

# Deep Reinforcement Learning

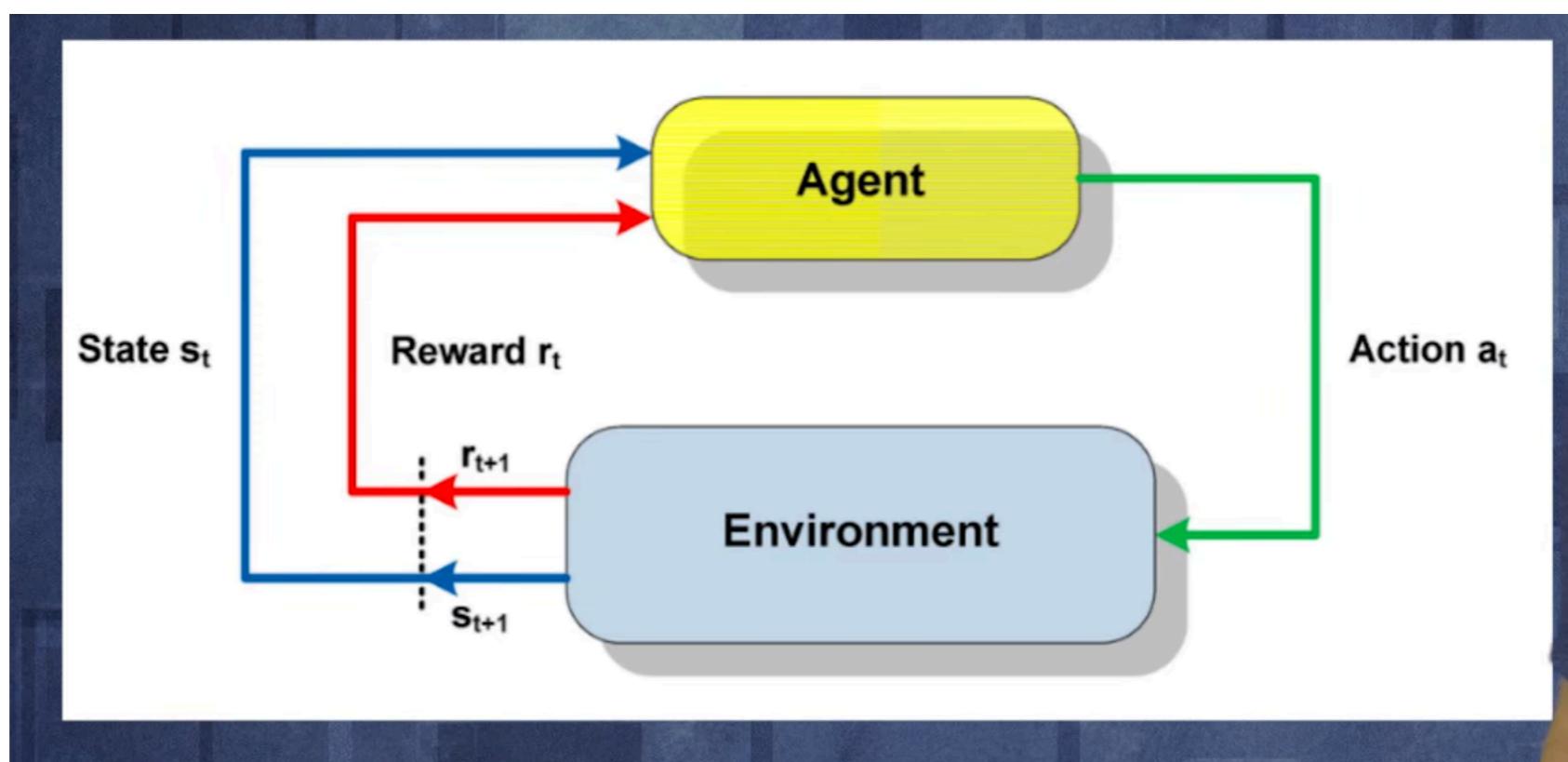
romeo kienzler

# Outline

- 8:30-12:00: 2 Theory Modules + 2 Exercises
  - Reinforcement Learning Overview
  - Introduction to AlphaGo Zero
  - Exercises on Policy Gradient Methods using the OpenAI Gym
- 13:00-16:00: 1 short Theory Module + microHackathon
  - Understanding influence on human priors
  - microHackahon: Implement a full Policy Gradient DeepRL agent for OpenAI Gym

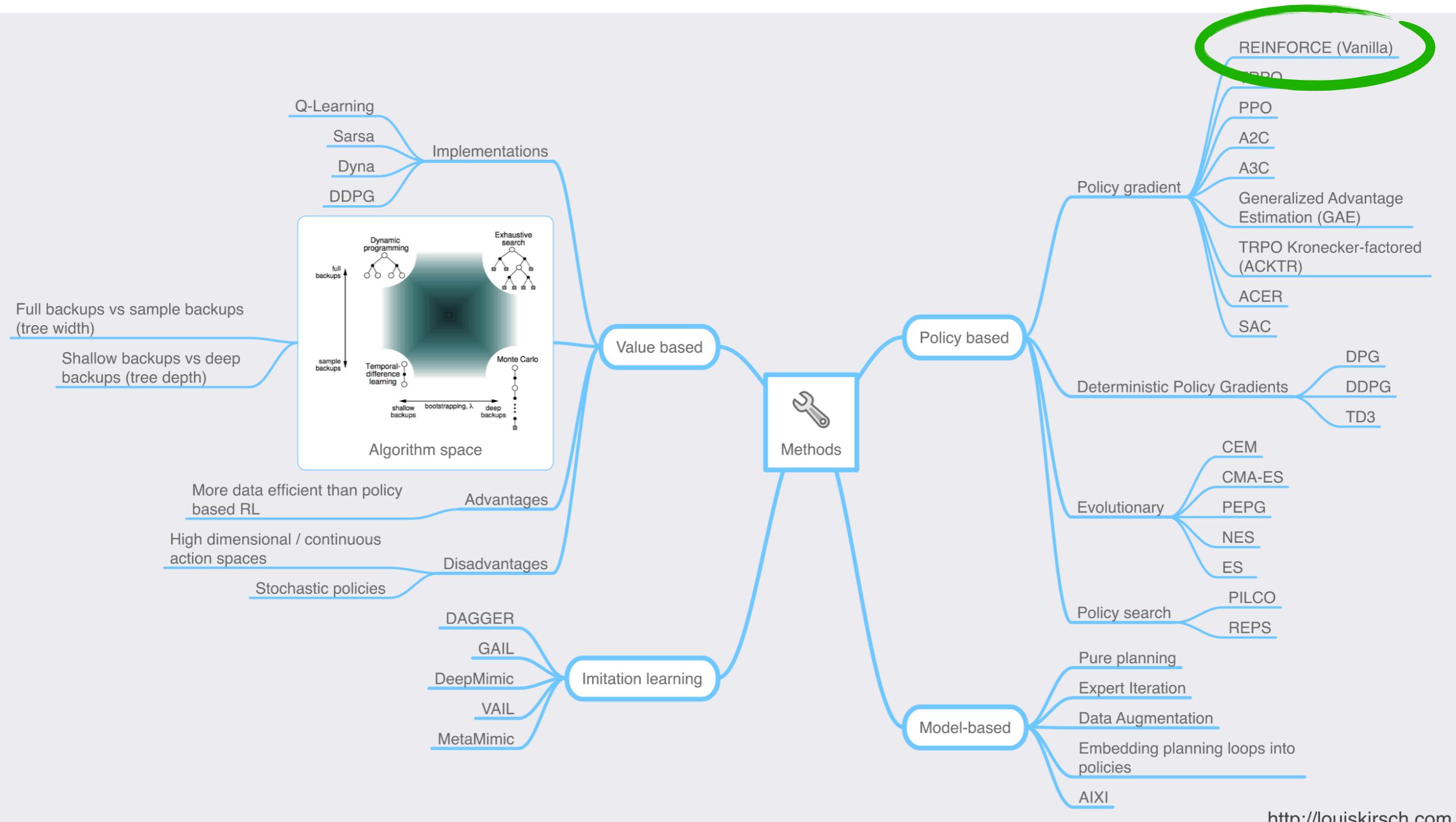
Reinforcement learning involves an agent, a set of *states*, and a set of *actions* per state. By performing an action  $a \in A$ , the agent transitions from state to state. Executing an action in a specific state provides the agent with a *reward* (a numerical score).

The agent is trained to choose the optimal action  $a$  in a given state  $s$  in order to maximize reward  $r$ .

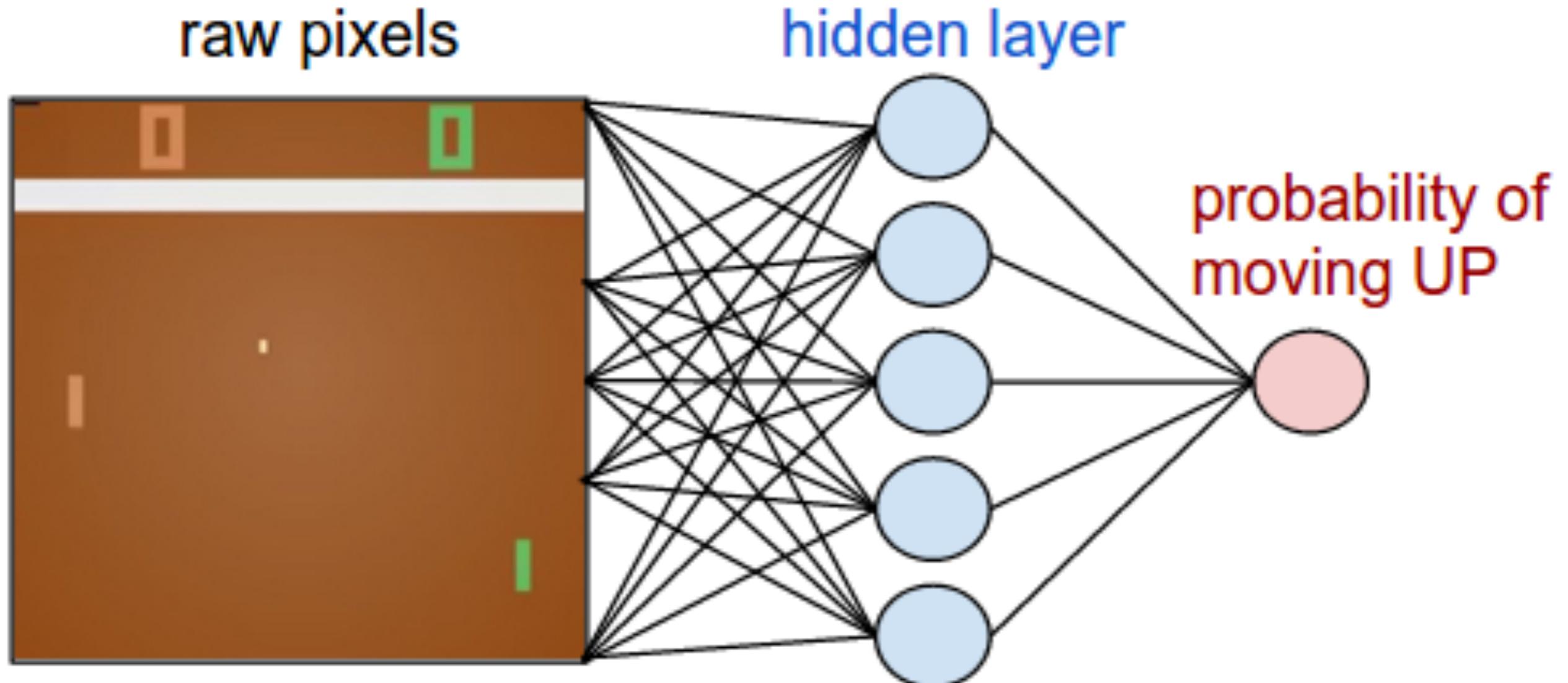


# latest achievements

- March 2016: AlphaGo Lee beats 18-time world champion Lee Sedol in the game of GO October 2017: AlphaGo Zero beats AlphaGo Lee 100:0
- July 2018: Dactyl, trained entirely in simulation but applied to real-world physics
- 2018: Show and Tell, a agent capable of following social laws
- 2018: IBM Reinforced Reader-Ranker for Open-Domain Question Answering ( $R^3$ )
- December 2018: AlphaFold won the 13th Critical Assessment of Techniques for Protein Structure Prediction (CASP)
- April 2019: OpenAI Five wins back-to-back games versus Dota 2 world champions OG at Finals, becoming the first AI to beat the world champions in an esports game (PPO)
- 2019: DQ Scheduler, a Deep Reinforcement Learning Based Controller Synchronization in Distributed SDN
- 2019: Dialog-Based Interactive Image Retrieval
- October 2019: Solving Rubik's Cube with a Robot Hand

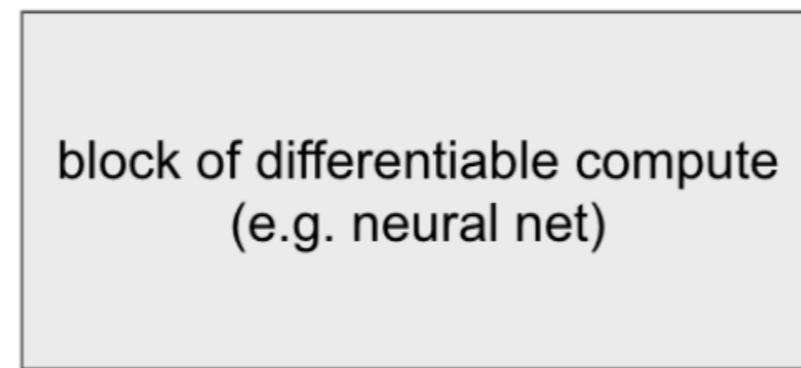


# pong from pixels



Source: <http://karpathy.github.io/2016/05/31/rl/>

### forward pass



log probabilities

-1.2	-0.36
------	-------

gradients

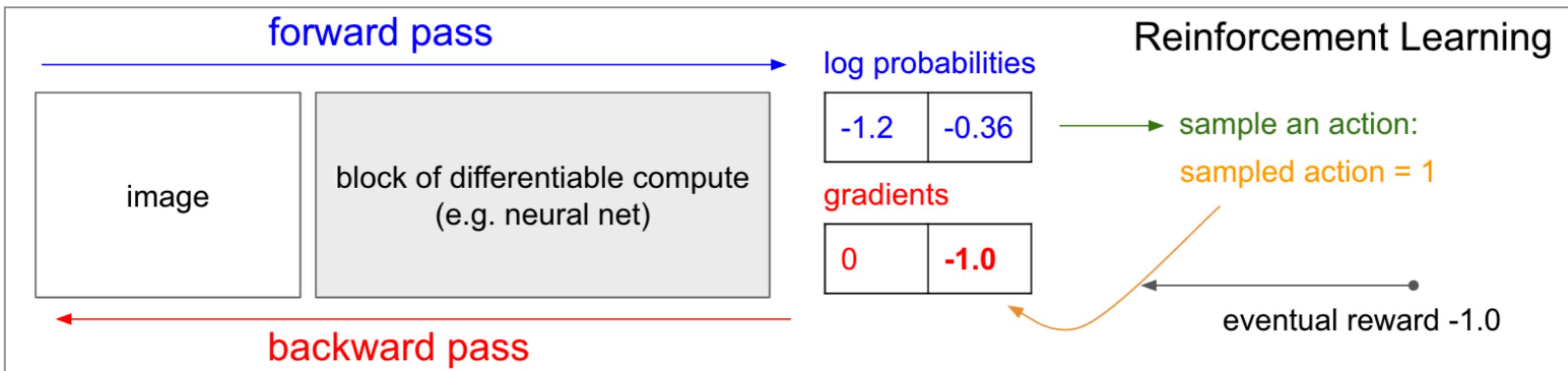
1.0	0
-----	---

Supervised Learning  
(correct label is provided)

correct action  
label = 0

### backward pass

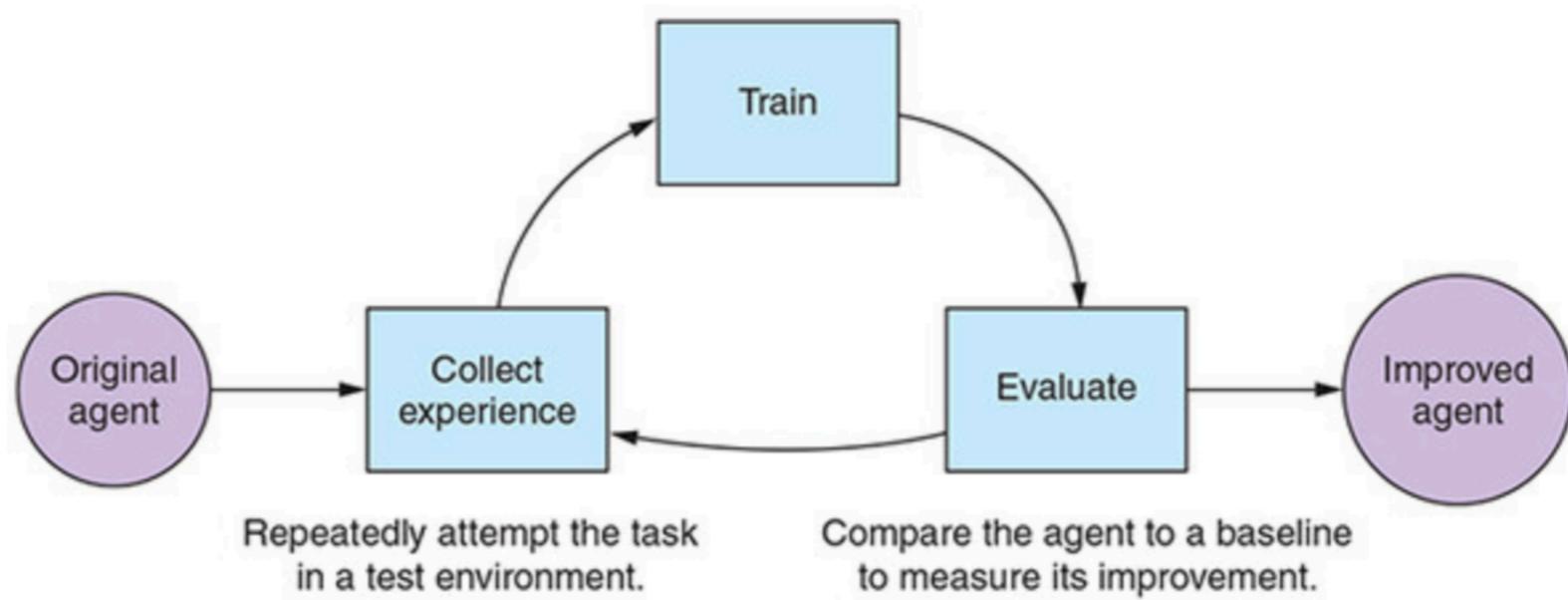
Source: <http://karpathy.github.io/2016/05/31/r/>



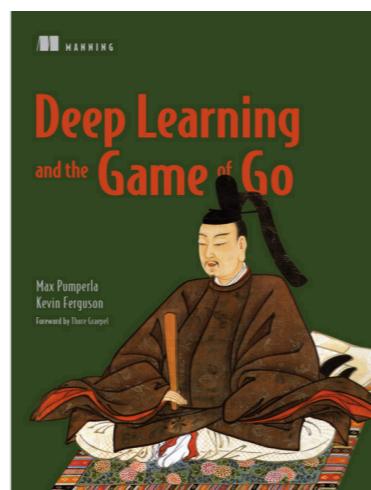
Source: <http://karpathy.github.io/2016/05/31/rl/>

### General reinforcement-learning cycle

Update the agent's behavior in response to the experience results.



**source:**



# Exercise

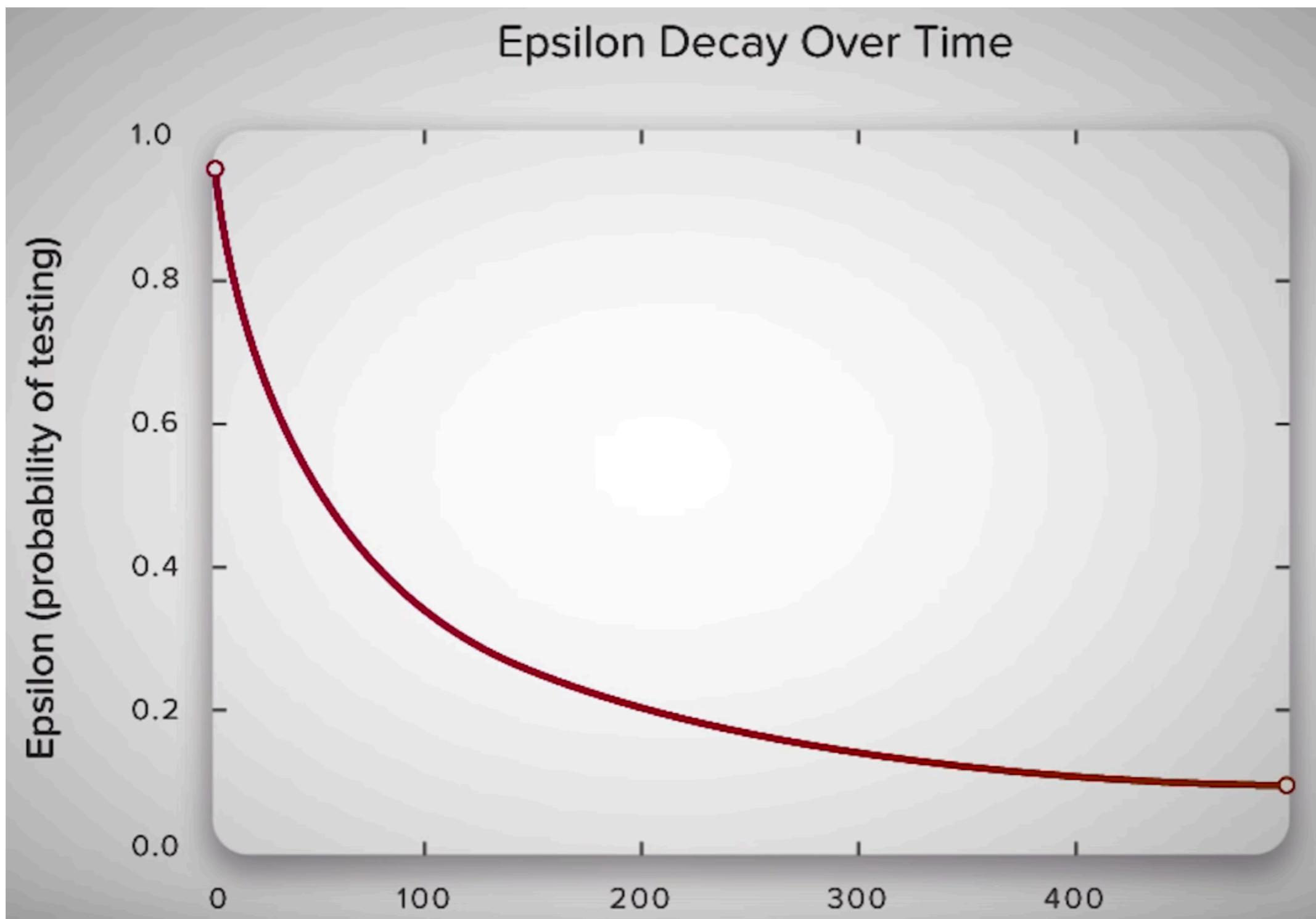
**<https://github.com/romeokienzler/DeepRL> => DeepRL.ipynb**

# epsilon greedy exploration

**Given policy vector  $\pi$ , with probability  $1 - \epsilon$   
choose best action, with probability  $\epsilon$   
choose any other random action**

$$\pi = (0.2, 0.3, 0.1, 0.2, 0.2)$$

## Epsilon Decay Over Time



# **credit assignment problem**

**Problem: Sparse Extrinsic Rewards**

# **credit assignment problem**

**Solution 1: Dense Intrinsic Reward Shaping**

# **credit assignment problem**

**Solution 2: auxiliary reward signals**

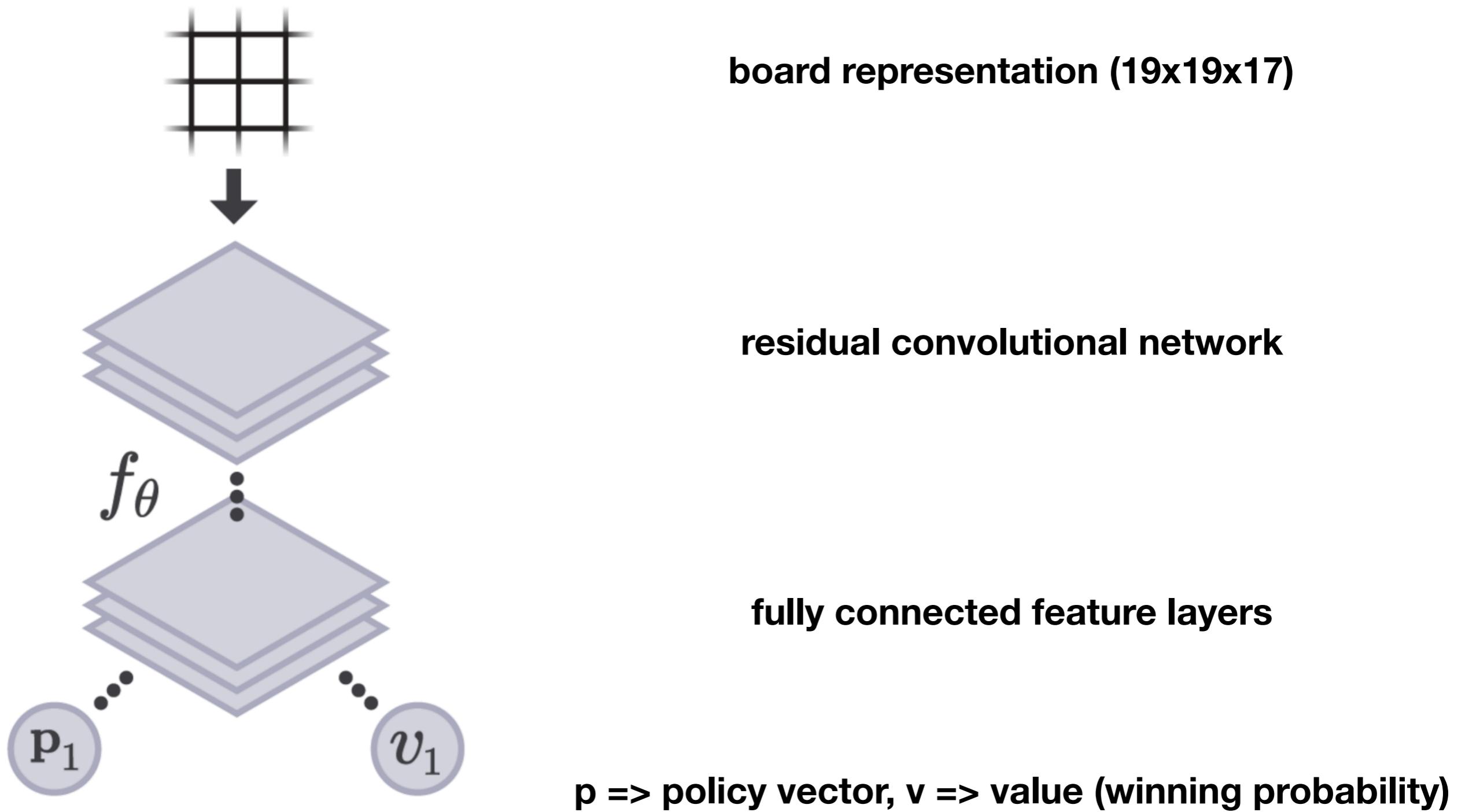
# **credit assignment problem**

**Solution 3: intrinsic curiosity**

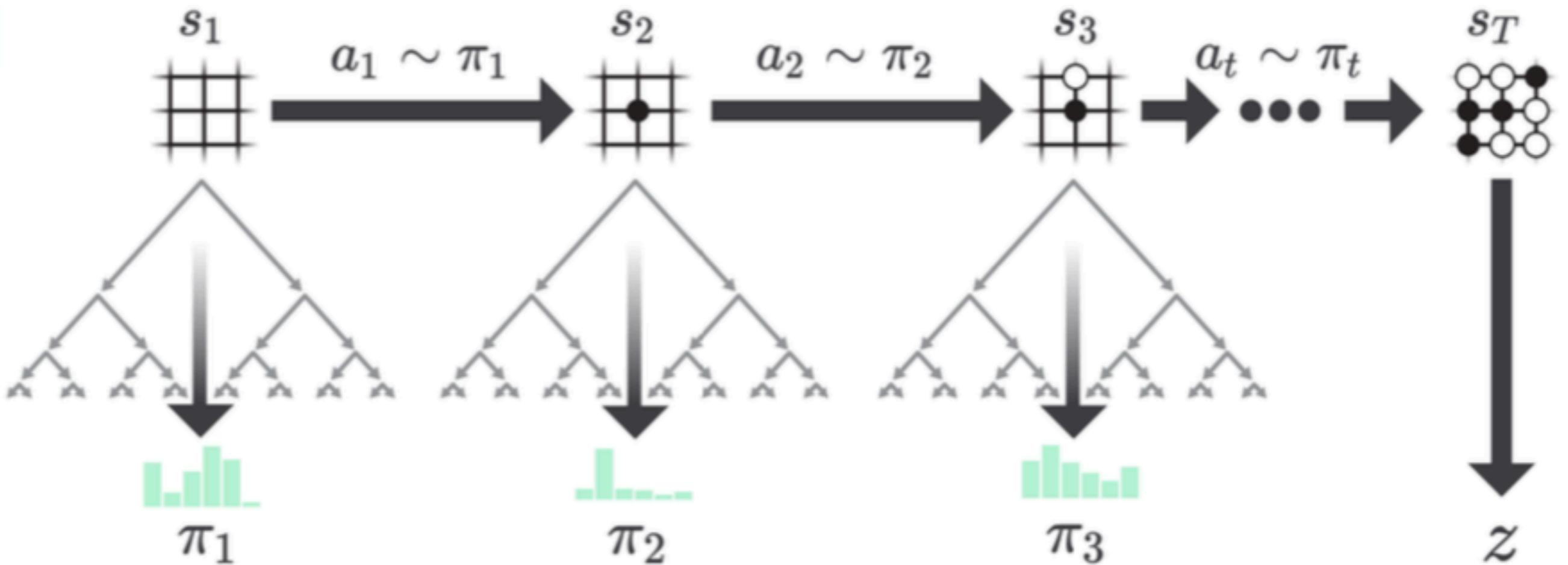
# **Understanding AlphaGo Zero**

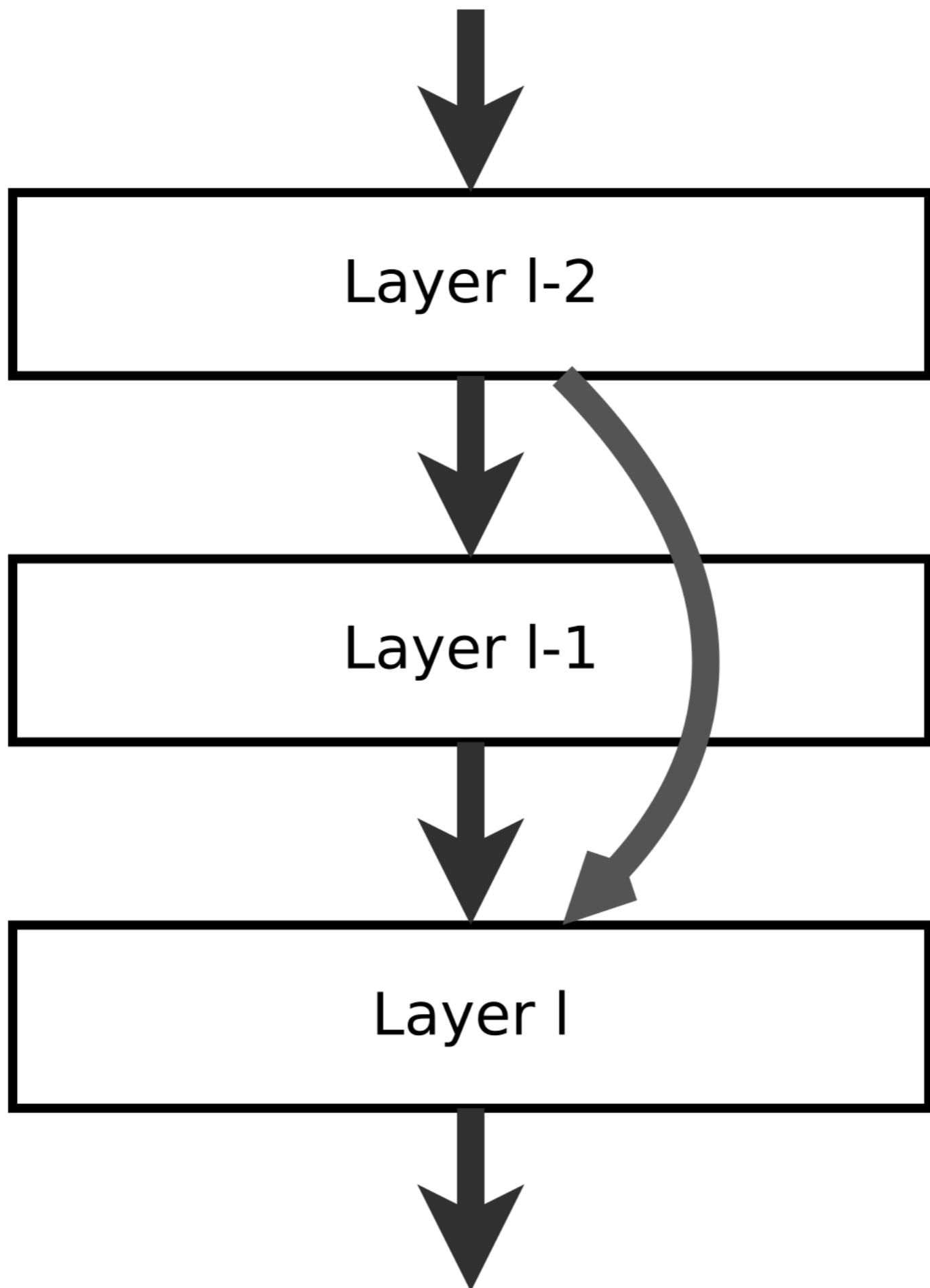
- no hand crafted features
- no hand crafted reward
- no human training data
- alpha go zero learns entirely from scratch (self play only)

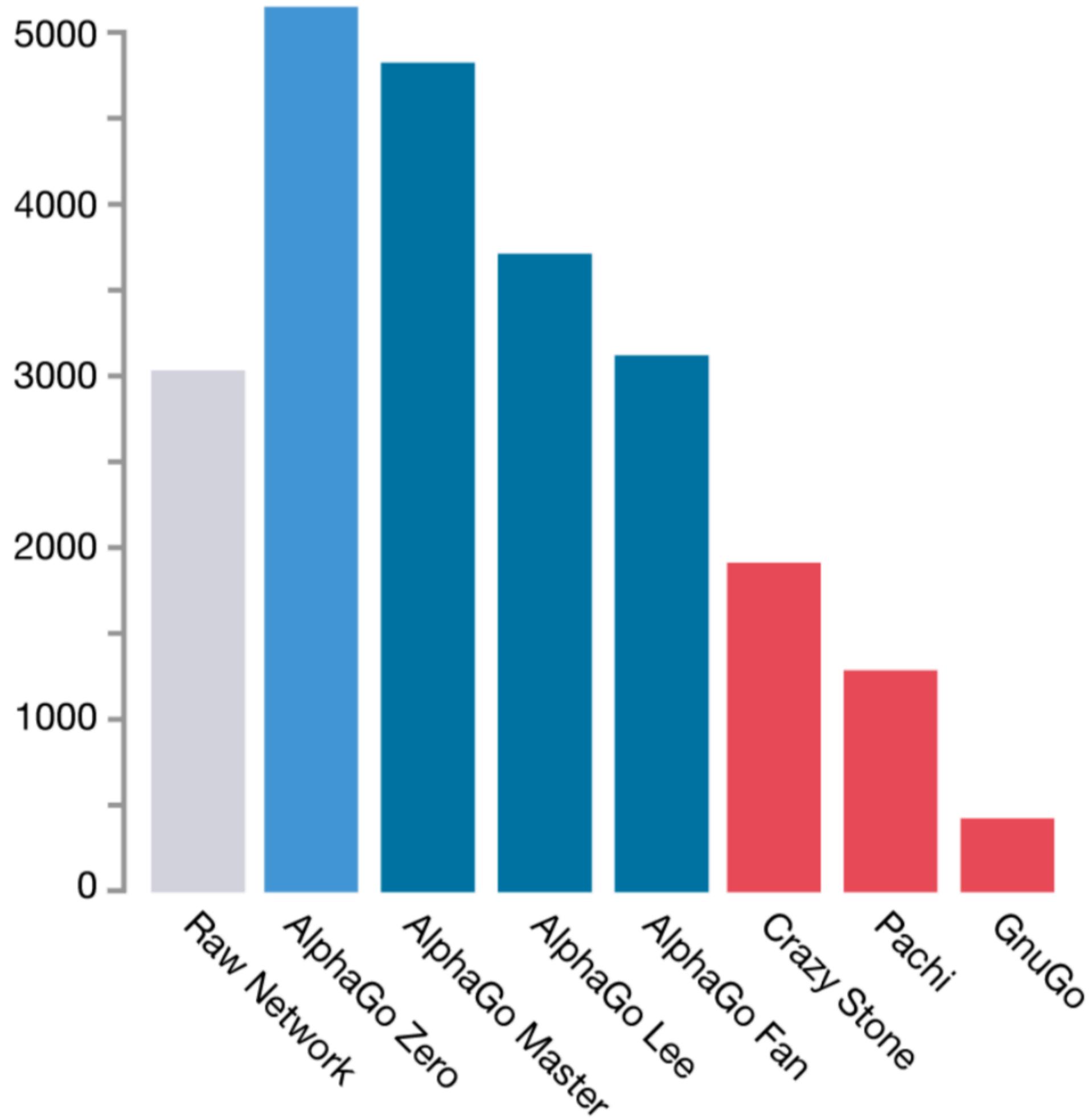
# AlphaGo Zero Architecture

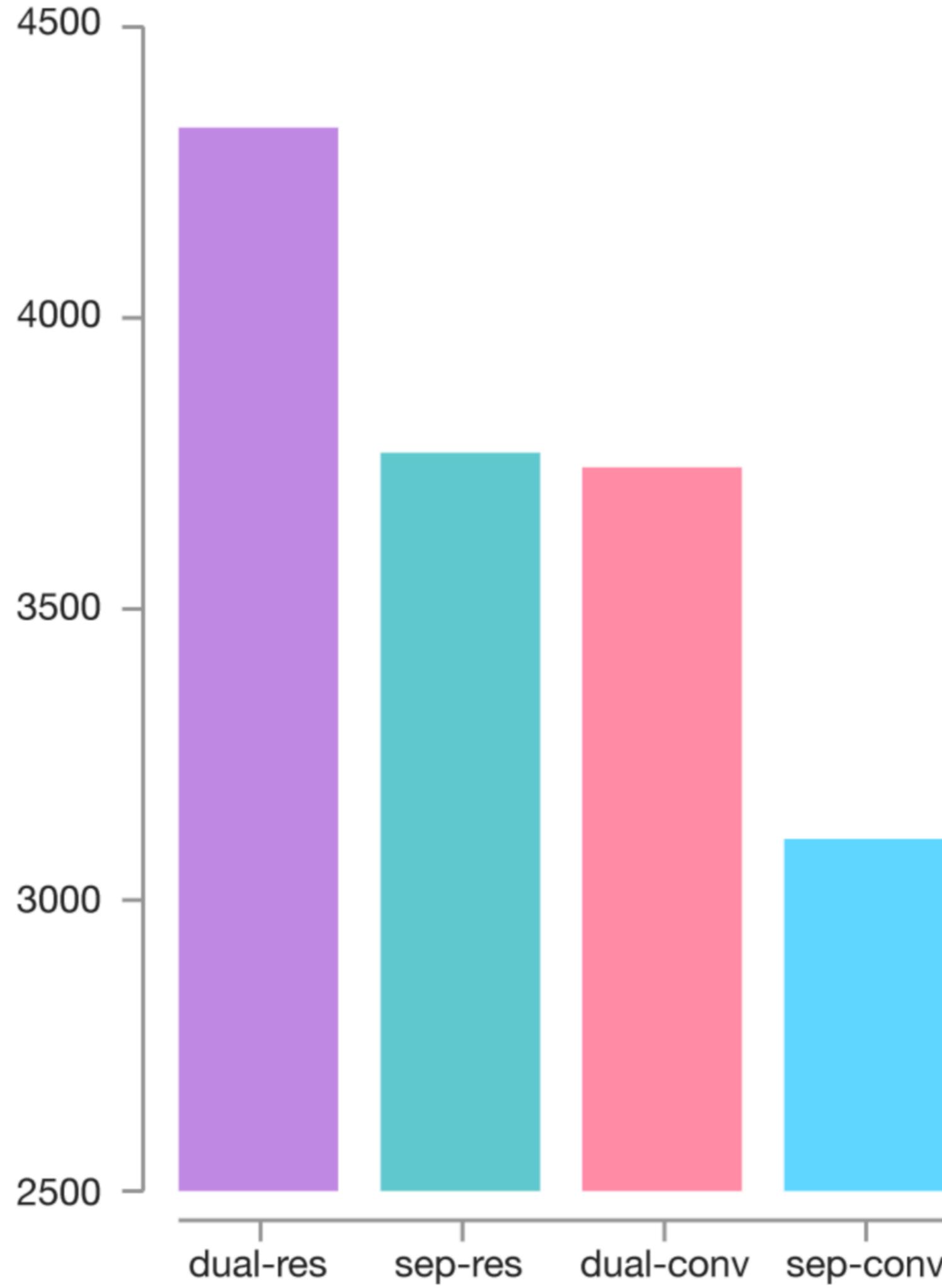


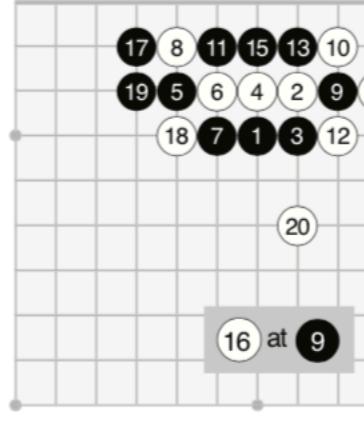
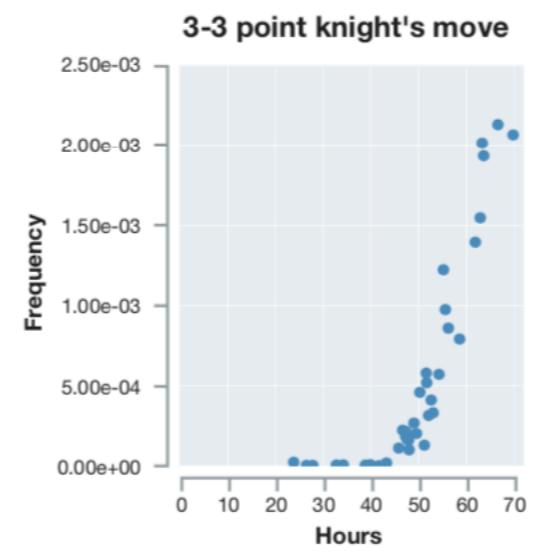
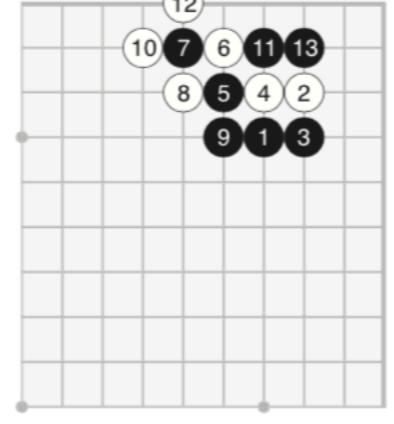
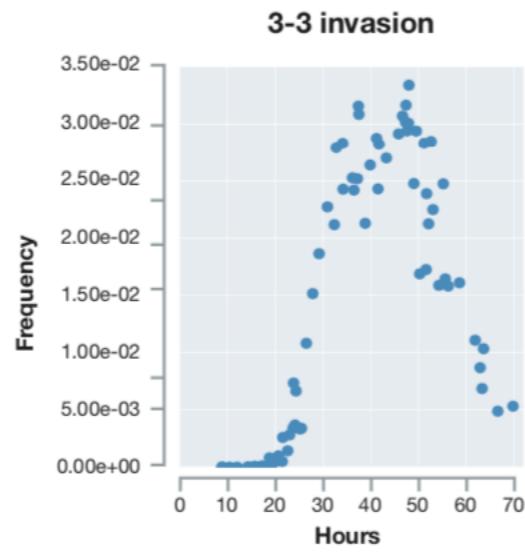
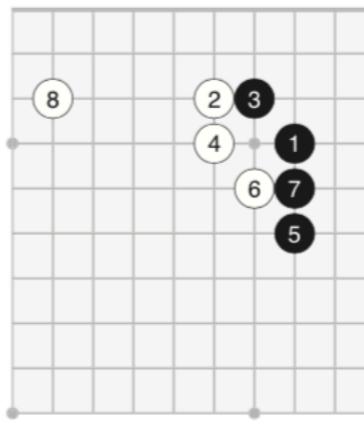
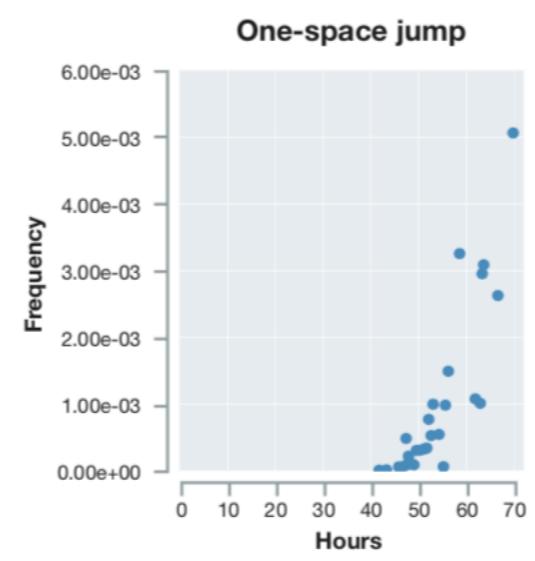
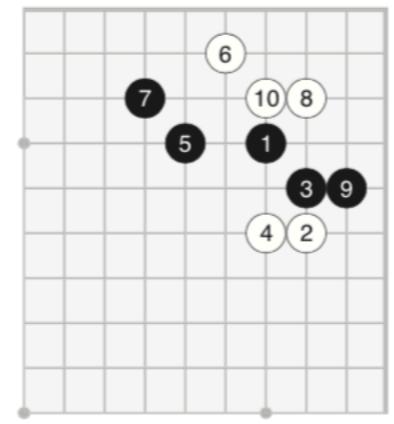
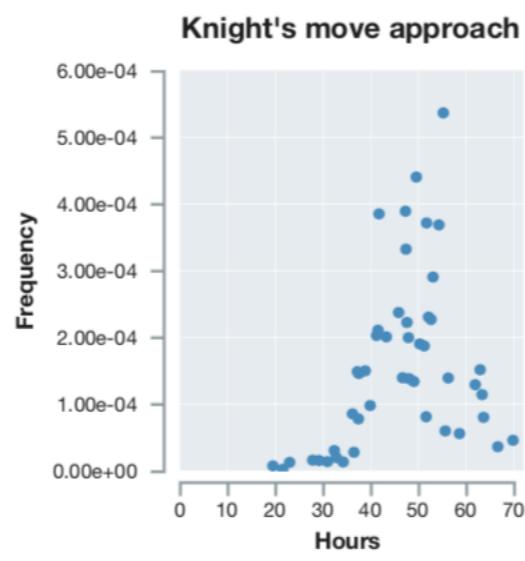
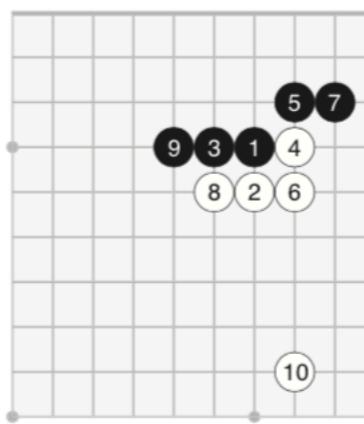
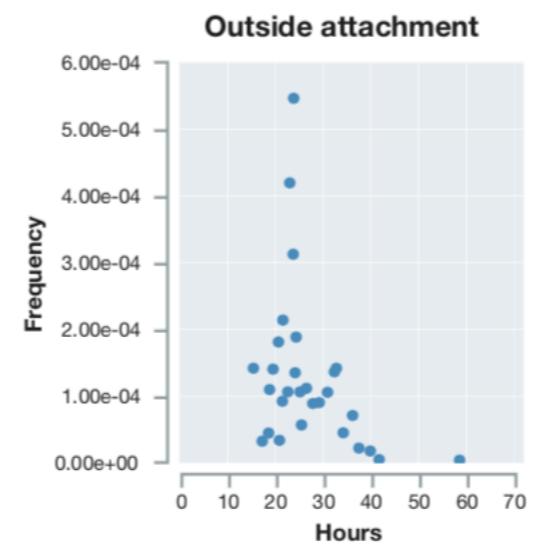
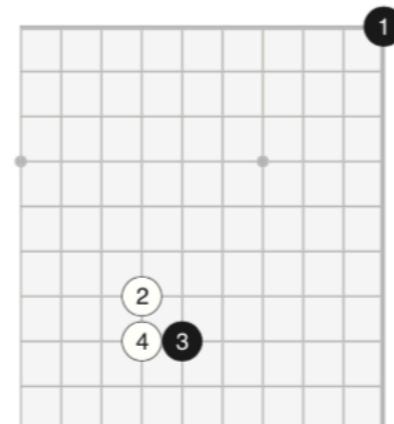
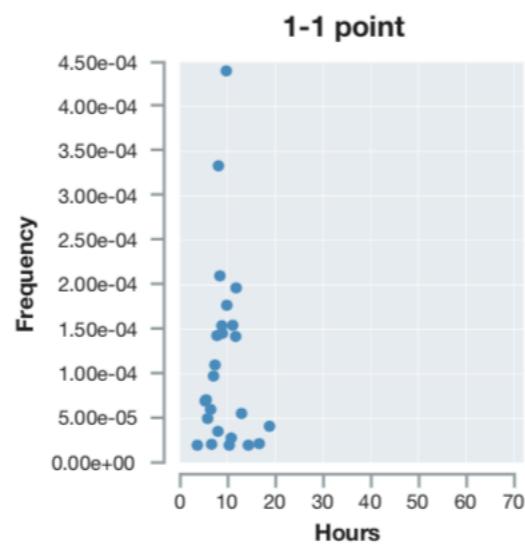
# AlphaGo Zero training

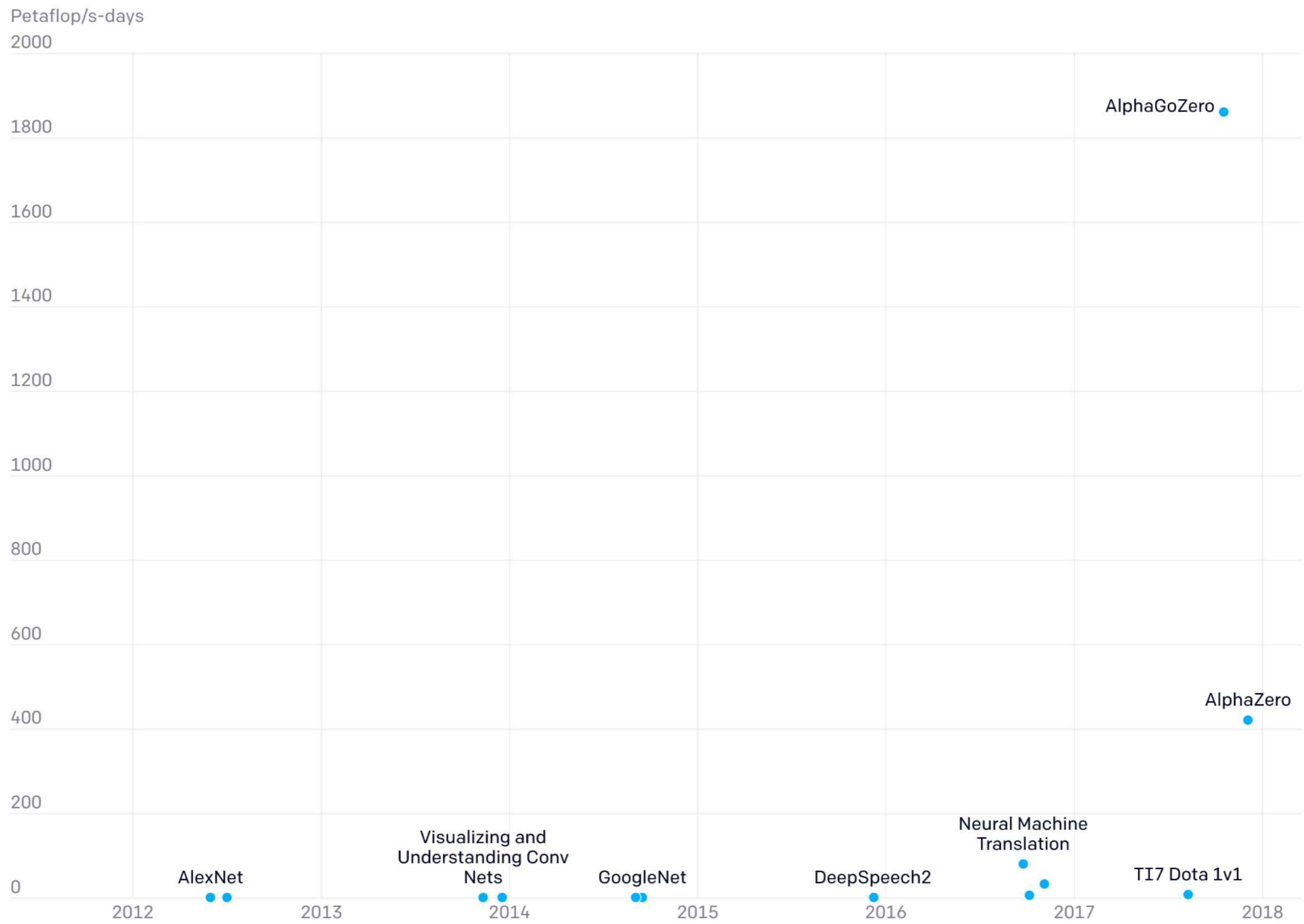










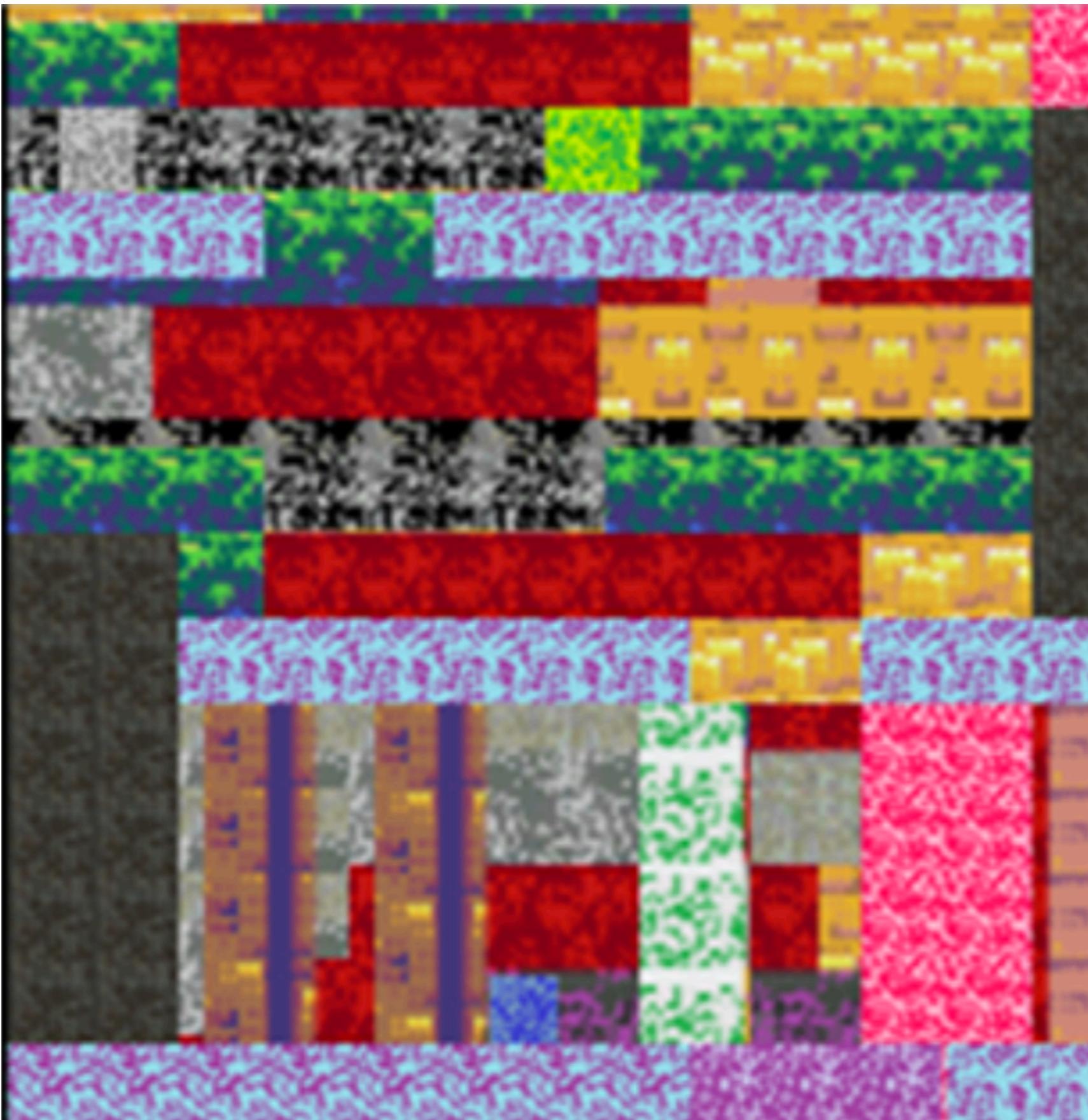


**Source: <https://openai.com/blog/ai-and-compute/>**

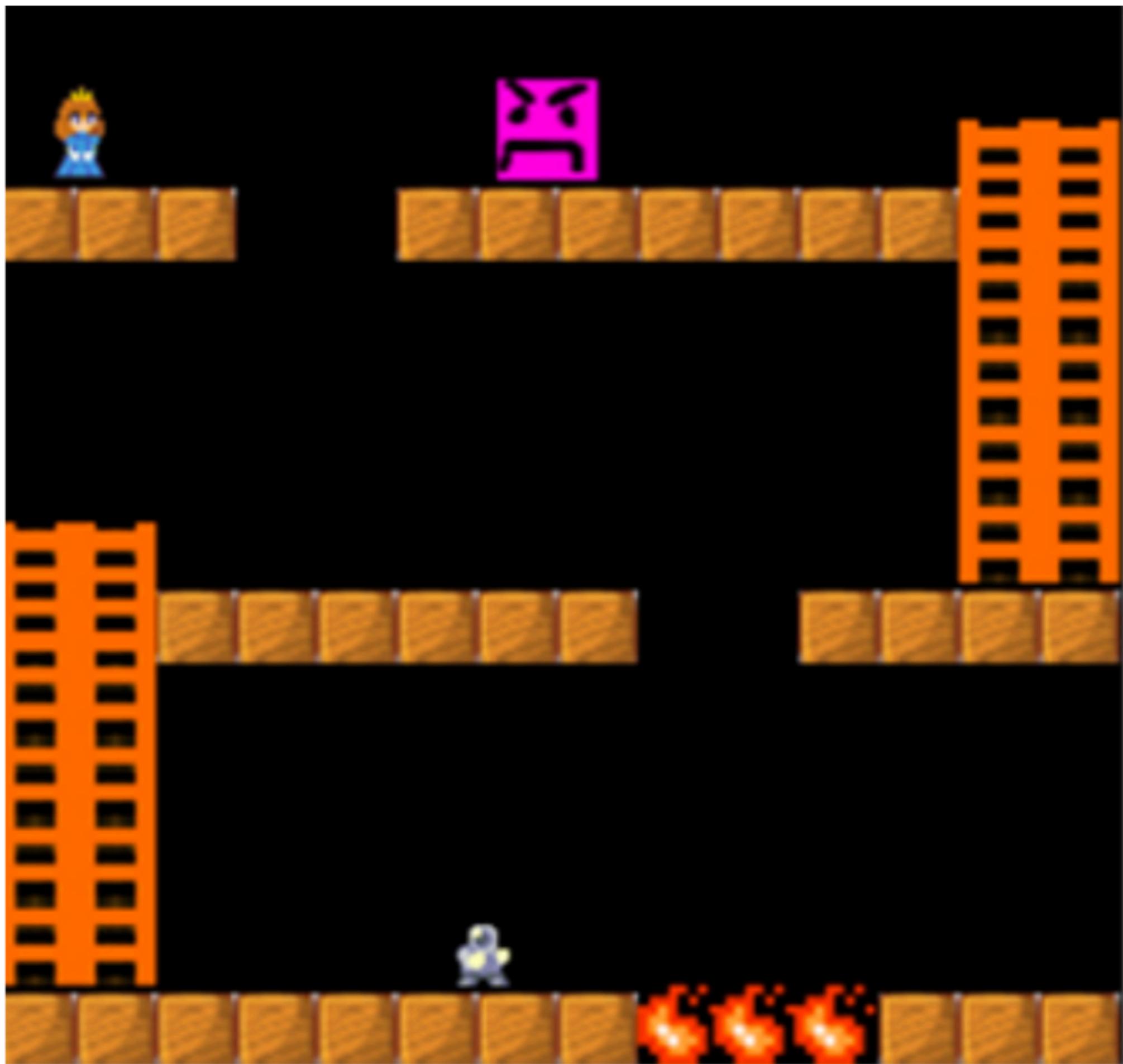
# Exercise

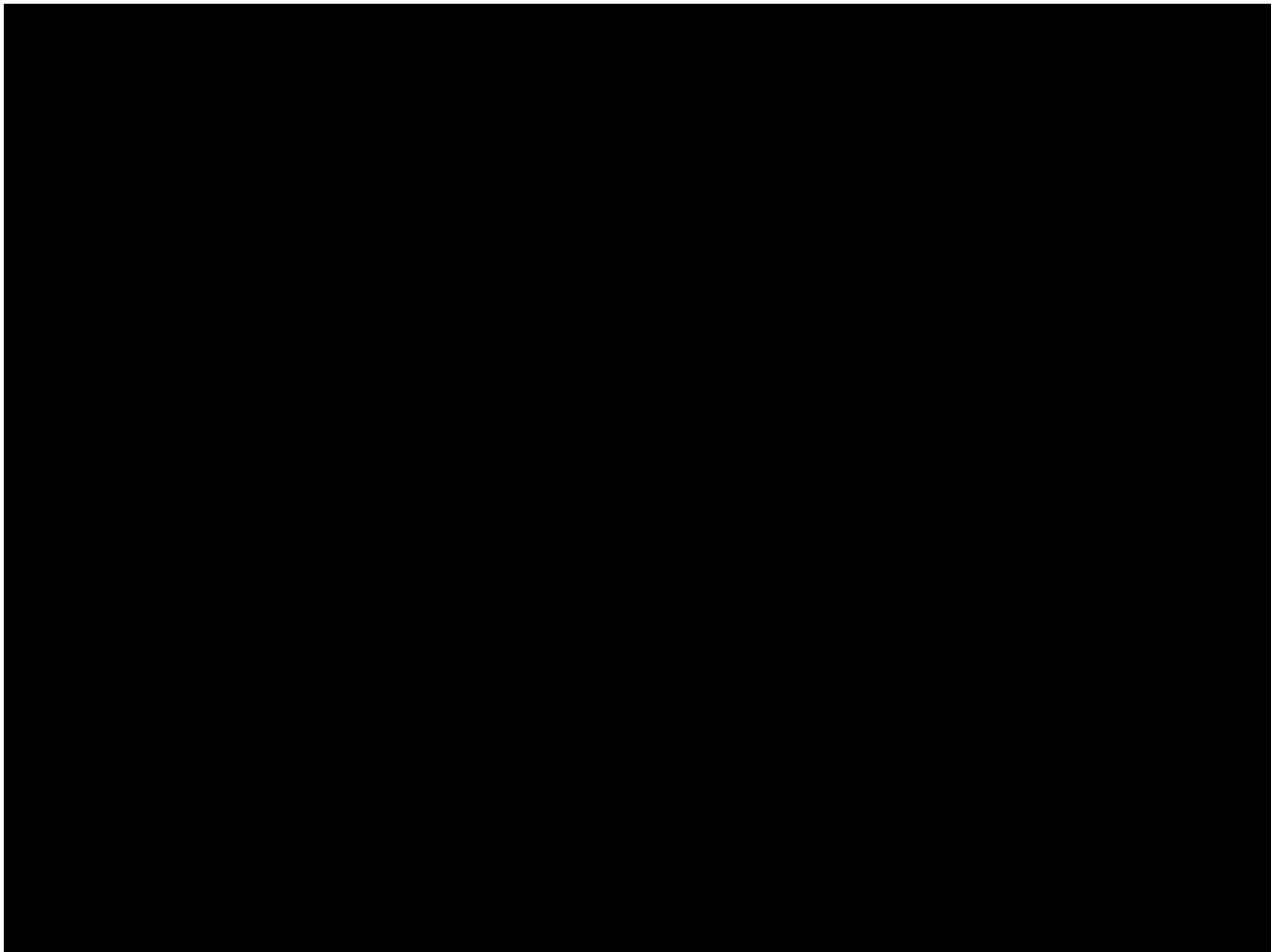
- Get familiar with the OpenAI gym (pip install gym)
  - <https://github.com/romeokienzler/DeepRL> => t-randomplay.py
- Create training data by randomly sample the environment (using random actions)
  - <https://github.com/romeokienzler/DeepRL> => t-trainingdata.ipynb
- Implement an agent using a policy network
  - <https://github.com/romeokienzler/DeepRL> => t-dlmodel.py

# Influence of Human Priors



<http://bit.ly/playhardcore>



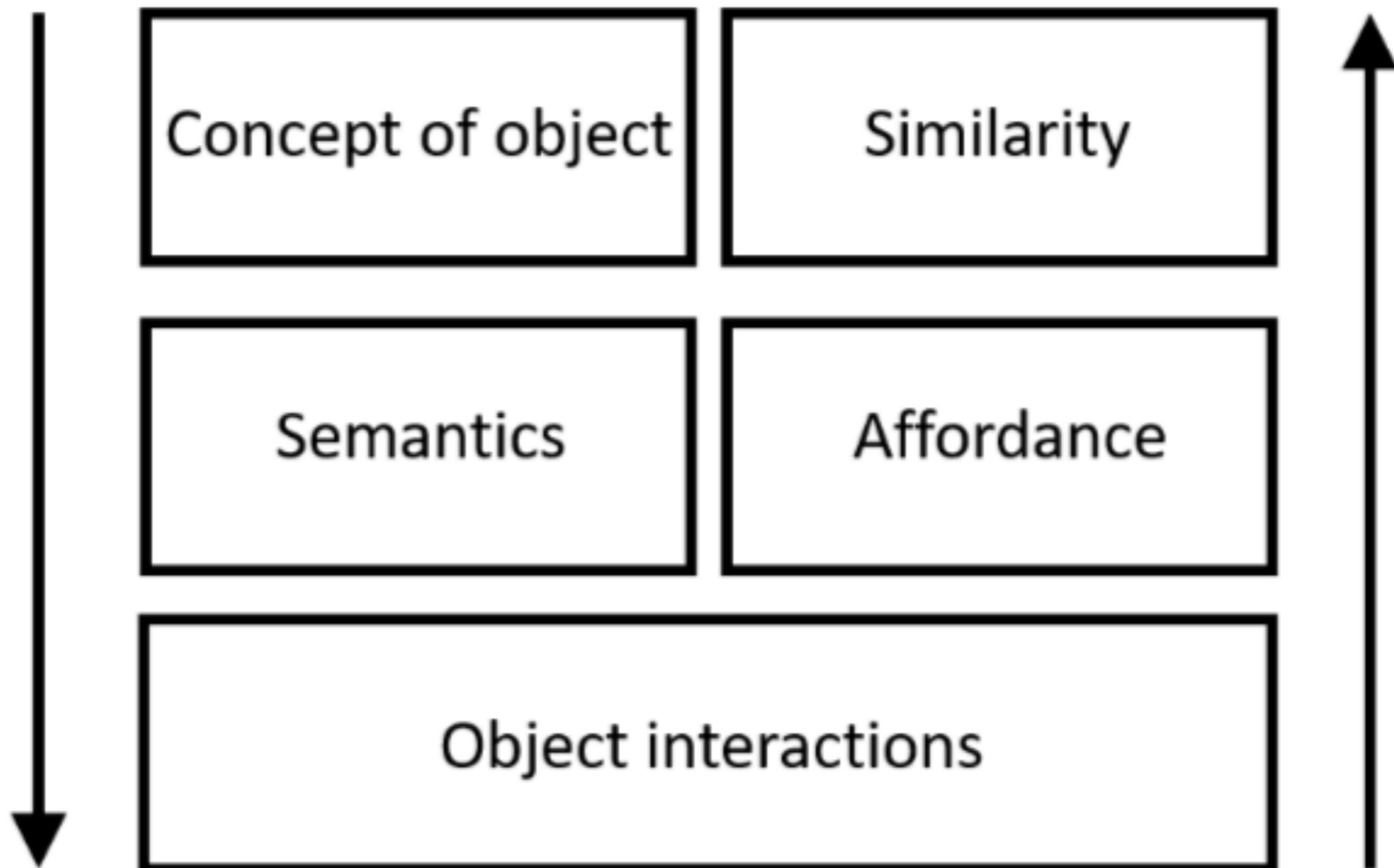


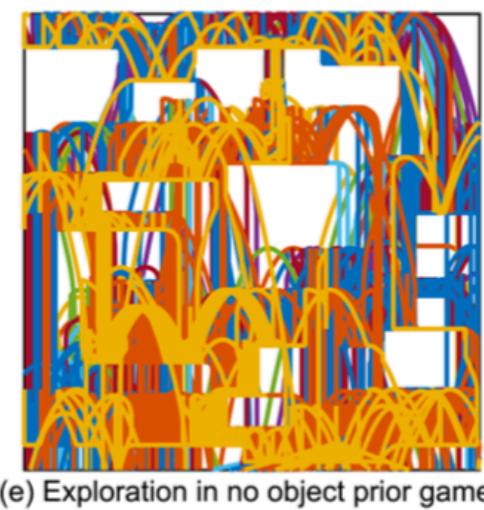
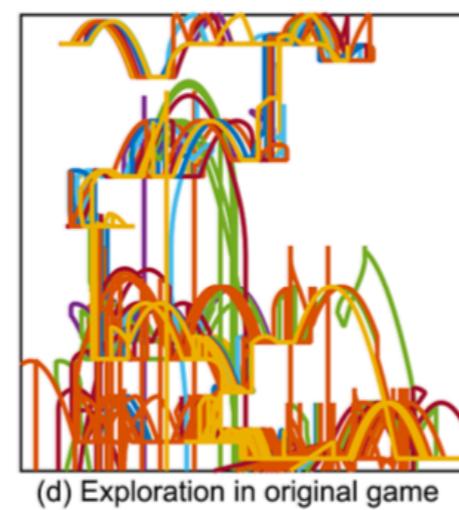
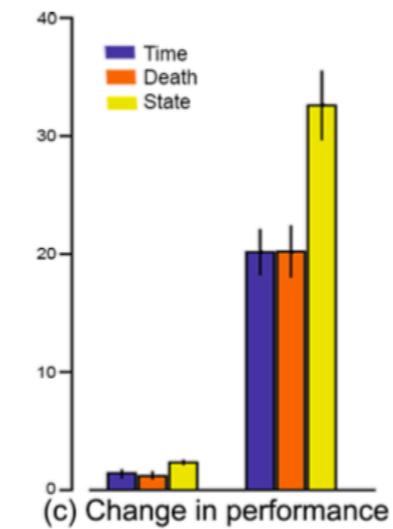
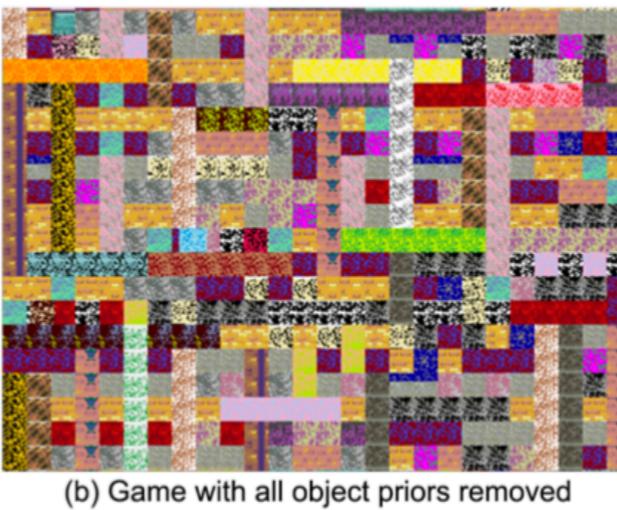
<https://www.youtube.com/watch?v=cJ3Ec70Mnqw>



Learnt first

Importance





# Current Research Focus

- **Gradient update strategies, e.g. TRPO (Trusted Region Policy Optimization), PPO (Proximal Policy Optimization)**
- **Reward shaping**

# micro Hackathon

- In the last exercise you've used a randomly sampled training set of environment interactions
- In this Hackathon, we'll change the behaviour and create an intelligent agent
- The main difference is, that the intelligent agent is used to determine the next action (and therefore state and reward) used for computing the weights updates of the neural network
- Once implemented, please compare (ideally plot) the learning progress of your agent vs. the random sampled training data
- btw. using your agent to decide which part of the environment to explore is called “on-policy”