# Mastering the Game of Go Without Human Knowledge

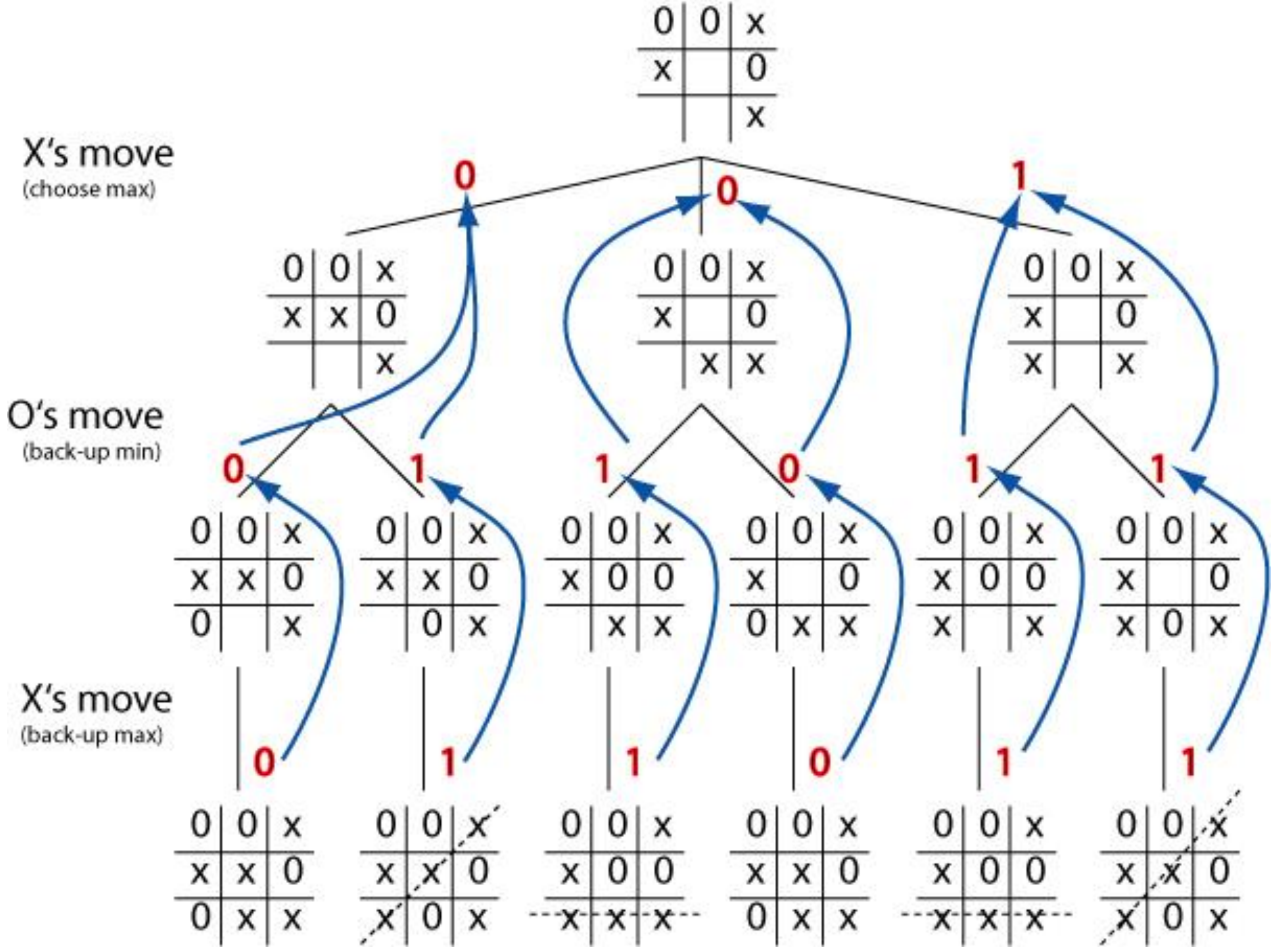
Lead: Liam Hinzman
Facilitators: Tahseen Shabab and Susan Cheng

# AlphaGo Zero

# Overview

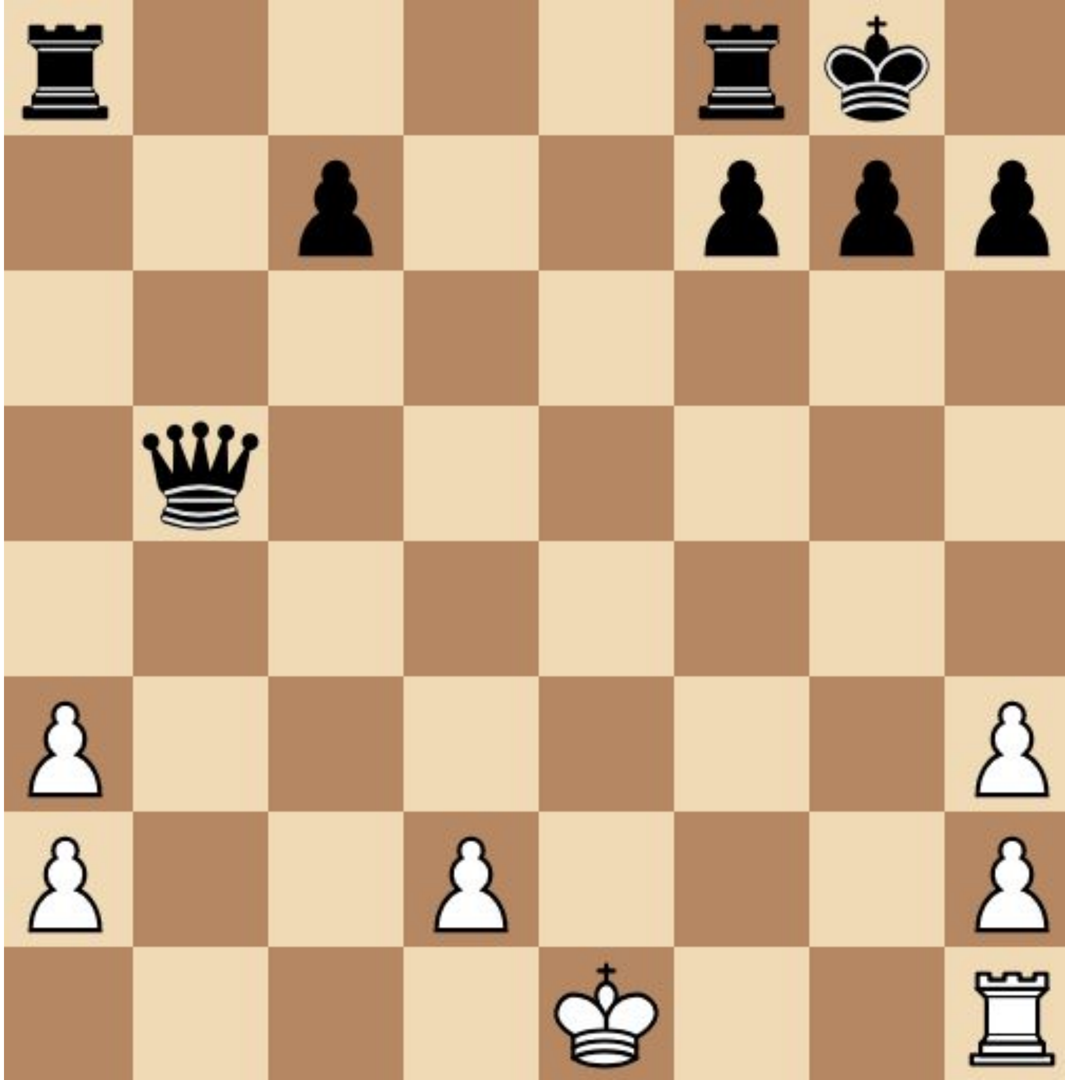
- Brief History of AI in Games
- What is Go and Why Should You Care?
- How AlphaGo Zero Works
- Results
- Discussion

# Minimax


X's move
(choose max)
O's move
(back-up min)
X's move
(back-up max)
0
0
1
0
1
1
0
1
1
0
1
1
0
1
1

# Heuristics

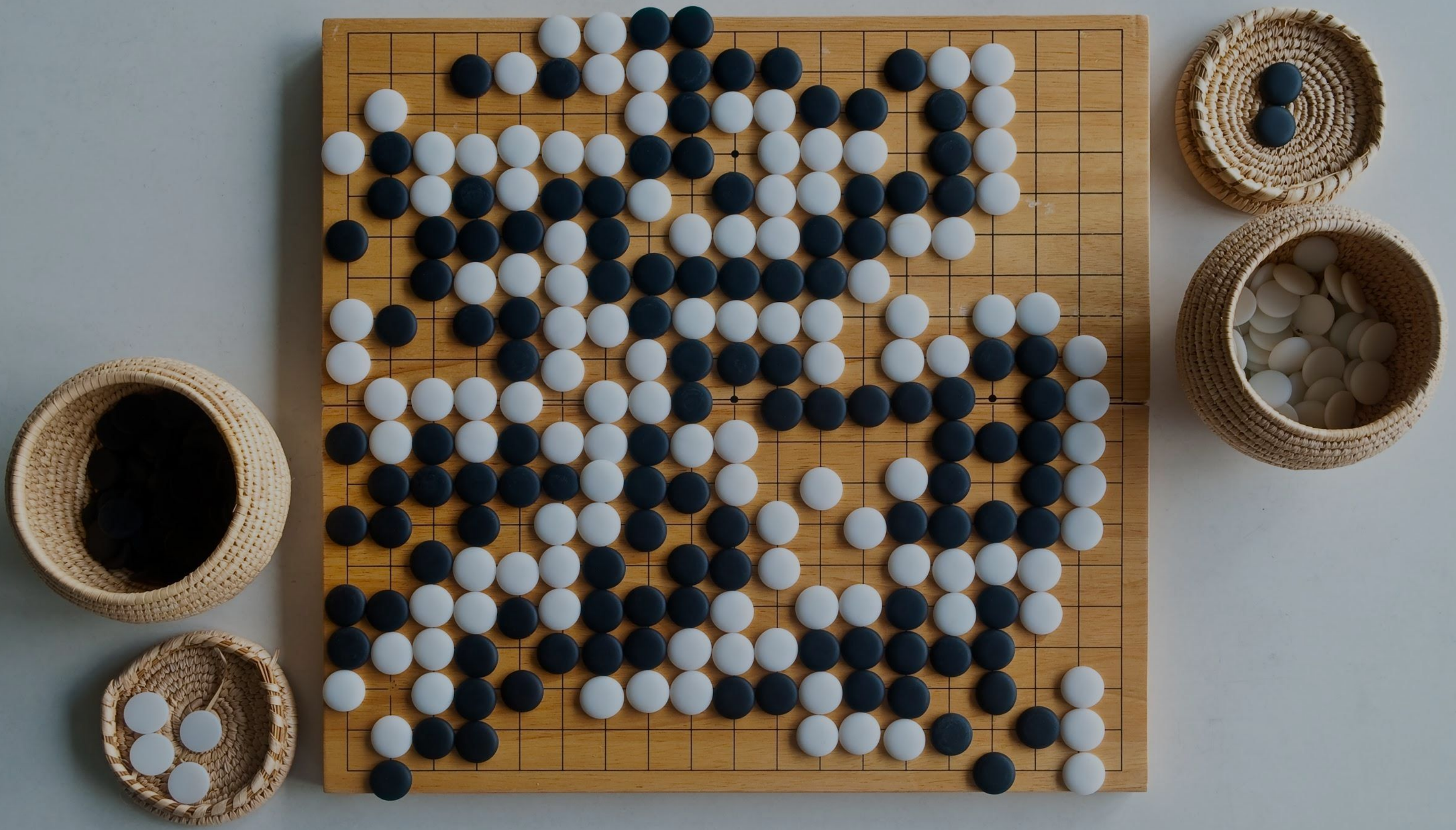
Reduces search depth

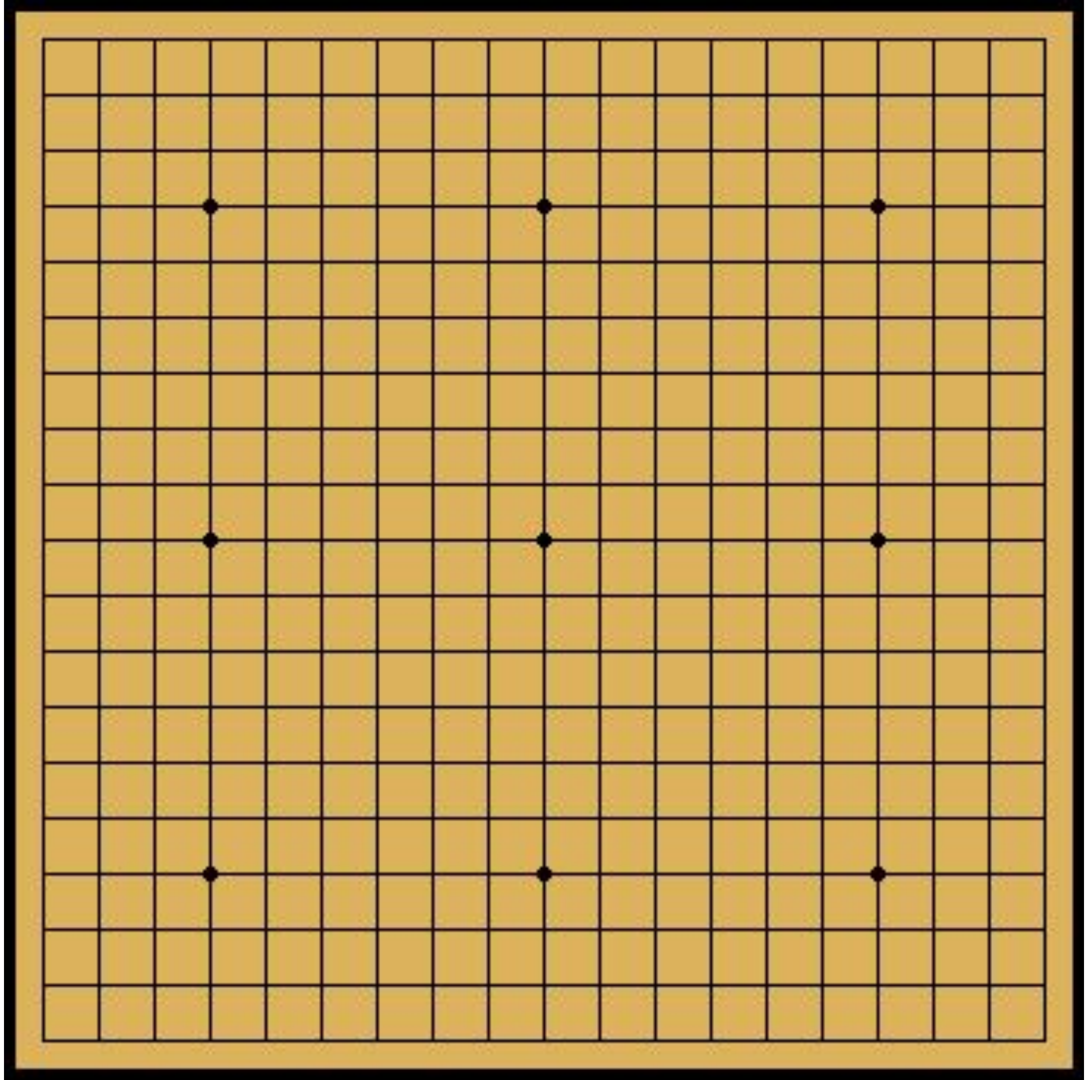


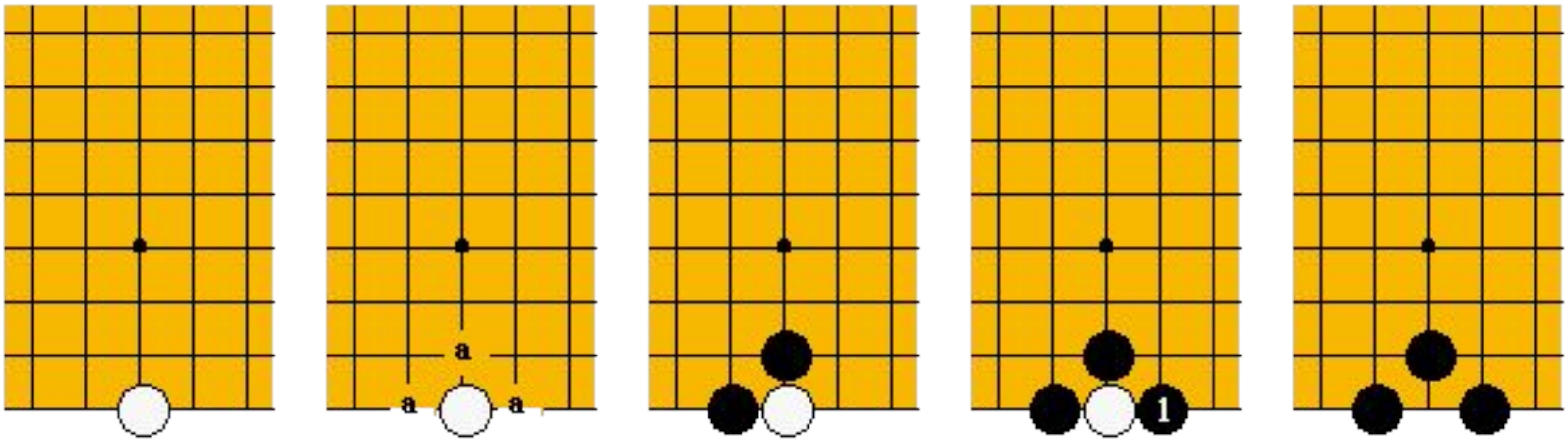
# Deep Blue

- 126 million positions per second
- Hand-designed Heuristics

# The Game of Go

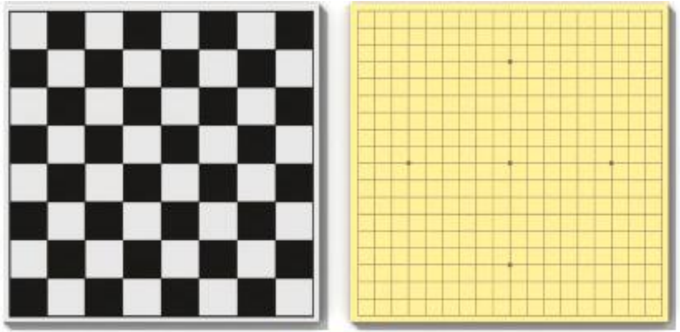





a
a
a
1

Go is Incredibly Complex

Go is Hard for Computers

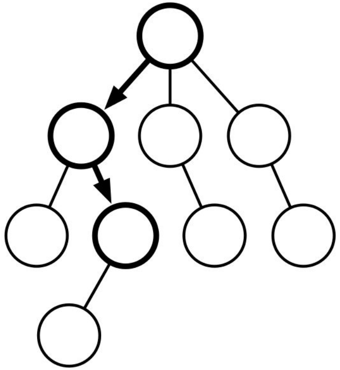


| | Chess | Go |
|---|---|---|
| **GRID SIZE** | 8 x 8 | 19 x 19 |
| **AVERAGE NUMBER OF MOVE CHOICES PER TURN** | 35 | 200–300 |
| **LENGTH OF TYPICAL GAME** | 60 moves | 200 moves |
| **NUMBER OF POSSIBLE GAME POSITIONS** | $10^{44}$ | $10^{170}$ |

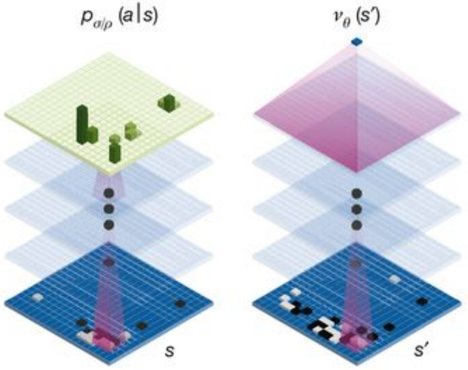
**EXPLOSION OF CHOICES**
(starting from average game position)

| | | |
|---|---|---|
| 35 | Move 1 | 200 |
| 1225 | Move 2 | 40 000 |
| 42 875 | Move 3 | 8 000 000 |
| 1 500 625 | Move 4 | 1 600 000 000 |

# How AlphaGo Zero Works



Monte-Carlo Tree Search

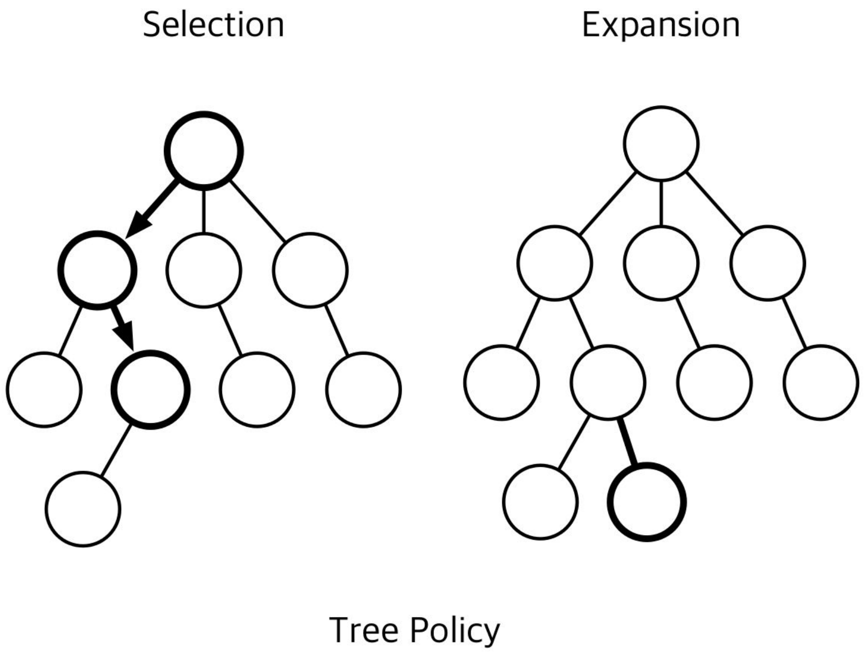



Residual Network

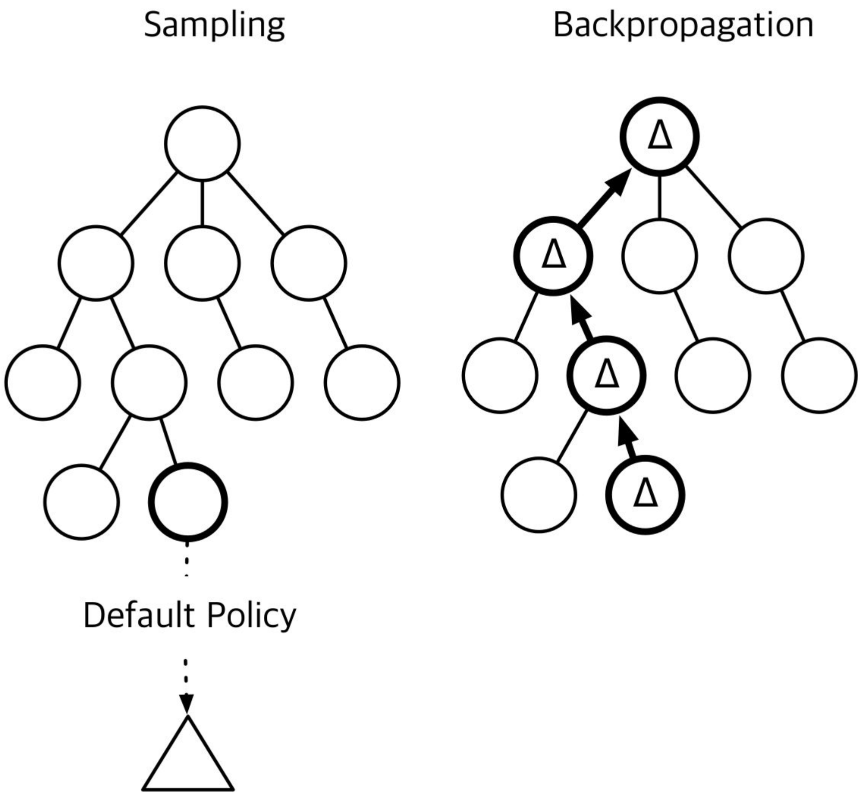
$$\pi^* \rightleftarrows V^*$$

Policy Iteration

# Monte-Carlo Tree Search (MCTS)



# Monte-Carlo Tree Search (MCTS)



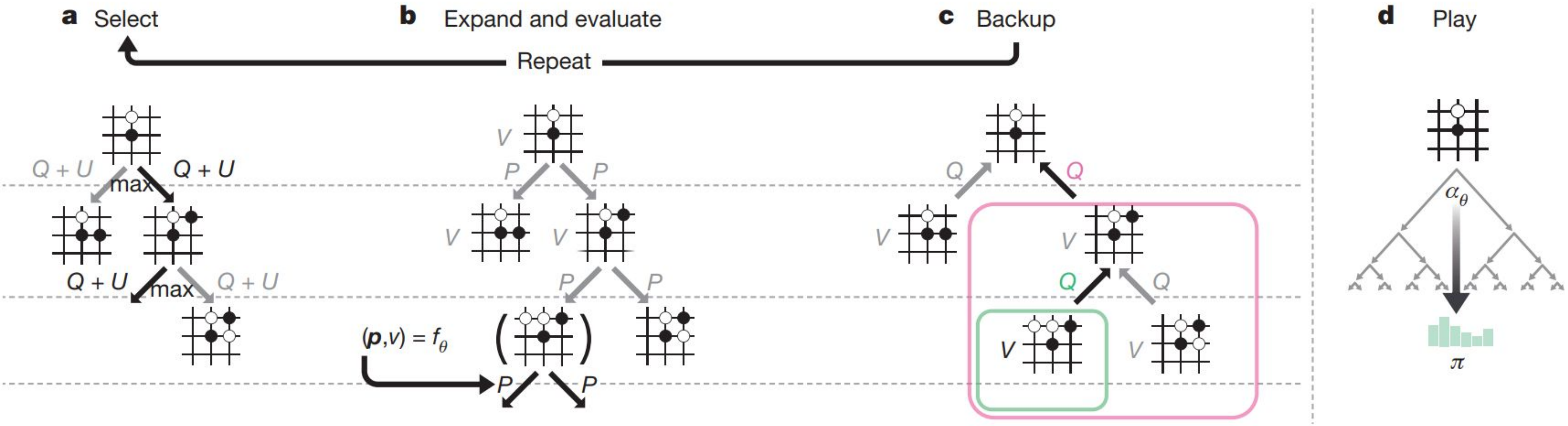
# MCTS: Advantages

- Aheuristic
- Online-search
- Works well on large trees

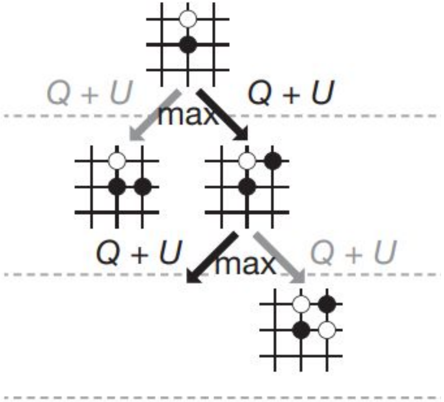
# MCTS: Disadvantages

- Many simulation are required
- No generalization between similar states
- Performance is dependent on “rollout” policy

# MCTS in AlphaGo Zero

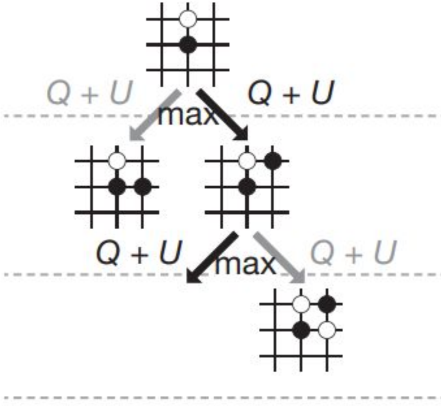


# Upper Confidence Bound for Trees (UCT)




$$Q(s,a) + c_{puct} \cdot P(s,a) \cdot \frac{\sqrt{\Sigma_b N(s,b)}}{1 + N(s,a)}$$

# Upper Confidence Bound for Trees (UCT)



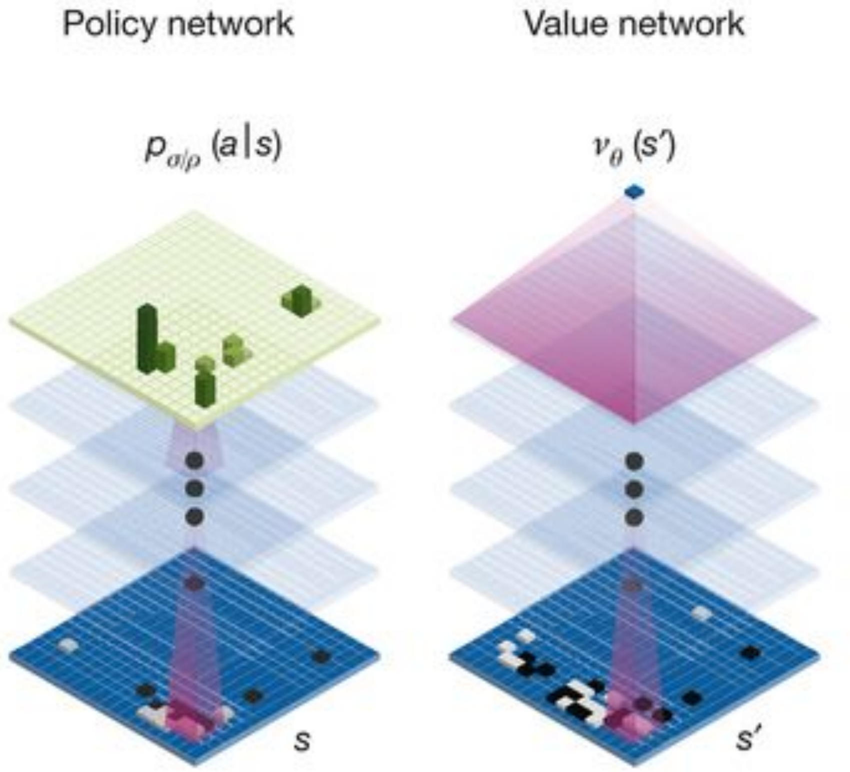

$$\underbrace{Q(s,a)}_{\text{Exploitation}} + \underbrace{c_{puct} \cdot P(s,a) \cdot \frac{\sqrt{\Sigma_b N(s,b)}}{1 + N(s,a)}}_{\text{Exploration}}$$

# Upper Confidence Bound for Trees (UCT)

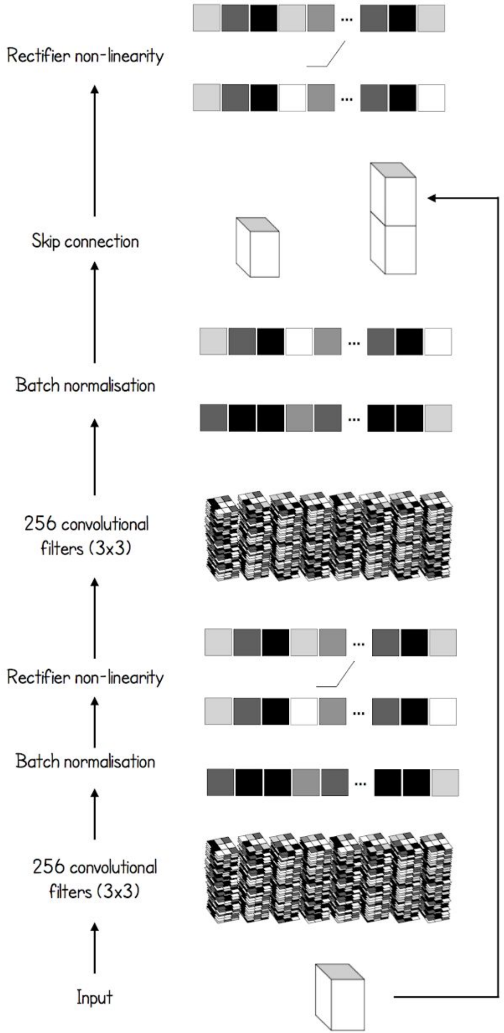
| | |
|---|---|
| s | State |
| a | Action |
| Q(s, a) | Expected Reward |
| P(s, a) | Policy |
| N(s, a) | # of state visits |
| $c_{puct}$ | Hyperparameter |

$$\underbrace{Q(s,a)}_{\text{Exploitation}} + \underbrace{c_{puct} \cdot P(s,a) \cdot \frac{\sqrt{\Sigma_b N(s,b)}}{1 + N(s,a)}}_{\text{Exploration}}$$

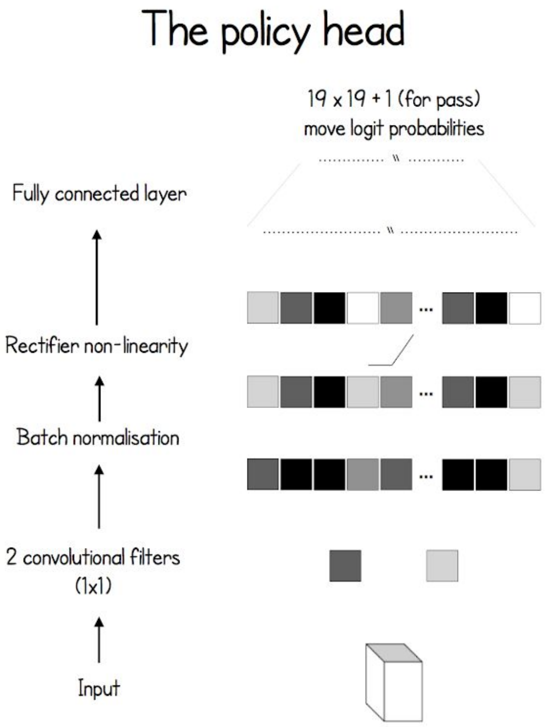
# AlphaGo Zero’s Network Architecture

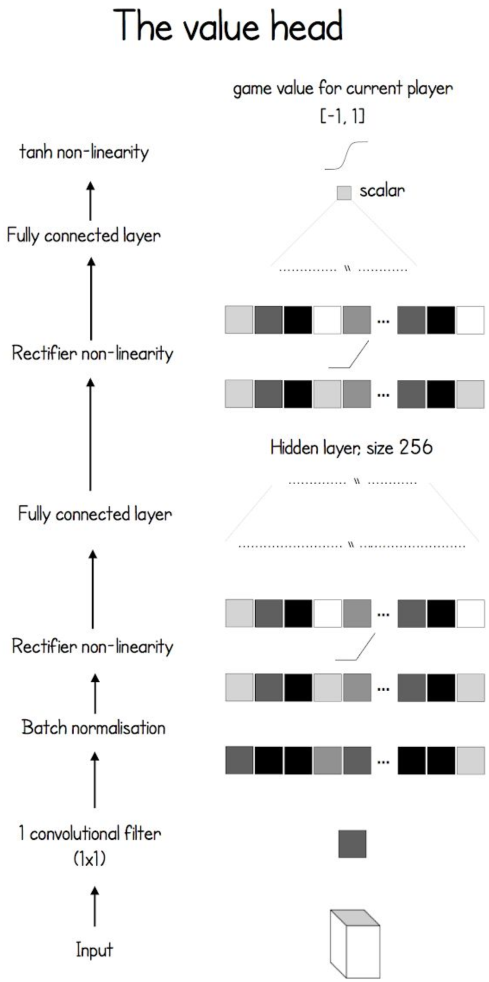


# Residual Layer



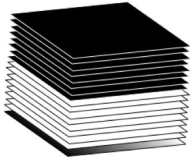
# Dual Heads


The policy head
19 x 19 + 1 (for pass)
move logit probabilities
Fully connected layer
Rectifier non-linearity
Batch normalisation
2 convolutional filters
(1x1)
Input


The value head
game value for current player
[-1, 1]
tanh non-linearity
scalar
Fully connected layer
Rectifier non-linearity
Hidden layer, size 256
Fully connected layer
Rectifier non-linearity
Batch normalisation
1 convolutional filter
(1x1)
Input

# Training

Self-play Worker



$\pi$

Training Worker

$$l = (z - v)^2 - \boldsymbol{\pi}^{\mathrm{T}} \log \boldsymbol{p} + c\|\theta\|^2$$

Evaluator

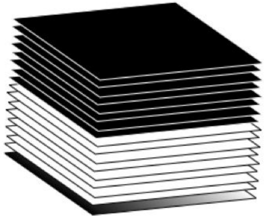
$$\pi' > \pi$$

# How AlphaGo Zero Chooses a Move

1600
Simulations

$$\pi \sim N^{1/\tau}$$

# Self-Play Workers



The game state

$\pi$

The search probabilities

The winner

# Training Worker

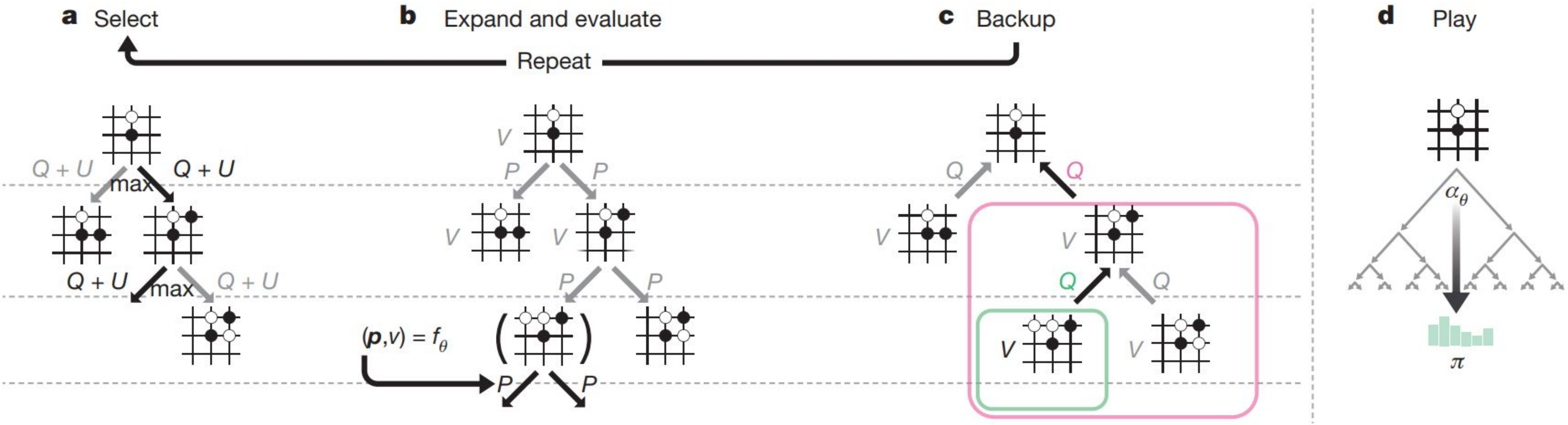
$$l = (z - v)^2 - \boldsymbol{\pi}^{\mathrm{T}} \log \boldsymbol{p} + c\|\theta\|^2$$

# Evaluator

400 Games

55% Win Rate

# MCTS in AlphaGo Zero

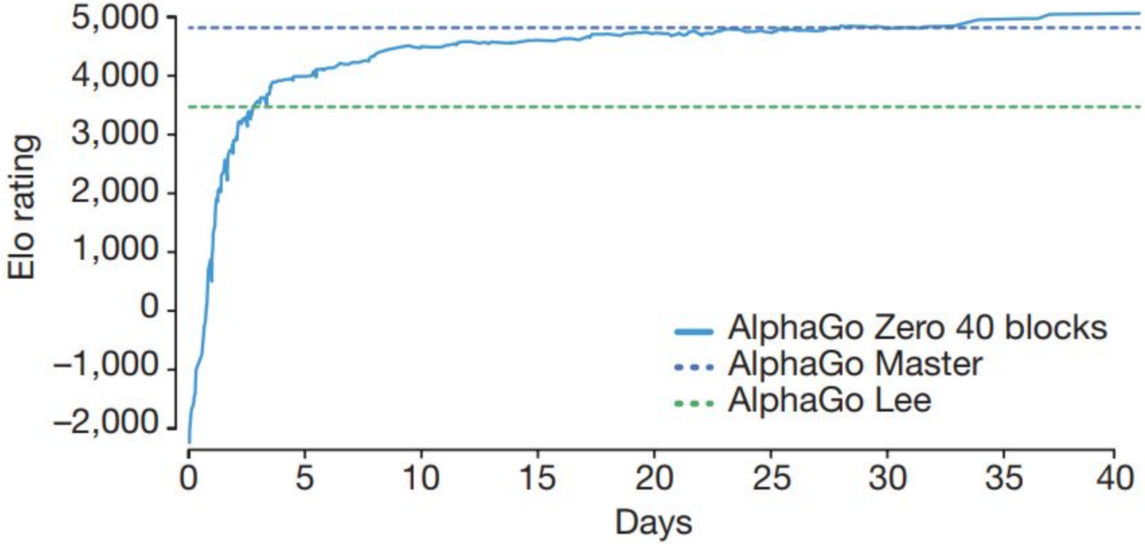


# 5 Minute Break

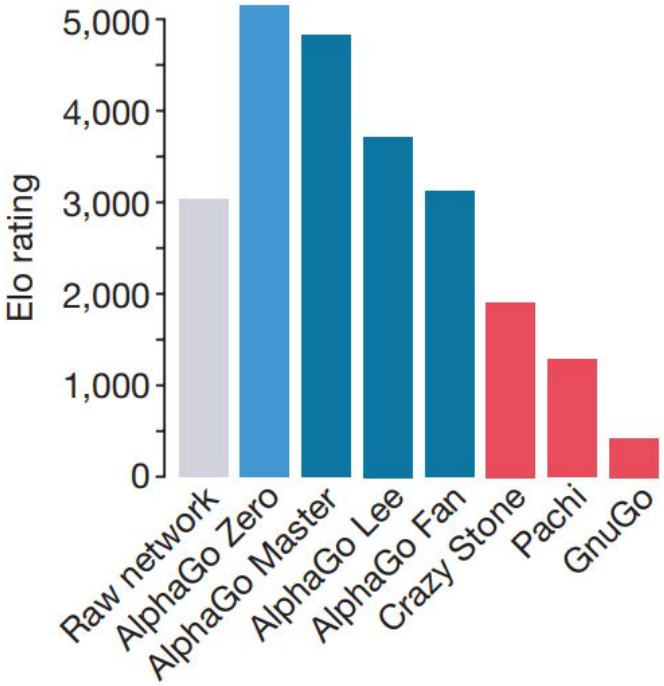
# AlphaGo Zero vs AlphaGo

| AlphaGo Zero | AlphaGo |
| --- | --- |
| Entirely self-play | Supervised learning + self-play |
| Input is game board | Input is hand-crafted features |
| Single network | Two networks |
| No rollouts | Rollouts were used |

# Results

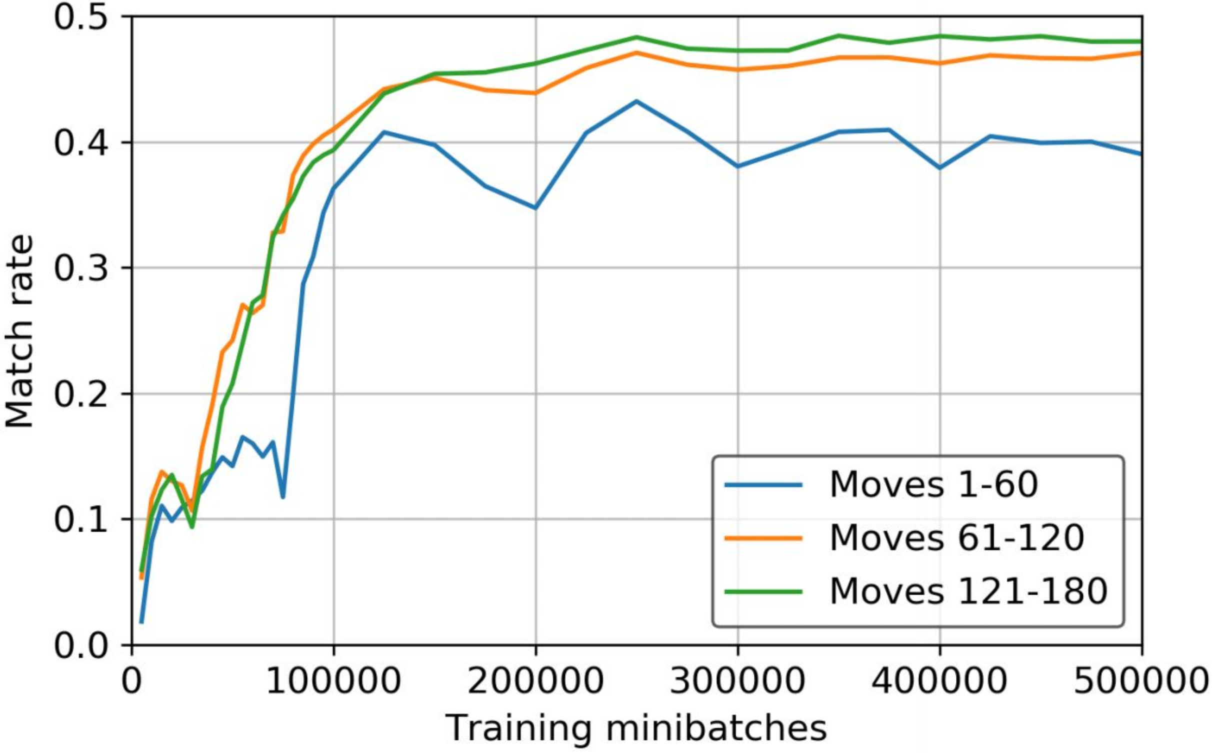


5,000
4,000
3,000
2,000
1,000
0
−1,000
−2,000
Elo rating
AlphaGo Zero 40 blocks
AlphaGo Master
AlphaGo Lee
0
5
10
15
20
25
30
35
40
Days

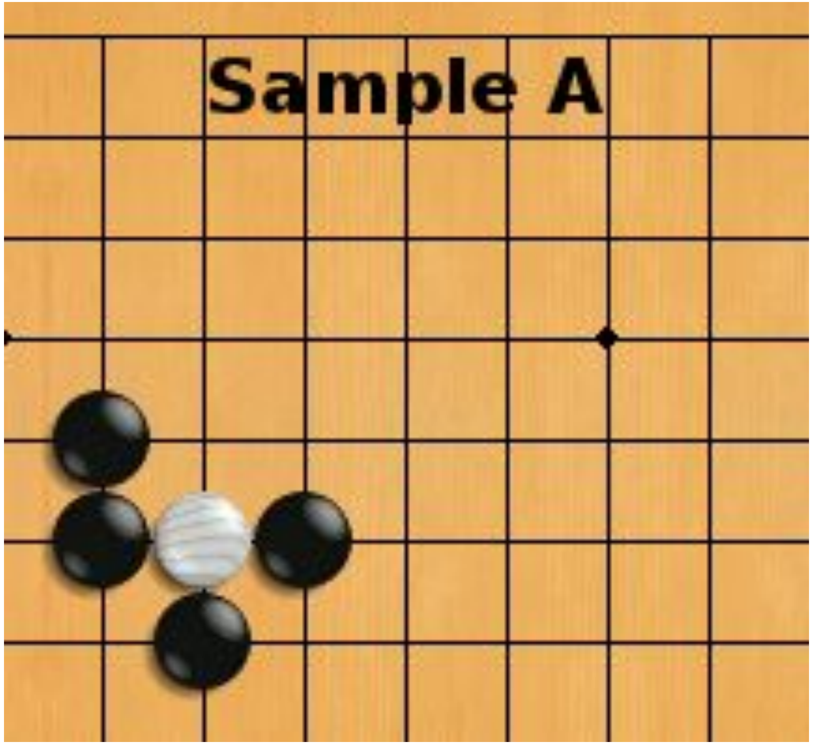



5,000
4,000
3,000
2,000
1,000
0
Elo rating
Raw network
AlphaGo Zero
AlphaGo Master
AlphaGo Lee
AlphaGo Fan
Crazy Stone
Pachi
GnuGo

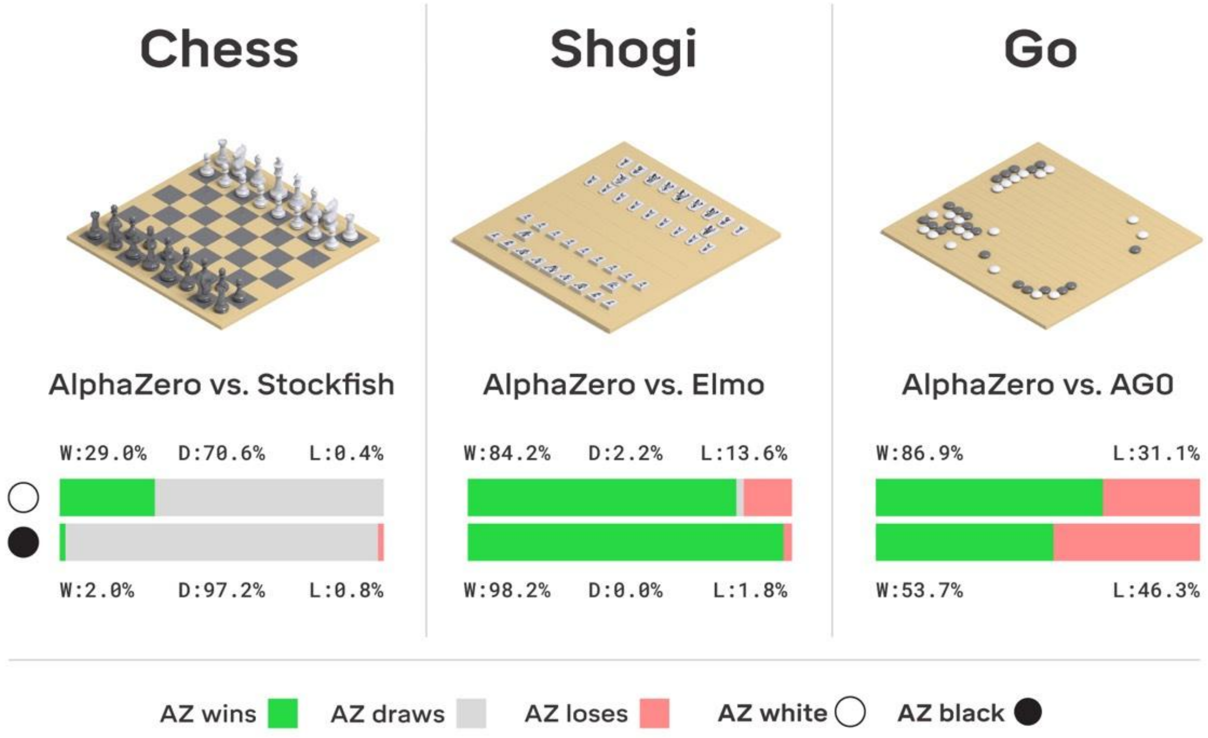
# Learning Stages



# Ladders



# AlphaZero



Chess
Shogi
Go
AlphaZero vs. Stockfish
AlphaZero vs. Elmo
AlphaZero vs. AGO
W:29.0% D:70.6% L:0.4%
W:2.0% D:97.2% L:0.8%
W:84.2% D:2.2% L:13.6%
W:98.2% D:0.0% L:1.8%
W:86.9% L:31.1%
W:53.7% L:46.3%
AZ wins
AZ draws
AZ loses
AZ white
AZ black

# AlphaGo Zero’s Gift

# Discussion

# Discussion

How can the AlphaGo Zero algorithm be extended to different games?

How can the sample efficiency of AlphaGo Zero be improved?

A very stable training environment is need for the algorithm.
Can this be alleviated to let AlphaZero applied to real-world problems?

# Resources

Mastering the Game of Go without Human Knowledge

David Silver 2017 NIPS Talk

ELF OpenGo: An Analysis and Open Reimplementation of AlphaZero

David Silver's PhD Thesis: Reinforcement Learning and Simulation-Based Search in Computer Go

A Brief History of Game AI Up To AlphaGo - Andrey Kurenkov

AlphaGo Zero Demystified - Dylan Djian