AMAC Journal Club on

« Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., & Blundell, C. (2020, November). **Agent57: Outperforming the atari human benchmark.** In *International Conference on Machine Learning* (pp. 507-517). PMLR. »

S. Doncieux

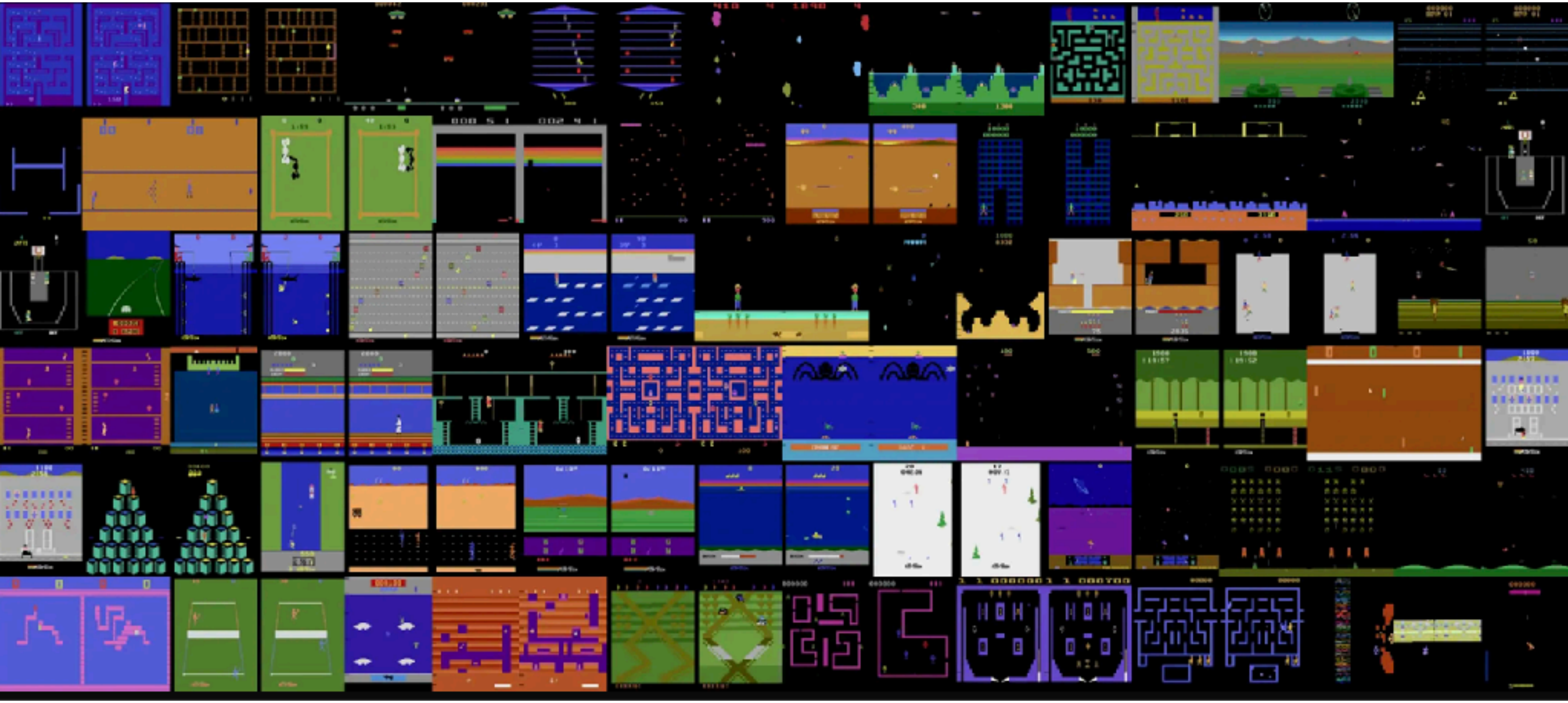
Sources

- Article: Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., & Blundell, C. (2020, November). Agent57: Outperforming the atari human benchmark. In *International Conference on Machine Learning* (pp. 507-517). PMLR.

- Blog: https://deepmind.com/blog/article/Agent57-Outperforming-the-human-Atari-benchmark

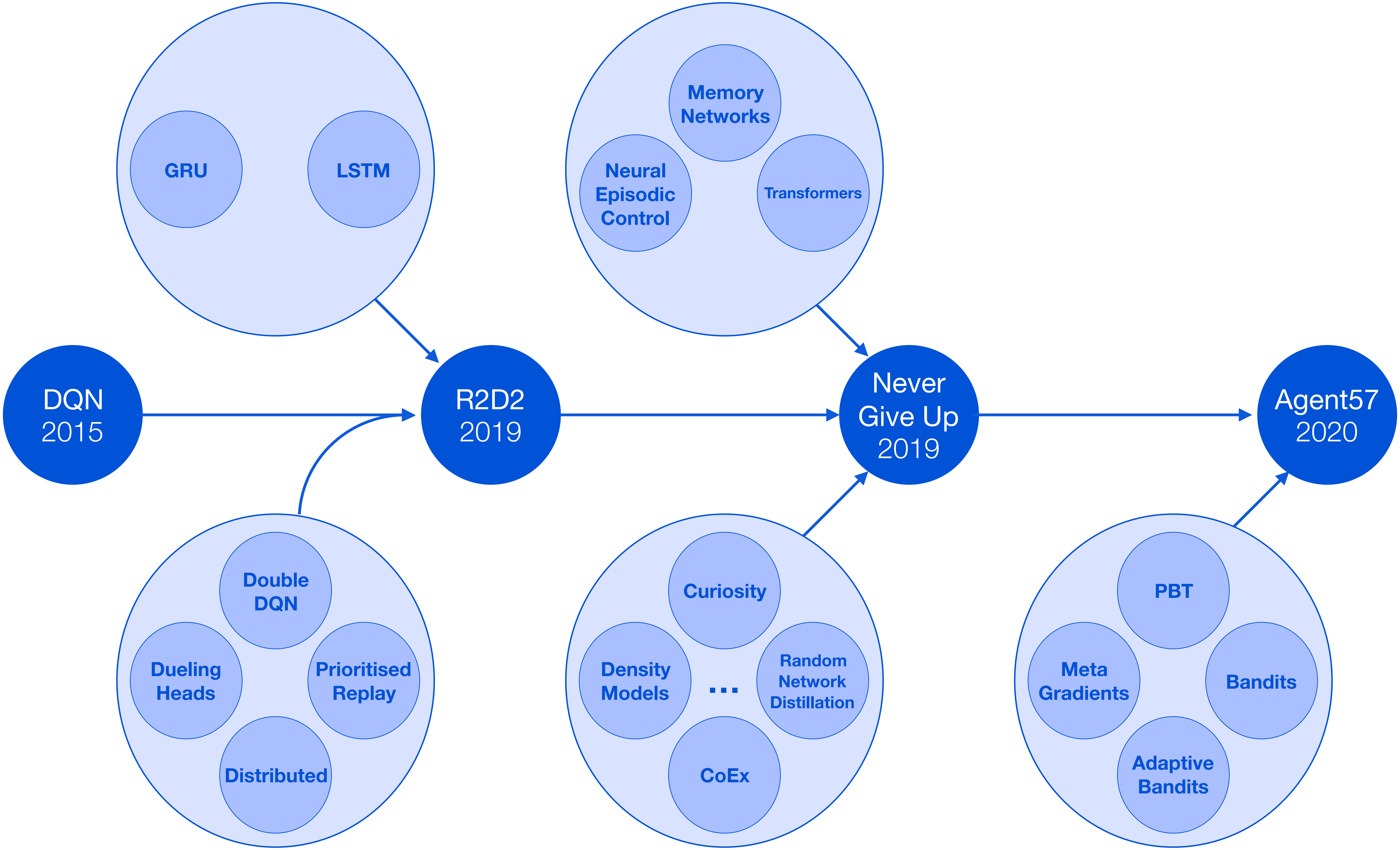
- and some of the articles it depends on…

Atari Benchmark

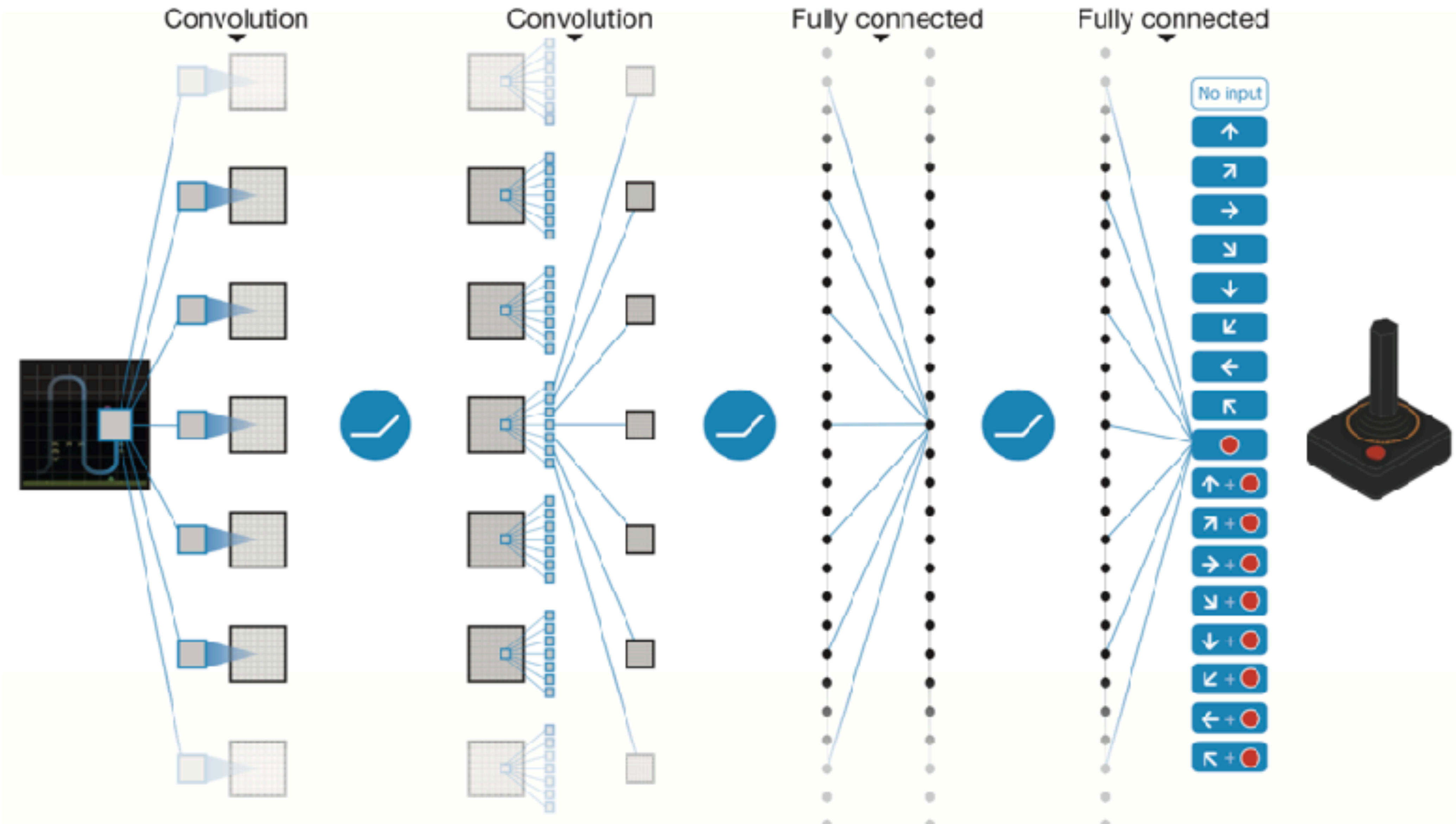


- Atari games as a proxy to study Artificial General Intelligence
- Can we find a same learning algorithm that could defeat humans on **all** games ?

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation plat- form for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 06 2013.

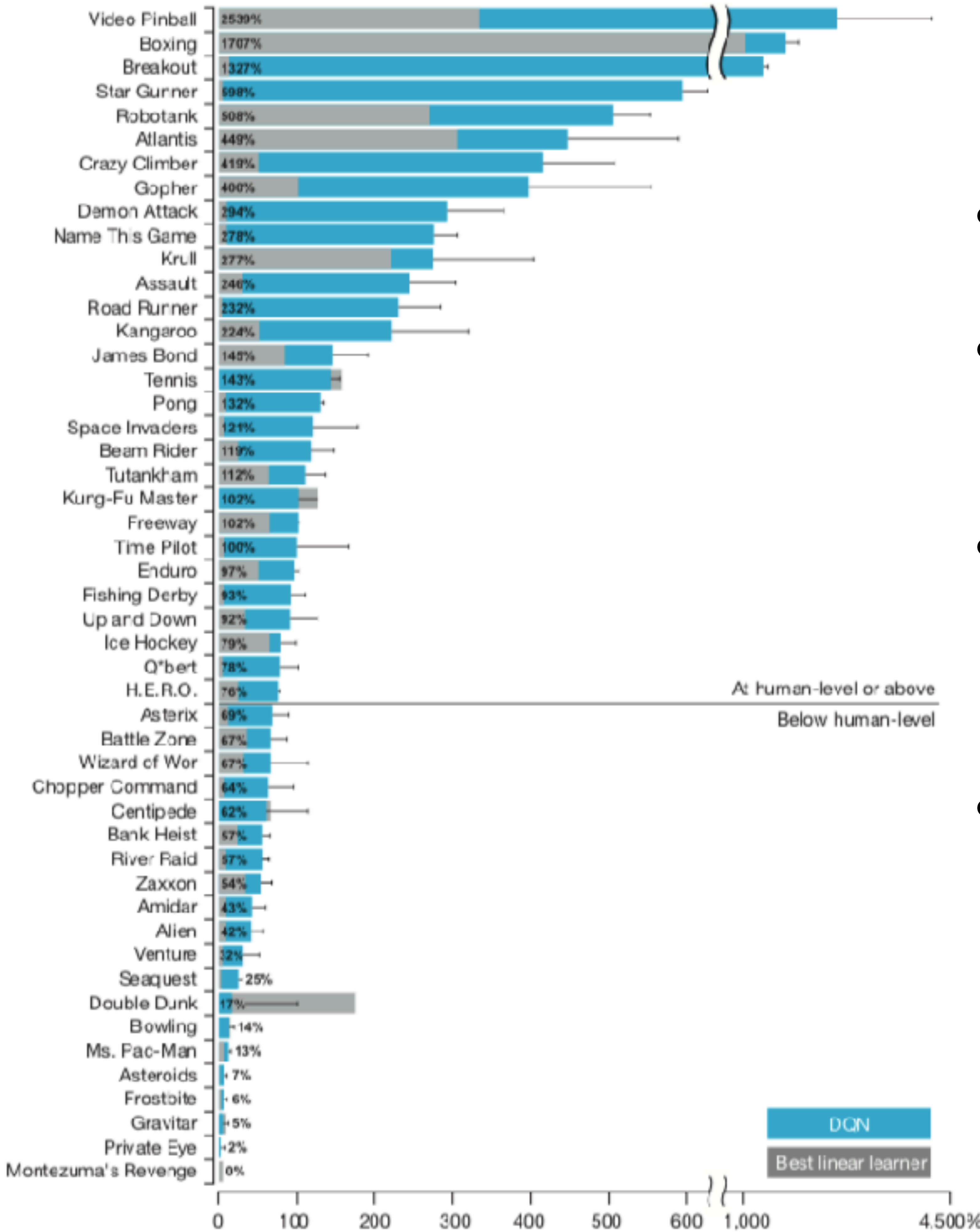

GRU
LSTM
Memory Networks
Neural Episodic Control
Transformers
DQN 2015
R2D2 2019
Never Give Up 2019
Agent57 2020
Double DQN
Dueling Heads
Prioritised Replay
Distributed
Curiosity
Density Models
...
Random Network Distillation
CoEx
PBT
Meta Gradients
Bandits
Adaptive Bandits

At the beginning was DQN...

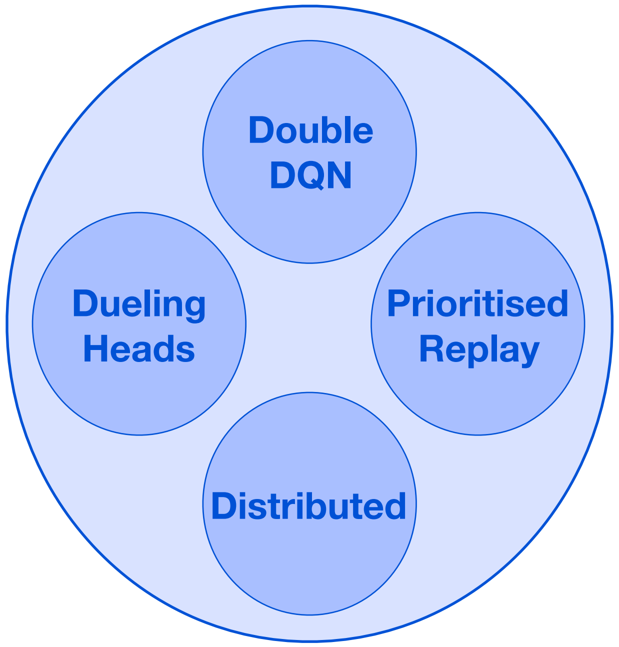


- Deep Q-Learning with experience replay and iterative update
- One Q value for each action
- s is the sequence of observations (84x84x4 = 84x84 images at 4 time steps)

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529-533.



- Tested on 49 games
- > human expert on 23 games
- Same architecture, same meta-parameters for all games
- Trained during 50million frames on each game (~38 days)

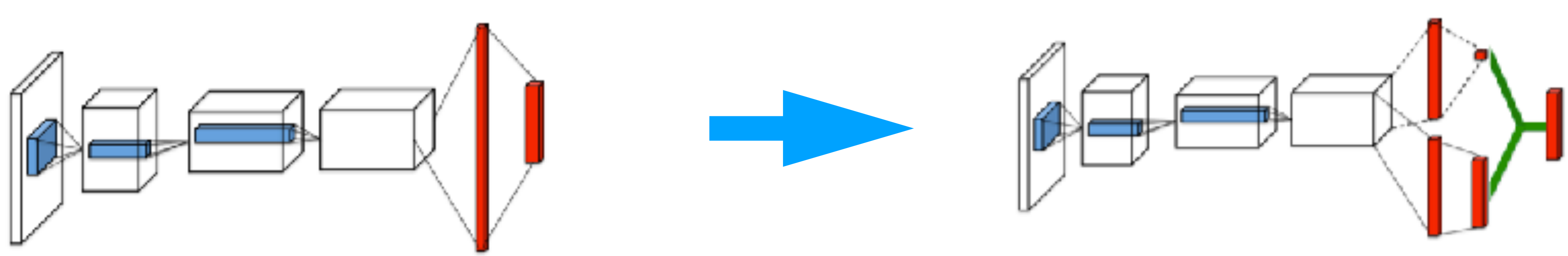


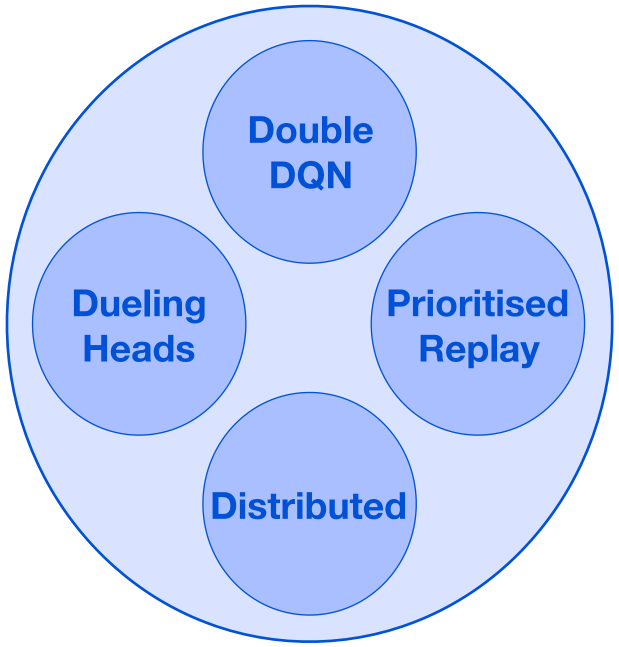
DQN improvements

- **Double DQN**: using 2 Deep Q-Networks one for policy determination, one for the value update (and switch them on a regular basis)

- **Prioritised Replay**: transitions are selected thanks to a probability that depends on the TD-error

- **Dueling Heads:** decompose Q into the value and advantage functions

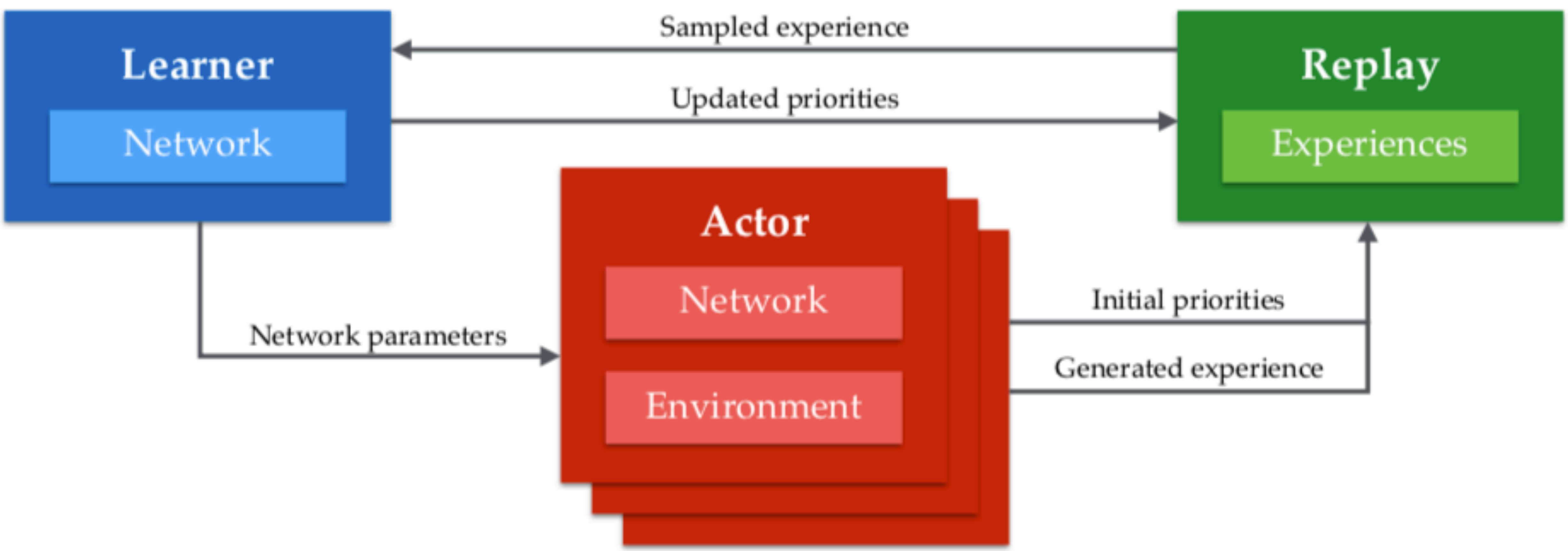




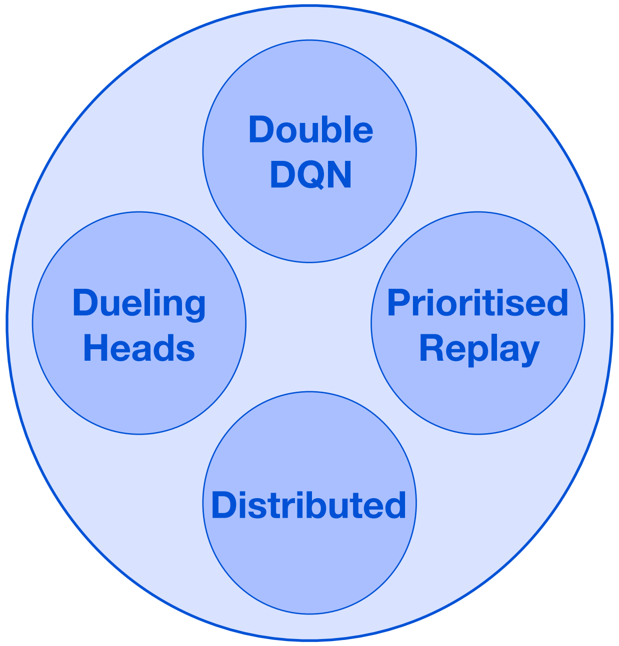
DQN improvements

Ape-X

- **Distributed Deep RL:** several actors, several learners, each in charge of updating a part of the parameters

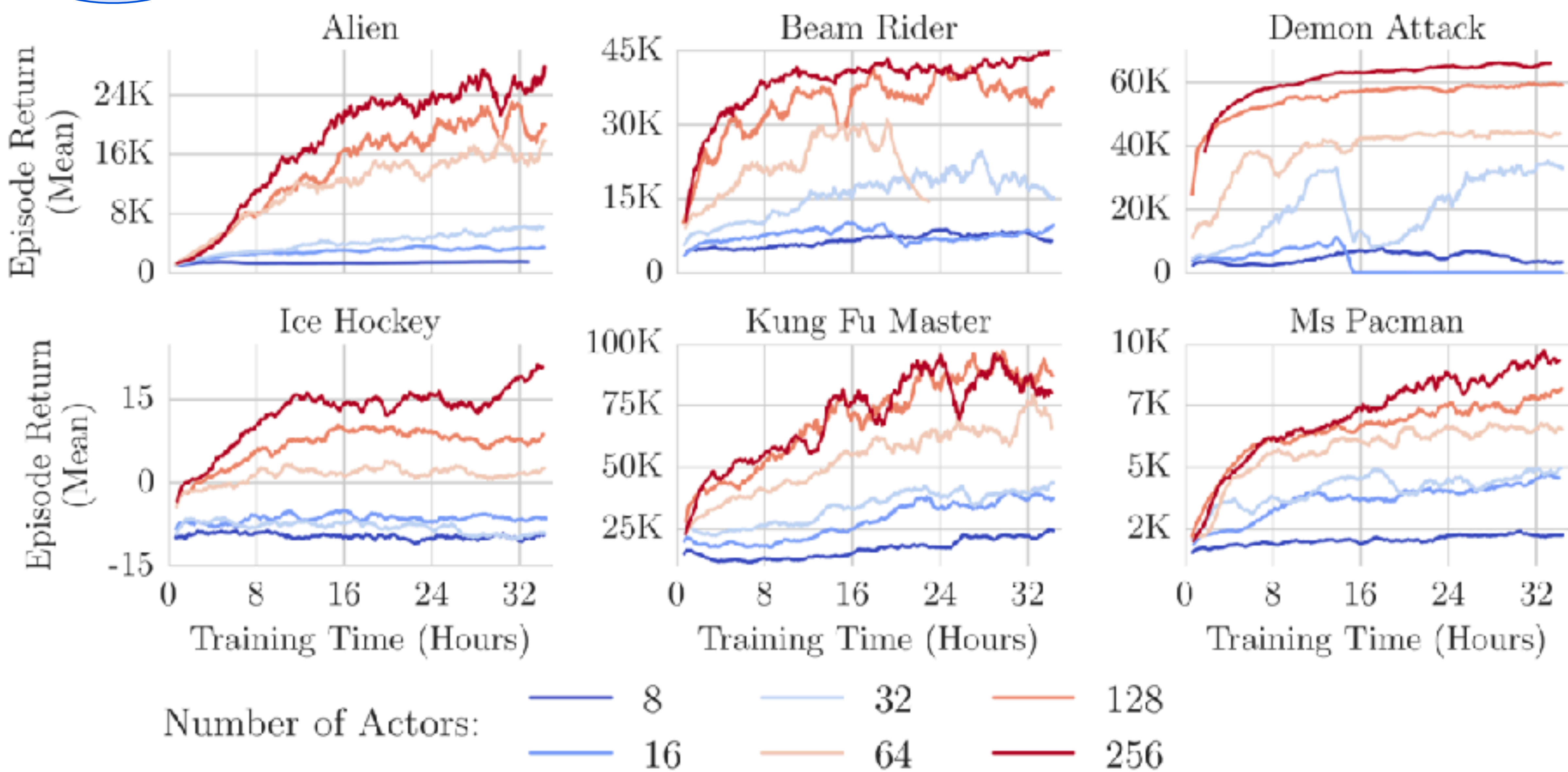


Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, 2018.



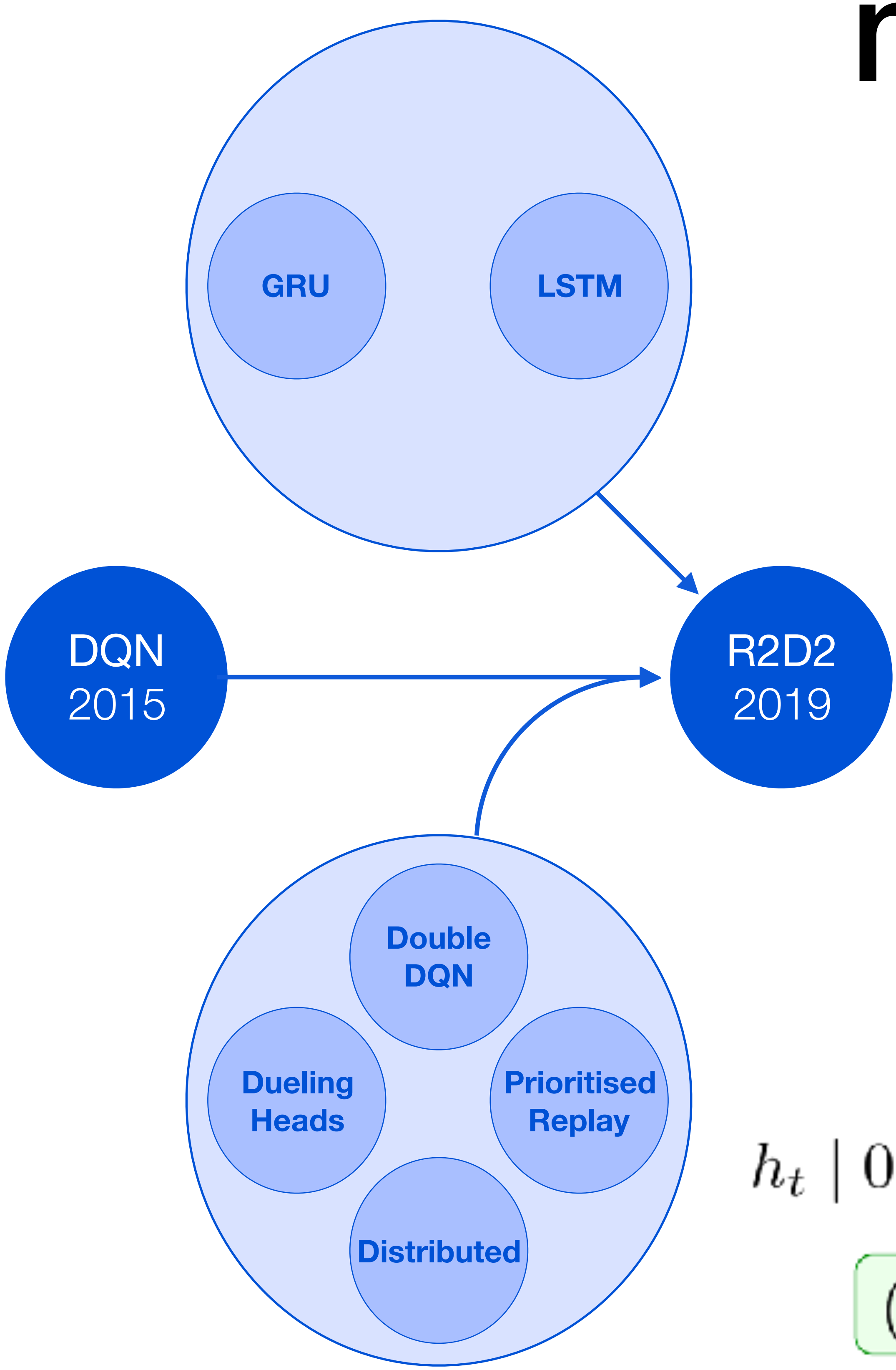
DQN improvements

Ape-X

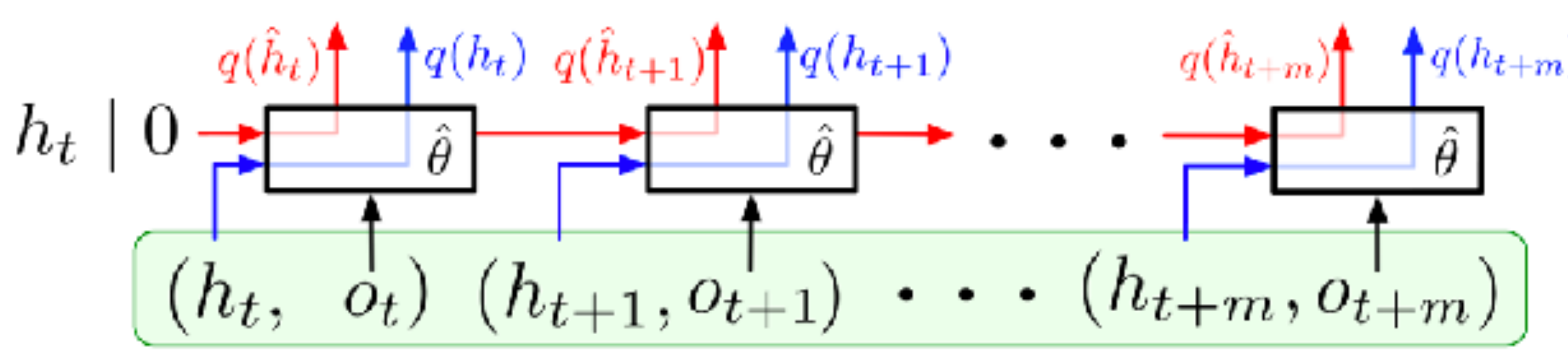


Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, 2018.

R2D2: Adding a short-term memory



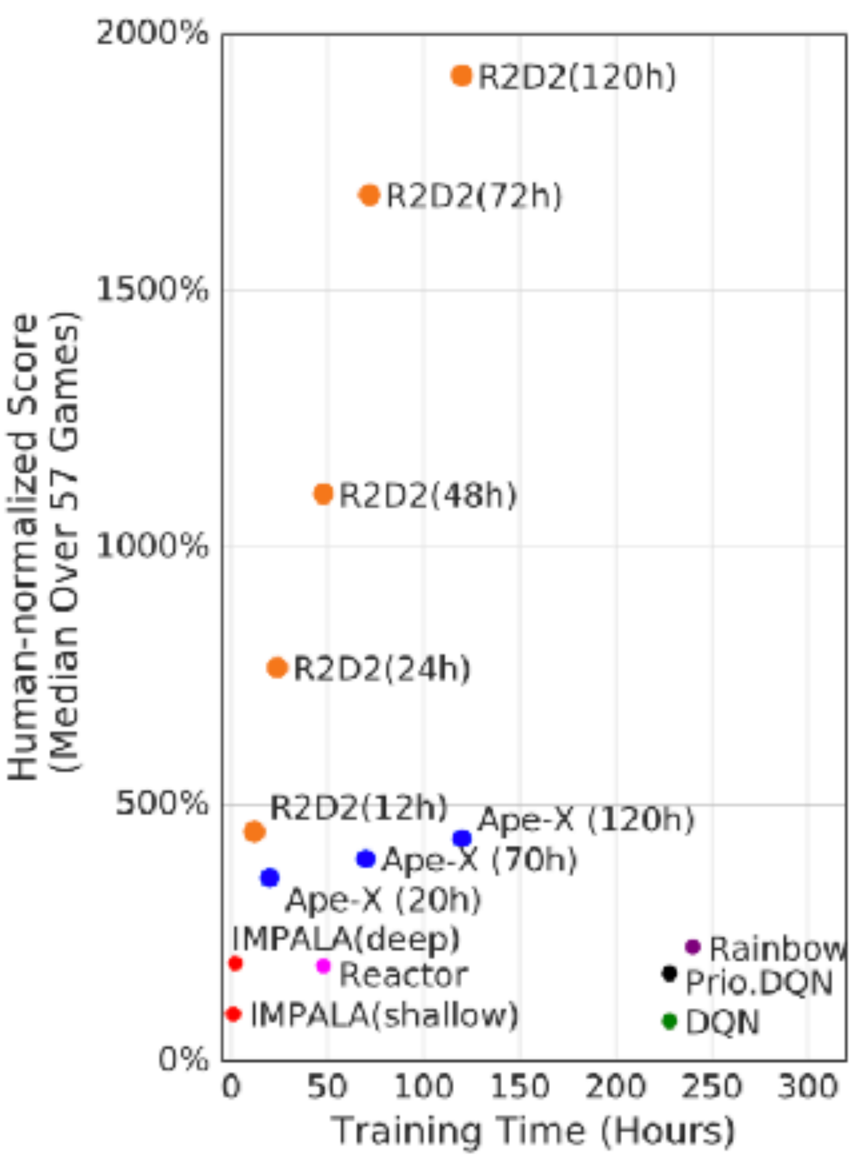
- Recurrent Replay Distributed DQN (R2D2)
- Formalize the problem as a POMDP, i.e. a partially observable MDP, $(\mathscr{S}, \mathscr{A}, T, R, \Omega, \mathscr{O})$:
  - $\mathscr{S}$: state (unobserved)
  - $\mathscr{A}$: actions
  - T: transition function
  - $R : \mathscr{S} \times \mathscr{A} \to \mathbb{R}$
  - Ω: observation set
  - $\mathscr{O}$: observation function mapping (unobserved) states to probability distributions over Ω
- Use a Recurrent NN (LSTM) to learn a representation that disambiguates the true state of the POMDP
- Propose mechanisms to train the LSTM from randomly sampled sequences (what initial internal state to use ?):
  - Store internal state in the replay buffer
  - Use a « burn-in » strategy to recover the state

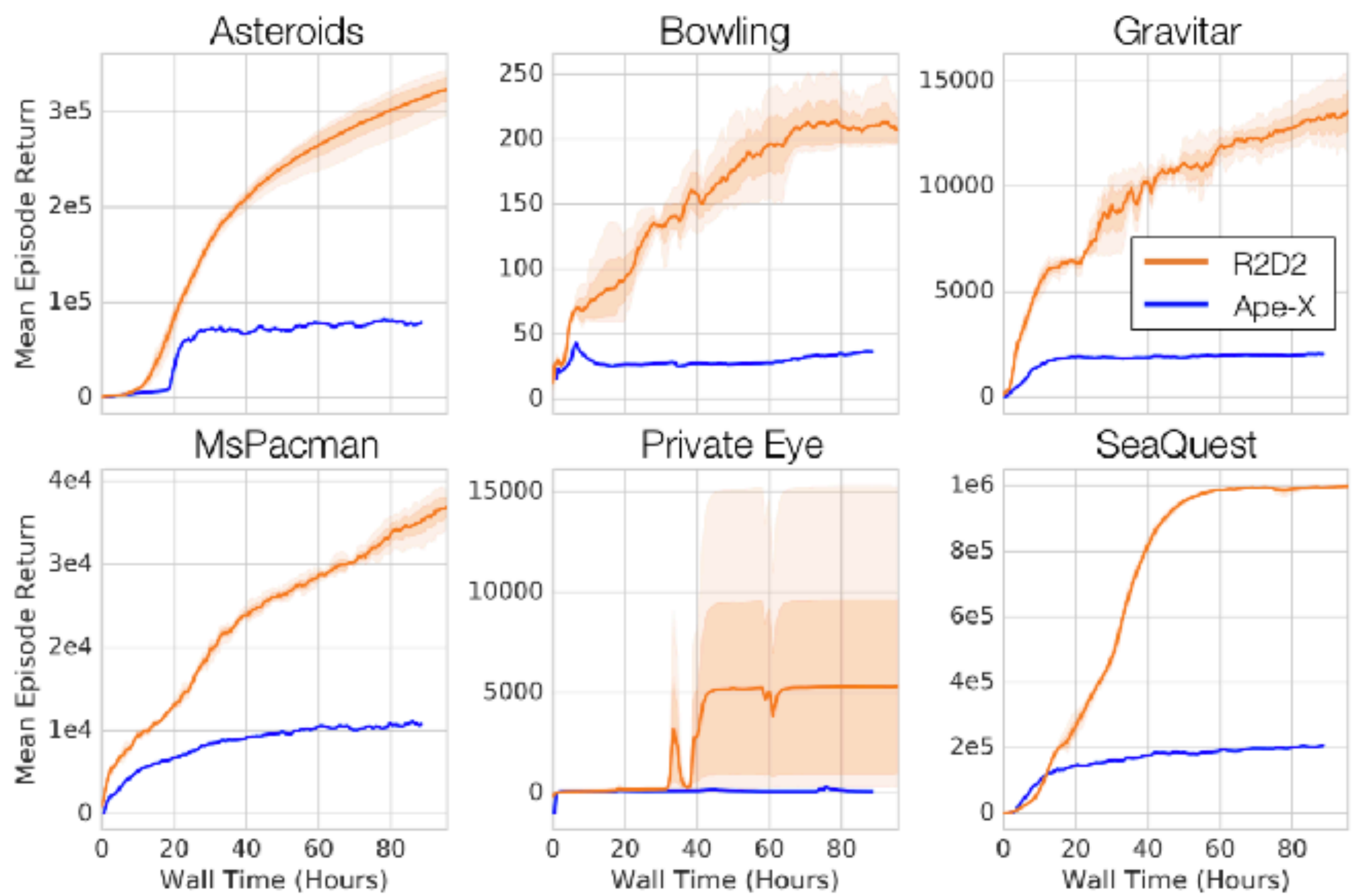


R2D2: Kapturowski, Steven, et al. "Recurrent experience replay in distributed reinforcement learning." ICLR (2019).

R2D2
2019

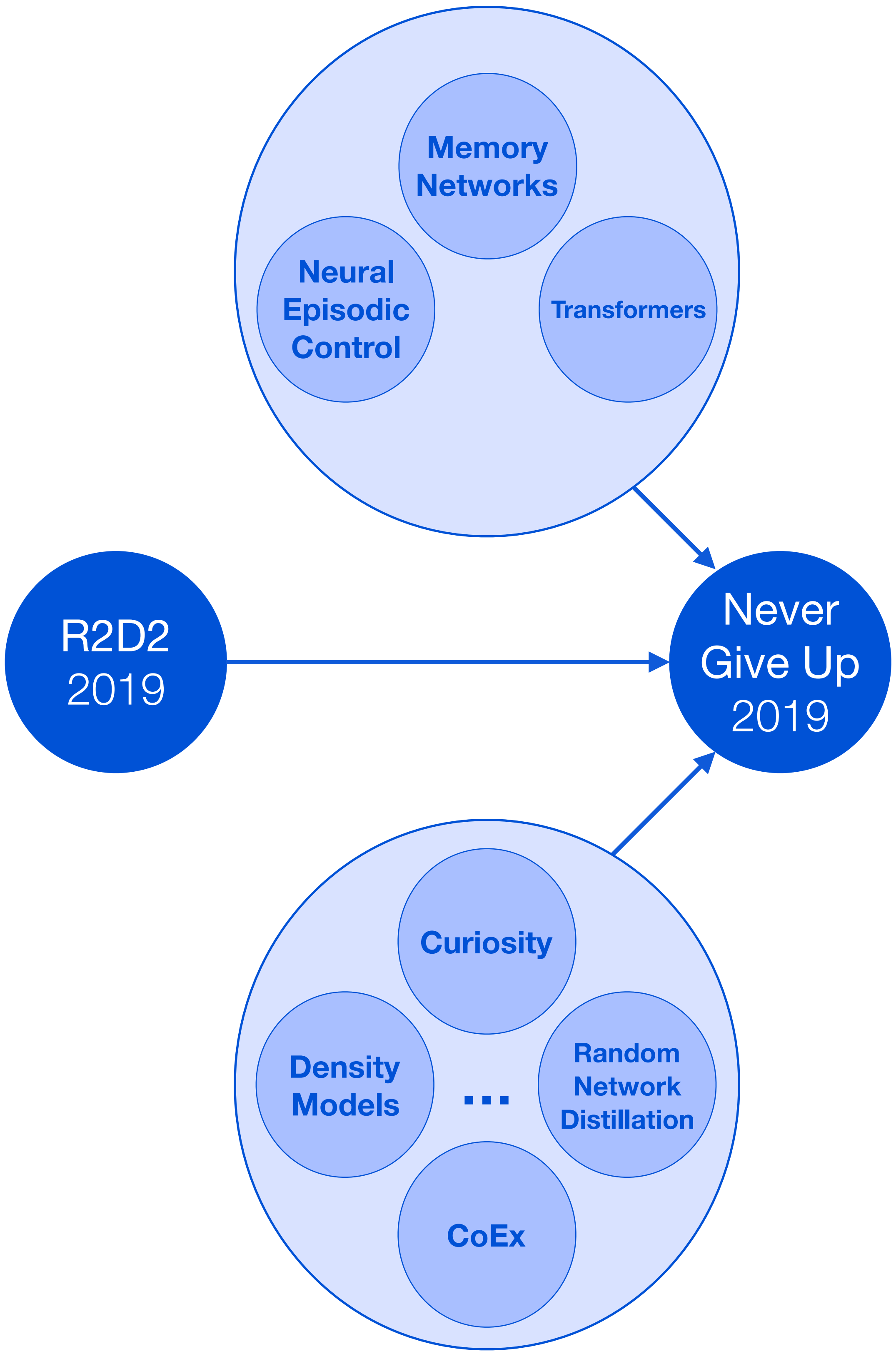
R2D2: Adding a short-term memory





R2D2: Kapturowski, Steven, et al. "Recurrent experience replay in distributed reinforcement learning." ICLR (2019).

Never Give Up:
Some more memory and exploration



- How to make a better exploration than ϵ-greedy strategies ?
- Propositions:
  - Combine extrinsic reward with an intrinsic reward:

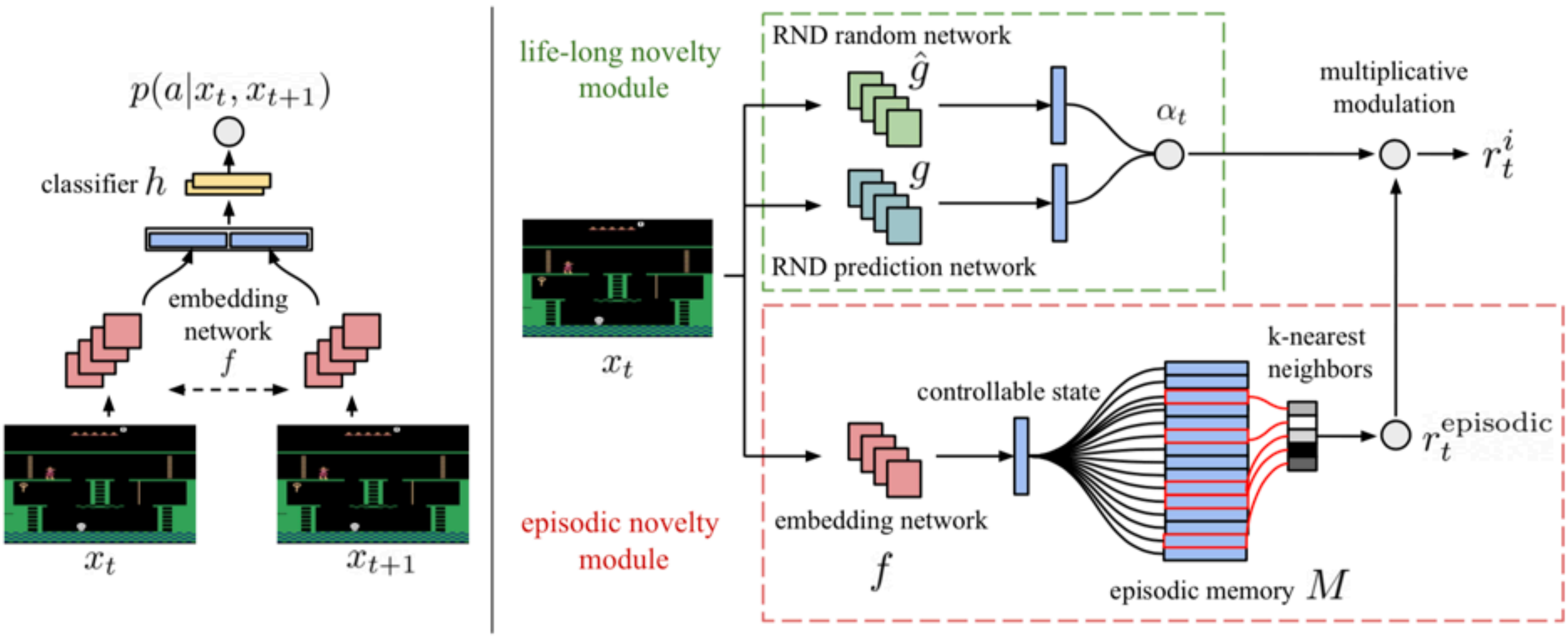
$$r_t = r_t^e + \beta r_t^i$$

  - r_t^i includes long-term and short-term novelty over *controllable states*
  - Learn $Q(x, a, \beta_i)$ to be able to act greedily (following $Q(x, a, 0)$) or not

Never Give Up: Puigdomènech Badia, Adrià, et al. "Never Give Up: Learning Directed Exploration Strategies." arXiv (2020): arXiv-2002.

Never Give Up 2019

Never Give Up: Intrinsic reward

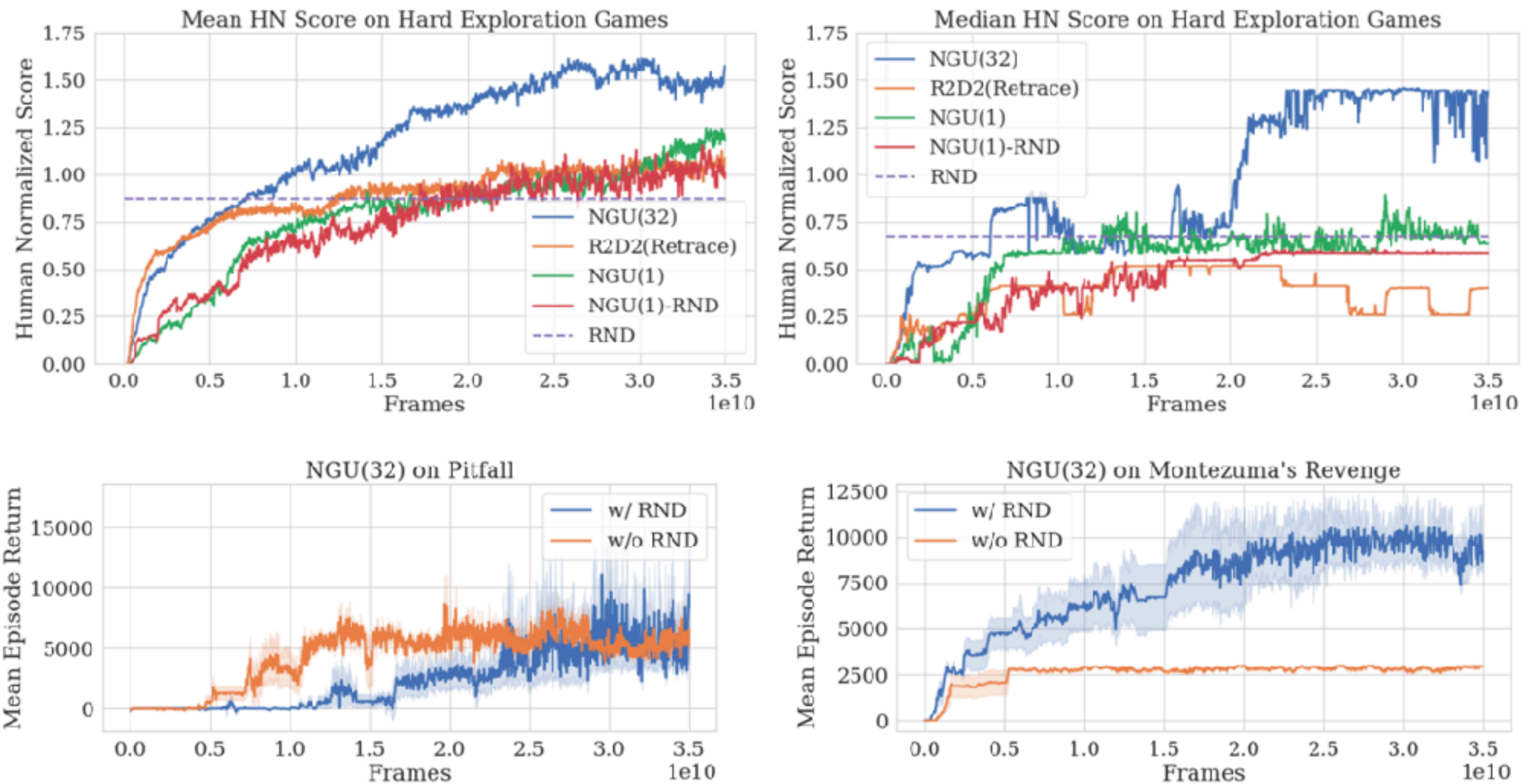


$$r_t^{episodic} \approx \frac{1}{\sqrt{\sum_{f_i \in N_k} K(f(x_t), f_i)} + c} \quad \text{with} \quad K(x, y) = \frac{\epsilon}{\frac{d^2(x,y)}{d_m^2} + \epsilon}$$

Never Give Up: Puigdomènech Badia, Adrià, et al. "Never Give Up: Learning Directed Exploration Strategies." arXiv (2020): arXiv-2002.

Never
Give Up
2019

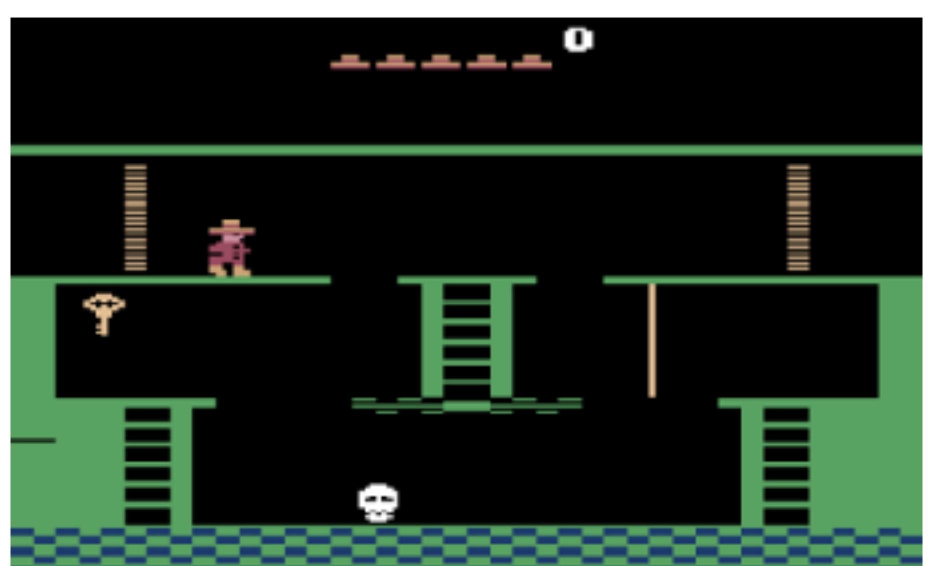
Never Give Up: Intrinsic reward



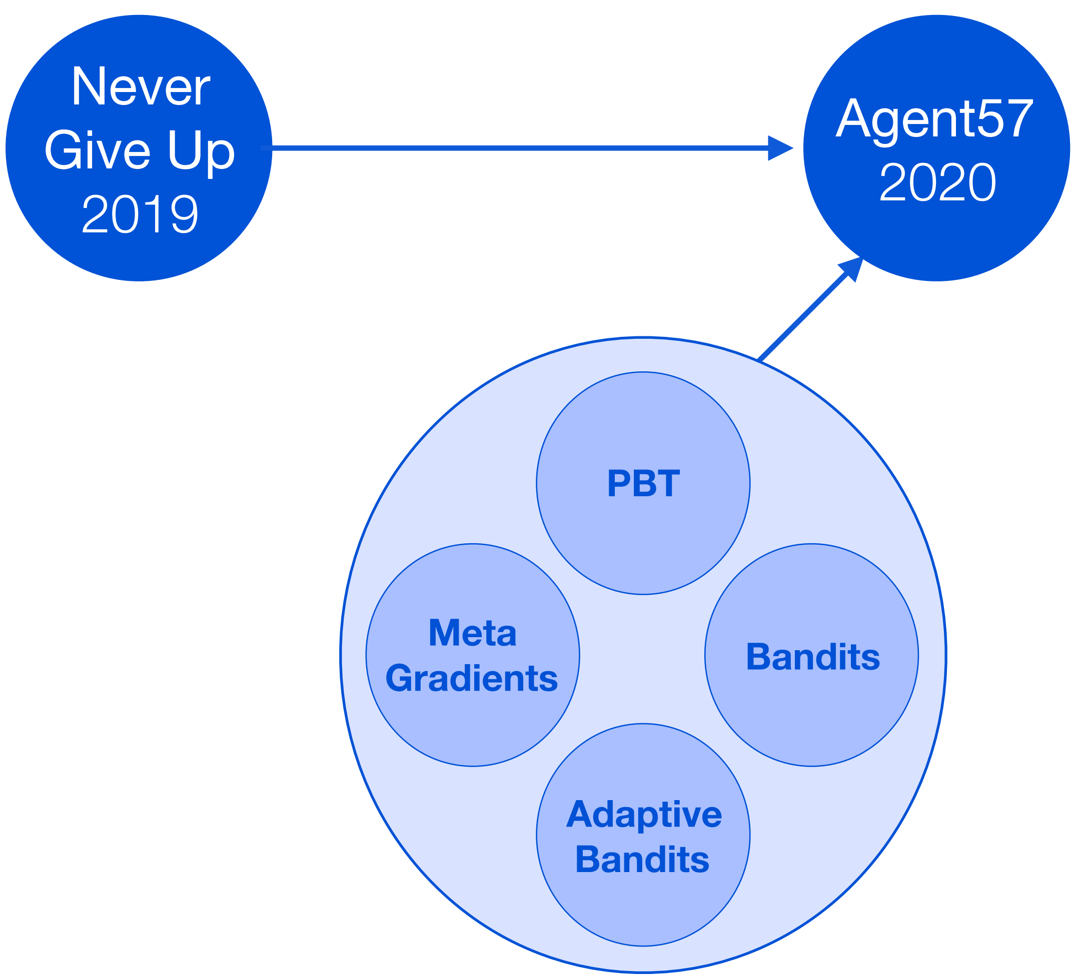
Never Give Up: Puigdomènech Badia, Adrià, et al. "Never Give Up: Learning Directed Exploration Strategies." arXiv (2020): arXiv-2002.

Atari Benchmark: what's the situation before Agent57 ?

- Some games haven't been solved yet: Pitfall, Skiiing, Montezuma revenge …
- The main challenges at this point:
  - long term credit assignment
  - exploration



Agent 57
Improvements over NGU



- Increase of the backpropagation through time window $(80 \rightarrow 160)$
- Decomposition of the Q network
- Dynamical adjustment of the discount factor and of the exploration/exploitation trade-off

➡ Relies on a multi-arm bandit

Agent57
2020

Improvement over NGU:

1. State-Action Value Function Parameterization

- Splitting the state-value function:

$$Q(x, a, j; \theta) = Q(x, a, j; \theta^e) + \beta_j Q(x, a, j; \theta^i)$$

with one NN per Q function term where:

  - $Q(x, a, j; \theta^e)$ is the extrinsic reward
  - $Q(x, a, j; \theta^i)$ is the intrinsic reward
  - $\theta = \theta^i \cup \theta^e$
  - optimized separately with resp. rewards r^e and r^i and same target policy:

$$\pi(x) = argmax_{a \in \mathcal{A}} Q(x, a, j; \theta)$$

- Training with the same sequence of transitions sampled from the replay buffer, but with 2 different transformed Retrace loss functions (with r^e and target policy π and with r^i and target policy π)

Agent57
2020

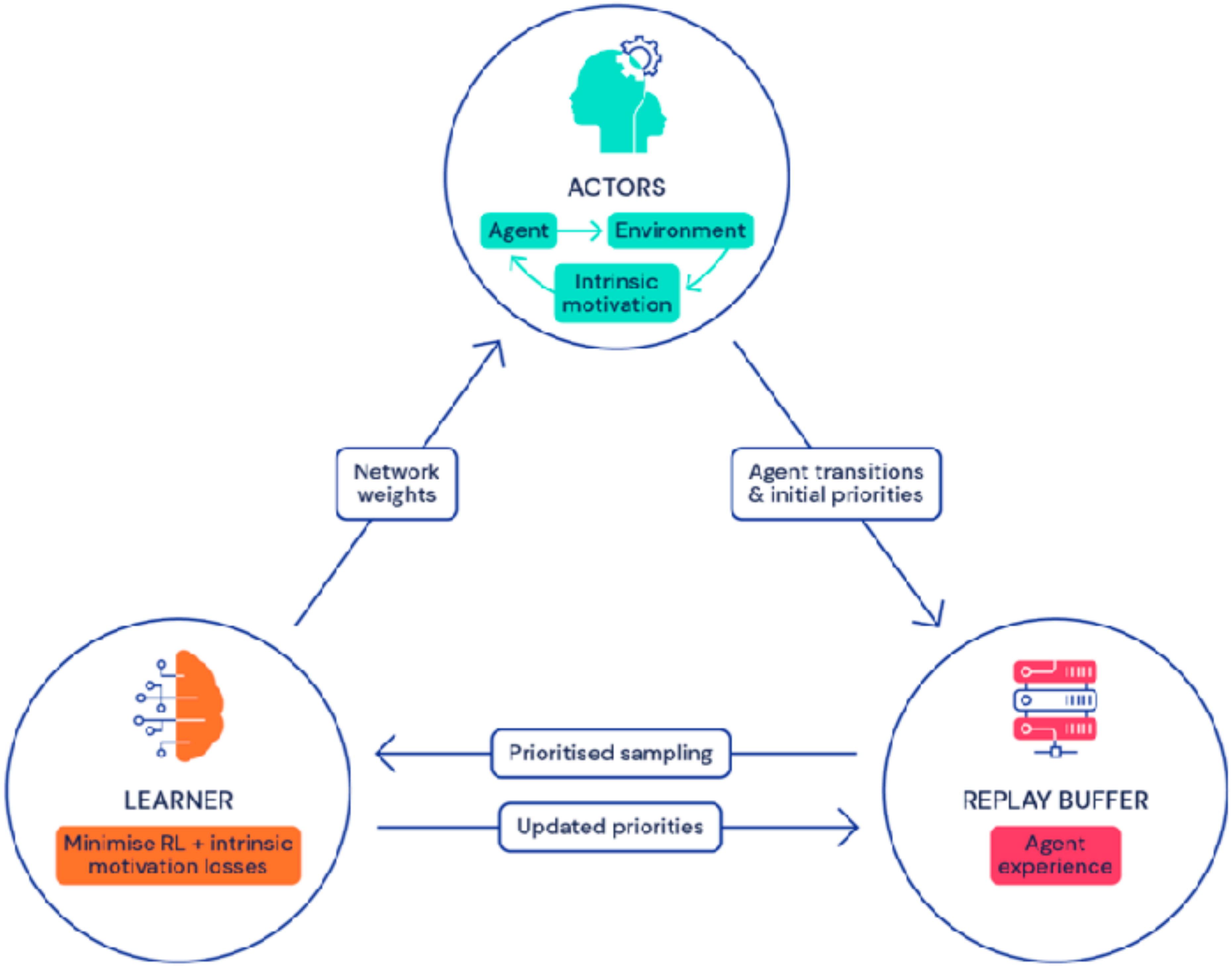
Improvement over NGU:

2. Adaptive Exploration over a Family of Policies

- Select which policy to use at training and evaluation times
- Policies represented by 32 different (β_j, γ_j)
- Non stationary multi-arm bandit running on each of the 256 actors:
    - at episode k, the meta-controller selects J_k
    - l-th actor acts ϵ_l-greedily w.r.t. $Q(x, a, J_k; \theta_l)$
    - undiscounted extrinsic reward $R_k^e(J_k)$ used to train the multi-arm bandit (sliding window UCB with ϵ_{UCB}-greedy exploration

Agent57
2020

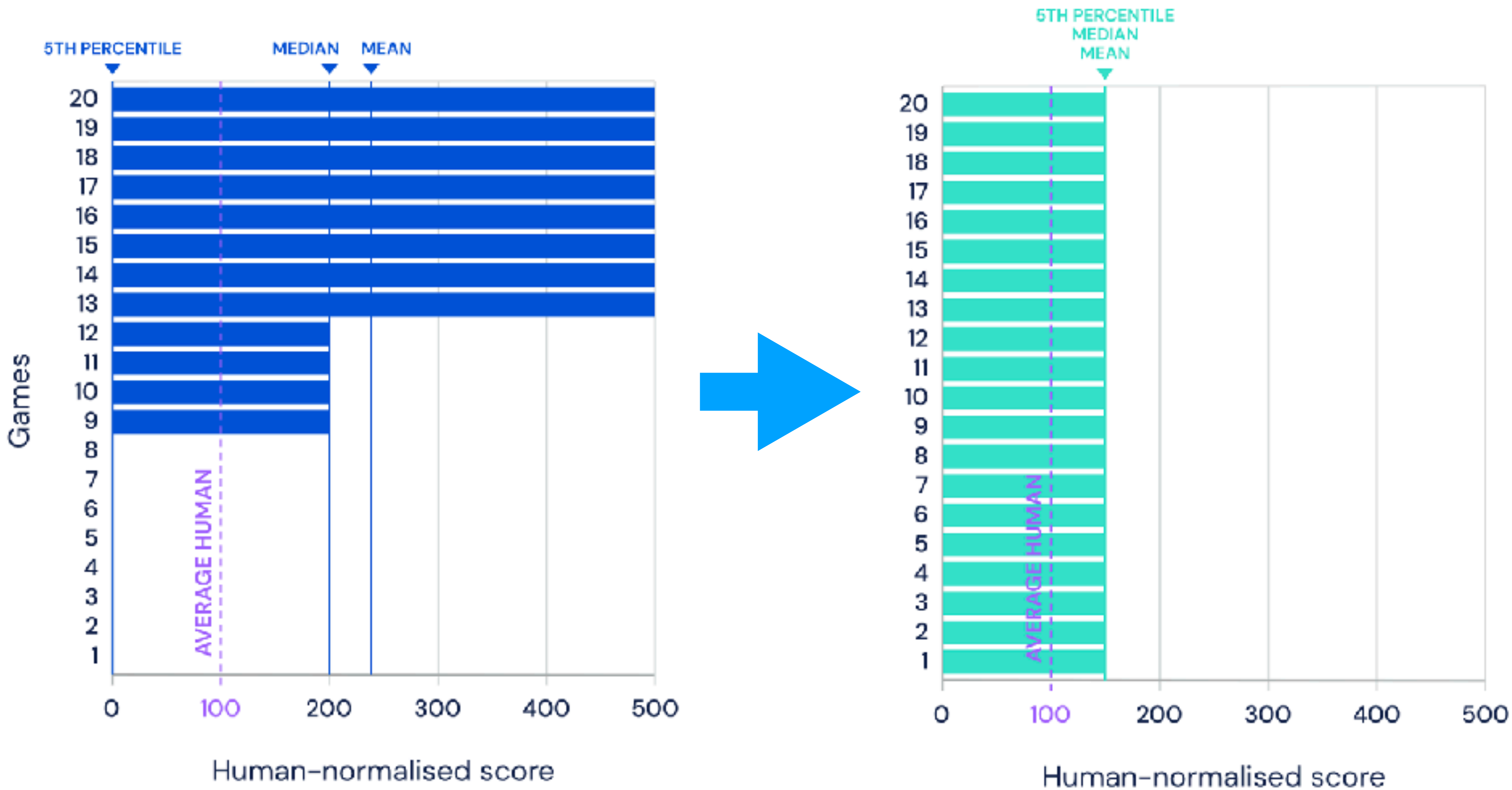
Agent 57 overview



Some results

Statistics	Agent57	R2D2 (bandit)	NGU	R2D2 (Retrace)	R2D2	MuZero
Number of games > human	**57**	54	51	52	52	51
Mean	4766.25	5461.66	3421.80	3518.36	4622.09	**5661.84**
Median	1933.49	2357.92	1359.78	1457.63	1935.86	**2381.51**

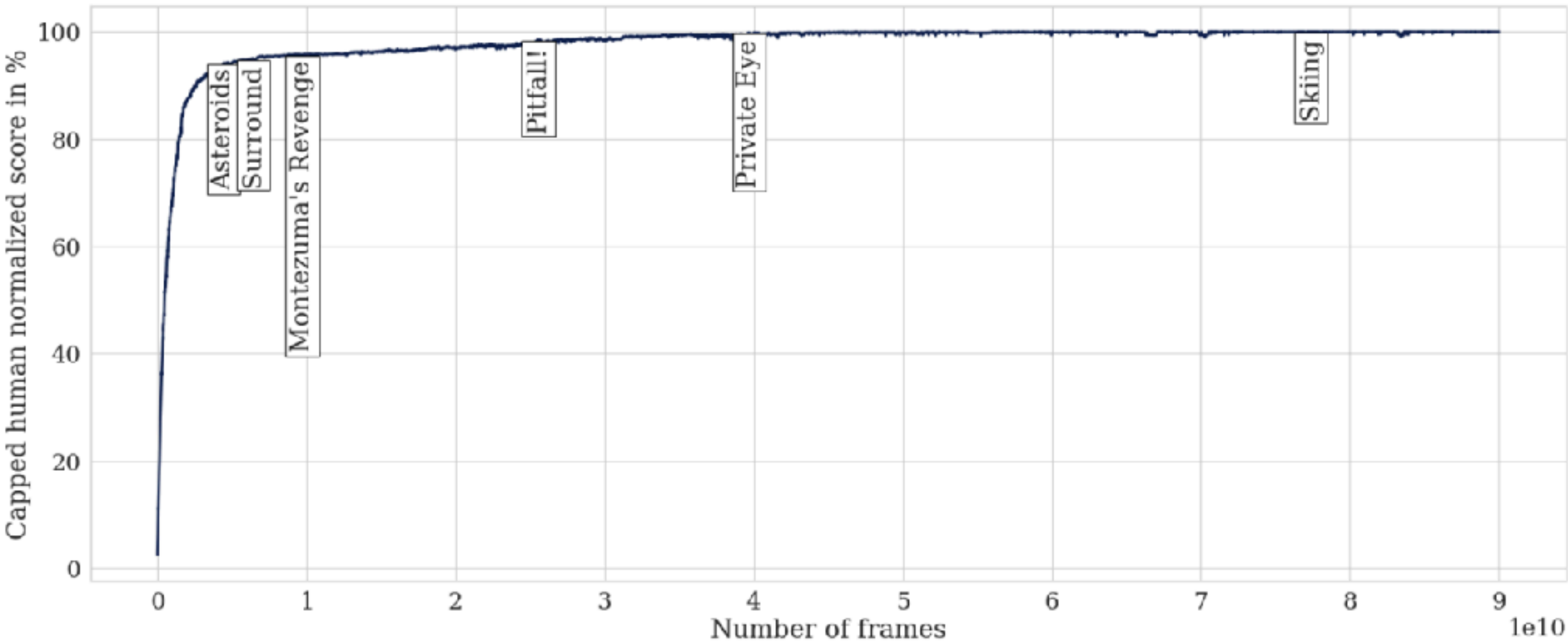
On the importance of an appropriate choice of the quality measure…



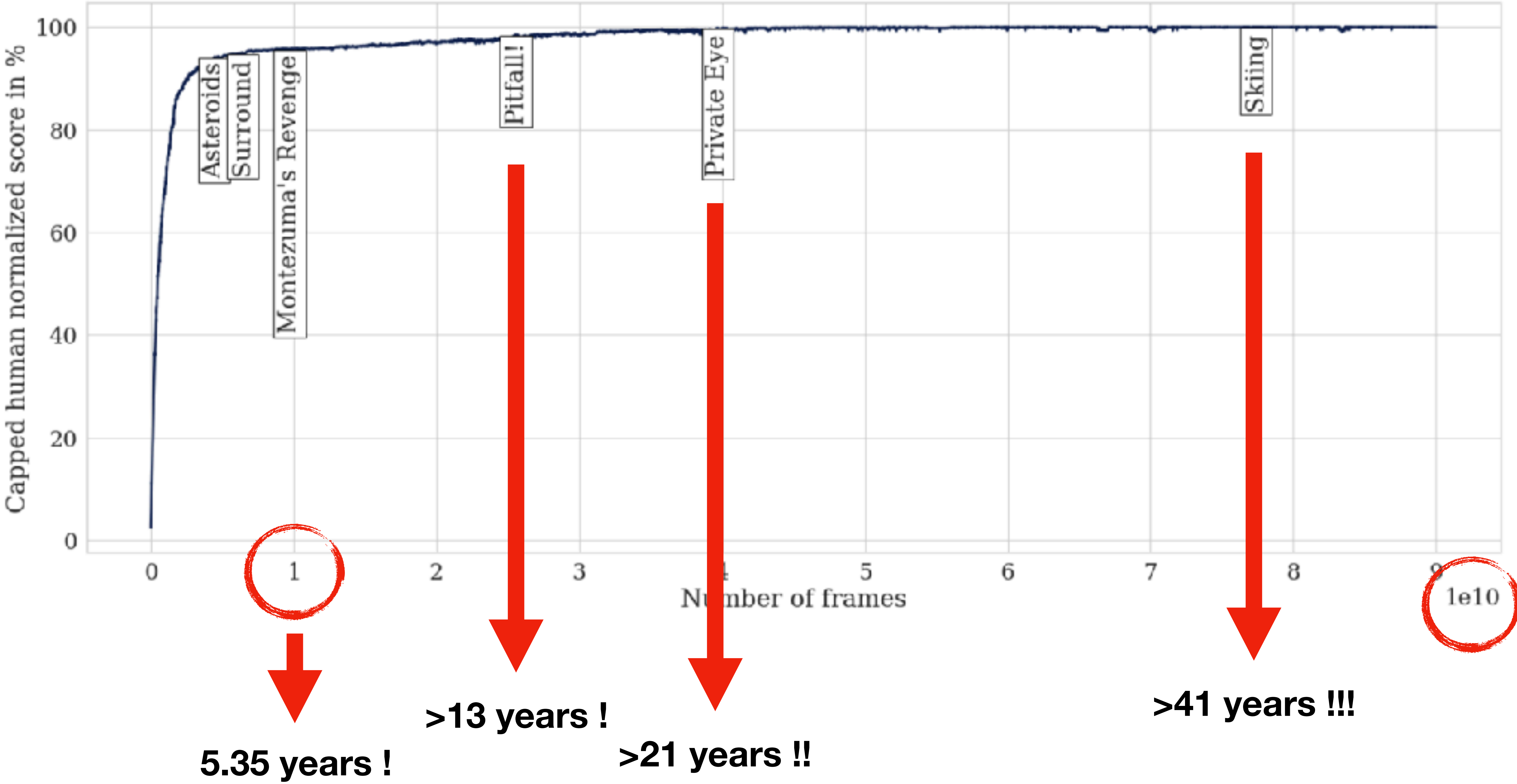
Some results

Statistics	Agent57	R2D2 (bandit)	NGU	R2D2 (Retrace)	R2D2	MuZero
Capped mean	**100.00**	96.93	95.07	94.20	94.33	89.92
Number of games > human	**57**	54	51	52	52	51
Mean	4766.25	5461.66	3421.80	3518.36	4622.09	**5661.84**
Median	1933.49	2357.92	1359.78	1457.63	1935.86	**2381.51**
40th Percentile	1091.07	**1298.80**	610.44	817.77	1176.05	1172.90
30th Percentile	614.65	**648.17**	267.10	420.67	529.23	503.05
20th Percentile	**324.78**	303.61	226.43	267.25	215.31	171.39
10th Percentile	**184.35**	116.82	107.78	116.03	115.33	75.74
5th Percentile	**116.67**	93.25	64.10	48.32	50.27	0.03

Some results



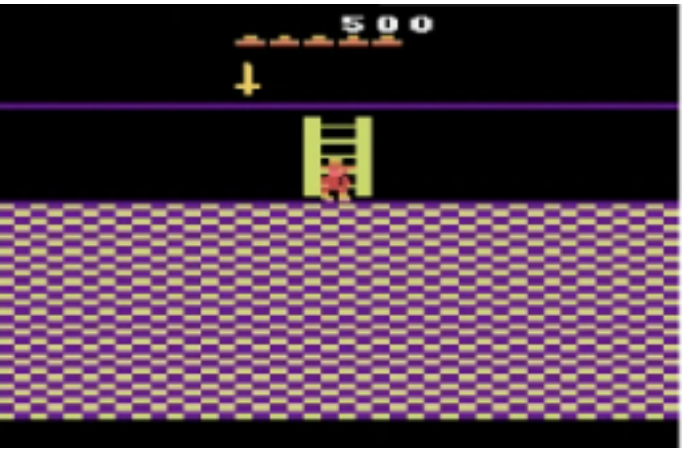
Some results

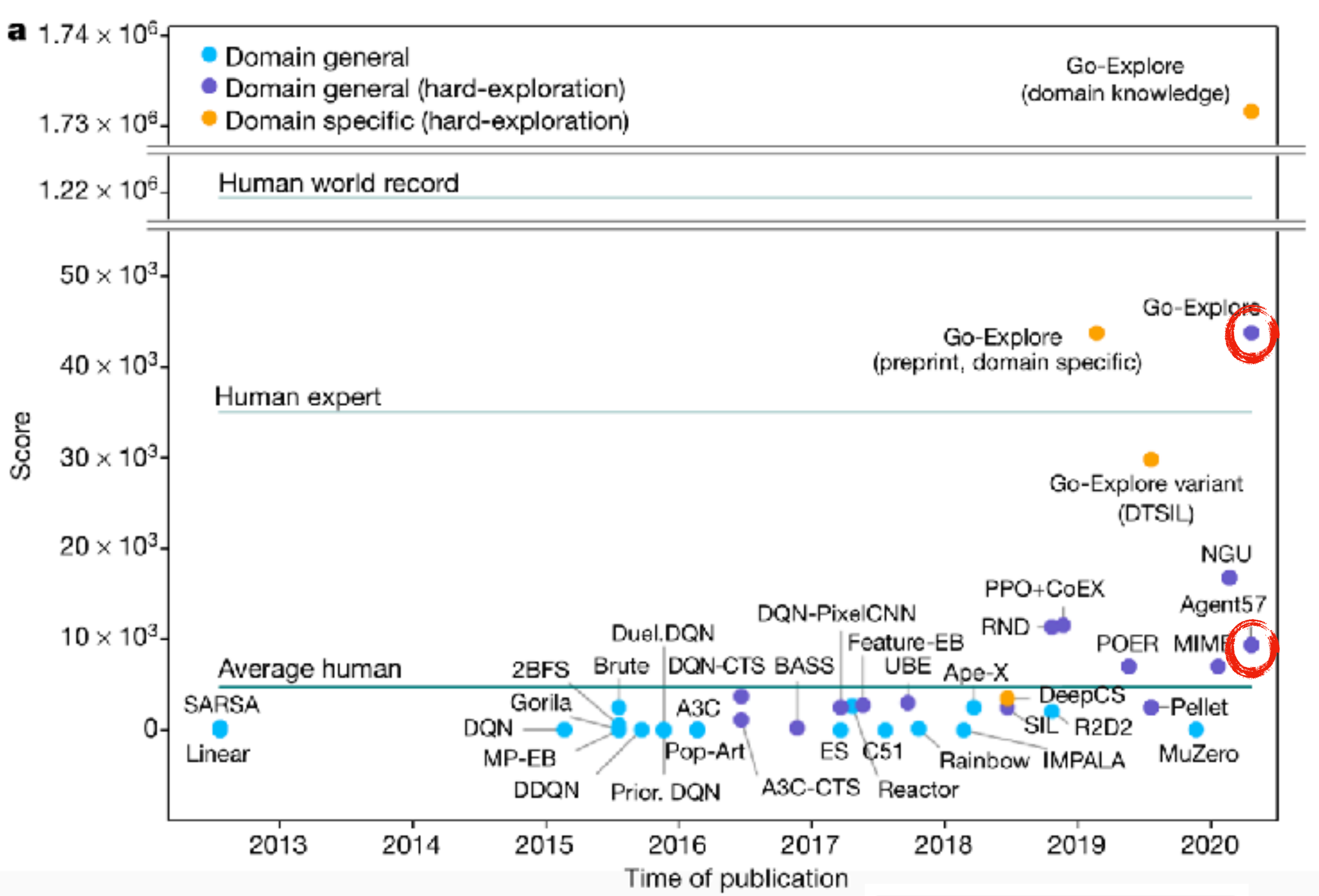


at 60 frames per second (averaged over 6 seeds)

Discussion

- Useful for robotics ? Some questions:
    - How does it scale to higher number of actions ? Continuous actions ?
    - Can it scale to more realistic images ?
    - How robust is it to perturbations ?
    - To what extent can the data efficiency be improved ?
- Is it possible to transfer the knowledge acquired ?





nature

Explore Content ⌄ Journa

nature › articles › article

Article | Published: 24 February 2021

First return, then explore

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley & Jeff Clune

Nature **590**, 580–586(2021) | Cite this article

Thank you !