“The ultimate fate of all intelligent beings has always been to become as grand as their thoughts.”

Liu Cixin, *Death's End*

https://danwang.co/liu-cixins-three-body/

CMPUT 628
Deep RL

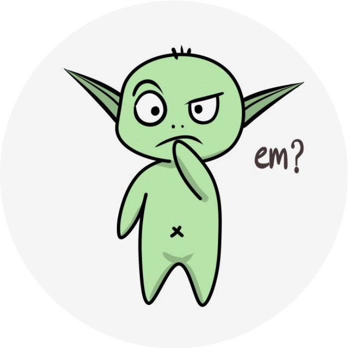
Marlos C. Machado

Class 13/ 25

Reminders & Notes

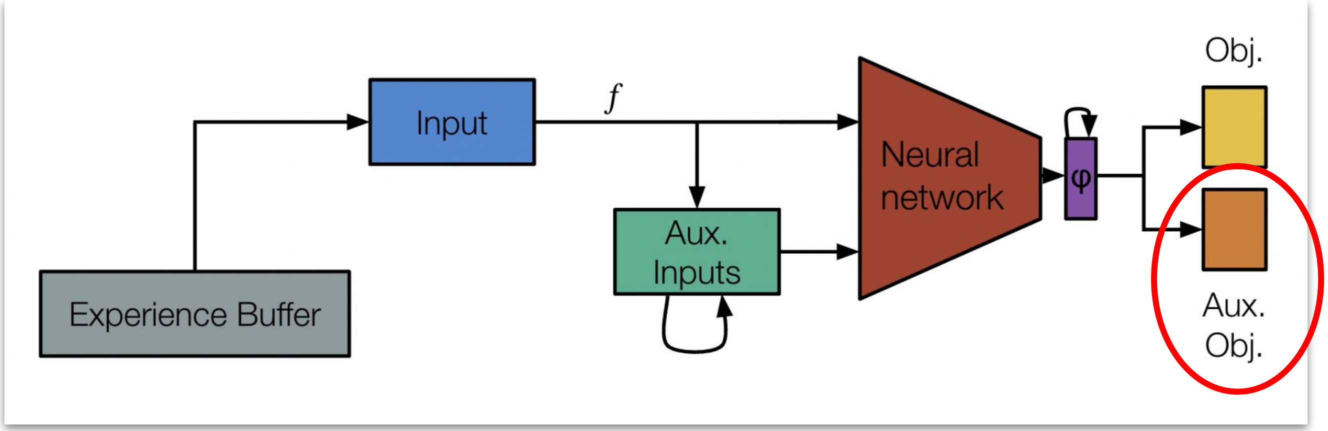
- Assignment 3 is out; it's due date is **March 7**
- I released the instructions for the Seminar and Paper Commentary
- Lecture notes: I have released v0.31 with the discussion about auxiliary tasks, but it will very likely be my last release — I don't think I'll have time to write about *distributional RL and what I discuss starting today*
- Mid-course Evaluation: No one complained, so I'll keep doing the same thing
  A reading list was requested, so I compiled it, in case it is useful
- Next Monday, A. Rupam Mahmood will give a guest lecture on streaming deep reinforcement learning (I won't be here, but you should come)

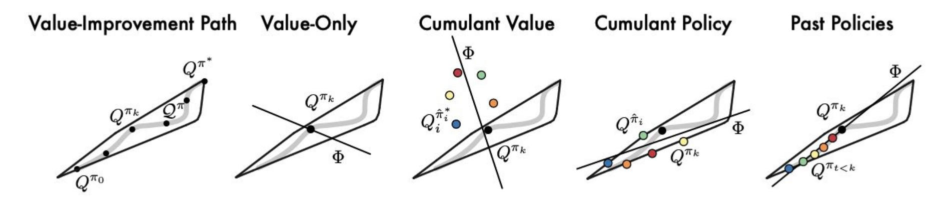
Please, interrupt me at any time!

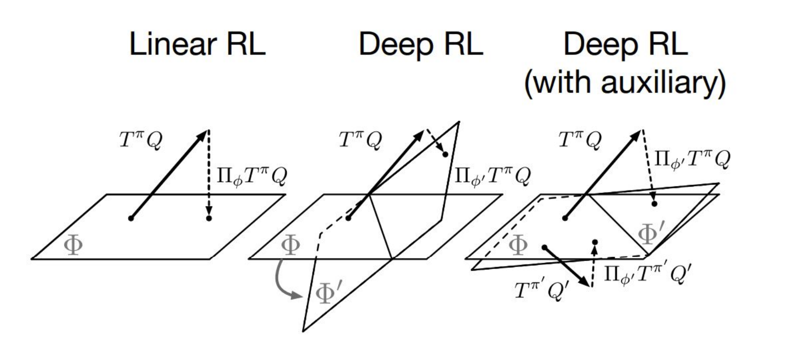


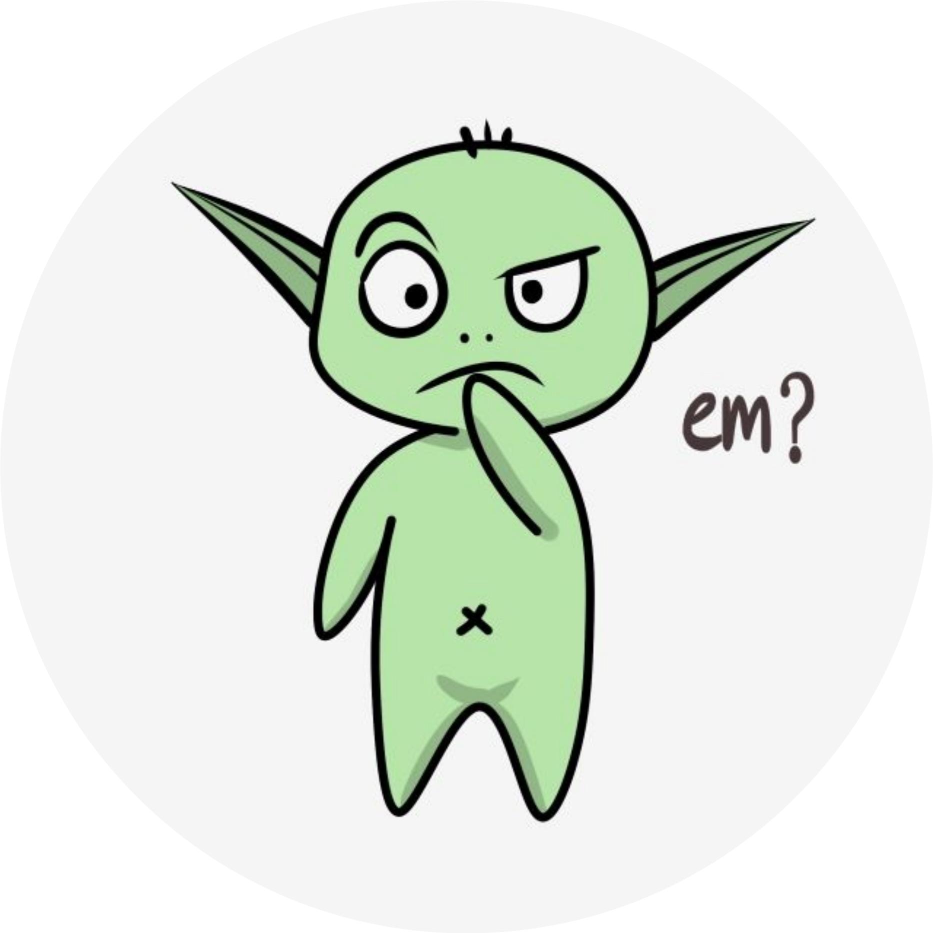
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Last Class: Auxiliary Tasks





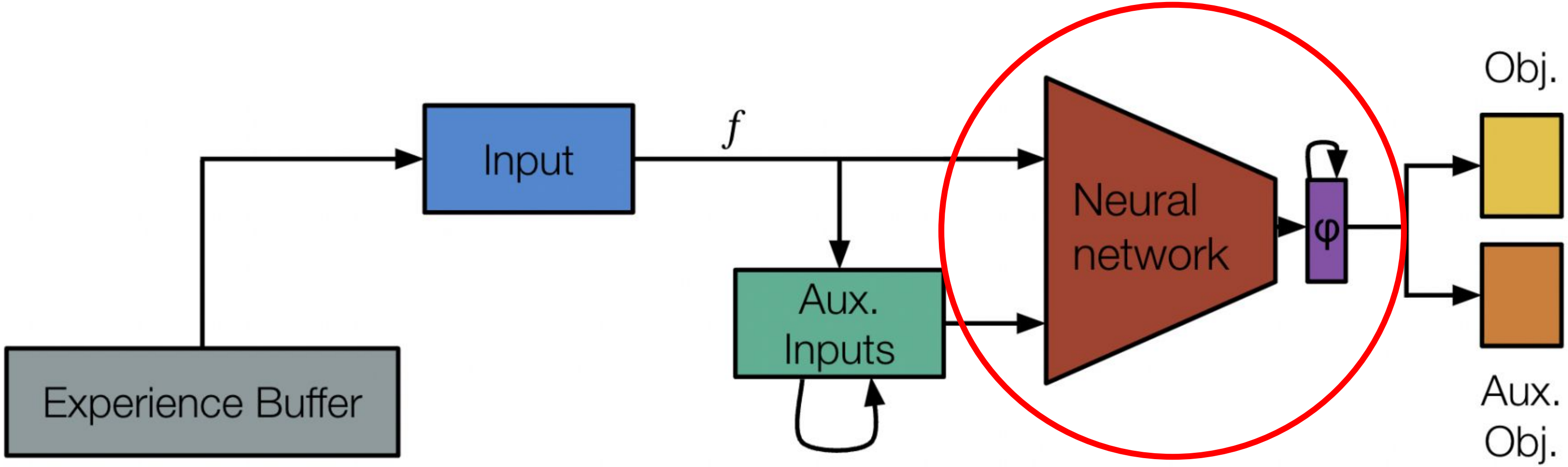




https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Neural Network Architecture

- Instead of changing objective functions, we can change the NN architecture

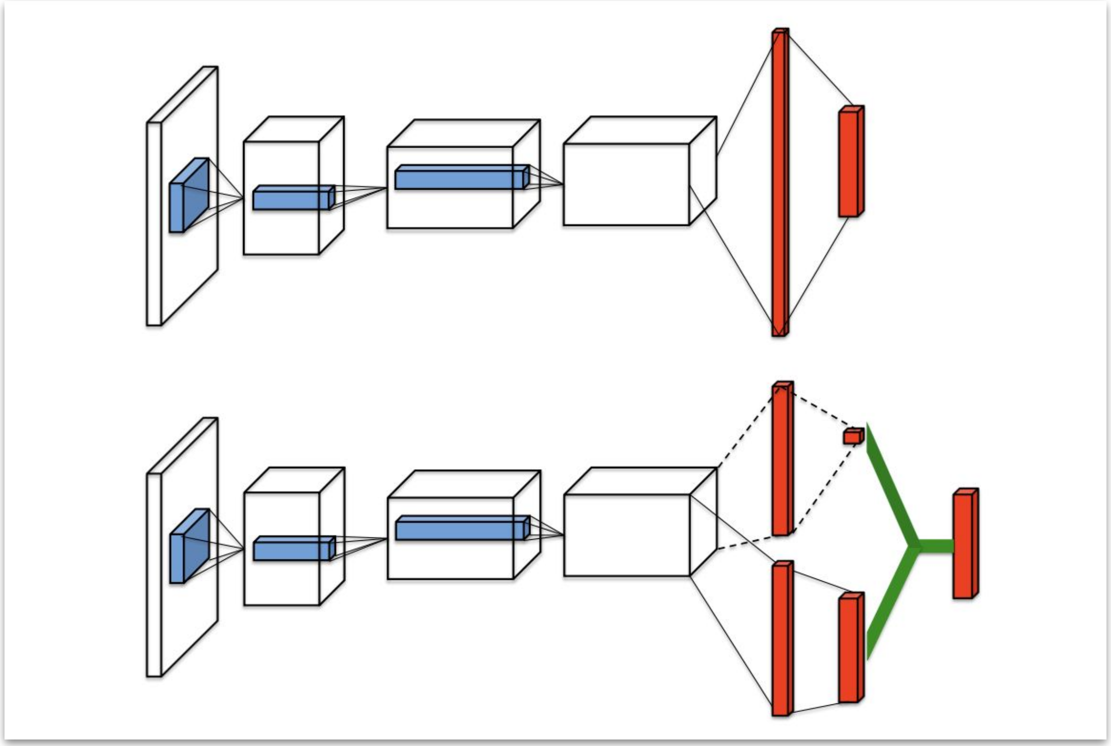


Duelling Networks [Wang et al., 2016]

- “Here, we take an alternative but complementary approach of focusing primarily on innovating a neural network architecture that is better suited for model-free RL”
- The goal is “to generalize learning across actions without imposing any change to the underlying reinforcement learning algorithm”
- The main idea is to have different estimators for state values and (state-dependent) action advantages, leveraging the fact that $a_\pi(s,a) = q_\pi(s,a) - v_\pi(s)$

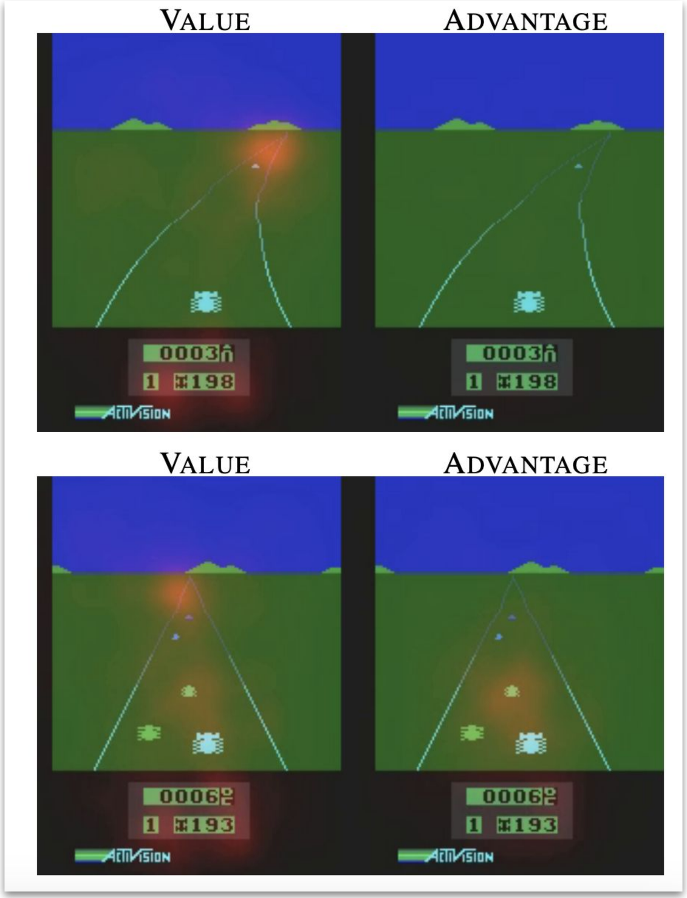
  Note that $\mathbb{E}_\pi [a_\pi(s,a)] = 0$

Duelling Networks [Wang et al., 2016]

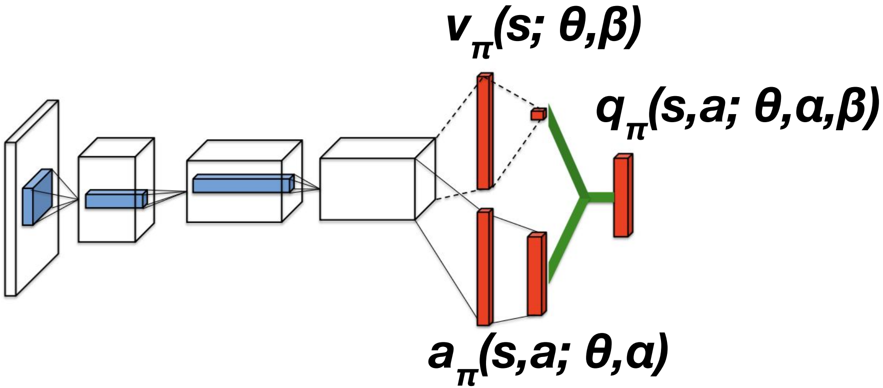


Duelling Networks [Wang et al., 2016]

- “Intuitively, the dueling architecture can learn which states are (or are not) valuable, without having to learn the effect of each action for each state. This is particularly useful in states where its actions do not affect the environment in any relevant way.”
- “The key insight [...] is that for many states, it is unnecessary to estimate the value of each action choice.”



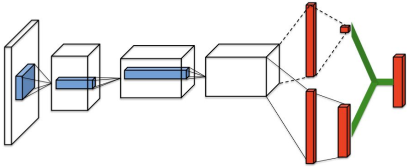
Duelling Networks [Wang et al., 2016]


$v_\pi(s; \theta,\beta)$
$q_\pi(s,a; \theta,\alpha,\beta)$
$a_\pi(s,a; \theta,\alpha)$

- But if we are naive about it, this equation is unidentifiable
- Instead, they force the advantage function estimator to have zero advantage at the chosen action. They consider implementing

$$q_\pi(s,a; \theta,\alpha,\beta) = v_\pi(s; \theta,\beta) + \left(a_\pi(s,a; \theta,\alpha) - \max_{a'} a_\pi(s,a'; \theta,\alpha)\right)$$

Duelling Networks [Wang et al., 2016]



- The correct thing to do would be

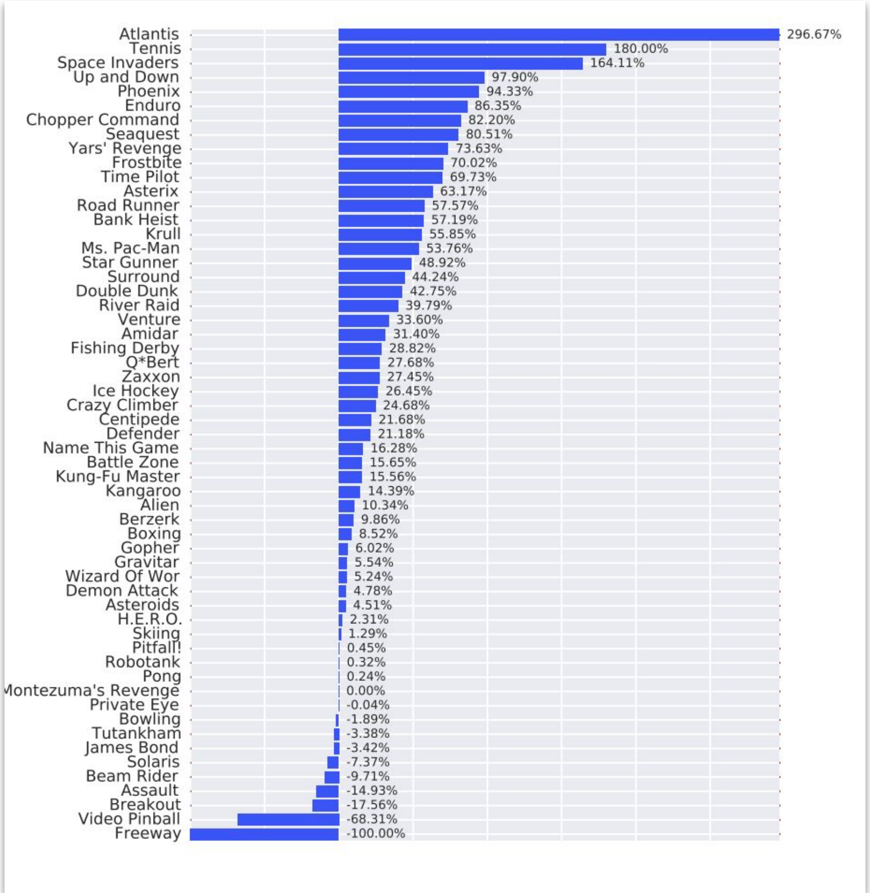
$$q_\pi(s,a;\theta,\alpha,\beta) = v_\pi(s;\theta,\beta) + \Big(a_\pi(s,a;\theta,\alpha) - \max_{a'} a_\pi(s,a';\theta,\alpha)\Big)$$

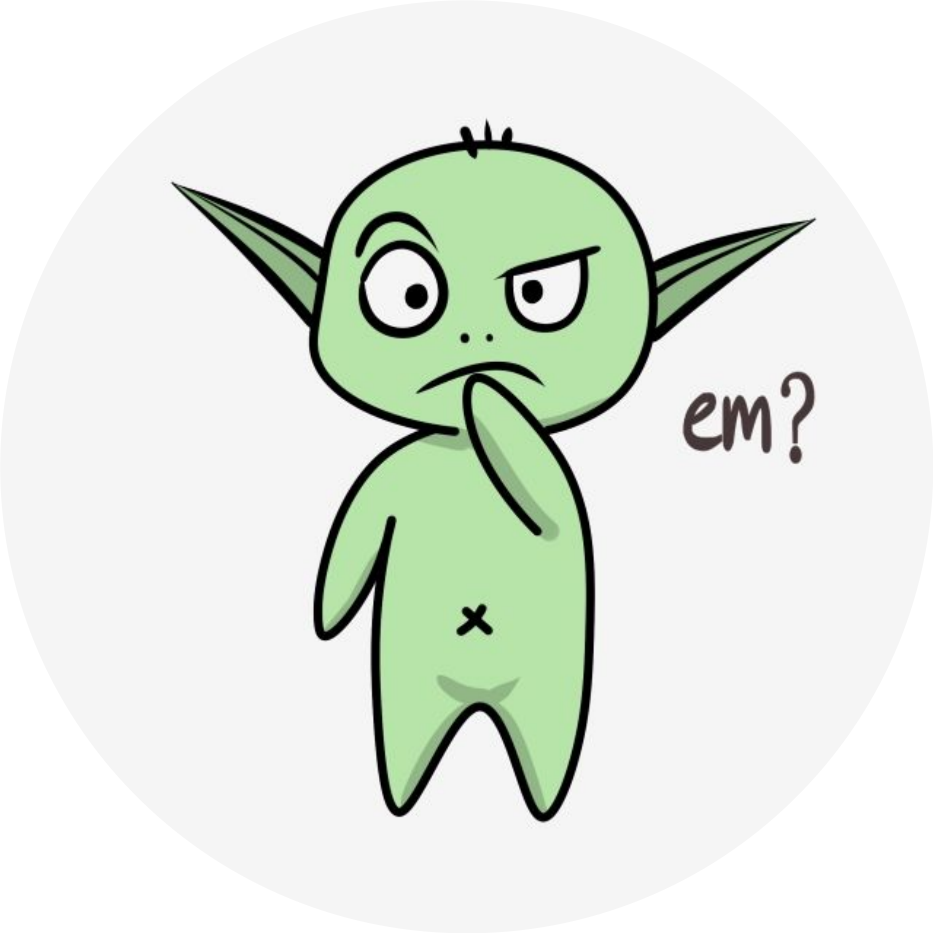
- In practice, for stability, they implement

$$q_\pi(s,a;\theta,\alpha,\beta) = v_\pi(s;\theta,\beta) + \Big(a_\pi(s,a;\theta,\alpha) - (1/|\mathscr{A}|) \sum_{a'} a_\pi(s,a';\theta,\alpha)\Big)$$

It Works [Wang et al., 2016] ¯_(ツ)_/¯

- “We follow closely the setup of van Hasselt et al. (2015)”
  - Minimal action set
  - 30 no-ops performance measure
- “Since both the advantage and the value stream propagate gradients to the last convolutional layer in the backward pass, we rescale the combined gradient entering the last convolutional layer by 1/√2. This simple heuristic mildly increases stability. In addition, we **clip the gradients** to have their norm less than or equal to 10. This clipping is not standard practice in deep RL....”





https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

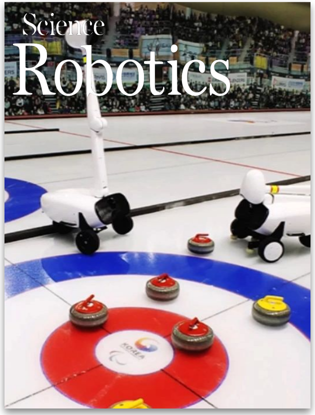
Reinforcement Learning in the “Real-World”

nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
LEARNING CURVE
Self-taught AI software attains human-level performance in video games
PAGES 486 & 529
SHARE DATA IN OUTBREAKS
A GIANT IN THE EARLY UNIVERSE
TELEPORTATION FOR TWO

[Mnih et al., 2020]

The international journal of science / 14 November 2019
nature
GAME PLAN
AI program learns to play StarCraft II to Grandmaster level
Pharmaceuticals
3D printing
Cancer imaging

[Vinyals et al., 2019]



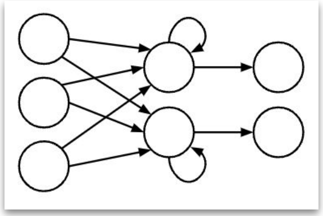
[Won et al., 2020]

[Bellemare et al., 2020]

[Degrave et al., 2022]

Partial observability is one of main challenges we face

- In most realistic sequential decision-making tasks, the agent is faced with a partial observable problem
- The agent must construct an agent-state in order to succeed
  - The agent can still leverage the MDP formalism; it is not a limitation of MDPs
  - $\mathbf{x}_{t+1} \overset{\text{def}}{=} u_{\phi}(\mathbf{x}_t, a_t, \mathbf{o}_{t+1})$
- Recurrent neural networks are the most obvious things to consider here (again, a change in architecture, not* anything else) , often we use LSTMs
  - $\boldsymbol{h}^{(t)} = f(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \boldsymbol{\theta})$

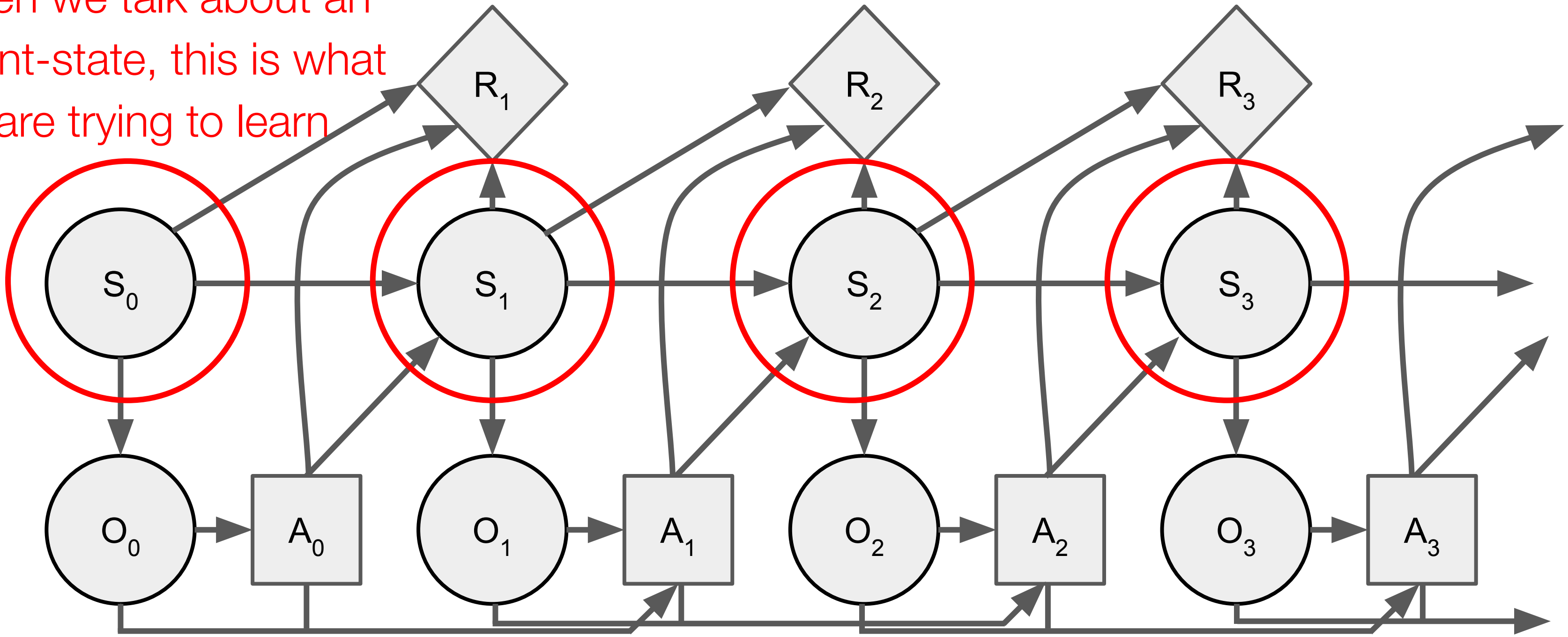


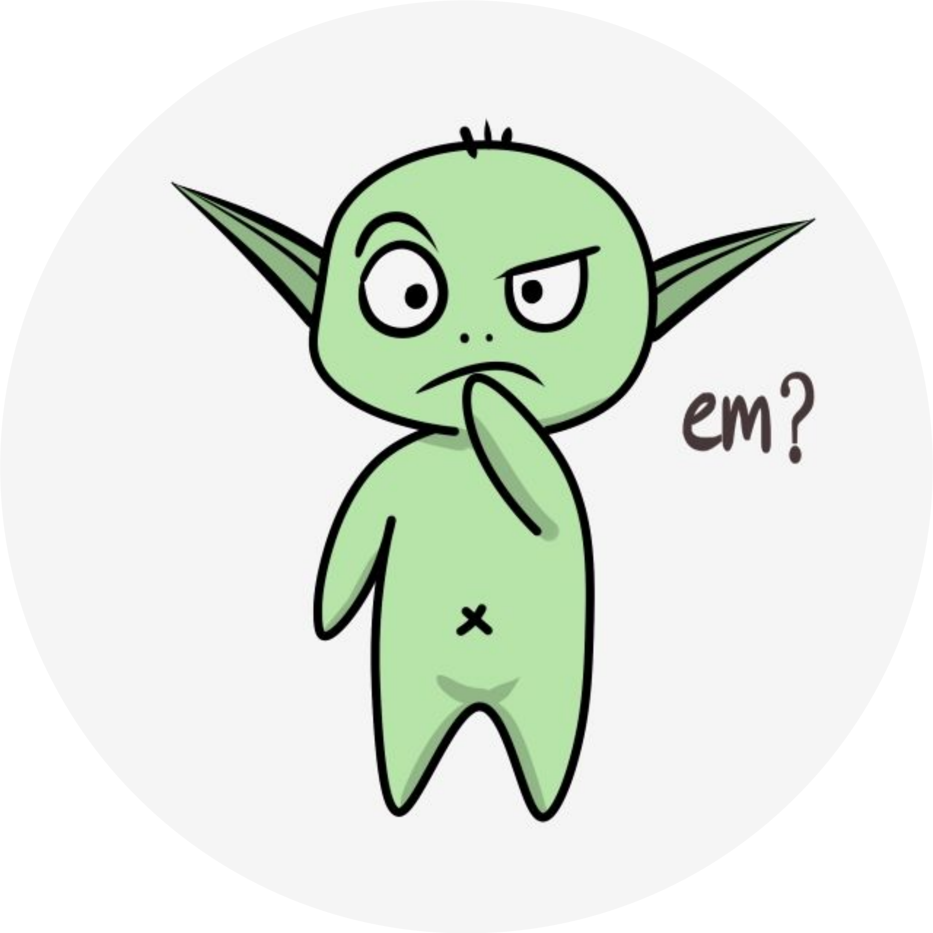
[Vinyals et al., 2019]

[Degrave et al., 2022]

Partially-Observable Markov Decision Processes

When we talk about an agent-state, this is what we are trying to learn

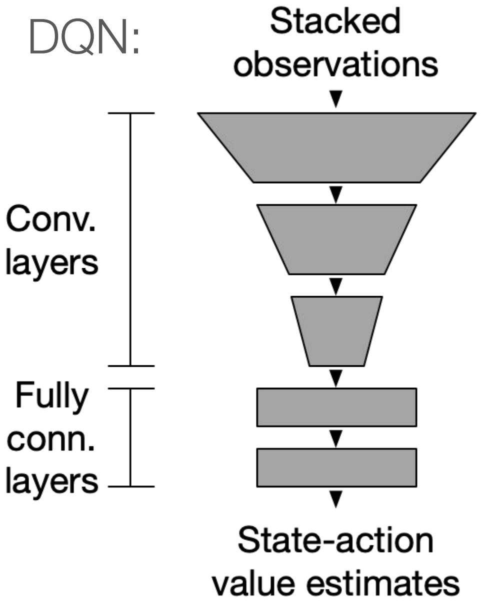




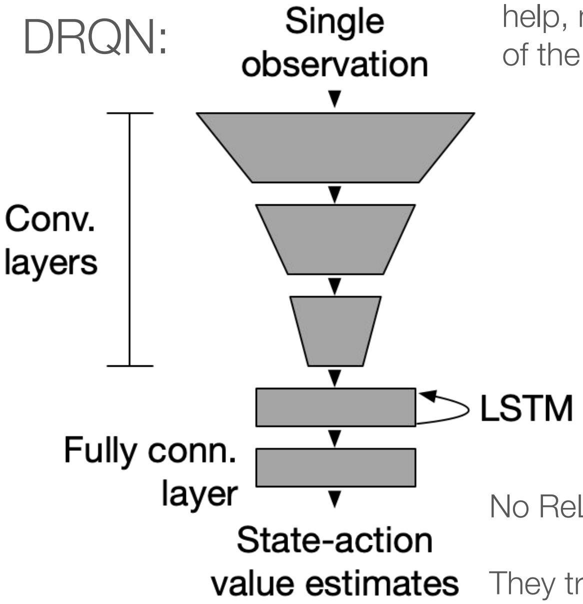
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Deep Recurrent Q-Network (DRQN) [Hausknecht and Stone, 2015]

DQN:



DRQN:



Frame stacking didn't help, maybe because of the # of params.

No ReLU after LSTM

They tried adding an LSTM between two fully conn. layers, didn't work that well

The Benefits of Using Recurrence Instead of Frame Stacking

- More general, can *learn*
- Much more flexibility on what to remember (and how long ago)
- Can better adapt “at evaluation time” if the quality of observations change

- But, training them can be quite tricky …
- … and, at least Hausknecht and Stone (2015) were not able to demonstrate performance gains

The Different Choices for Updating a Recurrent Agent

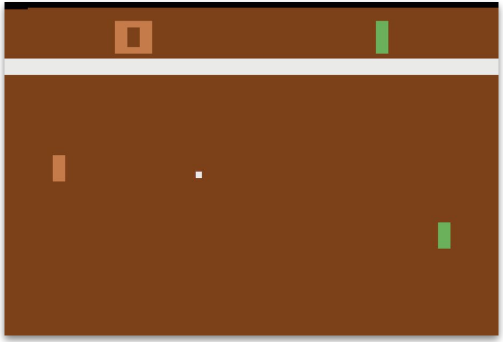
Hausknecht & Stone
eval. these two

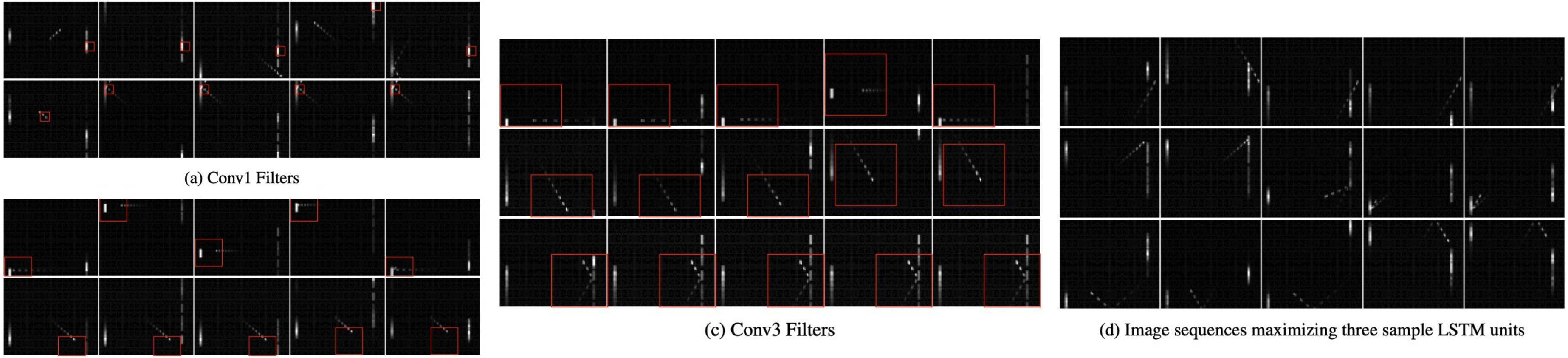
- Train agent sequentially by randomly selecting an episode and going through the episode's transitions serially
- Randomly sample transitions and do updates for *k* steps
  - Reset the state of the LSTM by zero-ing it out (more on this later)
  - Save the state of the LSTM on the experience replay buffer

    (can suffer from *representational drift,* leading to *recurrent state staleness)*
- Allow for a "burn-in" period where some observations are used just to produce a start state for the LSTM (to be discussed later)
- Do multiple sequential updates in parallel (to be discussed later)

Flickering Pong [Hausknecht and Stone, 2015]

- Screen is fully obscured with 0.5 chance
- DRQN trained with backpropagation through time for 10 the last ten time steps is able to do it





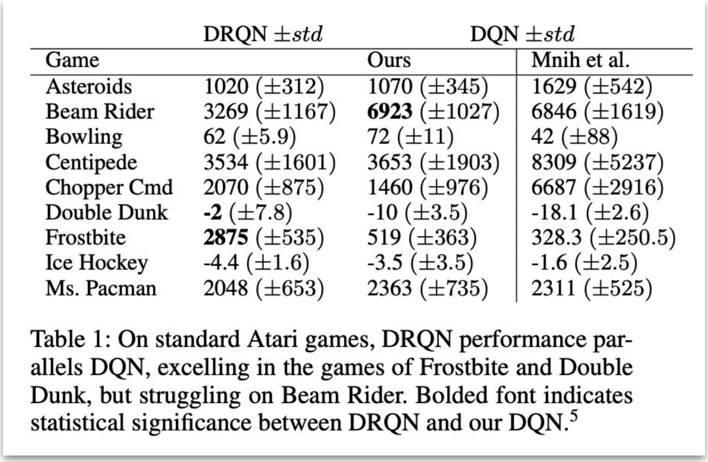
(a) Conv1 Filters

(b) Conv2 Filters

(c) Conv3 Filters

(d) Image sequences maximizing three sample LSTM units

Standard Atari 2600 Games [Hausknecht and Stone, 2015]



	DRQN ±*std*	DQN ±*std*	
Game		Ours	Mnih et al.
Asteroids	1020 (±312)	1070 (±345)	1629 (±542)
Beam Rider	3269 (±1167)	**6923** (±1027)	6846 (±1619)
Bowling	62 (±5.9)	72 (±11)	42 (±88)
Centipede	3534 (±1601)	3653 (±1903)	8309 (±5237)
Chopper Cmd	2070 (±875)	1460 (±976)	6687 (±2916)
Double Dunk	**-2** (±7.8)	-10 (±3.5)	-18.1 (±2.6)
Frostbite	**2875** (±535)	519 (±363)	328.3 (±250.5)
Ice Hockey	-4.4 (±1.6)	-3.5 (±3.5)	-1.6 (±2.5)
Ms. Pacman	2048 (±653)	2363 (±735)	2311 (±525)

Table 1: On standard Atari games, DRQN performance parallels DQN, excelling in the games of Frostbite and Double Dunk, but struggling on Beam Rider. Bolded font indicates statistical significance between DRQN and our DQN.[5]

"Policies were evaluated every 50,000 iterations by playing 10 episodes and averaging the resulting scores."

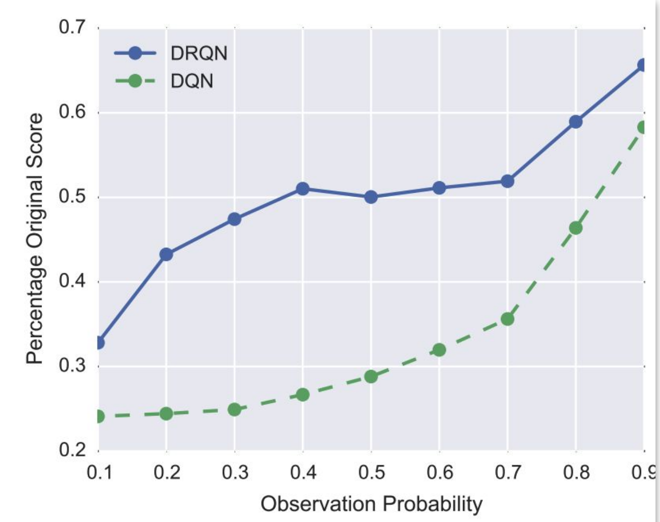
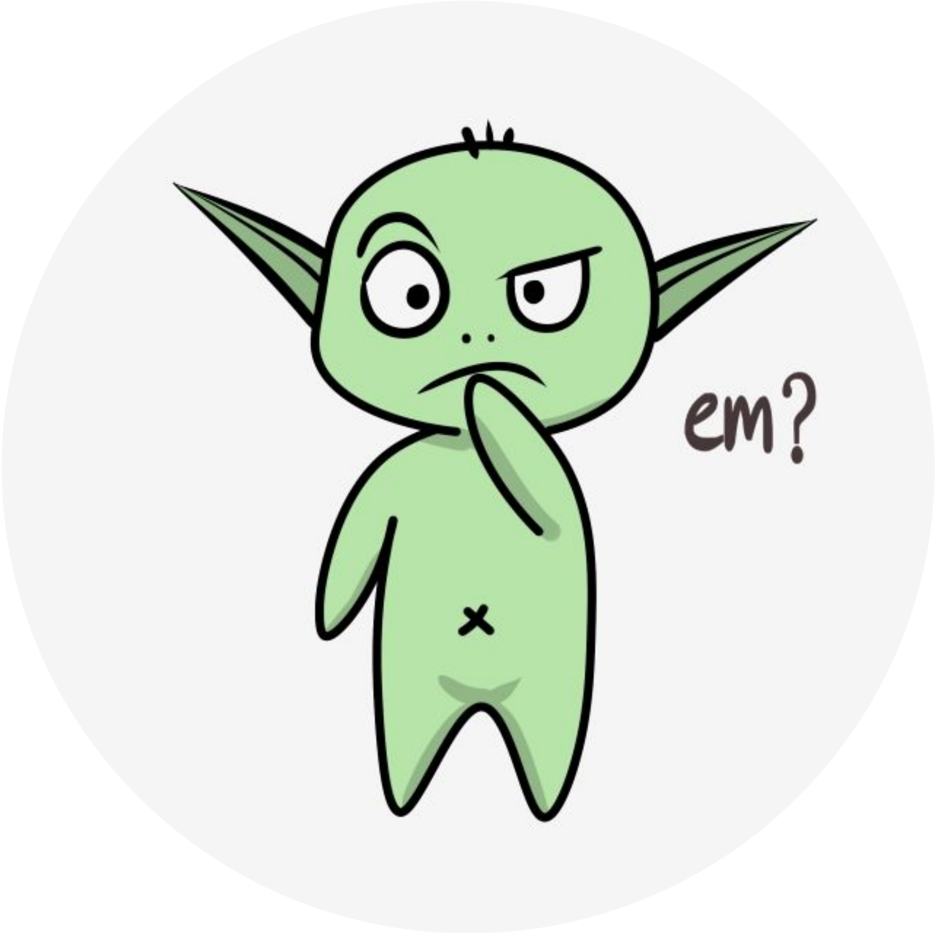


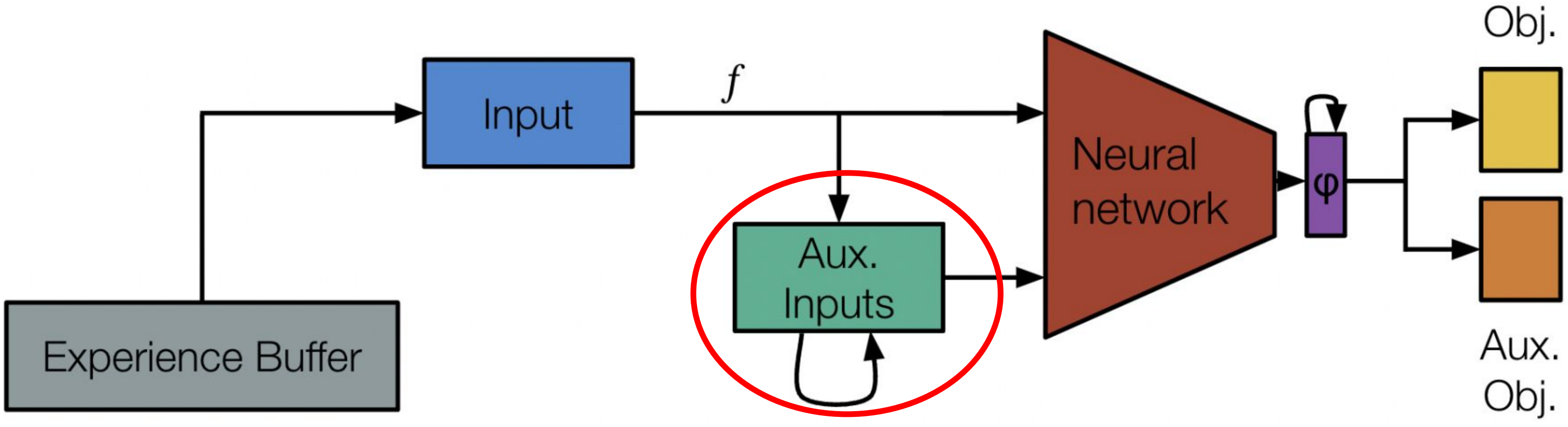
Figure 5: When trained on normal games (MDPs) and then evaluated on flickering games (POMDPs), DRQN's performance degrades more gracefully than DQN's. Each data point shows the average percentage of the original game score over all 9 games in Table 1.



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Auxiliary Inputs

- People often use a recurrent neural network to learn an agent-state, but another common approach is to use environment-specific functions to aid the agent's inputs for history summarization



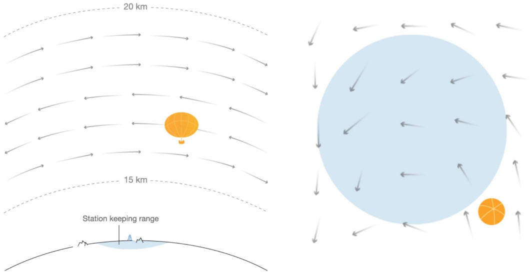
(Tao et al., 2023)

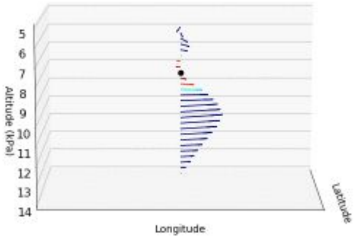
A couple of examples

Navigating Balloons in the Stratosphere

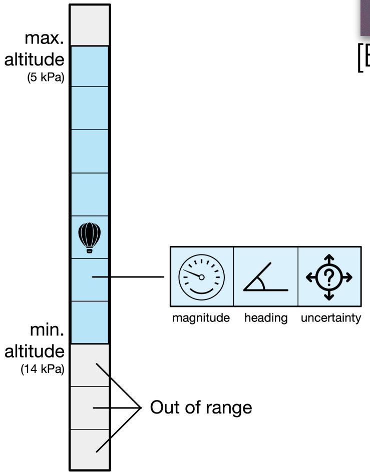
[Bellemare et al., 2020]

Observations





Forecast + measurements + Gaussian process = wind column



max. altitude (5 kPa)

min. altitude (14 kPa)

magnitude

heading

uncertainty

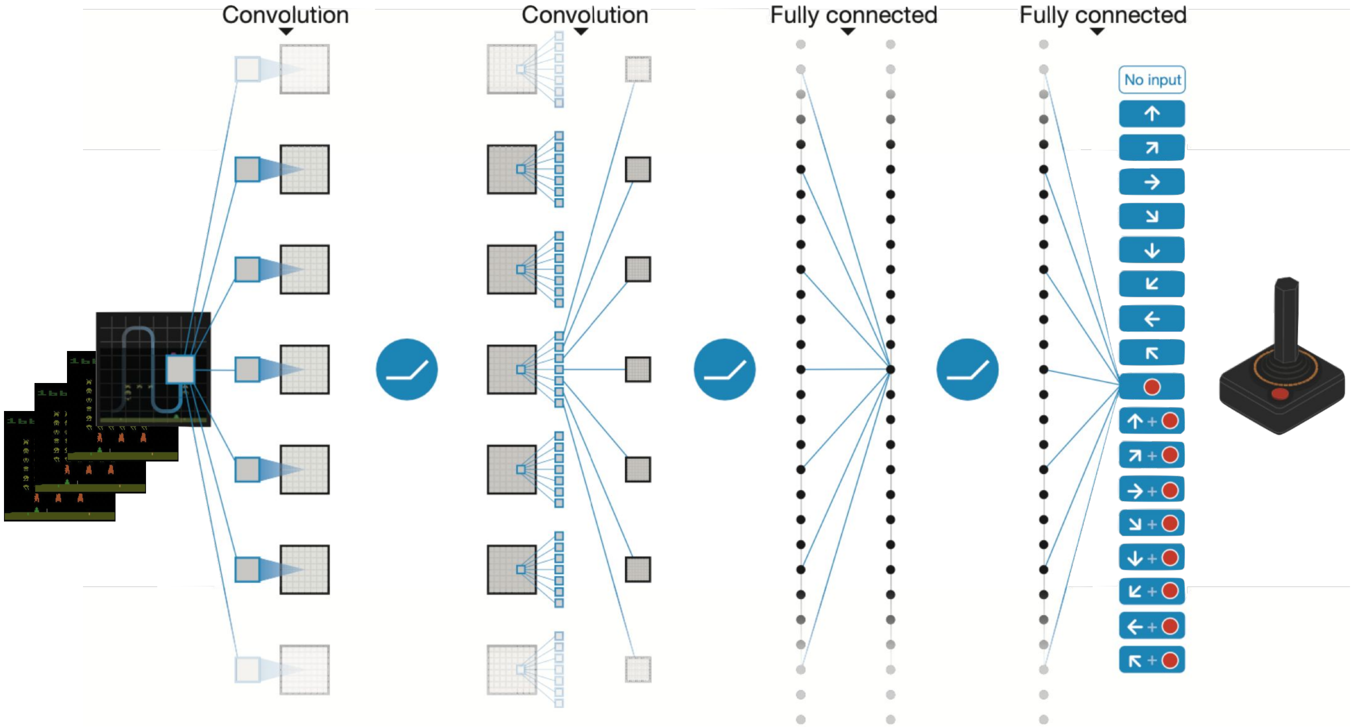
Out of range

+16 ambient variables

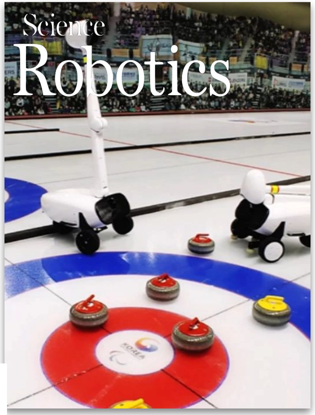
Playing Atari 2600 Games

nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
LEARNING CURVE
Self-taught AI software attains human-level performance in video games
SHARE DATA IN OUTBREAKS
A GIANT IN THE EARLY UNIVERSE
TELEPORTATION FOR TWO

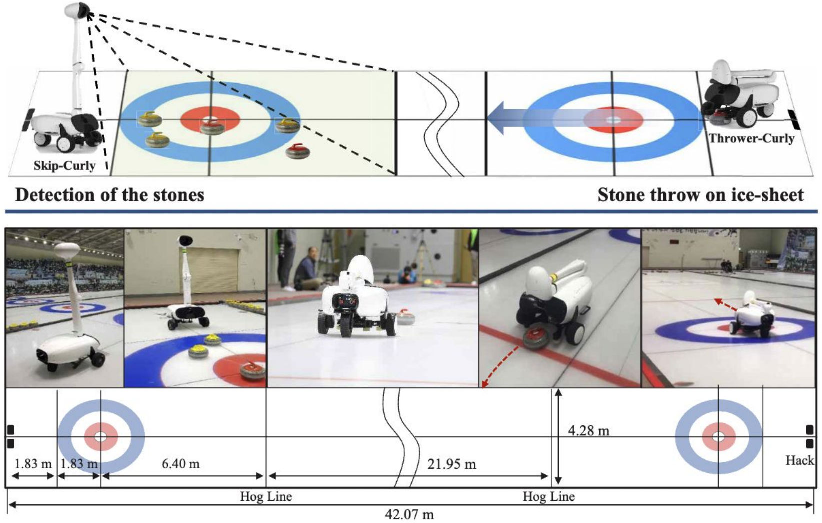
[Mnih et al., 2015]

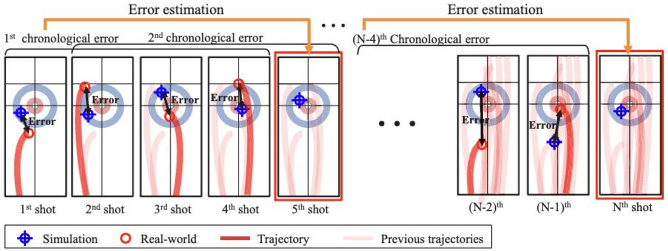


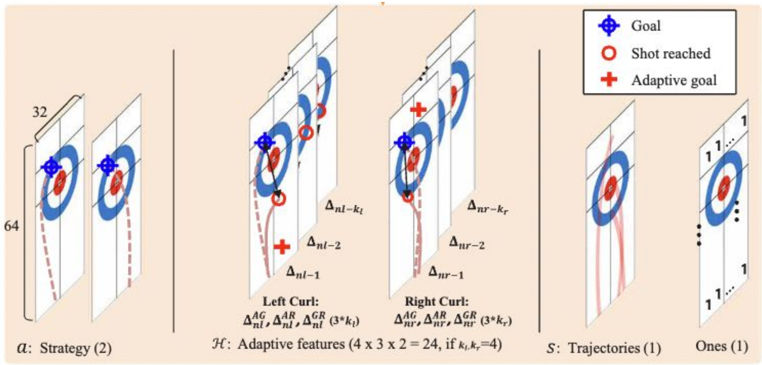
Playing Real Curling Against Olympians

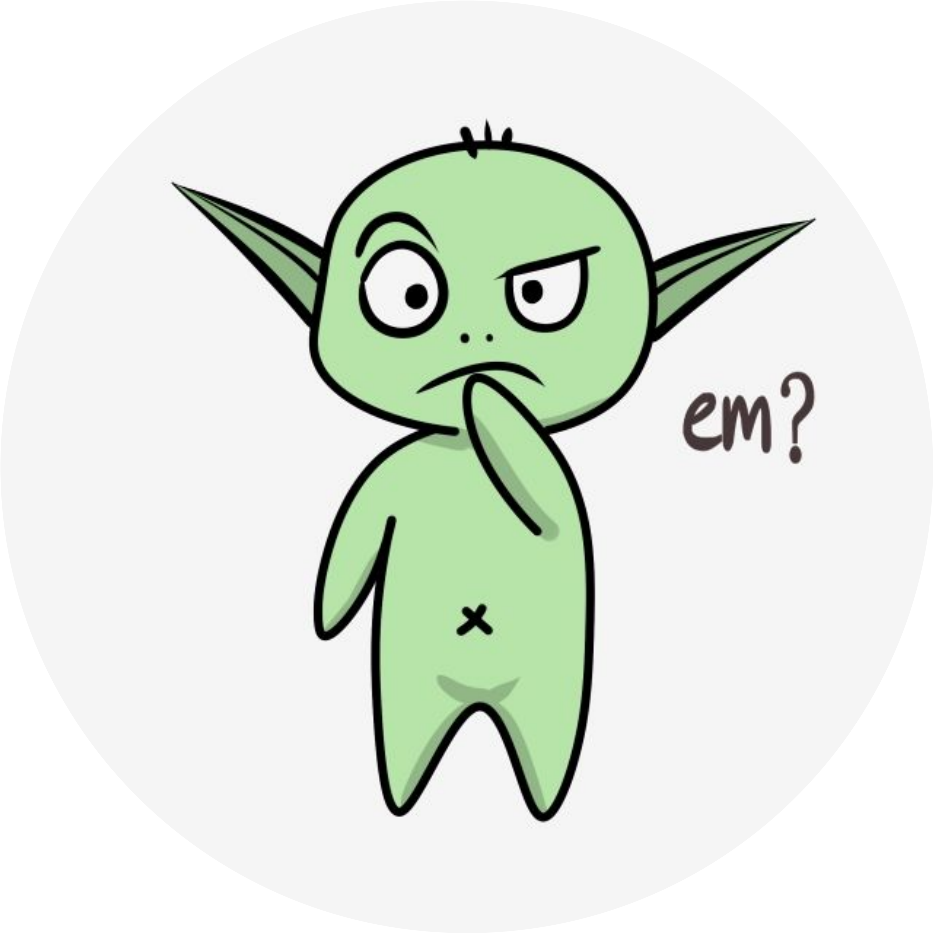


[Won et al., 2020]









https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Auxiliary Inputs

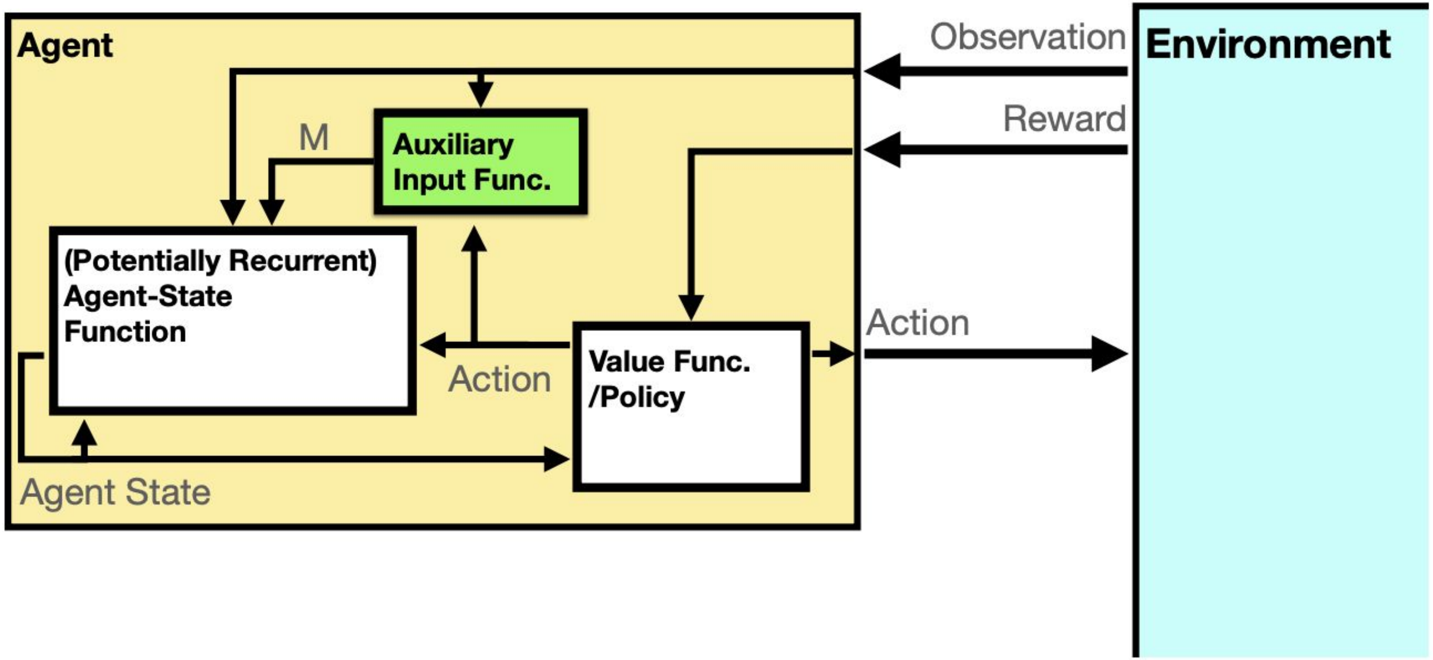
“additional inputs, beyond environment observations, that incorporate or model information regarding the past, present and/or future of a reinforcement learning agent.”

– Tao et al. (2023)

Auxiliary Inputs [Tao et al., 2023]

- “Allow an agent to discriminate between observations that would otherwise be aliased”
- “Allow for a smooth interpolation in the value function between different states”
- Auxiliary inputs do not necessarily summarize only the past, they can also try to summarize the present or the future
  - We can summarize entire trajectories, Tt = $\{\mathbf{o}_0, a_0, \ldots, \mathbf{o}_t, a_t, \mathbf{O}_{t+1}, \ldots, \mathbf{O}_L\}$
- The key idea here is that many methods incorporate auxiliary inputs into the learning process, such that $\mathbf{x}_{t+1} \overset{\text{def}}{=} u_\phi(\mathbf{x}_t, a_t, \mathbf{o}_{t+1}, \mathbf{M}_{t+1})$, where $\mathbf{M}\text{:}\ \mathscr{T} \rightarrow \mathbb{R}^N$.

Auxiliary Inputs [Tao et al., 2023]

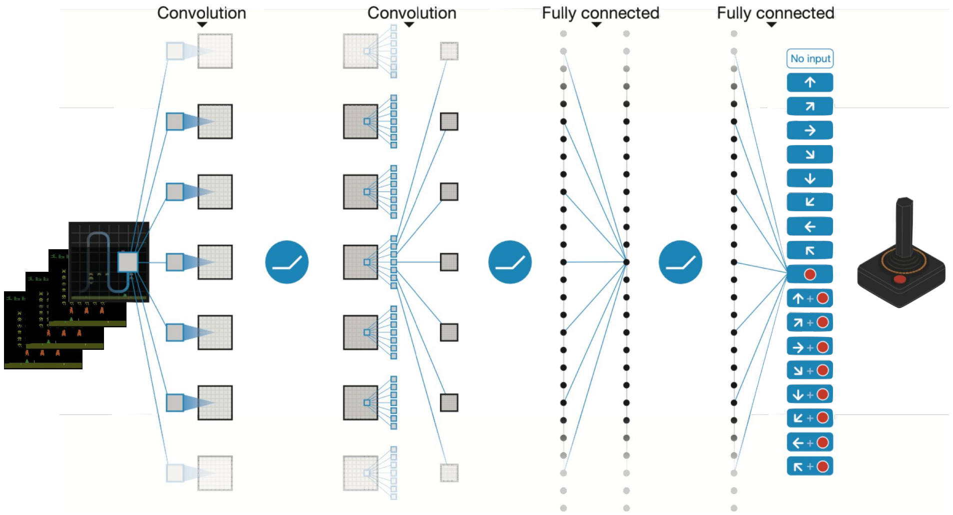


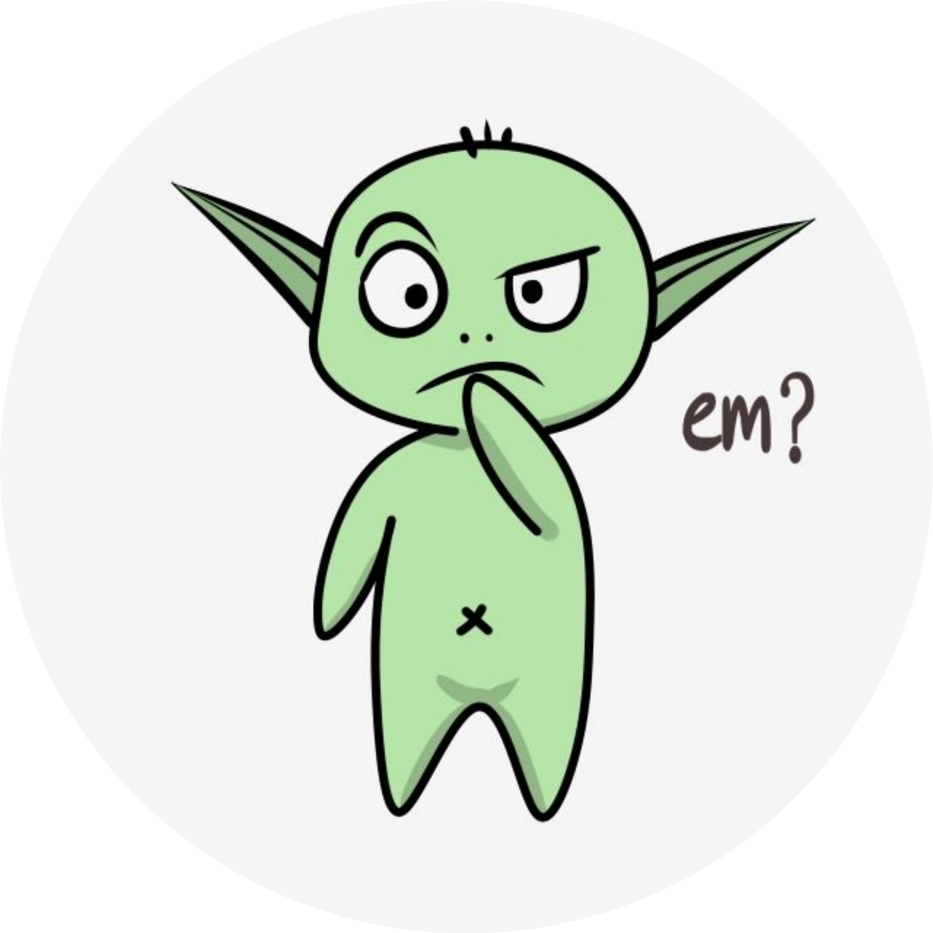
Auxiliary Inputs [Tao et al., 2023]

- We can see auxiliary inputs in terms of *resolution* and *depth:*
  - *Resolution: "the extent that information regarding individual observation-action pairs are preserved by the auxiliary inputs"*
  - *Depth: "the temporal length these auxiliary inputs retain information with regards to the agent's trajectory"*

Example: Frame Stacking as Auxiliary Input

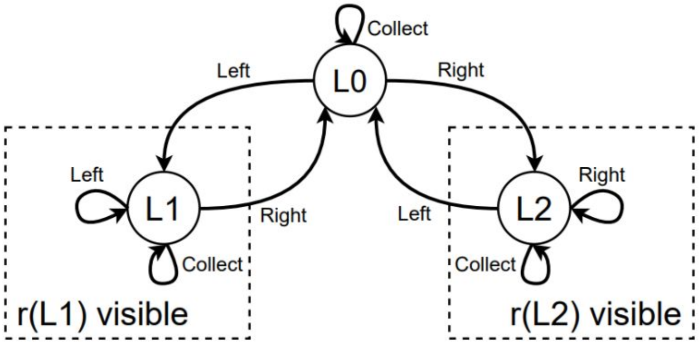
- $\mathbf{M}h(\{\mathbf{o}_0, a_0, \ldots, \mathbf{o}_t, a_t\}) \overset{\text{def}}{=} \mathbf{o}_{t-3} \,||\, \mathbf{o}_{t-2} \,||\, \mathbf{o}_{t-1}$, where || denotes concatenation
- Frame stacking has low depth and high resolution, for example





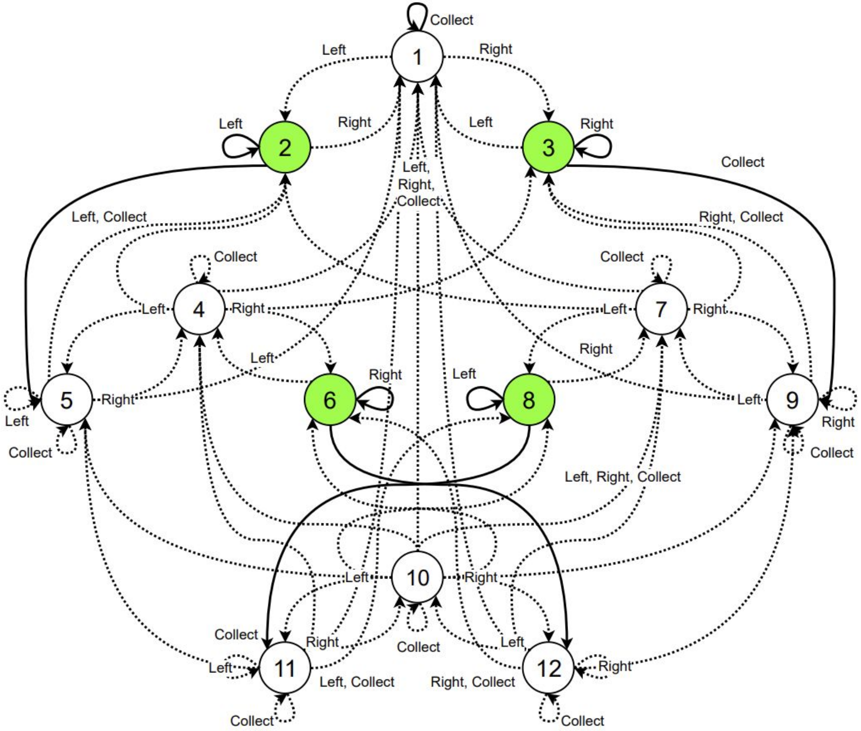
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Lobster Environment [Tao et al., 2023]



$$\boldsymbol{o}_t \doteq \begin{bmatrix} \text{0. Is the agent in location 0?} \\ \text{1. Is the agent in location 1?} \\ \text{2. Is the agent in location 2?} \\ \text{3. Is the reward in location 1 observable and missing?} \\ \text{4. Is the reward in location 1 observable and present?} \\ \text{5. Is the reward in location 1 unobservable?} \\ \text{6. Is the reward in location 2 observable and missing?} \\ \text{7. Is the reward in location 2 observable and present?} \\ \text{8. Is the reward in location 2 unobservable?} \end{bmatrix}$$

Lobster Environment [Tao et al., 2023]



Different Types of Auxiliary Inputs [Tao et al., 2023]

- Decaying traces
  - High depth, low resolution

$$\boldsymbol{M}_t \doteq \sum_{\tau=0}^{t} \lambda^{t-\tau} g(\boldsymbol{o}_\tau, a_\tau) \qquad \boldsymbol{M}_t \doteq \lambda \boldsymbol{M}_{t-1} + g(\boldsymbol{o}_\tau, a_\tau)$$

- Uncertainty
  - Variable depth (per #particles), low resolution

$$\boldsymbol{M}_{t+1} \doteq \sum_{j=0}^{k} \boldsymbol{w}_{t+1}[j] \odot \mathbb{1}_{[\hat{\boldsymbol{s}}_t[j]]}$$

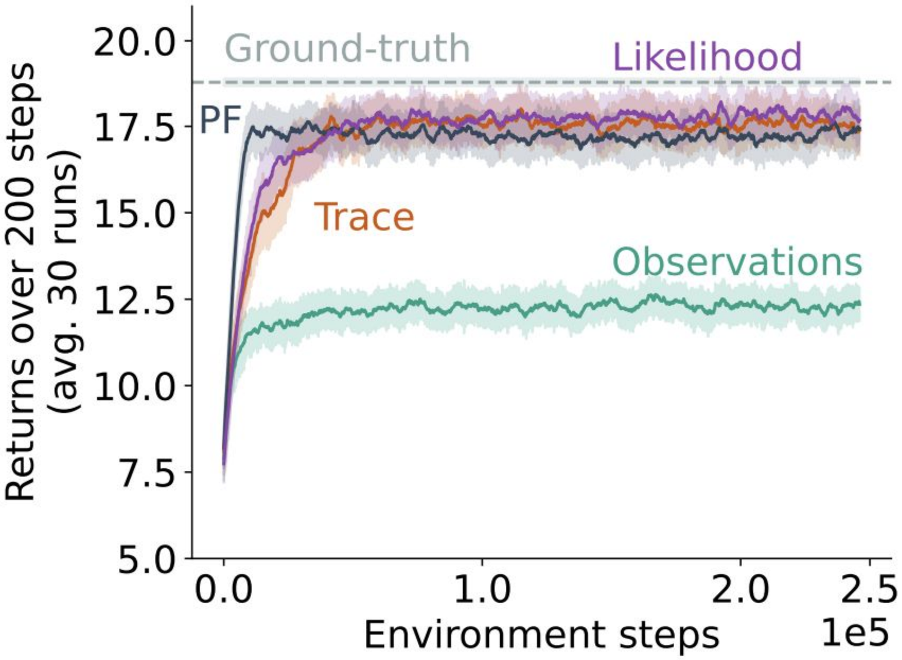
Particle Filtering

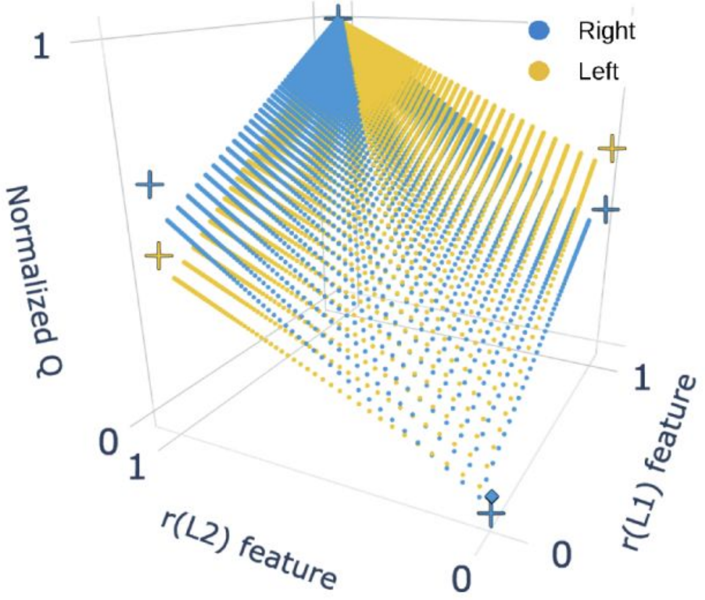
$$\hat{\boldsymbol{s}}_{t+1}[j] \sim p(\cdot \mid \hat{\boldsymbol{s}}_t[j], A_t)$$

$$\overline{\boldsymbol{w}}_{t+1}[j] \doteq P\{\boldsymbol{O}_t = \boldsymbol{o}_t \mid S_t = \hat{\boldsymbol{s}}_t[j]\} \cdot \boldsymbol{w}_t[j]$$

- GVFs (Sutton et al., 2011)
  - High depth, low resolution

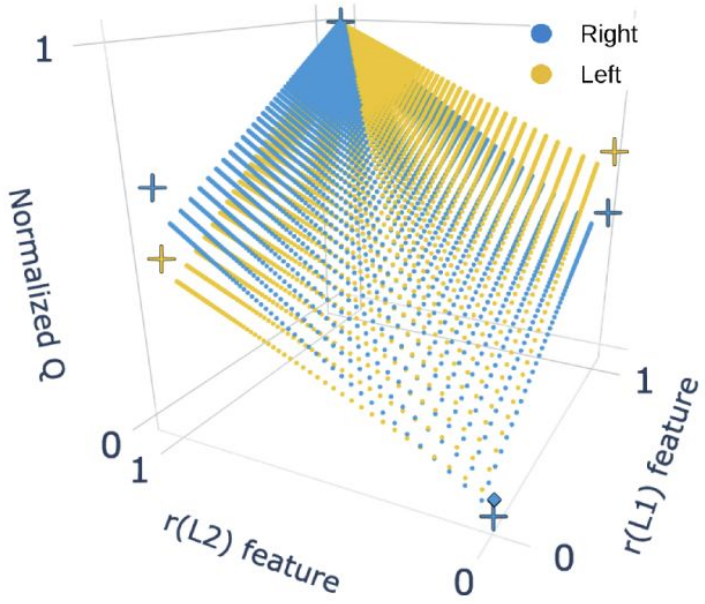
Auxiliary Inputs (LFA) [Tao et al., 2023]

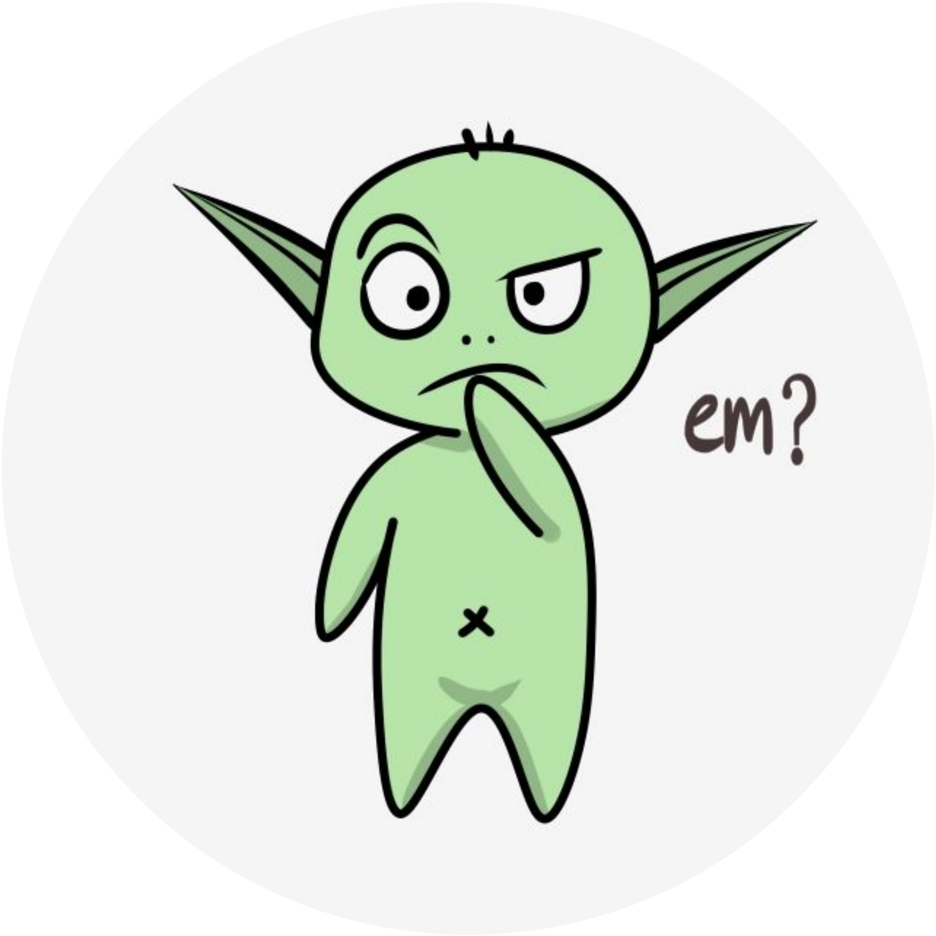




Auxiliary Inputs [Tao et al., 2023]

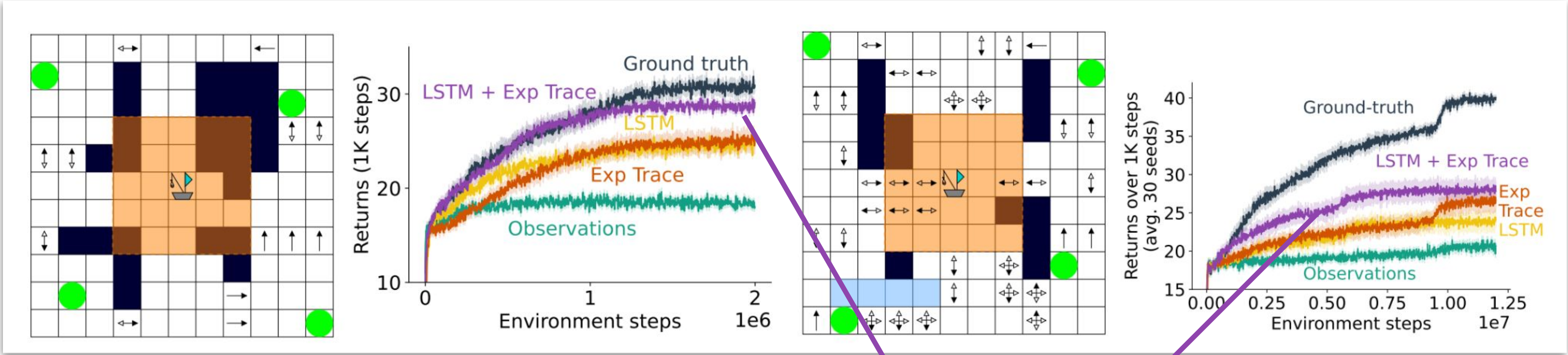
- “Allow an agent to discriminate between observations that would otherwise be aliased”
- “Allow for a smooth interpolation in the value function between different states”



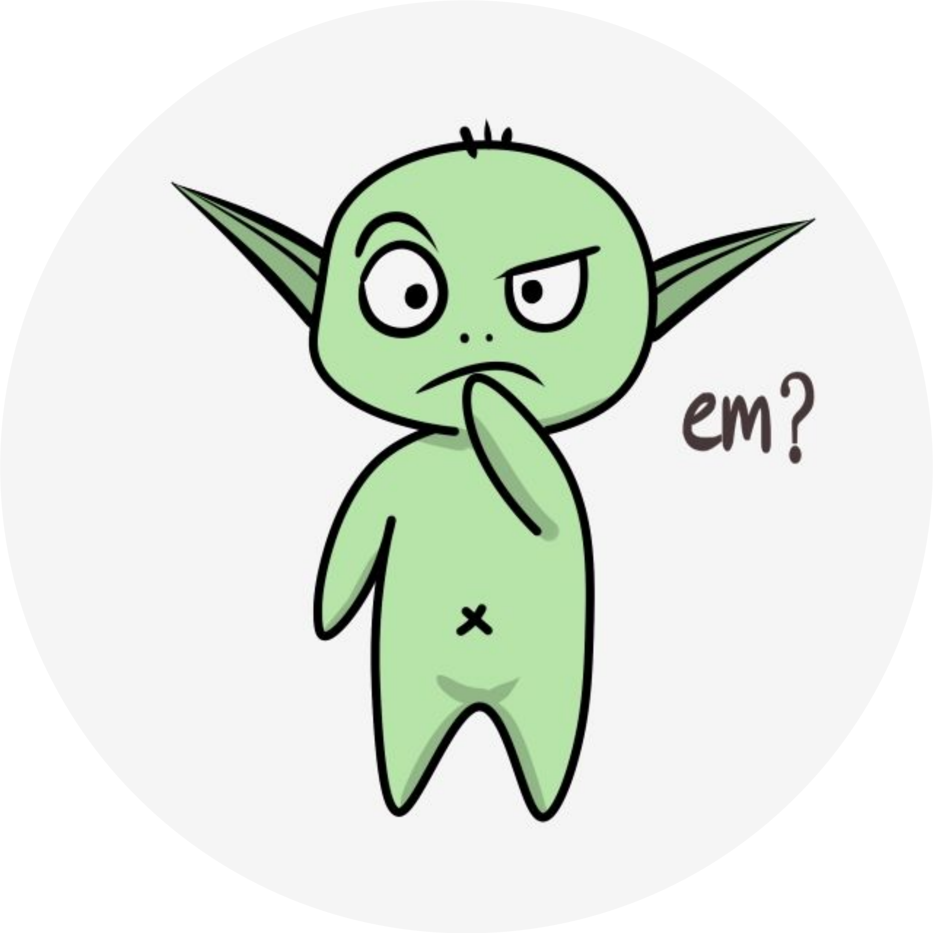


https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Auxiliary Inputs – Deep RL [Tao et al., 2023]

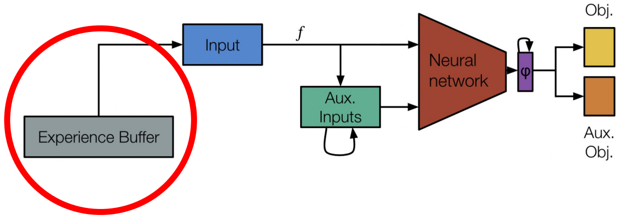


Auxiliary Inputs aren't trying to get rid of RNNs



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Next class



- What I plan to do:
  - Wrap up the discussion we started today
  - Talk about strategies around *experience replay buffers*
- What I recommend YOU to do for next class:
  - Read

    - Tom Schaul, John Quan, Ioannis Antonoglou, David Silver: Prioritized Experience Replay. ICLR 2016
  - *- William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, Will Dabney: Revisiting Fundamentals of Experience Replay. ICML 2020*