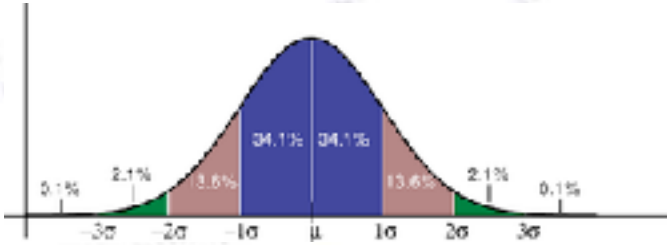
Applied ML

Reinforcement Learning

Troels C. Petersen (NBI)

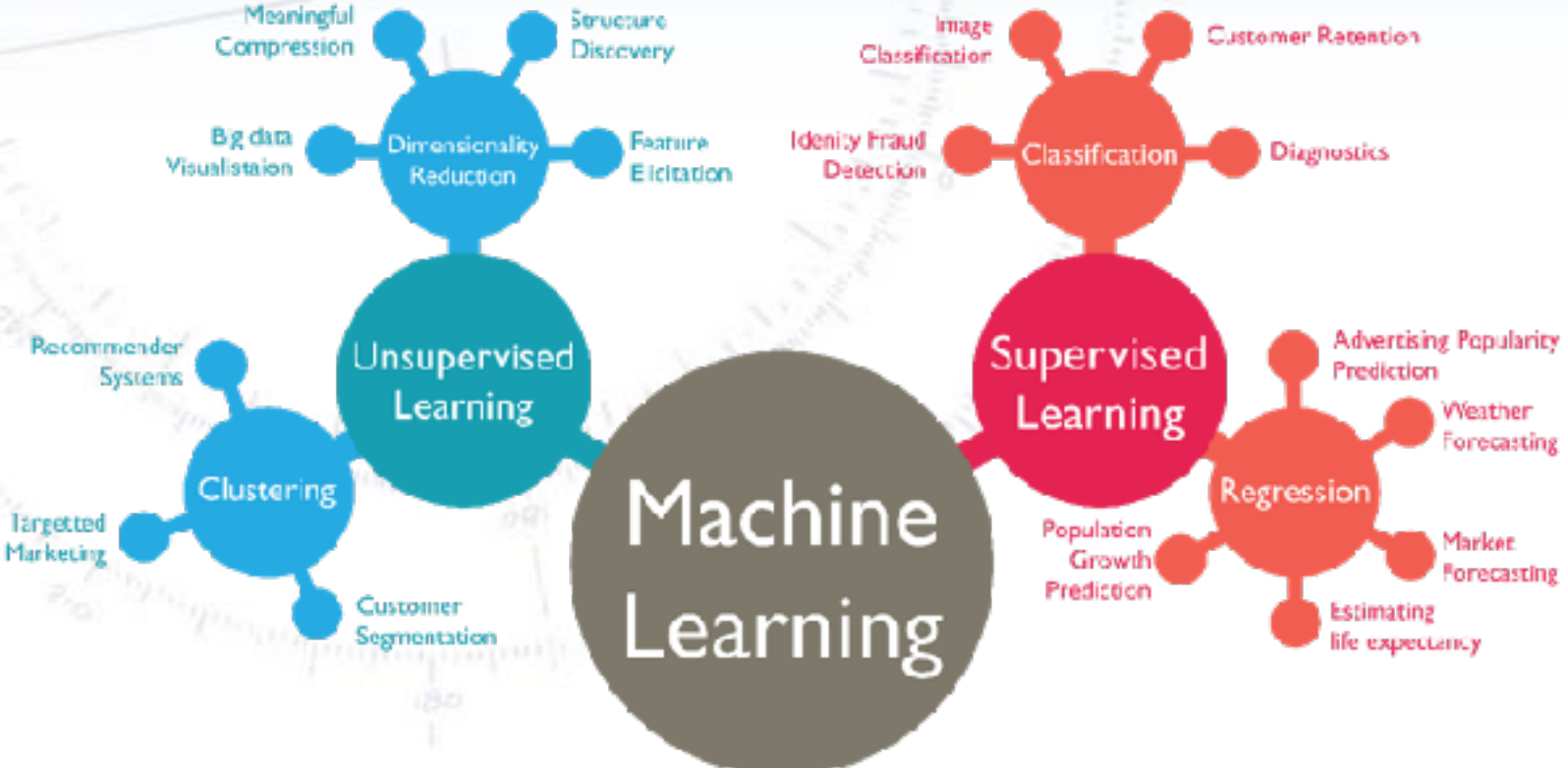


"Statistics is merely a quantisation of common sense - Machine Learning is a sharpening of it!"

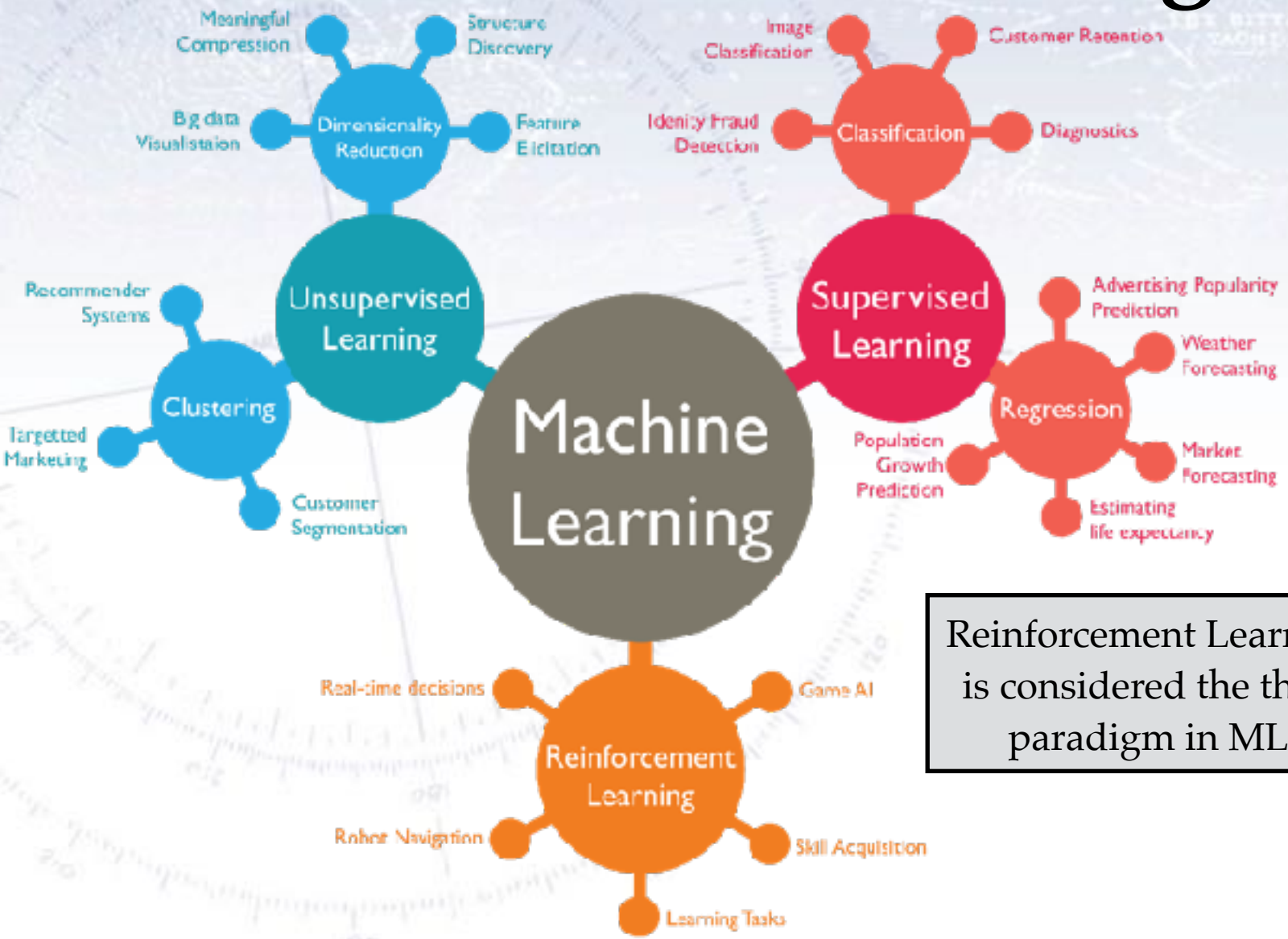
Classification vs. Regression

Unsupervised learning vs. supervised

Machine Learning can be supervised (you have correctly labelled examples) or unsupervised (you don't)… [or reinforced]. Following this, one can be using ML to either classify (is it A or B?) or for regression (estimate of X).

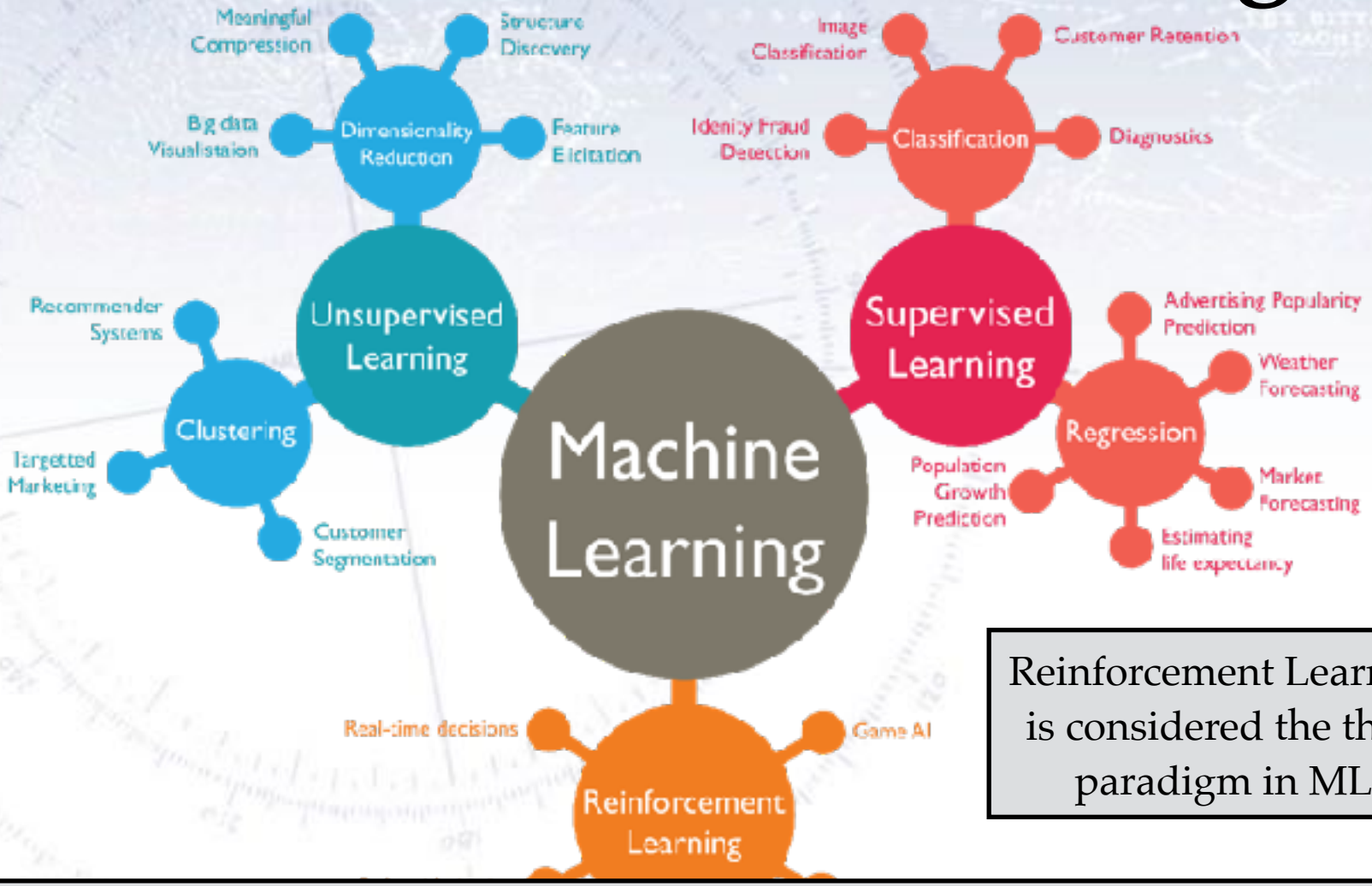


Reinforcement Learning



Reinforcement Learning is considered the third paradigm in ML.

Reinforcement Learning



Reinforcement Learning is considered the third paradigm in ML.

Essentially, the task is: "How should an agent (human or AI) act in this dynamic environment in order to optimise the cumulative score over time."

Reinforcement Learning

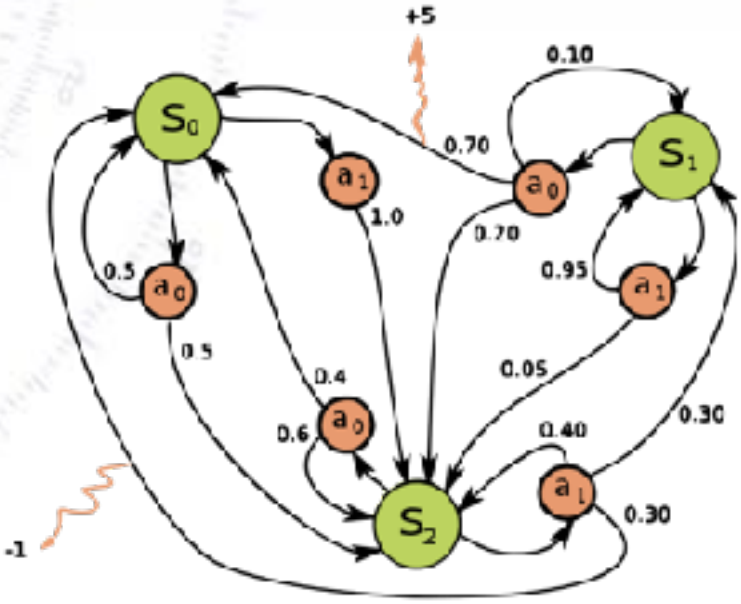
Reinforcement Learning (RL) does not rely on data, but rather an environment (a specific set of rules) in which it needs to optimise its actions/behaviour (model).

In doing so, the RL models needs to find a balance between exploration (of uncharted territory) and exploitation (of current knowledge), like all others.

The environment can be formulated as a Markov Decision Process (MDP), as shown below.

Reinforcement Learning does not assume knowledge of the MDP (i.e. it doesn't know what environment it is in - all it needs is a score).

And typically RF has great success in (potentially very) large environments, such as complicated games, but not in "real life", where the rules are unknown.



Formal requirements

Formally, RL Markov decision process is based on:

- A set of environment and agent states, $\mathbf{S_i}$
- A set of actions, **A**, of the agent
- $\mathbf{P_a(s, s') = Pr(S_t+1 \mid S_t = s, A_t = a)}$,
  the probability of transition (at time **t**) from state **s** to state **s'** under action **a**.
- $\mathbf{R_a(s, s')}$, the reward after transition from state **s** to state **s'** under action **a**.

The purpose of reinforcement learning is for the agent to learn an (near) optimal way of acting, that maximises the reward (loss) function.
Not unlike how animals/humans learning how to adapt to an environment!

It is assumed that the agent observes the "full" environmental state, in which case the problem is said to be **fully observable**. If the agent can only see a subset of states, or if the observed states are corrupted by noise, the agent is said to have **partial observability**, and formally the problem must be formulated as a Partially observable Markov decision process.

Formal requirements

In order to act (near) optimally, the agent must reason about the long-term consequences of its actions (i.e., maximise future income), *although the immediate reward associated with this might be negative*. Thus, reinforcement learning is particularly well-suited to problems that include a long-term versus short-term reward trade-off. It has been applied successfully to various problems, such as:

- Gaming (e.g. checkers (in 1956), backgammon, **chess, go**)
- Robot control (e.g. dexterity, **autonomous driving systems**)
- Realtime systems (e.g. energy storage, market response)

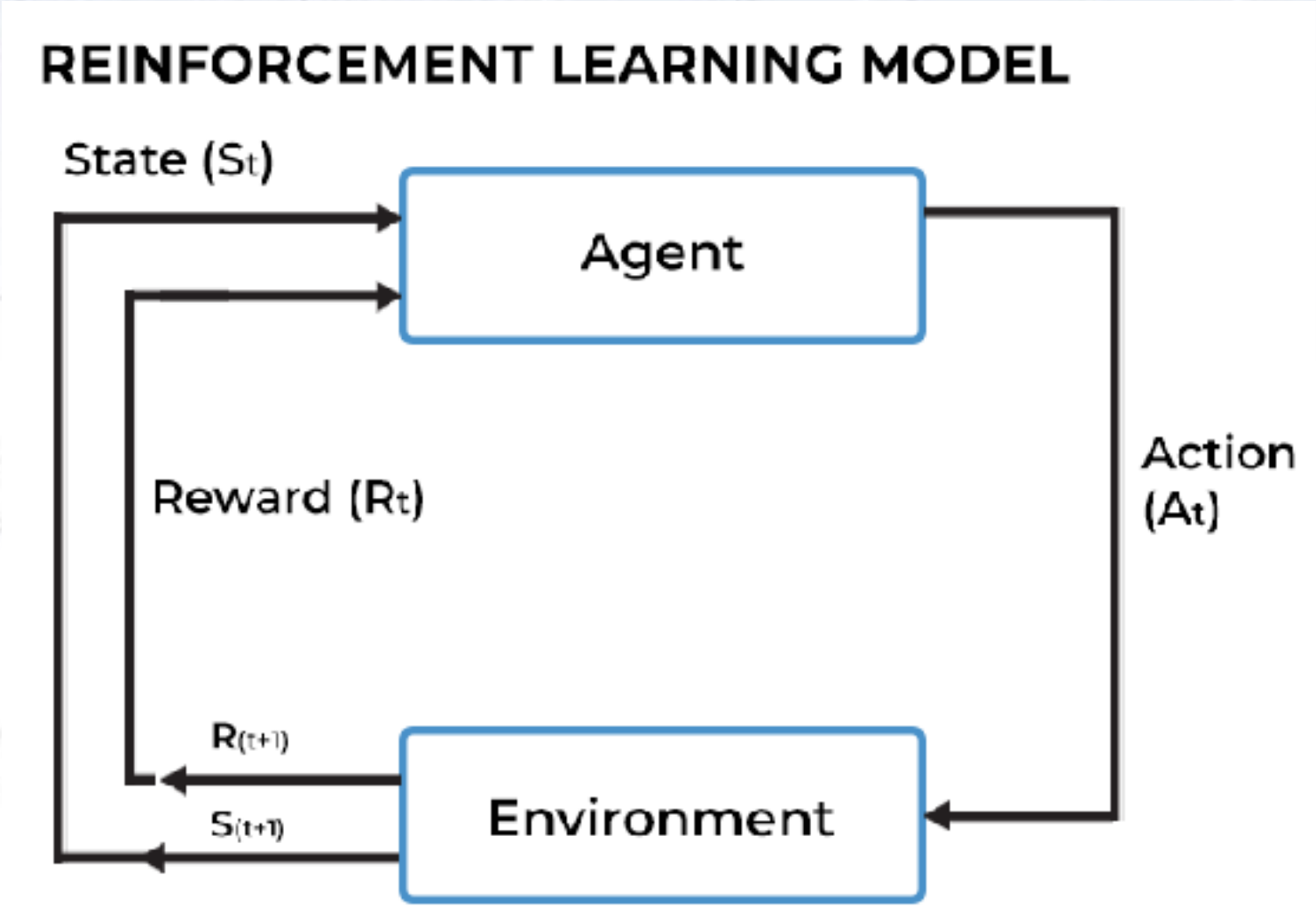
Two elements make reinforcement learning powerful: Use of **samples to optimise performance** and use of **functions to approximation of large environments**.
Thanks to these two key components, RL can be used in large environments when:

- A model of the environment is known (but not an analytic solution).
- A simulation model of the environment is given.
- The only way to collect information about the environment is to interact with it.

The first two involve a planning problem, while last one can be considered a genuine learning problem (partial observability). However, reinforcement learning converts these problems to machine learning problems.

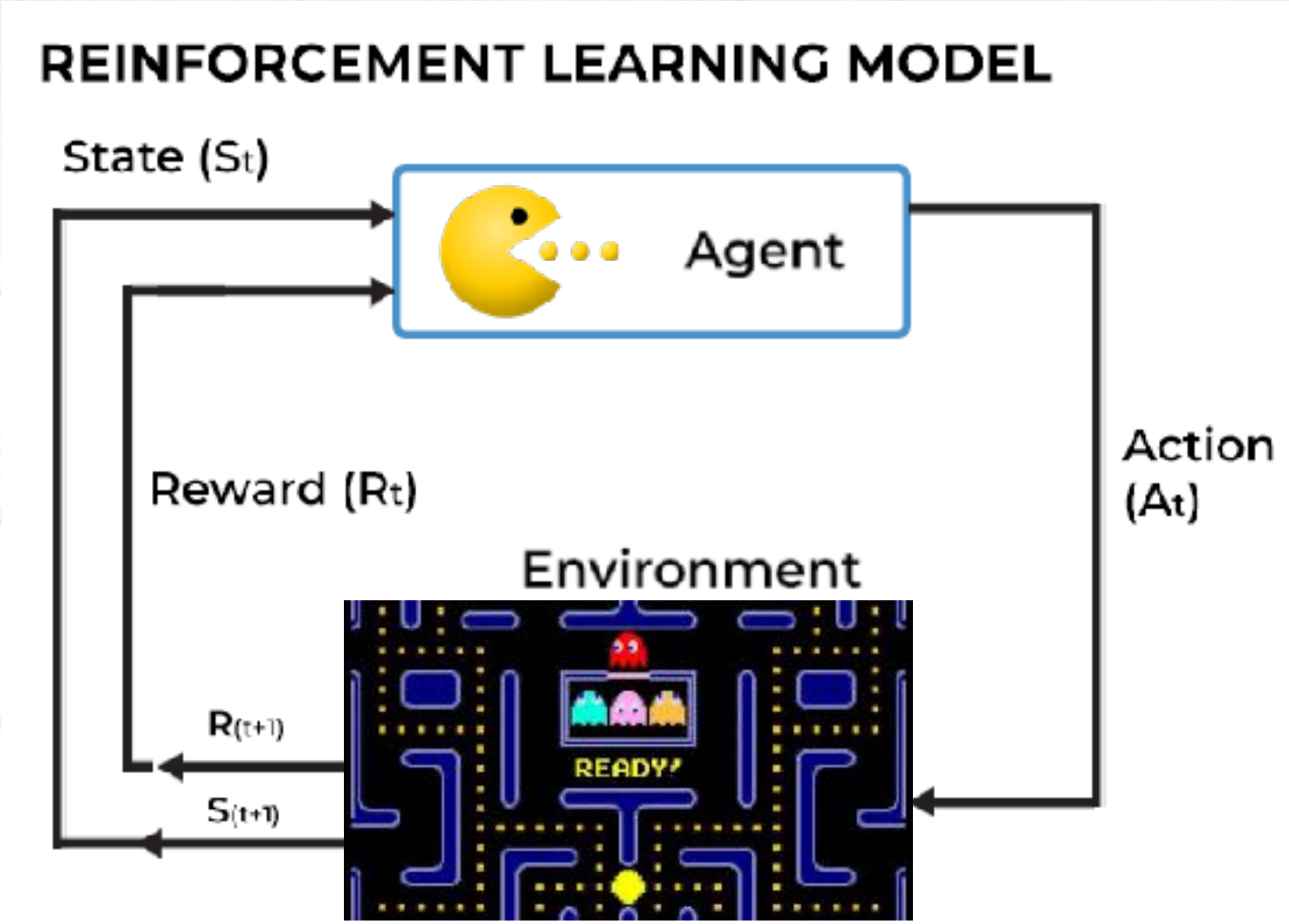
Reinforcement Learning

An Agent (AI model) takes action given state t. These affect the environment. The Agent gets feedback (reward), and the updated state of the environment at t+1.



Reinforcement Learning

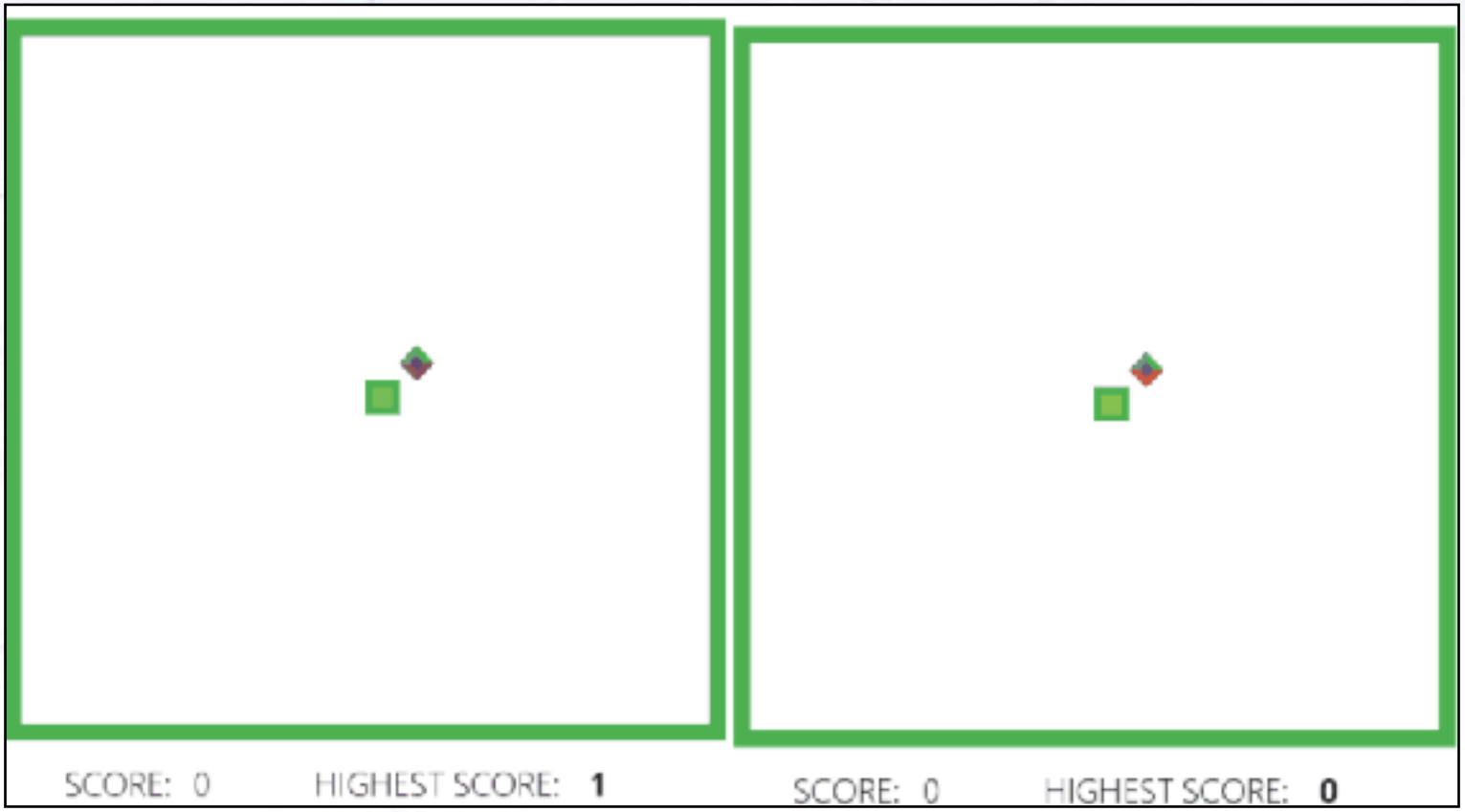
An Agent (AI model) takes action given state t. These affect the environment. The Agent gets feedback (reward), and the updated state of the environment at t+1.



The Snake/Worm Game

A classic “old school” video game is Snake (or Worm) Game, which due to its simplicity has been made in 100s of versions since the first inception in 1976.

Below is the untrained AI (left) and the same AI after training for 150 games.

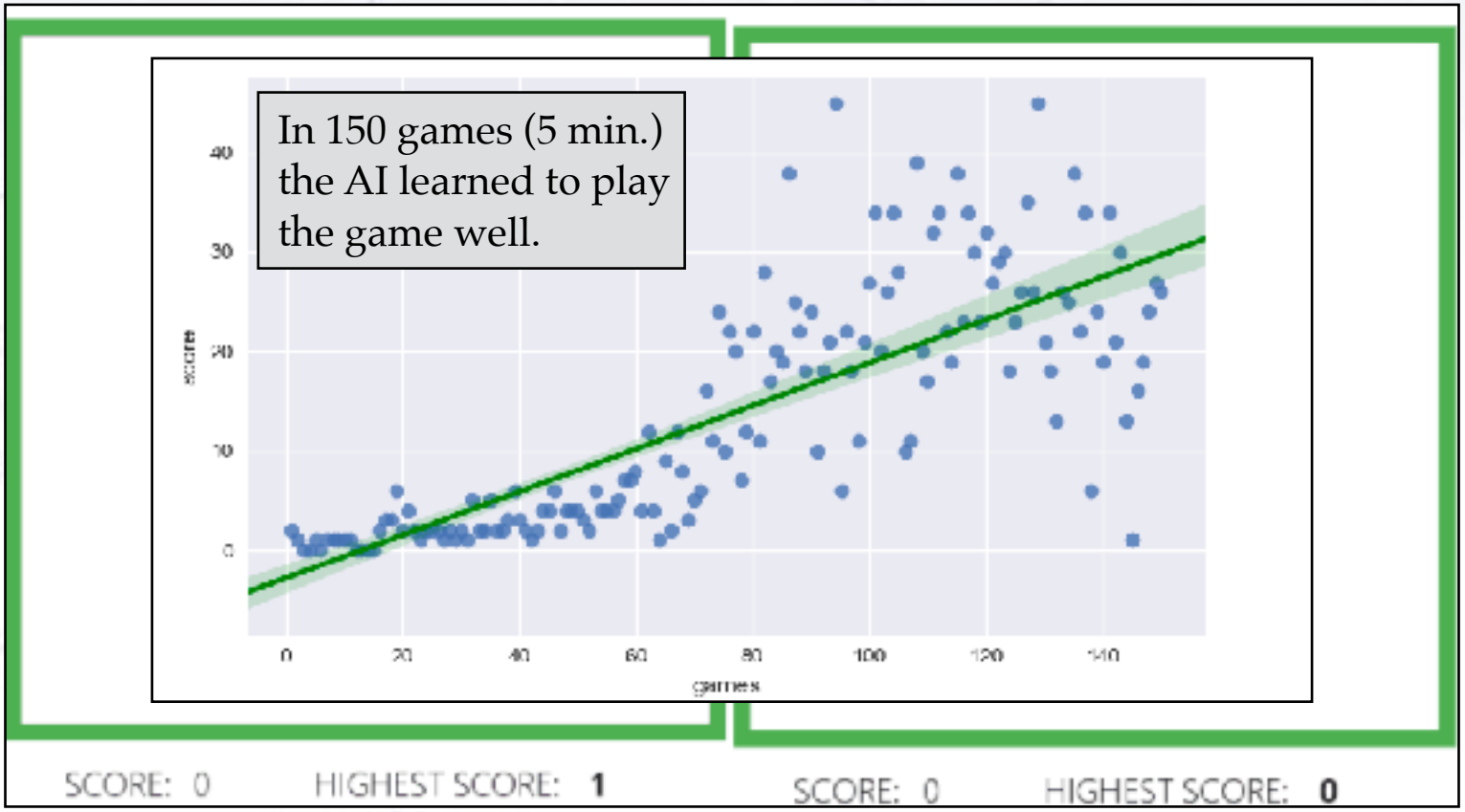


Towards Data Science: How to teach AI to play Games

The Snake/Worm Game

A classic "old school" video game is Snake (or Worm) Game, which due to its simplicity has been made in 100s of versions since the first inception in 1976.

Below is the untrained AI (left) and the same AI after training for 150 games.

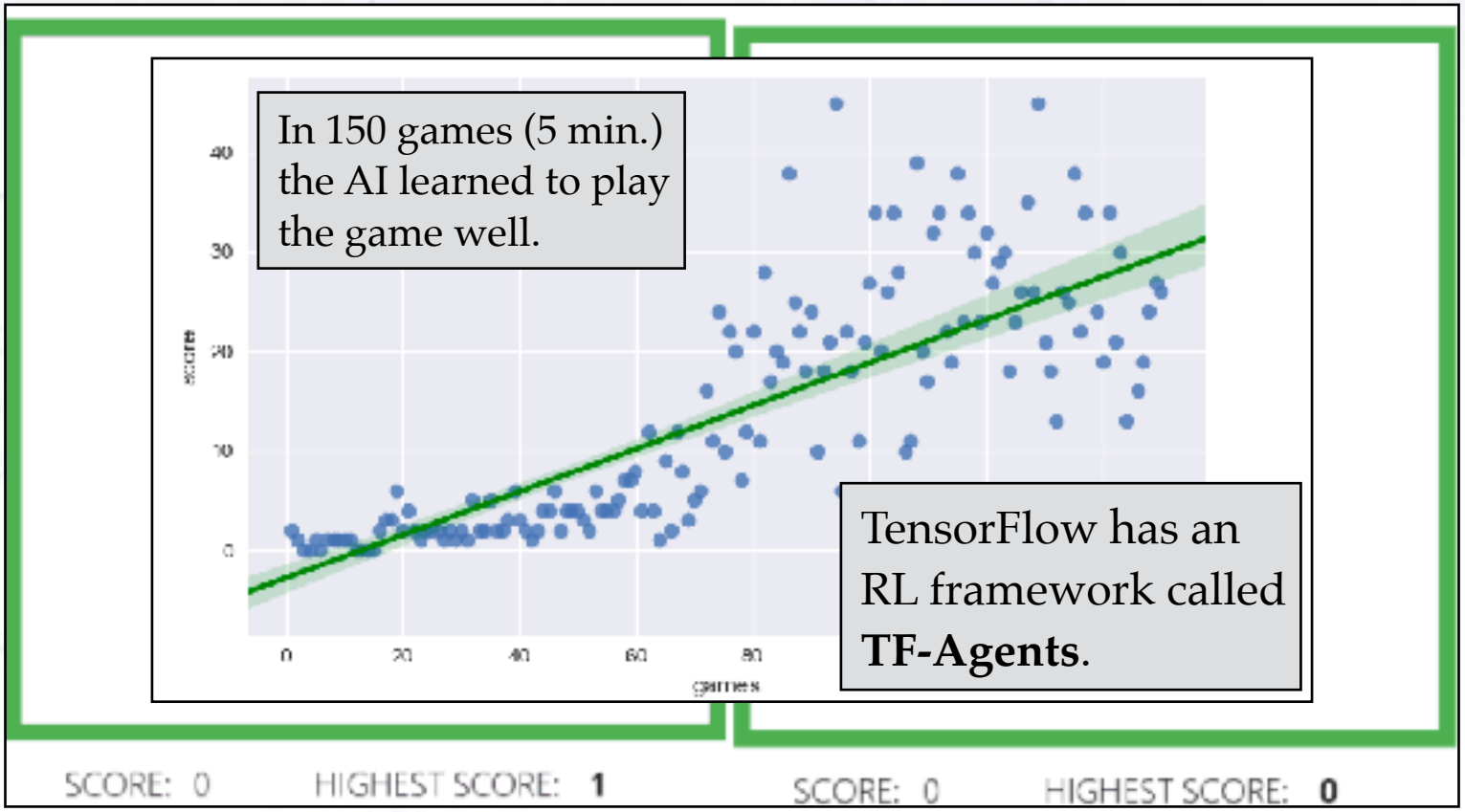


Towards Data Science: How to teach AI to play Games

The Snake/Worm Game

A classic "old school" video game is Snake (or Worm) Game, which due to its simplicity has been made in 100s of versions since the first inception in 1976.

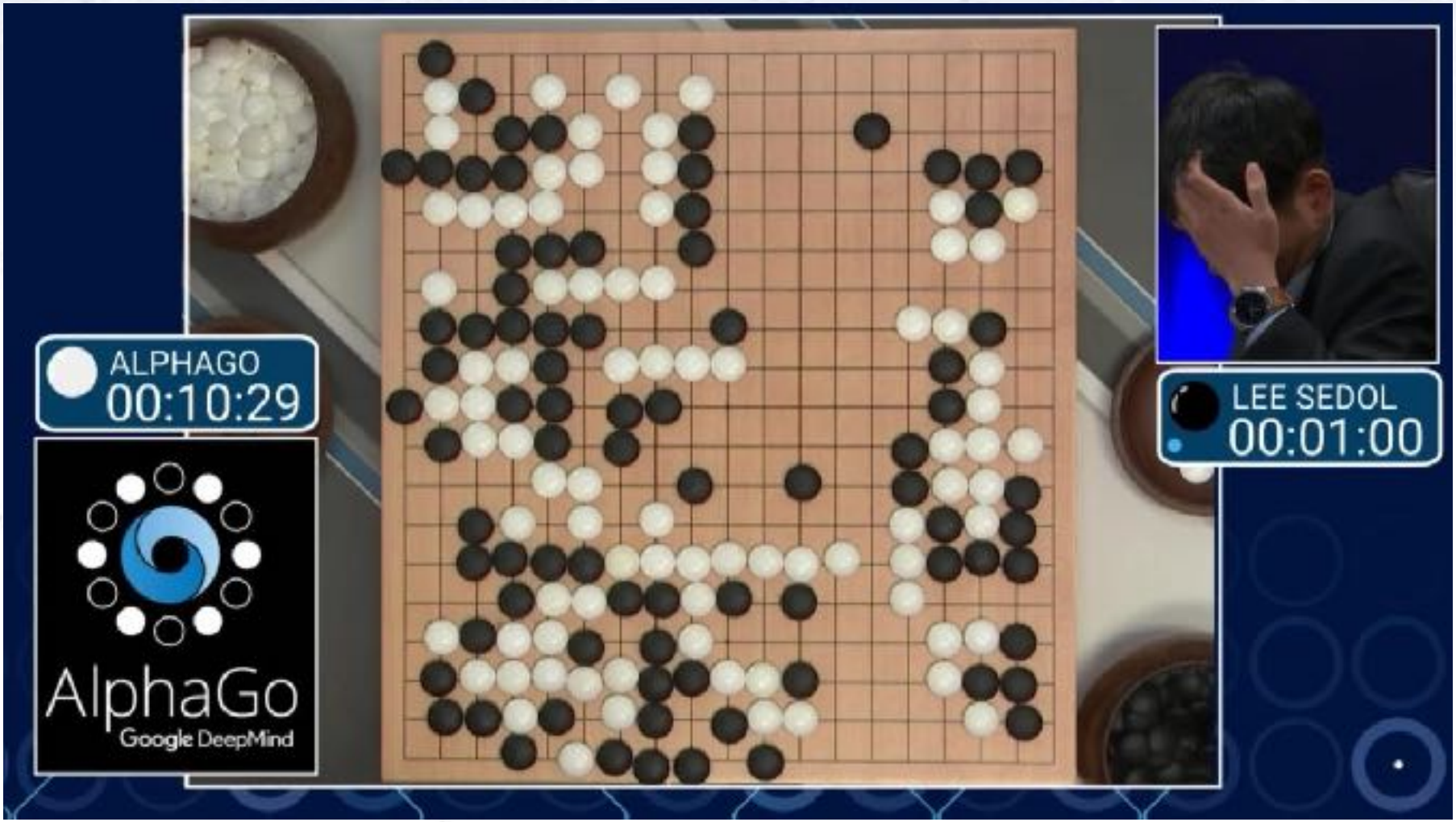
Below is the untrained AI (left) and the same AI after training for 150 games.



Towards Data Science: How to teach AI to play Games

AlphaGo

In October 2015, AlphaGo was introduced to play Go (considered hard for AI).
In March 2016 it won against the reigning world champion, Lee Sedol.



One program to rule them all

In December 2018, AlphaZero was introduced to play three classic strategy board games…

A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play

David Silver[1,2,*,†], Thomas Hubert[1,*], Julian Schrittwieser[1,*], Ioannis Antonoglou[1], Matthew Lai[1], Arthur Guez[1], Marc Lanctot[1], Laurent Sifre[1], Dharshan Kumaran[1], Thore Graepel[1], Timothy Lillicrap[1], Karen Simonyan[1], Demis Hassabis[1,†]

[1]DeepMind, 6 Pancras Square, London N1C 4AG, UK.

[2]University College London, Gower Street, London WC1E 6BT, UK.

↵[†]Corresponding author. Email: davidsilver@google.com (D.S.); dhcontact@google.com (D.H.)

↵[*] These authors contributed equally to this work.

– Hide authors and affiliations

Science 07 Dec 2018:
Vol. 362, Issue 6419, pp. 1140-1144
DOI: 10.1126/science.aar6404

One program to rule them all

In December 2018, AlphaZero was introduced to play three classic strategy board games…

A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play

David Silver[1,2,*,†], Thomas Hubert[1,*], Julian Schrittwieser[1,*], Ioannis Antonoglou[1], Matthew Lai[1], Arthur Guez[1], Marc Lanctot[1], Laurent Sifre[1], Dharshan Kumaran[1], Thore Graepel[1], Timothy Lillicrap[1], Karen Simonyan[1], Demis Hassabis[1,†]

[1]DeepMind, 6 Pancras Square, London N1C 4AG, UK.

[2]University College London, Gower Street, London WC1E 6BT, UK.

↵[†]Corresponding author. Email: davidsilver@google.com (D.S.); dhcontact@google.com (D.H.)

↵[*] These authors contributed equally to this work.

– Hide authors and affiliations

Science 07 Dec 2018:
Vol. 362, Issue 6419, pp. 1140-1144
DOI: 10.1126/science.aar6404

After four hours of training it beat the best chess program in the world at the time: 72 draws, 28 wins, and… 0 losses.

Within 24 hours AlphaZero achieved a superhuman level of play in ALL three games by defeating world-champion programs…. using only the rules!