Open-ended learning in symmetric zero-sum games

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech M. Czarnecki, Julien Perolat, Max Jaderberg, Thore Graepel

DeepMind

Long ago and far away (mid-1800s in Cambridge, England):

First tutor: **"I'm teaching the most brilliant boy in Britain"**
Second tutor: **"Well, I'm teaching the best test-taker"**

Depending on the version of the story, the first boy was either **Lord Kelvin** or **James Clerk Maxwell**. The second boy indeed *scored highest* on the Mathematical Tripos, but is otherwise long forgotten.

DeepMind

Long ago and far away (mid-1800s in Cambridge, England):

First tutor: **"I'm teaching the most brilliant boy in Britain"**
Second tutor: **"Well, I'm teaching the best test-taker"**

Depending on the version of the story, the first boy was either **Lord Kelvin** or **James Clerk Maxwell**. The second boy indeed *scored highest* on the Mathematical Tripos, but is otherwise long forgotten.

Modern learning algorithms are outstanding test-takers

But intelligence is about more than taking tests
It's also about formulating useful problems

Where do problems come from?

Answer #1:

Someone packages a dataset into a loss function

e.g. ImageNet, CIFAR, MNIST, ...

Where do problems come from?

Answer #1:

Someone packages a dataset into a loss function

e.g. ImageNet, CIFAR, MNIST, …

Answer #2:

Someone builds a task (that is, an environment sprinkled with rewards)

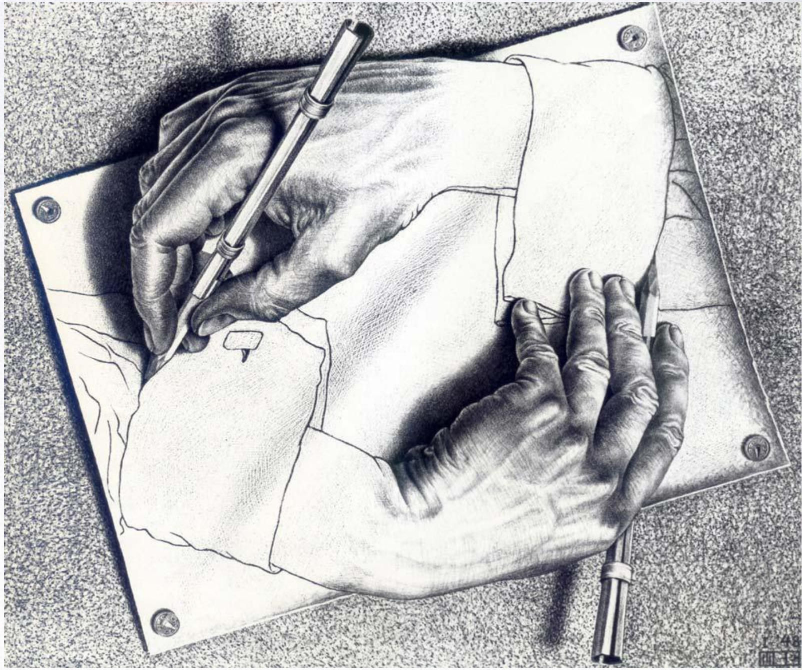
e.g. Arcade Learning Environment, DM-Lab, Open AI gym, …

Where do problems come from?

Answer #3:

Self-play in symmetric zero-sum games

The agent *is* the task -- create an outer loop that bends deep RL on itself



DeepMind

(Naive) self-play *is* an open-ended learning algorithm

It's pretty amazing

Algorithm 2 Self-play

input: agent $\mathbf{v}_1$
for $t = 1, \ldots, T$ **do**
  $\mathbf{v}_{t+1} \leftarrow \text{oracle}\left(\mathbf{v}_t, \phi_{\mathbf{v}_t}(\bullet)\right)$
end for
output: $\mathbf{v}_{T+1}$

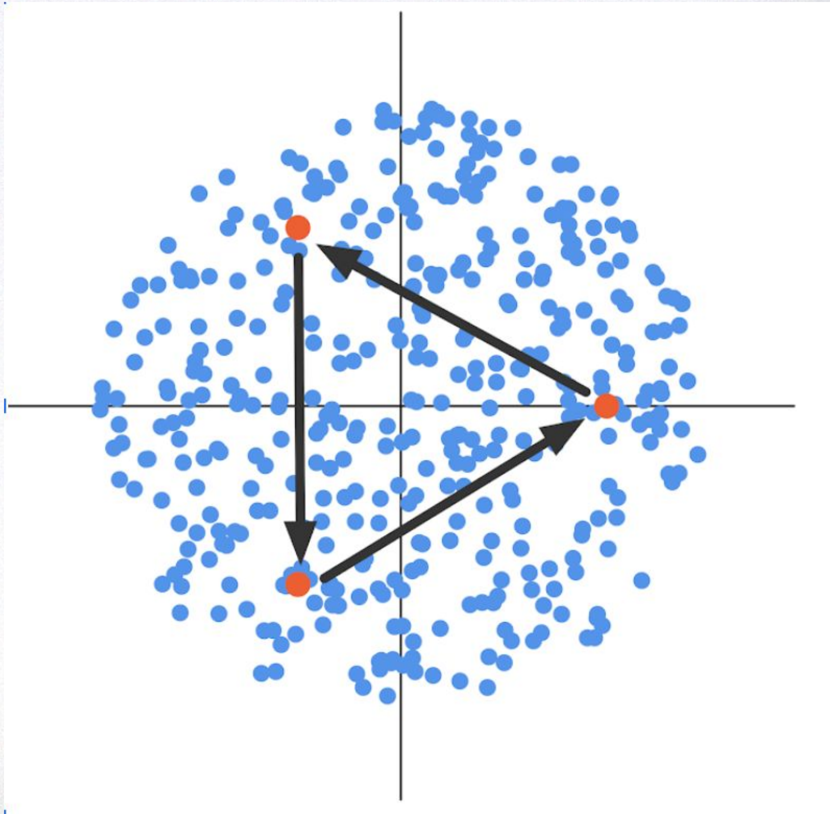
(Naive) self-play *is* an open-ended learning algorithm

but …

there are really simple examples

where it completely breaks down

It's **not** a general purpose learning algorithm, *not even* for zero-sum games



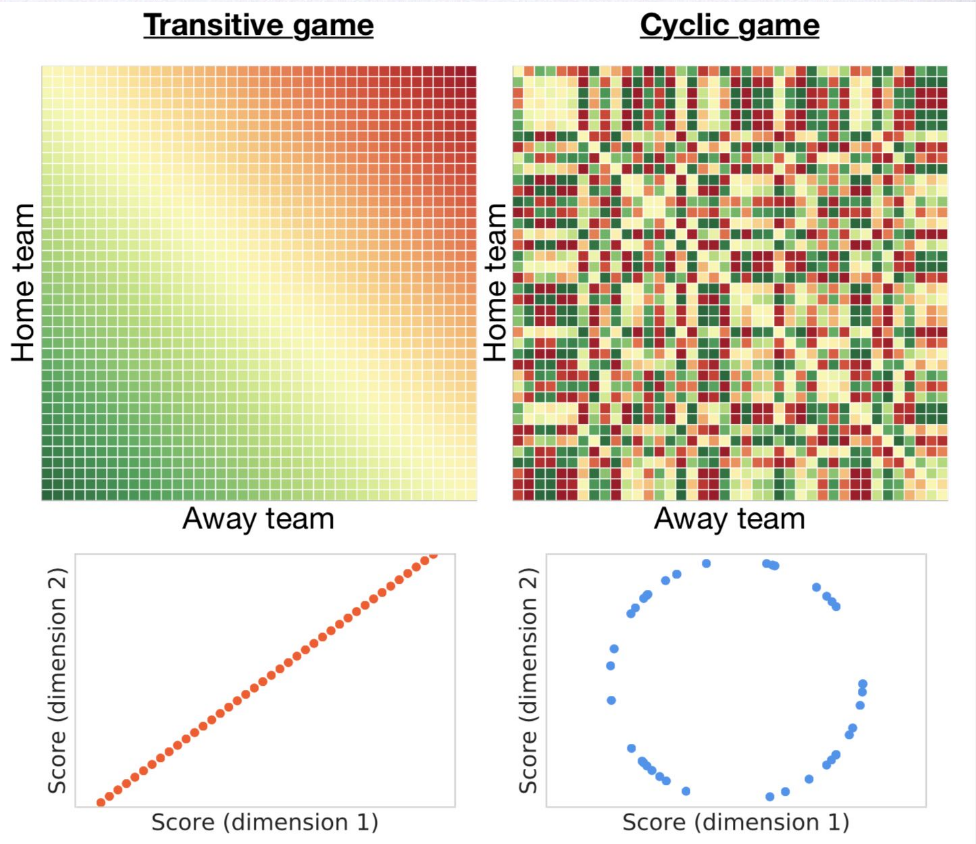
On the varieties of zero-sum games

transitive: “relative skill determines who wins”

cyclic: “every strategy has a counter-strategy”

DeepMind

Theorem: Any symmetric two-player zero-sum game decomposes into **[transitive]** + **[cyclic]** components



transitive: skill determines outcome

cyclic: every strategy has a counter-strategy

DeepMind

The paper:

How to formulate useful objectives in non-transitive games

New tools:

- **Gamescapes** (generalize landscapes, but represent many objectives)
- **Population-level performance** measures
- **Population-level training** algorithms