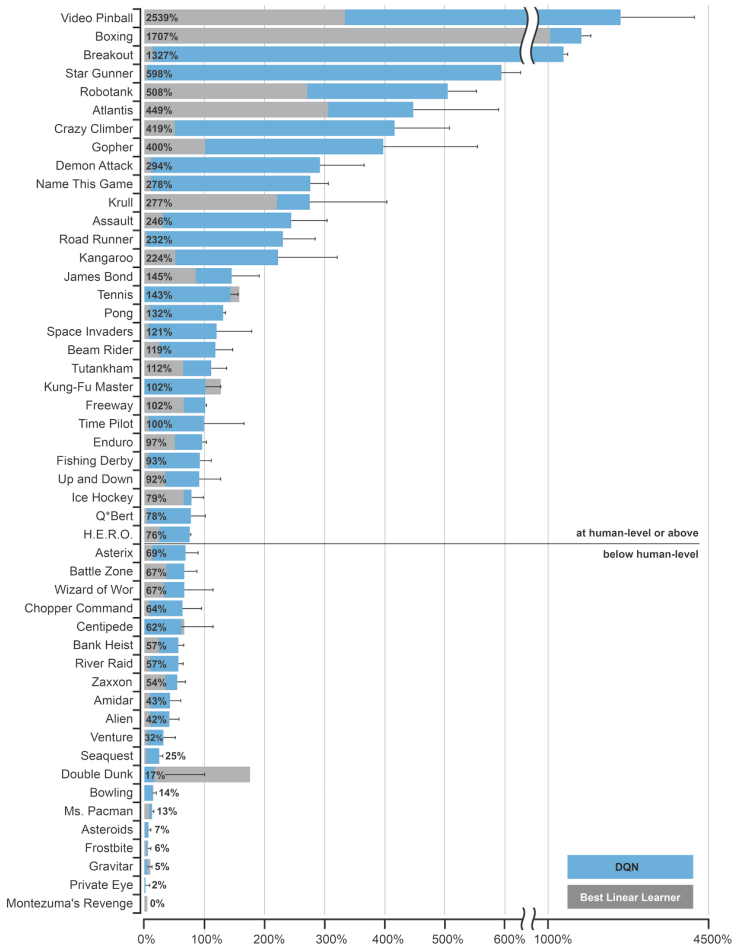
Humans excel at solving a wide variety of challenging problems, from low-level motor control through to high-level cognitive tasks. Our goal at DeepMind is to create artificial agents that can achieve a similar level of performance and generality. Like a human, our agents learn for themselves to achieve successful strategies that lead to the greatest long-term rewards. This paradigm of learning by trial-and-error, solely from rewards or punishments, is known as **reinforcement learning** (RL). Also like a human, our agents construct and learn their own knowledge directly from raw inputs, such as vision, without any hand-engineered features or domain heuristics. This is achieved by **deep learning** of neural networks. At DeepMind we have pioneered the combination of these approaches - deep reinforcement learning - to create the first artificial agents to achieve human-level performance across many challenging domains.

Our agents must continually make value judgements so as to select good actions over bad. This knowledge is represented by a Q-network that estimates the total reward that an agent can expect to receive after taking a particular action. Two years ago we introduced the first widely successful **algorithm** for deep reinforcement learning. The key idea was to use deep neural networks to represent the Q-network, and to train this Q-network to predict total reward. Previous attempts to combine RL with neural networks had largely failed due to unstable learning. To address these instabilities, our Deep Q-Networks (DQN) algorithm stores all of the agent's experiences and then randomly samples and replays these experiences to provide diverse and decorrelated training data. We applied DQN to learn to play games on the

Atari 2600 console. At each time-step the agent observes the raw pixels on the screen, a reward signal corresponding to the game score, and selects a joystick direction. In our **Nature paper** we trained separate DQN agents for 50 different Atari games, without any prior knowledge of the game rules.

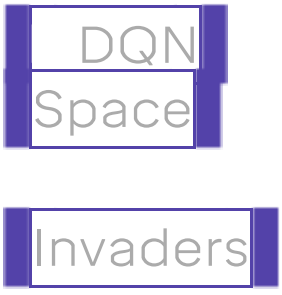


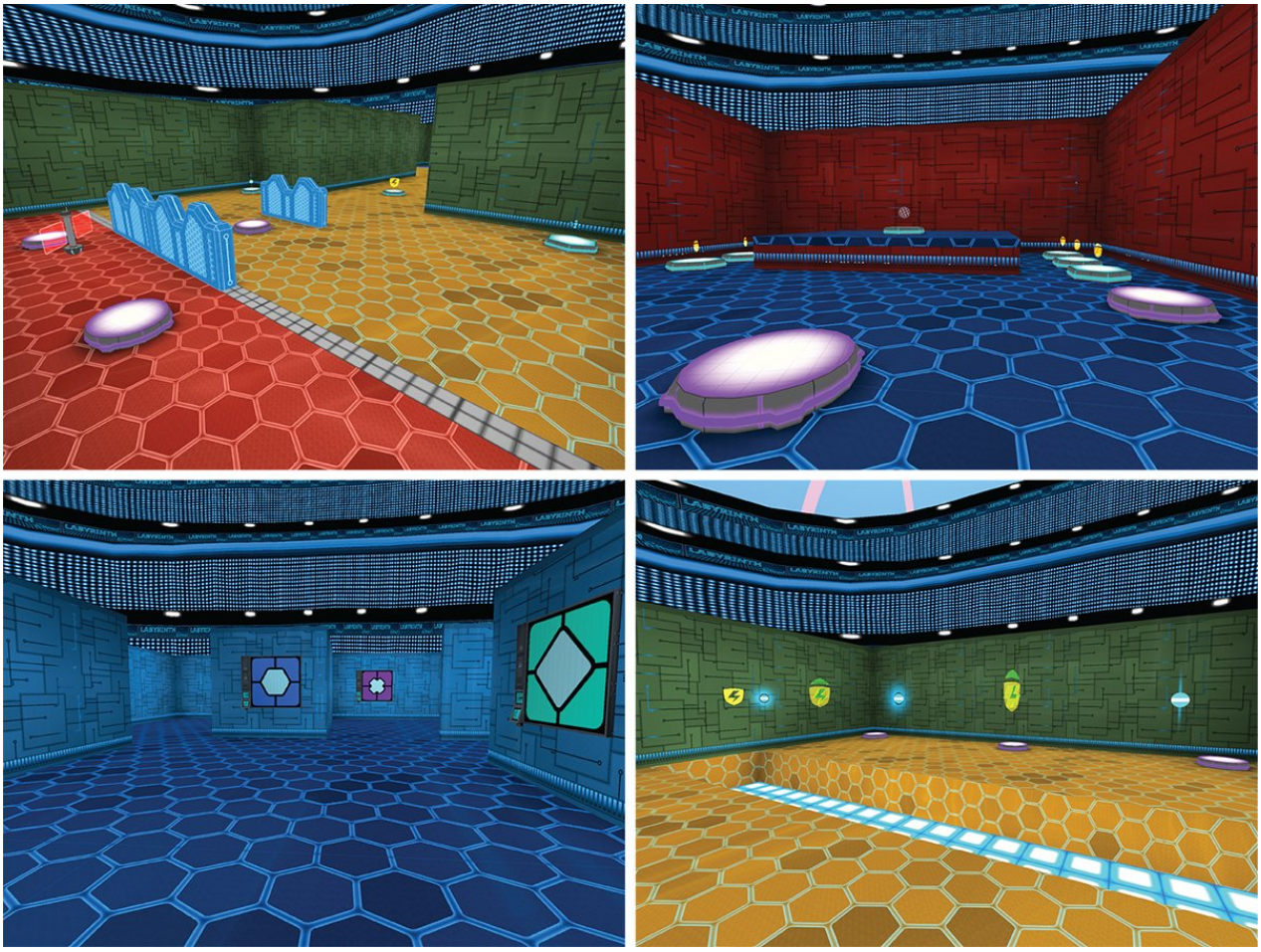
Amazingly, DQN achieved human-level performance in almost half of the 50 games to which it was applied; far beyond any previous method. The **DQN source code** and **Atari 2600 emulator** are freely available to anyone who wishes to experiment for themselves.

We have subsequently improved the DQN algorithm in many ways: further stabilising the **learning dynamics**; prioritising the **replayed experiences**; **normalising**, **aggregating** and **re-scaling** the outputs. Combining several of these improvements together led to a 300% improvement in mean score across Atari games; human-level performance has now been achieved in almost all of the Atari games. We can even train a **single neural network** to learn about **multiple Atari games**. We have also built a massively distributed deep RL system, known as **Gorila**, that utilises the Google Cloud platform to speed up training time by an order of magnitude; this system has been applied to recommender systems within Google.

However, deep Q-networks are only one way to solve the deep RL problem. We recently introduced an even more practical and effective method based on asynchronous RL. This approach exploits the multithreading capabilities of standard CPUs. The idea is to execute many instances of our agent in parallel, but using a shared model. This provides a viable alternative to experience replay, since parallelisation also diversifies and decorrelates the data. Our asynchronous actor-critic algorithm, **A3C**, combines a deep Q-network with a deep policy network for selecting actions. It achieves state-of-the-art results, using a fraction of the training time of DQN and a fraction of the resource consumption of Gorila. By building novel approaches to **intrinsic motivation** and **temporally abstract planning**, we have also achieved breakthrough results in the most notoriously challenging Atari games, such as Montezuma's Revenge.

While Atari games demonstrate a wide degree of diversity, they are limited to 2D sprite-based video games. We have recently introduced Labyrinth: a challenging suite of 3D navigation and puzzle-solving environments. Again, the agent only observes pixel-based inputs from its immediate field-of-view, and must figure out the map to discover and exploit rewards.





Amazingly, the A3C algorithm achieves human-level performance, out-of-the-box, on many Labyrinth tasks. An **alternative approach** based on episodic memory has also proven successful. Labyrinth will also be released open source in the coming months.

Asynchronous Methods for Deep Reinforcement Learning: Labyrinth

Follow   

Research

Applied

News & Blog

About Us

Careers

Press
Terms and Conditions
Privacy Policy – Updated
Alphabet Inc