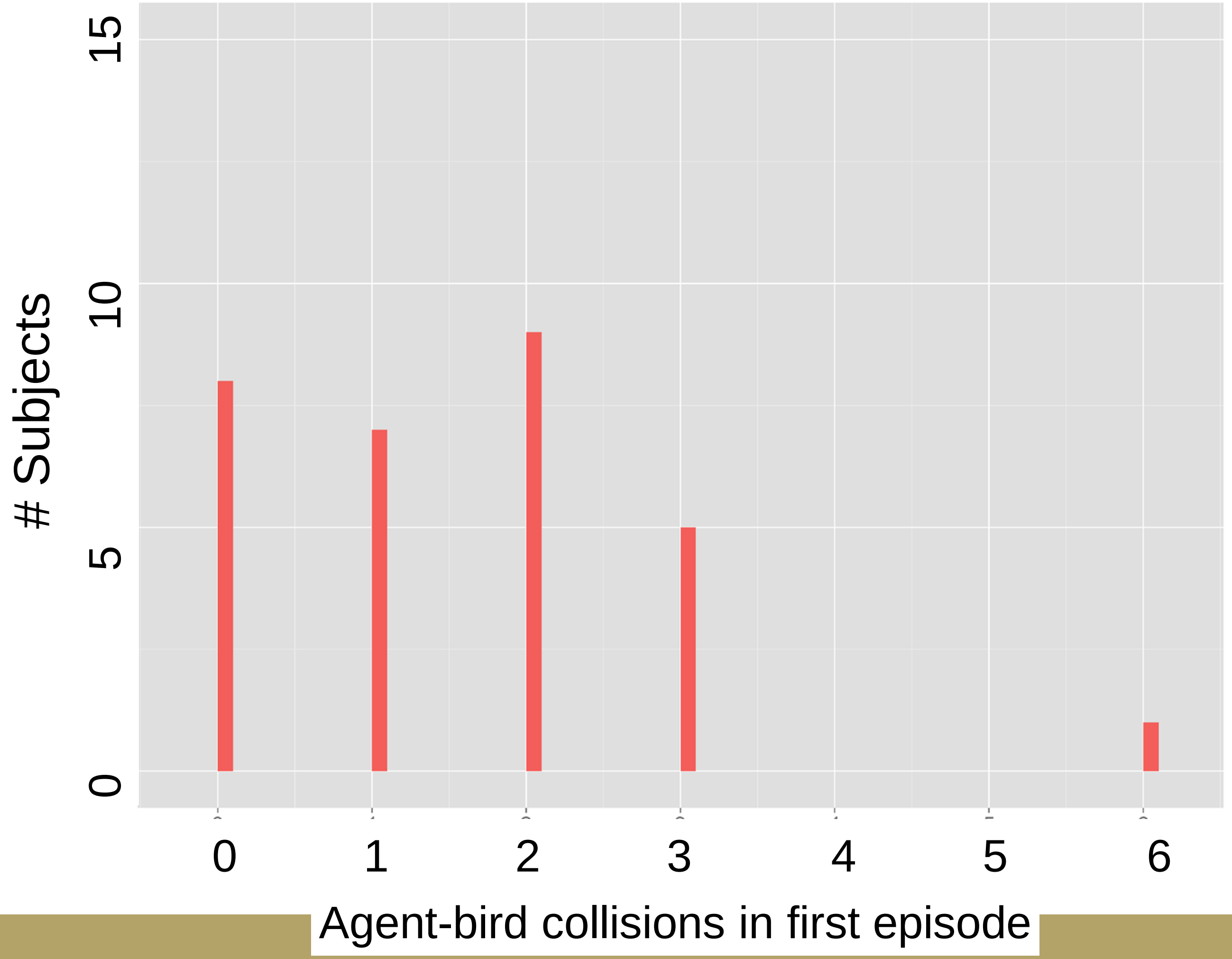
(He et al., 2016)

Frostbite

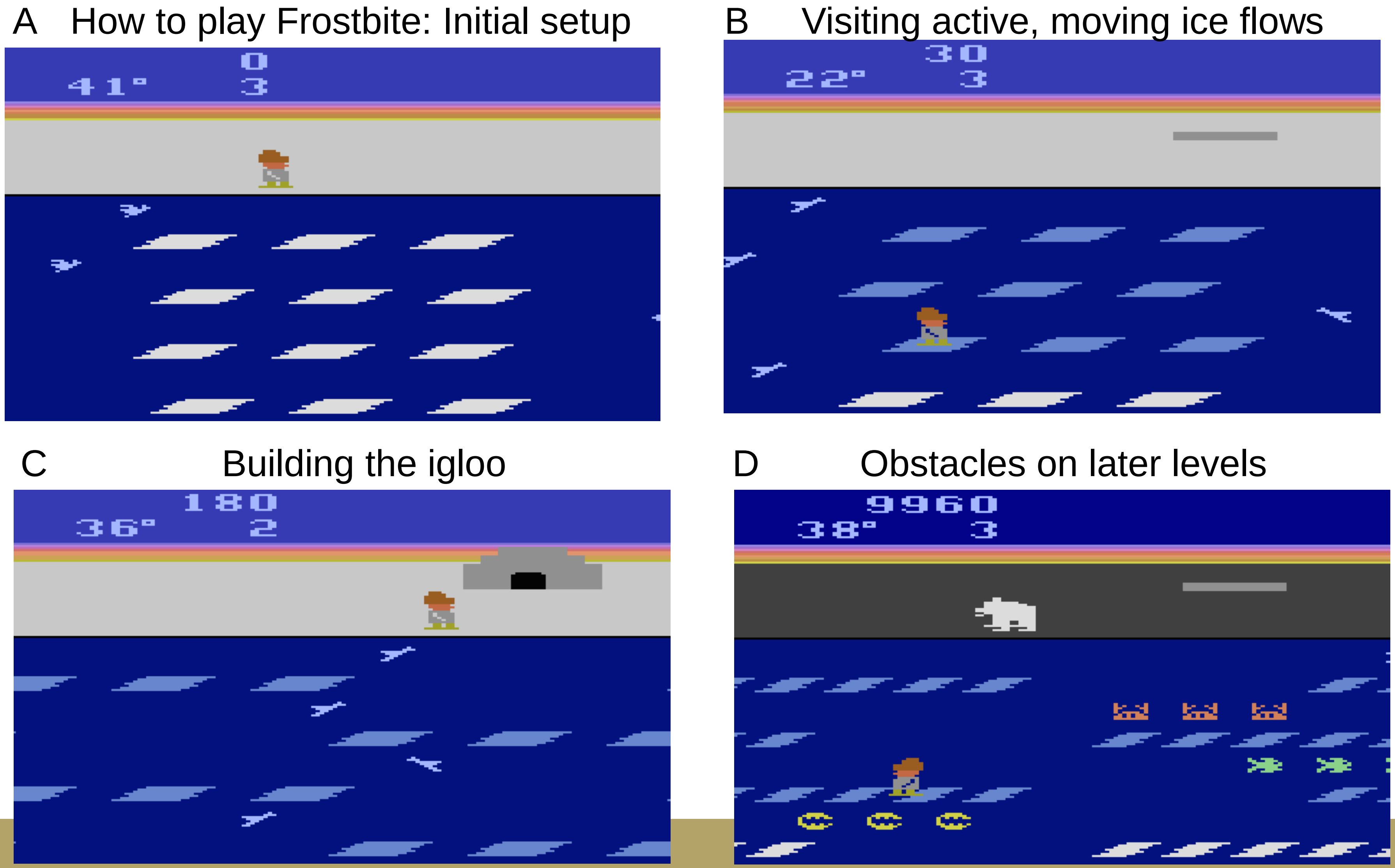

6000
4000
2000
0
Score
0
5
10
15
20
25
Experience (hours of gameplay)
DQN
Human
OpTightening

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

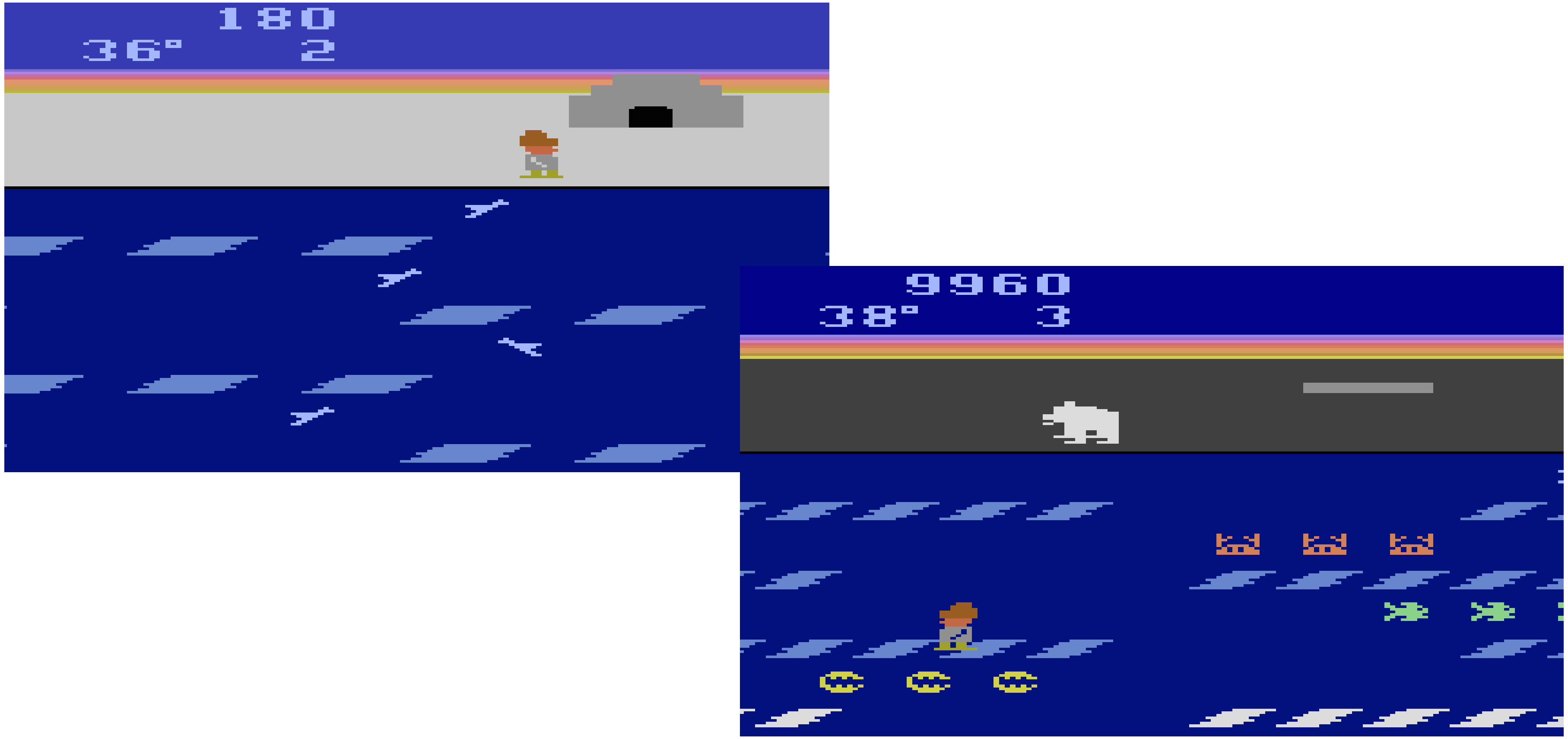
B Visiting active, moving ice flows

C Building the igloo

D Obstacles on later levels

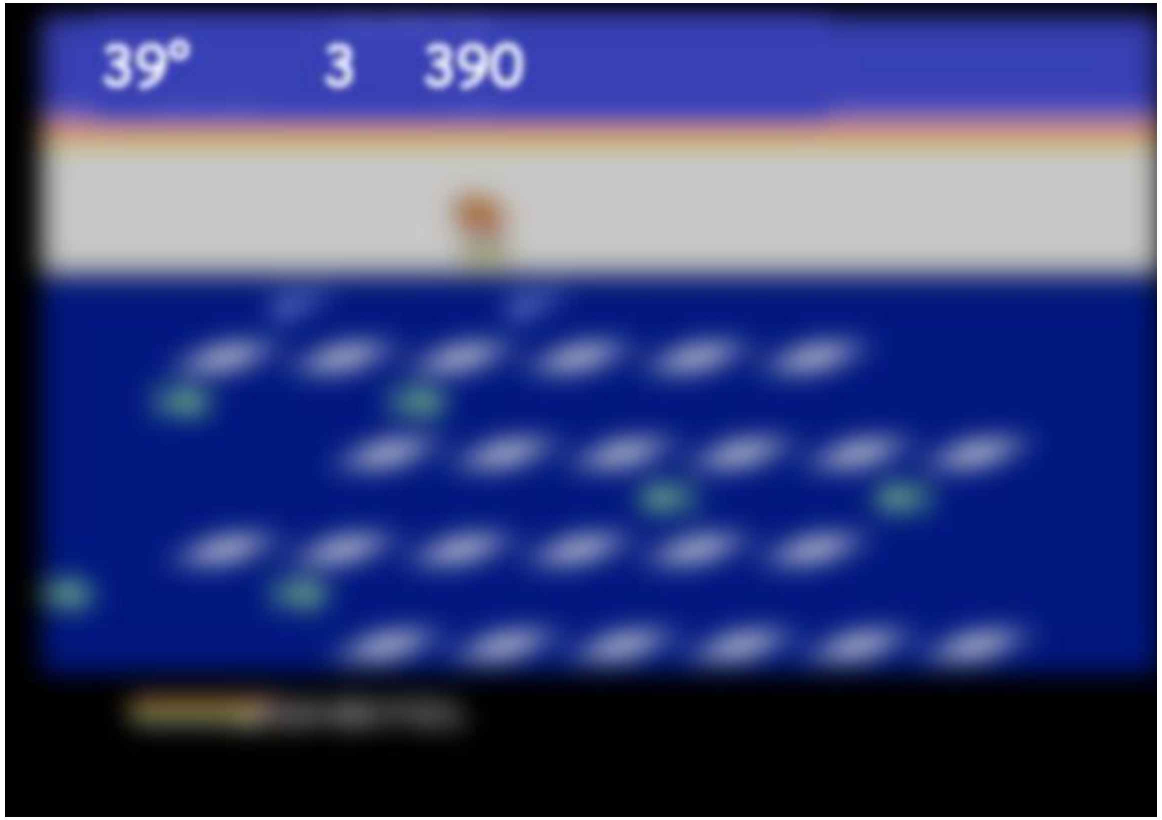
What drives such rapid learning?

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



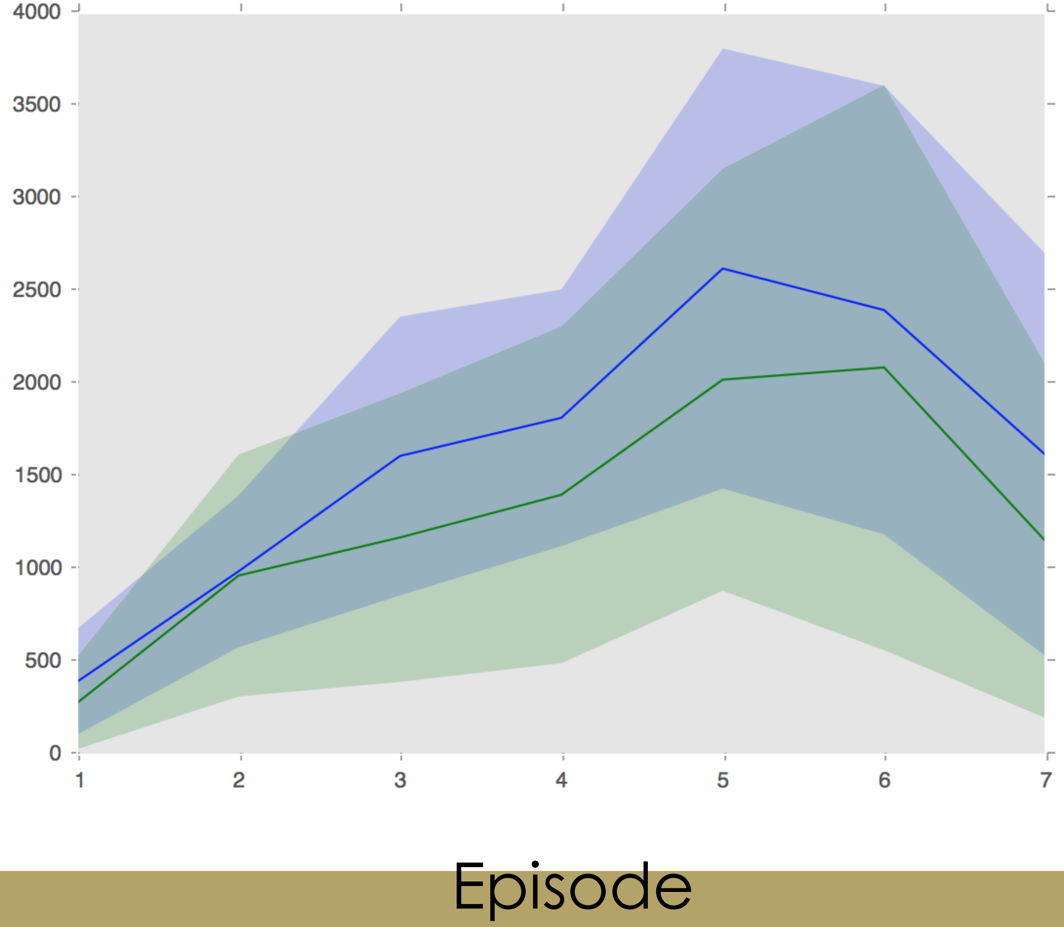
What drives such rapid learning?

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?



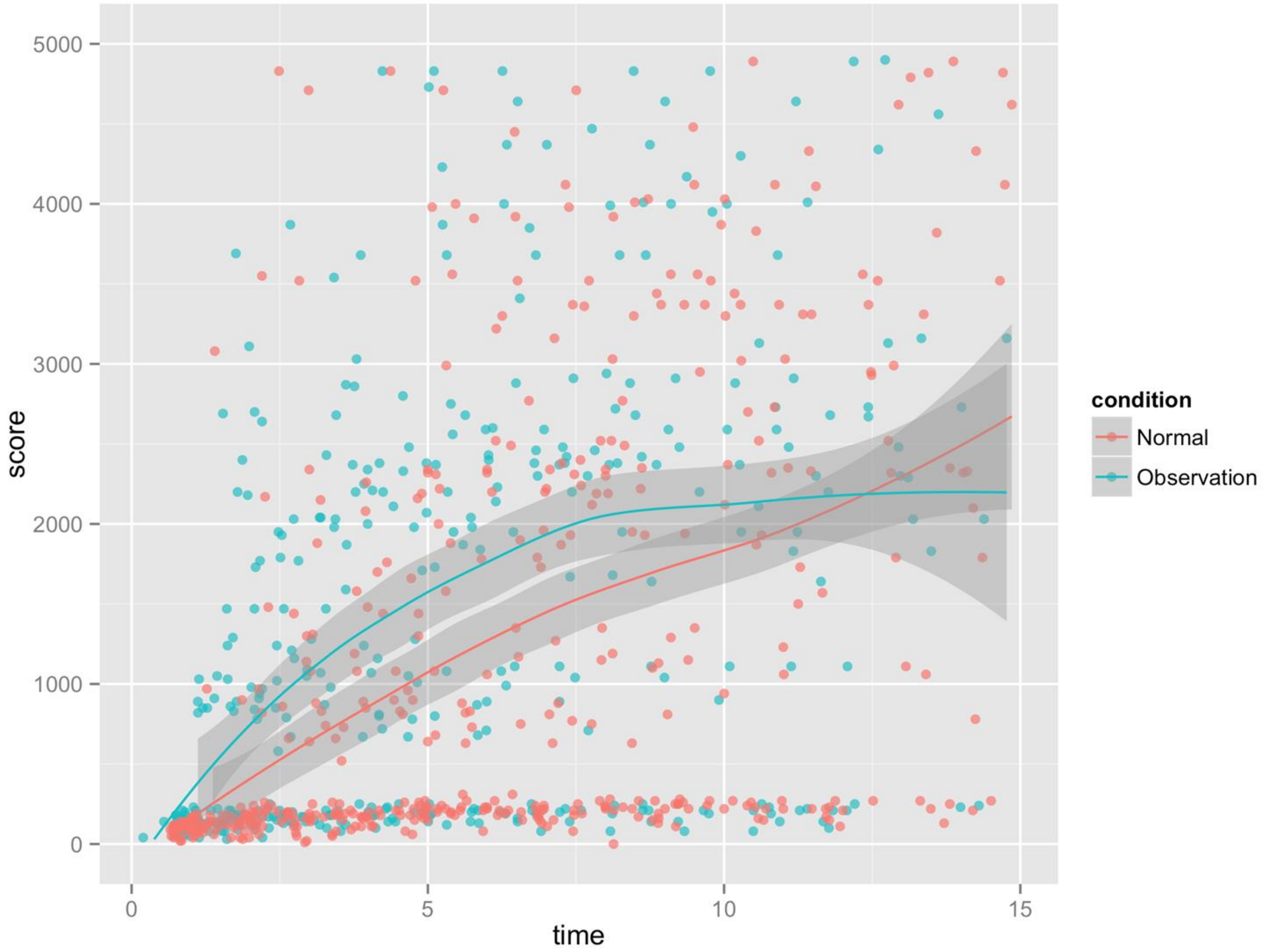
Being "object-oriented" in exploration matters, but prior world knowledge about specific object types doesn't so much!

What Drives Such Rapid Learning?

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

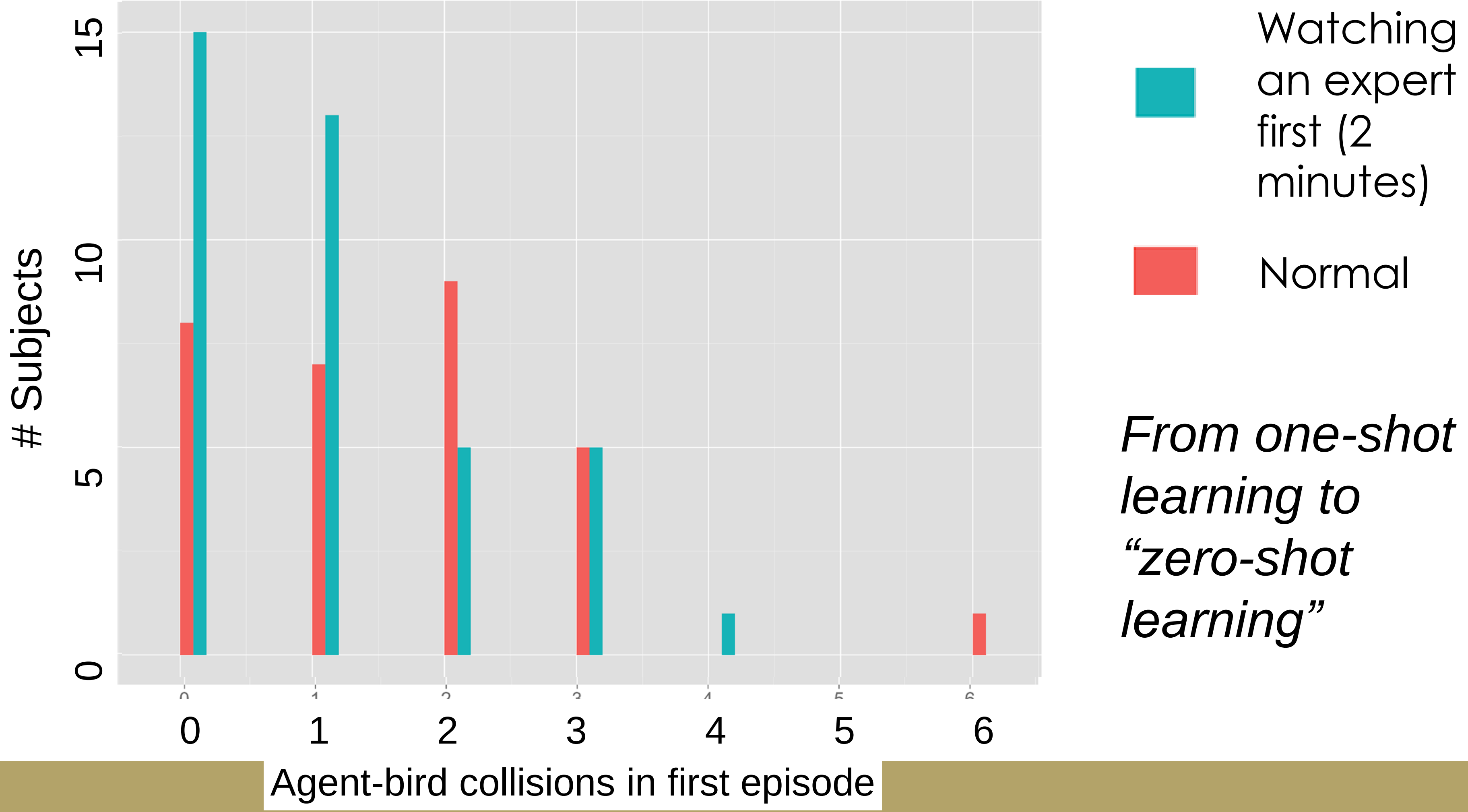
What drives such rapid learning?

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



What drives such rapid learning?

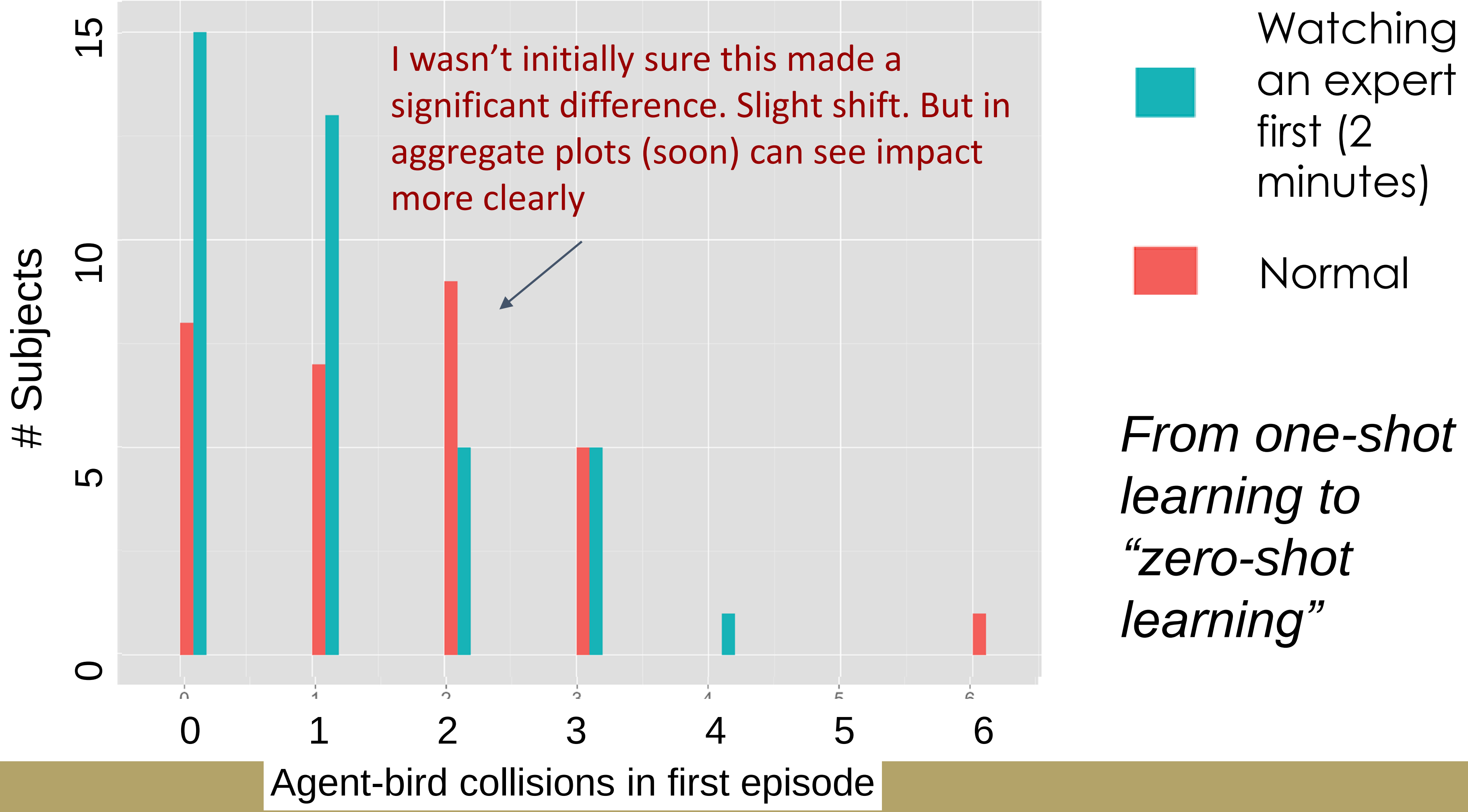
People can learn even faster if they combine their own experience with just a little help from others:



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:



I wasn’t initially sure this made a significant difference. Slight shift. But in aggregate plots (soon) can see impact more clearly

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.

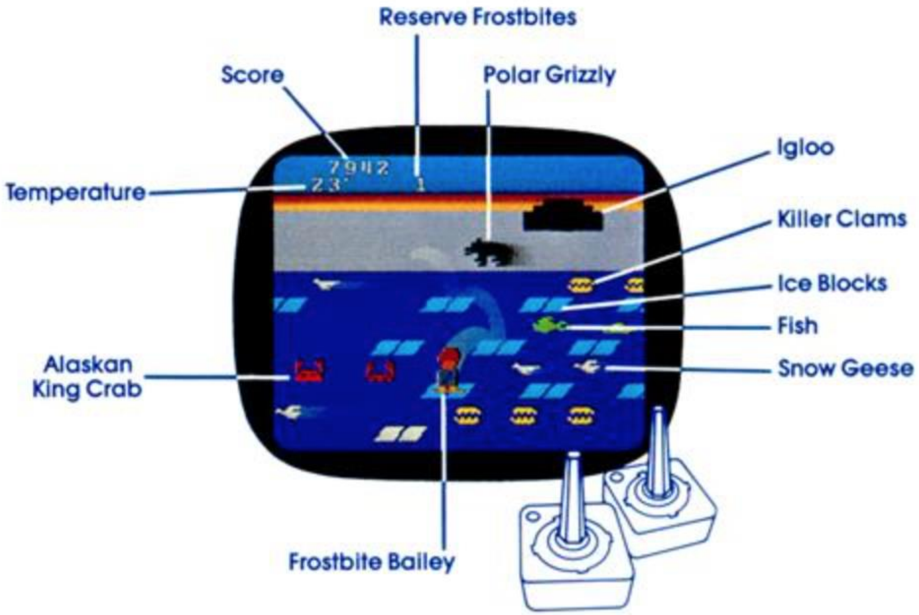
You begin the game with one active Frostbite Bailey and three on reserve. With each increase of 5,000 points, a bonus Frostbite is added to your reserves (up to a maximum of nine).

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.

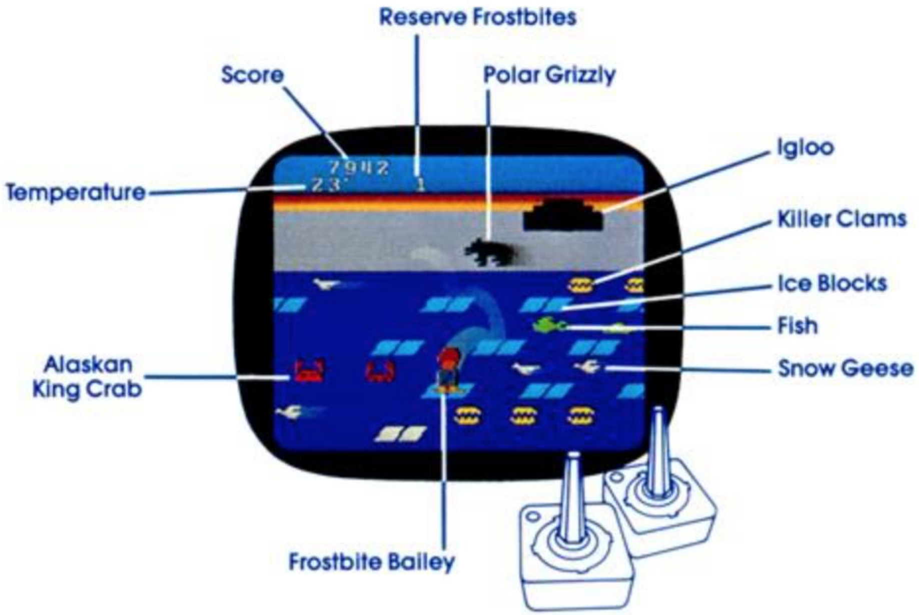
You begin the game with one active Frostbite Bailey and three on reserve. With each increase of 5,000 points, a bonus Frostbite is added to your reserves (up to a maximum of nine).

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).

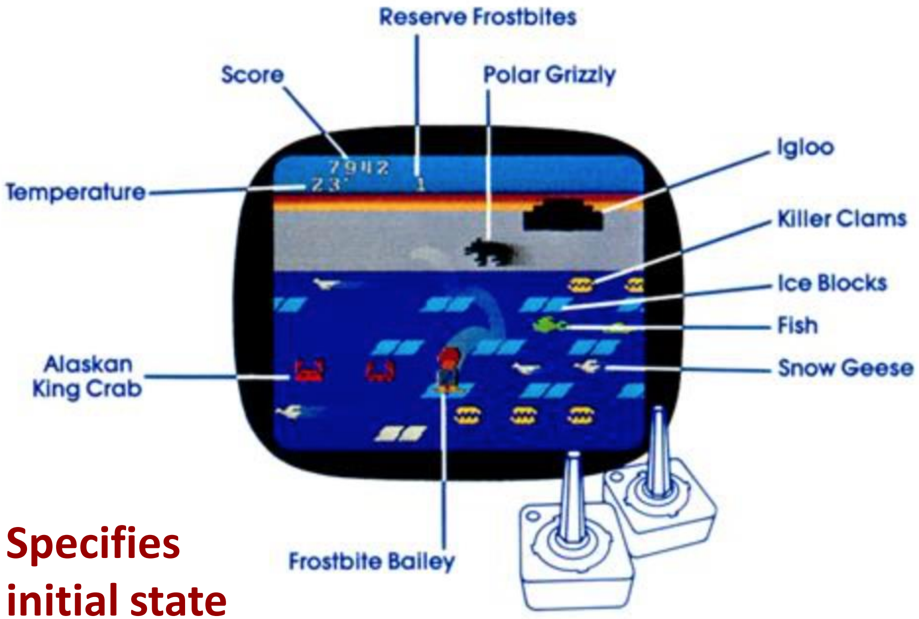
Specifies initial state

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.

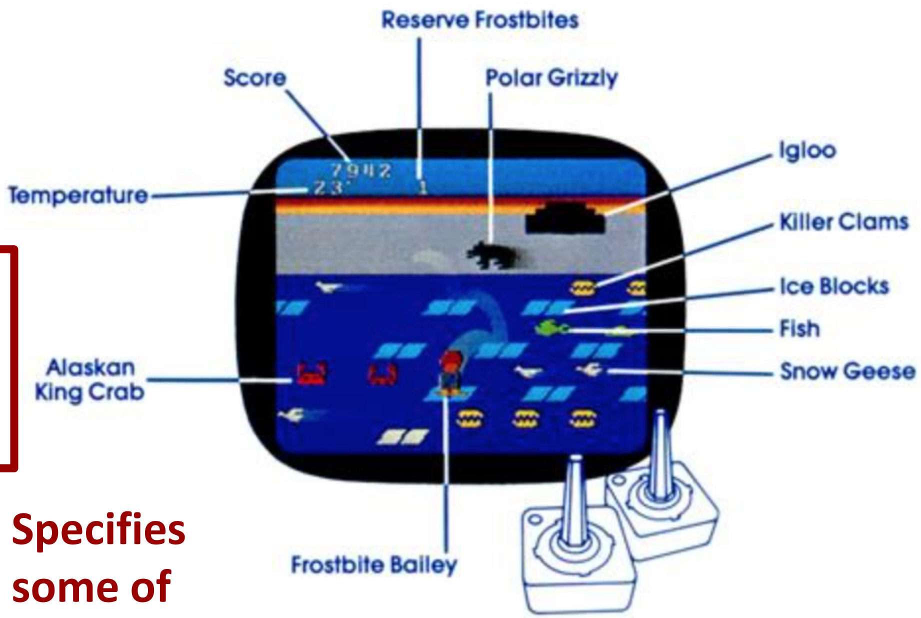
You begin the game with one active Frostbite Bailey and three on reserve. With each increase of 5,000 points, a bonus Frostbite is added to your reserves (up to a maximum of nine).

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 some of dynamics

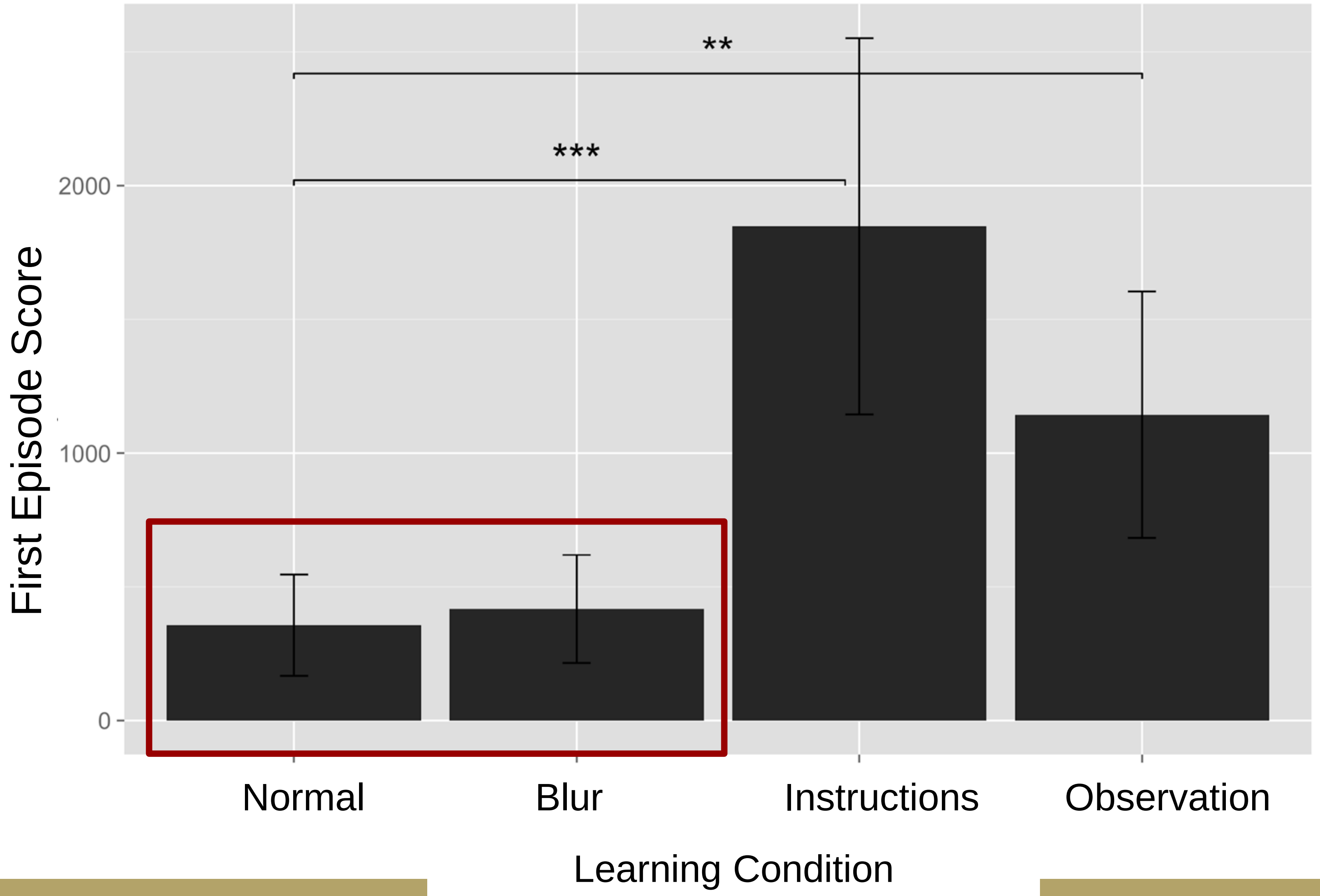
Work hazards. Avoid contact with Alaskan King Crabs, snow geese, and killer clams, as they will push Frostbite Bailey into the fatal Arctic Sea. The Polar Grizzlies come out of hibernation at level 4 and, upon contact, will chase Frostbite right off-screen.

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!

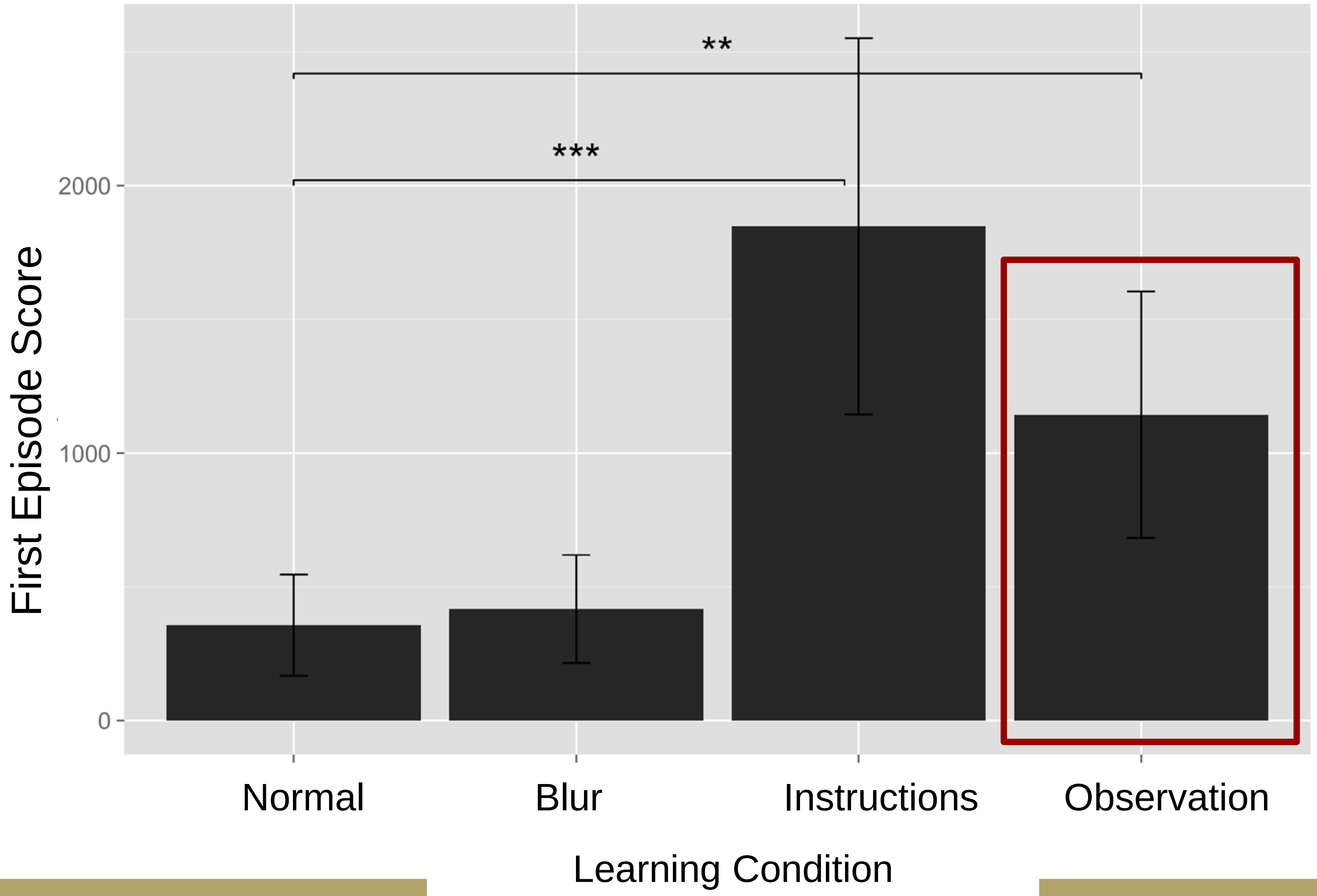
SPECIAL FEATURES OF FROSTBITE

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.

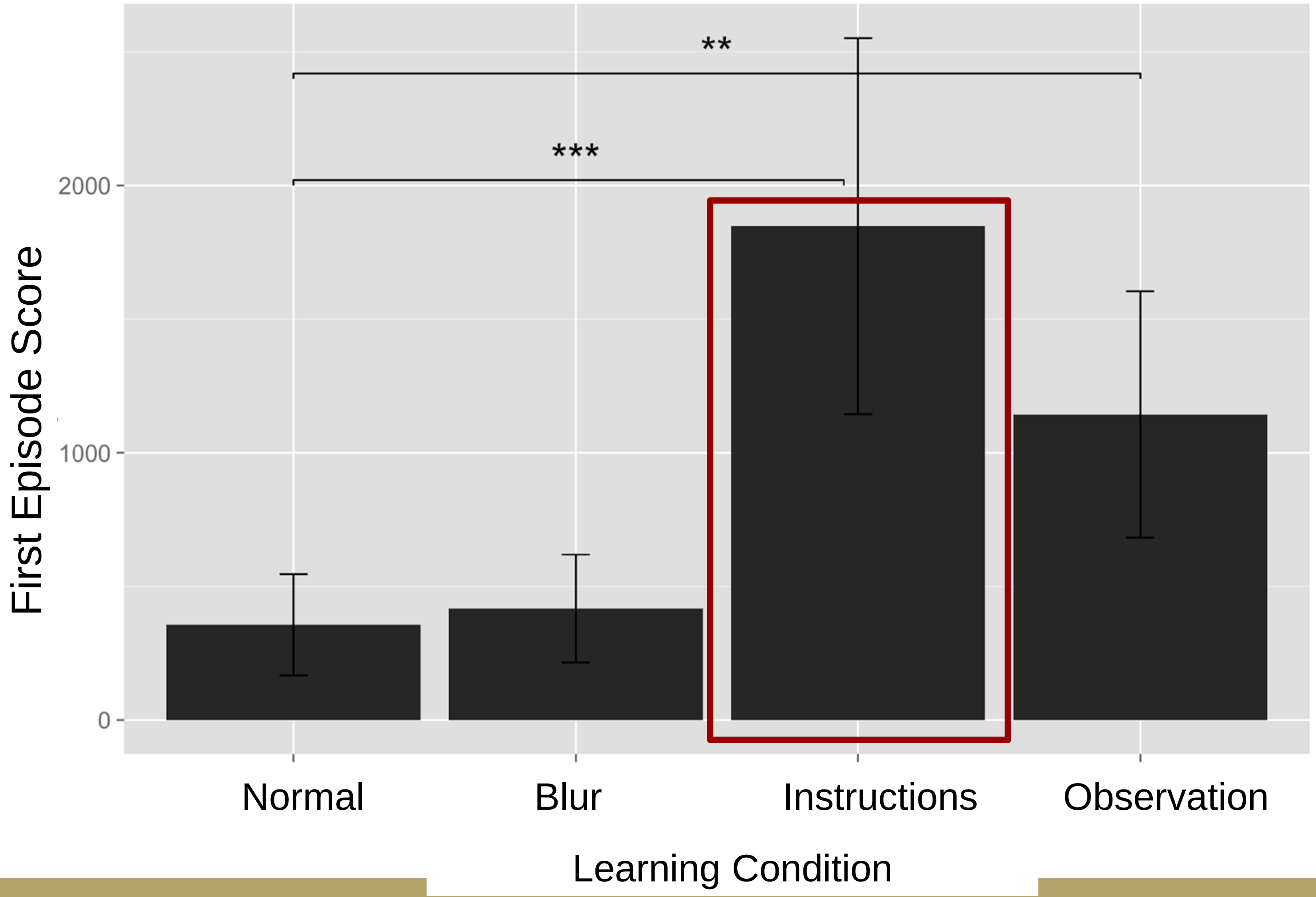
Humans aren’t relying on specific object knowledge



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?)

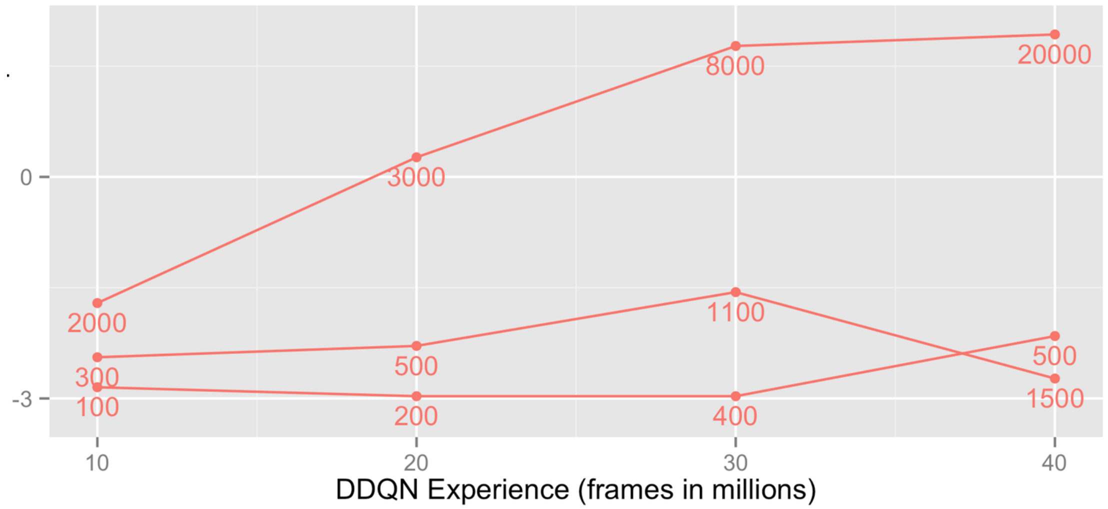
- If follow up work has addressed some of these limitations, include pointers to that. But don’t limit your discussion only to the problems / limitations that have already been addressed.

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)

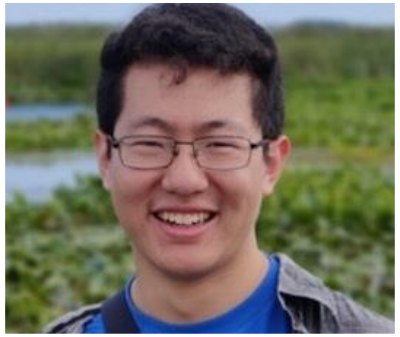
- Problem the reading is discussing
- Why is it important and hard
- What is the key limitation of prior work
- What is the key insight(s) (try to do in 1-3) of the proposed work
- What did they demonstrate by this insight? (tighter theoretical bounds, state of the art performance on X, etc)

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.

Course Logistics

Animesh Garg



Liqaun Wang

Albert Wilcox

Uzair Akbar

- Contact us at through CANVAS (direct email discouraged)
- For room information, office hours, etc., see website and canvas: https://pairlab.github.io/cs8803-f24/#

Note: The logistics info is subject to change in the first week of class.
The website and canvas will always contain the most up-to-date information, so please check frequently