Administrative:

- Survey - please fill survey in USOS.

- Exam free pass - threshold to be announced a few days before the exam.

- Exam - to be eligible for the exam you need to pass the homeworks:
  - 50% is guaranteed to be enough
  - Possibly lower threshold released before the exam

- Exam on Feb 8 - Saturday (5h), two parts:
  - Test (moodle, no materials allowed).
  - Programming tasks in pytorch.
  - Old exams are on on the course website.

Old exams

Both coding tasks and tests from editions 2020/21-2023/24 are here

Model based RL

Plan for today:

- Exploration & Exploitation (briefly) -> AlphaGo Zero
- Model Based Reinforcement Learning (briefly) -> From GPT to ChatGPT

Exploration & exploitation

Online decision making

- Exploitation - make the best decision given the current knowledge

- Exploration - gather more information

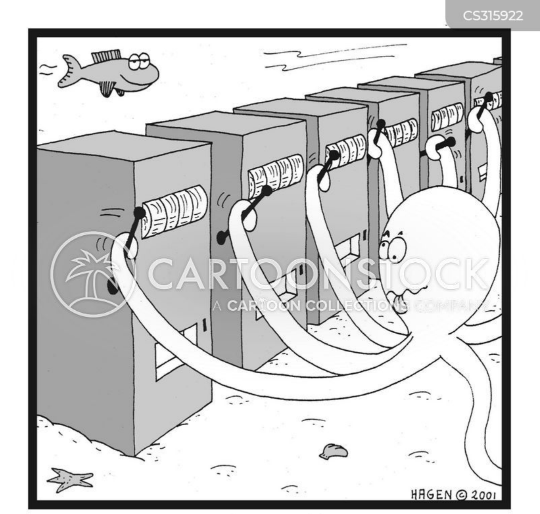
- Optimum long term strategy requires a trade-off between them

Examples

- Lunch selection:
  - Exploitation - pick your favourite dish
  - Exploration - try something new

- Game playing:
  - Exploitation - play the best move (as you currently know)
  - Exploration - try an experimental move

Multi-armed bandits



Compulsive gambling

- Set of bandits (actions) A
- Each turn an action is chosen independently
- The environment generates a reward $Q(a) = \mathbb{E}[r|a]$
  - **Reward depends only on most recent action!**
- Goal: maximize total rewards
  - e.g., recommender systems
  - exploration vs exploitation

Regret minimization

- Maximizing reward is equivalent to minimizing regret
  - a* - best action in hindsight

$$R_t = \sum_{i=1}^{t} (\mathbb{E}[a^*] - \mathbb{E}[r|a_t])$$

- Regret minimization is an established mathematical problem.

Greedy action selection

- Maintain Monte Carlo estimates of the action values:

$$\bar{Q}_t(a) = 1/N_t(a) \sum_{i=1}^{t} [r_t \text{ if } a_t = a \text{ else } 0]$$

- Greedy always picks action with highest estimate

$$a_t = \text{argmax}_a \bar{Q}_t(a)$$

- Problem - greedy can use suboptimal action forever

Epsilon greedy

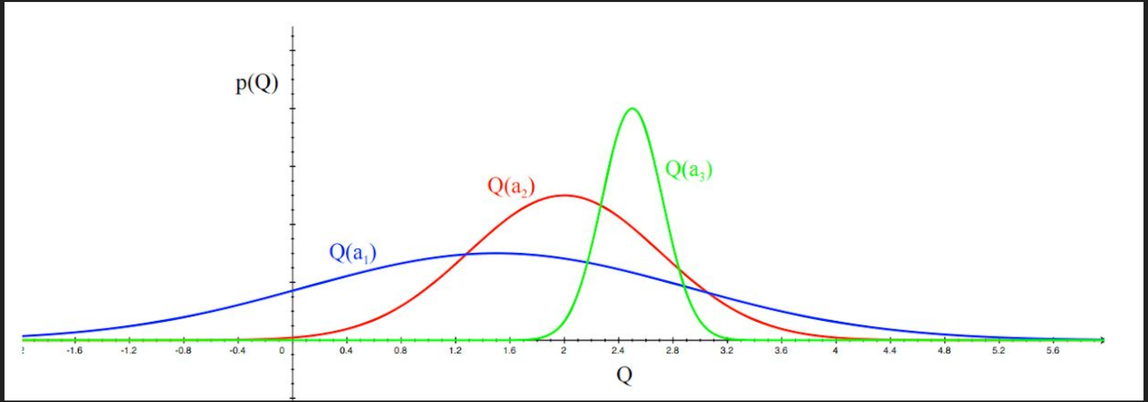
ε-greedy exploration strategy:

- With probability ε act randomly

- With probability (1-ε) act greedily
    - according to the current estimates

Modeling rewards as distributions

- Consider having distribution modeling current belief about the expected rewards

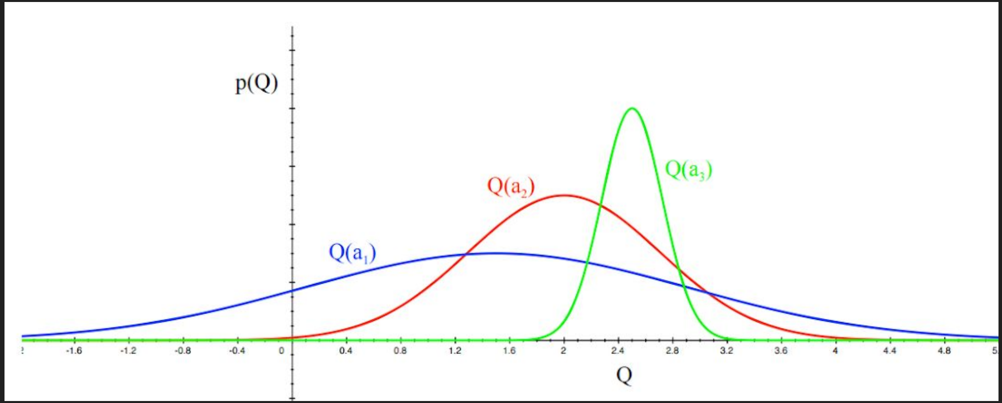


- Which action should we pick?
    - Uncertain actions are worth exploring

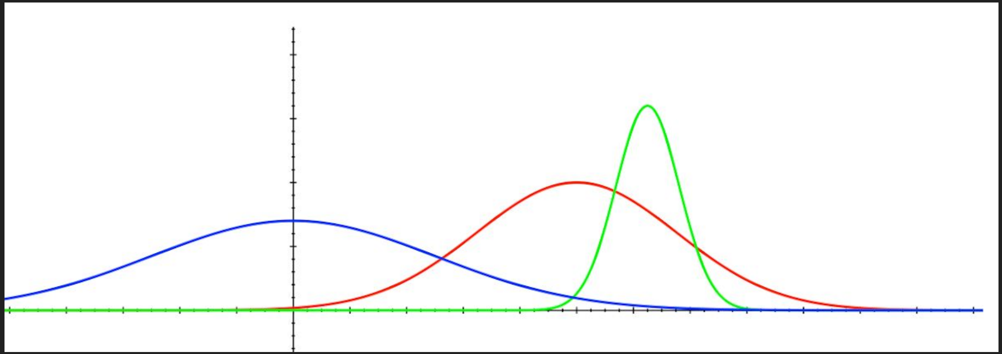
source

Modeling rewards as distributions

- Current belief:



- Updated belief after picking blue



source

UCB - Upper Confidence Bounds

- Maintain uncertainty of each action $\bar{U}_t(a)$ so that with high probability $Q(a) \leq \bar{Q}_t(a) + \bar{U}_t(a)$
- Select an action maximizing UCB

$$a_t = \text{argmax}_a \, \bar{Q}_t(a) + \bar{U}_t(a)$$

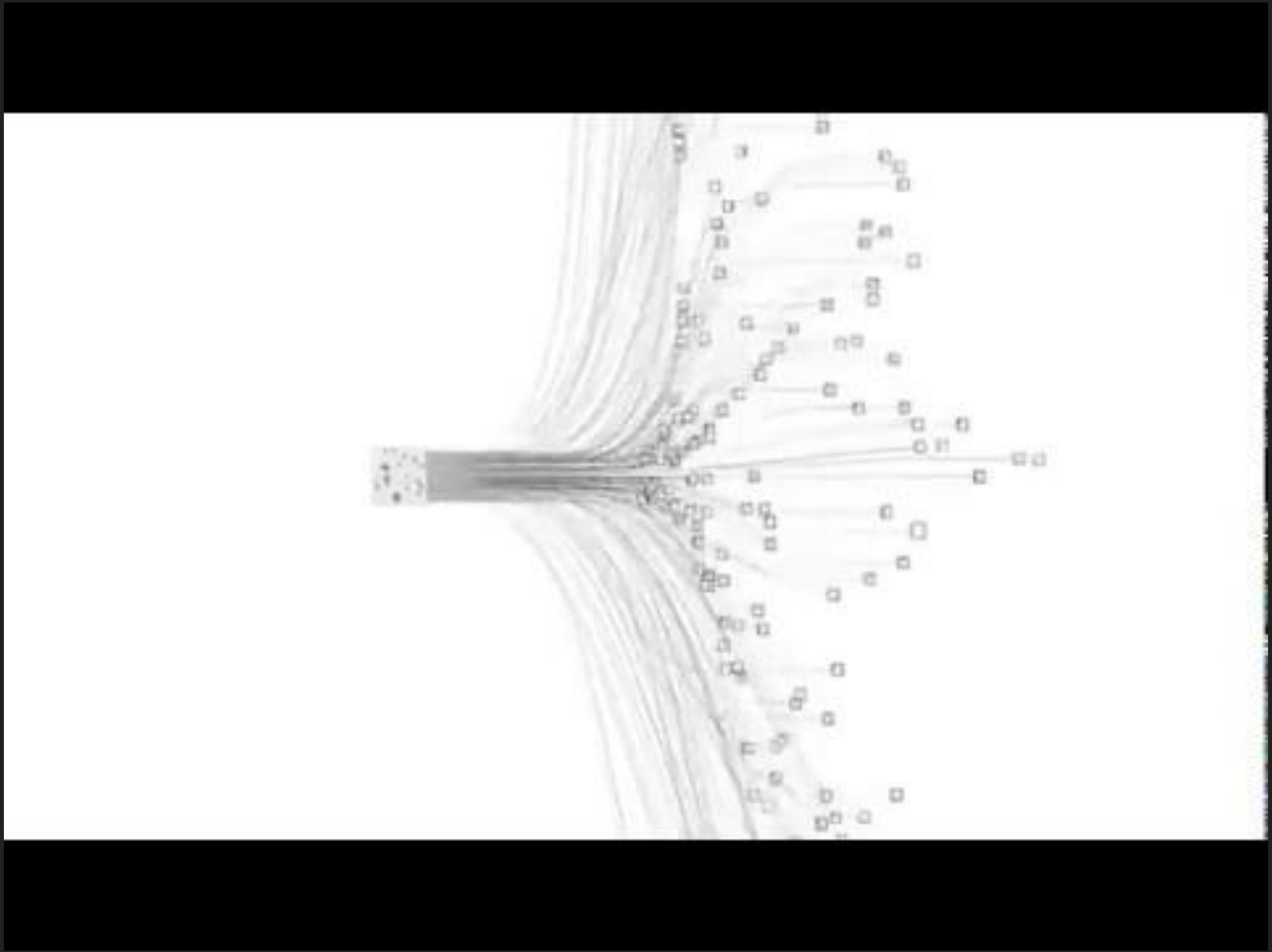
- Update bounds for **all** actions (not only the selected one)

$$\bar{U}_t(a) = \sqrt{2 \log t / N_t(a)}$$

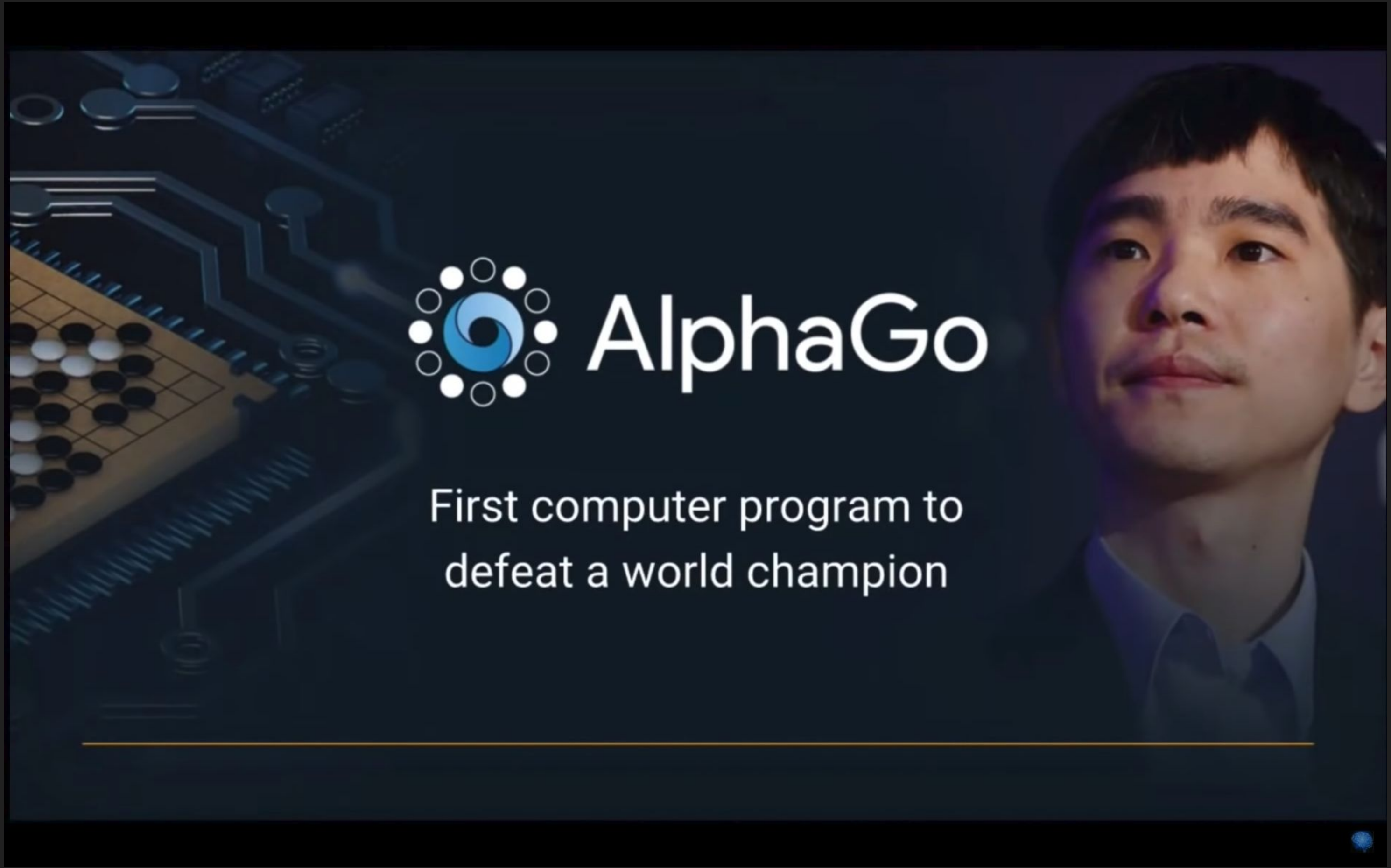
AlphaGo Zero


Go in numbers
3,000
Years Old
40M
Players
10^170
Positions
Google DeepMind

source

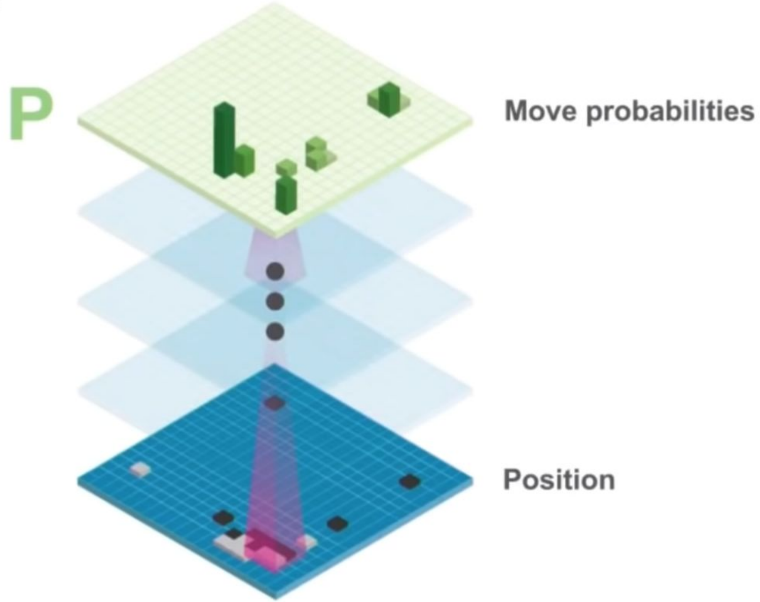


source



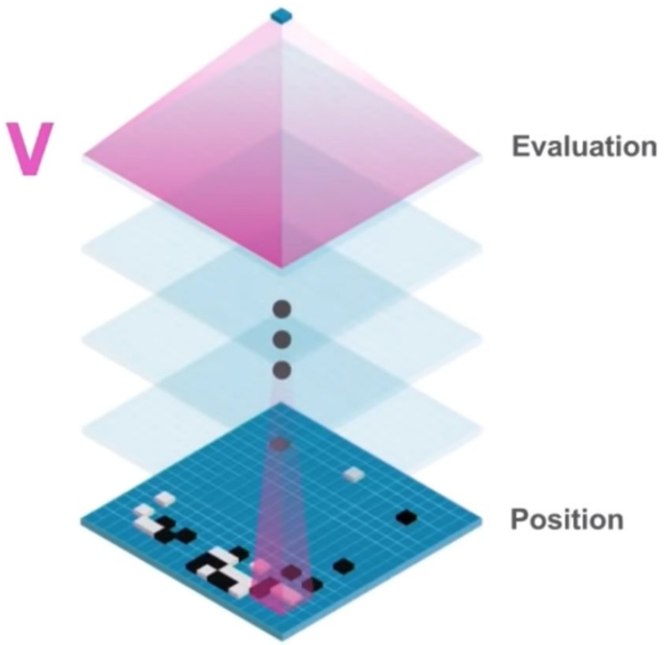
source

Policy network

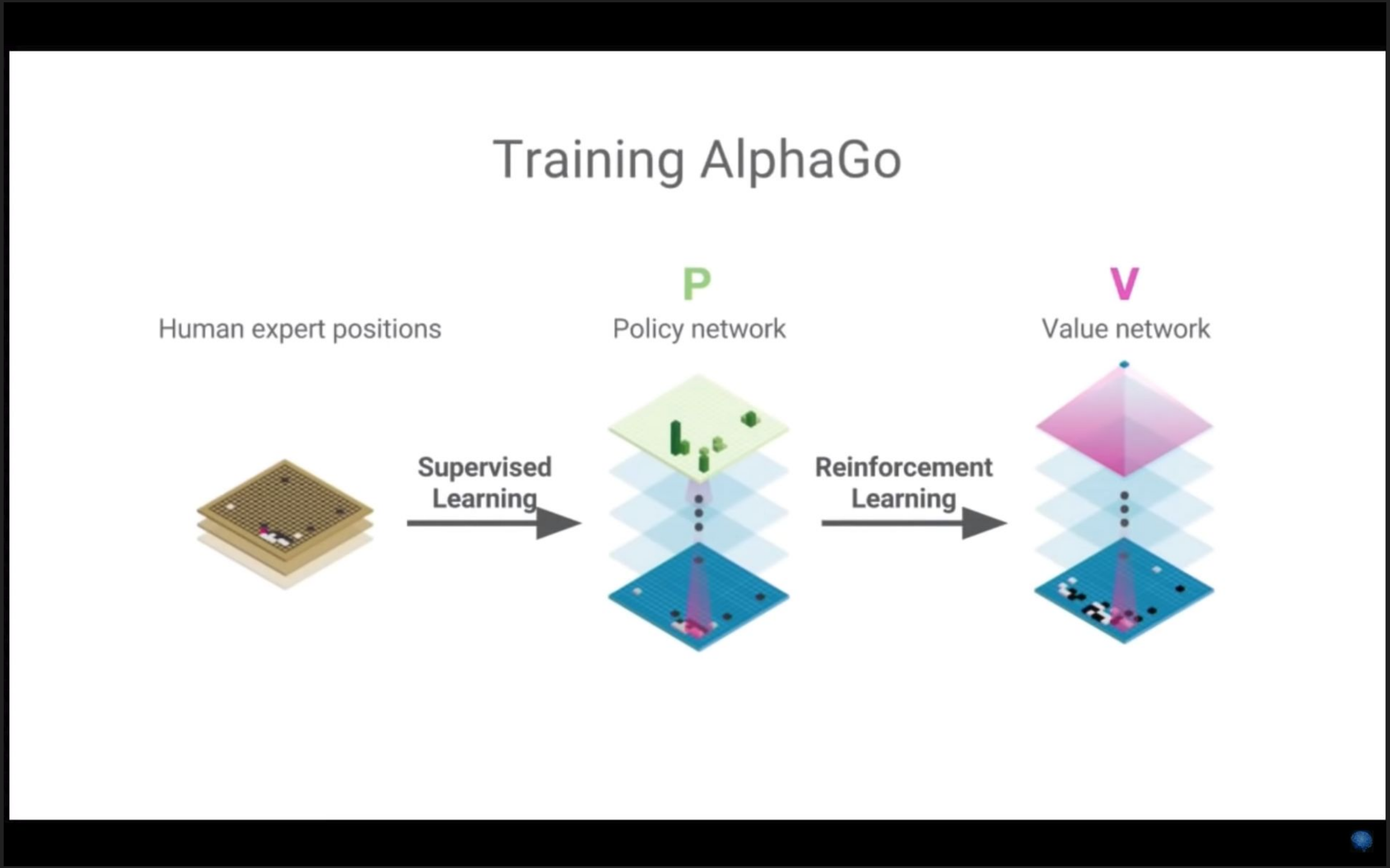


source

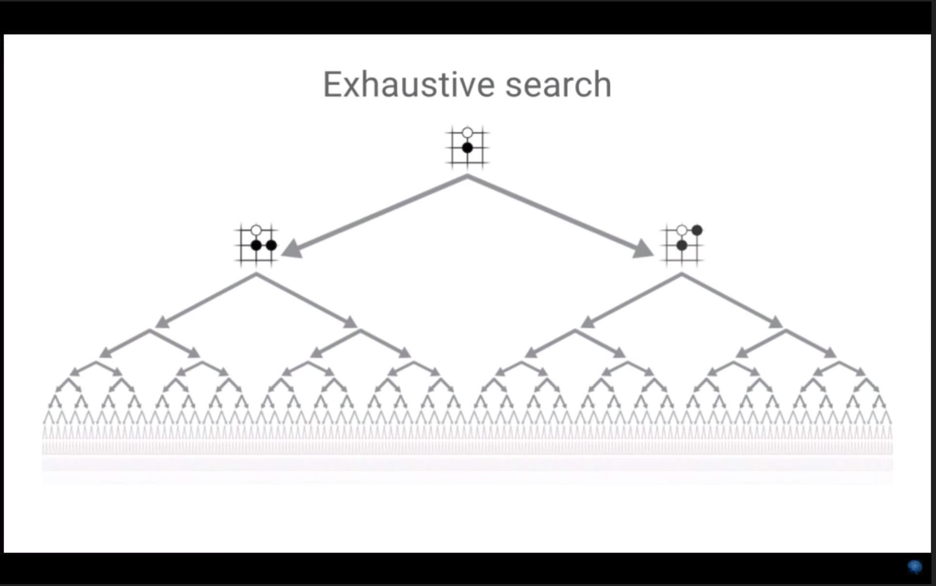
Value network

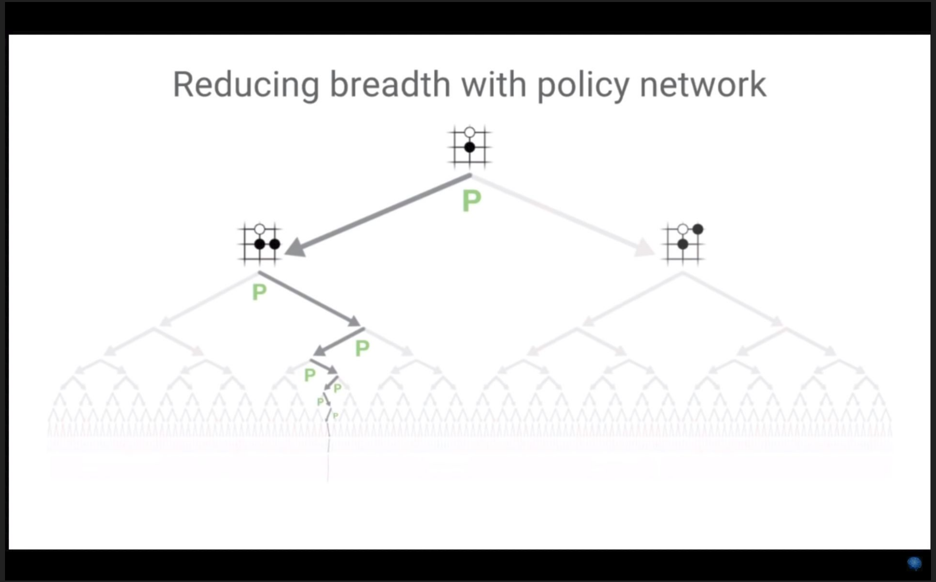


source

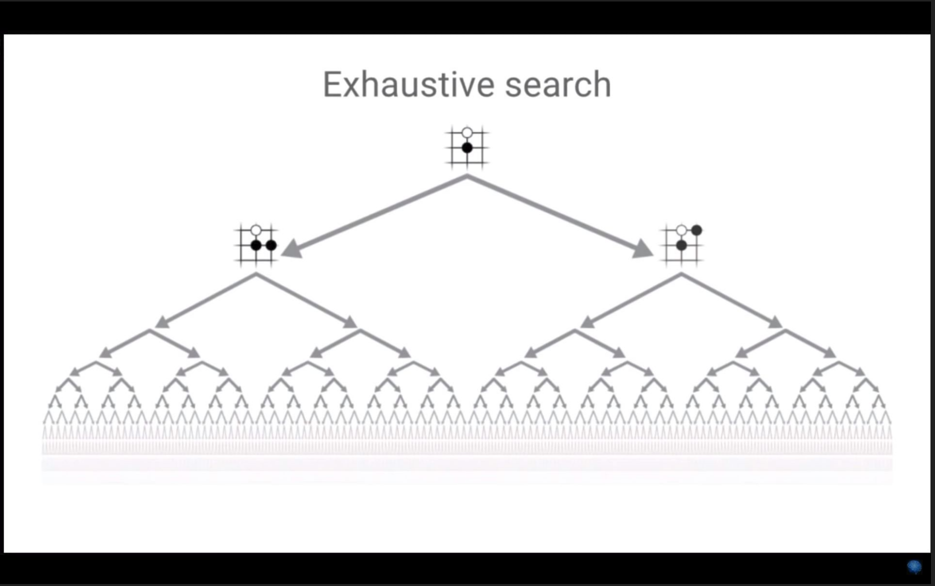

Training AlphaGo
Human expert positions
Supervised Learning
P
Policy network
Reinforcement Learning
V
Value network

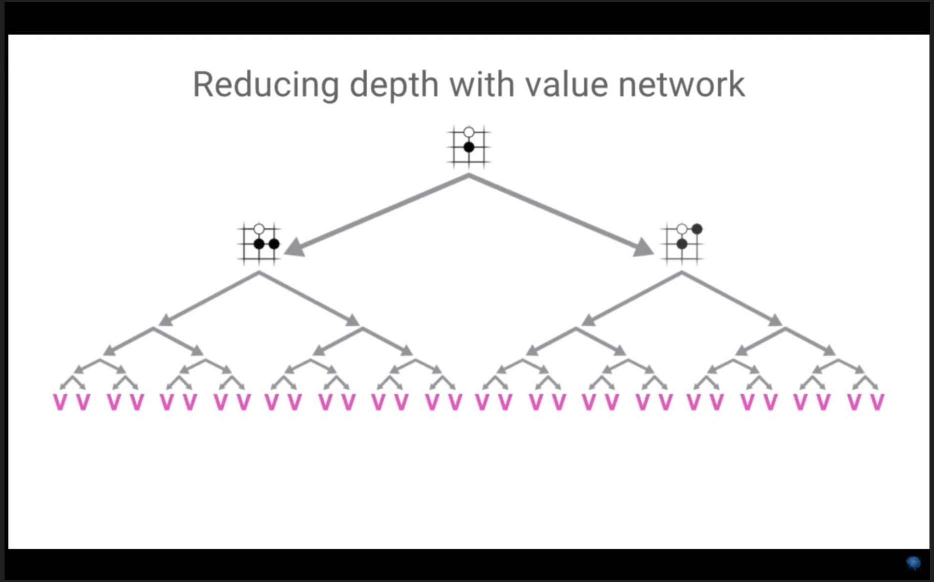
source


Exhaustive search


Reducing breadth with policy network
P
P
P
P

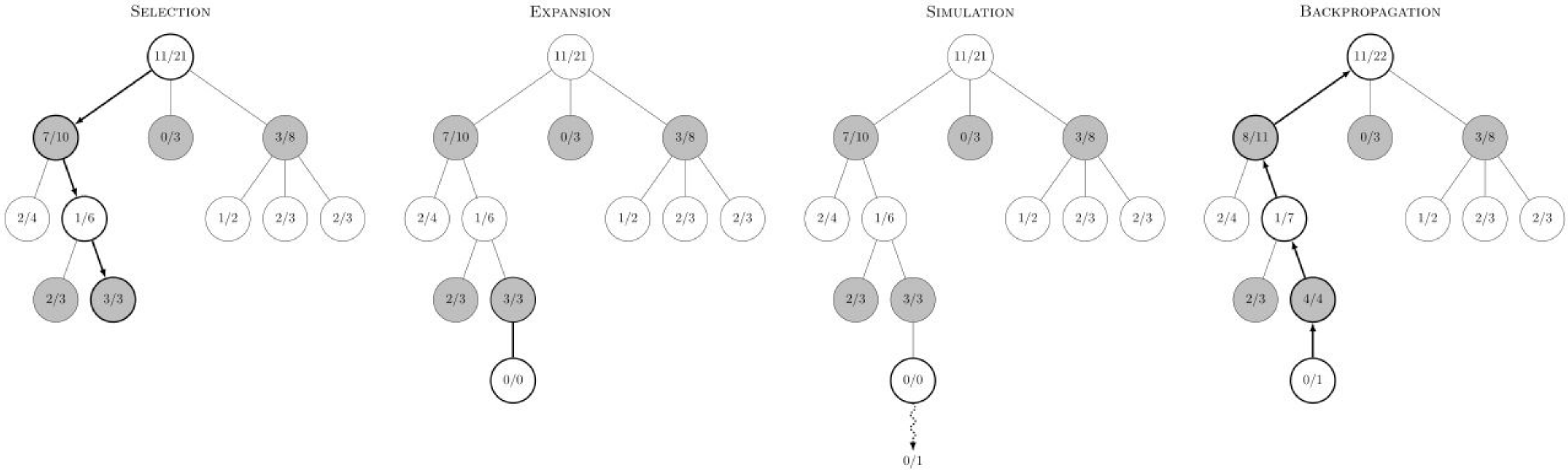
source


Exhaustive search


Reducing depth with value network
V V V V V V V V V V V V V V V V V V V V V V V V V V V V V V V V

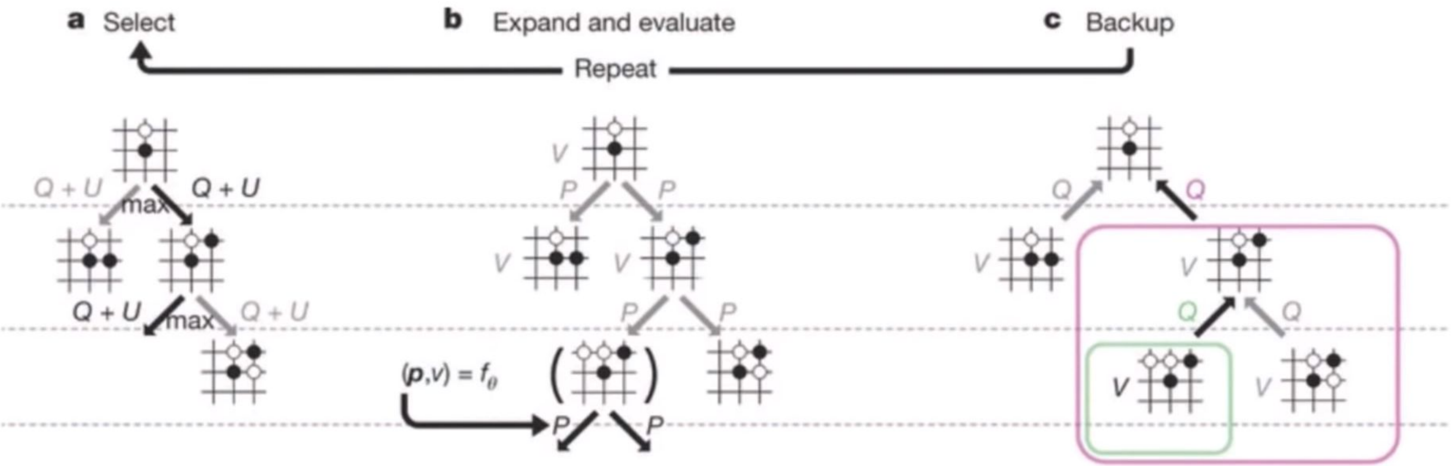
source

MCTS - Monte Carlo Tree Search



source

Monte-Carlo tree search in AlphaGo



source

AlphaGo vs Lee Sedol

Lee Sedol (9p): winner of 18 world titles

Match was played in Seoul, March 2016

AlphaGo won the match 4-1

source

AlphaGo Master vs Ke Jie

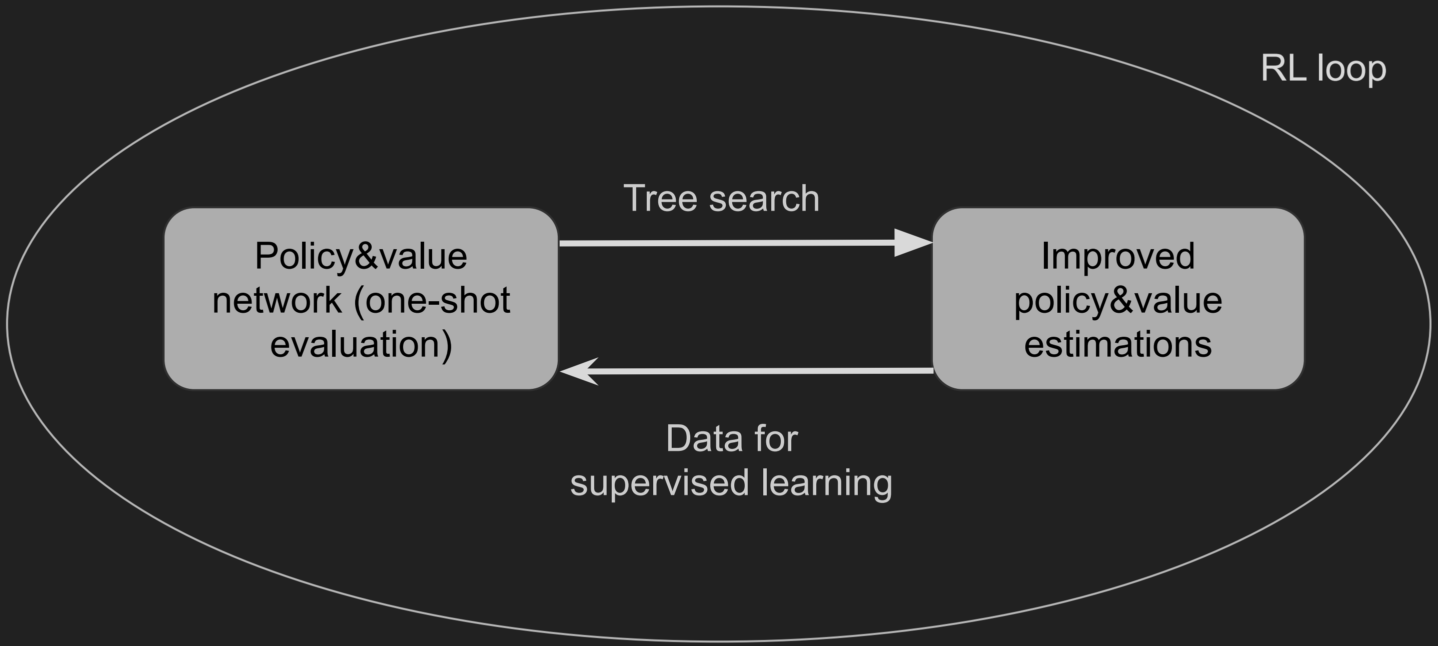
Ke Jie (9p): player ranked #1 in world

Match was played in China, May 2017

AlphaGo Master won the match 3-0

source

source


RL loop
Tree search
Policy&value network (one-shot evaluation)
Improved policy&value estimations
Data for supervised learning

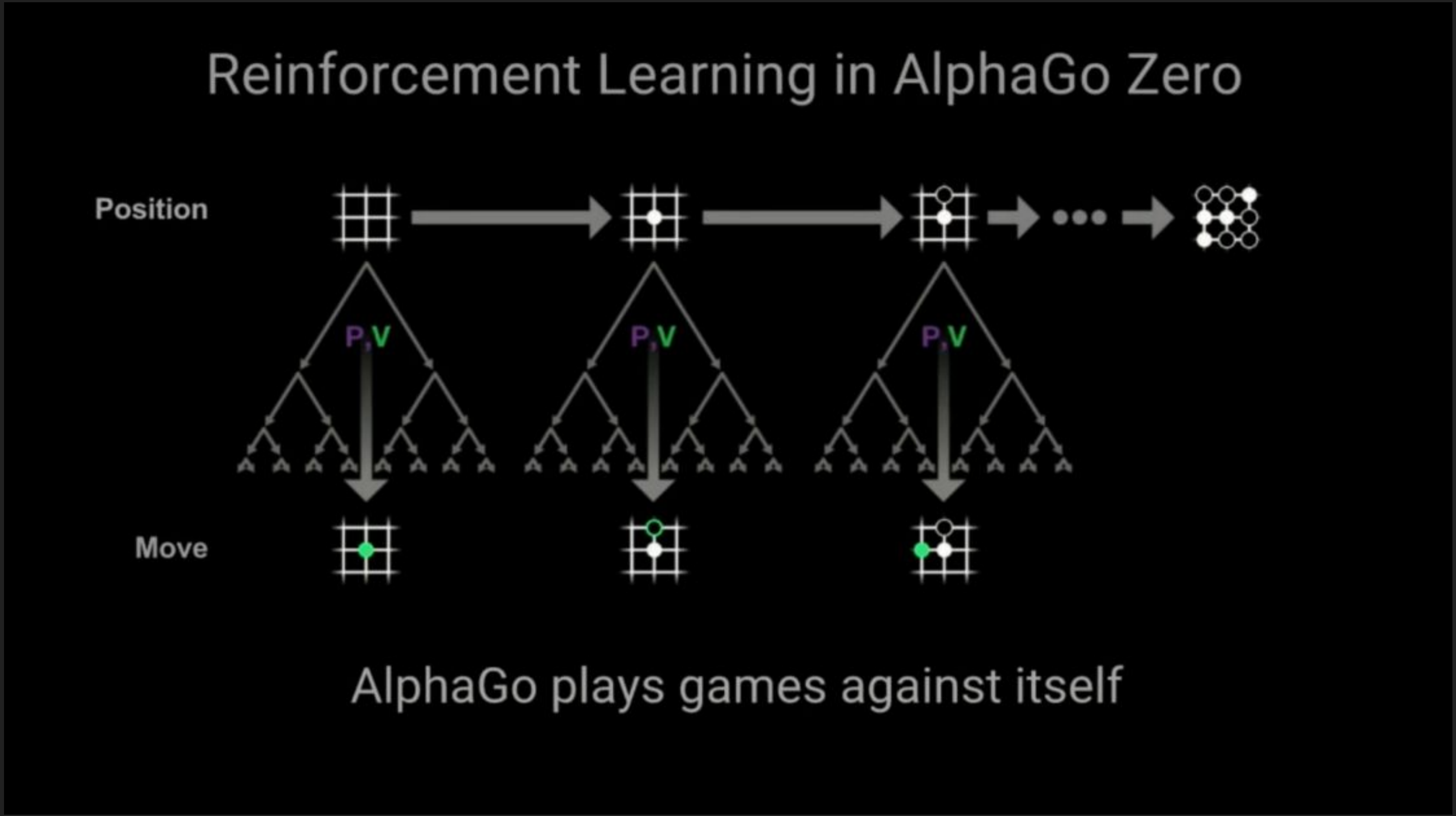
source

AlphaGo Zero: learning from first principles

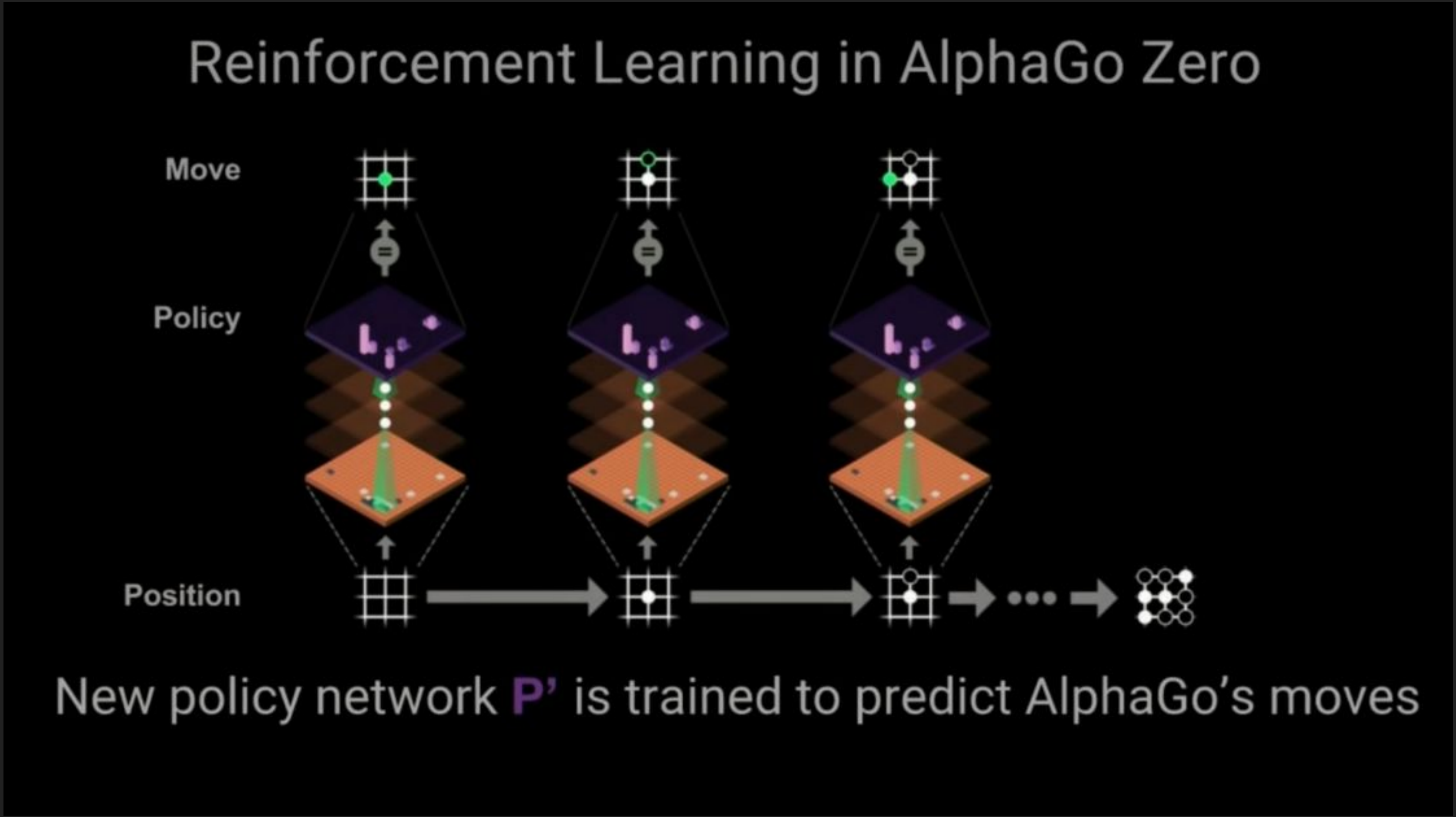
- **No human data**
  - Learns solely by self-play reinforcement learning, starting from random
- **No human features**
  - Only takes raw board as an input
- **Single neural network**
  - Policy and value networks are combined into one neural network (resnet)
- **Simpler search**
  - No randomised Monte-Carlo rollouts, only uses neural network to evaluate

Less complexity => more generality

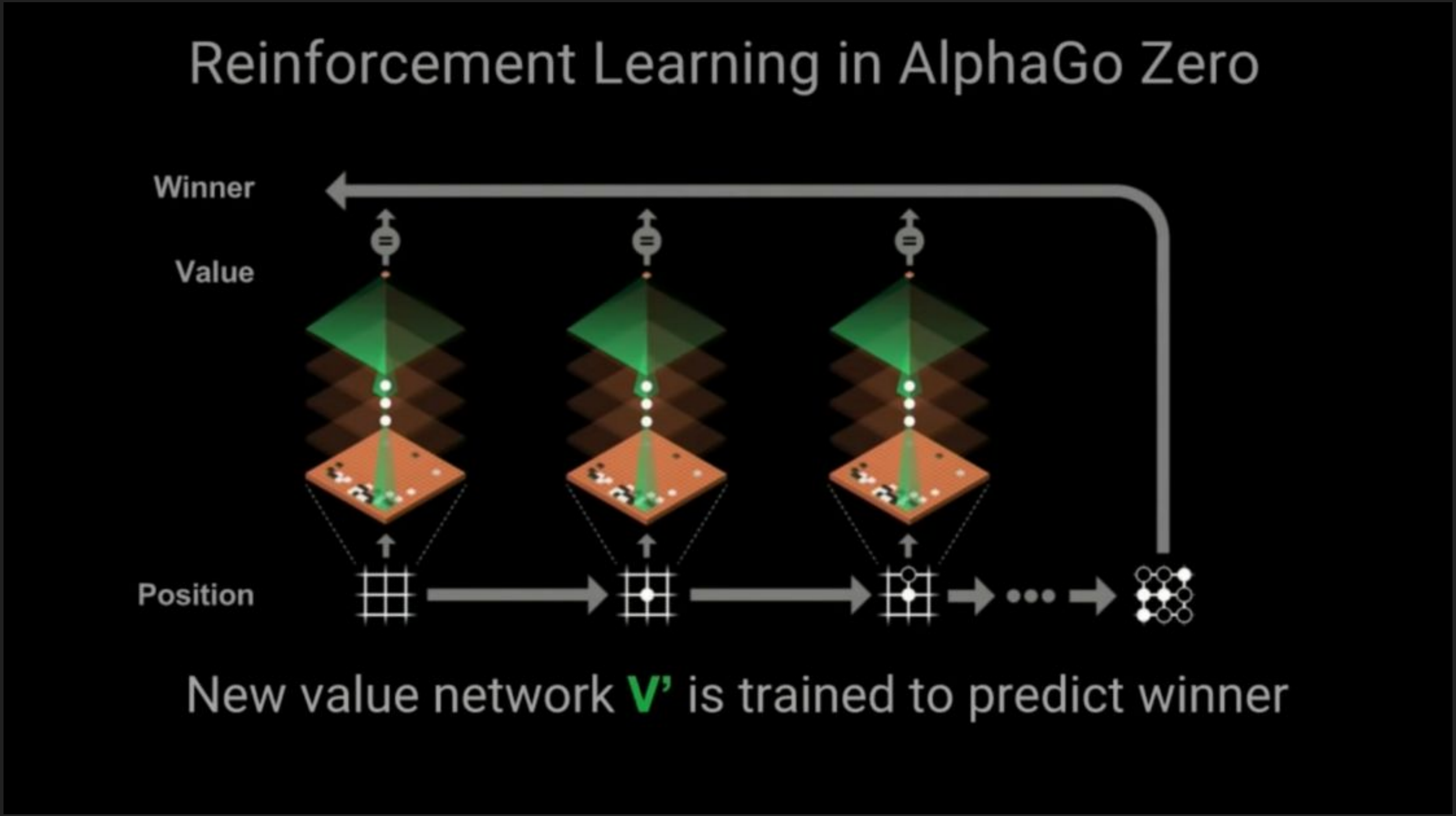
source


Reinforcement Learning in AlphaGo Zero
Position
P, V
P, V
P, V
Move
AlphaGo plays games against itself

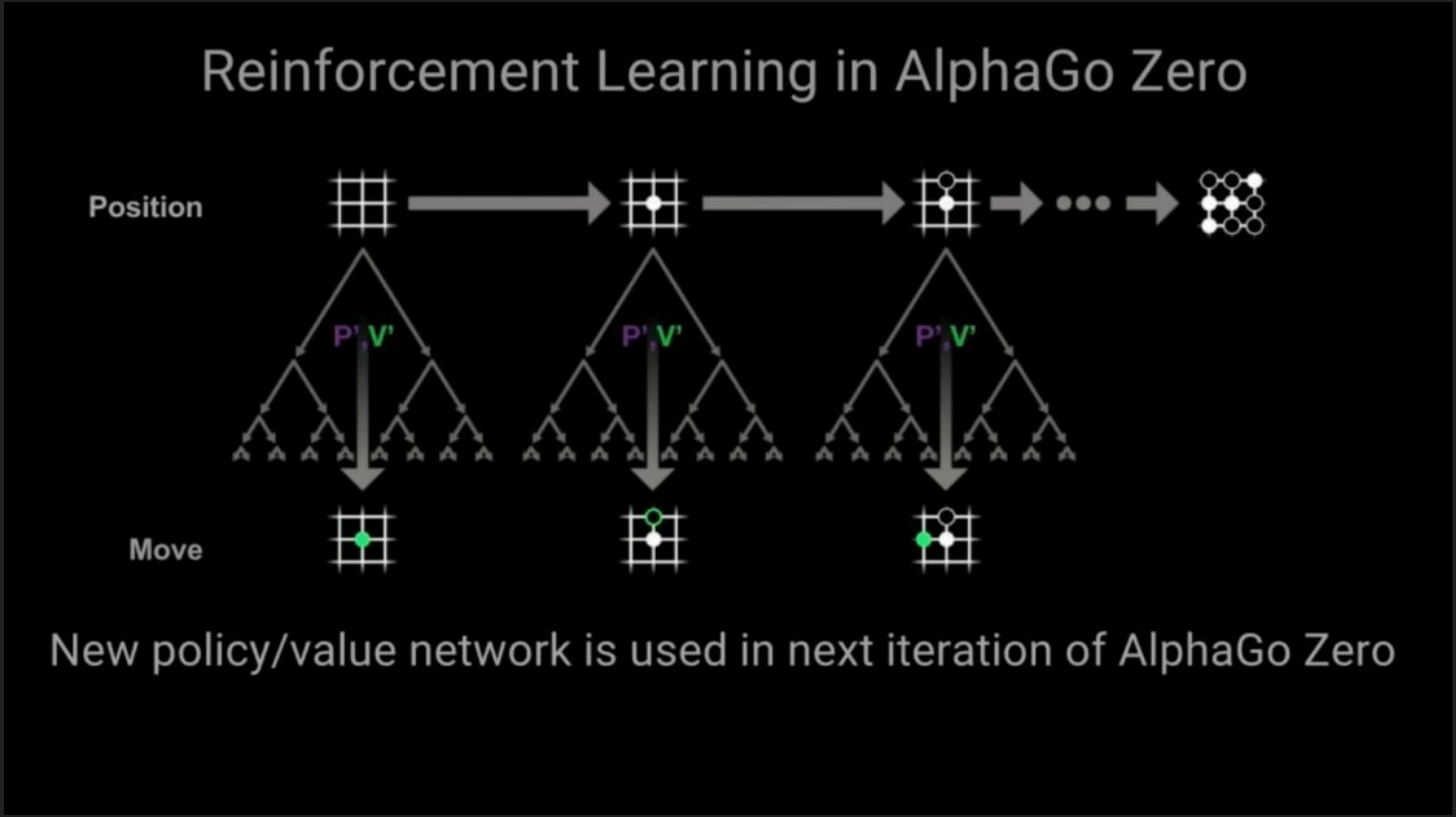
source


Reinforcement Learning in AlphaGo Zero
Move
Policy
Position
New policy network P' is trained to predict AlphaGo's moves

source


Reinforcement Learning in AlphaGo Zero
Winner
Value
Position
New value network V' is trained to predict winner

source


Reinforcement Learning in AlphaGo Zero
Position
P', V'
P', V'
P', V'
Move
New policy/value network is used in next iteration of AlphaGo Zero

source

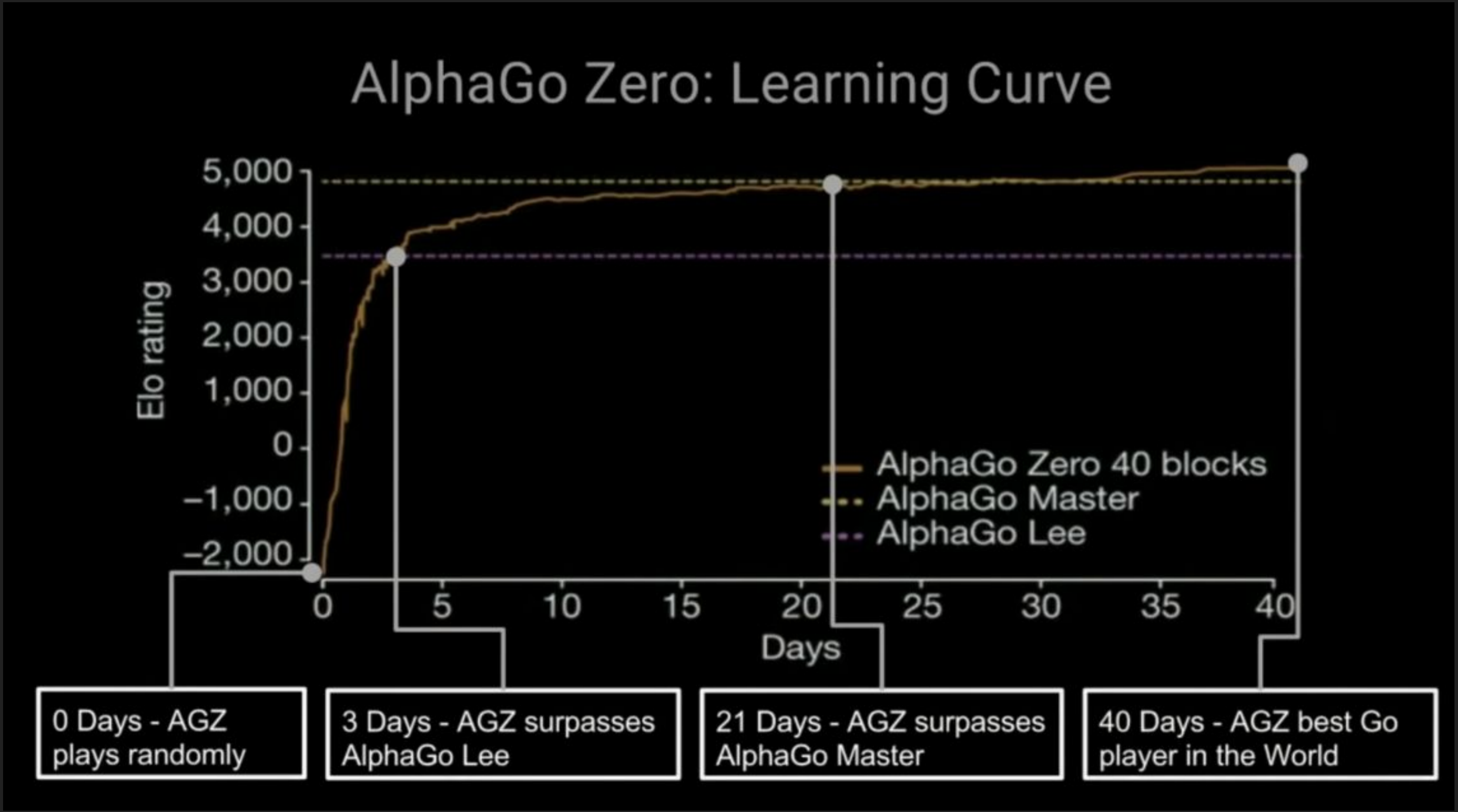
Search-Based Policy Iteration

- **Search-Based Policy Improvement**
  - Run MCTS search using current network
  - Actions selected by MCTS > actions selected by raw network

- **Search-Based Policy Evaluation**
  - Play self-play games with AlphaGo search
  - Evaluate improved policy by the average outcome

See also: Lagoudakis 03, Scherrer 15, Anthony 17

source


AlphaGo Zero: Learning Curve
5,000
4,000
3,000
2,000
1,000
0
−1,000
−2,000
Elo rating
0
5
10
15
20
25
30
35
40
Days
AlphaGo Zero 40 blocks
AlphaGo Master
AlphaGo Lee
0 Days - AGZ plays randomly
3 Days - AGZ surpasses AlphaGo Lee
21 Days - AGZ surpasses AlphaGo Master
40 Days - AGZ best Go player in the World

[source]

Compute

- Alpha Go:
  Policy network: 10,000 minibatches of 128 games, using 50 GPUs, 1 day.
  Evaluation (MCTS): 40 search threads, 48 CPUs, and 8 GPUs

- AlphaGo Zero:
  Over the course of training, 4.9 million games of self-play were generated, using 1,600 simulations for each MCTS, which corresponds to approximately 0.4s thinking time per move. Parameters were updated from 700,000 mini-batches of 2,048 positions.

  AlphaGo Zero and AlphaGo Master ... single machine with 4 TPUs; AlphaGo Fan and AlphaGo Lee .. 176 GPUs and 48 TPUs respectively.

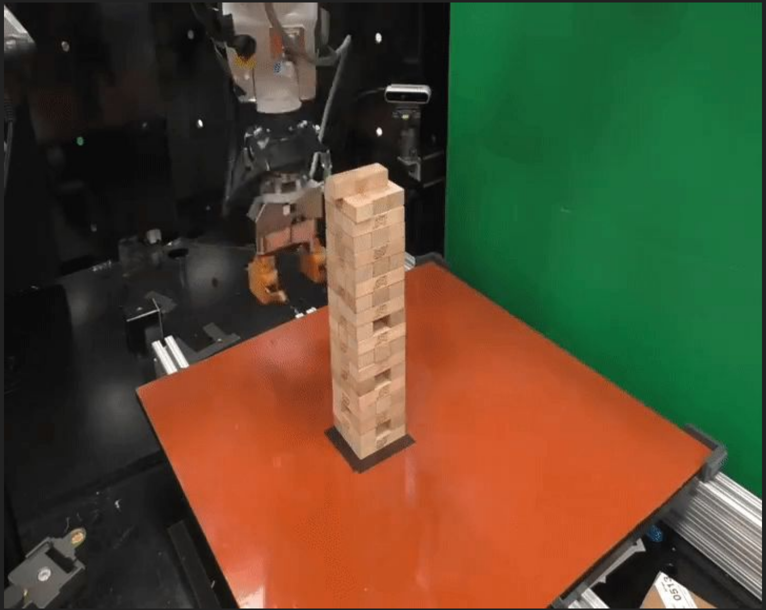
Model based RL

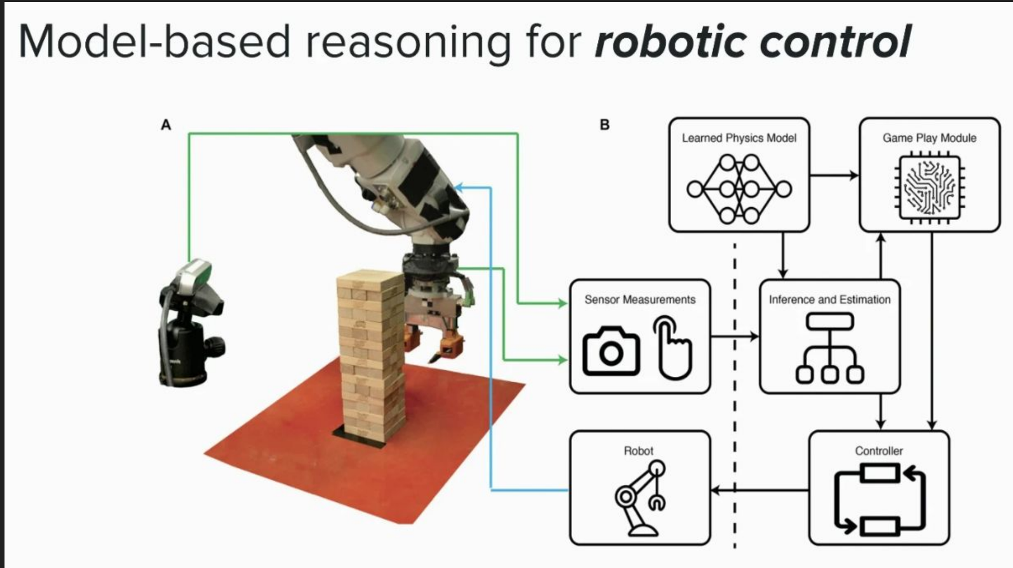
Learning resources

- ICML 2020 Tutorial on
  Model Based Reinforcement Learning:

  https://sites.google.com/view/mbrl-tutorial

Motivation - sample efficiency





Fazeli et al. (2019). See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion. Science Robotics, 4(26).

Why do we want to learn a model?

Planning with real robots

(too expensive, too risky)

Simulating complex physical dynamics

(too expensive)

Interactions with humans

(no access)

https://sites.google.com/view/mbrl-tutorial

Model-free vs model-based

- Collect data $D=\{s_t, a_t, r_{t+1}, s_{t+1}\}$

- Model-free
  - data → policy

- Model -based
  - Data→ model → policy

What is a model?

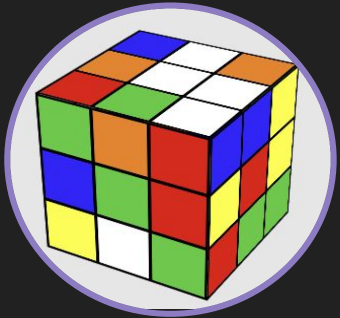
- Some explicit representation of the knowledge about the environment and task
- Usually:
  - Transition/dynamics model: $s_{t+1}=f(s_t,a_t)$
  - Reward model: $r_{t+1}=f(s_t,a_t)$
- Other cases:
  - Inverse transition (going backwards in time)
  - Modeling distance between states

Environment simulators

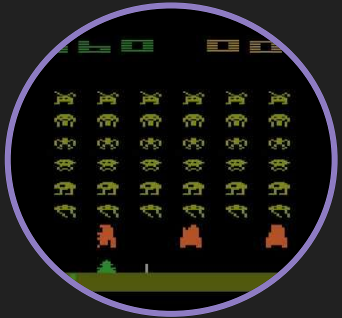
Sometimes we know the ground truth dynamics and rewards.

Might as well use them!

Silver et al. (2016). Mastering the game of Go with deep neural networks and tree search. Nature.



Agostinelli et al. (2019). Solving the Rubik's cube with deep reinforcement learning and search. Nature Machine Intelligence.



Bellemare et al. (2013). The Arcade Learning Environment: An Evaluation Platform for General Agents. JAIR.

Todorov et al. (2012). MuJoCo: A physics engine for model-based control. IROS.

Dimensions of learned models

States vs. **Observations** vs. **Latent States**

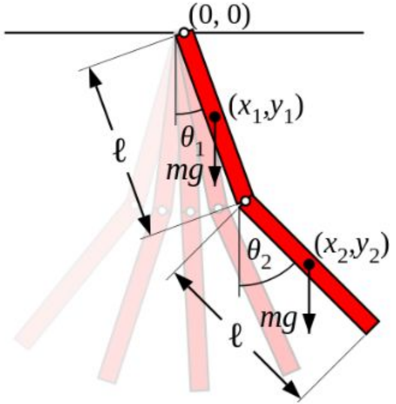
$$s = [x, \dot{x}, \phi]$$

$$o =$$ 

$$z \in \mathbb{R}^{d_z}$$

State transitions: dynamical systems

Double pendulum (acrobot)



$$x_1 = \frac{l}{2}\sin\theta_1 \qquad x_2 = l\left(\sin\theta_1 + \tfrac{1}{2}\sin\theta_2\right)$$

$$y_1 = -\frac{l}{2}\cos\theta_1 \qquad y_2 = -l\left(\cos\theta_1 + \tfrac{1}{2}\cos\theta_2\right)$$

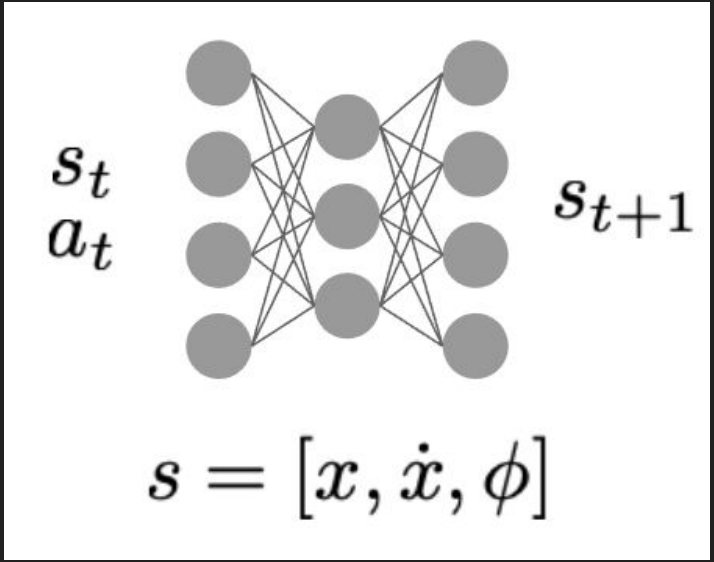
Equations of motion are assumed known.

Reward usually assumed known.

Use system identification to estimate unknown parameters (e.g. mass)

https://sites.google.com/view/mbrl-tutorial

State transitions: MLP-based

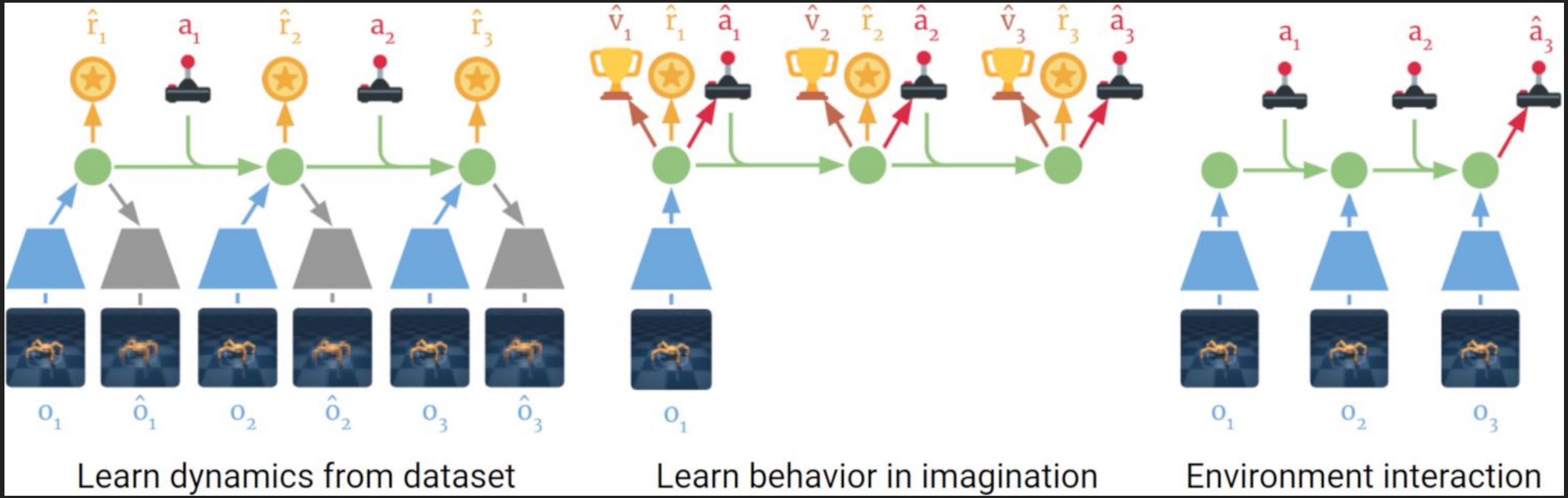


Note: typically better to predict the derivative (change in *s*), and then integrate to obtain s_{t+1}

This is the same idea behind using skip connections / residual blocks!

https://sites.google.com/view/mbrl-tutorial

Latent-space transition models



https://github.com/google-research/dreamer

Model-free vs. model-based RL

Model-free	Model-based	
+	?	Asymptotic rewards*
+	- / +	Computation at deployment*
-	+	Data efficiency*
-	+	Speed to adapt to changing rewards*
-	+	Speed to adapt to changing dynamics*
-	+	Exploration*

*but it depends a lot on the specific method!

https://sites.google.com/view/mbrl-tutorial

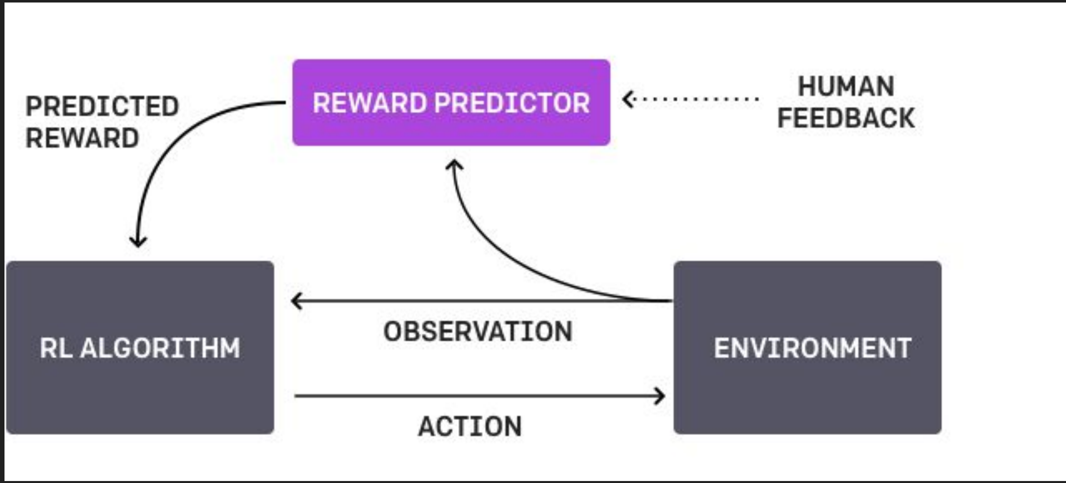
From GPT-3 to ChatGPT

https://openai.com/blog/instruction-following/
https://openai.com/blog/chatgpt/

InstructGPT

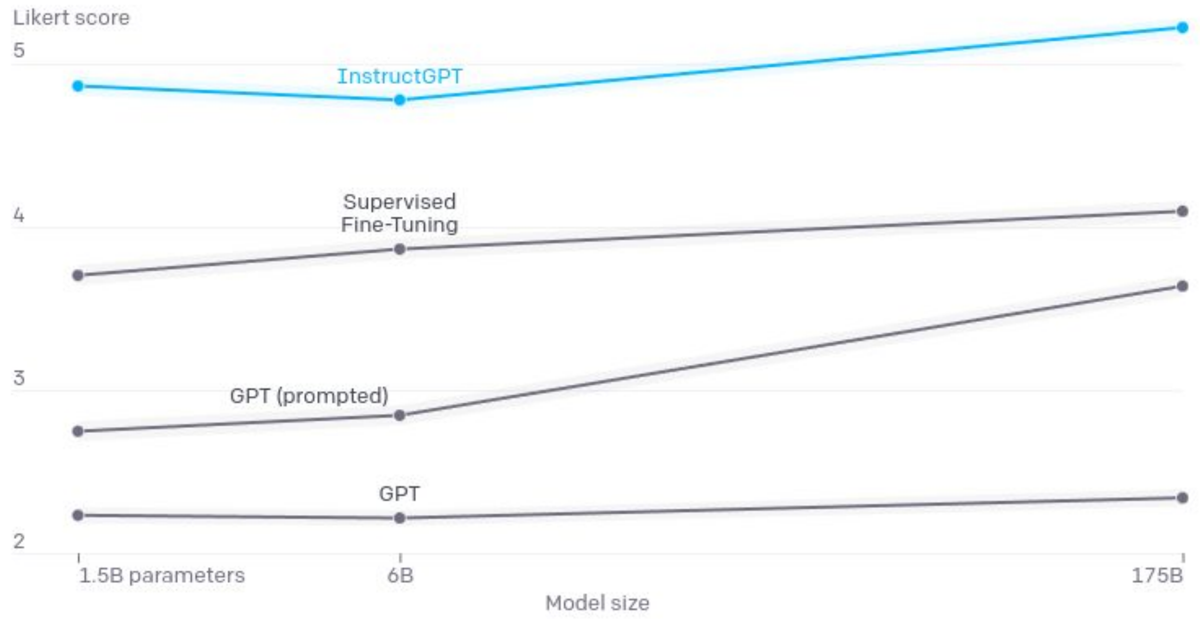
- GPT is trained to predict the next token, not to solve a problem posed by a user

- Goal - use human feedback to guide a higher level policy



https://openai.com/blog/deep-reinforcement-learning-from-human-preferences/

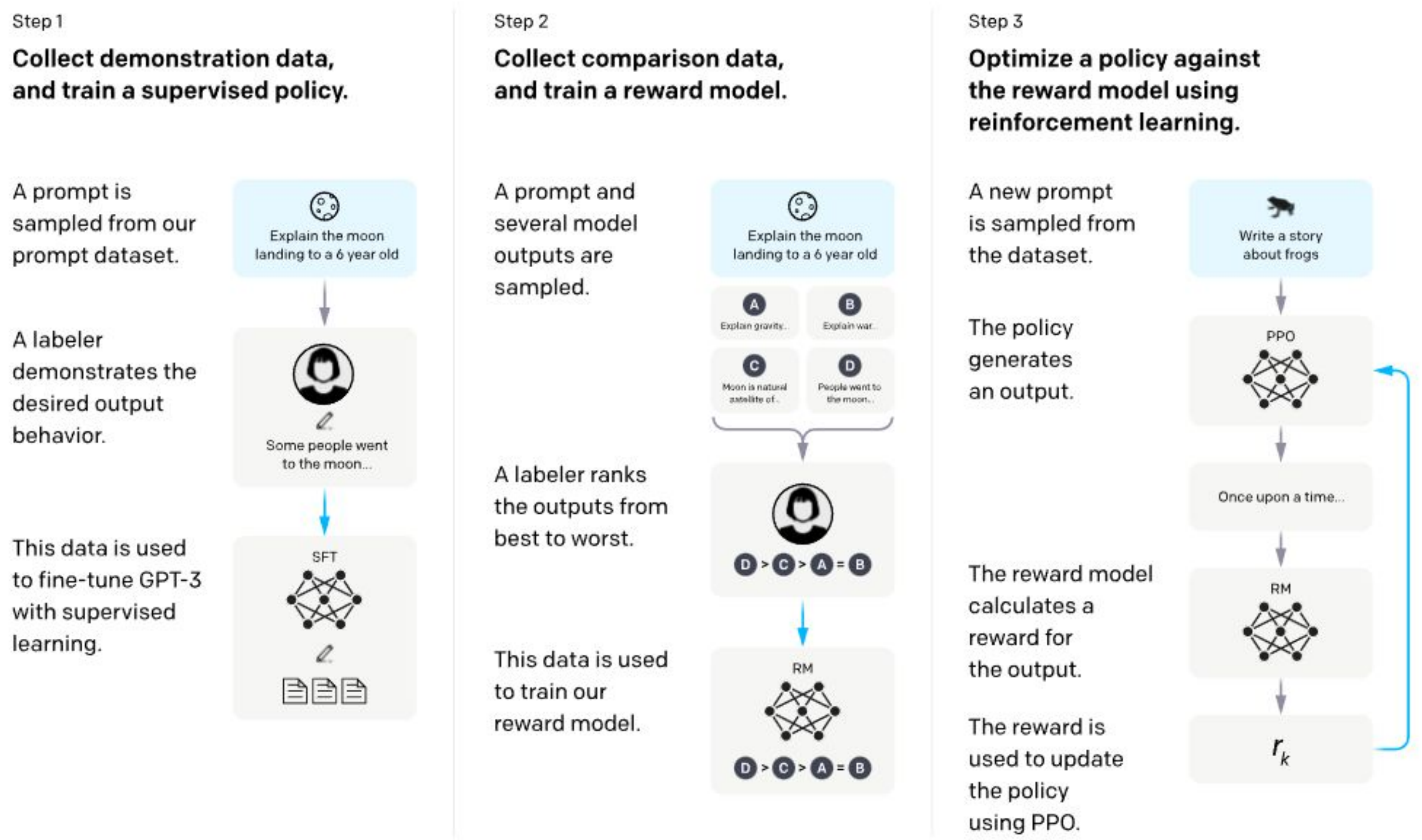
InstructGPT - results



Quality ratings of model outputs on a 1–7 scale (y-axis), for various model sizes (x-axis), on prompts submitted to InstructGPT models on our API. InstructGPT outputs are given much higher scores by our labelers than outputs from GPT-3 with a few-shot prompt and without, as well as models fine-tuned with supervised learning. We find similar results for prompts submitted to GPT-3 models on the API.

https://openai.com/blog/instruction-following/

Methods

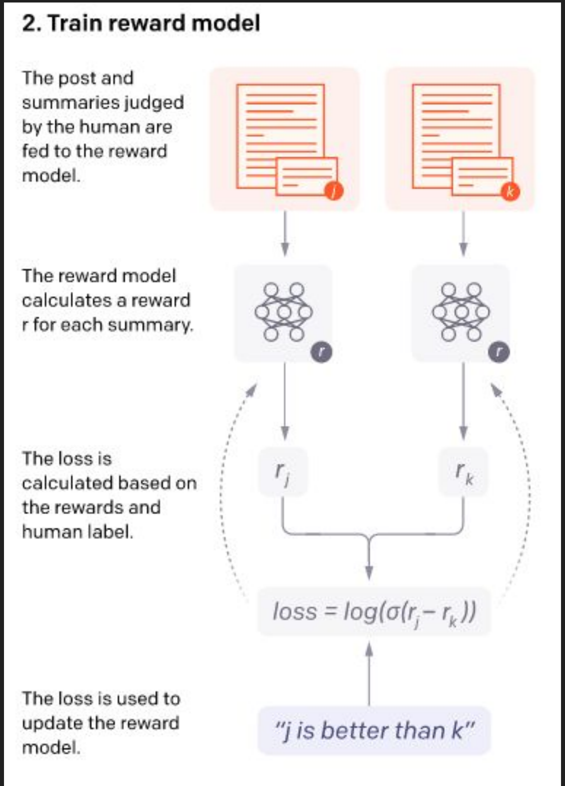


https://openai.com/blog/instruction-following/

Reward model details - work prior to GPT-3

- The reward model calculates absolute reward estimate

- The difference between the absolute values is used to propagate the ranking-based loss.



https://openai.com/blog/learning-to-summarize-with-human-feedback

RL 3-lecture series recap

- Q-learning (DQN)

- Policy Gradients

- Model based RL:
  - Alpha Go Zero
  - Human Feedback RL → InstructGPT (ChatGPT)

Feedback is a gift

Please fill
the course survey in USOS