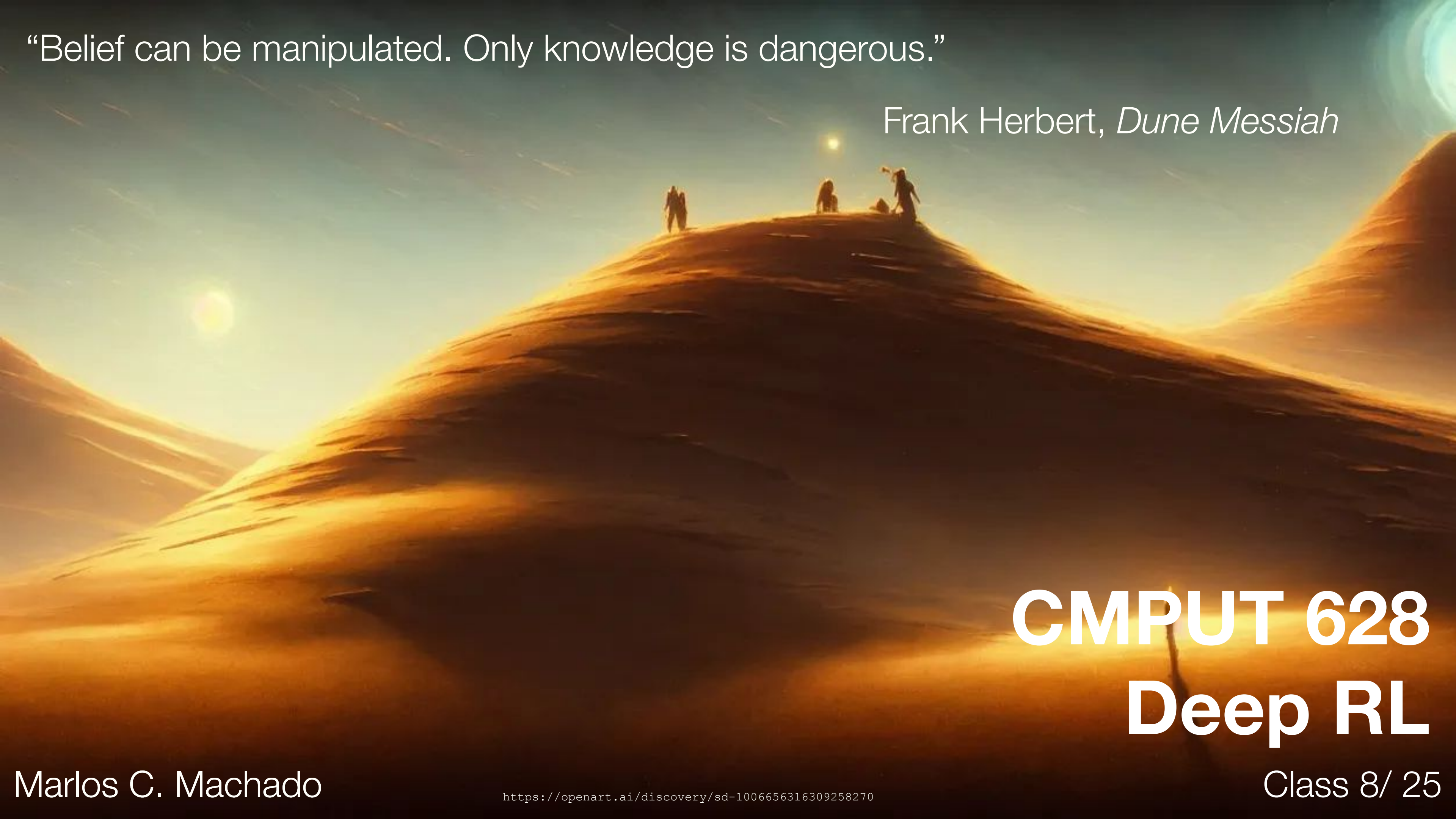


"Belief can be manipulated. Only knowledge is dangerous."

Frank Herbert, *Dune Messiah*

CMPUT 628
Deep RL

Marlos C. Machado

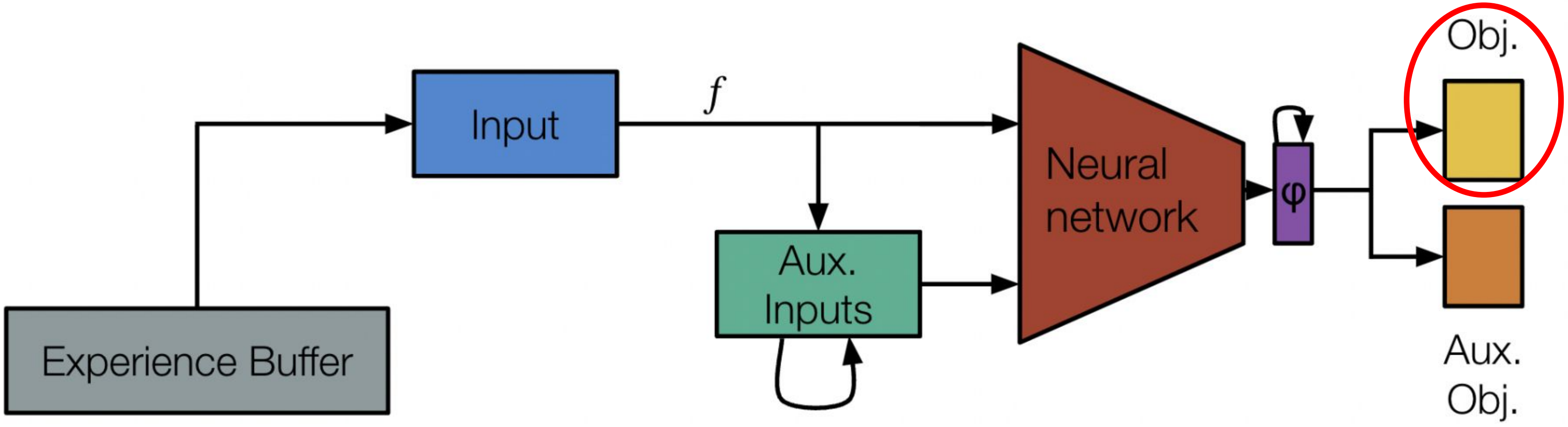
https://openart.ai/discovery/sd-1006656316309258270

Class 8/ 25

Reminders & Notes

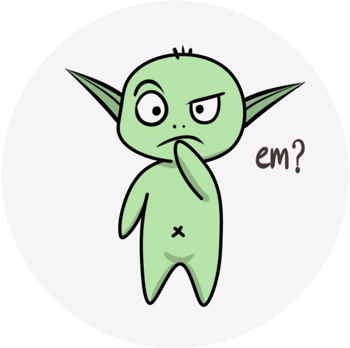
- Assignment 2 is out and you should have already started it :-)
- I already marked Assignment 1 and the result is on eClass
  If something looks wrong to you, we should talk
- I will release instructions about seminar and paper review
  during the reading week (Feb 18 – Feb 21)
- Lecture notes v0.2 are available. I'm working on v0.3
  Feedback is more than welcome
- I will be travelling on March 3rd (Monday), 2025
  A. Rupam Mahmood will give a guest lecture on streaming deep RL

Plan



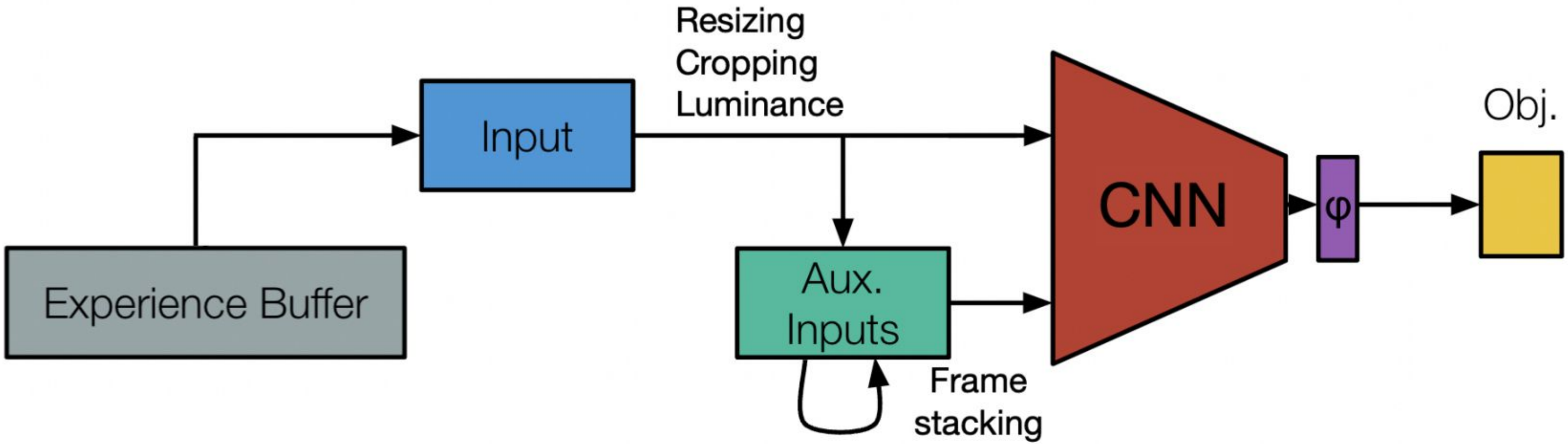
Exploring different choices of objective functions, specifically double learning to address maximization bias

Please, interrupt me at any time!



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

Last class: DQN



DQN's objective function

$$\mathcal{L}^{\text{DQN}} = \mathbb{E}_{(o,a,r,o')\sim U(\mathcal{D})}\Big[\Big(R_{t+1} + \gamma \max_{a'\in\mathcal{A}} Q(O_{t+1}, a'; \boldsymbol{\theta}^{-}) - Q(O_t, A_t; \boldsymbol{\theta}_t)\Big)^2\Big]$$

or , equivalently,

$$Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}^{-}) = R_{t+1} + \gamma \max_{a'\in\mathcal{A}} Q\big(O_{t+1}, a'; \boldsymbol{\theta}^{-}\big)$$

$$\mathcal{L}^{\text{DQN}} = \mathbb{E}_{(o,a,r,o')\sim U(\mathcal{D})}\Big[\Big(Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}^{-}) - Q(O_t, A_t; \boldsymbol{\theta}_t)\Big)^2\Big].$$

Maximization bias

- Q-learning uses a maximization to get its target policies.
- *Maximization bias*: A maximum over estimated values is used implicitly as an estimate of the maximum value, which can lead to a significant positive bias.

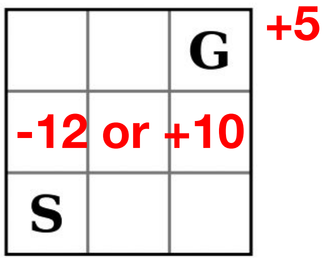
Example: Maximization bias

Actions	*Action 1*	*Action 2*	*Action 3*	*Action 4*	*Action 5*
Outcome	3, 1, 3, 1, 3	6, 3, 3, 2, 4	7, 1, 4, 6, 7	5, 3, 2, 5, 1	1, 10, 6, 2, 4
Average	2.2	3	5	3.2	4.6

3, 8, 2, 4, 8, 4, 2, 6, 4, 1 → Avg = 4.2

Q-learning overestimates [van Hasselt 2010]



Double learning

- The issue is that we use the same samples to determine the maximizing action and to estimate its value. One should not be using the maximum of the estimates as an estimate of the maximum of true values.
- In Bandits:
  - Split the data, learn $Q_1(a)$ and $Q_2(a)$ to estimate $q(a)$.
  - Choose actions according to one estimate and get estimate from the other:
    $A^* = \text{argmax}_a\ Q_1(a)$ $\qquad Q_2(A^*) = Q_2(\text{argmax}_a\ Q_1(a))$
  - This leads to unbiased estimates, that is: $\mathbb{E}[Q_2(A^*)] = q(A^*)$

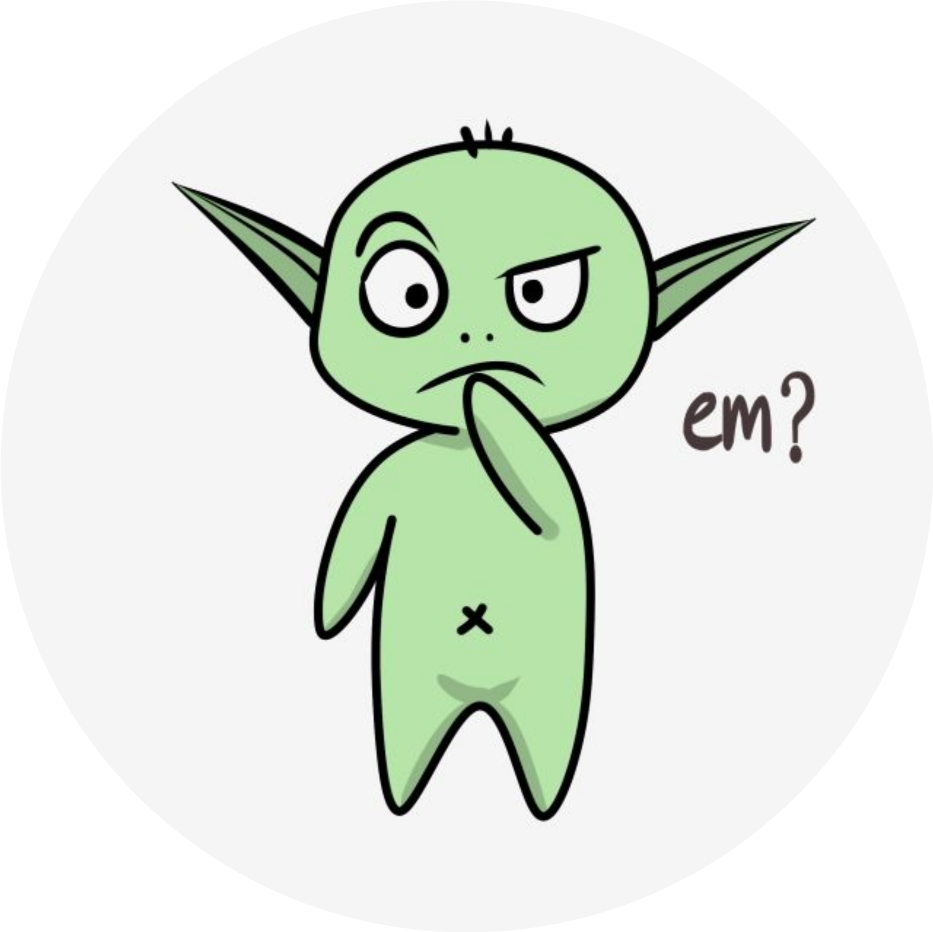
$$Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma Q_2\big(S_{t+1}, \arg\max_a Q_1(S_{t+1}, a)\big) - Q_1(S_t, A_t) \Big]$$

Double Q-Learning [van Hasselt 2010]

Double Q-learning, for estimating $Q_1 \approx Q_2 \approx q_*$

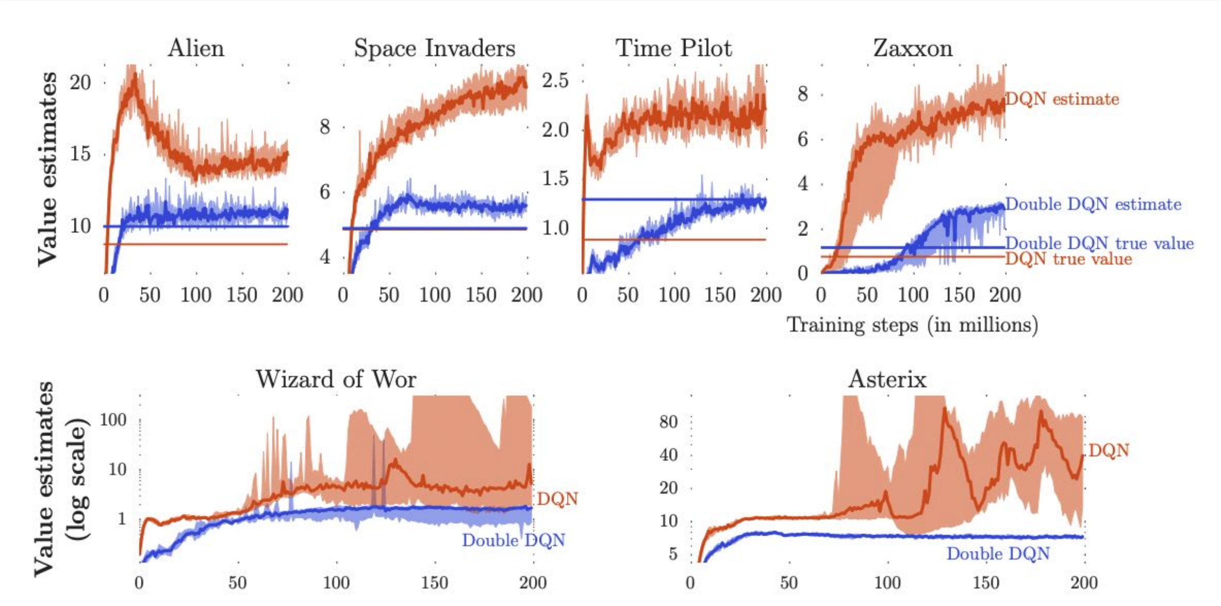
Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q_1(s, a)$ and $Q_2(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, such that $Q(terminal, \cdot) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        Choose A from S using the policy ε-greedy in $Q_1 + Q_2$
        Take action A, observe R, S'
        With 0.5 probabilility:
            $Q_1(S, A) \leftarrow Q_1(S, A) + \alpha\Big(R + \gamma Q_2\big(S', \arg\max_a Q_1(S', a)\big) - Q_1(S, A)\Big)$
        else:
            $Q_2(S, A) \leftarrow Q_2(S, A) + \alpha\Big(R + \gamma Q_1\big(S', \arg\max_a Q_2(S', a)\big) - Q_2(S, A)\Big)$
        $S \leftarrow S'$
    until S is terminal



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

Do we observe maximization bias in DQN? [van Hasselt, Guez, & Silver 2016]



* "6 different random seeds with the hyper-parameters employed by Mnih et al. (2015). The darker line shows the median over seeds and we average the two extreme values to obtain the shaded area (i.e., 10% and 90% quantiles with linear interpolation)."

Now what?

$$Y_t^{\mathrm{Q}} = R_{t+1} + \gamma Q(S_{t+1}, \operatorname*{argmax}_a Q(S_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}_t)$$

$$Y_t^{\mathrm{DoubleQ}} \equiv R_{t+1} + \gamma Q(S_{t+1}, \operatorname*{argmax}_a Q(S_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}'_t)$$

Double DQN [van Hasselt, Guez, & Silver 2016]

- Main idea: use the target network as a second value function
- Update is the same as in DQN, but with a different target:

$$Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}_t) = R_{t+1} + \gamma Q\Big(O_{t+1}, \arg\max_{a \in \mathcal{A}} Q(O_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}^-\Big)$$

$$\mathcal{L}^{\text{DDQN}} = \mathbb{E}_{\tau \sim U(\mathcal{D})}\Big[\big(Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}_t) - Q(O_t, A_t; \boldsymbol{\theta}_t)\big)^2\Big]$$

- “This version of Double DQN is perhaps the minimal possible change to DQN towards Double Q-learning. The goal is to get most of the benefit of Double Q-learning, while keeping the rest of the DQN algorithm intact for a fair comparison, and with minimal computational overhead.”

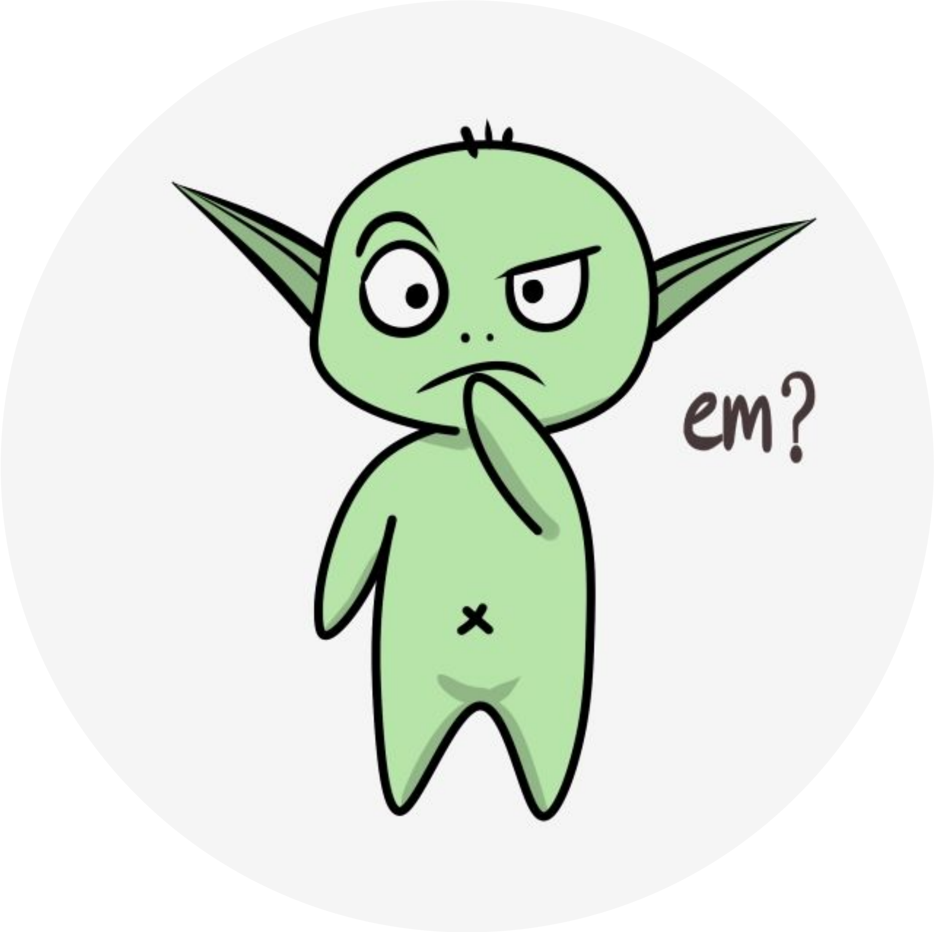
DQN vs Double DQN

$$Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}^-) = R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q\big(O_{t+1}, a'; \boldsymbol{\theta}^-\big)$$

$$\mathcal{L}^{\text{DQN}} = \mathbb{E}_{(o,a,r,o') \sim U(\mathcal{D})}\Big[\big(Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}^-) - Q(O_t, A_t; \boldsymbol{\theta}_t)\big)^2\Big].$$

$$Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}_t) = R_{t+1} + \gamma Q\big(O_{t+1}, \arg\max_{a \in \mathcal{A}} Q(O_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}^-\big)$$

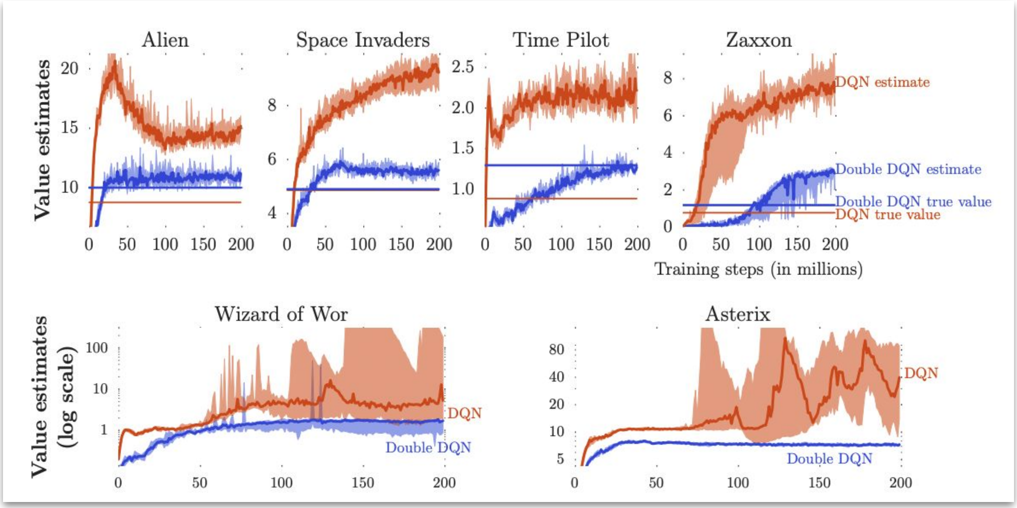
$$\mathcal{L}^{\text{DDQN}} = \mathbb{E}_{\tau \sim U(\mathcal{D})}\Big[\big(Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}_t) - Q(O_t, A_t; \boldsymbol{\theta}_t)\big)^2\Big],$$



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

It seems to work ¯_ (ツ)_/¯

- “We closely follow the experimental setting and network architecture outlined by Mnih et al. (2015). (...) The network takes the last four frames as input and outputs the action value of each action. On each game, the network is trained on a single GPU for 200M frames, or approximately 1 week.” [van Hasselt, Guez, & Silver 2016]



Quality of the learned policies

[van Hasselt, Guez, & Silver 2016]

- “Overoptimism does not always adversely affect the quality of the learned policy. For example, DQN achieves optimal behavior in Pong despite slightly overestimating the policy value. Nevertheless, reducing overestimations can significantly benefit the stability of learning.” [van Hasselt, Guez, & Silver 2016]
- “As described by Mnih et al. (2015) each evaluation episode starts by executing a special no-op action that does not affect the environment up to 30 times, to provide different starting points for the agent. Some exploration during evaluation provides additional randomization. For Double DQN we used the exact same hyper-parameters as for DQN, to allow for a controlled experiment focused just on reducing overestimations. The learned policies are evaluated for 5 mins of emulator time (18,000 frames) with an ε-greedy policy where ε = 0.05. The scores are averaged over 100 episodes.”

Quality of the learned policies

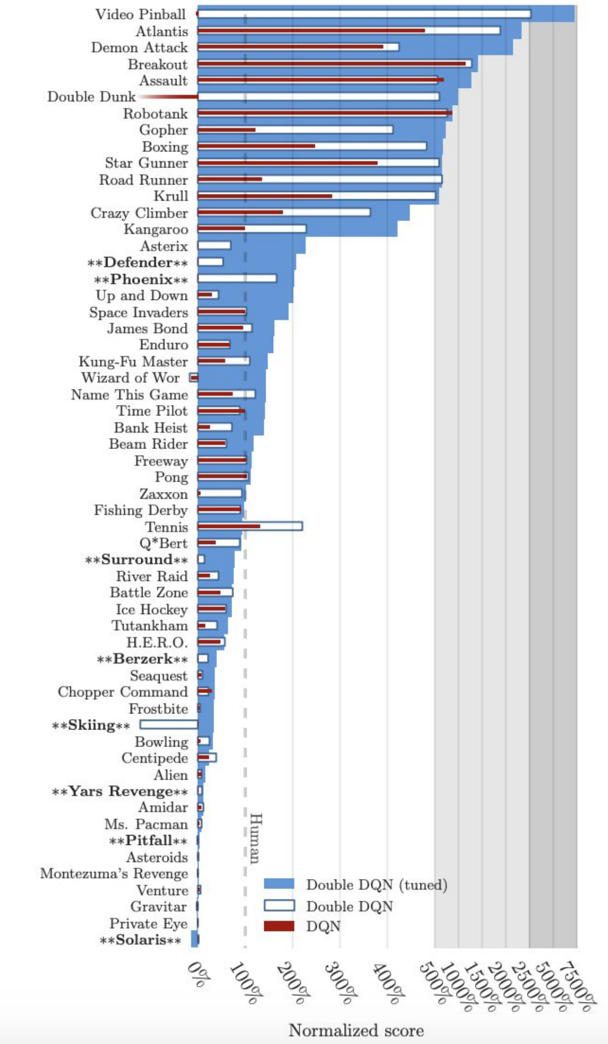
[van Hasselt, Guez, & Silver 2016]

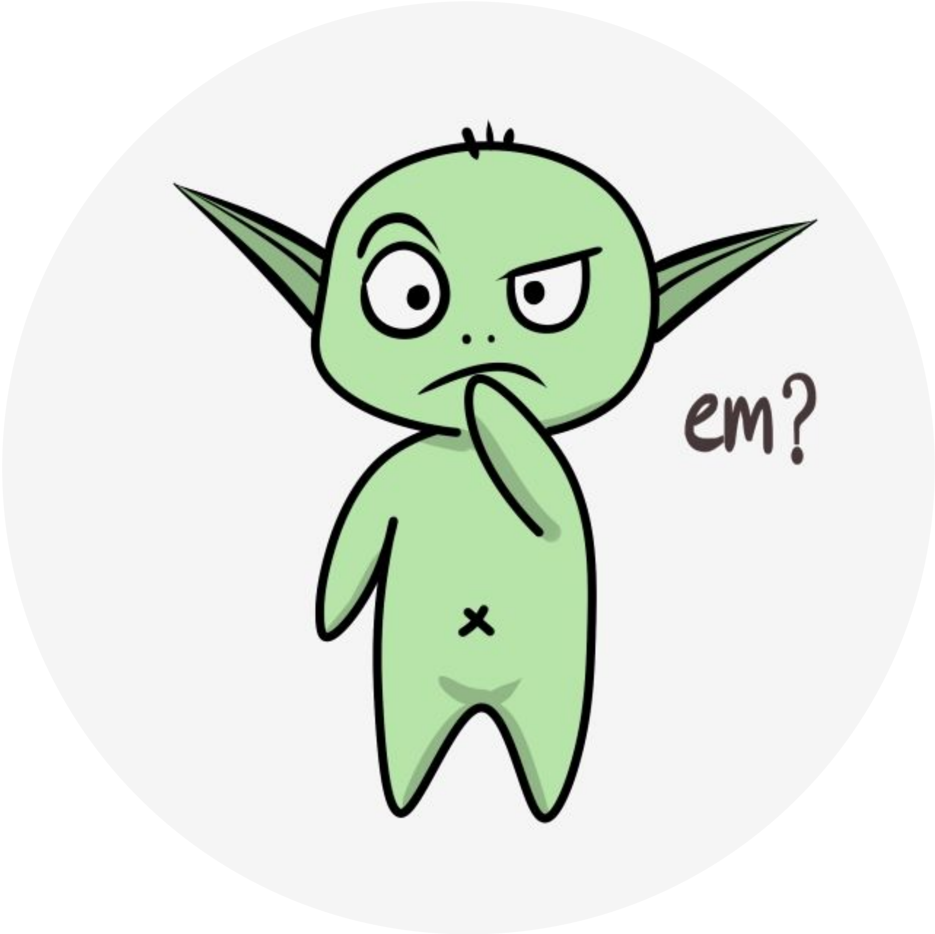
	DQN	Double DQN	Double DQN (tuned)
Median	47.5%	88.4%	116.7%
Mean	122.0%	273.1%	475.2%

$$\text{score}_{\text{normalized}} = \frac{\text{score}_{\text{agent}} - \text{score}_{\text{random}}}{\text{score}_{\text{human}} - \text{score}_{\text{random}}}$$

Game	Random	Human	DQN	Double DQN
Alien	227.80	6875.40	3069.33	2907.30
Amidar	5.80	1675.80	739.50	702.10
Assault	222.40	1496.40	3358.63	5022.90
Asterix	210.00	8503.30	6011.67	15150.00
Asteroids	719.10	13156.70	1629.33	930.60
Atlantis	12850.00	29028.10	85950.00	64758.00
Bank Heist	14.20	734.40	429.67	728.30
Battle Zone	2360.00	37800.00	26300.00	25730.00
Beam Rider	363.90	5774.70	6845.93	7654.00
Bowling	23.10	154.80	42.40	70.50
Boxing	0.10	4.30	71.83	81.70
Breakout	1.70	31.80	401.20	375.00
Centipede	2090.90	11963.20	8309.40	4139.40

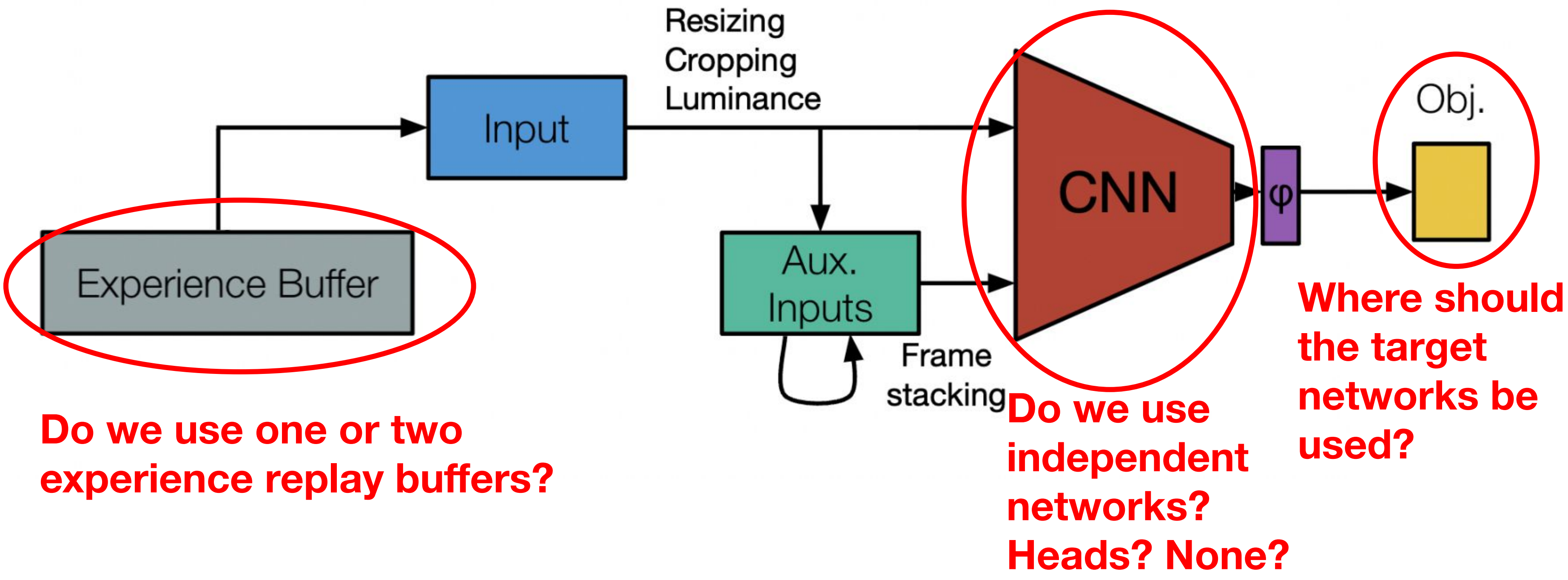
* 1 seed?





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

But this is deep RL: Which components should we change?



The Different Update Rules [Nagarajan et al., In Preparation]

- Two experience replay buffers, no target network:

$$Y_{v_1}(R_{t+1}, O_{t+1}; \boldsymbol{\theta}_t^1, \boldsymbol{\theta}_t^2) = R_{t+1} + \gamma Q\Big(O_{t+1}, \arg\max_{a \in \mathcal{A}} Q(O_{t+1}, a; \boldsymbol{\theta}_t^1); \boldsymbol{\theta}_t^2\Big)$$

$$\mathcal{L}_{v_1}^{\text{DDQL}} = \mathbb{E}_{\tau \sim U(\mathcal{D}^1)}\Big[\Big(Y_{v_1}(R_{t+1}, O_{t+1}; \boldsymbol{\theta}_t^1, \boldsymbol{\theta}_t^2) - Q(O_t, A_t; \boldsymbol{\theta}_t^1)\Big)^2\Big]$$

- Two independent estimators, but using a target network:

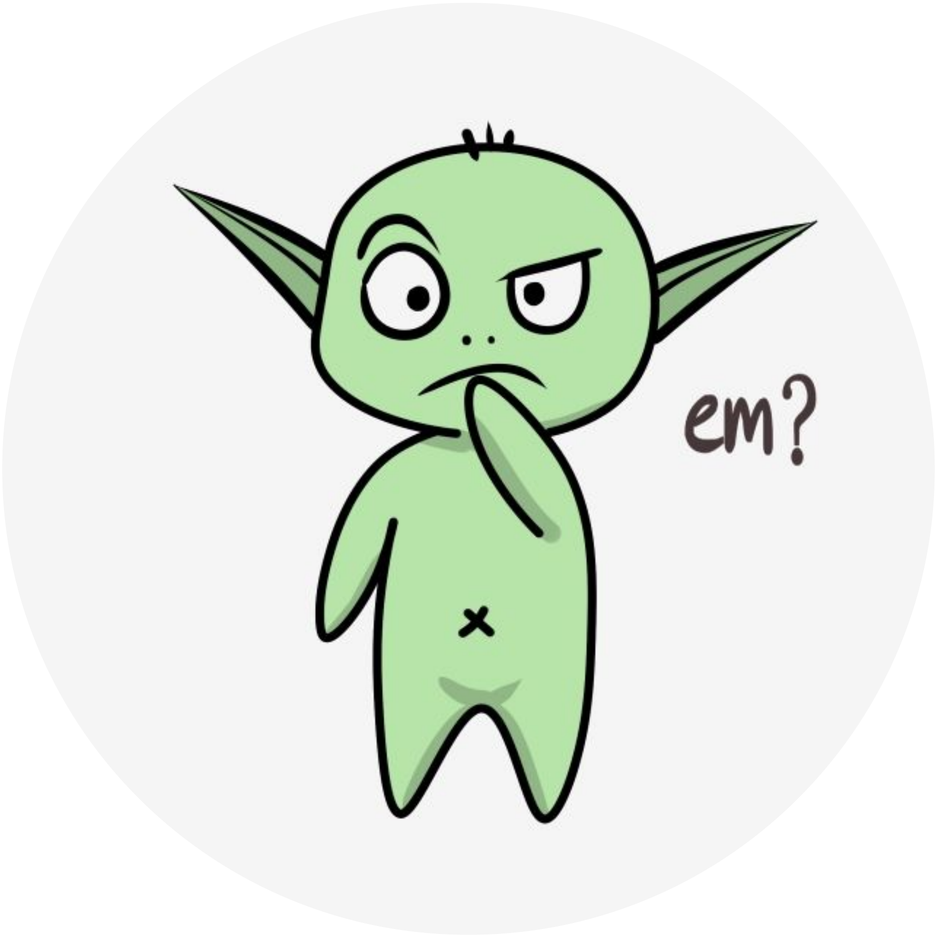
$$Y_{v_2}(R_{t+1}, O_{t+1}; \boldsymbol{\theta}_t^{-1}, \boldsymbol{\theta}_t^{-2}) = R_{t+1} + \gamma Q\Big(O_{t+1}, \arg\max_{a \in \mathcal{A}} Q(O_{t+1}, a; \boldsymbol{\theta}_t^{-1}); \boldsymbol{\theta}_t^{-2}\Big)$$

$$\mathcal{L}_{v_2}^{\text{DDQL}} = \mathbb{E}_{\tau \sim U(\mathcal{D}^1)}\Big[\Big(Y_{v_2}(R_{t+1}, O_{t+1}; \boldsymbol{\theta}_t^{-1}, \boldsymbol{\theta}_t^{-2}) - Q(O_t, A_t; \boldsymbol{\theta}_t^1)\Big)^2\Big]$$

The Different Update Rules [Nagarajan et al., In Preparation]

Action selector \ Action-value estimator	θ	θ^-
θ	DQN (no target)	Double DQN
θ^-	Inverse Double DQN	DQN

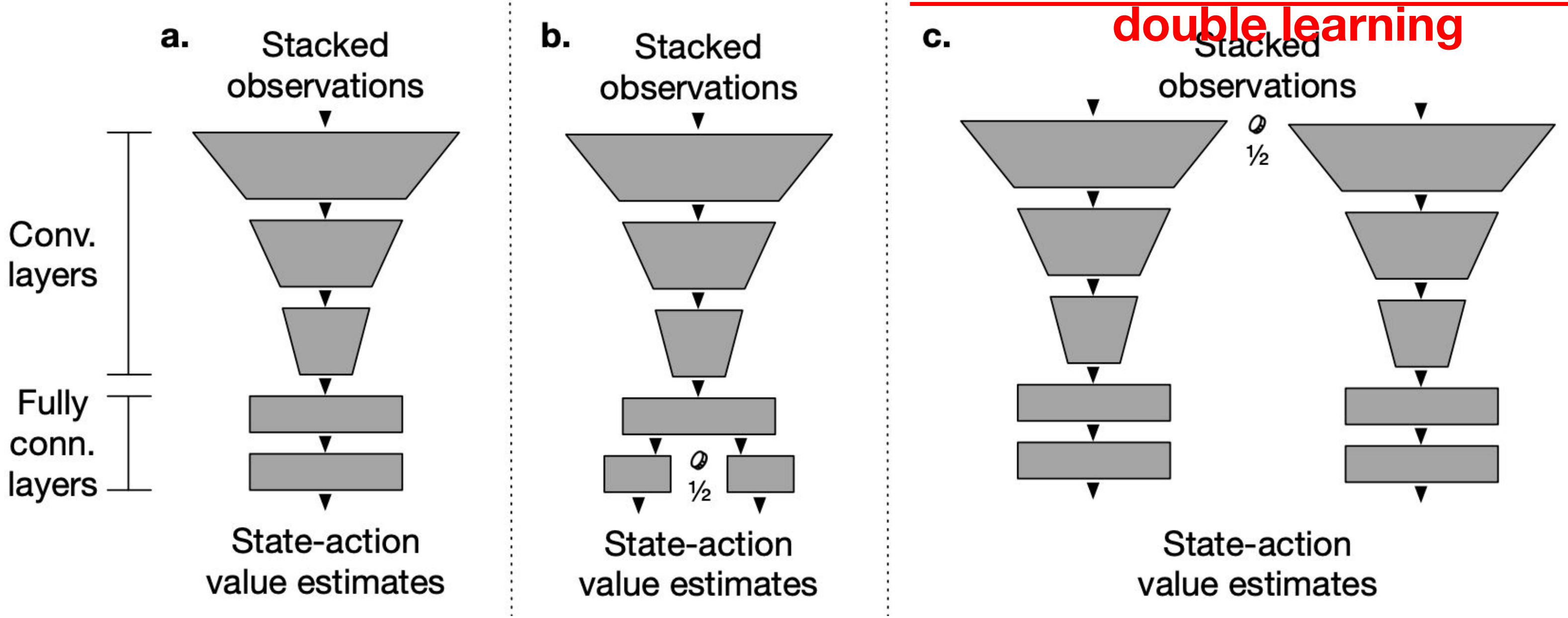
Action selector \ Action-value estimator	θ_2	θ_2^-
θ_1	DDQL (no target)	True DDQN?
θ_1^-	Inverse True DDQN?	DDQL



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

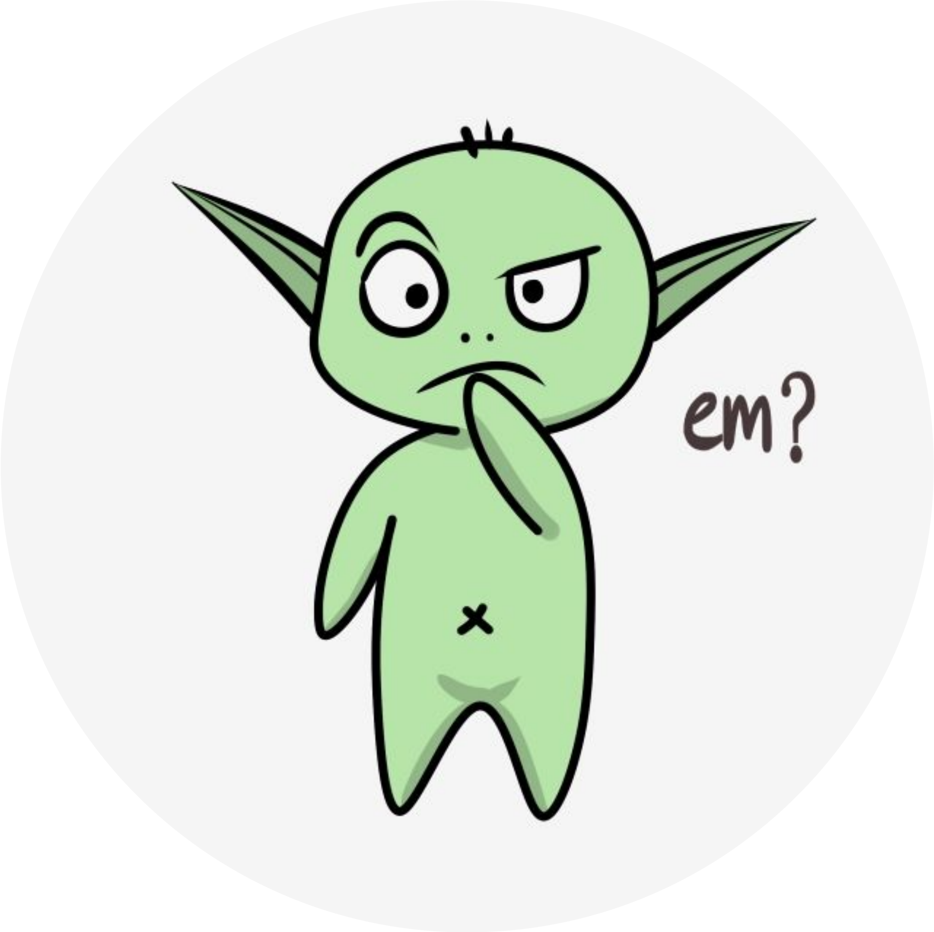
Different Neural Network Architectures [Nagarajan et al., In Preparation]

Arguably, this is a more faithful implementation of double learning



The Many Instantiation Choices [Nagarajan et al., In Preparation]

Double Q-learning feature	DDQN	DH-DDQL	DN-DDQL
(1) Decoupling of action-selection and Evaluation in target?	✓	✓	✓
(2) Trains two Q-functions?	✗	✓	✓
(2a) No shared parameters?	✗	✗	✓
(3a) Train on different minibatches?	✗	✓	✓
(3b) Train on partitioned datasets?	✗	✗	✗



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

Some Preliminary Results [Nagarajan et al., In Preparation]

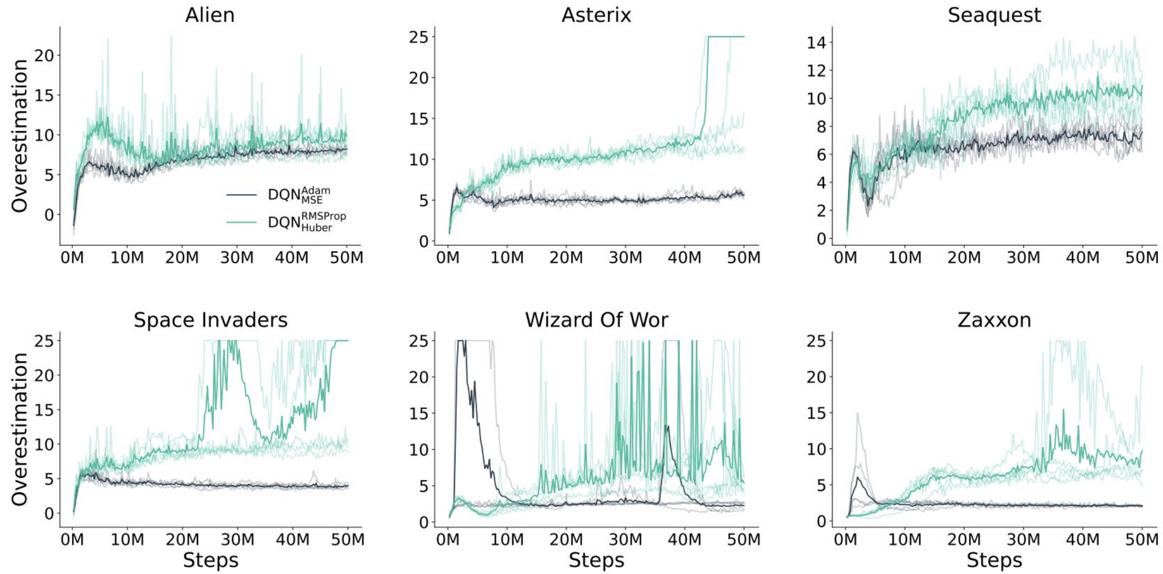


Figure 3: Overestimation comparison of $\text{DQN}^{\text{Adam}}_{\text{MSE}}$ versus $\text{DQN}^{\text{RMSProp}}_{\text{Huber}}$. The darker curves depict the mean overestimation across 5 seeds and the lighter curves depict the individual runs. The overestimation is capped at 25 for visibility.

Some Preliminary Results [Nagarajan et al., In Preparation]

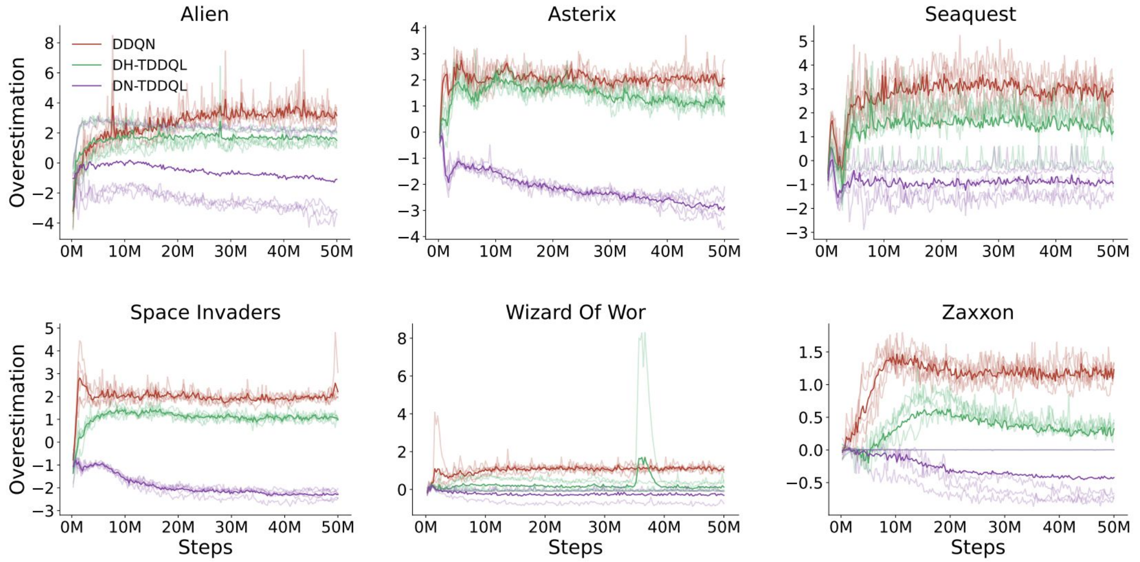


Figure 6: Overestimation comparison of different adaptations of Double Q-learning to the DQN-style algorithm setting. The mean overestimation across 5 seeds are the darker curves, and the lighter curves correspond to the individual runs.

Some Preliminary Results [Nagarajan et al., In Preparation]

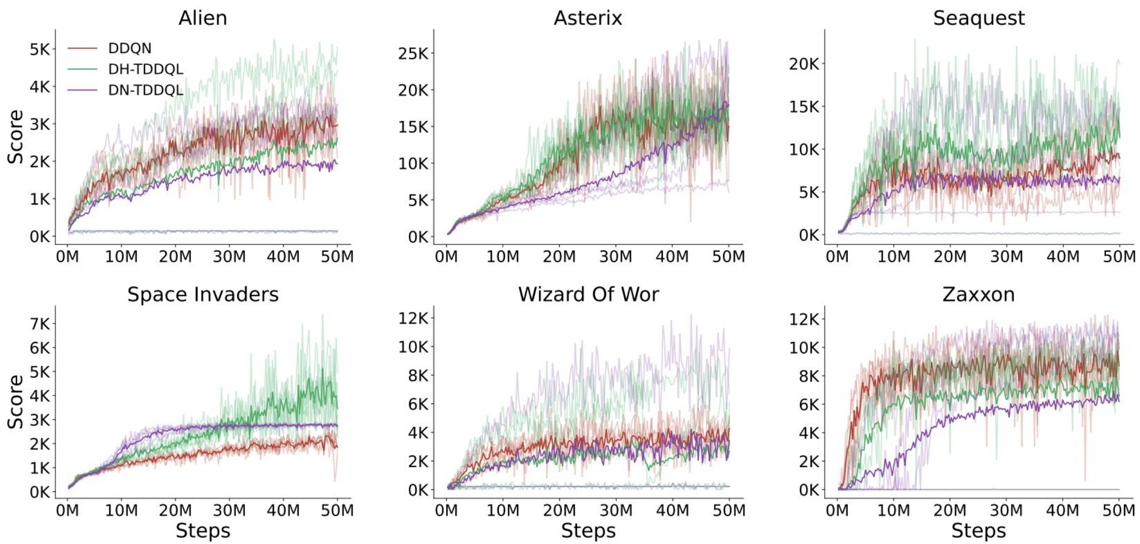


Figure 9: Score comparison of different adaptations of Double Q-learning in the DQN setting. The mean scores across 5 seeds are the opaque curves, and the translucent curves correspond to individual runs.

Some Preliminary Results [Nagarajan et al., In Preparation]

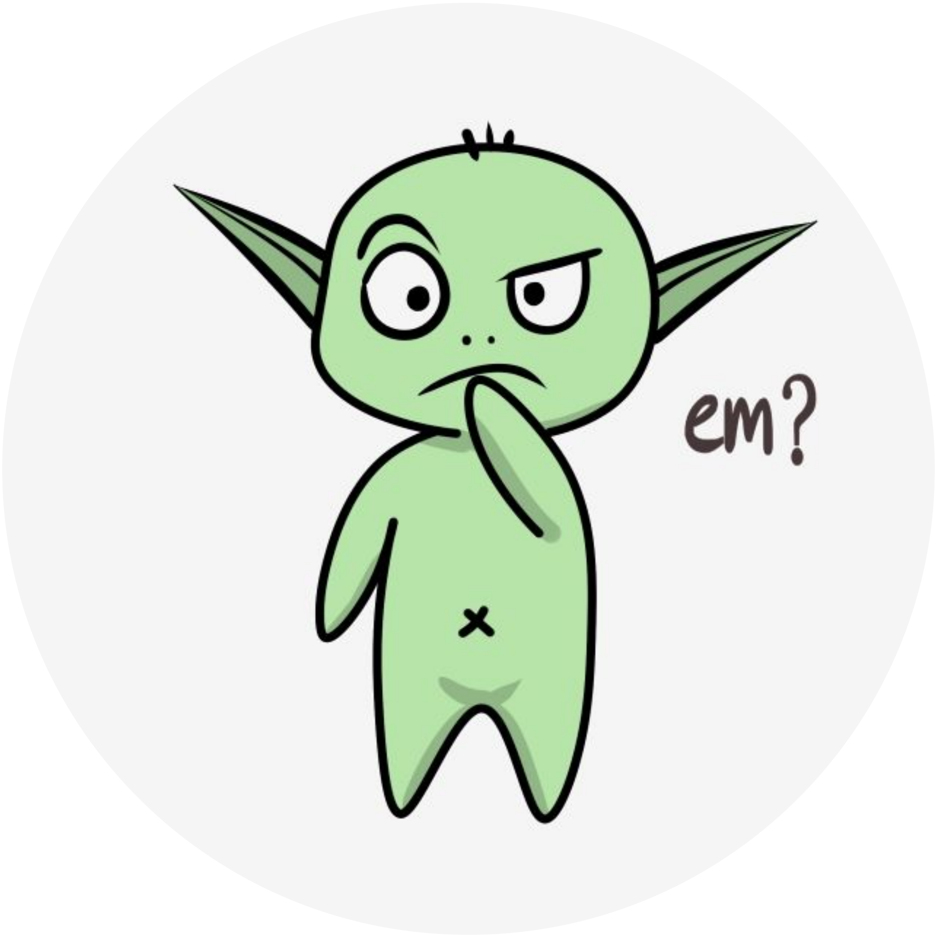
- The optimizer (Adam vs RMSProp) and the loss (MSE vs Huber) can have a big impact not only on performance, but also on overestimation
- Using two independent experience replay buffers has very little impact on the performance we observe (both for overestimation and the policy itself)
- Using two different networks drastically reduces overestimation, sometimes even leading to underestimation
- Reducing overestimation does not immediately lead to performance improvements

Some Preliminary Results [Nagarajan et al., In Preparation]

On the Development and Analysis of Algorithm Variations

Making definitive claims about an algorithm is difficult because:

- Hyperparameters need to be properly swept and hyperparameters are not independent from each other.
- Sometimes it is not easy to make apples-to-apples comparison (e.g., number of networks vs number of gradient updates)



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

Next class

- What I plan to do:
  - Continue discussing different instantiations of deep RL algorithms through the objective function, more specifically, multi-step methods.
- What I recommend YOU to do for next class:
  - Read
    - *Chapters 7 and 12 by Sutton and Barto (2018)*
    - *[Optional] Daley, B., White, M., Amato, C., and Machado, M. C. (2023). Trajectory-aware eligibility traces for off-policy reinforcement learning. In Proceedings of the International Conference on Machine Learning*
  - Work on Assignment 2!