< Blog

BLOG POST
RESEARCH

31 MAR 2020

Agent57: Outperforming the human Atari benchmark

The Atari57 suite of games is a long-standing benchmark to gauge agent performance across a wide range of tasks. We've developed Agent57, the first deep reinforcement learning agent to obtain a score that is above the human baseline on all 57 Atari 2600

PLAYLIST: AGENT57 PLAYING ALL GAMES IN THE ATARI57 BENCHMARK

How to measure Artificial General Intelligence?

At DeepMind, we're interested in building agents that do well on a wide range of tasks. An agent that performs *sufficiently well* on a *sufficiently wide* range of tasks is classified as intelligent. Games are an excellent testing ground for building adaptive algorithms: they provide a rich suite of tasks which players must develop sophisticated behavioural strategies to master, but they also provide an easy progress metric – game score – to optimise against. The ultimate goal is not to develop systems that excel at games, but rather to use games as a stepping stone for developing systems that learn to excel at a broad set of challenges. Typically, human performance is taken as a baseline for what doing "sufficiently well" on a task means: the score obtained by an agent on each task can be measured relative to representative human performance, providing a human normalised score: 0% indicates that an agent performs at random, while 100% or above indicates the agent is performing at human level or better.

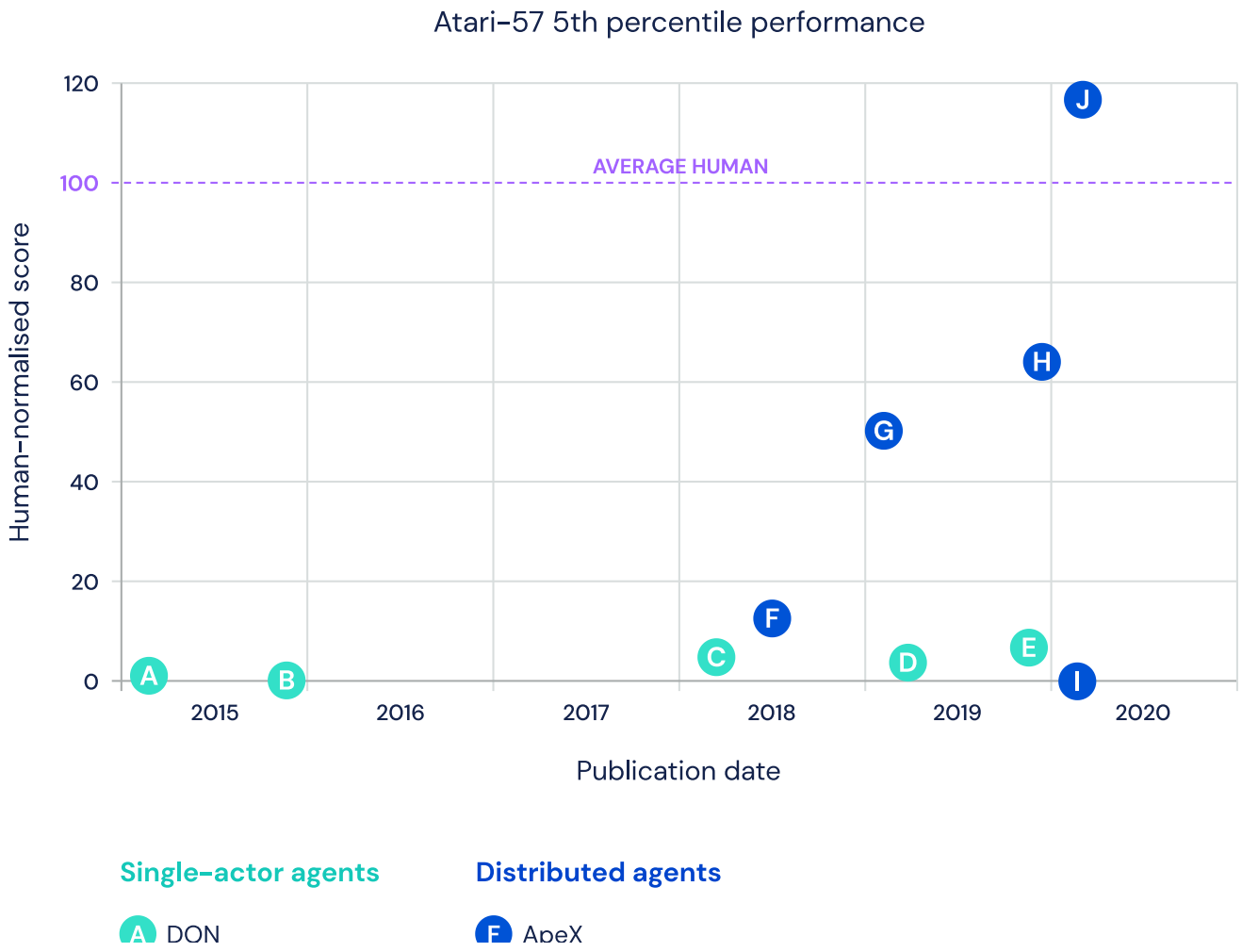
In 2012, the Arcade Learning environment – a suite of 57 Atari 2600 games (dubbed Atari57) – was proposed as a benchmark set of tasks: these canonical Atari games pose a broad range of challenges for an agent to master. The research community commonly uses this benchmark to measure progress in building successively more intelligent agents. It's often desirable to summarise the performance of an agent on a wide range of tasks as a single number, and so average performance (either mean or median score

0% on eight tasks (mean = 240%, median = 200%), while agent B obtains a score of 150% on all tasks (mean = median = 150%). On average, agent A performs better than agent B. However, agent B possesses a more general ability: it obtains human-level performance on more tasks than agent A.

FIGURE 1: ILLUSTRATION OF THE MEAN, MEDIAN AND 5TH PERCENTILE PERFORMANCE OF TWO HYPOTHETICAL AGENTS ON THE SAME BENCHMARK SET OF 20 TASKS.

This issue is exacerbated if some tasks are much easier than others. By performing very well on very easy tasks, agent A can apparently outperform agent B, which performs well on both easy and hard tasks.

the past eight years. But, like the illustrative example above, not all Atari games are equal, with some games being much easier than others. Instead of examining the average performance, if we examine the performance of agents on the bottom 5% of games, we see that not much has changed since 2012: in fact, agents published in 2019 were struggling on the same games with which agents published in 2012 struggled. Agent57 changes this, and is a more general agent in Atari57 than any agent since the inception of the benchmark. Agent57 finally obtains above human-level performance on the very hardest games in the benchmark set, as well as the easiest ones.

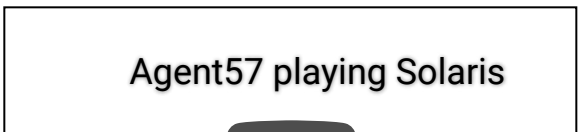


Agent57 ancestry

Back in 2012, DeepMind developed the Deep Q-network agent (DQN) to tackle the Atari57 suite. Since then, the research community has developed many extensions and alternatives to DQN. Despite these advancements, however, all deep reinforcement learning agents have consistently failed to score in four games: Montezuma's Revenge, Pitfall, Solaris and Skiing.

Montezuma's Revenge and Pitfall require extensive exploration to obtain good performance. A core dilemma in learning is the exploration-exploitation problem: should one keep performing behaviours one knows works (exploit), or should one try something new (explore) to discover new strategies that might be even more successful? For example, should one always order their same favourite dish at a local restaurant, or try something new that might surpass the old favourite? Exploration involves taking many suboptimal actions to gather the information necessary to discover an ultimately stronger behaviour.

Solaris and Skiing are long-term credit assignment problems: in these games, it's challenging to match the consequences of an agents' actions to the rewards it receives. Agents must collect information over long time scales to get the feedback necessary to learn.



For Agent57 to tackle these four challenging games in addition to the other Atari57 games, several changes to DQN were necessary.

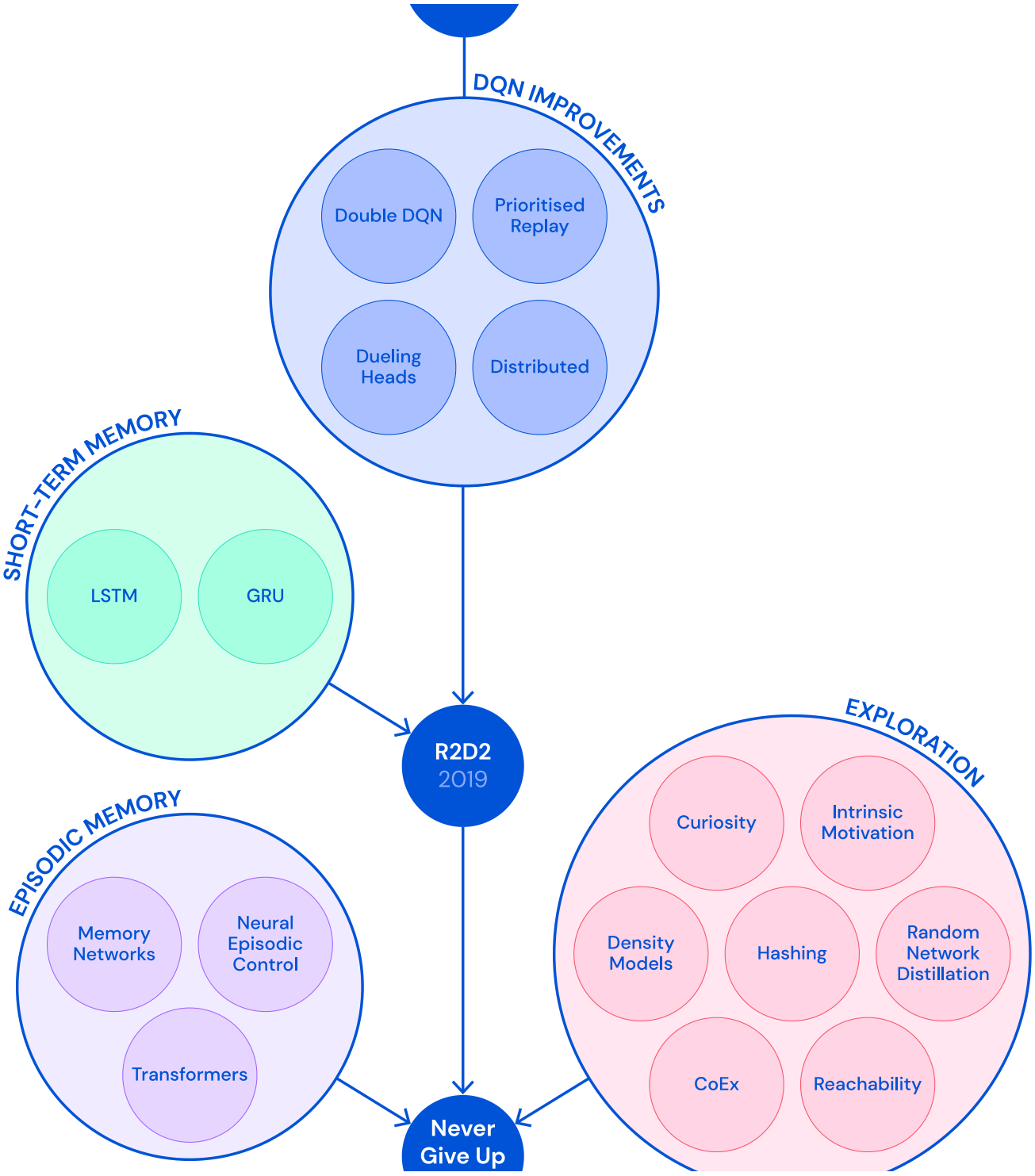


FIGURE 3. CONCEPTUAL ADVANCEMENTS TO DQN THAT HAVE RESULTED IN THE DEVELOPMENT OF MORE GENERALLY INTELLIGENT AGENTS.

DQN improvements

Early improvements to DQN enhanced its learning efficiency and stability, including double DQN, prioritised experience replay and dueling architecture. These changes allowed agents to make more efficient and effective use of their experience.

Distributed agents

Next, researchers introduced **distributed** variants of DQN, Gorila DQN and ApeX, that could be run on many computers simultaneously. This allowed agents to acquire and learn from experience more quickly, enabling researchers to rapidly iterate on ideas. Agent57 is also a distributed RL agent that decouples the data collection and the learning processes. Many actors interact with independent copies of the environment, feeding data to a central 'memory bank' in the form of a prioritized replay buffer. A learner then samples training data from this replay buffer, as shown in Figure 4, similar to how a person might recall memories to better learn from them. The learner uses these replayed experiences to construct loss functions, by which it estimates the cost of actions or events. Then, it updates the parameters of its neural network by minimizing losses. Finally,

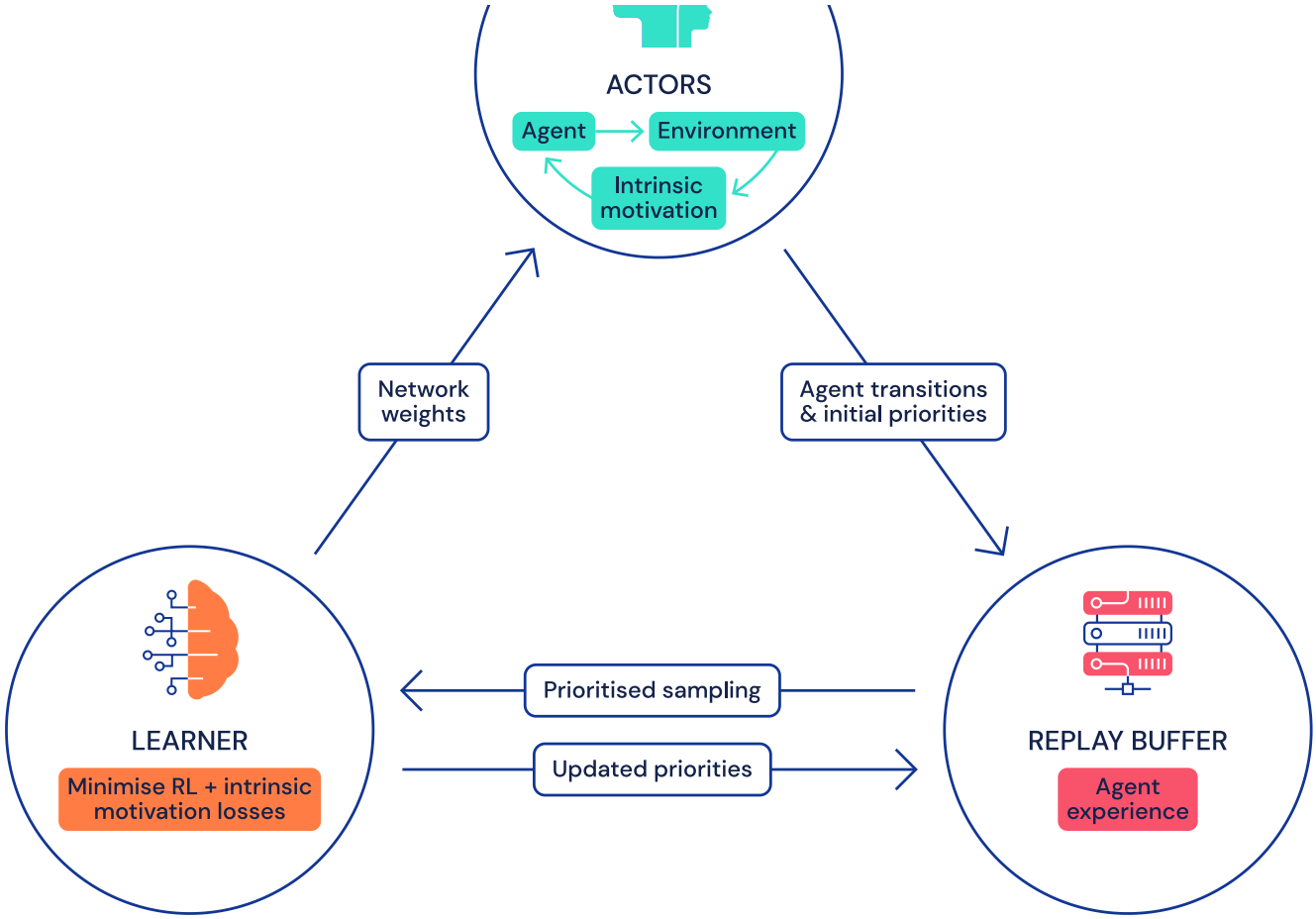


FIGURE 4. DISTRIBUTED SETUP FOR AGENT 57.

Short-term memory

Agents need to have memory in order to take into account previous observations into their decision making. This allows the agent to not only base its decisions on the present observation (which is usually partial, that is, an agent only sees some of its world), but

Interfacing memory with behaviour is crucial for building systems that self-learn. In reinforcement learning, an agent can be an on-policy learner, which can only learn the value of its direct actions, or an off-policy learner, which can learn about optimal actions even when not performing those actions – e.g., it might be taking random actions, but can still learn what the best possible action would be. Off-policy learning is therefore a desirable property for agents, helping them learn the best course of action to take while thoroughly exploring their environment. Combining off-policy learning with memory is challenging because you need to know what you might remember when executing a different behaviour. For example, what you might choose to remember when looking for an apple (e.g., where the apple is located), is different to what you might choose to remember if looking for an orange. But if you were looking for an orange, you could still learn how to find the apple if you came across the apple by chance, in case you need to find it in the future. The first deep RL agent combining memory and off-policy learning was Deep Recurrent Q-Network (DRQN). More recently, a significant speciation in the lineage of Agent57 occurred with Recurrent Replay Distributed DQN (R2D2), combining a neural network model of short-term memory with off-policy learning and distributed training, and achieving a very strong average performance on Atari57. R2D2 modifies the replay mechanism for learning from past experiences to work with short term memory. All together, this helped R2D2 efficiently learn profitable behaviours, and **exploit** them for reward.

Episodic memory

We designed Never Give Up (NGU) to augment R2D2 with another form of memory:

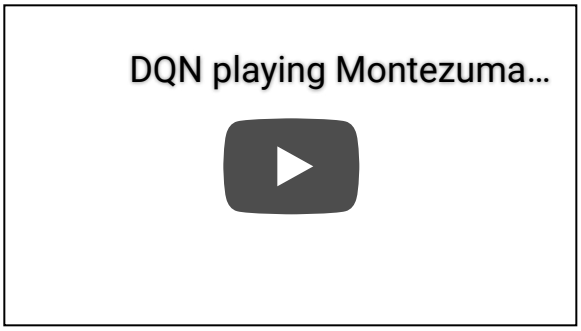
Intrinsic motivation methods to encourage directed exploration

In order to discover the most successful strategies, agents must explore their environment–but some exploration strategies are more efficient than others. With DQN, researchers attempted to address the exploration problem by using an undirected exploration strategy known as epsilon-greedy: with a fixed probability (epsilon), take a random action, otherwise pick the current best action. However, this family of techniques do not scale well to hard exploration problems: in the absence of rewards, they require a prohibitive amount of time to explore large state-action spaces, as they rely on undirected random action choices to discover unseen states. In order to overcome this limitation, many directed exploration strategies have been proposed. Among these, one strand has focused on developing **intrinsic motivation rewards** that encourage an agent to explore and visit as many states as possible by providing more dense "internal" rewards for novelty-seeking behaviours. Within that strand, we distinguish two types of rewards: firstly, *long-term novelty* rewards encourage visiting many states throughout training, across many episodes. Secondly, *short-term novelty* rewards encourage visiting many states over a short span of time (e.g., within a single episode of a game).

Seeking novelty over long time scales

Long-term novelty rewards signal when a previously unseen state is encountered in the

new experiences), as well as an inability to produce precise outputs for all inputs. For example, in Montezuma's Revenge, unlike undirected exploration strategies, long-term novelty rewards allow the agent to surpass the human baseline. However, even the best performing methods on Montezuma's Revenge need to carefully train a density model at the *right* speed: when the density model indicates that the states in the first room are *familiar*, the agent should be able to consistently get to unfamiliar territory.



PLAYLIST: DQN VS. AGENT57 PLAYING MONTEZUMA'S REVENGE

Seeking novelty over short time scales

Short-term novelty rewards can be used to encourage an agent to explore states that have not been encountered in its recent past. Recently, neural networks that mimic some properties of episodic memory have been used to speed up learning in reinforcement learning agents. Because episodic memories are also thought to be important for recognising novel experiences, we adapted these models to give Never Give Up a notion of short-term novelty. Episodic memory models are efficient and reliable candidates for

If an agent is programmed to use a notion of distance wherein every tiny visual variation is taken into account, that agent would visit a large number of different states simply by passively observing the environment, even standing still – a fruitless form of exploration. To avoid this scenario, the agent should instead learn features that are seen as important for exploration, such as controllability, and compute a distance with respect to those features only. Such models have previously been used for exploration, and combining them with episodic memory is one of the main advancements of the Never Give Up exploration method, which resulted in above-human performance in Pitfall!.

PLAYLIST: NGU VS. AGENT57 PLAYING PITFALL!

Never Give Up (NGU) used this short-term novelty reward based on controllable states, mixed with a long term novelty reward, using Random Network Distillation. The mix was achieved by multiplying both rewards, where the long term novelty is bounded. This way the short-term novelty reward's effect is preserved, but can be down-modulated as the agent becomes more familiar with the game over its lifetime. The other core idea of NGU is that it learns a family of policies that range from purely exploitative to highly exploratory. This is achieved by leveraging a distributed setup: by building on top of R2D2, actors produce experience with different policies based on different importance

that adapts the exploration-exploitation trade-off, as well as a time horizon that can be adjusted for games requiring longer temporal credit assignment. With this change, Agent57 is able to get the best of both worlds: above human-level performance on both easy games and hard games.

Specifically, intrinsic motivation methods have two shortcomings:

- **Exploration:** Many games are amenable to policies that are purely exploitative, particularly after a game has been fully explored. This implies that much of the experience produced by exploratory policies in Never Give Up will eventually become wasteful after the agent explores all relevant states.
- **Time horizon:** Some tasks will require long time horizons (e.g. Skiing, Solaris), where valuing rewards that will be earned in the far future might be important for eventually learning a good exploitative policy, or even to learn a good policy at all. At the same time, other tasks may be slow and unstable to learn if future rewards are overly weighted. This trade-off is commonly controlled by the discount factor in reinforcement learning, where a higher discount factor enables learning from longer time horizons.

This motivated the use of an online adaptation mechanism that controls the amount of experience produced with different policies, with a variable-length time horizon and importance attributed to novelty. Researchers have tried tackling this with multiple methods, including training a population of agents with different hyperparameter values, directly learning the values of the hyperparameters by gradient descent, or using a centralized bandit to learn the value of hyperparameters.

We used a bandit algorithm to select which policy our agent should use to generate

PLAYLIST: NGU VS. AGENT57 PLAYING SKIING

Agent57: putting it all together

To achieve Agent57, we combined our previous exploration agent, Never Give Up, with a meta-controller. This agent computes a mixture of long and short term intrinsic motivation to explore and learn a family of policies, where the choice of policy is selected by the meta-controller. The meta-controller allows each actor of the agent to choose a different trade-off between near vs. long term performance, as well as exploring new states vs. exploiting what's already known (Figure 4). Reinforcement learning is a feedback loop: the actions chosen determine the training data. Therefore, the meta-controller also determines what data the agent learns from.

Statisitics	Agent57	NGU	R2D2	MuZero
Number of games > human	57	51	52	51
Mean HNS	4766.25%	3421.80%	4622.09%	**5661.84%**
Median HNS	1933.49%	1359.78%	1935.86%	**2381.51%**
5th percentile of HNS	**116.67%**	64.10%	50.27%	0.03%

Conclusions and the future

Agent57 was able to scale with increasing amounts of computation: the longer it trained, the higher its score got. While this enabled Agent57 to achieve strong general performance, it takes a lot of computation and time; the data efficiency can certainly be improved. Additionally, this agent shows better 5th percentile performance on the set of Atari57 games. This by no means marks the end of Atari research, not only in terms of data efficiency, but also in terms of general performance. We offer two views on this: firstly, analyzing the performance among percentiles gives us new insights on how general algorithms are. While Agent57 achieves strong results on the first percentiles of the 57 games and holds better mean and median performance than NGU or R2D2, as illustrated by MuZero, it could still obtain a higher average performance. Secondly, all current algorithms are far from achieving optimal performance in some games. To that end, key improvements to use might be enhancements in the representations that Agent57 uses for exploration, planning, and credit assignment.

Read the paper here.

Work done by: Adrià Puigdomènech, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Charles Blundell

Figure design by Paulo Estriga and Adam Cain

References

- Agent57 lineage:

  - Prioritised replay: Schaul, Tom, et al. "Prioritized experience replay." arXiv preprint arXiv:1511.05952 (2015).
  - Apex: Horgan, Dan, et al. "Distributed prioritized experience replay." arXiv preprint arXiv:1803.00933 (2018).
  - R2D2: Kapturowski, Steven, et al. "Recurrent experience replay in distributed reinforcement learning." ICLR (2019).
  - NGU: Badia, Adrià Puigdomènech, et al. "Never Give Up: Learning Directed Exploration Strategies." ICLR (2020).

- Episodic Memory related:

  - Memory Networks: Weston, Jason, Sumit Chopra, and Antoine Bordes. "Memory networks." arXiv preprint arXiv:1410.3916 (2014).
  - Neural Episodic Control: Pritzel, Alexander, et al. "Neural episodic control." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.
  - Transformer: Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
  - Wayne, Greg, et al. "Unsupervised predictive memory in a goal-directed agent." *arXiv preprint arXiv:1803.10760* (2018).

- Exploration related:

  - Curiosity: Schmidhuber, Jürgen. "A possibility for implementing curiosity and boredom in model-building neural controllers." Proc. of the international conference on simulation of adaptive behavior: From animals to animats. 1991.

- Density models: Ostrovski, Georg, et al. "Count-based exploration with neural density models." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.
- Ex2: Fu, Justin, John Co-Reyes, and Sergey Levine. "Ex2: Exploration with exemplar models for deep reinforcement learning." Advances in neural information processing systems. 2017.
- Hashing: Tang, Haoran, et al. "# exploration: A study of count-based exploration for deep reinforcement learning." Advances in neural information processing systems. 2017.
- Random Network Distillation: Burda, Yuri, et al. "Exploration by random network distillation." arXiv preprint arXiv:1810.12894 (2018).
- CoEx: Choi, Jongwook, et al. "Contingency-aware exploration in reinforcement learning." arXiv preprint arXiv:1811.01483 (2018).
- Reachability: Savinov, Nikolay, et al. "Episodic curiosity through reachability." ICLR, 2019.
- Never Give Up: Puigdomènech Badia, Adrià, et al. "Never Give Up: Learning Directed Exploration Strategies." arXiv (2020): arXiv-2002.

- Meta-controller related:

  - Population-based training: Jaderberg, Max, et al. "Population based training of neural networks." arXiv preprint arXiv:1711.09846 (2017).
  - Meta-gradients: Xu, Zhongwen, Hado P. van Hasselt, and David Silver. "Meta-gradient reinforcement learning." Advances in neural information processing systems. 2018.
  - Bandits: Schaul, Tom, et al. "Adapting Behaviour for Learning Progress." arXiv

AP Adrià Puigdomènech

BP Bilal Piot

SK Steven Kapturowski

PS Pablo Sprechmann

AV Alex Vitvitskyi

DG Daniel Guo

CB Charles Blundell

FURTHER READING

Deep Reinforcement Learning

Multi-Task Learning

Overview

Technical safety

Ethics & Society

Careers

Search open roles

A unique mission

Teams

Diversity matters

Locations

Internships

PRESS

TERMS & CONDITIONS

PRIVACY POLICY

MODERN SLAVERY STATEMENT

ALPHABET INC