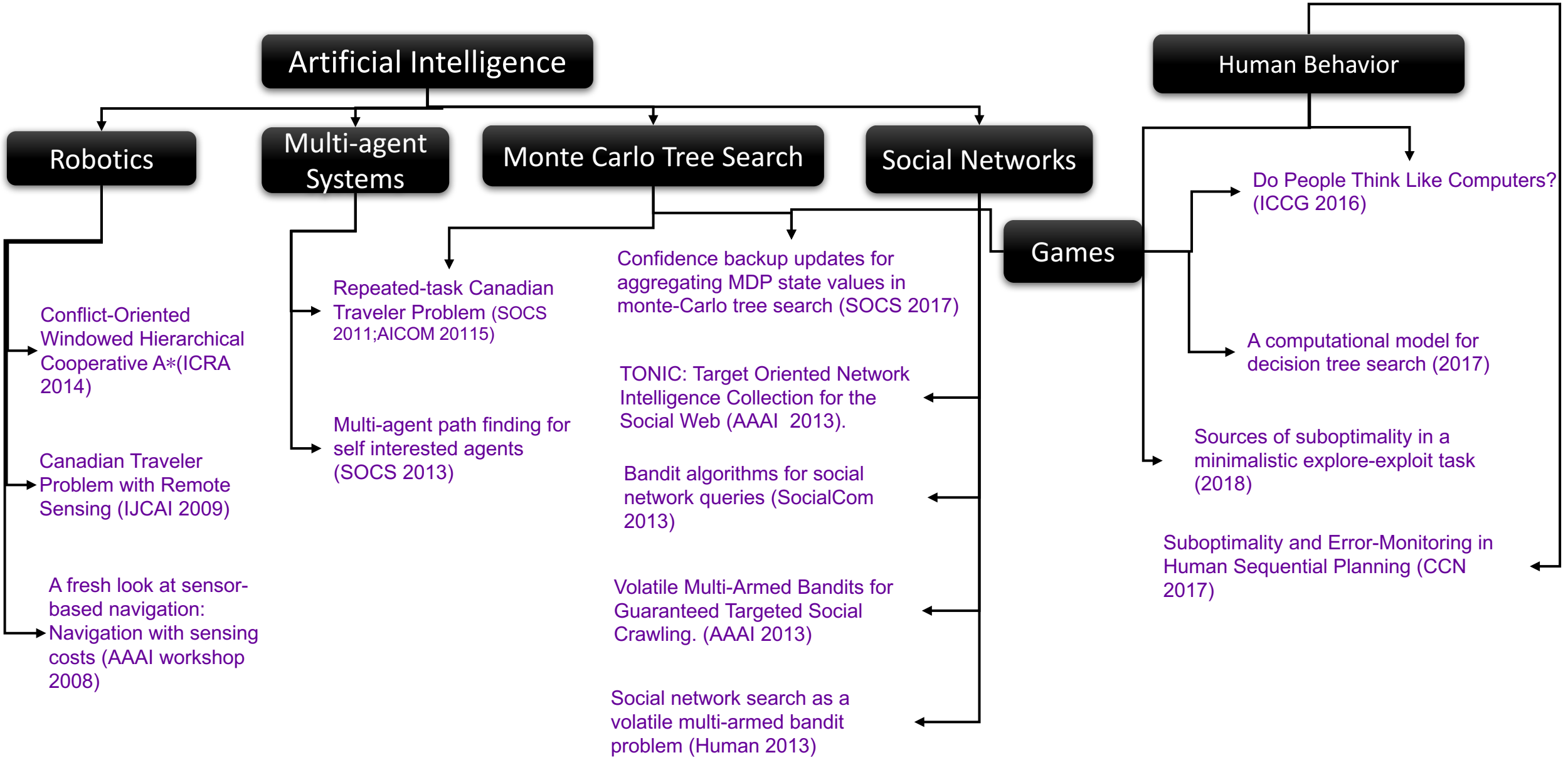
From heuristic search to models of sequential planning

Zahy Bnaya

zahy.bnaya@gmail.com


Artificial Intelligence
Robotics
Multi-agent Systems
Monte Carlo Tree Search
Social Networks
Games
Human Behavior
Conflict-Oriented Windowed Hierarchical Cooperative A*(ICRA 2014)
Canadian Traveler Problem with Remote Sensing (IJCAI 2009)
A fresh look at sensor-based navigation: Navigation with sensing costs (AAAI workshop 2008)
Repeated-task Canadian Traveler Problem (SOCS 2011;AICOM 20115)
Multi-agent path finding for self interested agents (SOCS 2013)
Confidence backup updates for aggregating MDP state values in monte-Carlo tree search (SOCS 2017)
TONIC: Target Oriented Network Intelligence Collection for the Social Web (AAAI 2013).
Bandit algorithms for social network queries (SocialCom 2013)
Volatile Multi-Armed Bandits for Guaranteed Targeted Social Crawling. (AAAI 2013)
Social network search as a volatile multi-armed bandit problem (Human 2013)
Do People Think Like Computers? (ICCG 2016)
A computational model for decision tree search (2017)
Sources of suboptimality in a minimalistic explore-exploit task (2018)
Suboptimality and Error-Monitoring in Human Sequential Planning (CCN 2017)

Come visit Tilburg

Pick a Date

Aug

Sep

Destination

Schiphol

Eindhoven

Other

Flight Carrier

Regular

Low Cost

Get to Tilburg

Car

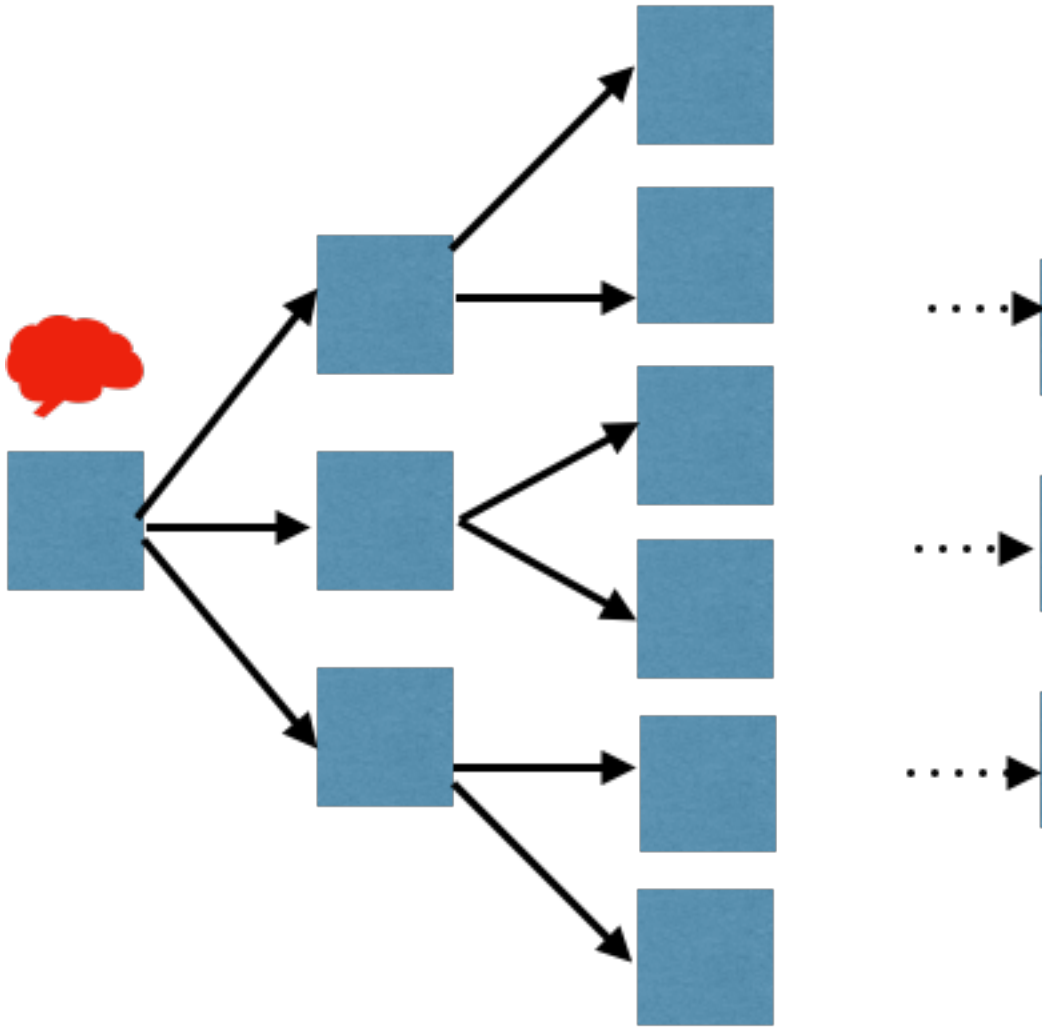
Train

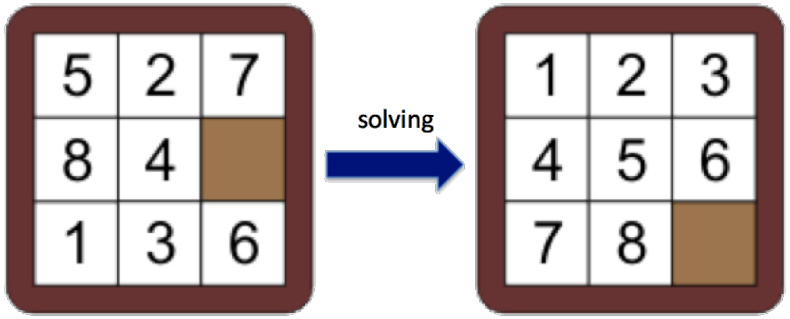
Bus

Departure Time

Location: tilburg
Arrival time: ?
Journey time: ?
Frustration level:?
Exhaustion level: ?
Total money spent:?

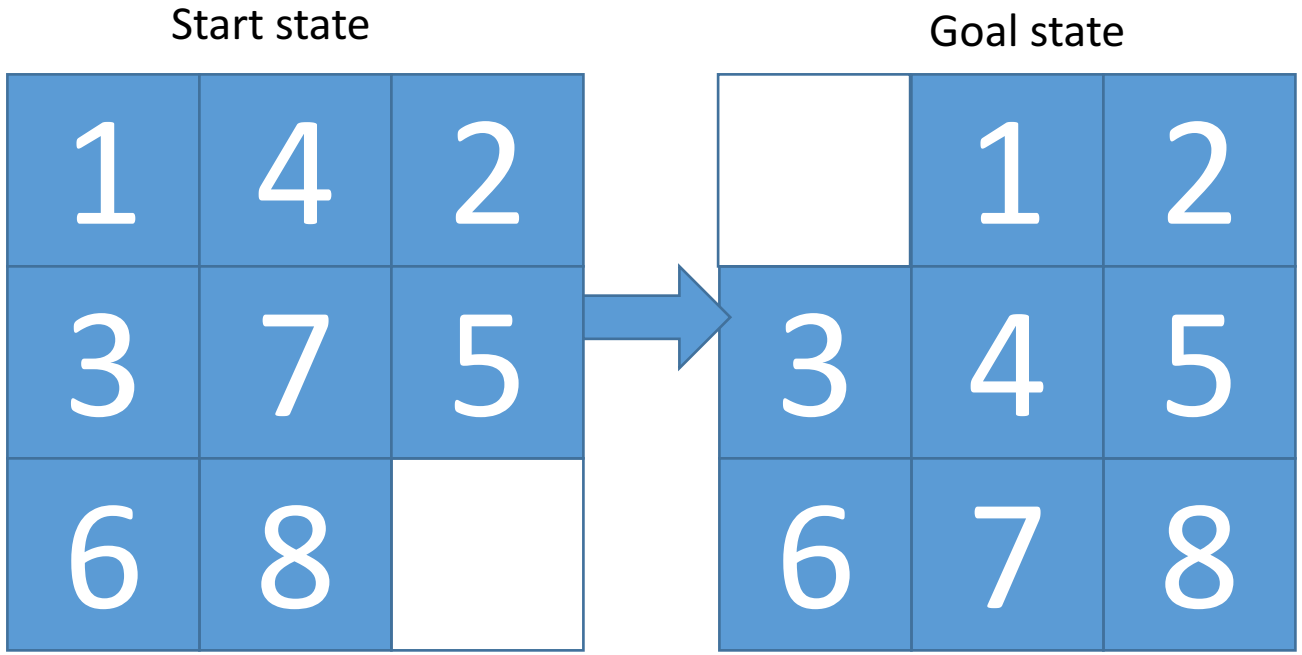
Sequential Decisions

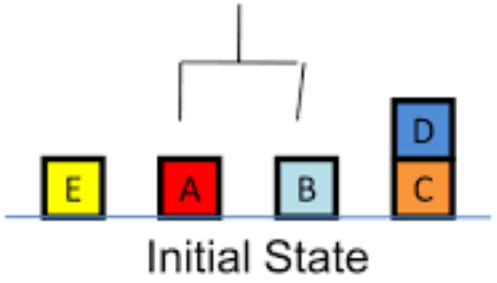




Classical Planning

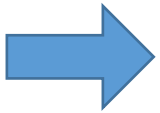
- *State Transition System*
  - *{States}*
  - {A*ctions*}
  - State-transition function
  - start state, goal state





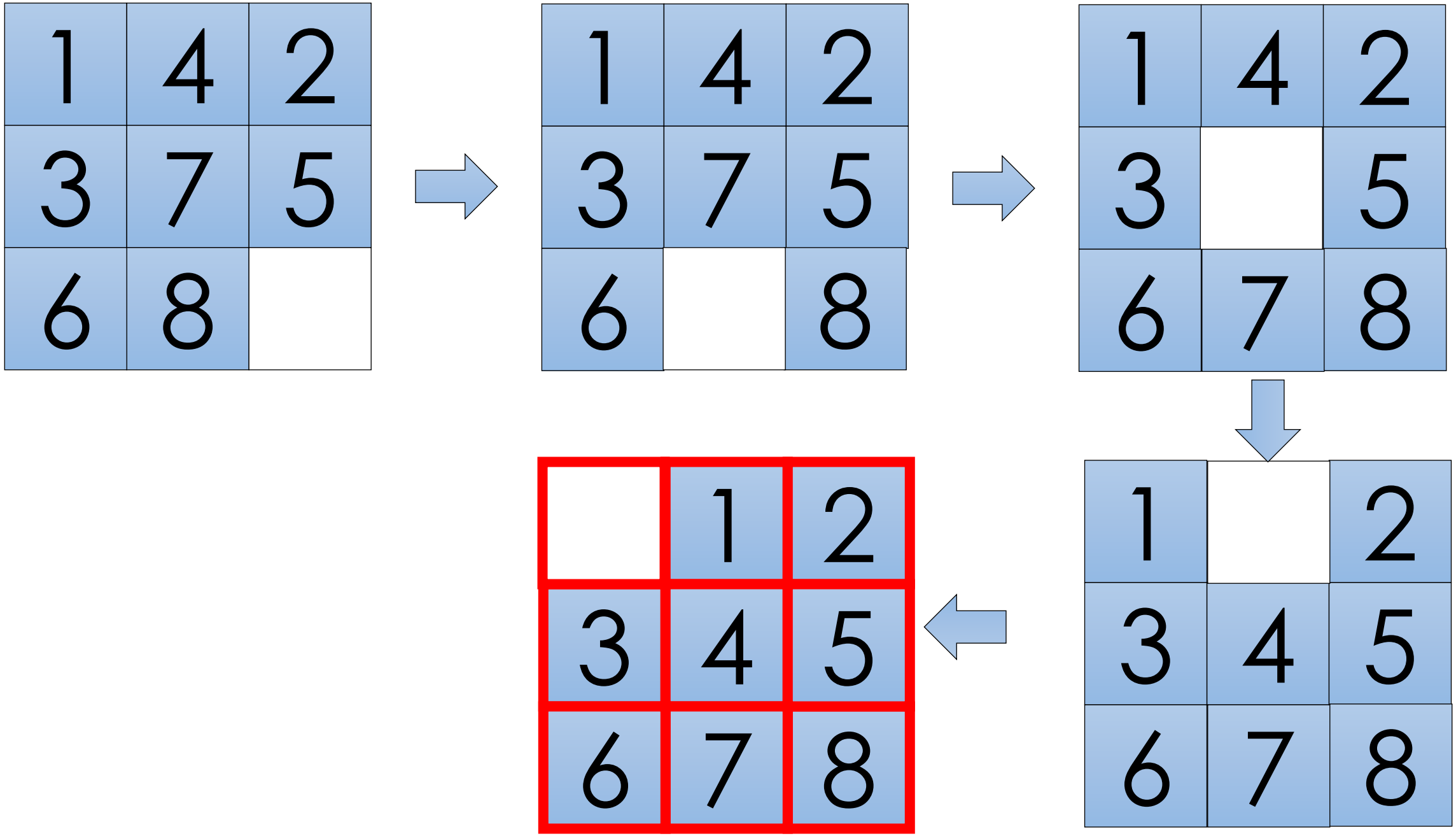
Start state

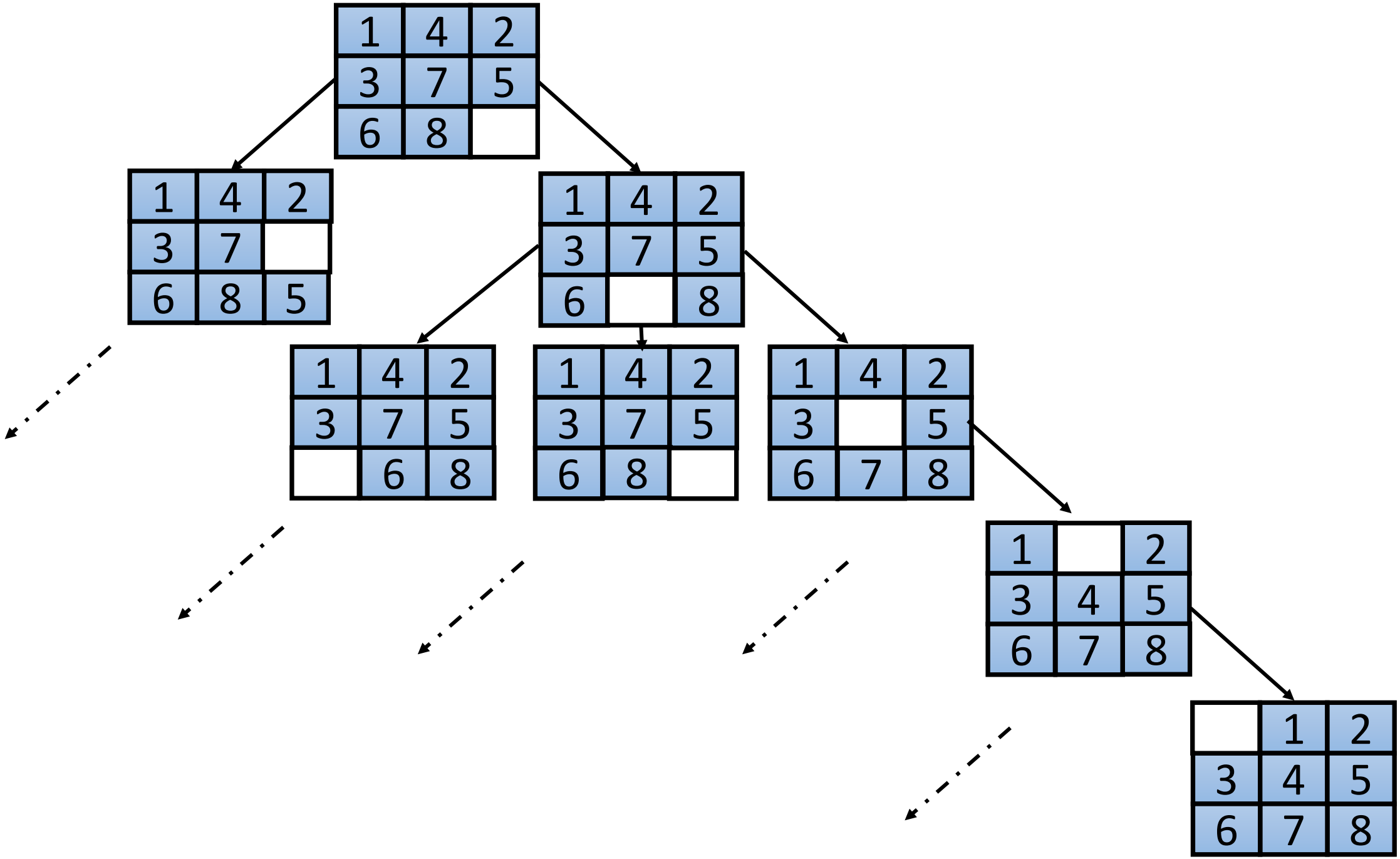
1	4	2
3	7	5
6	8	



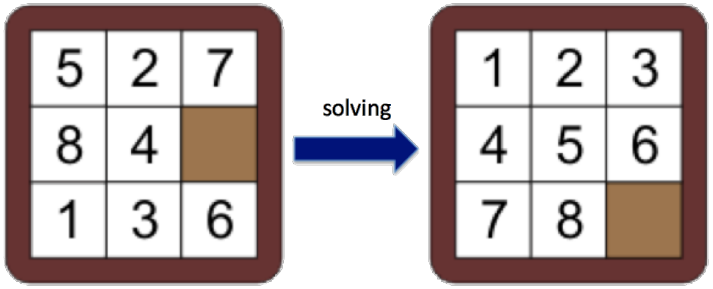
Goal state

	1	2
3	4	5
6	7	8

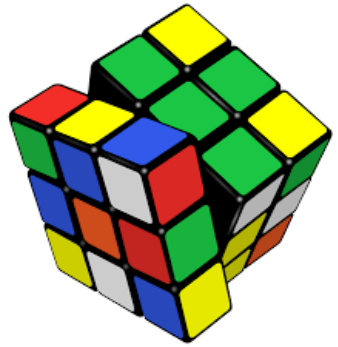

1 4 2
3 7 5
6 8
1 4 2
3 7 5
6 8
1 4 2
3 5
6 7 8
1 2
3 4 5
6 7 8
1 2
3 4 5
6 7 8


1 4 2
3 7 5
6 8
1 4 2
3 7
6 8 5
1 4 2
3 7 5
6 8
1 4 2
3 7 5
6 8
1 4 2
3 7 5
6 8
1 4 2
3 5
6 7 8
1 2
3 4 5
6 7 8
1 2
3 4 5
6 7 8

Sequential Decision Making

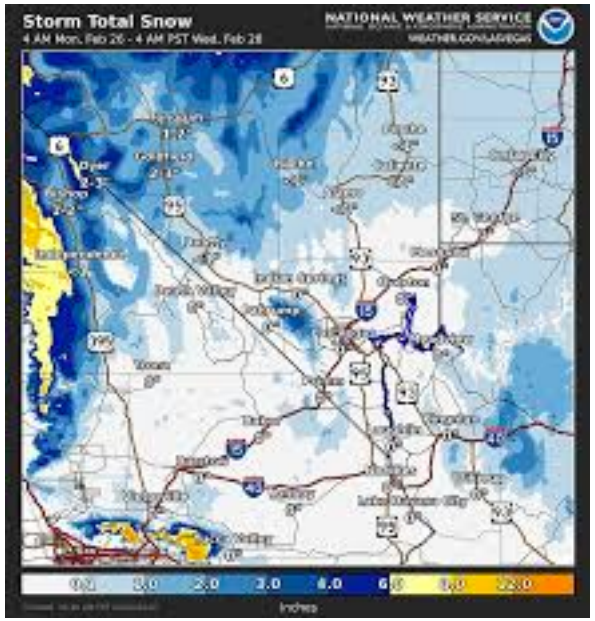


Tiles	#States
8	362,880
15	~10,461,395,000,000
24	$\sim 7.76 \times 10^{24}$

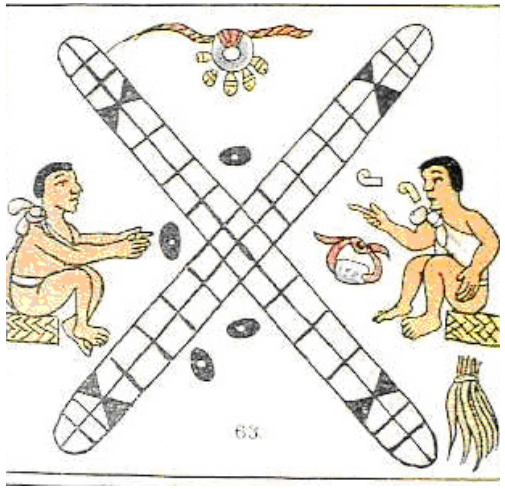


Cubies	#States
2x2x2	3,265,920
3x3x3	43,252,003,274,489,856,000
4x4x4	$\sim 7.4 \times 10^{45}$

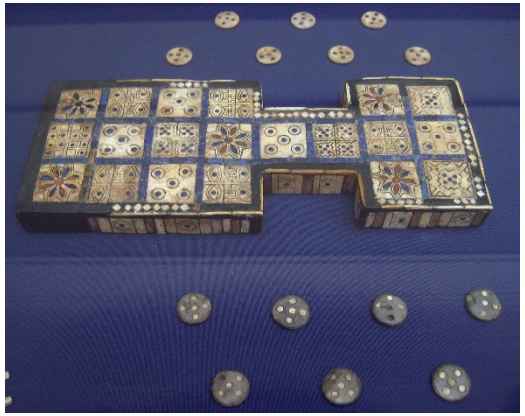
Even more complex Decision making domains



Games - complex decision making domains

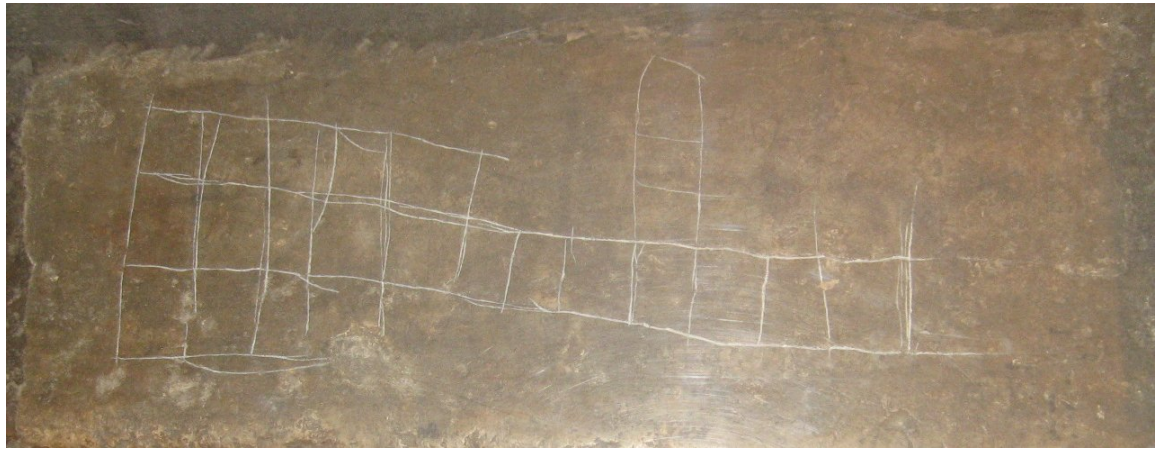


Patolli, played in Pre-Columbian America (Mexico)



The royal game of Ur (2600-2400 BC)

Sargon II (765-705 BC)



Found in the palace of Sargon II (now in the British Museum)

The rules (177 BC)
also in the British Muesum

Senet set from the tomb of Tutankhamun (~1323 BC)

Painting in tomb of Queen Nefertati (~1255 BC)

How did humans and animals evolved to deal with such tasks?

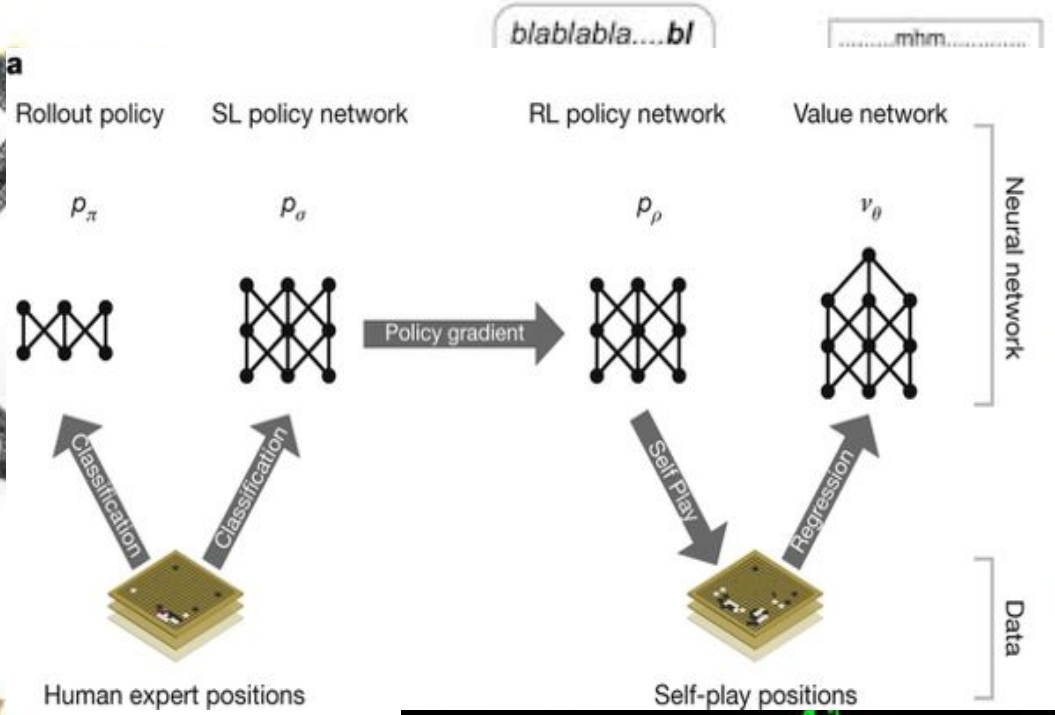
Chess research

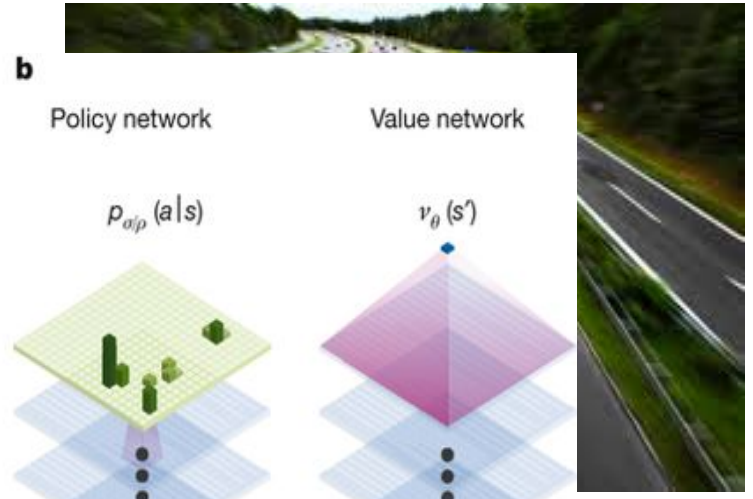
- Chase and Simon (1973) asked master chess players to verbally report what moves they were thinking of.
  - Subjective nature of verbal reports
  - No *quantitative* insights into evaluation function and selective search
- Adriaan de Groot (1946), Herb Simon (1989): the most important aspect of expertise is improved pattern recognition.
  - Memorize Chess Patterns (De Groot, Chase and Simon)

Newell and Simon: heuristic search hypothesis (1976)

- A physical symbol system will **repeatedly** generate and modify known symbol structures until the created structure matches the solution structure.
- A heuristic method can accomplish its task by using **search trees**.
- Instead of generating all possible solution branches, a **heuristic selects branches** more likely to produce outcomes than other branches.
- It is **selective** at each decision point, picking branches that are more likely to produce solutions.

Meanwhile in Artificial Intelligence



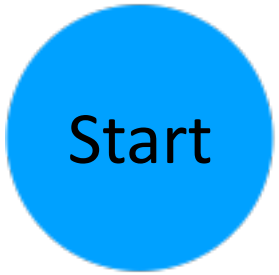


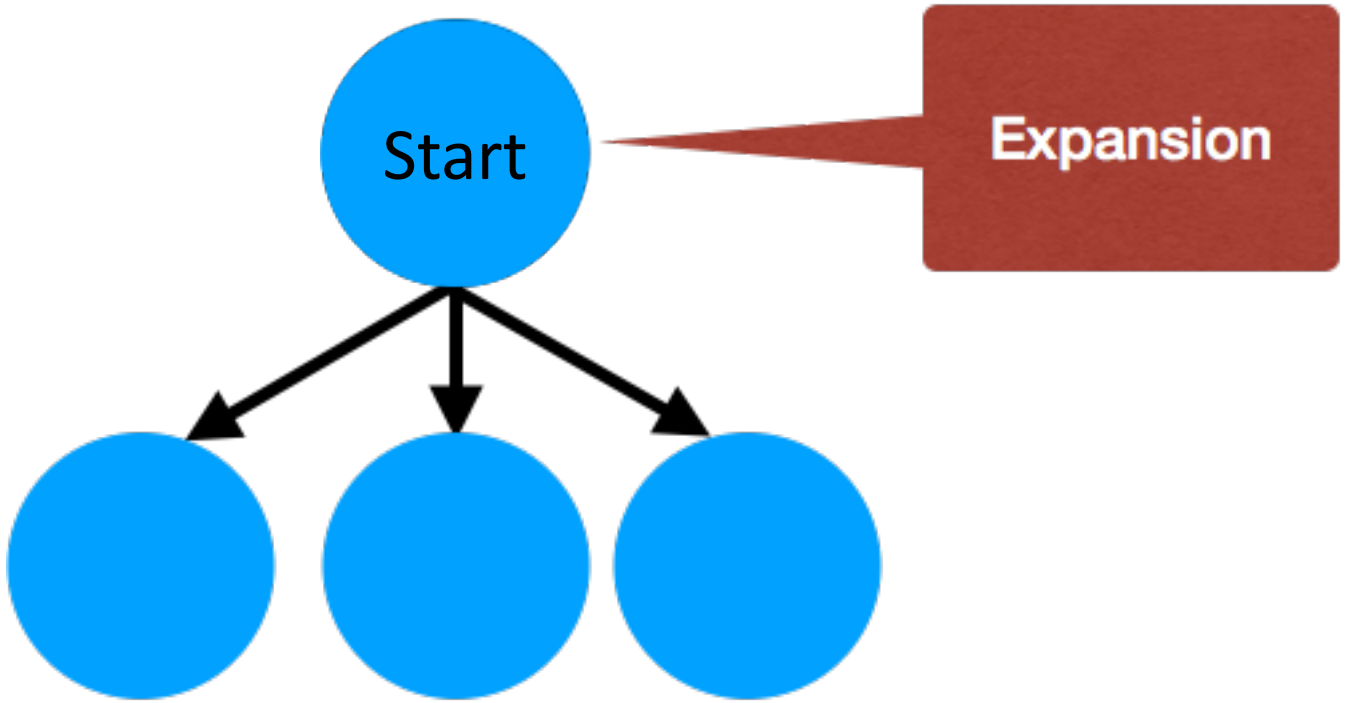
Impressive Results of AI solvers on Sequential Domains

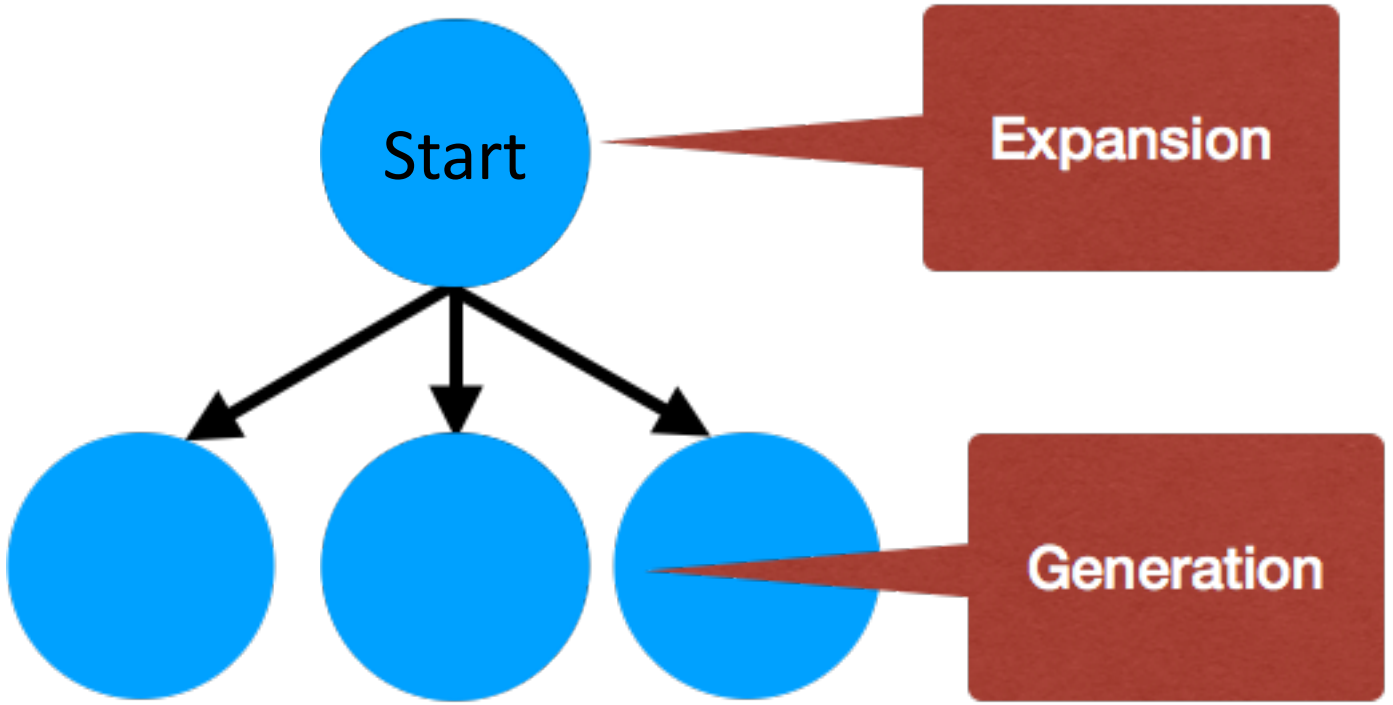
- Rubik's cube. - Any random state
- Sliding-Tile-Puzzle - Any random state of the 24-tile puzzle
- Checkers - Game is solved (Shafer et al. 2007)
- Heads-up limit hold'em poker - Game is solved (Bowling et al. 2015)

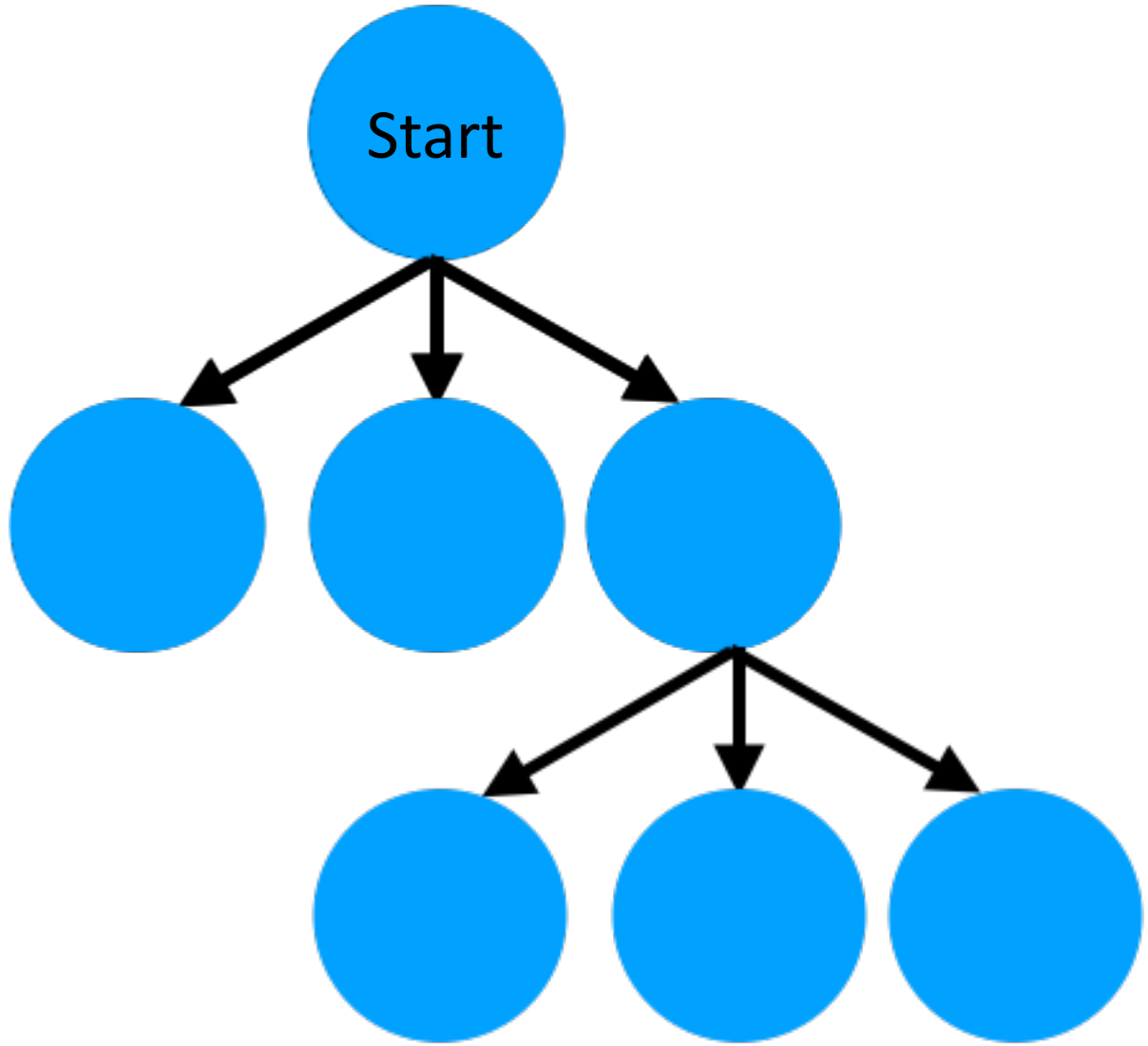
- Super-human strength even earlier:
  - Chess [Deep blue, 1996]
  - Checkers [Chinook, 1997]
  - Othello [Logistello, 1999]
  - Scrabble [Maven, 2002]
  - Go [AlphaGo, 2016]

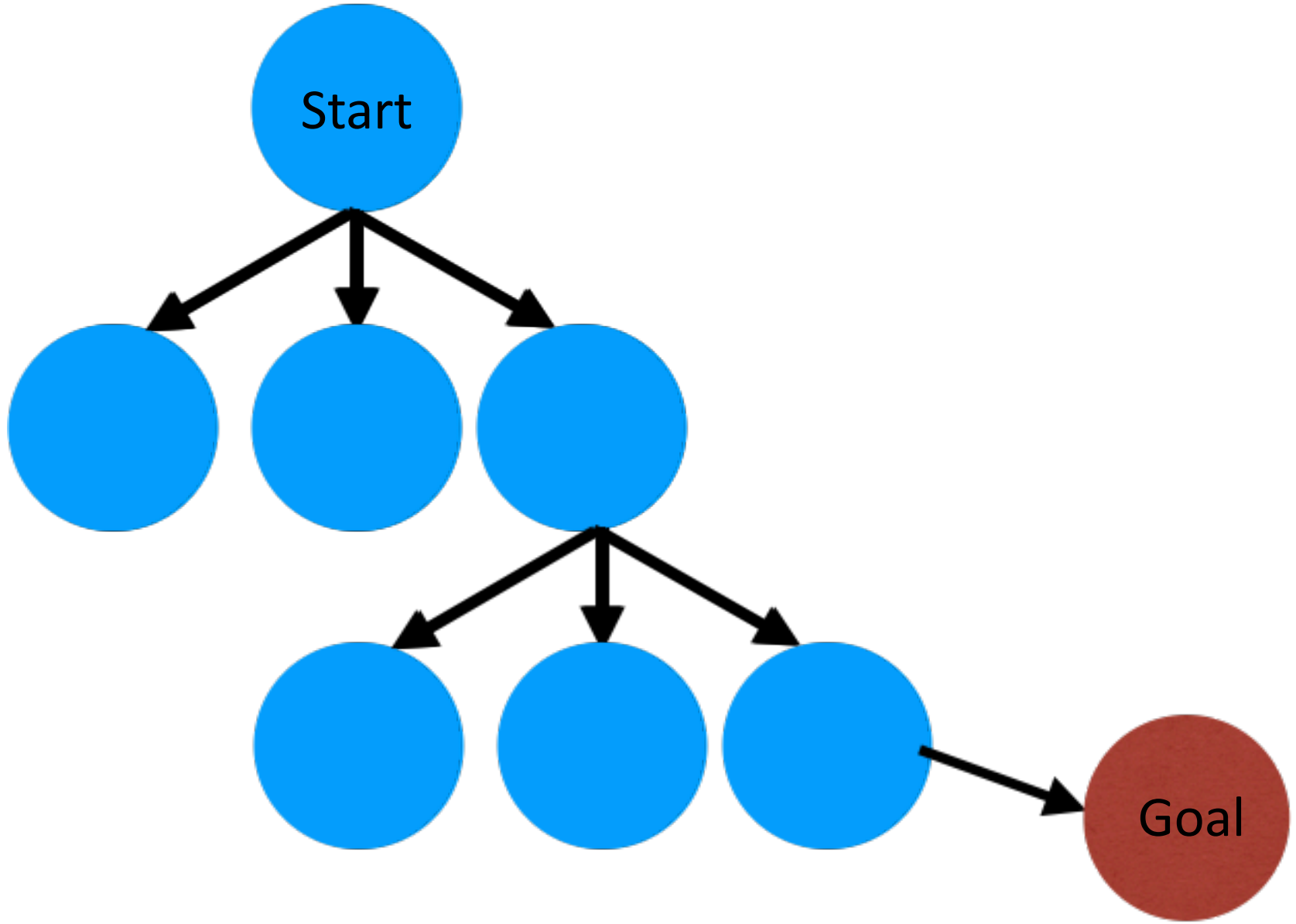
State-space search

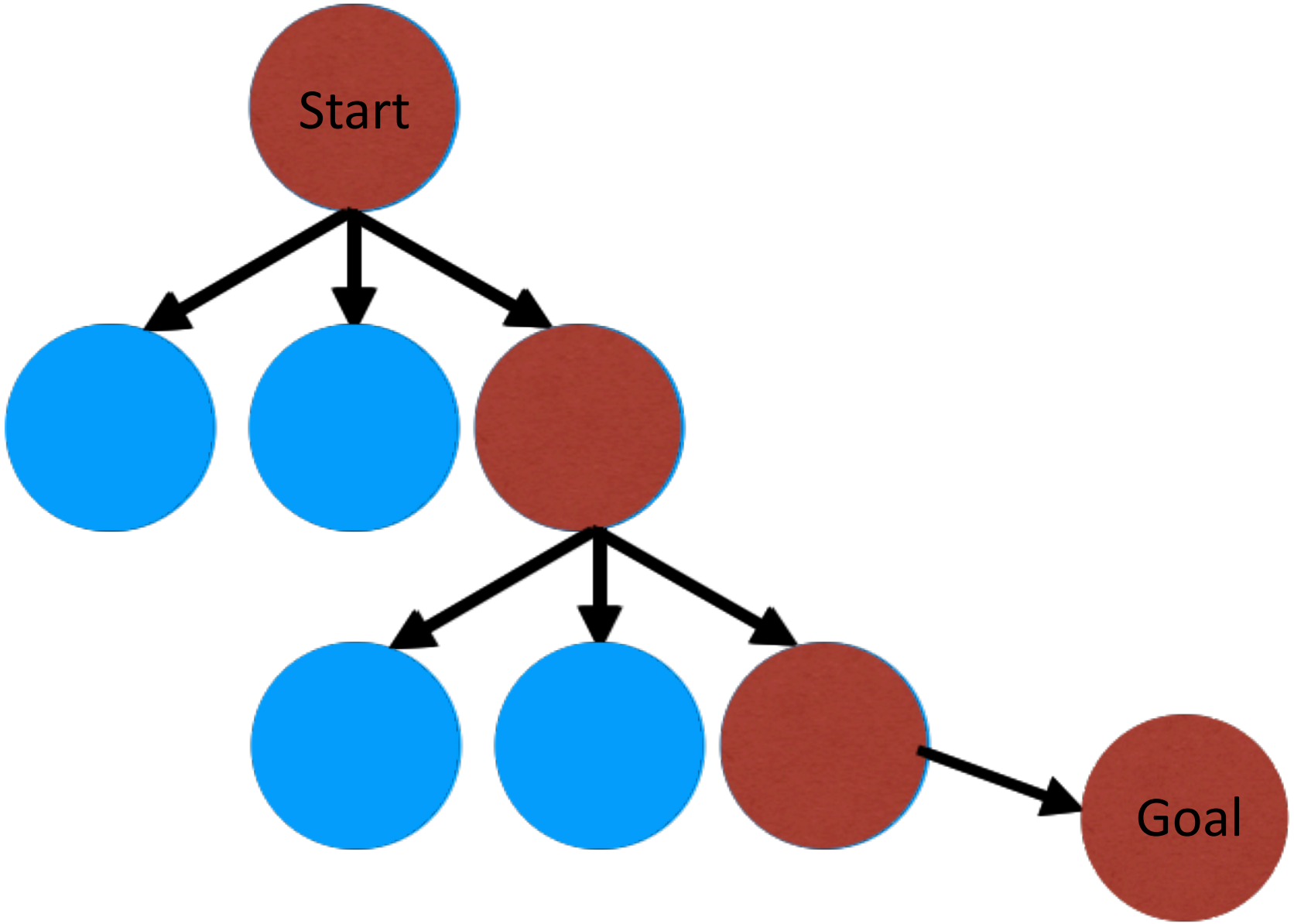



Start
Expansion


Start
Expansion
Generation

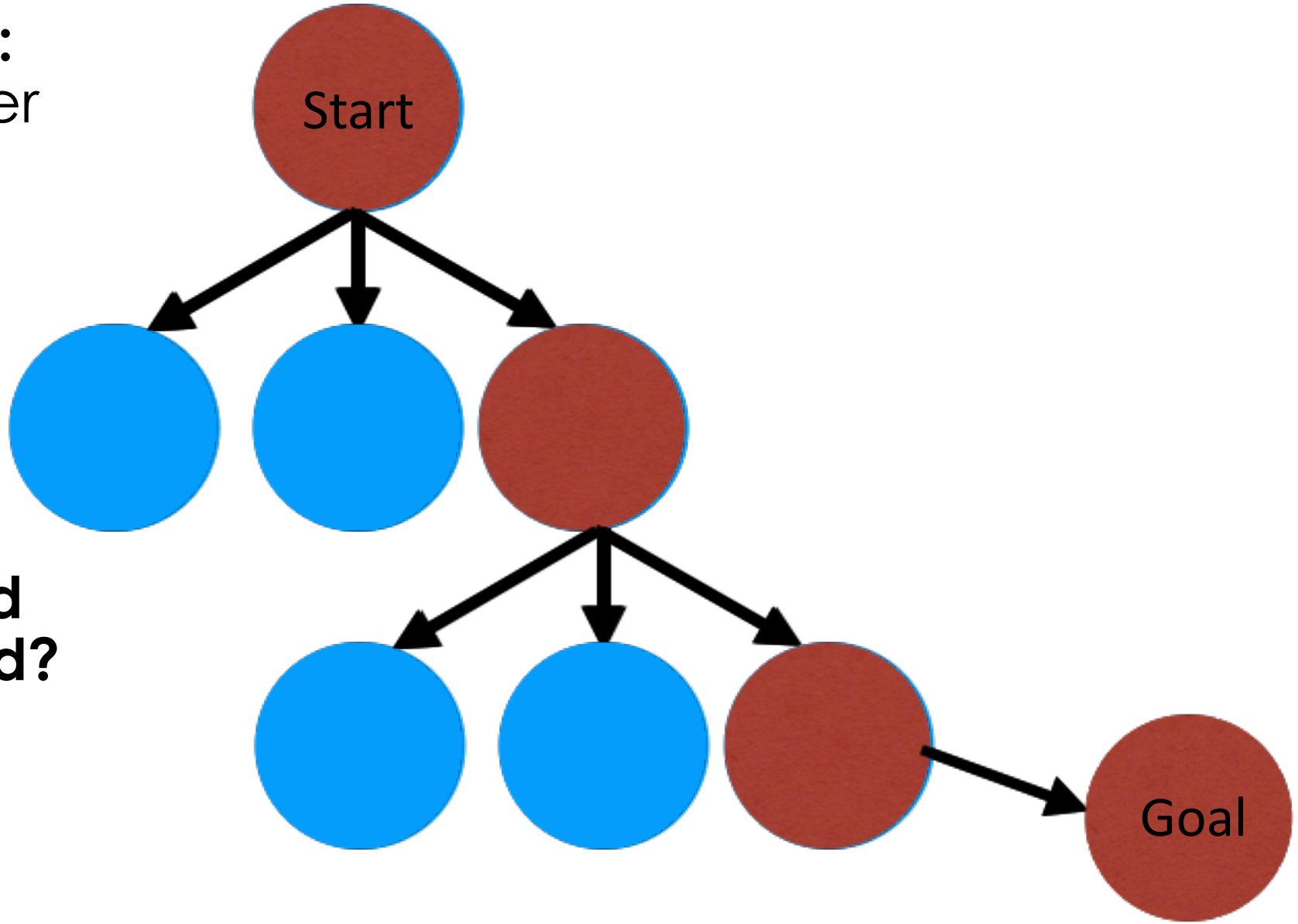

Start


Start
Goal


Start
Goal

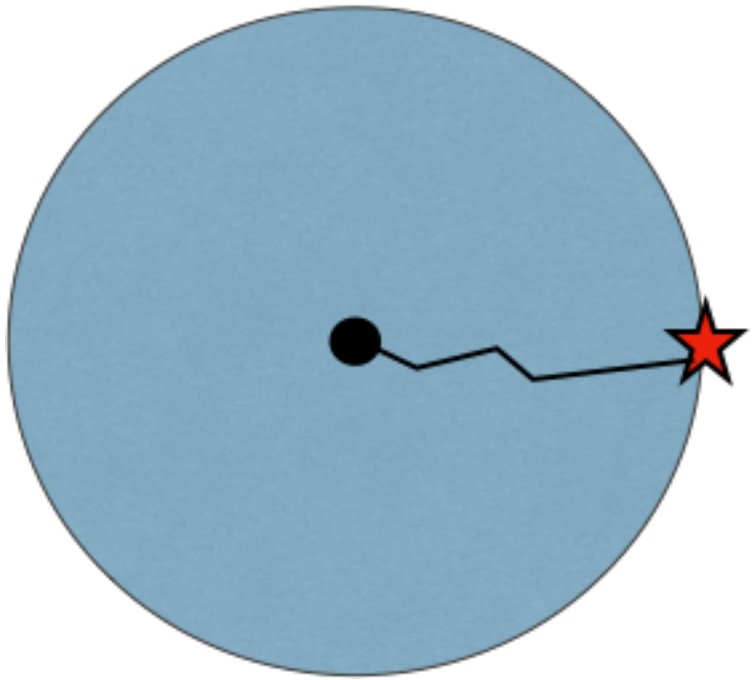
Common objective: minimize the number of **expanded** and **generated** nodes.

In what order should nodes be expanded?

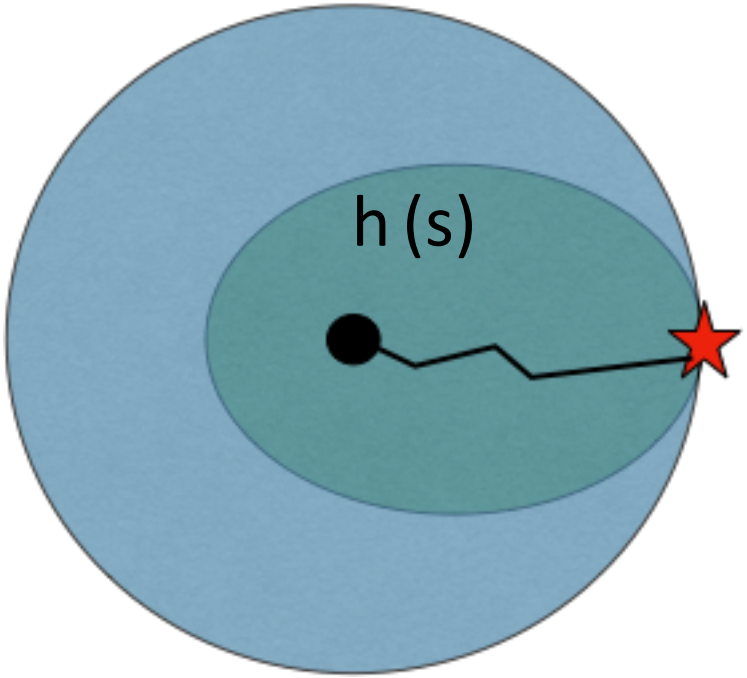


Uninformed Search

Uninformed Search

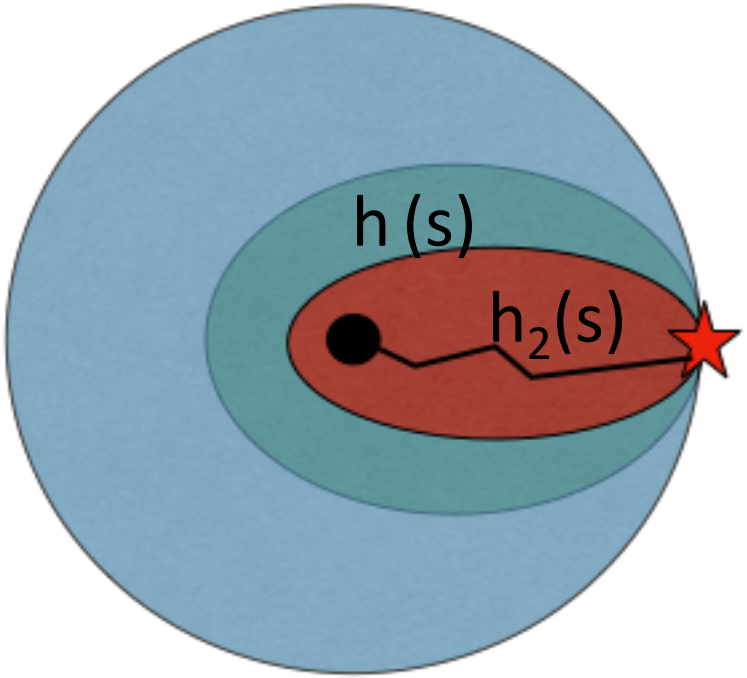


Heuristics - Informed Search



Heuristic function: Estimated distance from state *s* to the goal - h(s)

Heuristics - Informed Search

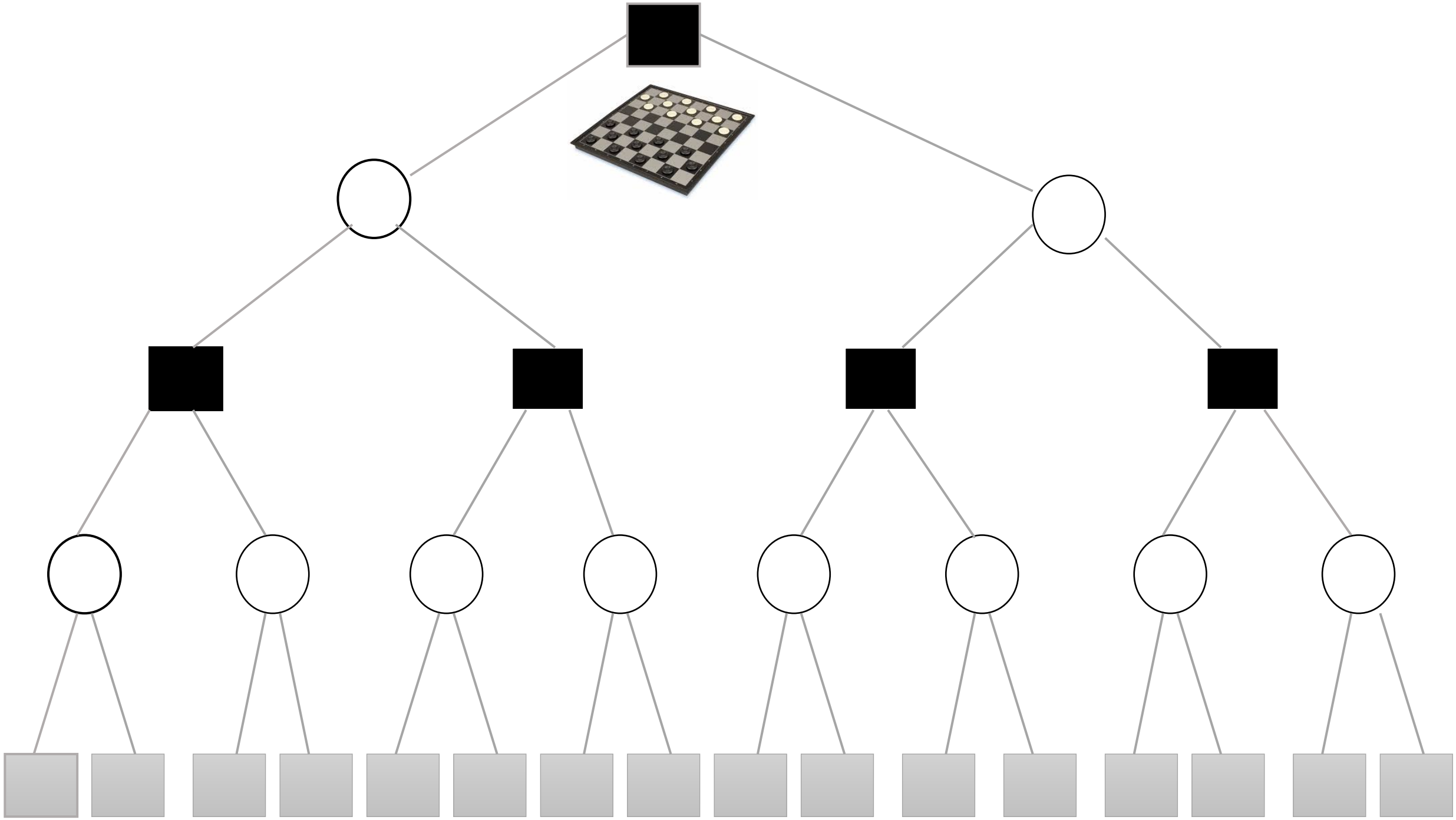


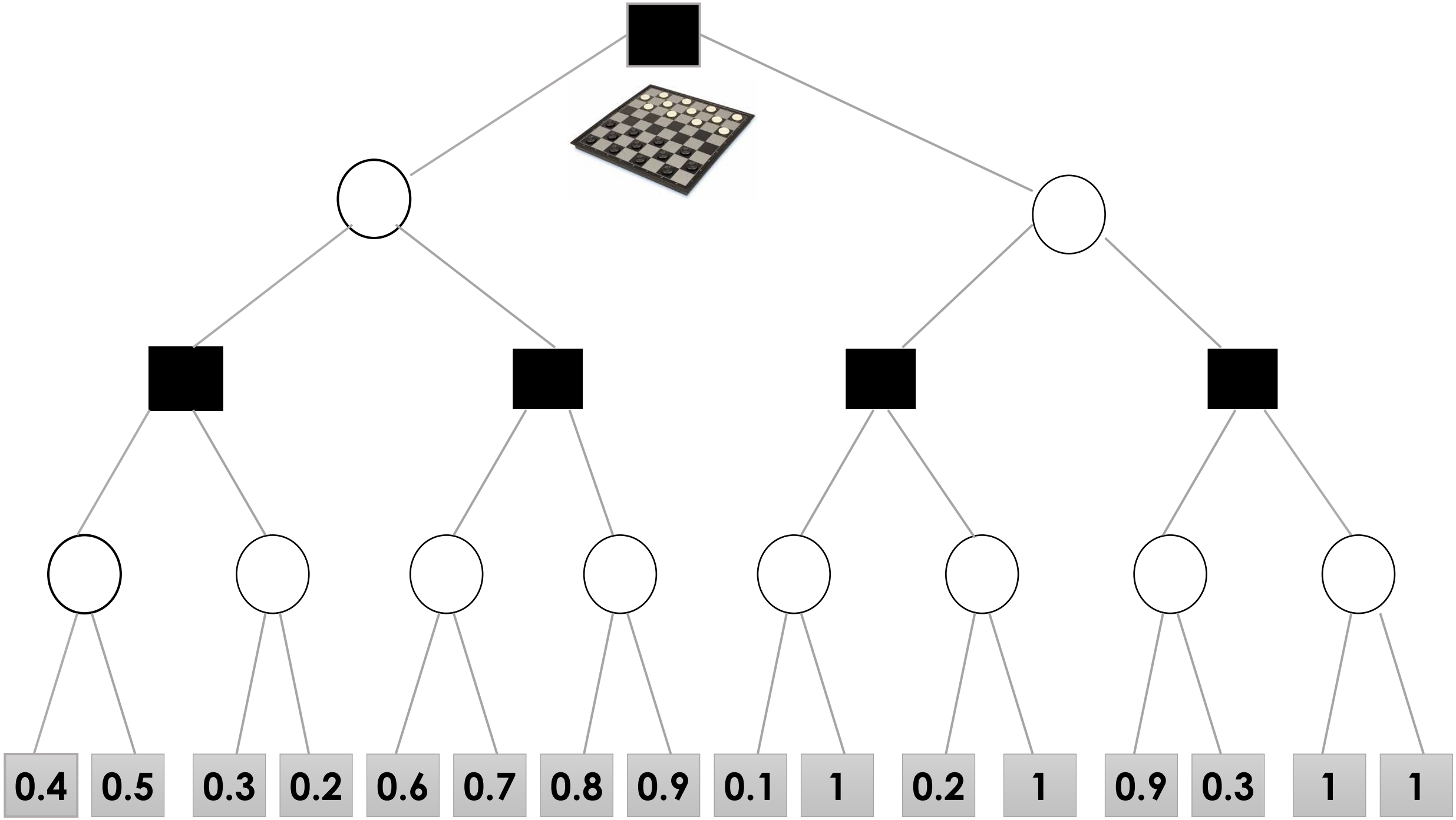
$h_2(s)$ a more accurate heuristic function

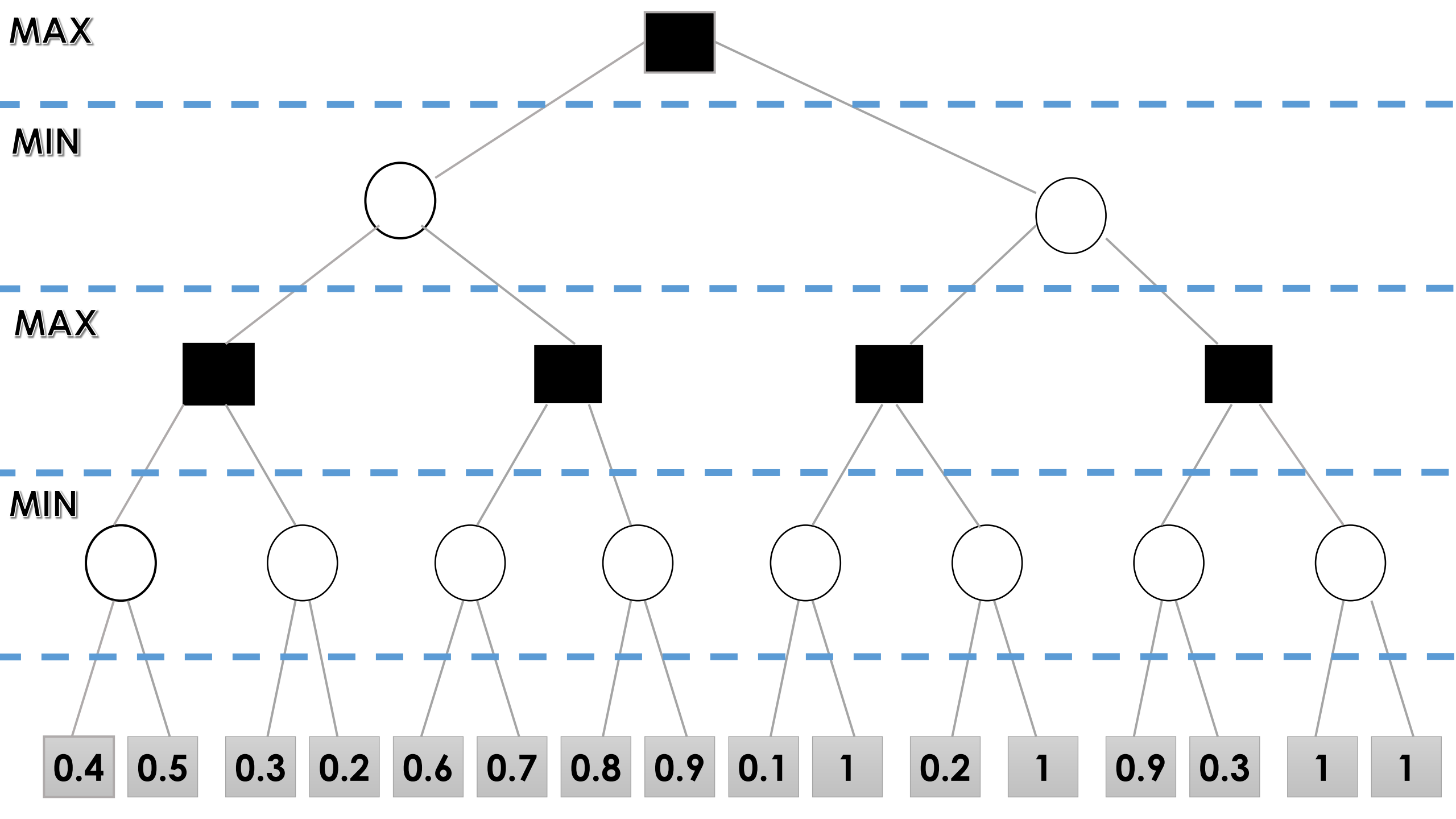
Two Player Zero-Sum Games

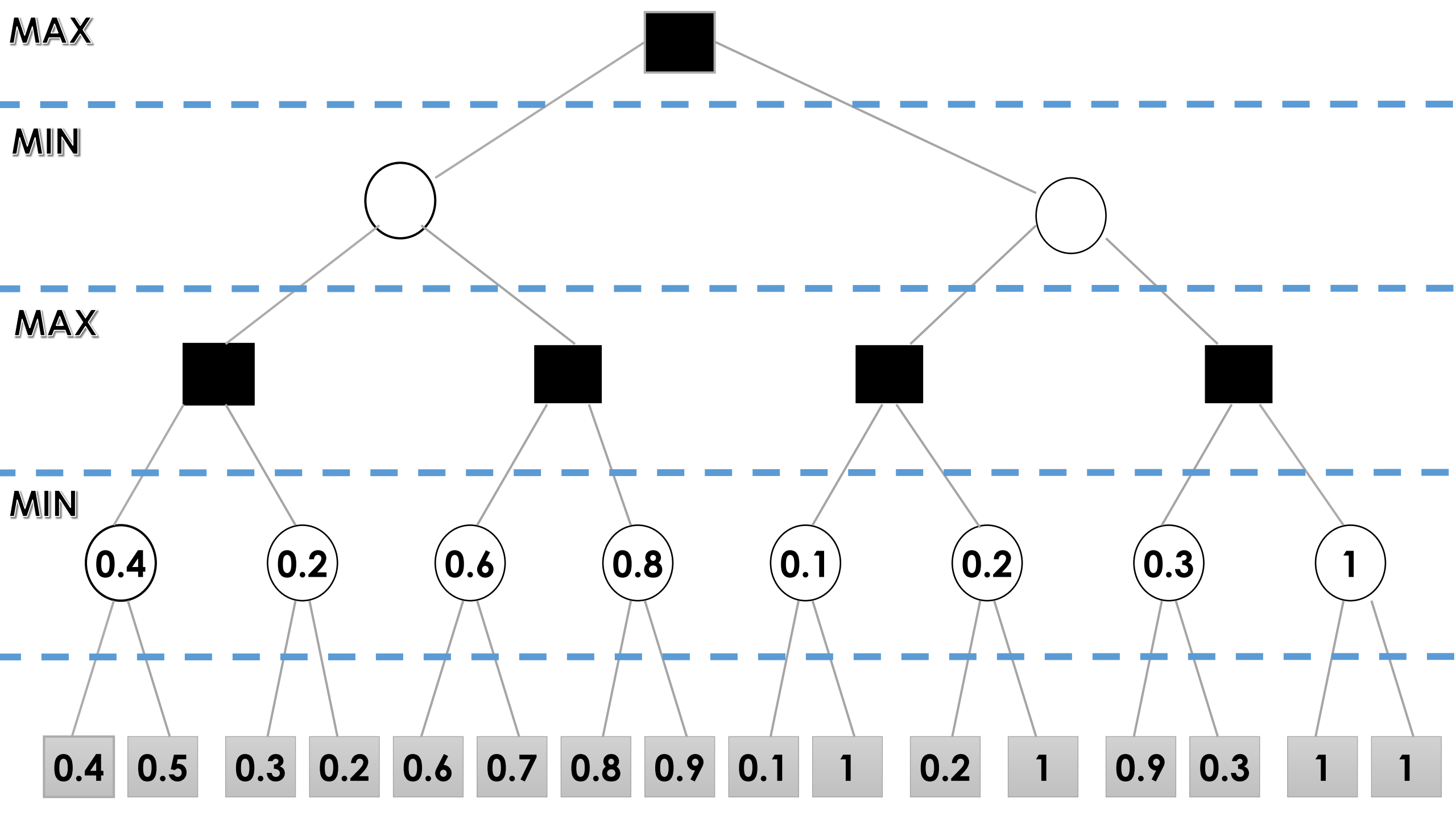
- One player is the MAX player
- The opposite player is the MIN player
- The value of a state in simple zero-sum games:

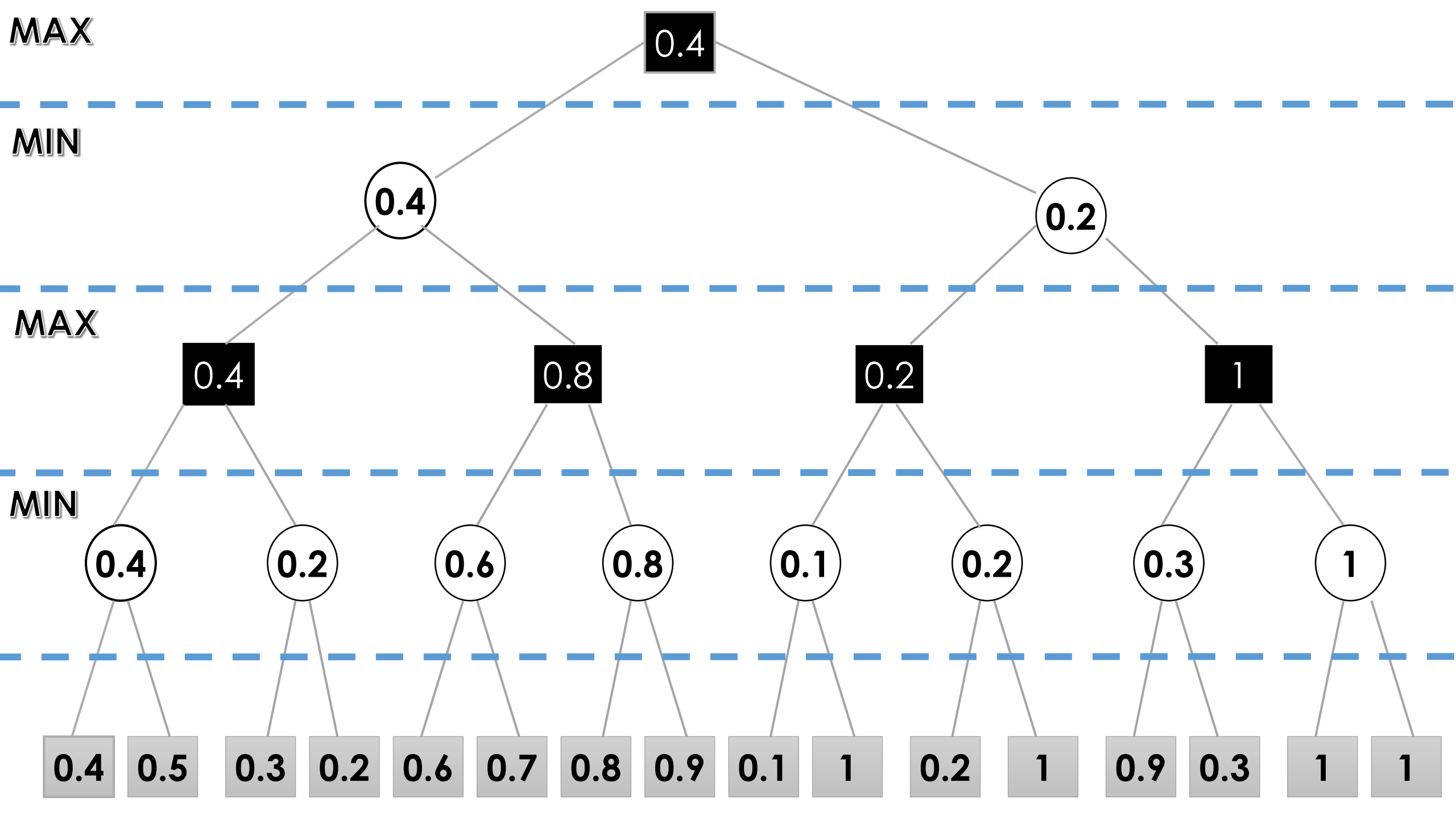
$$v(state) = \begin{cases} +1 & MAX\ wins \\ 0 & Draw \\ -1 & MIN\ wins \end{cases}$$

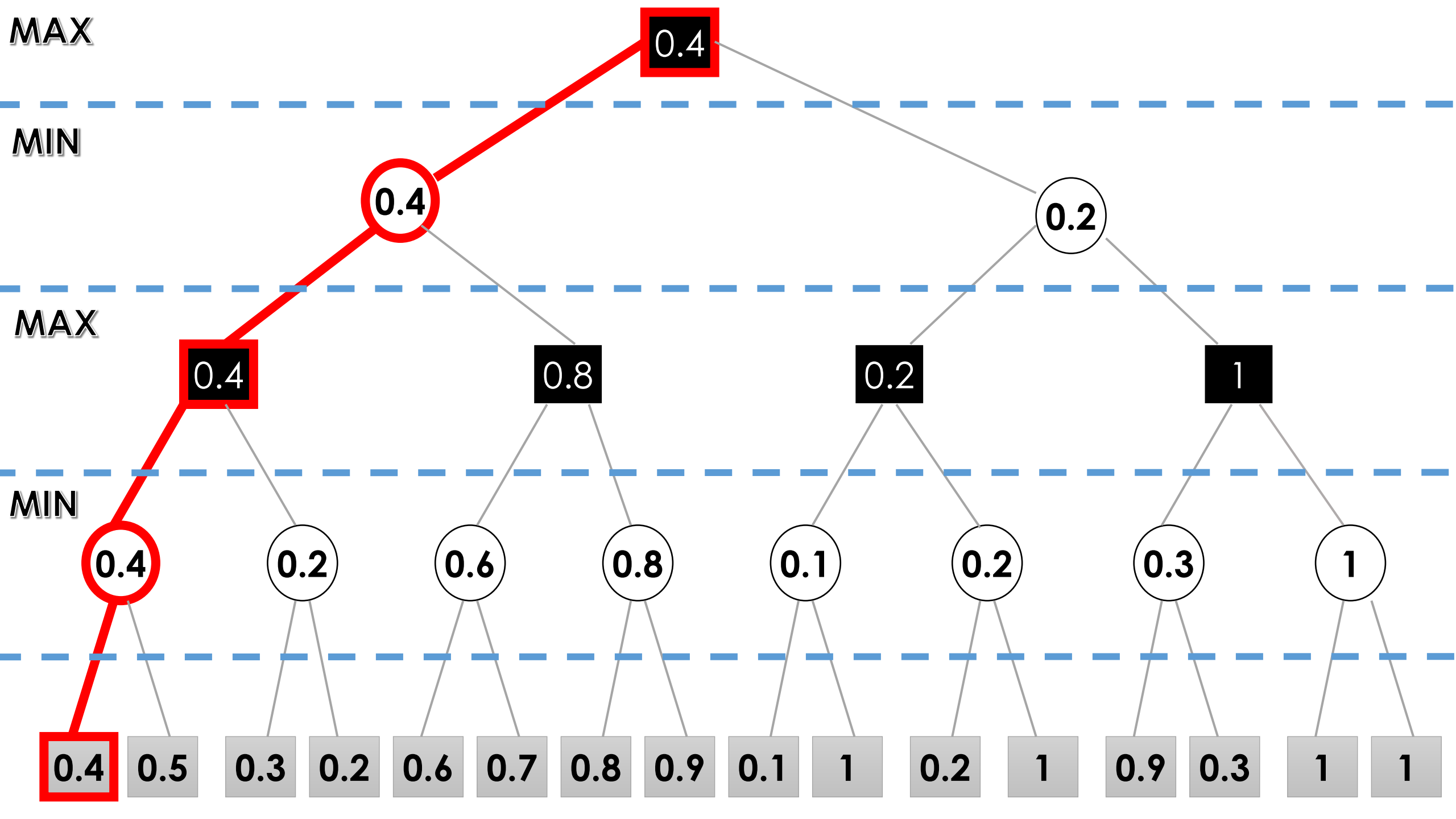



0.4
0.5
0.3
0.2
0.6
0.7
0.8
0.9
0.1
1
0.2
1
0.9
0.3
1
1


MAX
MIN
MAX
MIN
0.4
0.5
0.3
0.2
0.6
0.7
0.8
0.9
0.1
1
0.2
1
0.9
0.3
1
1


MAX
MIN
MAX
MIN
0.4
0.2
0.6
0.8
0.1
0.2
0.3
1
0.4
0.5
0.3
0.2
0.6
0.7
0.8
0.9
0.1
1
0.2
1
0.9
0.3
1
1


MAX
MIN
MAX
MIN
0.4
0.4
0.2
0.4
0.8
0.2
1
0.4
0.2
0.6
0.8
0.1
0.2
0.3
1
0.4
0.5
0.3
0.2
0.6
0.7
0.8
0.9
0.1
1
0.2
1
0.9
0.3
1
1


MAX
MIN
MAX
MIN
0.4
0.4
0.2
0.4
0.8
0.2
1
0.4
0.2
0.6
0.8
0.1
0.2
0.3
1
0.4
0.5
0.3
0.2
0.6
0.7
0.8
0.9
0.1
1
0.2
1
0.9
0.3
1
1

Case study: The game of GO

- 1968, Pattern recognition to detect Ko (Albert Zobrist)
  - A bunch of tries.
  - Not much success.
- 2006, Crazy Stone, Remi Coulom (MCTS)
- 2008, Mogo, (added expert knowledge)
- 2010, MogoTW won 19x19 (7 handicap)
- 2011, Zen reached 5 dan
- 2013, Crazy-stone beats Yoshio Ishida in 19x19 (4 handicap)
- 2015, AlphaGo. European Go champion
- 2016, AlphaGo. (no handicap)
- 2017, AlphaGo. World class

What do people know that we do not?

- Super strong heuristic?
- Better ability to explore huge trees?
- Model-free vs model-based learning?
- Better learning abilities in general?
- Other mechanisms?

Case study: The game of GO

- 1968, Pattern recognition to detect Ko (Albert Zobrist).
  - Bunch of others.
  - No much success.
- 2006, Crazy Stone, Remi Coulom (MCTS)
- 2008, Mogo, (added expert knowledge).
- 2010, MogoTW won 19x19 (7 handicap)
- 2011, Zen reached 5 dan.
- 2013, Crazy-stone beats Yoshio Ishida in 19x19 (4 handicap)
- 2015, AlphaGo. European Go champion.
- 2016, AlphaGo, (no handicap)
- 2017, AlphaGo. World class

Two major catalysts:

Case study: The game of GO

- 1968, Pattern recognition to detect Ko (Albert Zobrist).
  - Bunch of others.
  - No much success
- 2006, Crazy Stone, Remi Coulom (MCTS)
- 2008, Mogo, (added expert knowledge)
- 2010, MogoTW won 19x19 (7 handicap)
- 2011, Zen reached 5 dan
- 2013, Crazy-stone beats Yoshio Ishida in 19x19 (4 handicap)

Deep Learning

- 2015, AlphaGo. European Go champion
- 2016, AlphaGo. (no handicap)
- 2017, AlphaGo. World class

Two major catalysts:

Deep Learning

Case study: The game of GO

- 1968, Pattern recognition to detect Ko (Albert Zobrist).
  - Bunch of others.
  - No much success.

MCTS

- 2006, Crazy Stone, Remi Coulom (MCTS)
- 2008, Mogo, (added expert knowledge)
- 2010, MogoTW won 19x19 (7 handicap)
- 2011, Zen reached 5 dan
- 2013, Crazy-stone beats Yoshio Ishida in 19x19 (4 handicap)

Deep Learning

- 2015, AlphaGo. European Go champion
- 2016, AlphaGo. (no handicap)
- 2017, AlphaGo. World class

Two major catalysts:

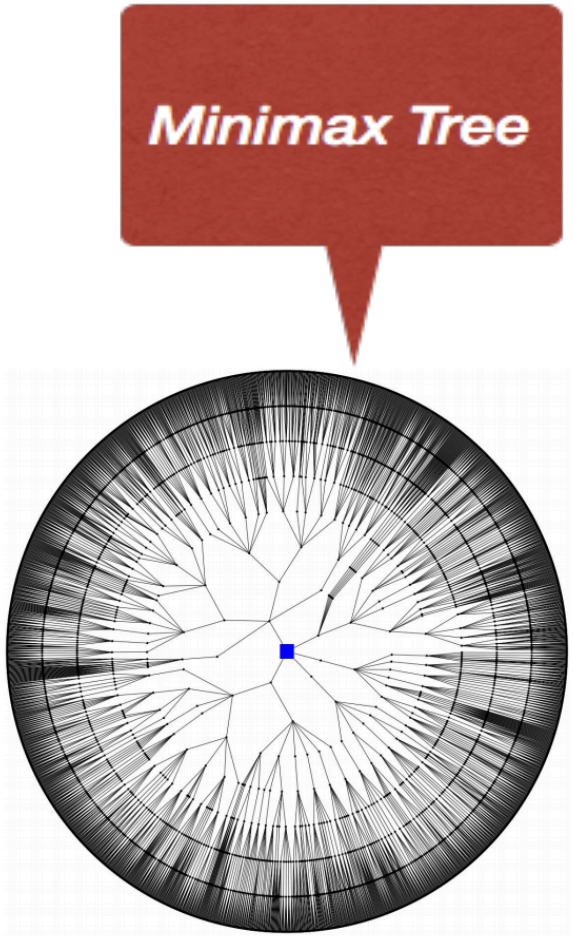
Deep Learning

Monte-Carlo Tree Search.

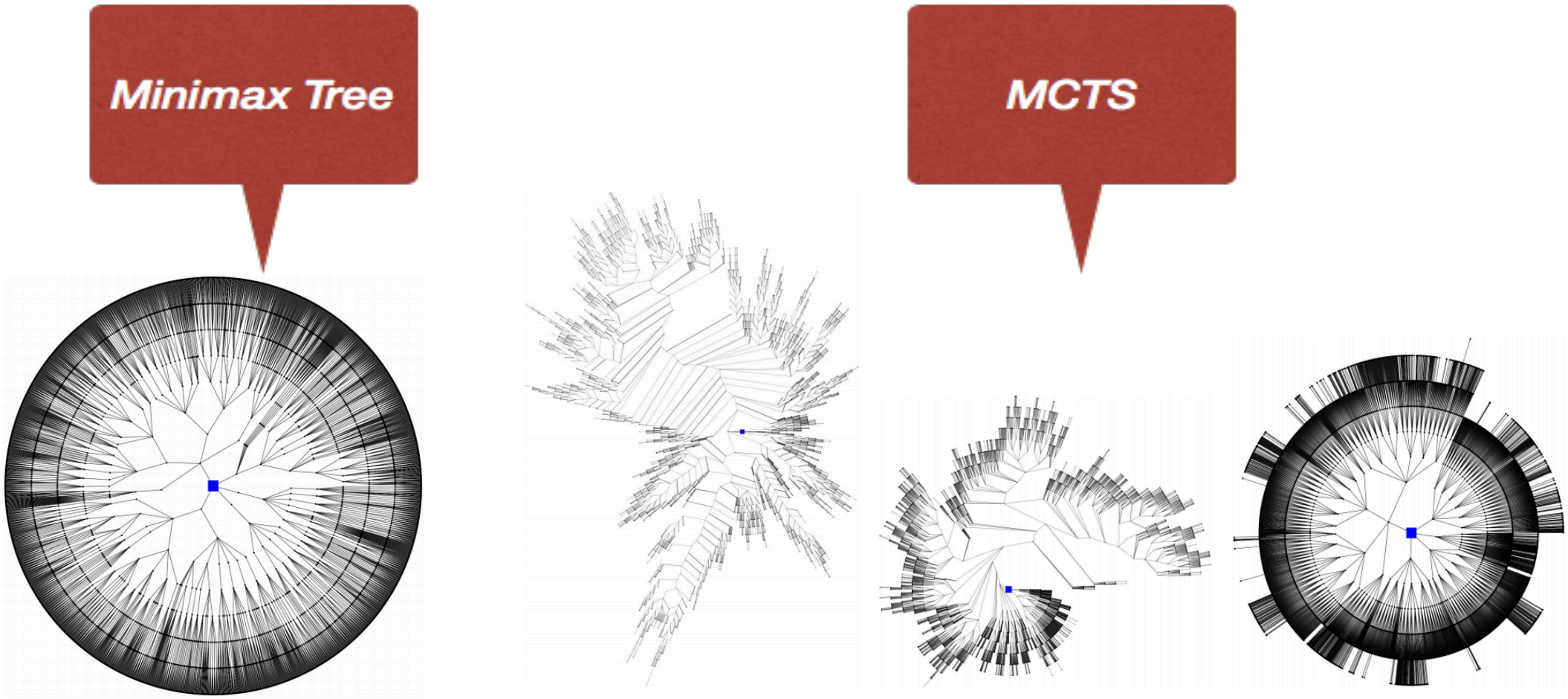
Monte-Carlo Tree Search

- Iterative process
- Builds **asymmetric** search trees
- Building is guided by **value estimations** stored in each node
- Estimations can be based on **simulations**

* For a survey paper on MCTS see (Browne et al).



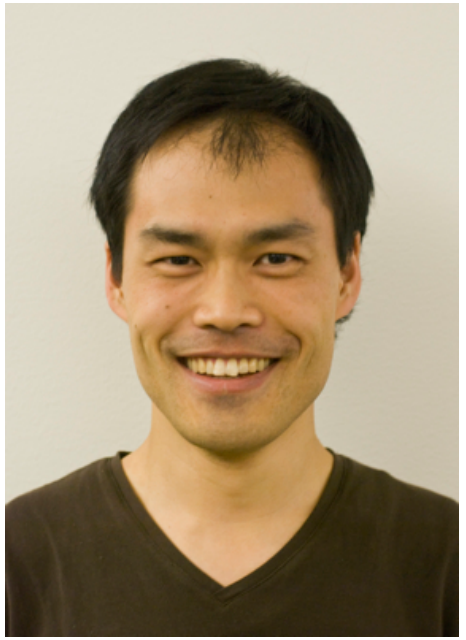
(Ramanujan et al. 2011)



(Ramanujan et al. 2011)

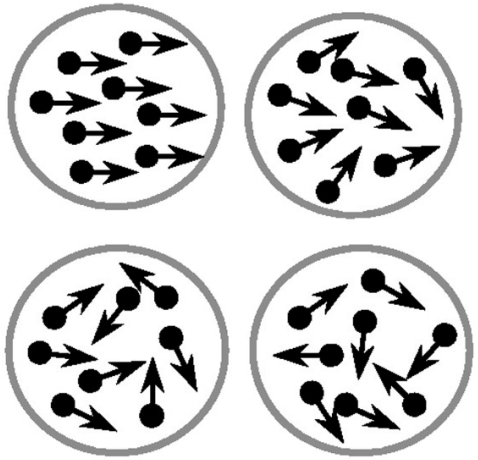
(See *Browne et al. 2012*)

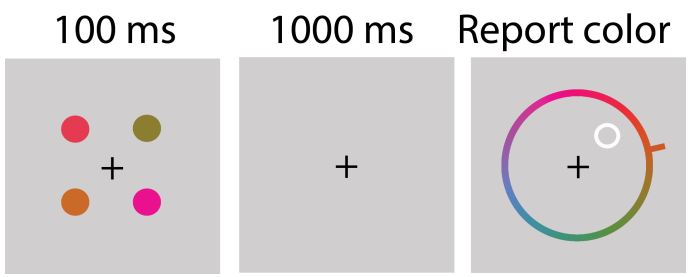
Methods that are algorithmically efficient are also “human-like”.



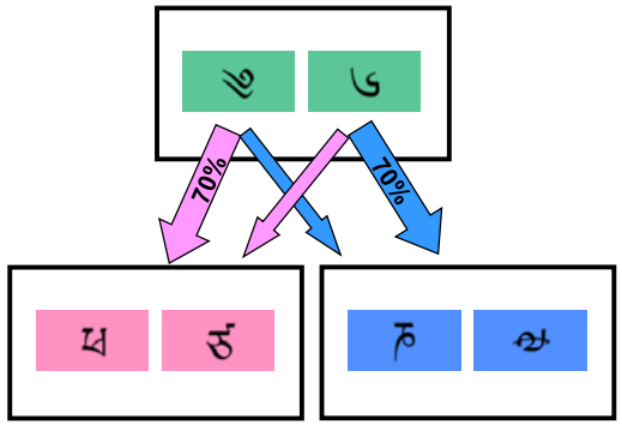
Wei Ji Ma

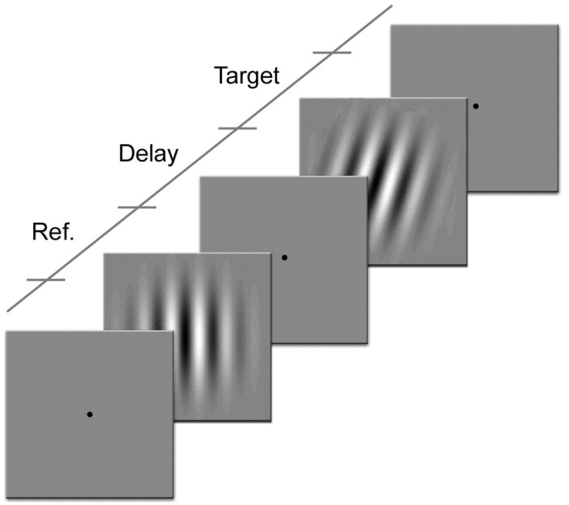
Perception/working memory/attention tasks





YELLOW BLUE ORANGE
BLACK RED GREEN
PURPLE YELLOW RED
ORANGE GREEN BLUE
BLUE RED PURPLE
YELLOW RED GREEN



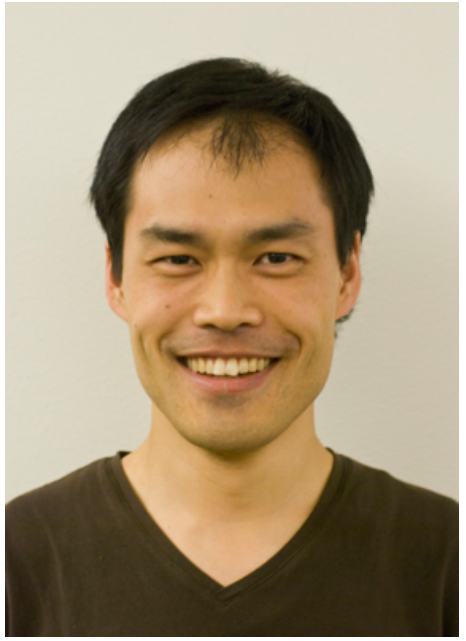


A Computational Model for Decision Tree Search

Bas Van Opheusden

Gianni Galbiati

Yunqi Li

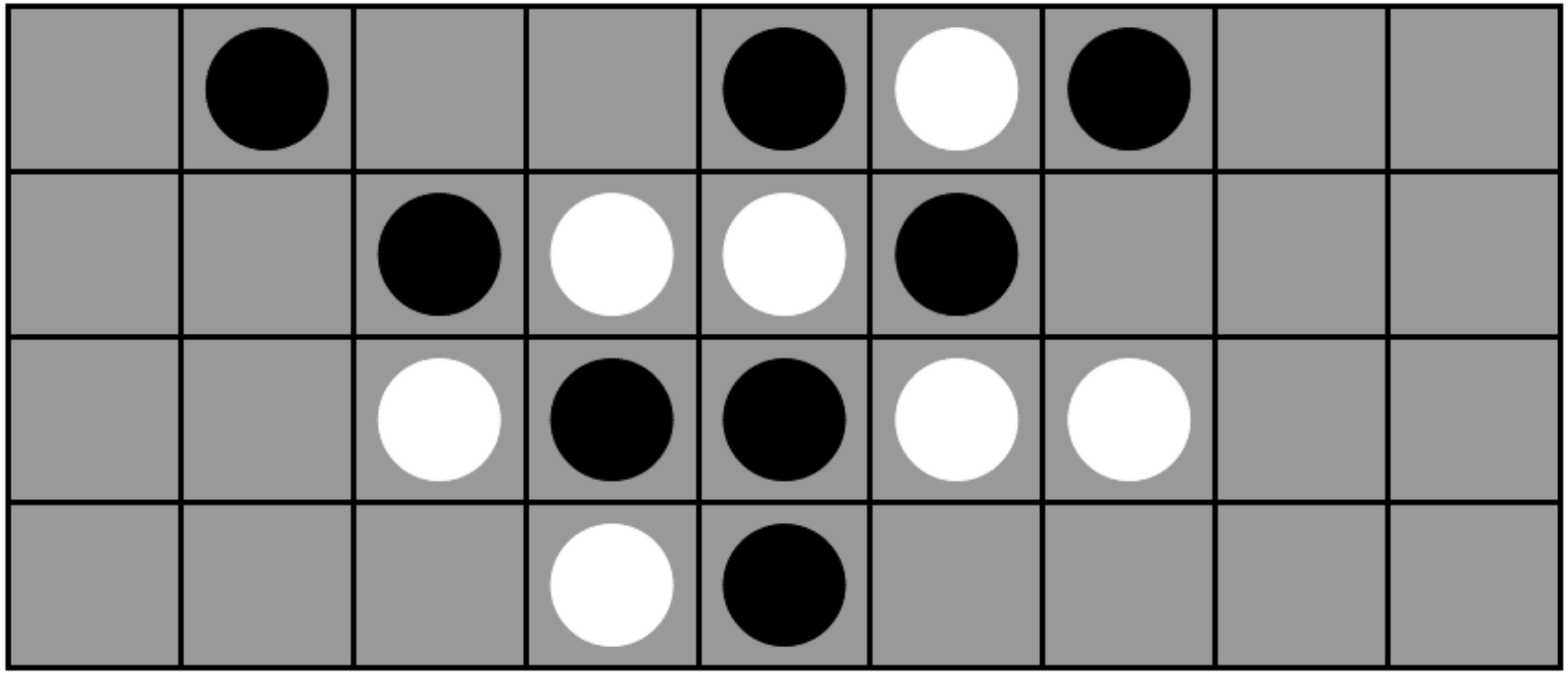


Wei Ji Ma

Human Game Playing

- Can we model such complex behavior?
- Are subjects performing tree search?
- Are subjects using heuristics?
- Are subjects using a similar process to MCTS?
- Methods:
  - Human behavior experiments
  - Eye tracking
  - Computational modeling
  - Model fitting
  - Model comparison

Task: make 4-in-a-row (horizontal, diagonal, vertical)



basvanopheusden.github.com

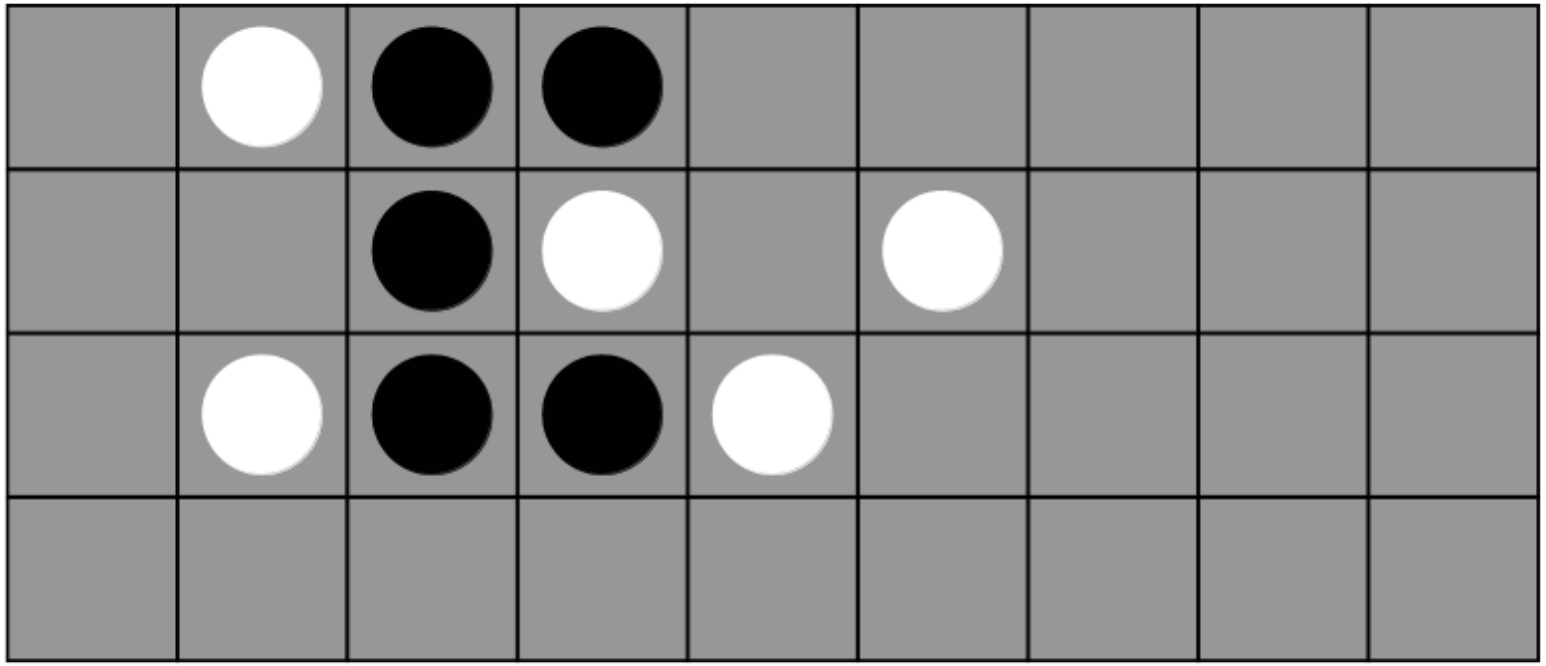
Experiment 1:
Human versus human

- 19 subject pairs, randomly paired
- Free play for 45 minutes, alternating colors
- Can we build a computational model for this behavior?

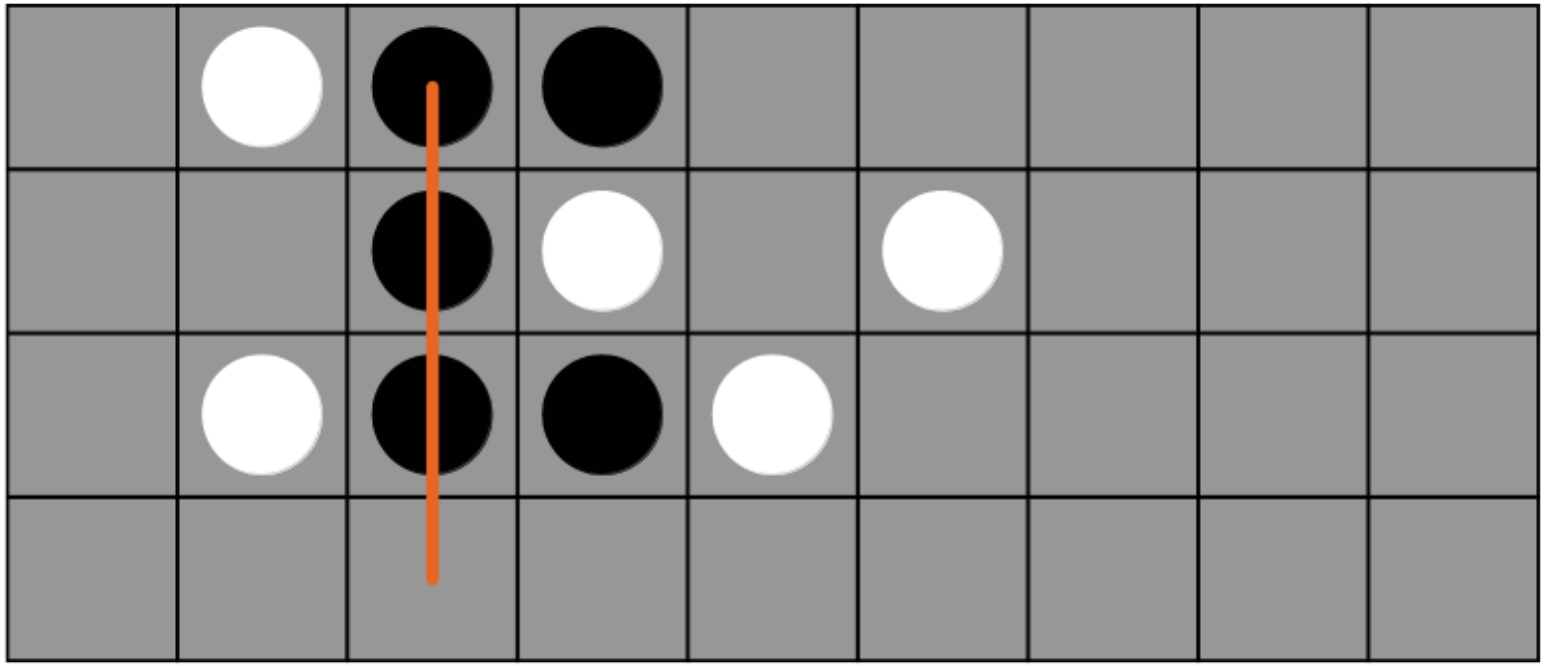
Components of the Model

1. ***Value* (heuristic)** function
   - Weighted linear sum of features of the board.
   - Features are "human-like"
2. **Tree search**
   - Mental simulation of expansions and value propagation.
3. **Sources of variability**: value noise, lapse rate, feature dropping.

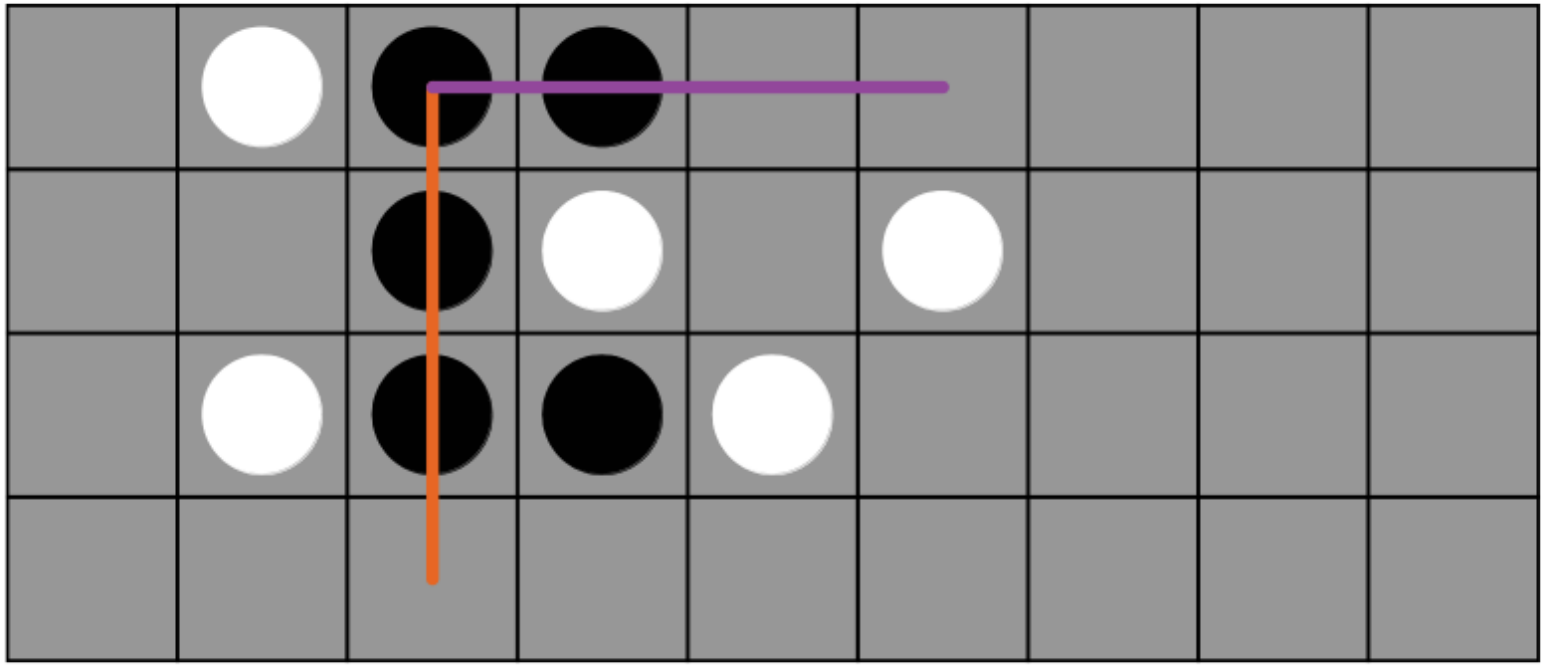
Model: evaluation function



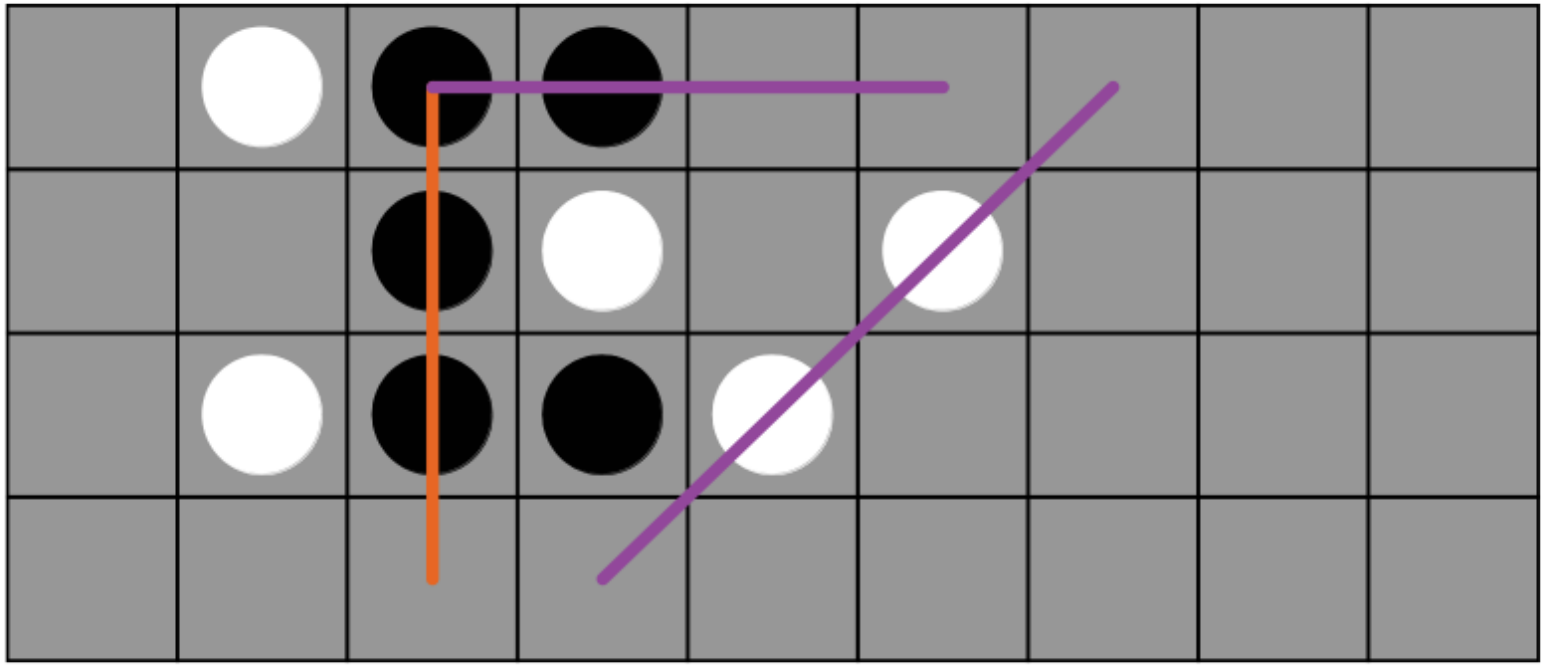
Model: evaluation function



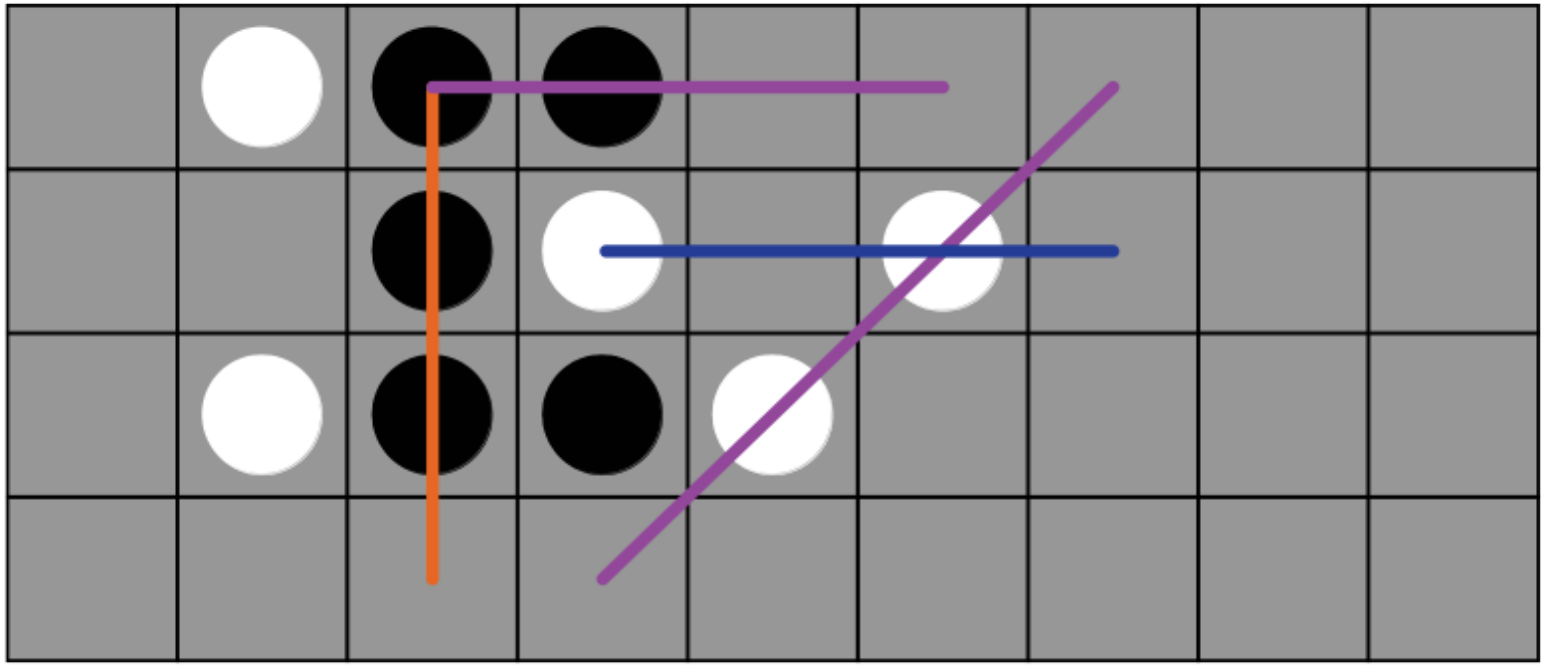
Model: evaluation function



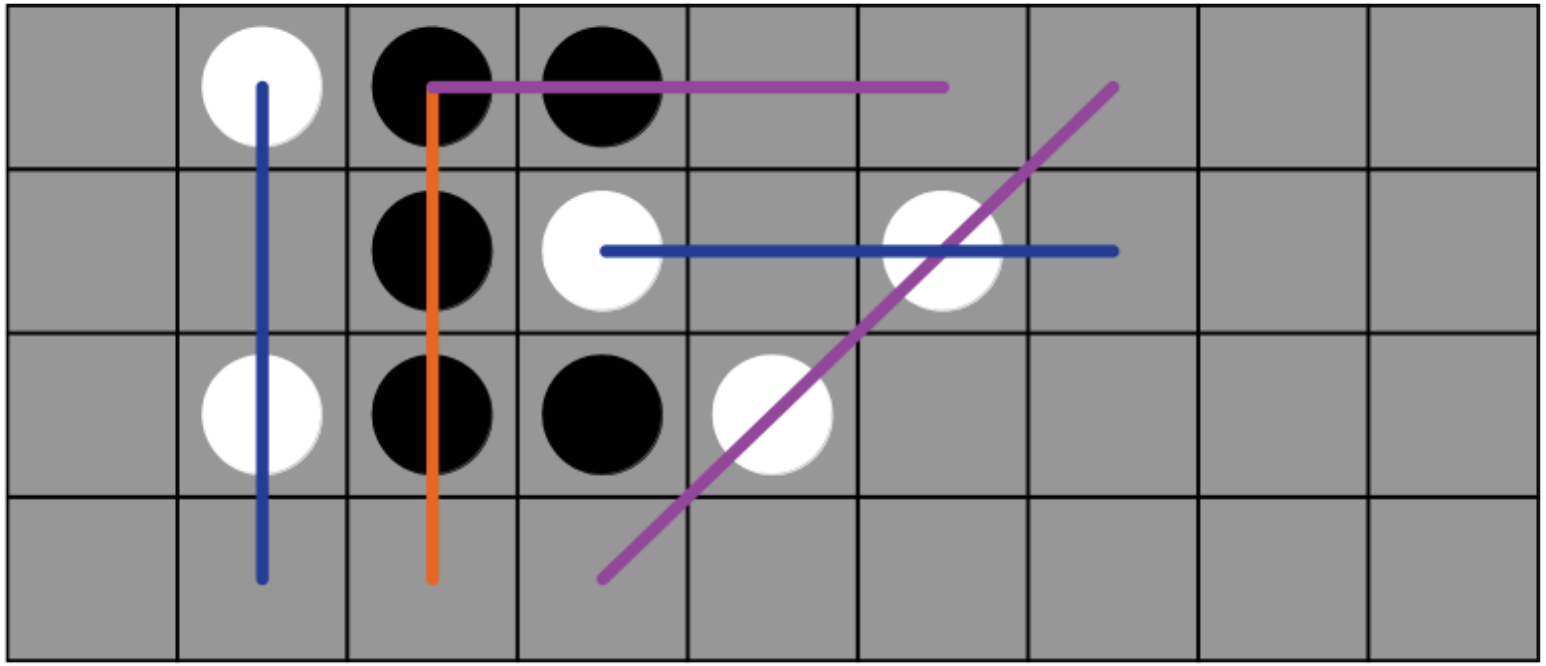
Model: evaluation function



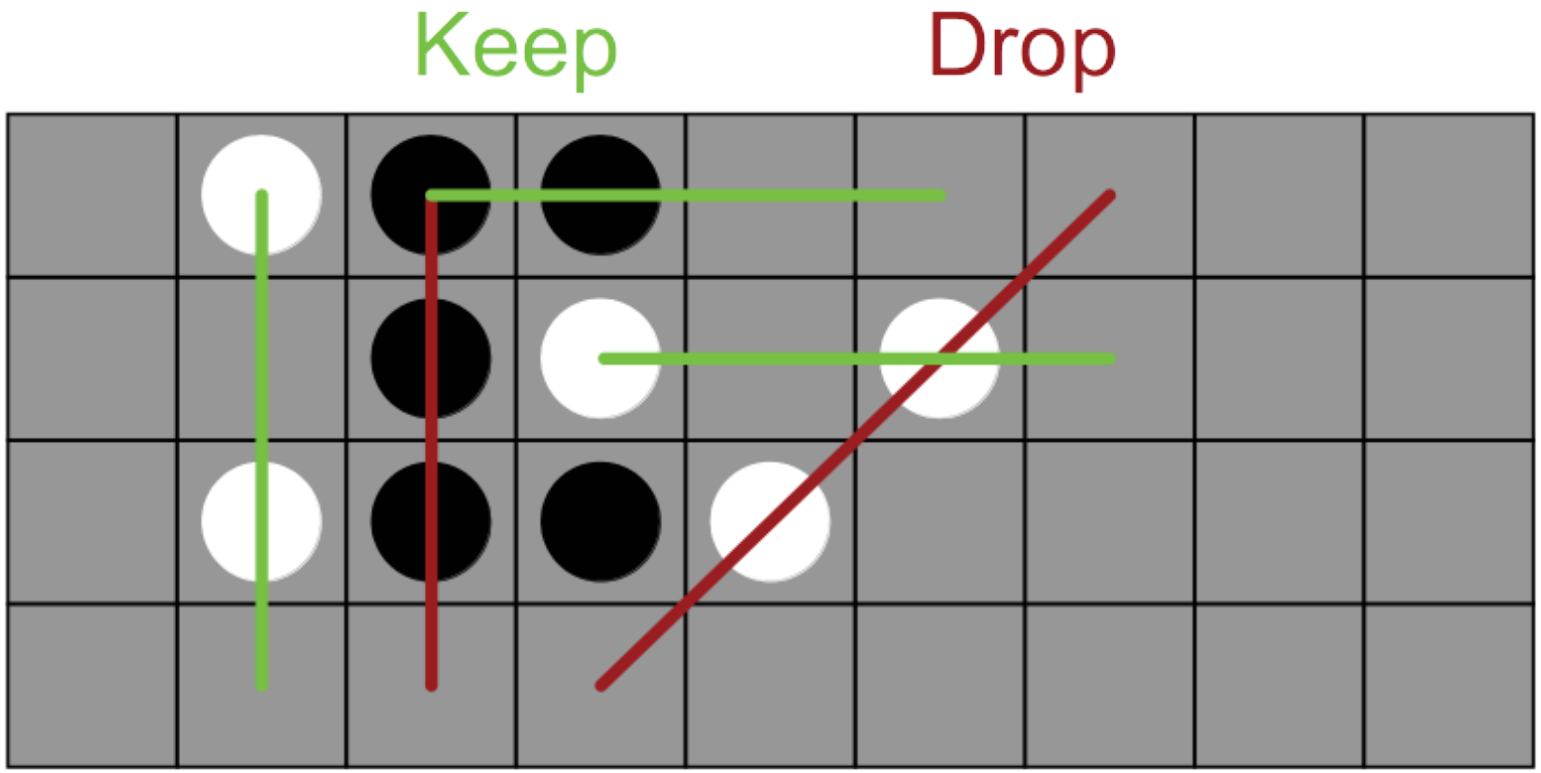
Model: evaluation function



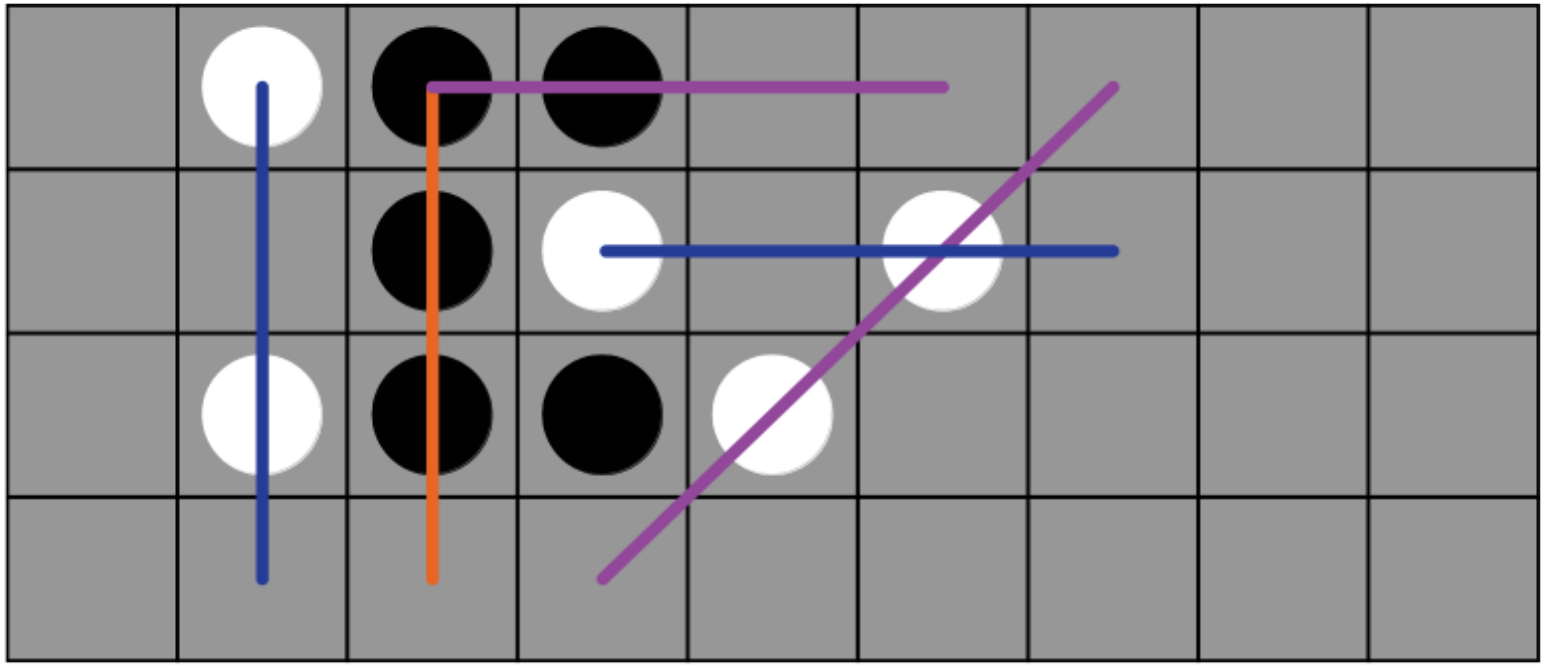
Model: evaluation function



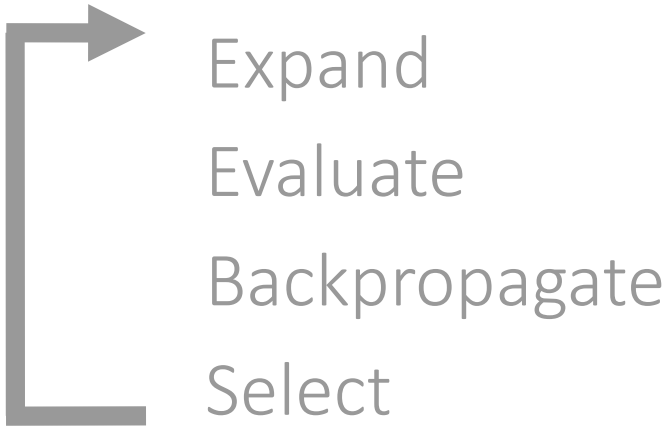
Model: feature dropping

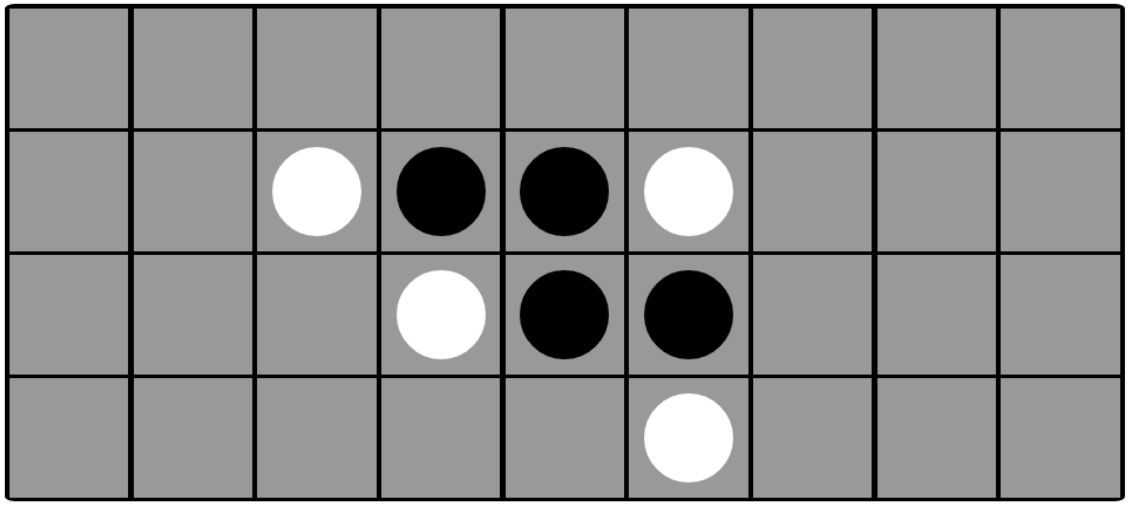


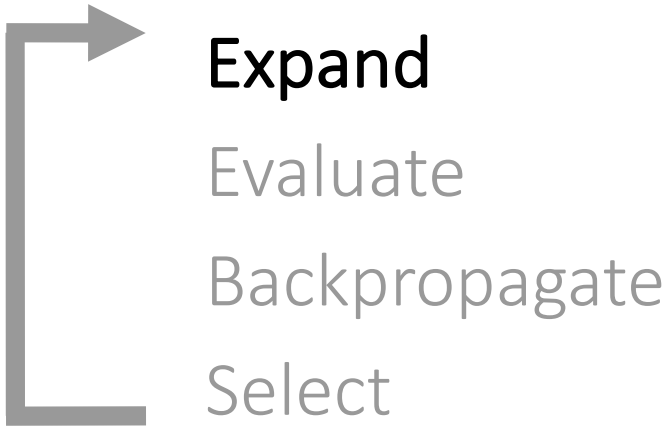
Model: evaluation function

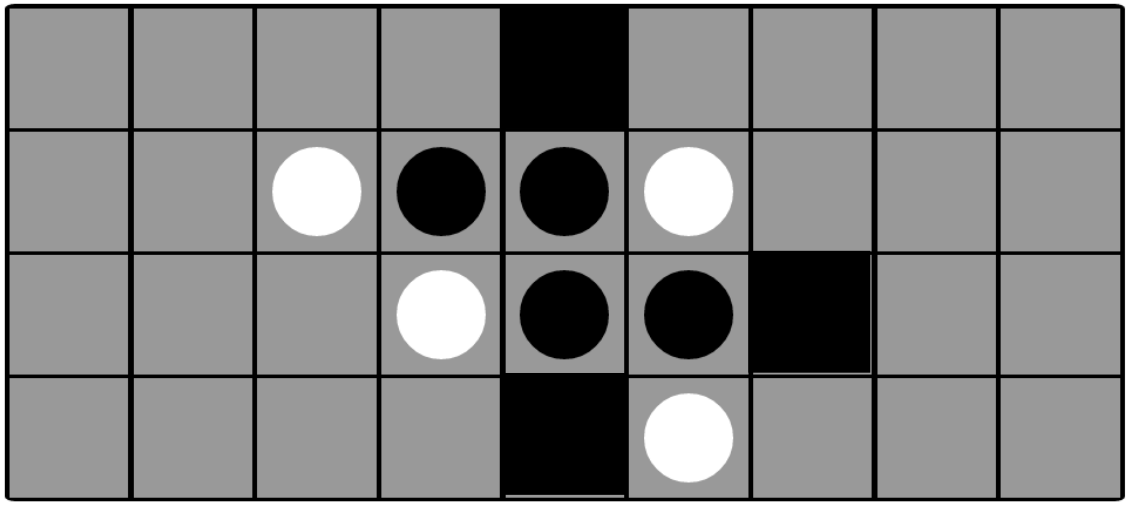


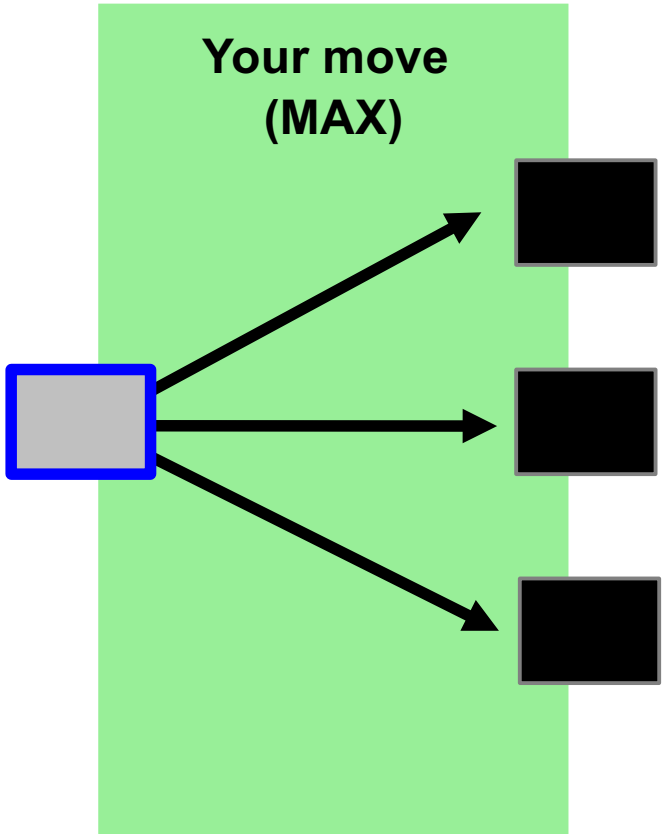
$$V(s) = \sum_{i=0}^{4} w_i f_i(s, \text{self}) - \sum_{i=0}^{4} w_i f_i(s, \text{opponent})$$

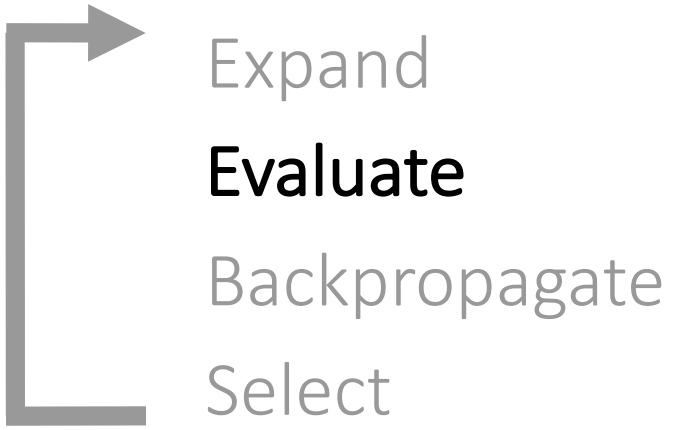

Expand
Evaluate
Backpropagate
Select

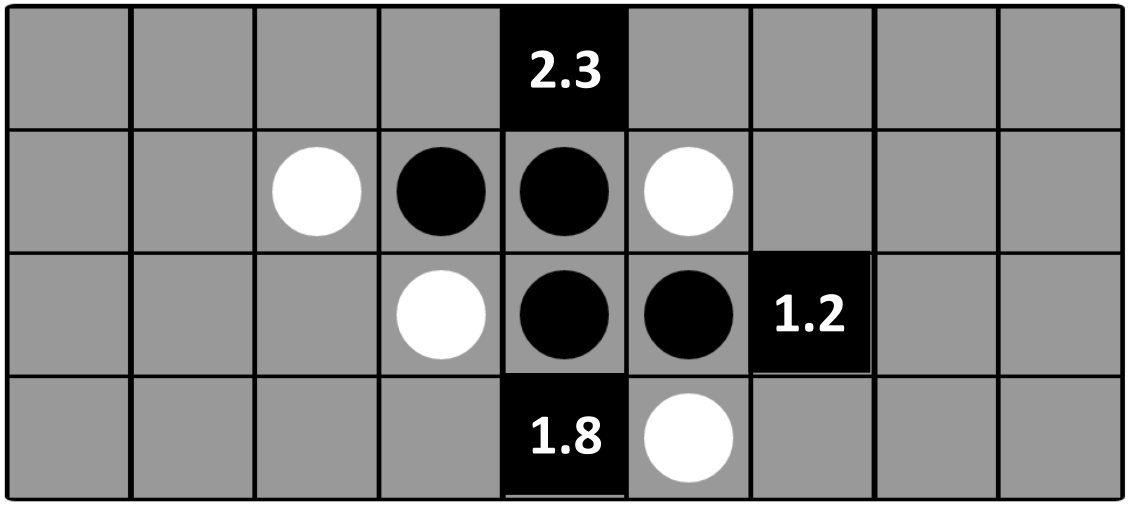


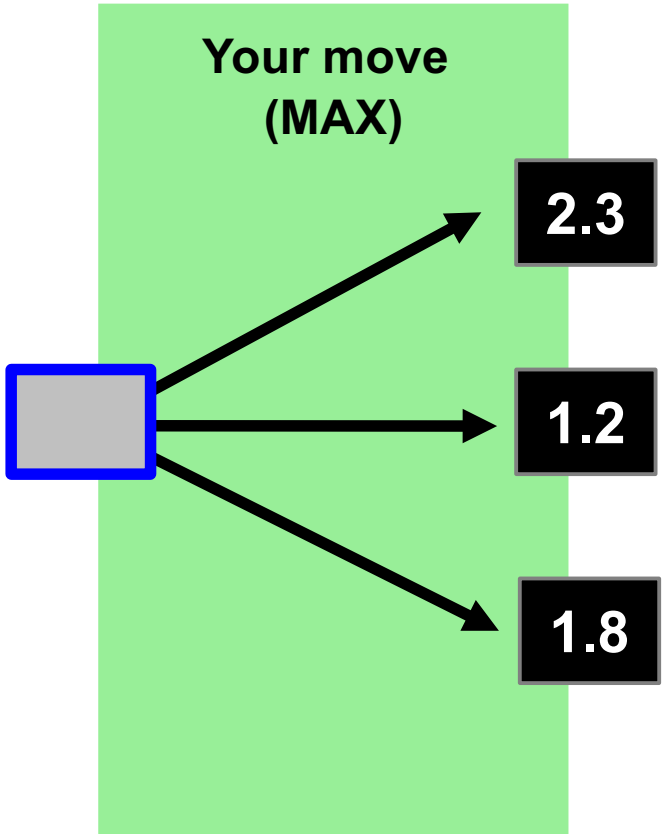

Expand
Evaluate
Backpropagate
Select

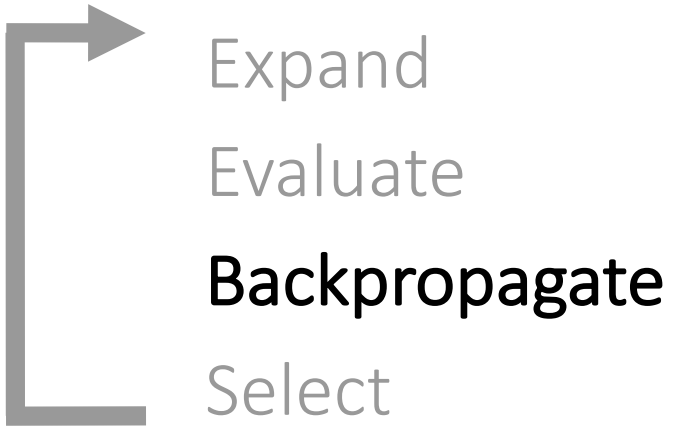


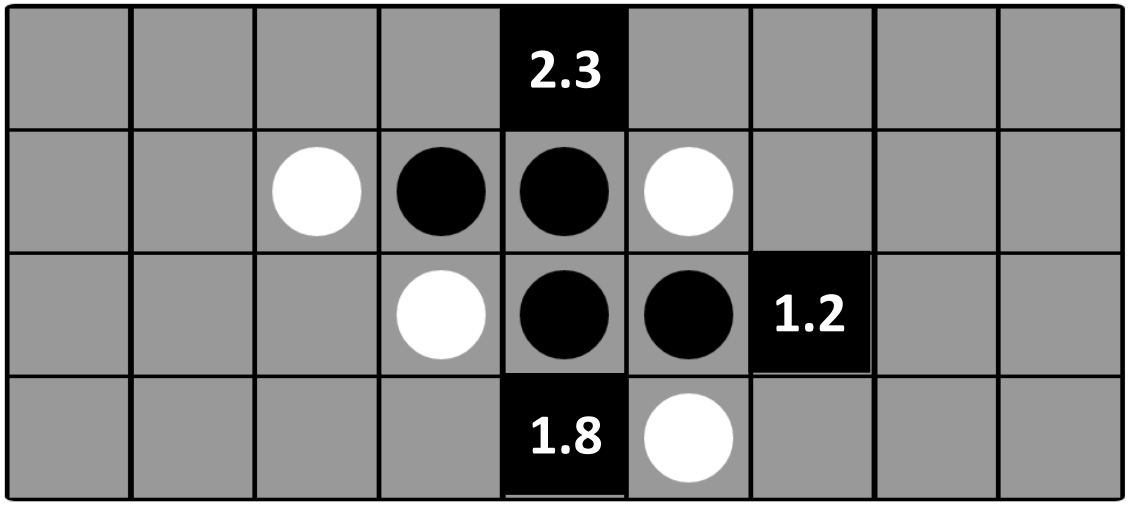

Your move
(MAX)

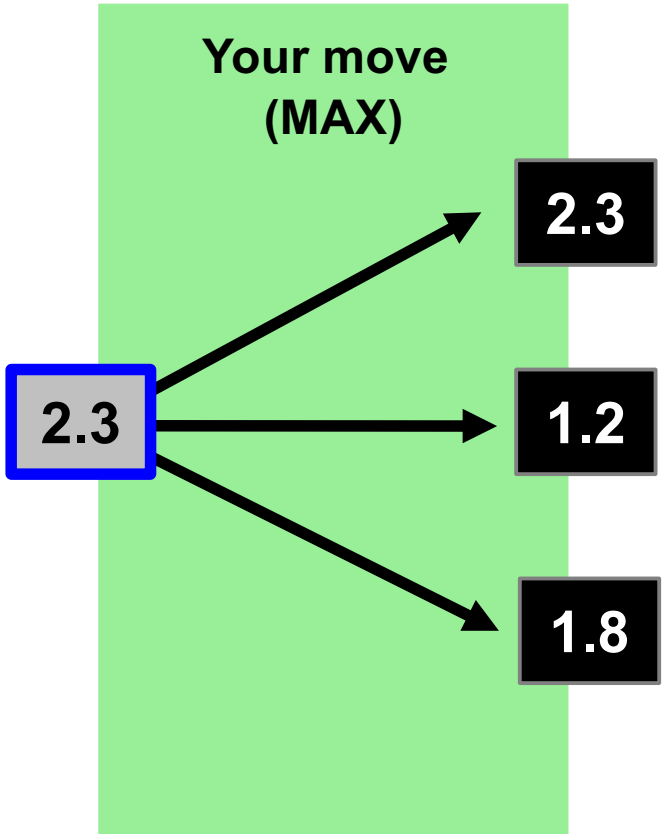

Expand
Evaluate
Backpropagate
Select

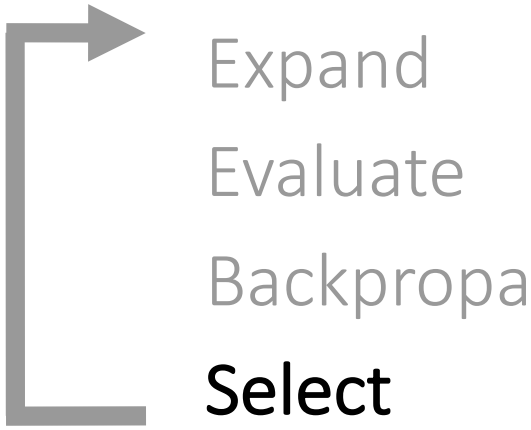

2.3
1.2
1.8

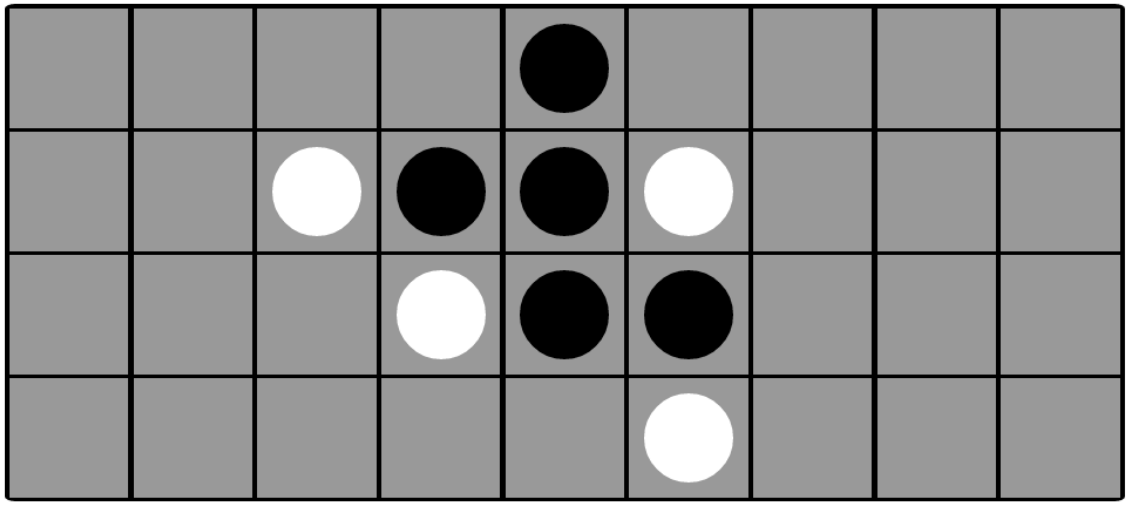

Your move
(MAX)
2.3
1.2
1.8

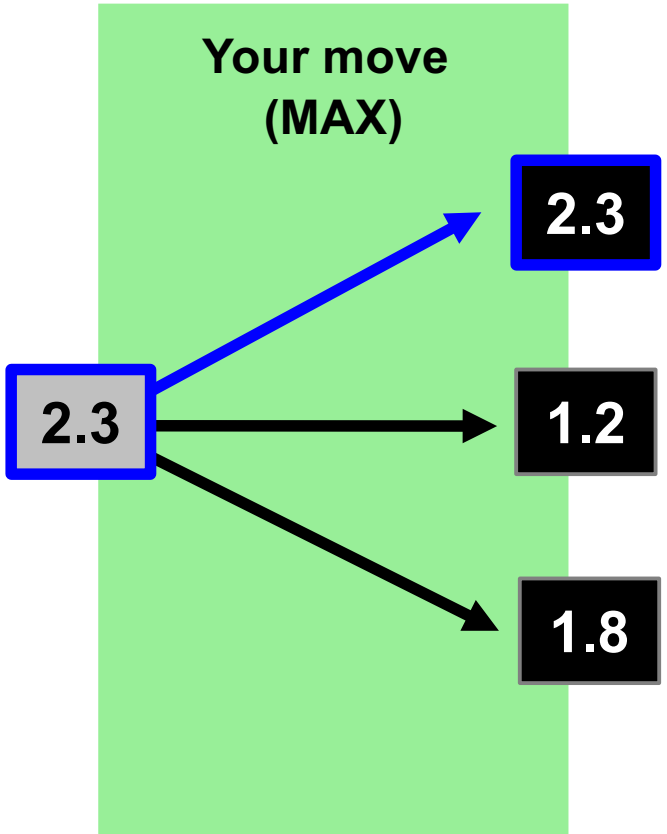

Expand
Evaluate
Backpropagate
Select

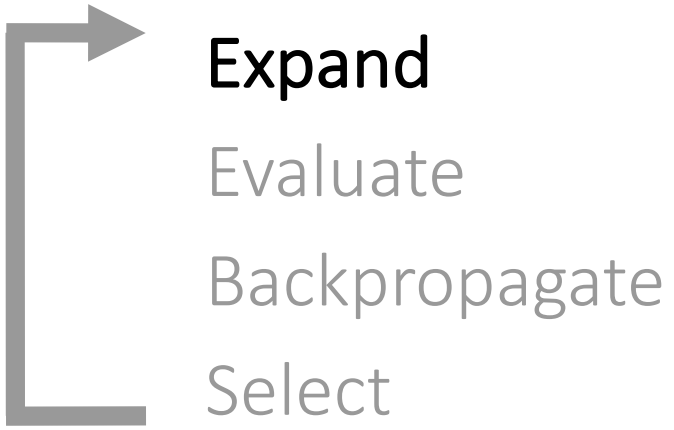

2.3
1.2
1.8

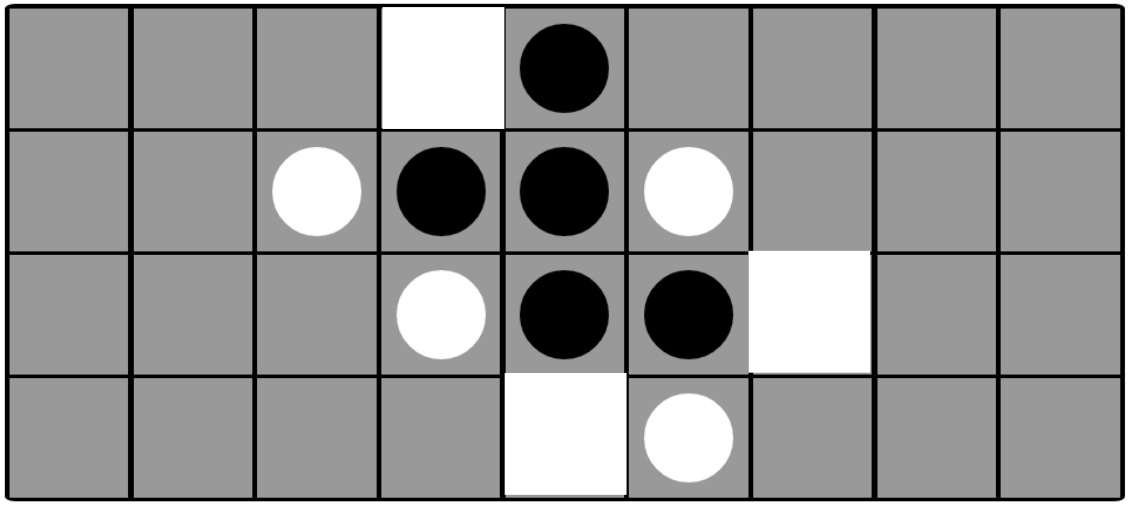

Your move
(MAX)
2.3
2.3
1.2
1.8

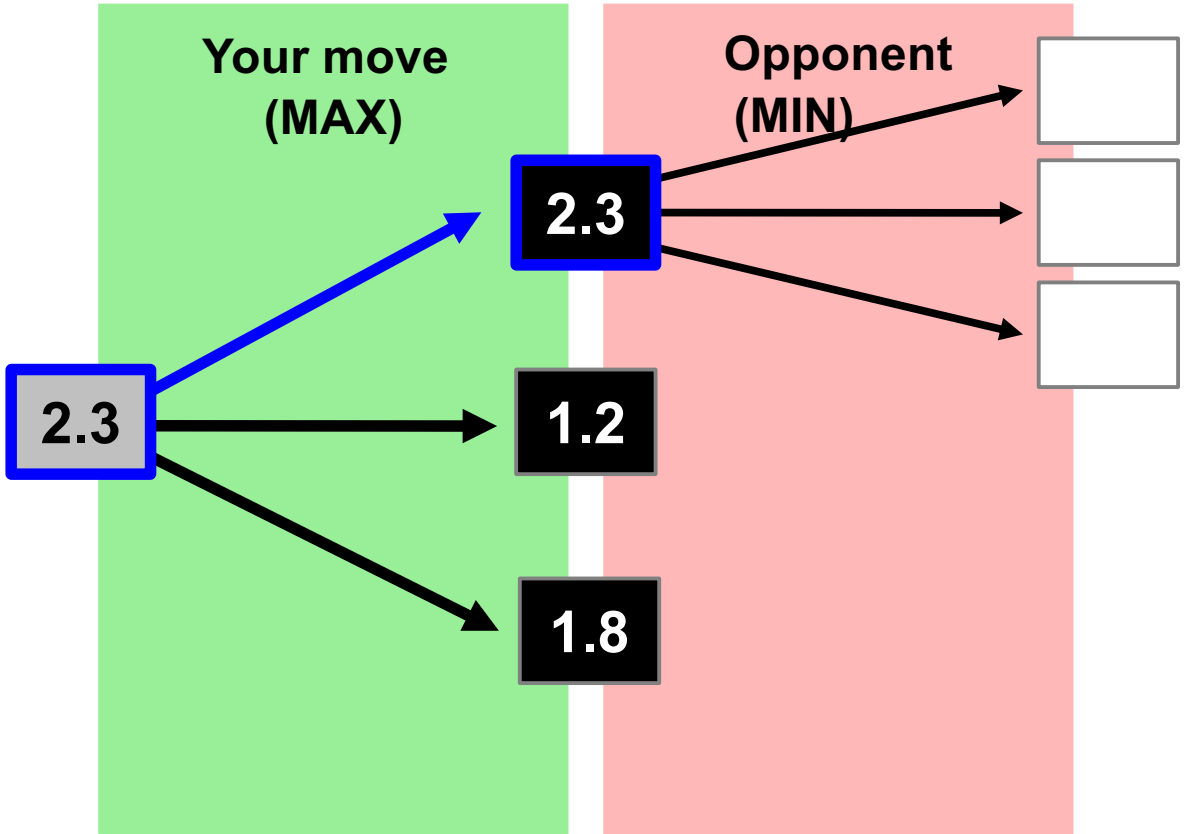

Expand
Evaluate
Backpropagate
Select

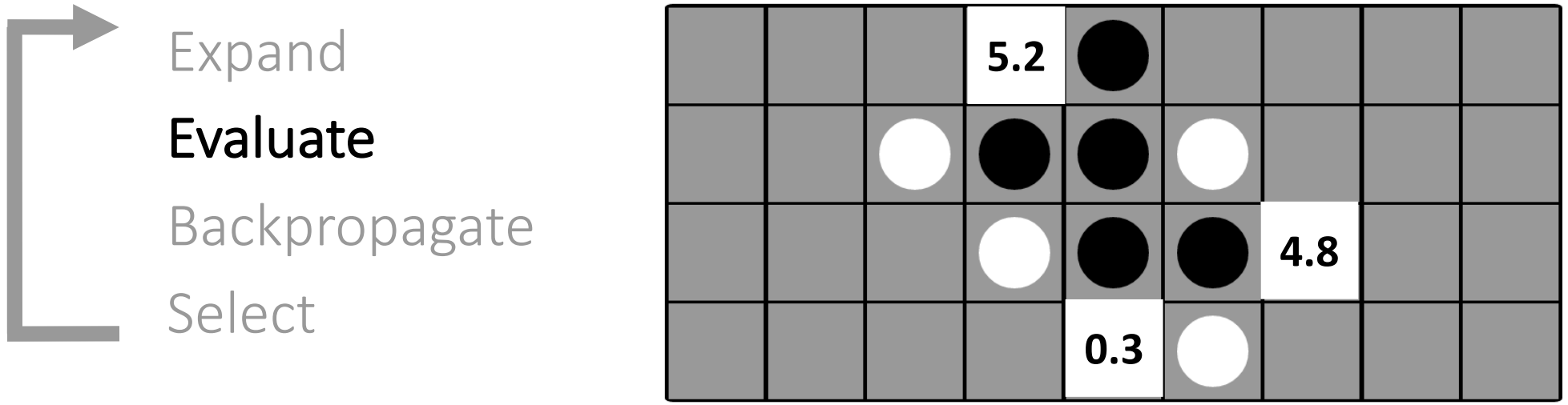


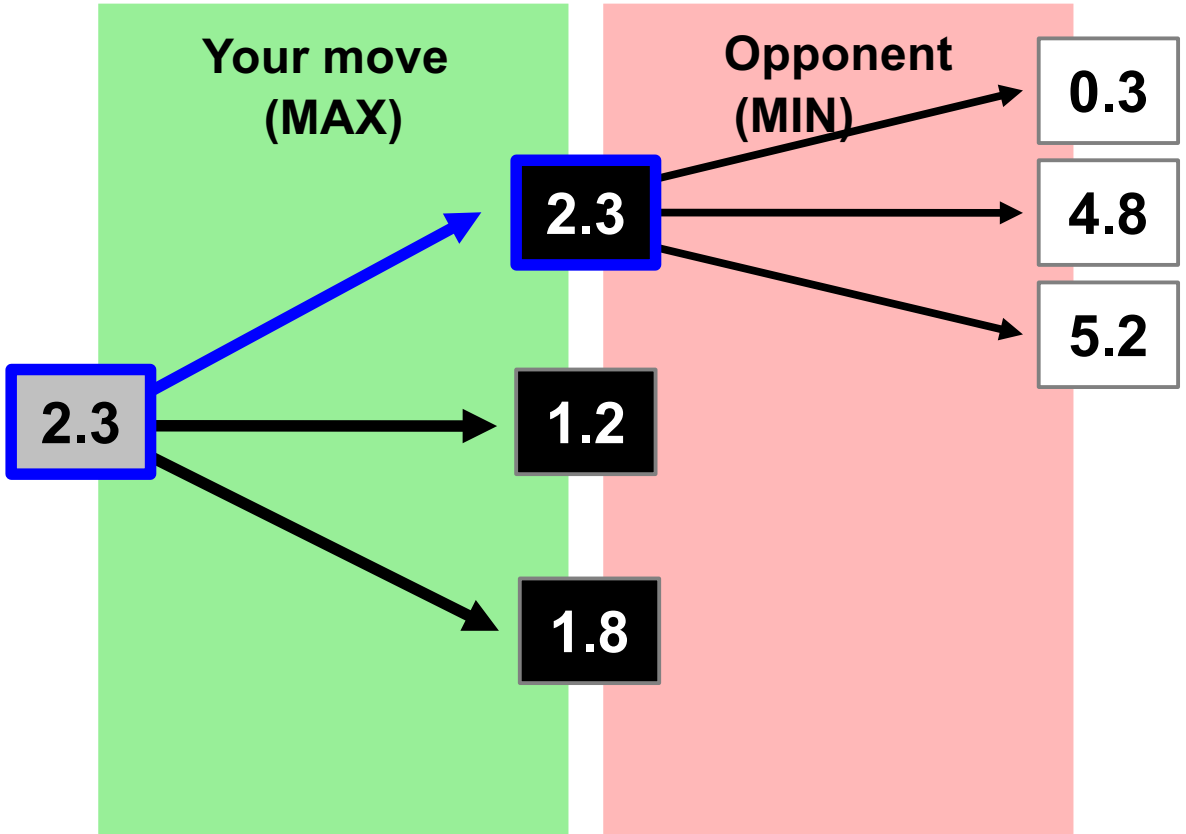

Your move
(MAX)
2.3
2.3
1.2
1.8

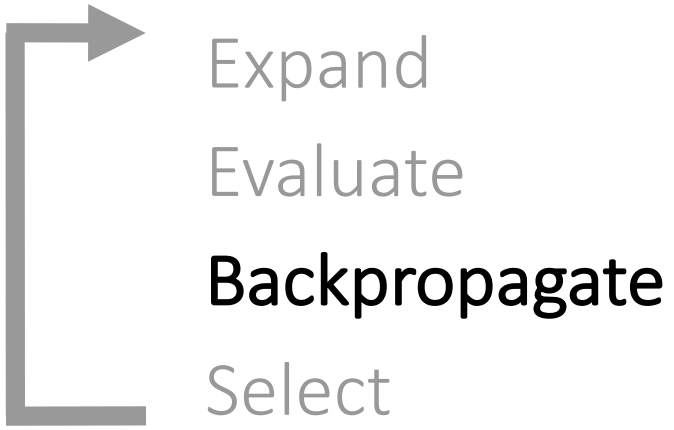

Expand
Evaluate
Backpropagate
Select

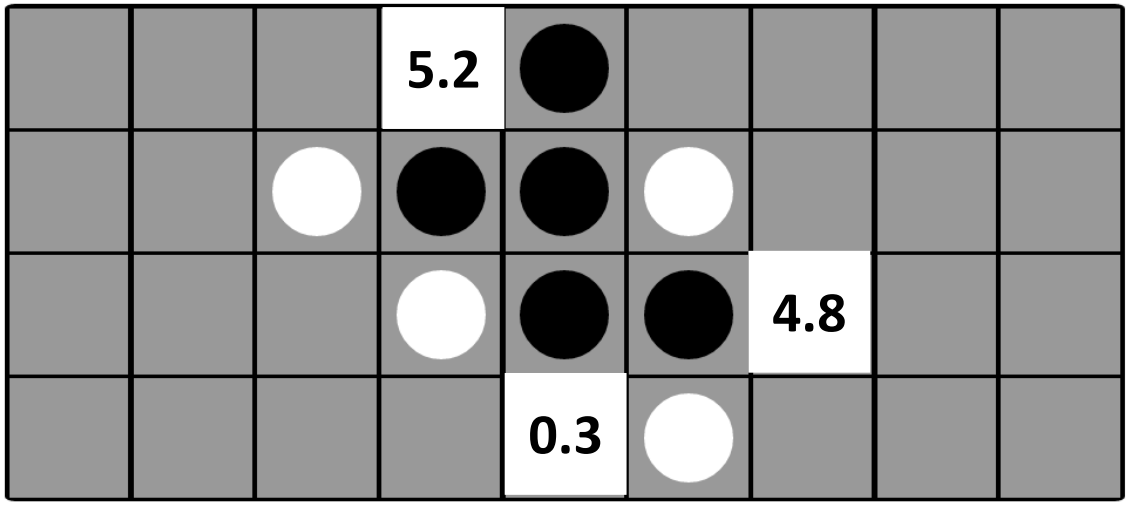


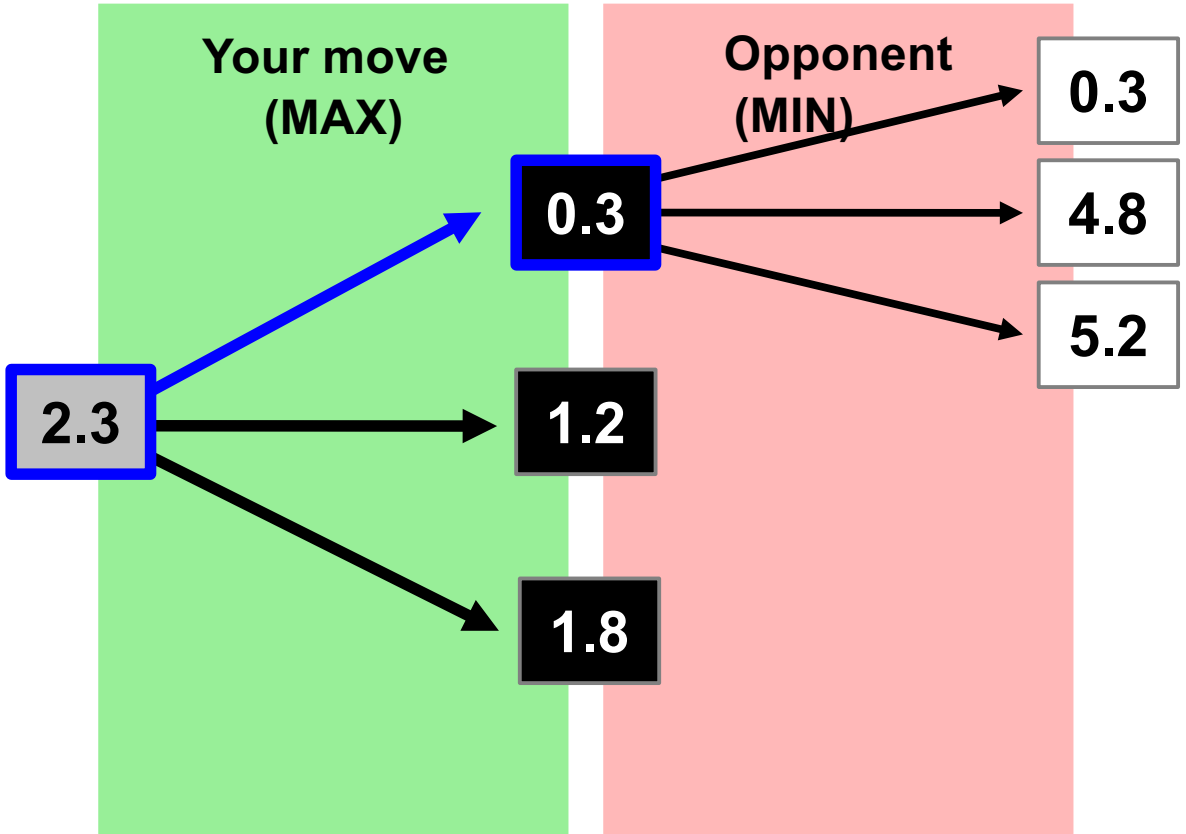

Your move
(MAX)
Opponent
(MIN)
2.3
2.3
1.2
1.8

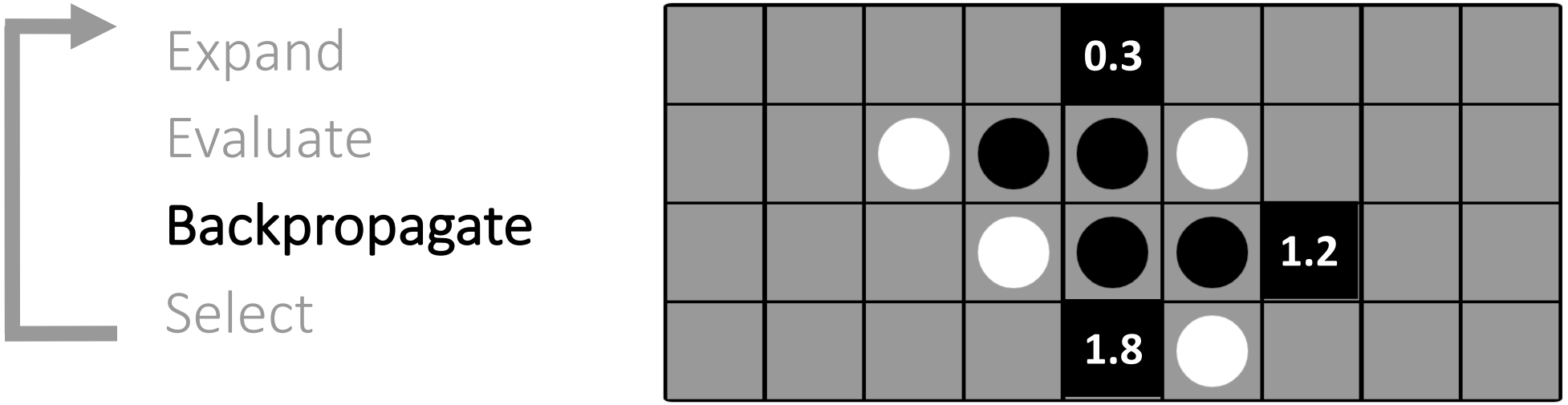

Expand
Evaluate
Backpropagate
Select
5.2
4.8
0.3

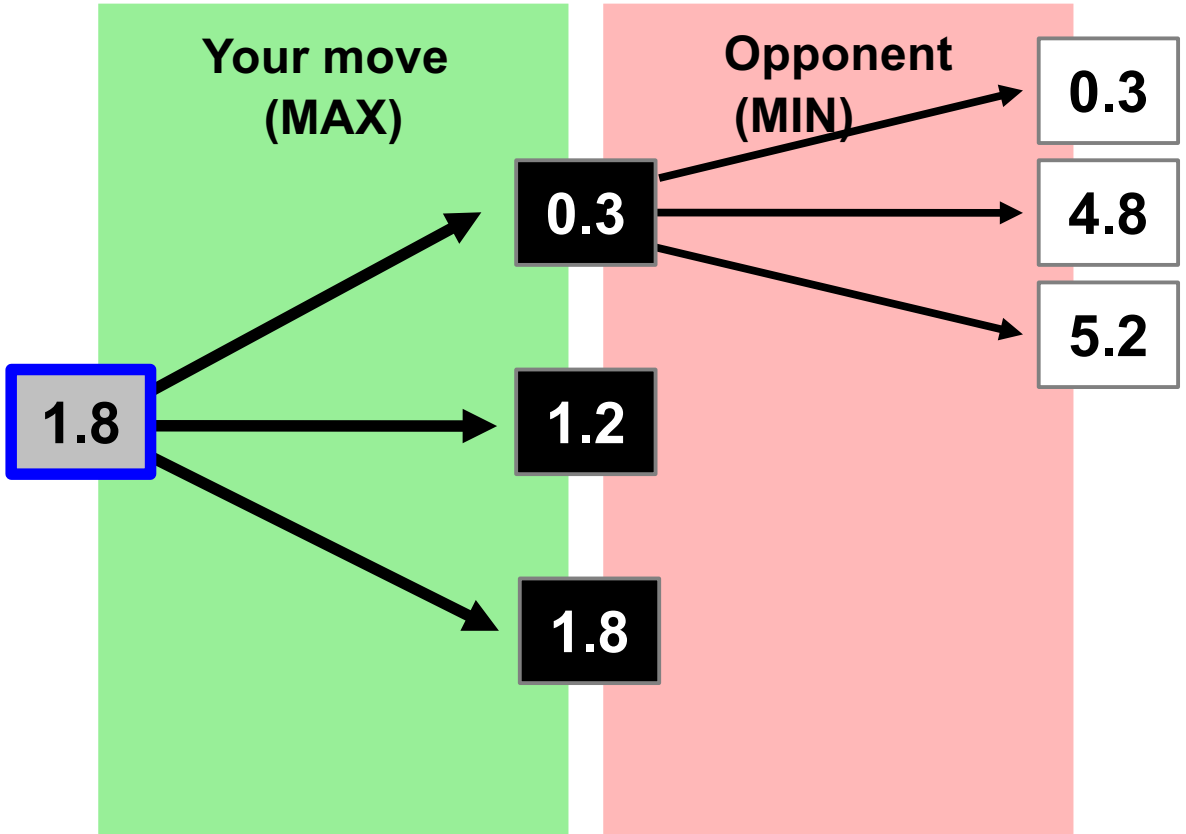

Your move
(MAX)
Opponent
(MIN)
2.3
2.3
1.2
1.8
0.3
4.8
5.2

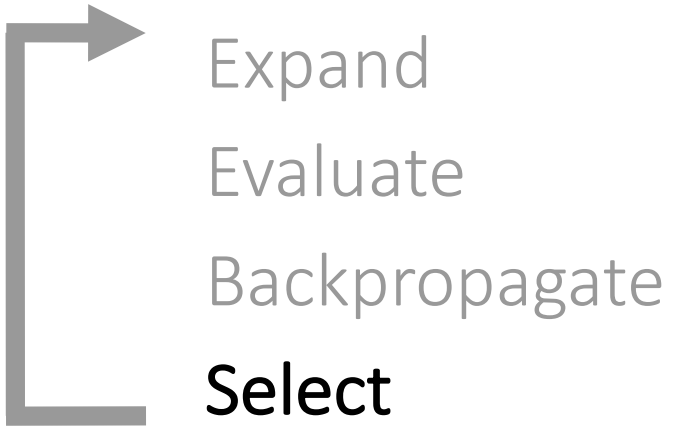

Expand
Evaluate
Backpropagate
Select

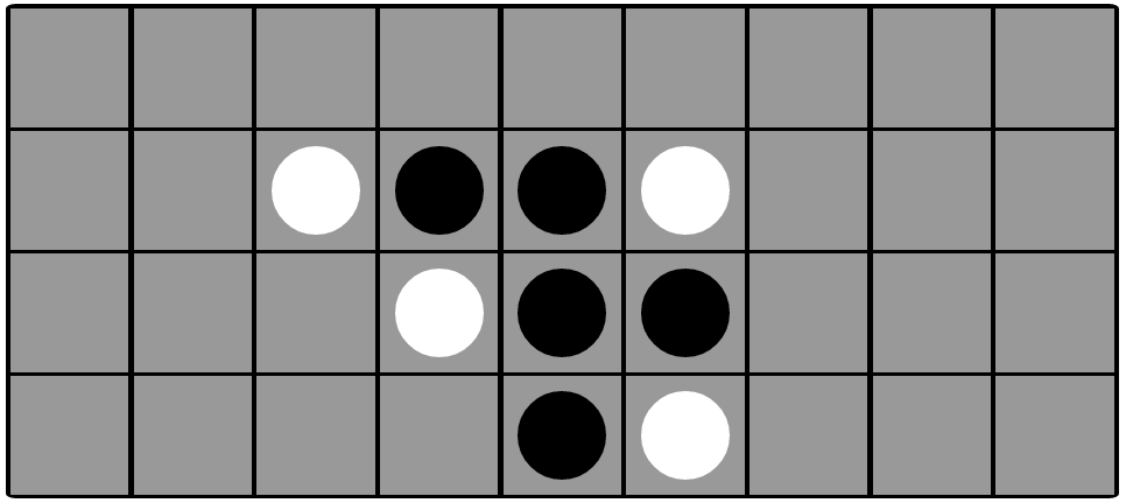

5.2
4.8
0.3

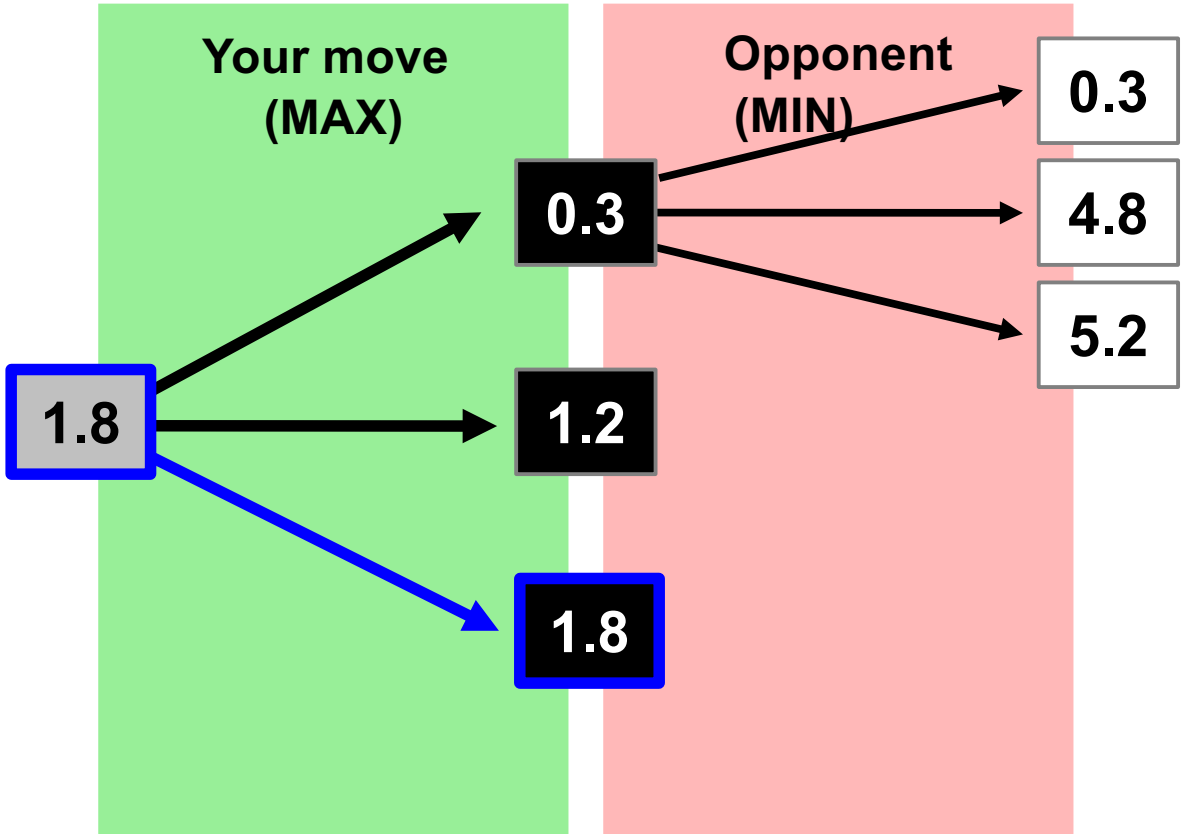

Your move
(MAX)
Opponent
(MIN)
2.3
0.3
1.2
1.8
0.3
4.8
5.2

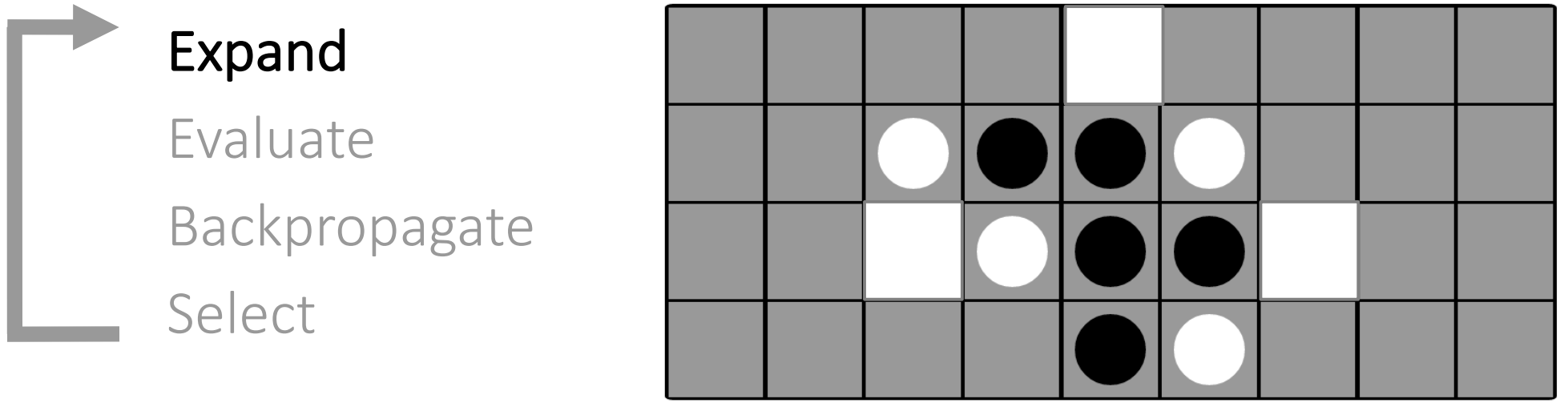

Expand
Evaluate
Backpropagate
Select
0.3
1.2
1.8

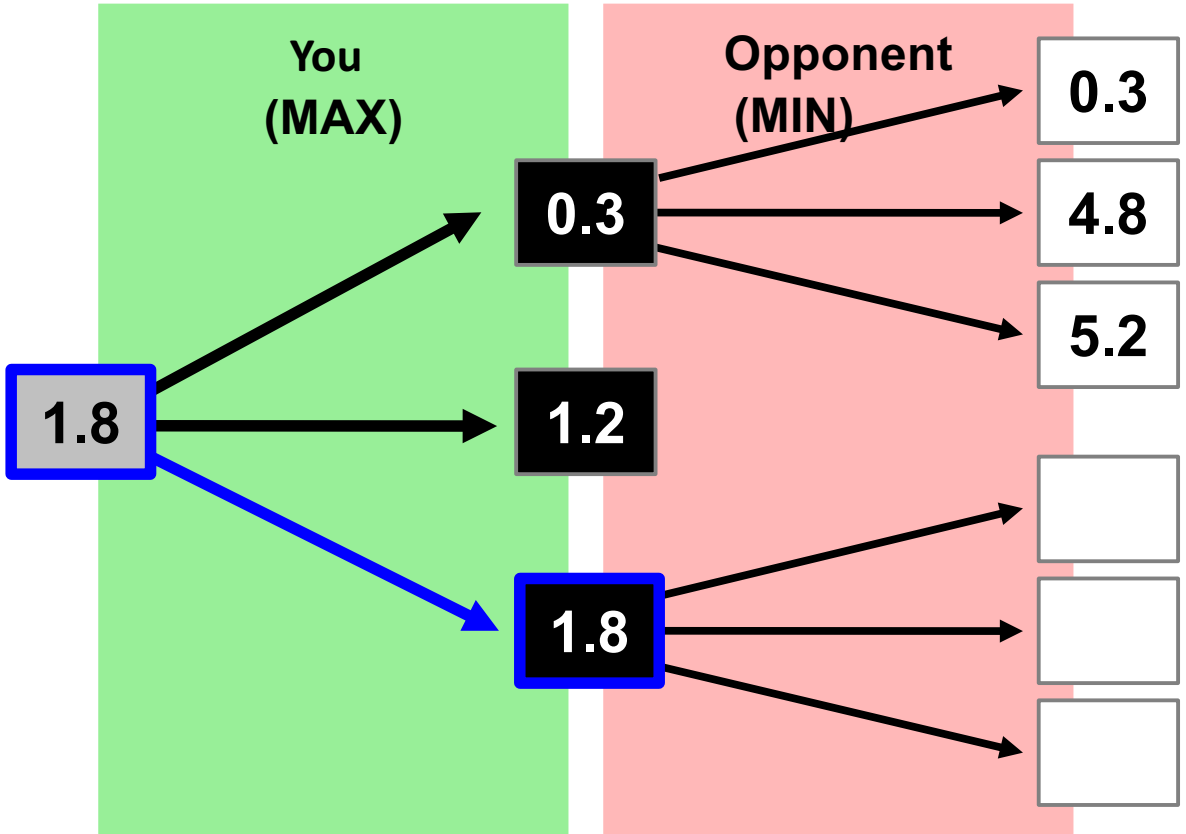

Your move
(MAX)
Opponent
(MIN)
1.8
0.3
1.2
1.8
0.3
4.8
5.2

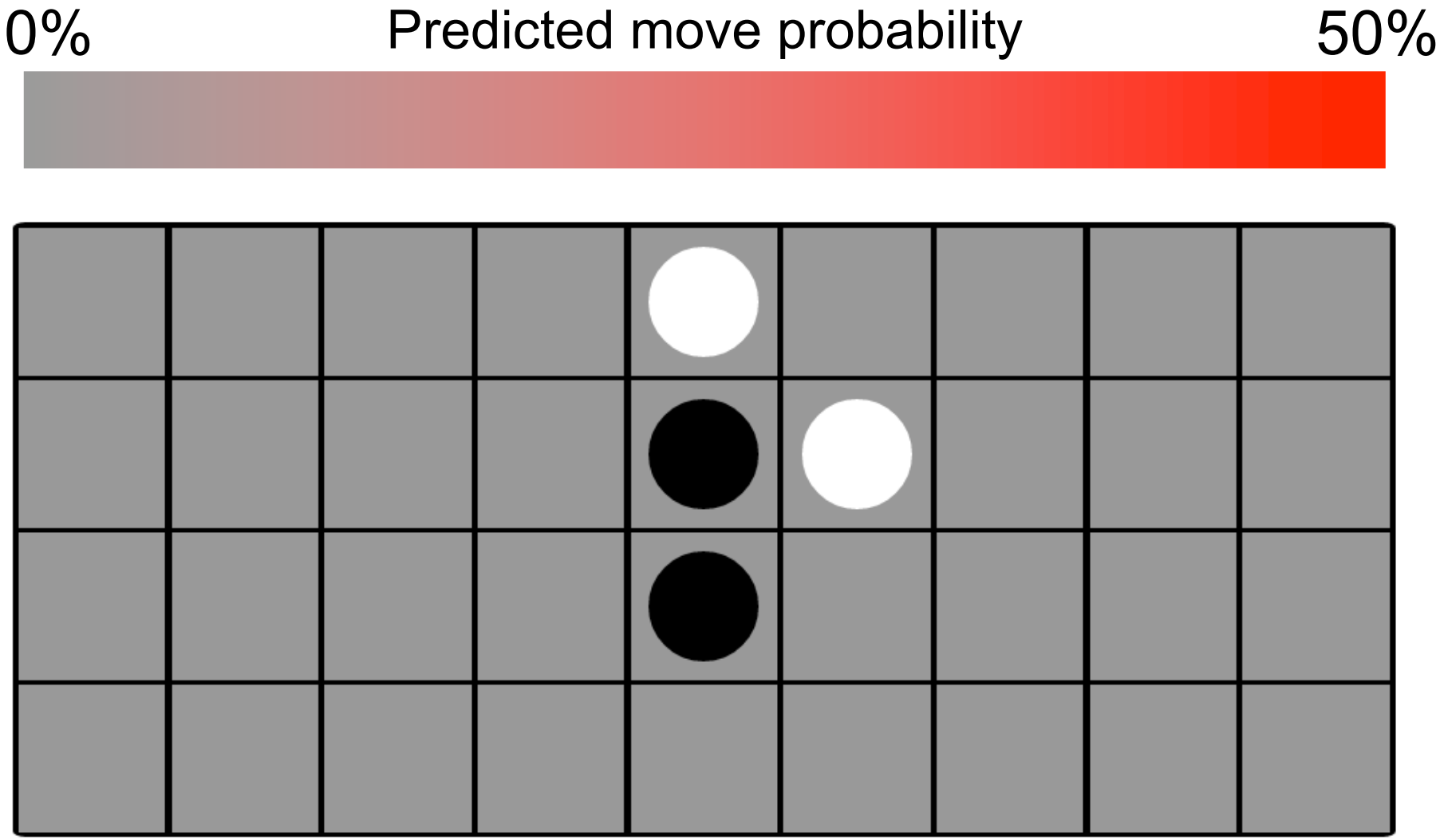

Expand
Evaluate
Backpropagate
Select

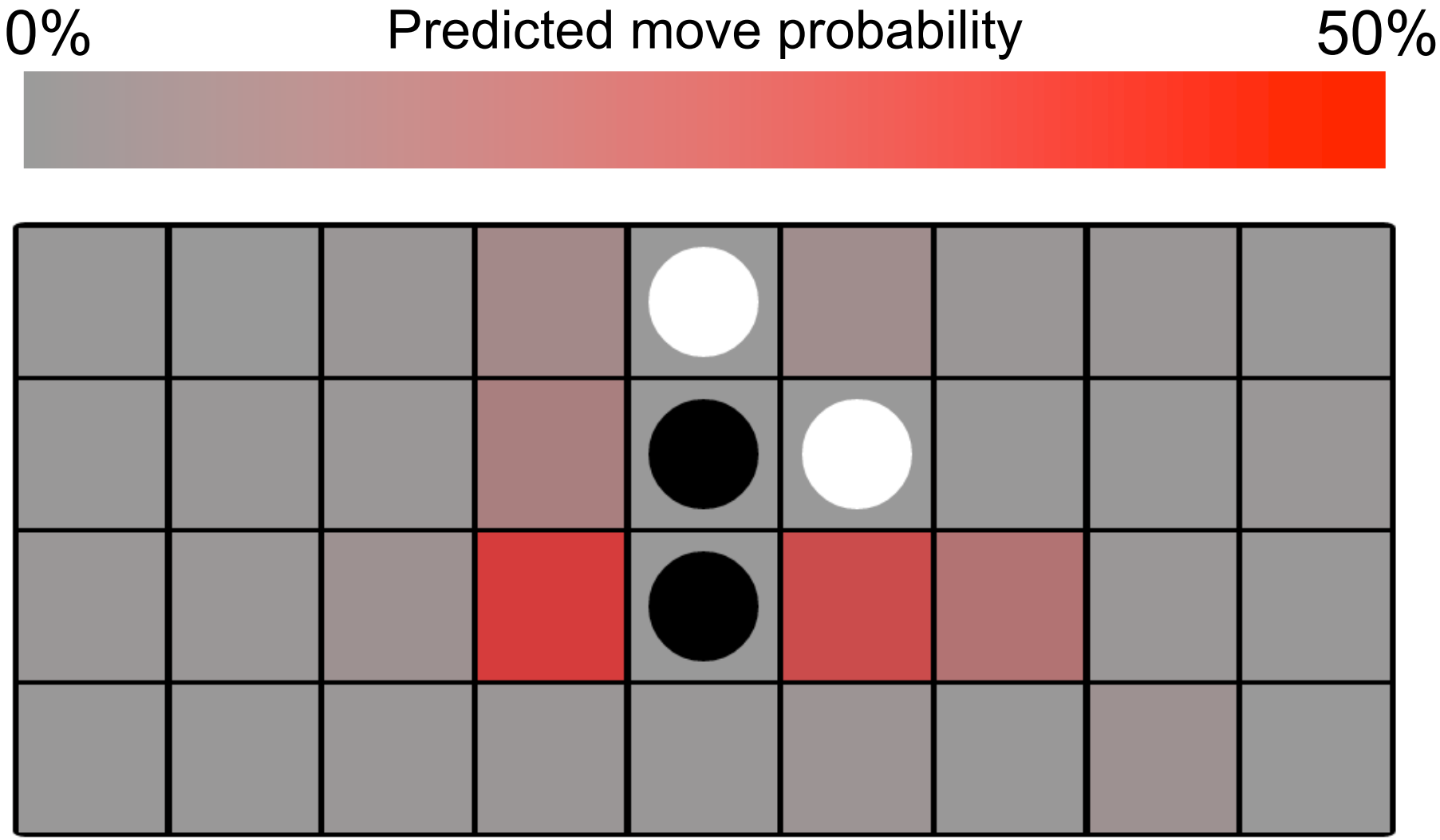


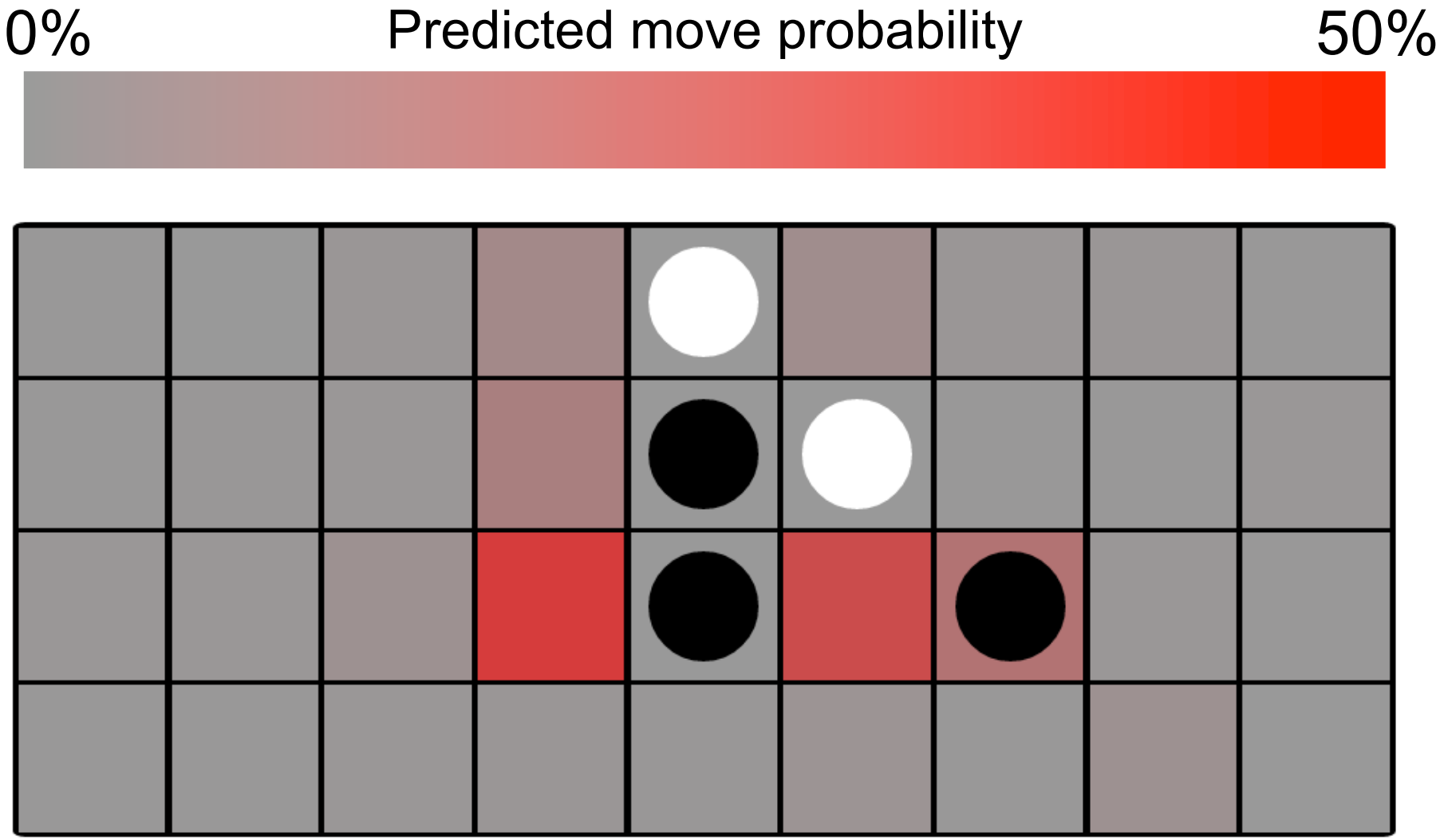

Your move
(MAX)
Opponent
(MIN)
0.3
4.8
5.2
0.3
1.8
1.2
1.8

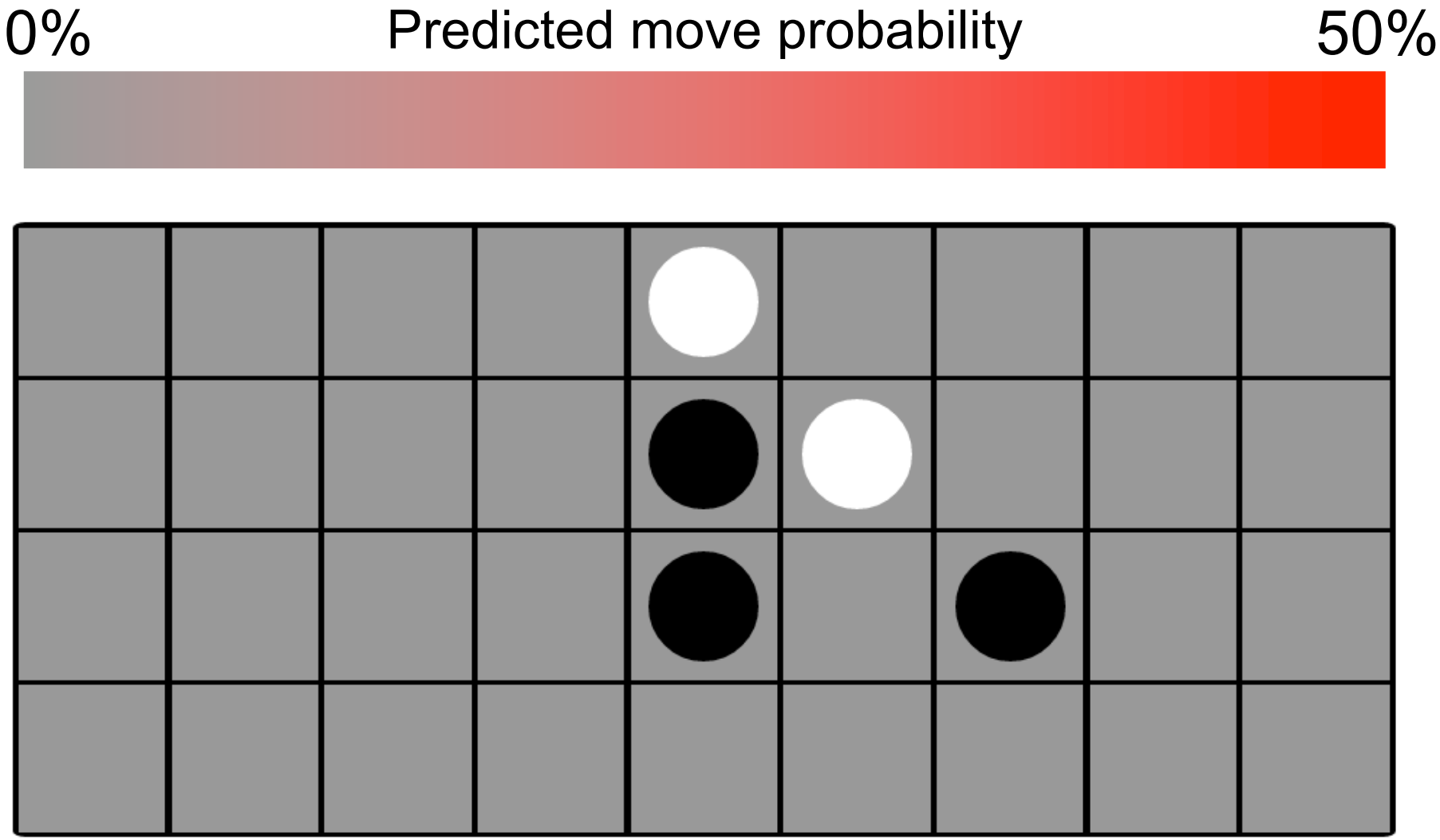

Expand
Evaluate
Backpropagate
Select

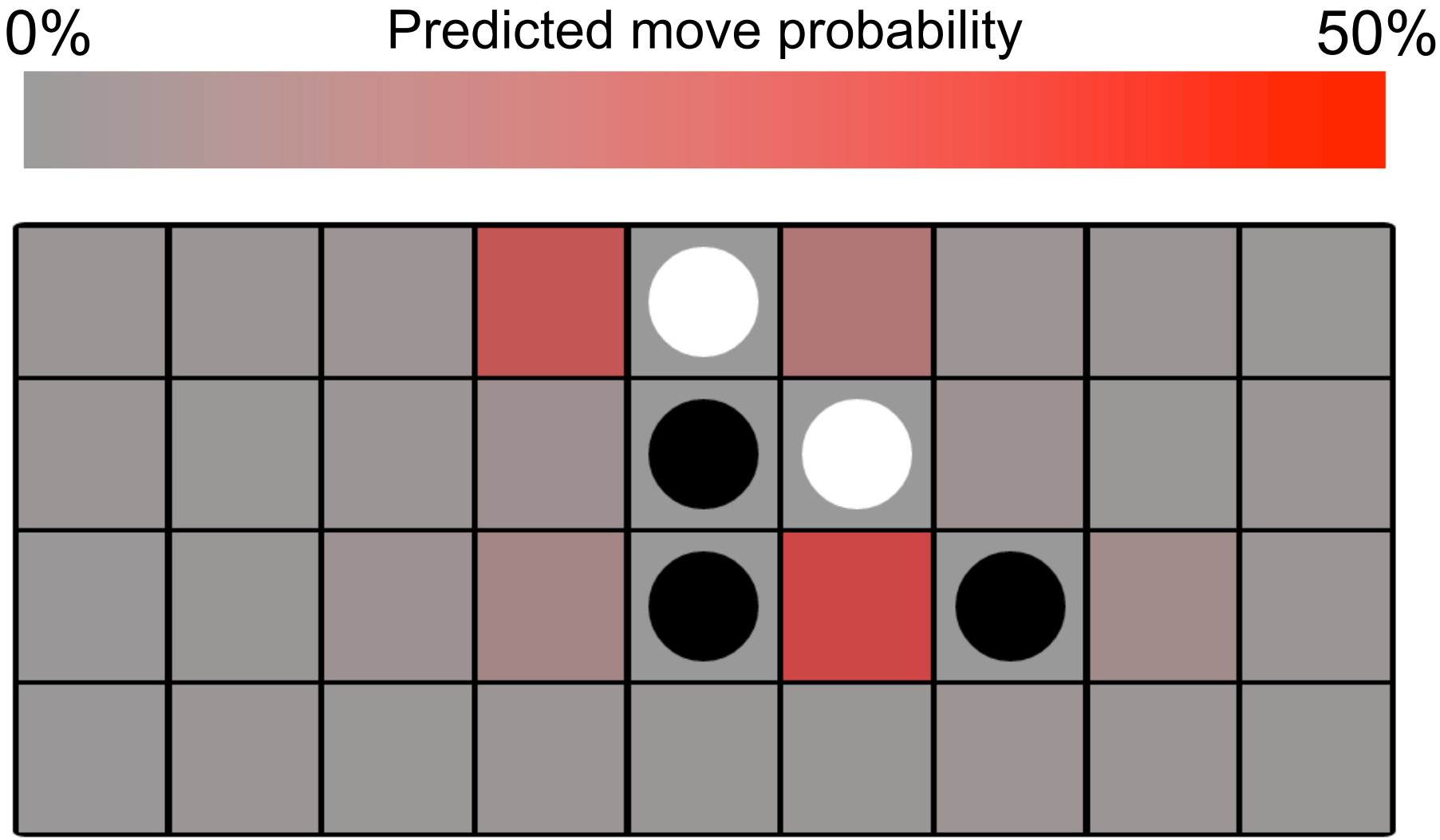

You
(MAX)
Opponent
(MIN)
0.3
4.8
5.2
0.3
1.2
1.8
1.8

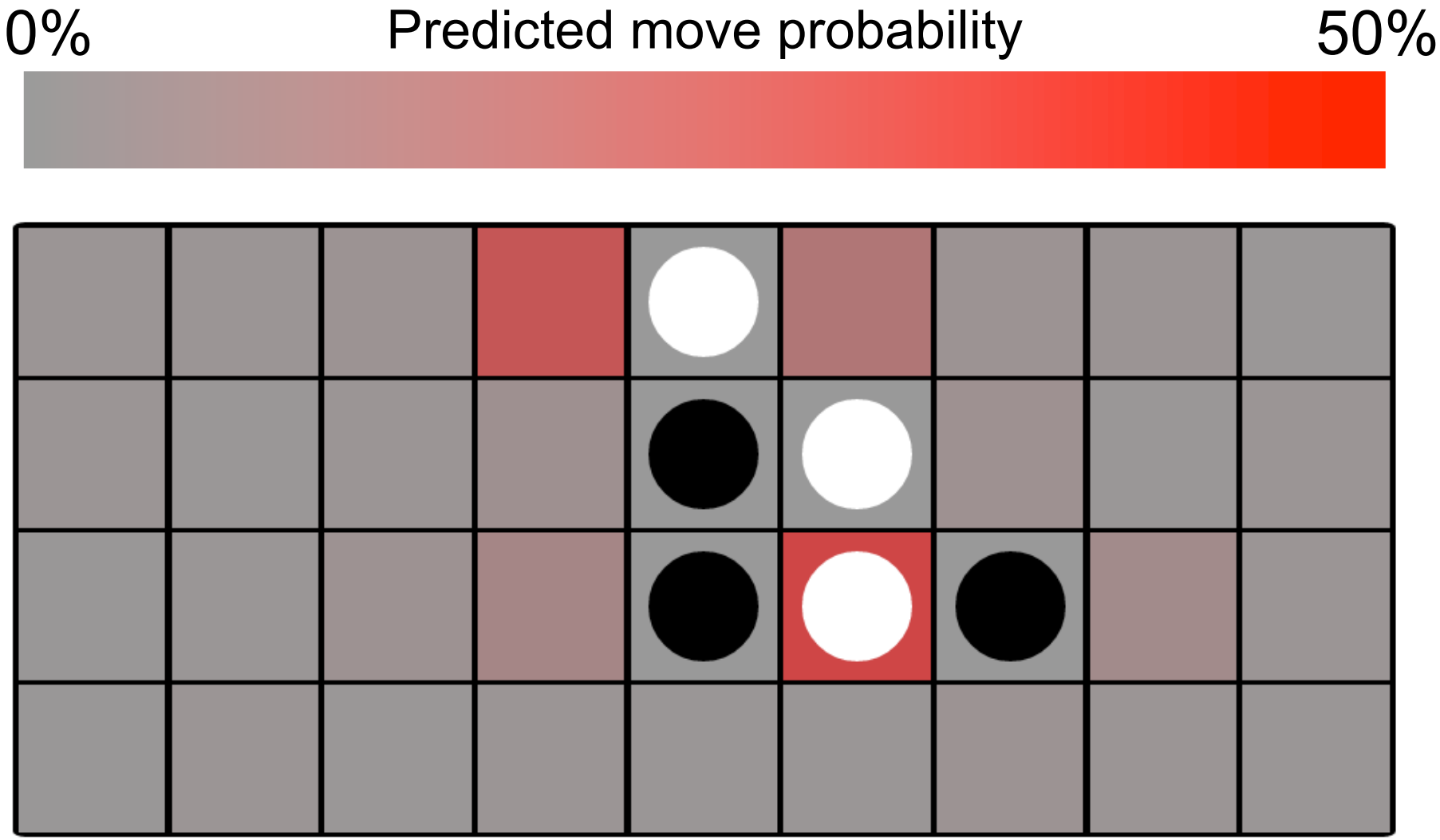

0%
Predicted move probability
50%

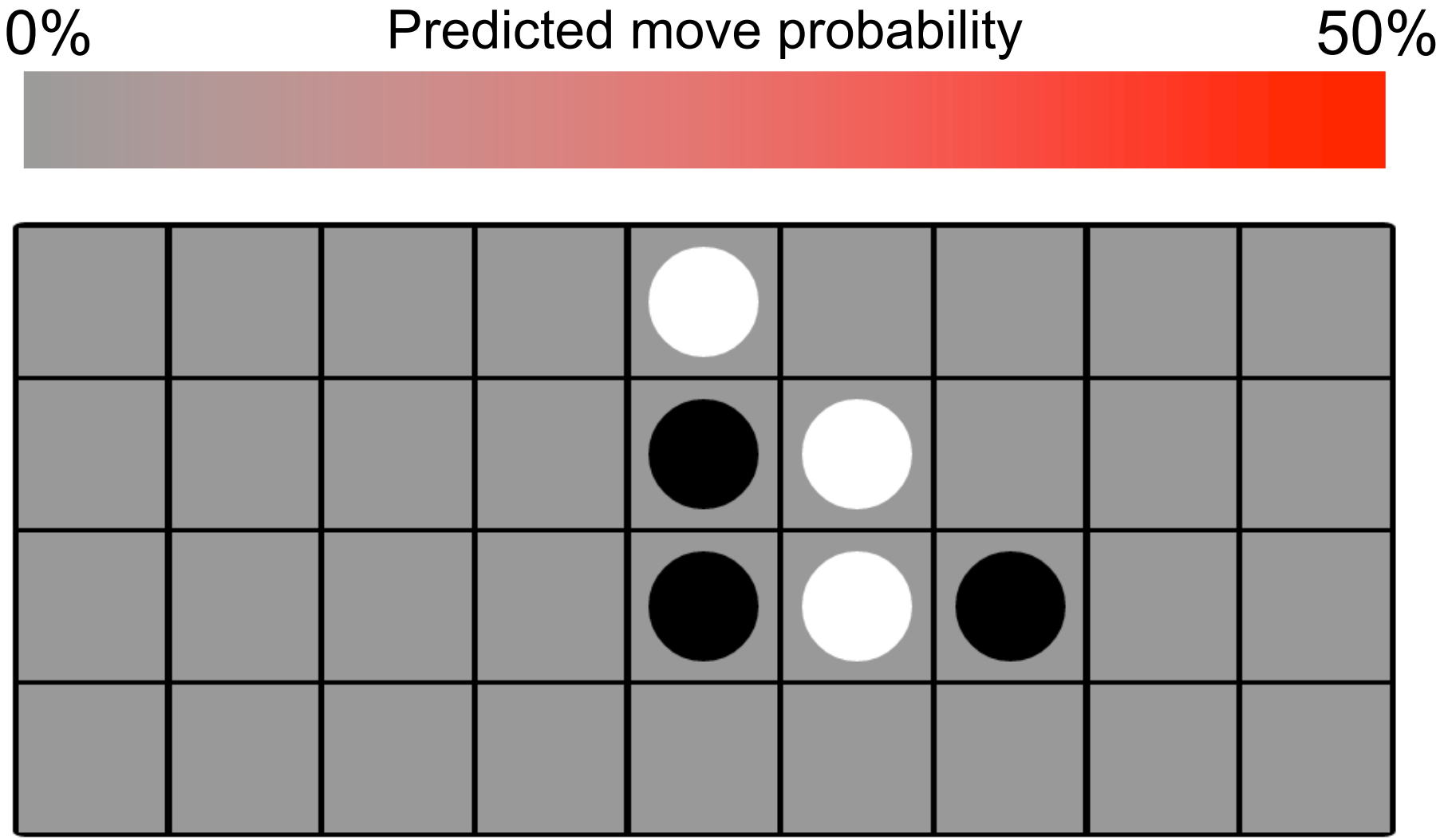

0%
Predicted move probability
50%

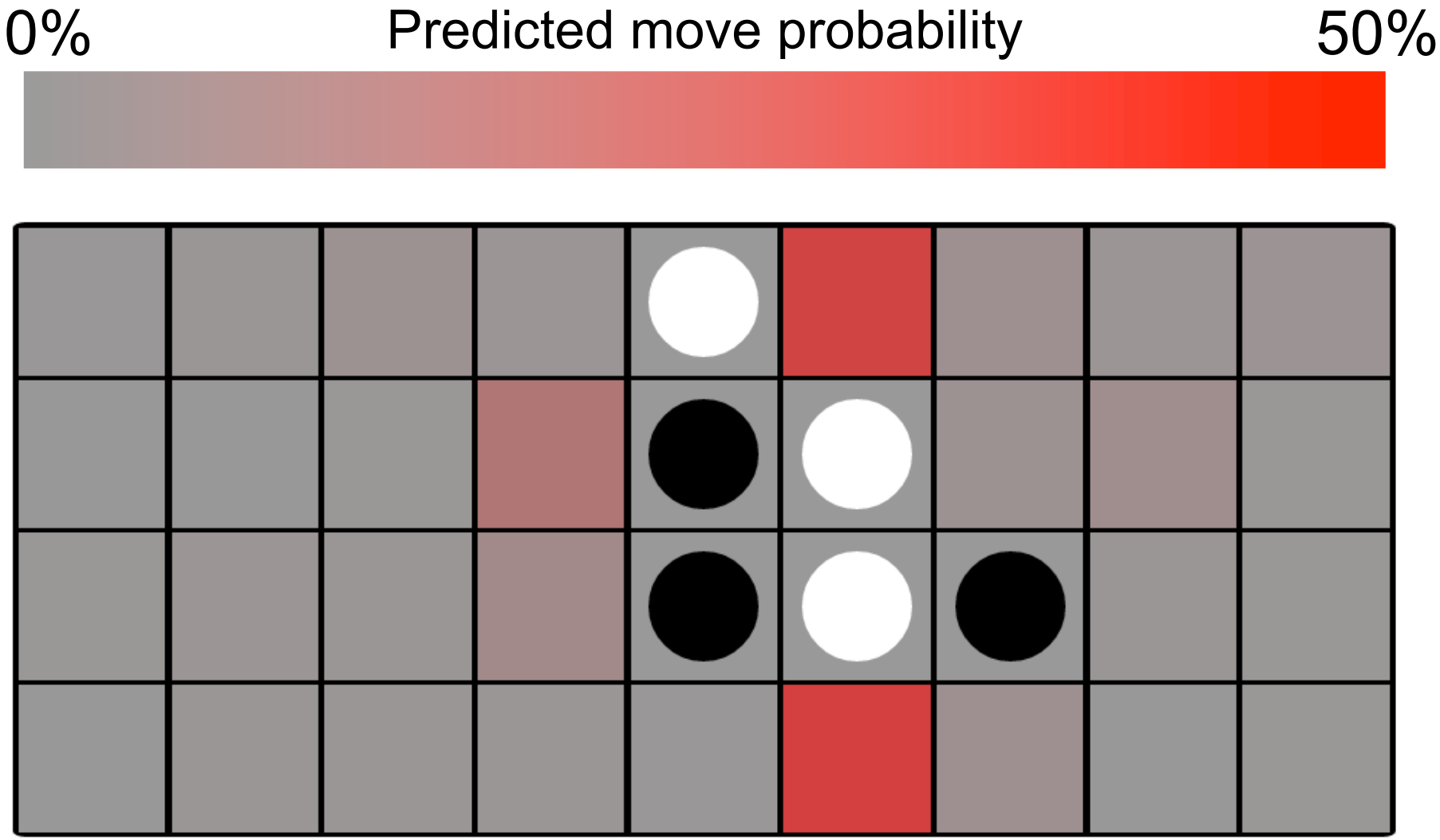

0%
Predicted move probability
50%

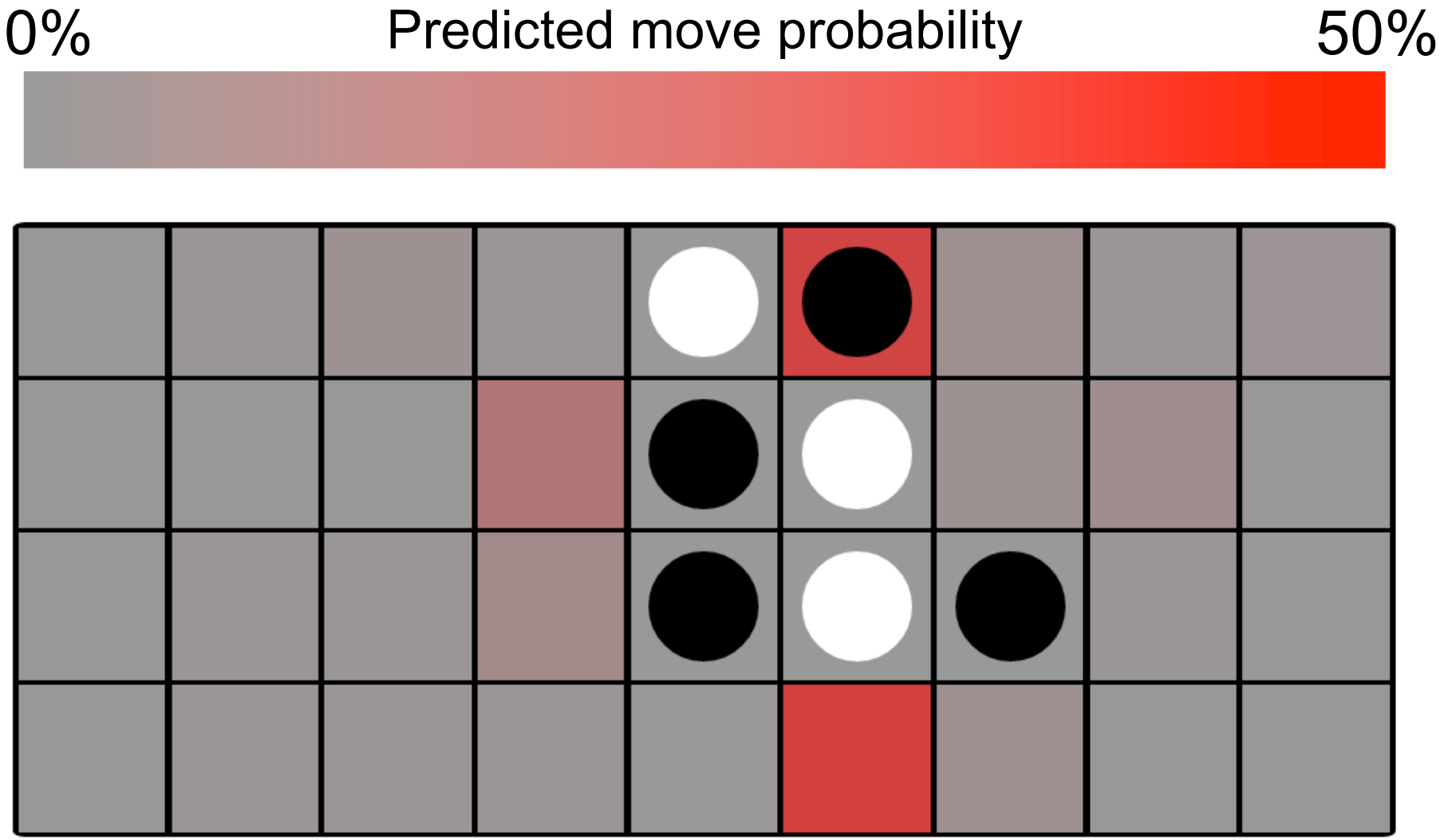

0%
Predicted move probability
50%


0%
Predicted move probability
50%

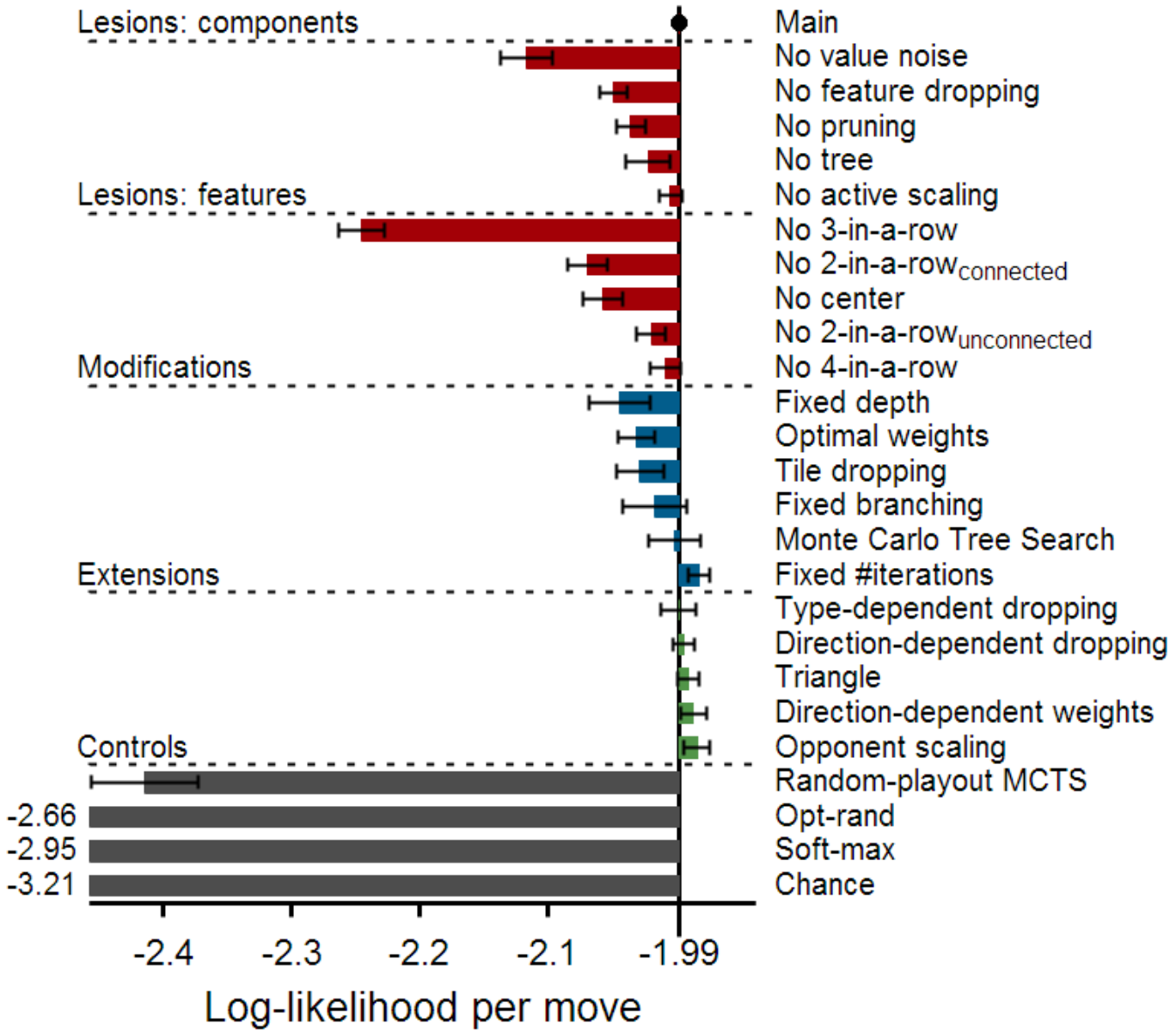

0%
Predicted move probability
50%


0%
Predicted move probability
50%


0%
Predicted move probability
50%


0%
Predicted move probability
50%

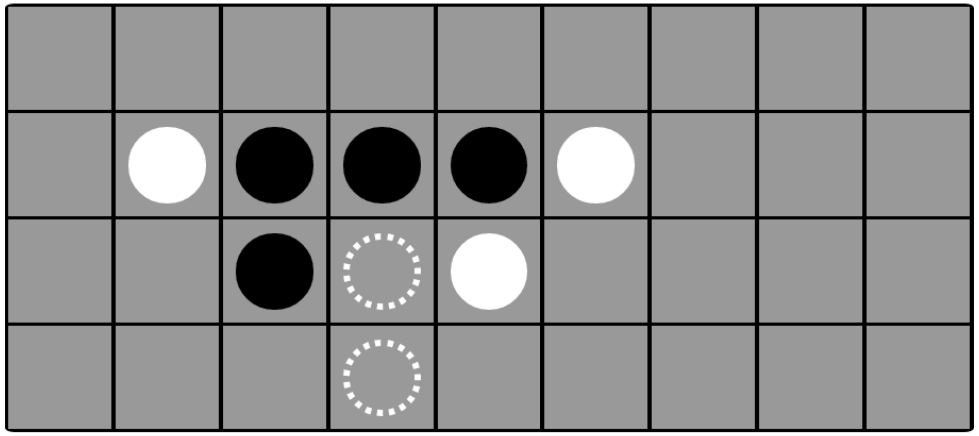
Model Comparison



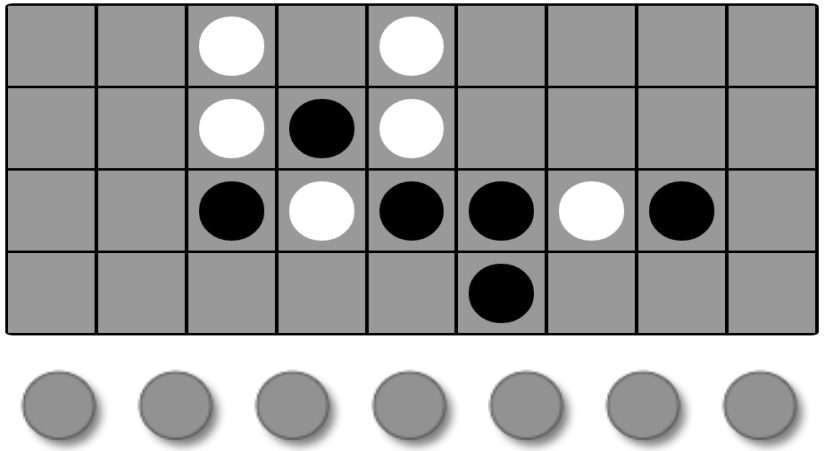
Experiment 2
Generalization

- Can models fitted to a subject's play predict other tasks?
- **Part 1**: Playing against AI agents (30 min)
- **Part 2:** 2AFC (84 trials)
- **Part 3:** Evaluating positions (84 trials)
- 12 subjects

You are playing **WHITE**

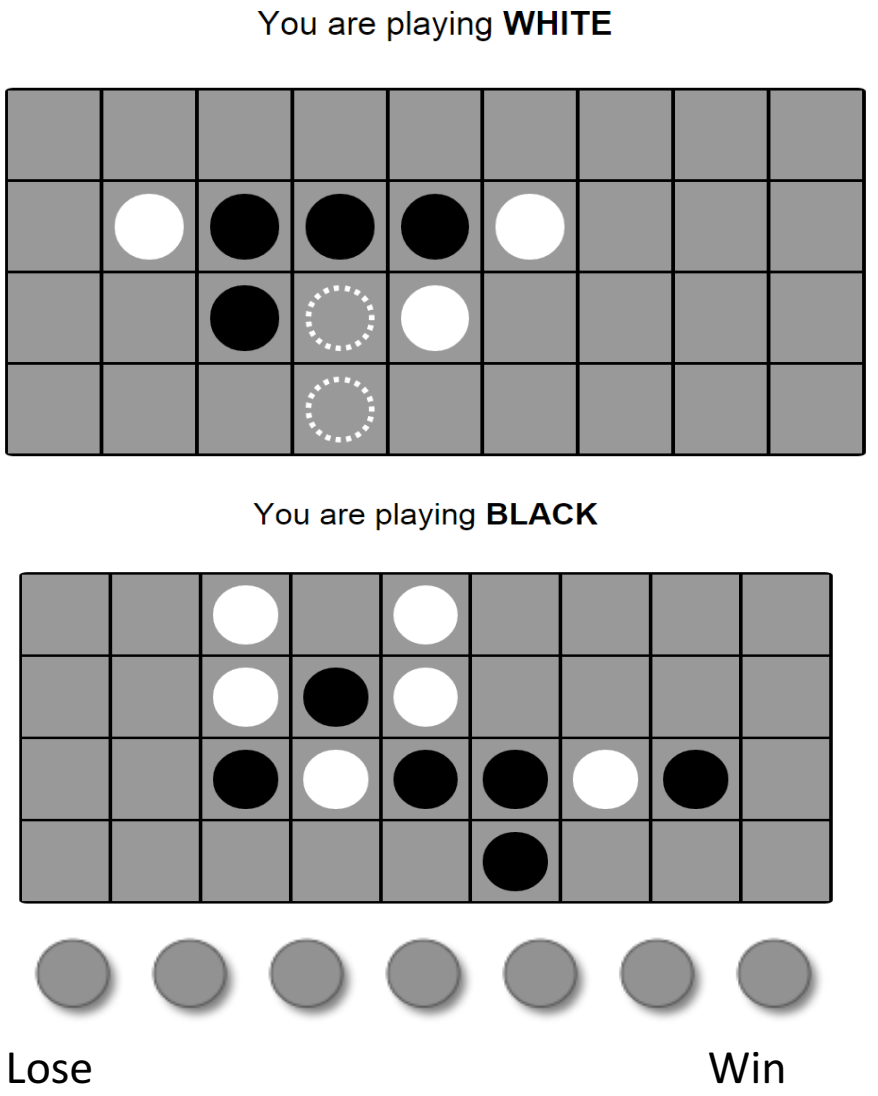


You are playing **BLACK**

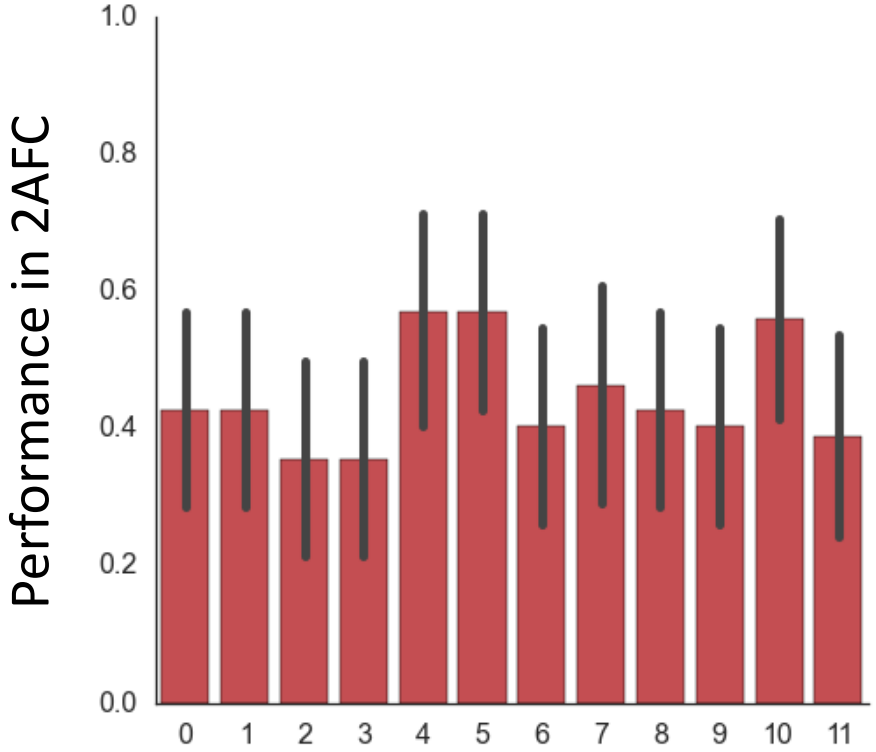


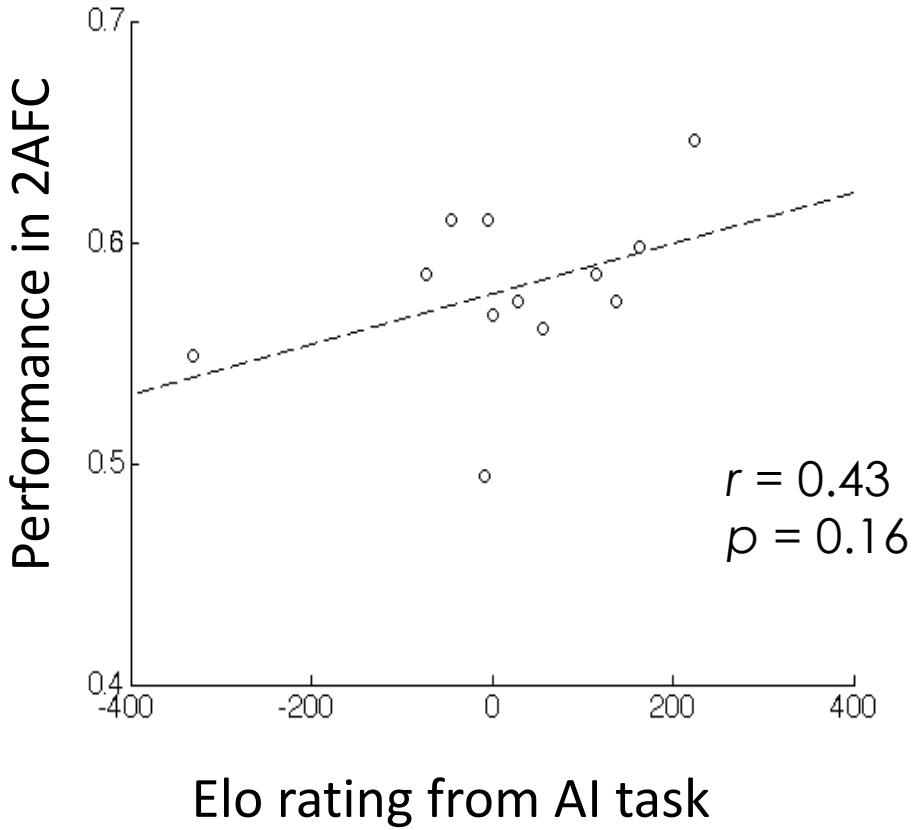
Experiment 2
Generalization

- Can models fitted to a subject's play predict other tasks?
- **Part 1**: Free play (30 min)
- **Part 2:** Two alternatives forced choice (2AFC) (84 trials)
- **Part 3:** Evaluating positions (84 trials)

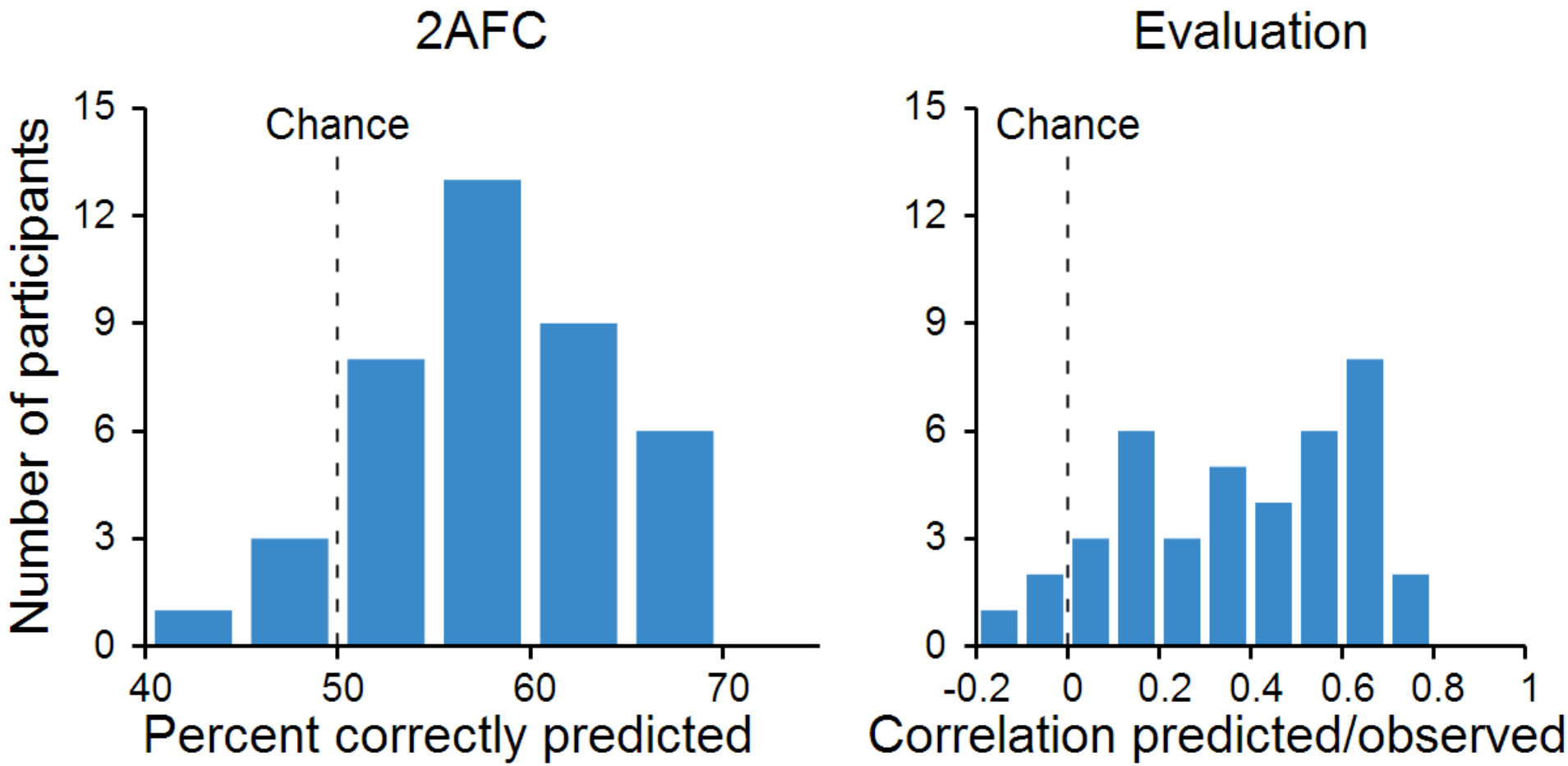


Subject performance in 2AFC task





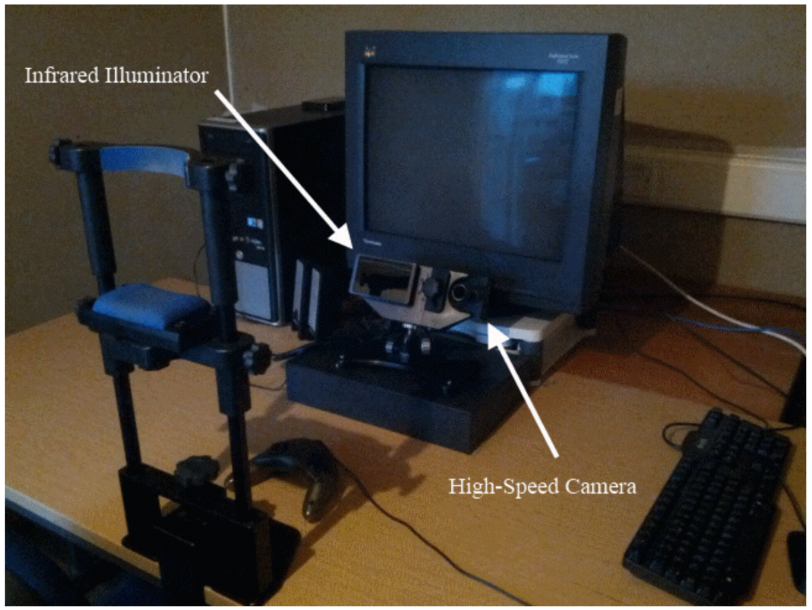
Fit on Games, Predict other Tasks

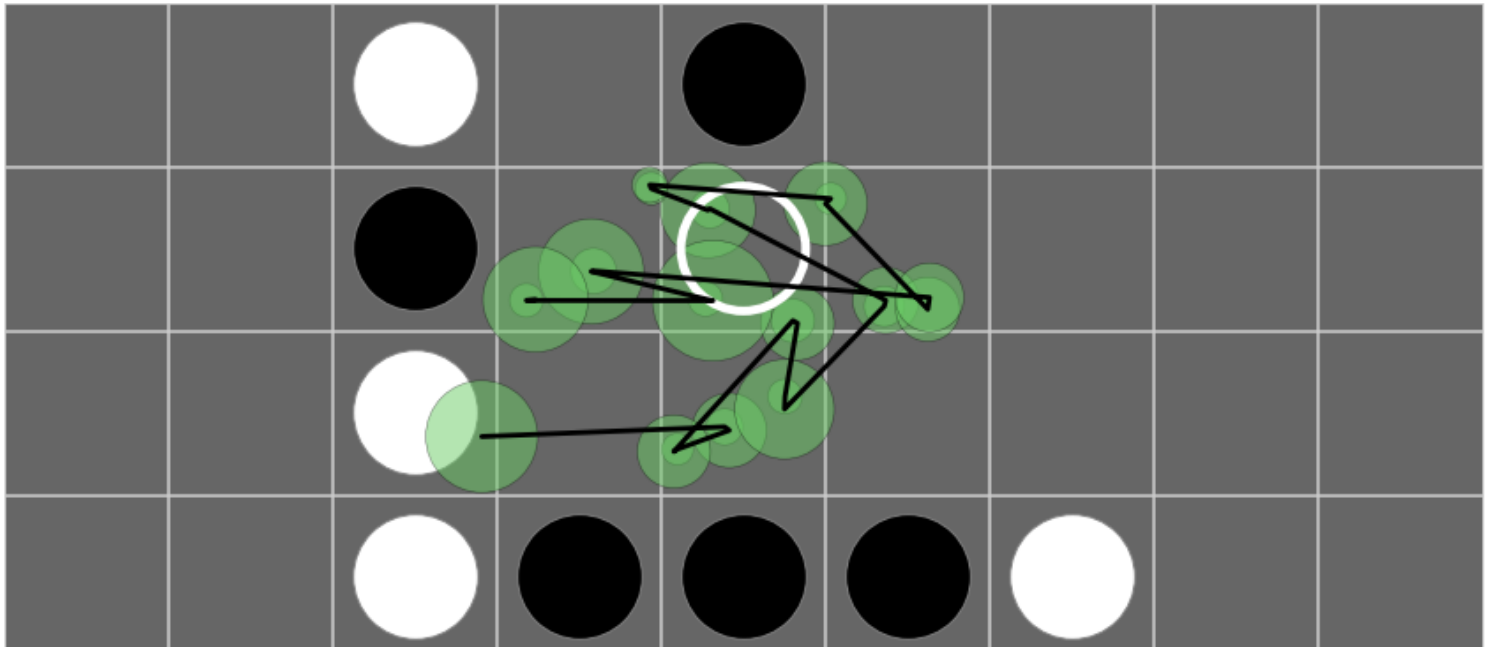


Experiment 3:
eye tracking

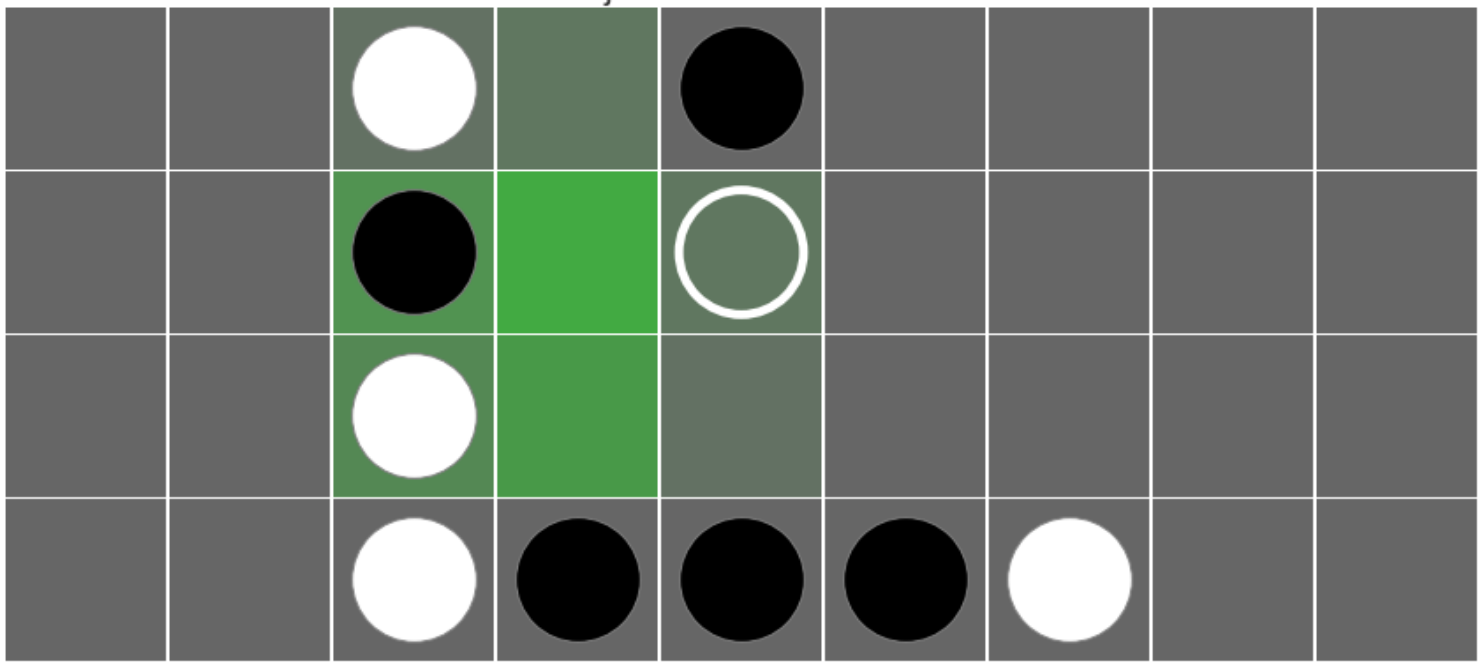
- Play and 2AFC parts from Experiment 2 (total of ~40 minutes)
- 10 subjects
- But with eyetracking (EyeLink II)
- Revisiting *Tichomirov and Poznyanskaya* using the model

Experiment 3: eye tracking

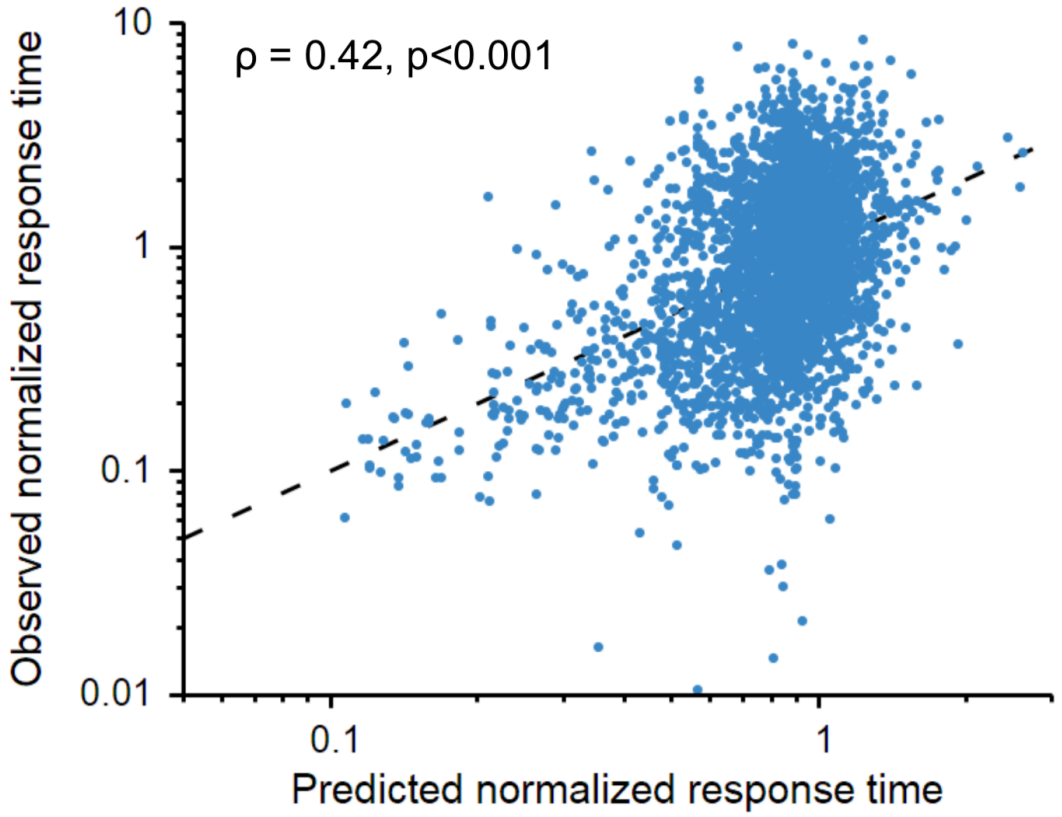




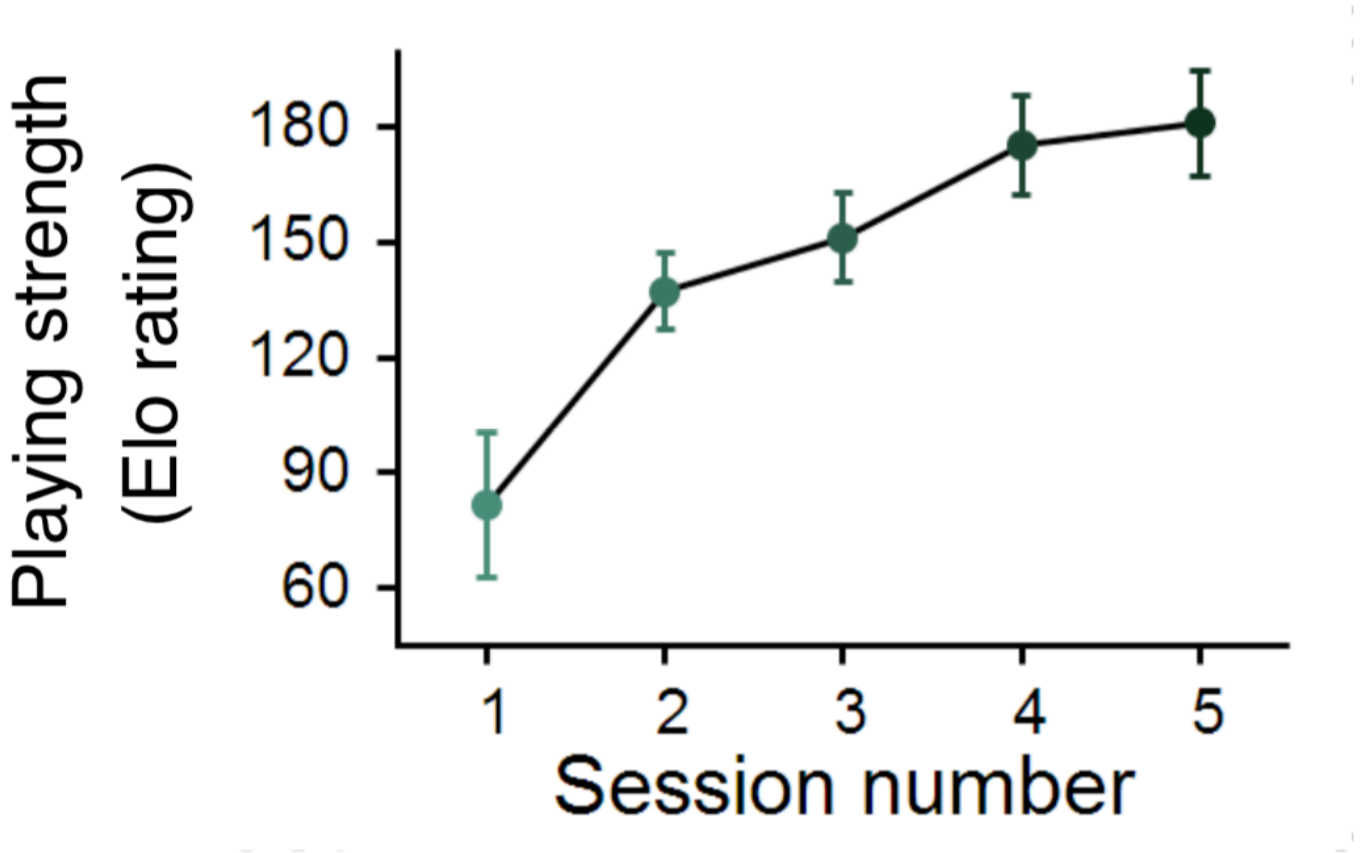
Subject 5 Game 3 Move 9



Predicting Response Times



What Makes a Player Strong?

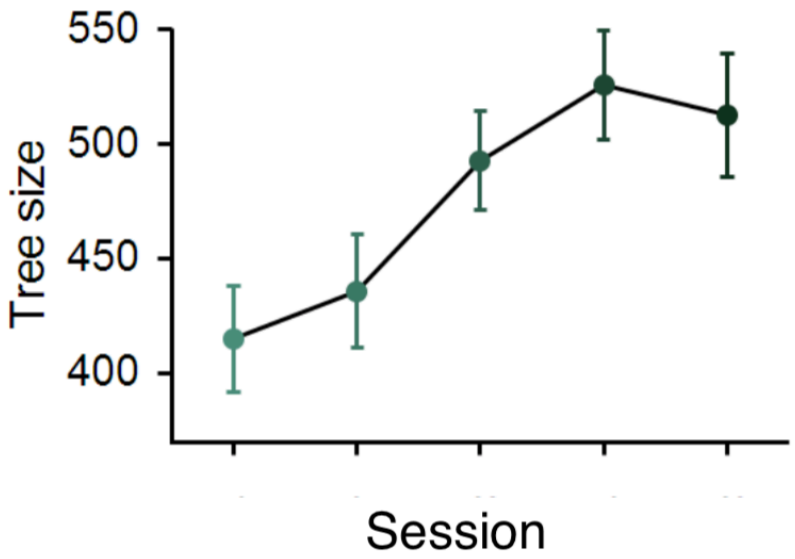


n=30

What Makes a Player Strong?

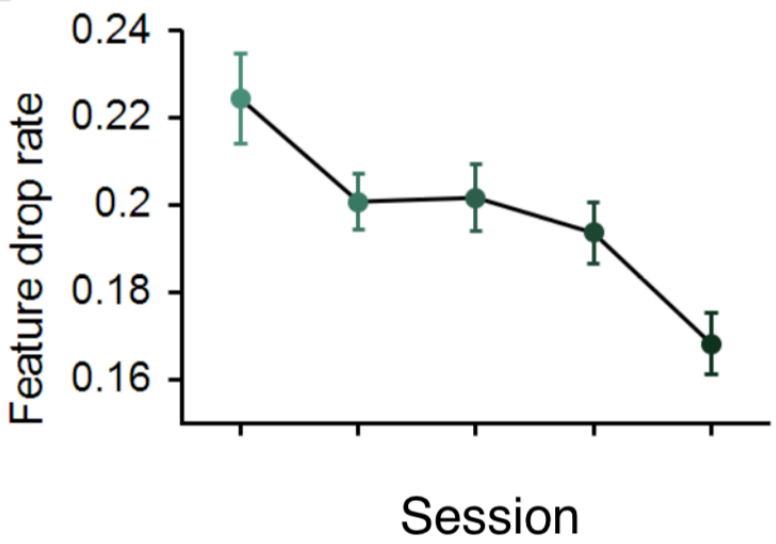
Tree Size

***

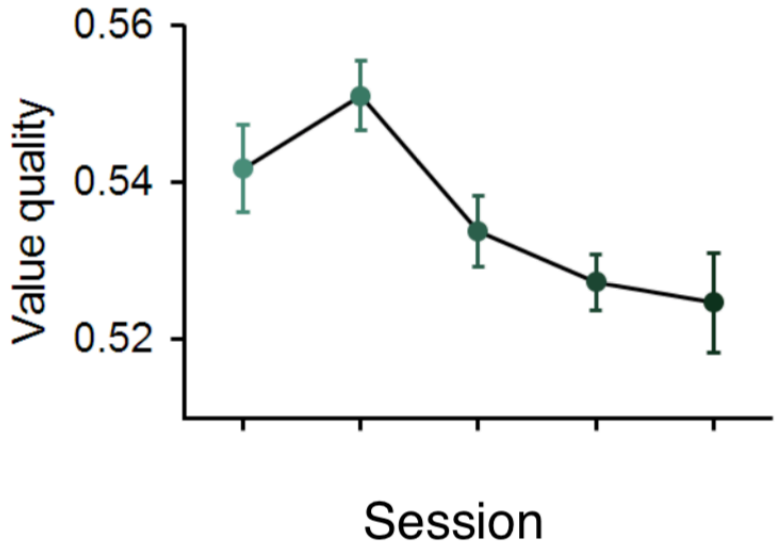


Feature Drop Rate

***



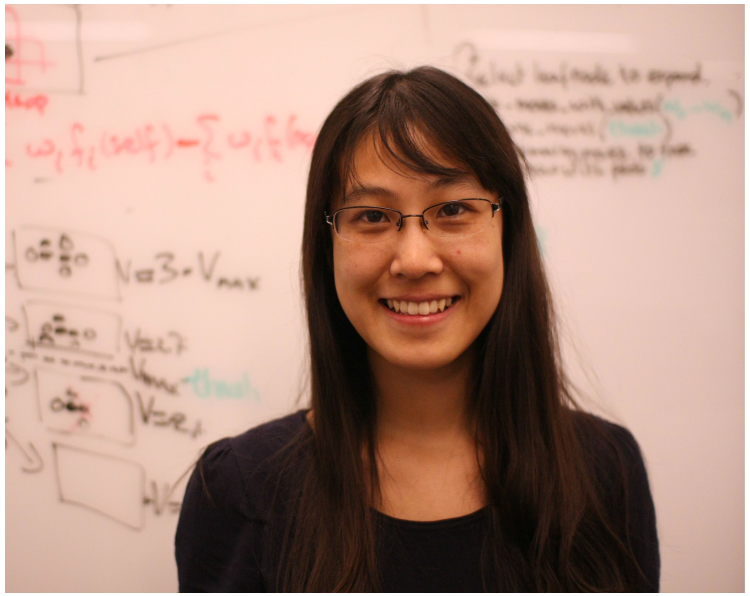
Heuristic Value Quality



What Have We Learned?

- We can model a complex behavior such as playing a game
- Model is able to generalize to novel tasks
- Eye movements correlate with search tree model
- Tree mechanism allows looking ahead
- But the model deals only with a single next action and ignores the long term **plan**

Sources of Sub-optimality in Minimalistic Exploration Exploitation Tasks



Mingyu Song

(Nature Human Behavior(In Revision))

https://www.biorxiv.org/content/early/2018/06/18/348474

Exploration Vs. Exploitation

- Fundamental and common dilemma
  - Where to park the car?
  - Get married or break up?
  - What movie to watch?
  - Keep foraging in this field?

- Actually a family of problems /processes (Cohen et al 2007)
- Might involve multiple neural mechanisms (Yu & Dayan 2005)

Experiment

Common complexities	Our task
Indefinite horizon	Number of days is known
Aggregation of rewards – memory limitations	Reward history is constantly shown on screen
Reward distribution drifts	Stationary environment
Exploration and exploitation are intermixed.	No exploration on exploitations
Both “expected” and “unexpected” uncertainty	Guaranteed rewards and known distributions

(see also Cohen 2007)

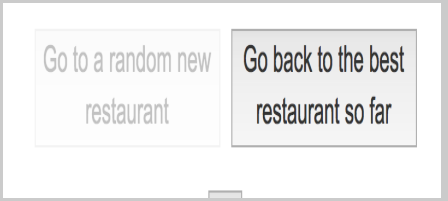
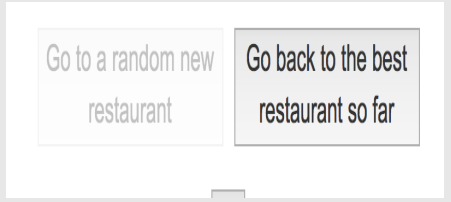
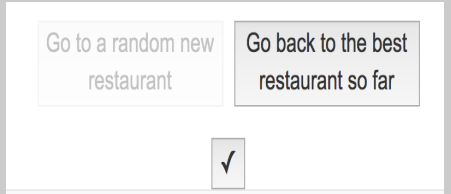
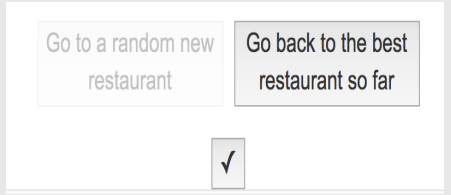
(see also Sang, Todd and Goldstone, 2011)

Additional Exploration/Exploitation complexities

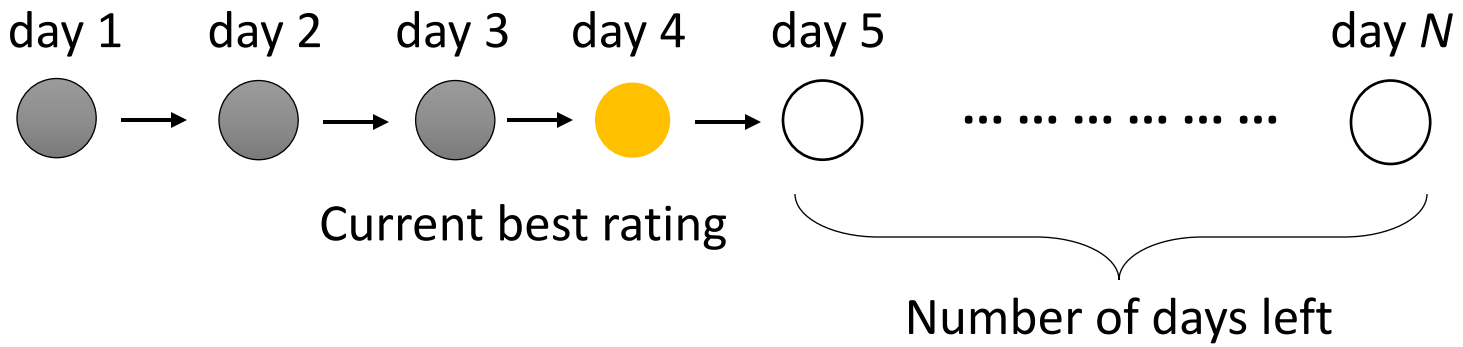
- Keeping track of rewards (0.4, 0.6, 0.1, ….. , 0.9)
  - Aggregation
  - Memory
- Non-stationary environments (e.g. Daw et al. 2006, Yu and Dayan 2005)
  - Anticipating changes
- Indefinite decision horizon (e.g. Acuna et al. 2008)
  - Estimating the number of potential trials
- Action costs
  - Incorporating costs to the computation
- Unknown reward distributions (e.g., Lee et al. 2011, Meyer et al. 1995 and most others)
  - Have to learn parameters

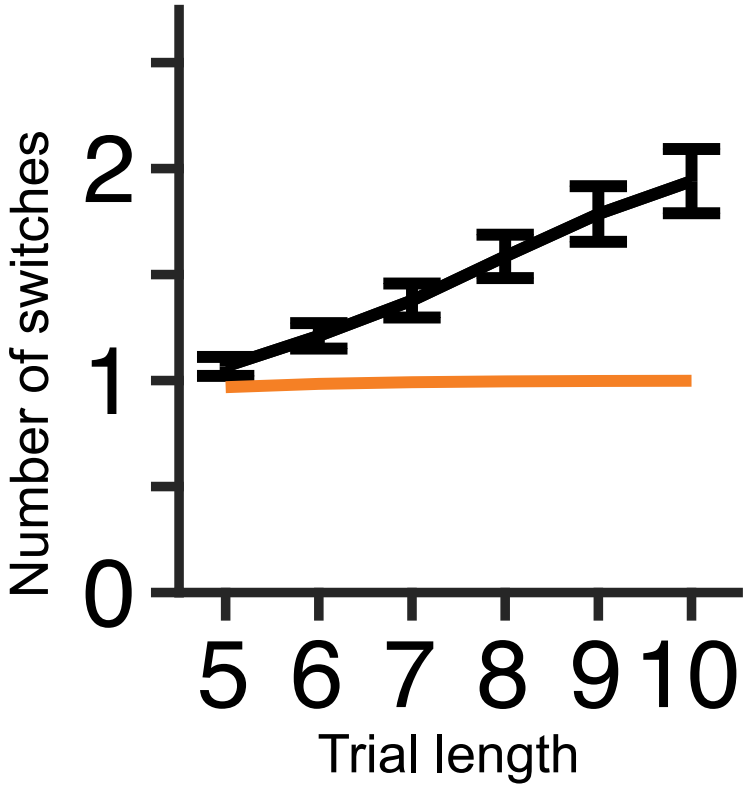
- Known number of days
- Distribution is known
  - No aggregation required
- Reward history is constantly shown on screen
  - No memory required
- Stationary environment
- No **exploration** when **exploiting**.
- Can always go back to best reward so far **(safe return)**

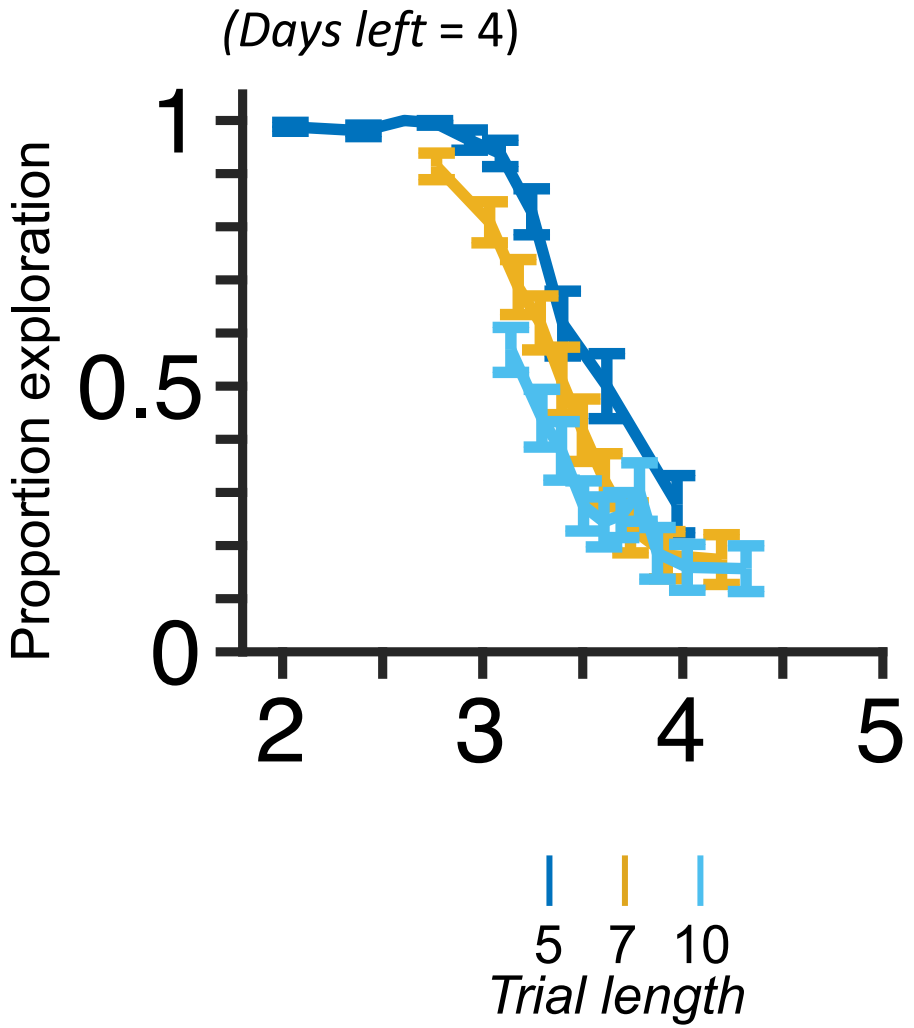
Experiments

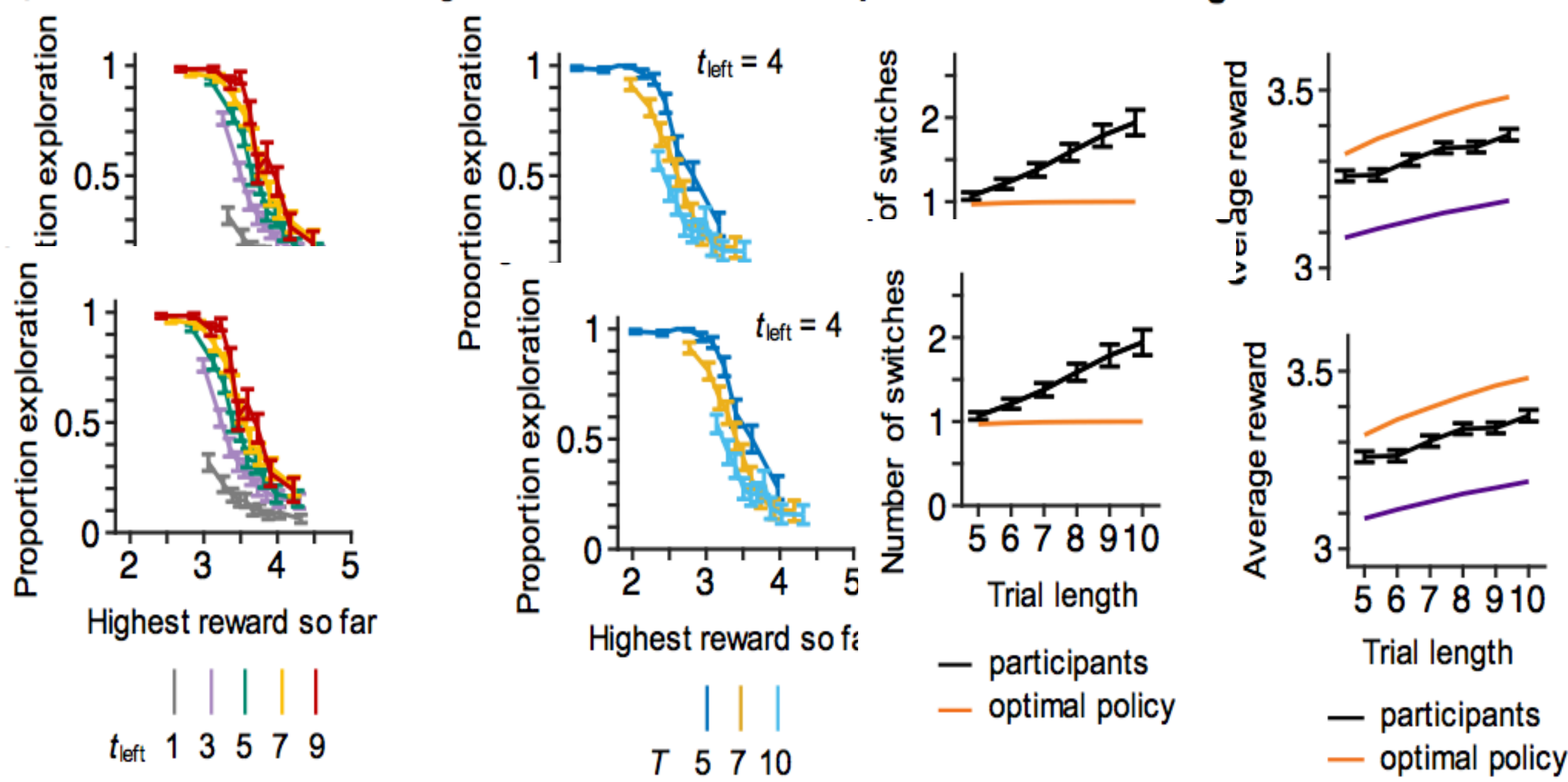
Conditions	Subjects	Trials per subject	Compensation	Setup
Lab	16	180	$10/hour. plus performance bonus up to $5	
Amazon Mechanical Turk (Mturk)	109	60	$1 plus performance bonus up to $4	
Lab	49	180	$10 per hour plus performance bonus up to $5	
Amazon Mechanical Turk	143	60	$1.5 plus performance bonus up to $5	

Information needed for optimal behavior

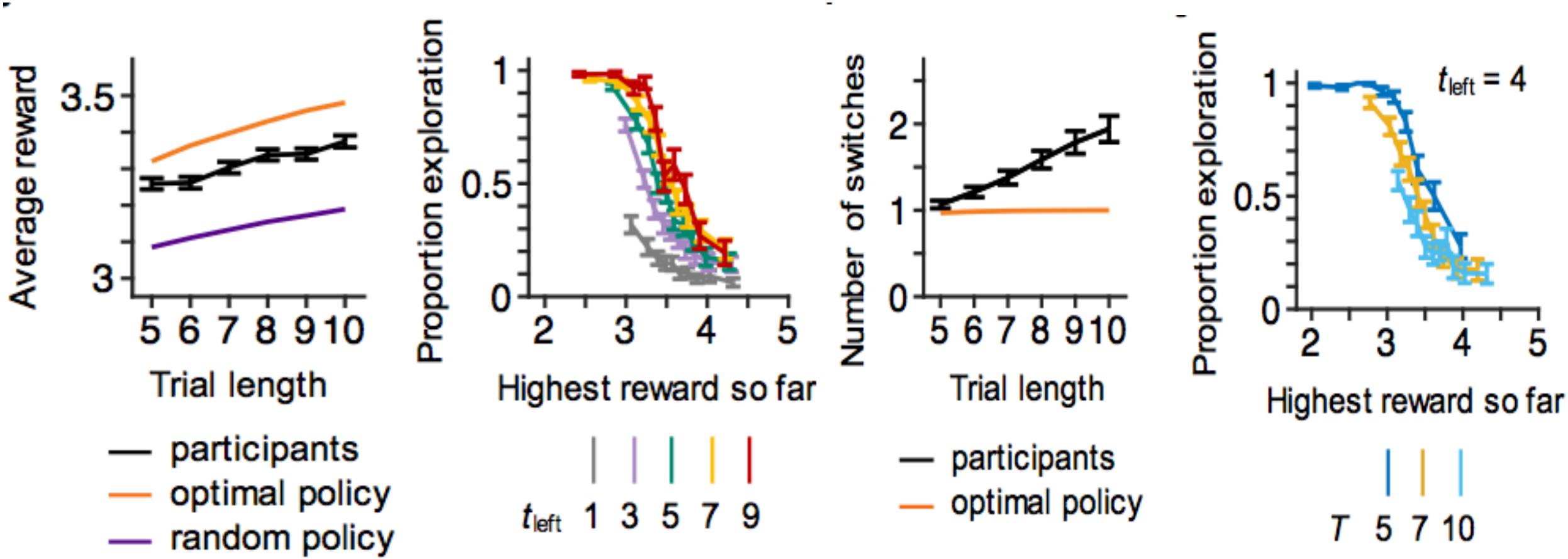



Number of switches
0
1
2
5 6 7 8 9 10
Trial length


(Days left = 4)
Proportion exploration
0
0.5
1
2 3 4 5
5 7 10
Trial length


tion exploration
1
0.5
Proportion exploration
1
0.5
0
2
3
4
5
Highest reward so far
t_{left} 1 3 5 7 9
Proportion exploration
1
0.5
t_{left} = 4
Proportion exploration
1
0.5
0
2
3
4
5
Highest reward so fa
t_{left} = 4
T 5 7 10
of switches
2
1
Number of switches
2
1
0
5 6 7 8 9 10
Trial length
participants
optimal policy
age reward
3.5
3
Average reward
3.5
3
5 6 7 8 9 10
Trial length
participants
optimal policy
random policy

TODO: Get better resolution images



"We are correct most of the time, but the mistakes we make are interesting."

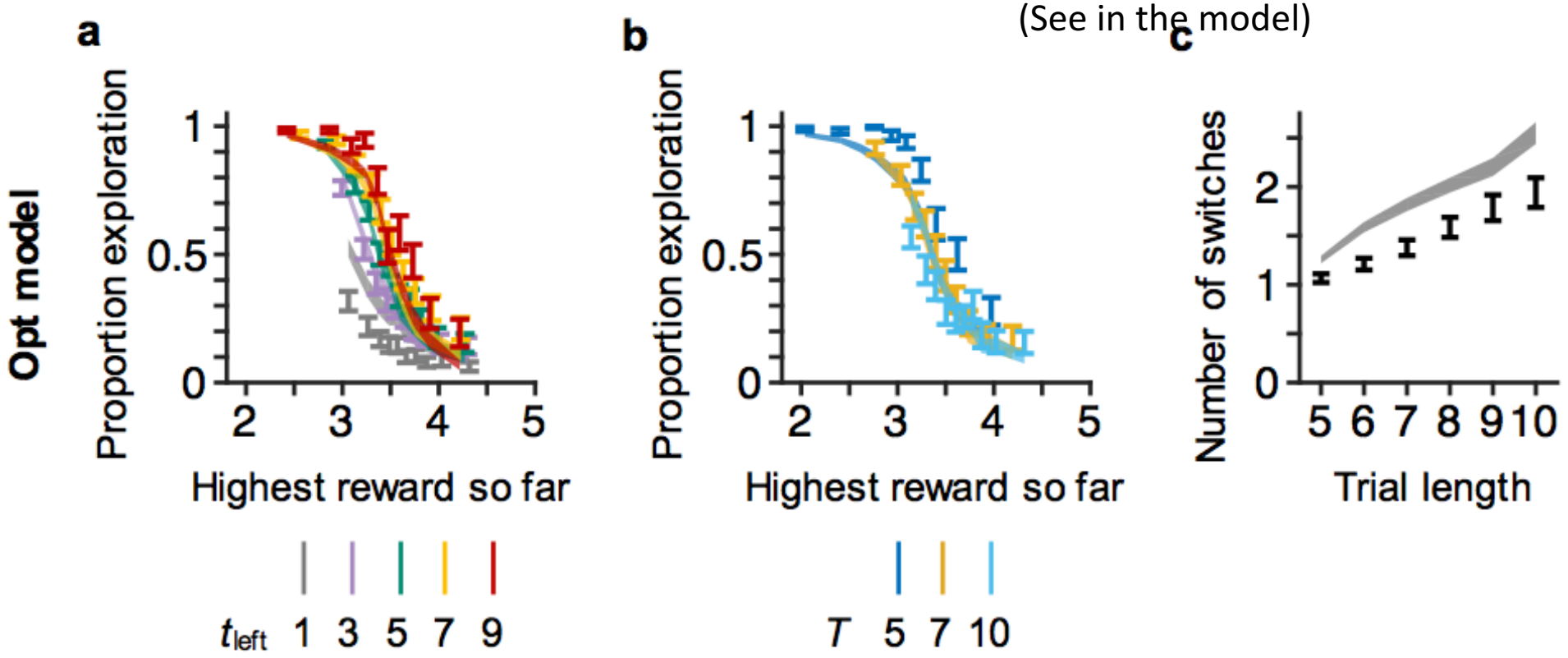
Daniel Kahneman

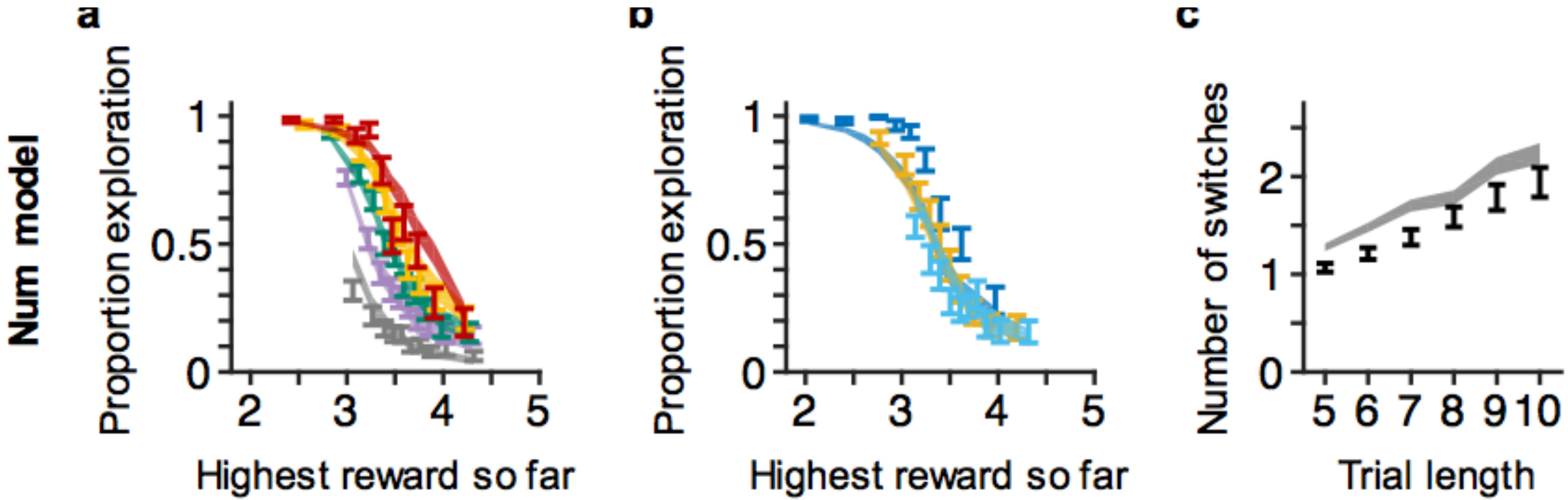
Models

- Optimal strategy with decision noise and bias.
- Threshold models
  - *Num* – Exploit when *highest observed reward* > Threshold(days left)
  - *Prop* – Exploit when *highest observed reward* > Threshold(days left/Trial length)
  - *Prop-V* – Adding trial-level variability.

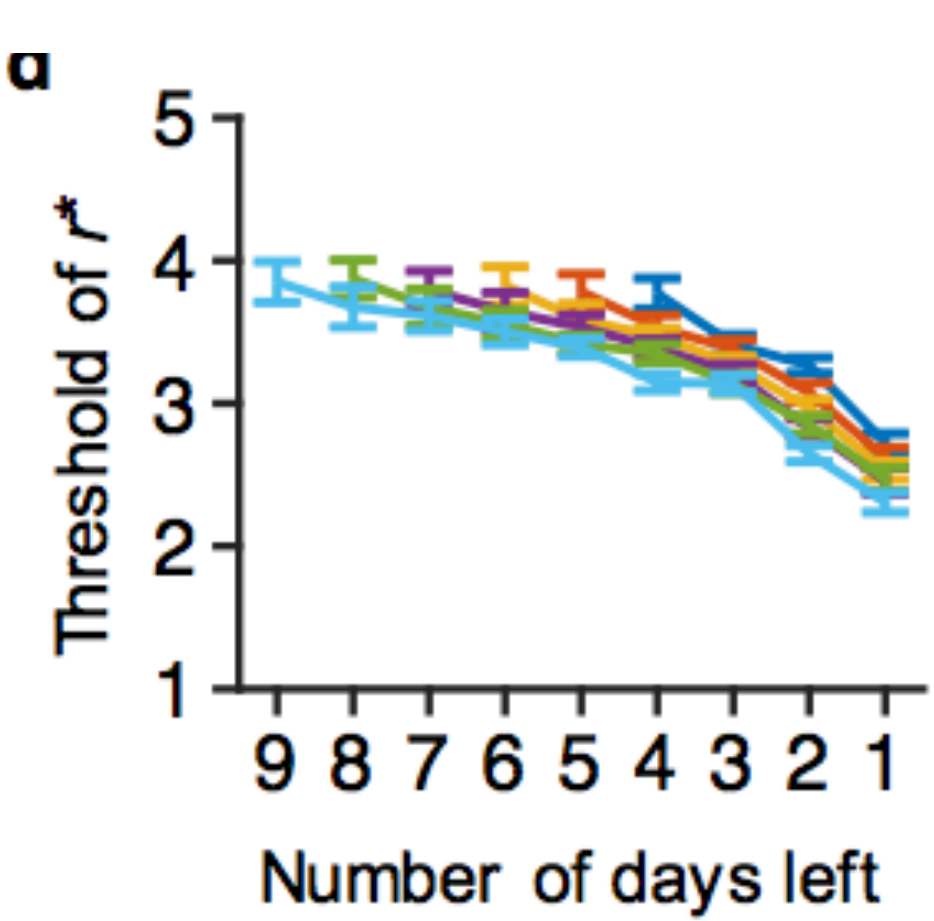
Optimal Solution with decision noise and bias

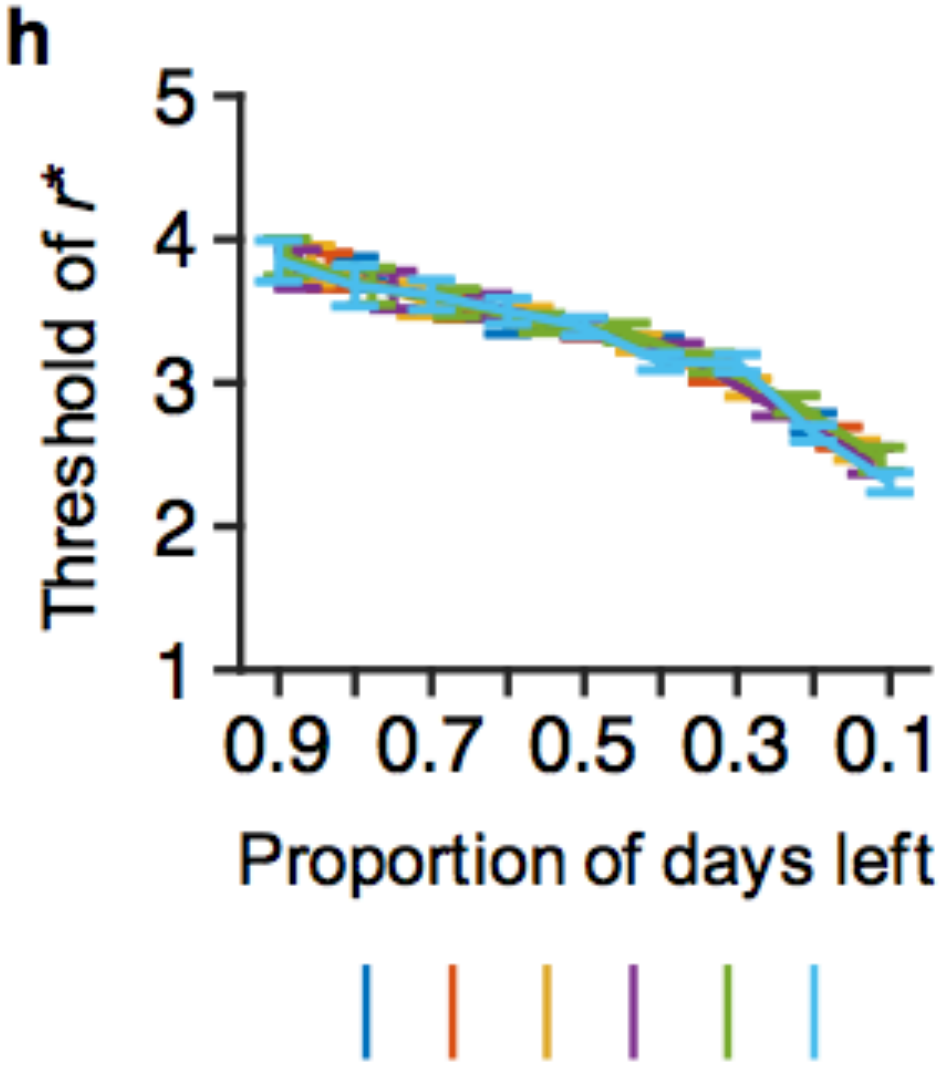
Decision Noise + bias: What does each of them mean? (See in the model)

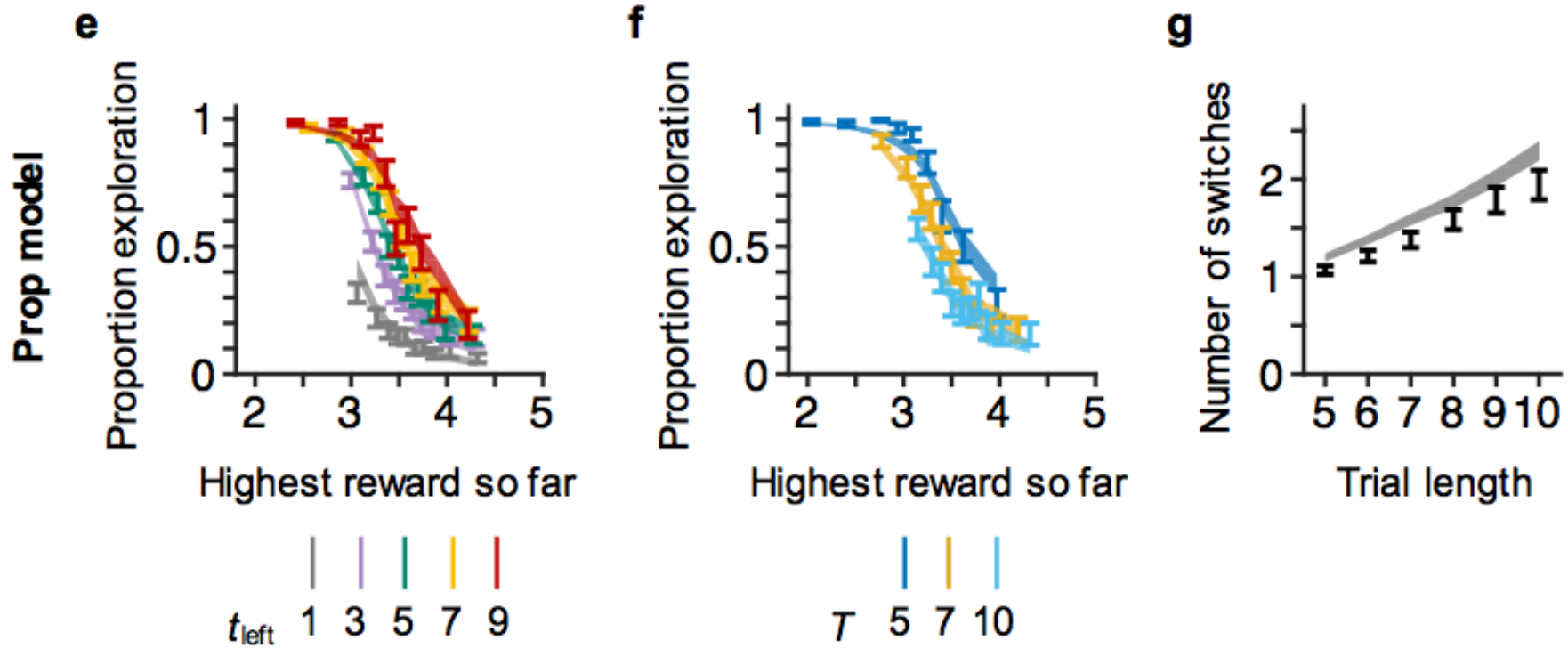



a
b
c
Num model
Proportion exploration
1
0.5
0
2
3
4
5
Highest reward so far
Proportion exploration
1
0.5
0
2
3
4
5
Highest reward so far
Number of switches
2
1
0
5
6
7
8
9
10
Trial length

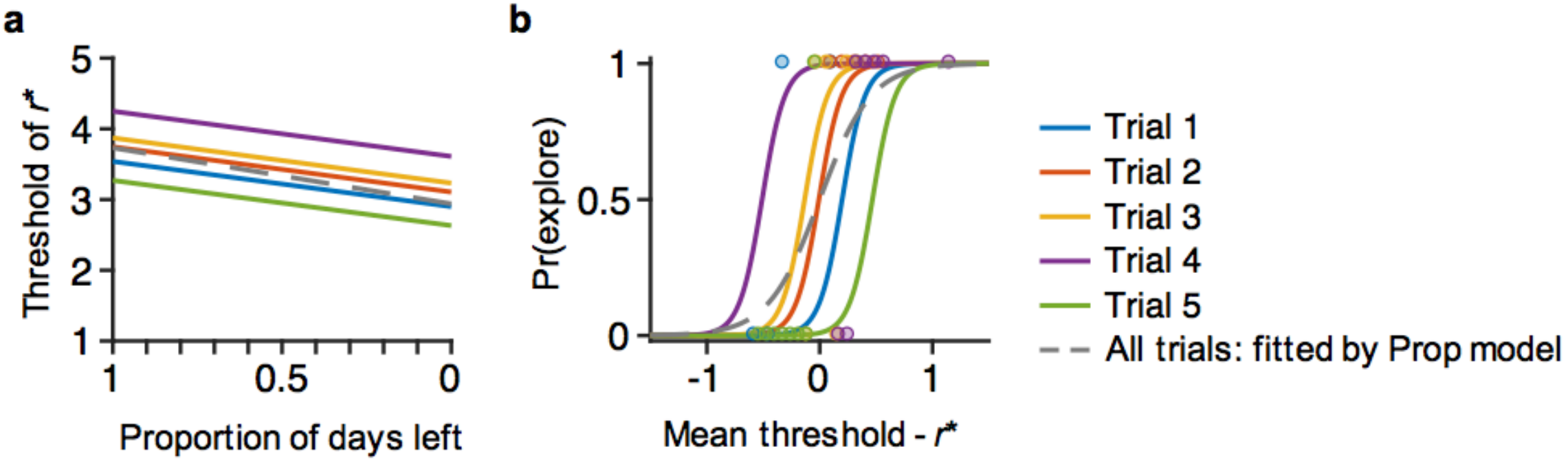
Using proportion

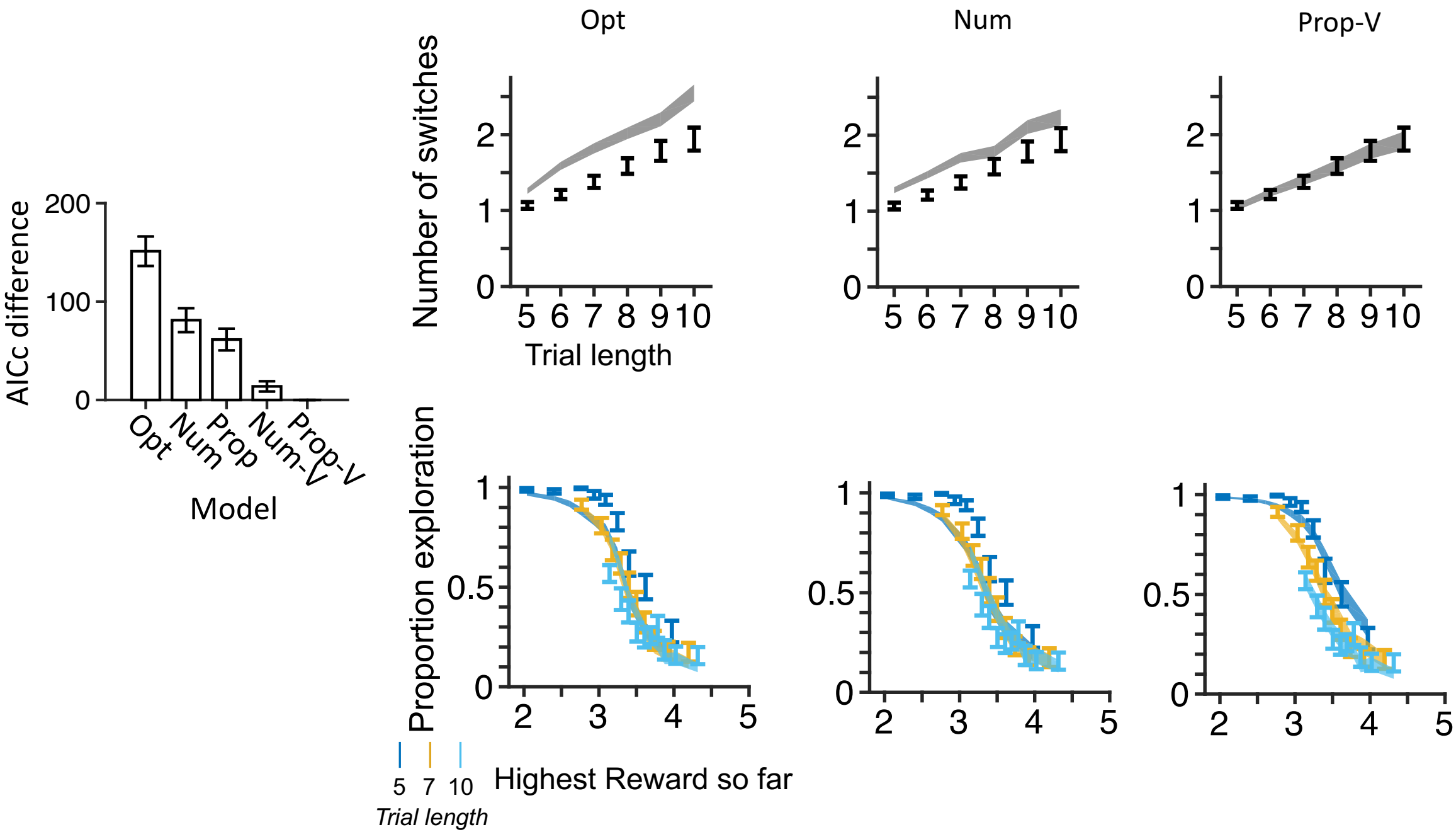





e
f
g
Prop model
Proportion exploration
1
0.5
0
2
3
4
5
Highest reward so far
t_{left} 1 3 5 7 9
T 5 7 10
Number of switches
2
1
0
5 6 7 8 9 10
Trial length

Sequence level variability




AICc difference
200
100
0
Opt
Num
Prop
Num-V
Prop-V
Model
Opt
Num
Prop-V
Number of switches
0
1
2
5 6 7 8 9 10
Trial length
Proportion exploration
0
0.5
1
2 3 4 5
5 7 10
Trial length
Highest Reward so far

What have we learned?

- People have systematic errors in exploration exploitation tasks.
- Proportion of days left (rather than number of days left) explains the threshold better.
- Sequence-level variability
- Indication that subjects might have a “**plan**” for each trial

(Nature Human Behavior (In Revision))

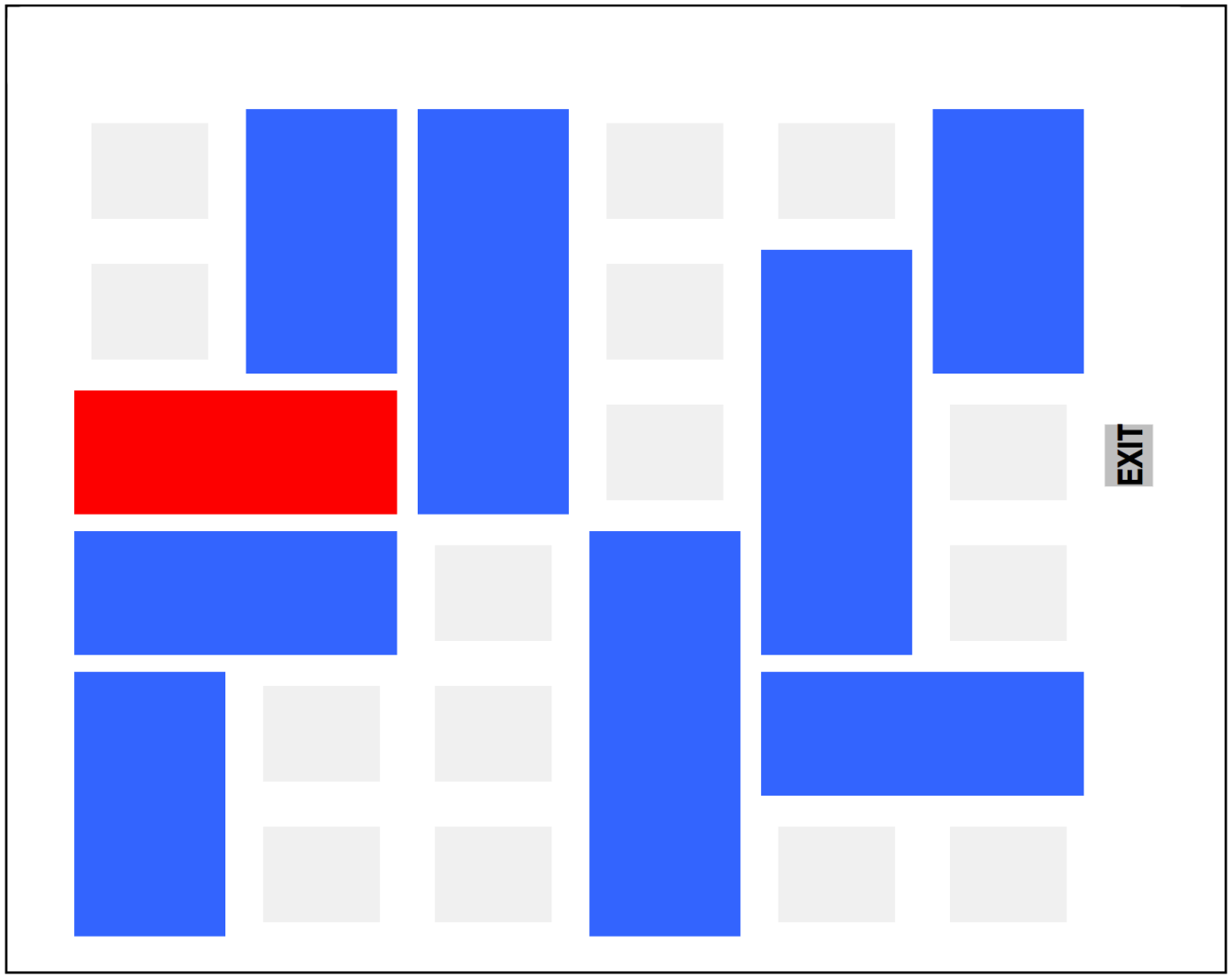
Artificial Intelligence and Cognitive Science

- Different goals
- Modelling of complex behavior is possible
  - But it is not trivial
  - Making conclusions/narrative is challenging.
- Many modes of failures play a role when it gets to human behavior.
  - Must be distinguishable
  - Based on a lot of assumptions
- Many interesting domains applicable to both fields
  - Games and puzzles
  - Daily routines

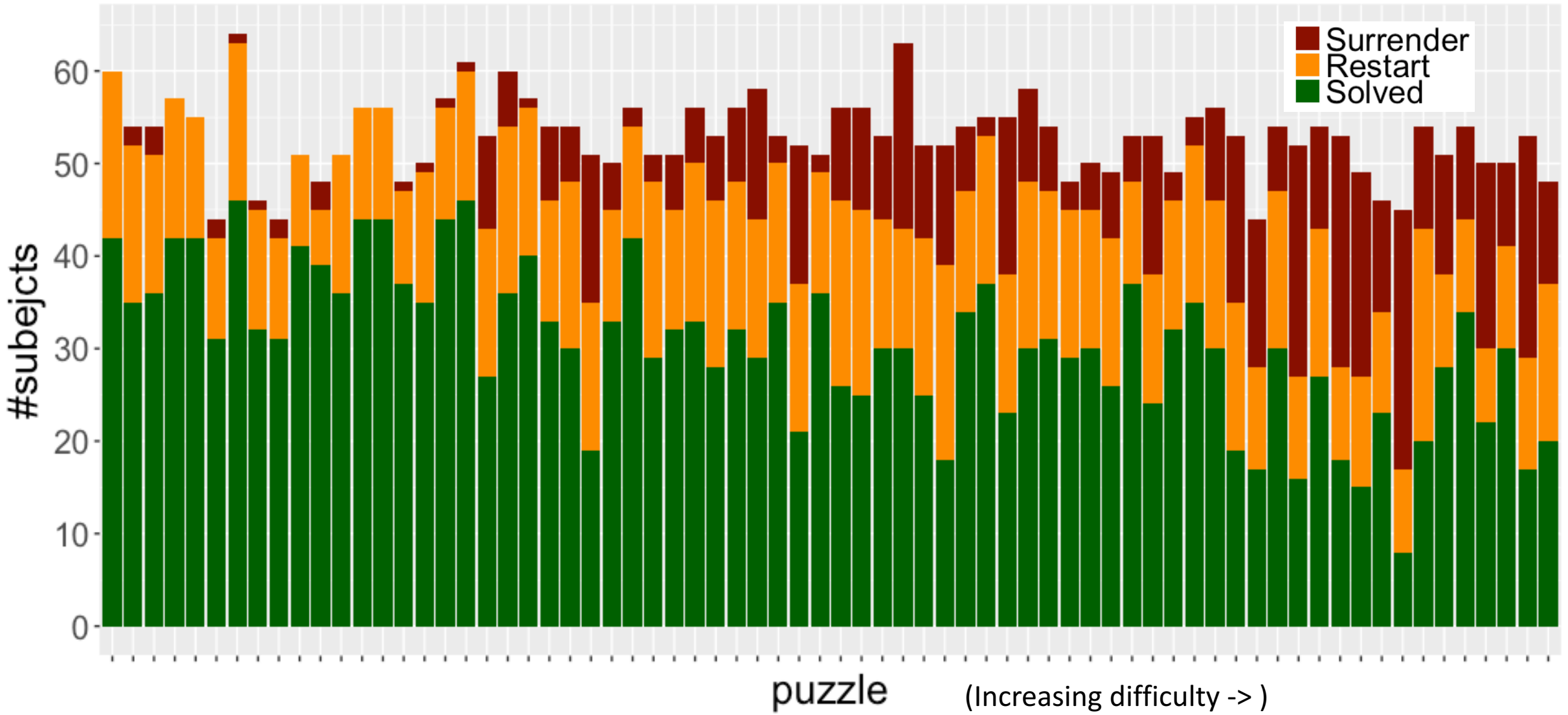
Future Directions

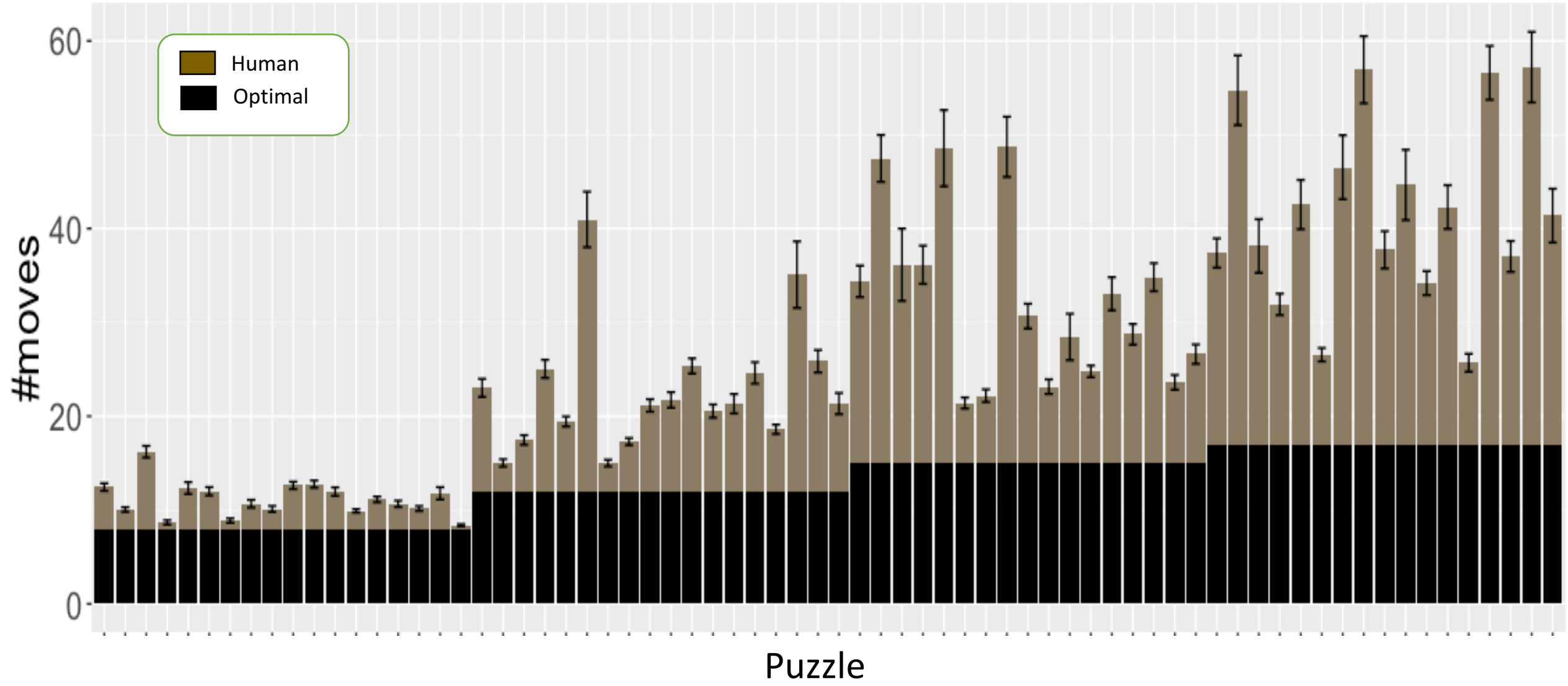
- How do humans "search" for a plan?
- What heuristics are they using?
  - Do people abstract and/or relax the state and action space?
- Are people using *Reinforcement Learning*? Classical Planning? Heuristic Search algorithms?
- What makes a task difficult for humans?
- Many interesting domains applicable to both fields
  - Games and puzzles
  - Daily routines

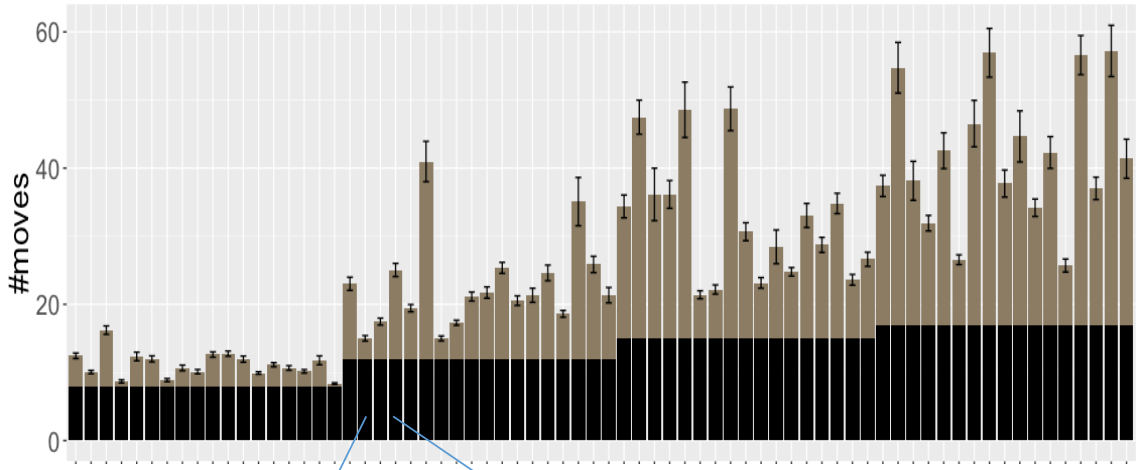
Rushhour



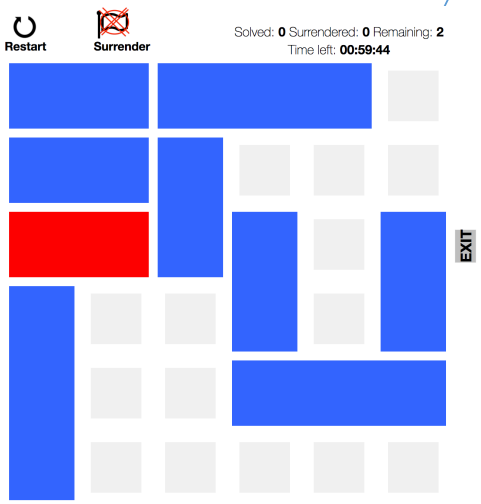
- Simple game
- Easy Abstraction
- No intuitive easy heuristics
- Fun and engaging


Surrender
Restart
Solved
#subejcts
60
50
40
30
20
10
0
puzzle
(Increasing difficulty ->)

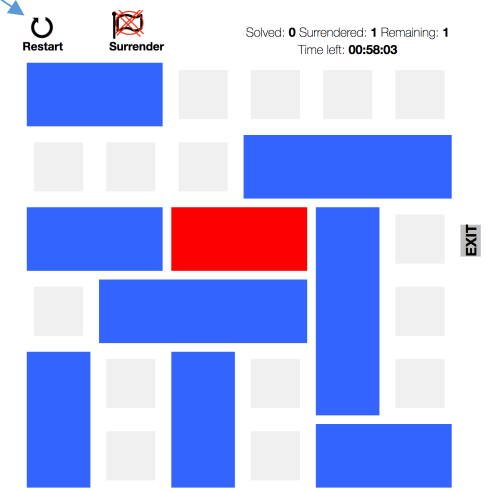

Human
Optimal
60
40
20
0
#moves
Puzzle



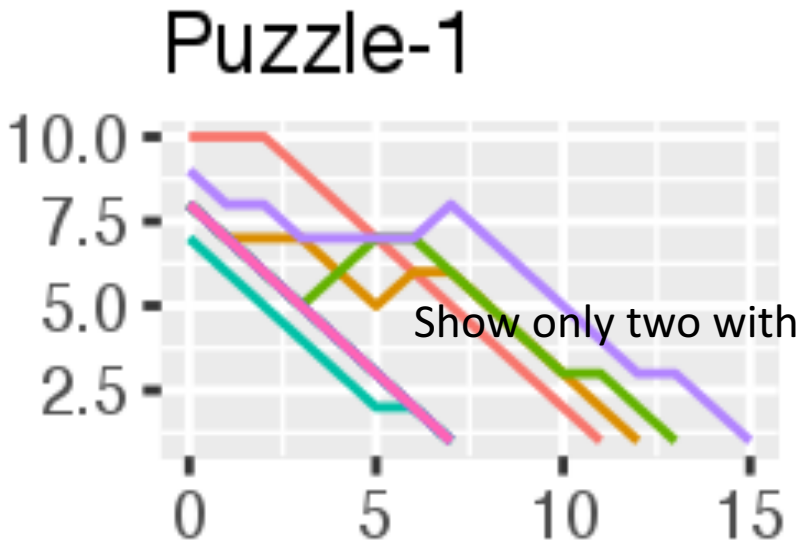
What makes a puzzle difficult?

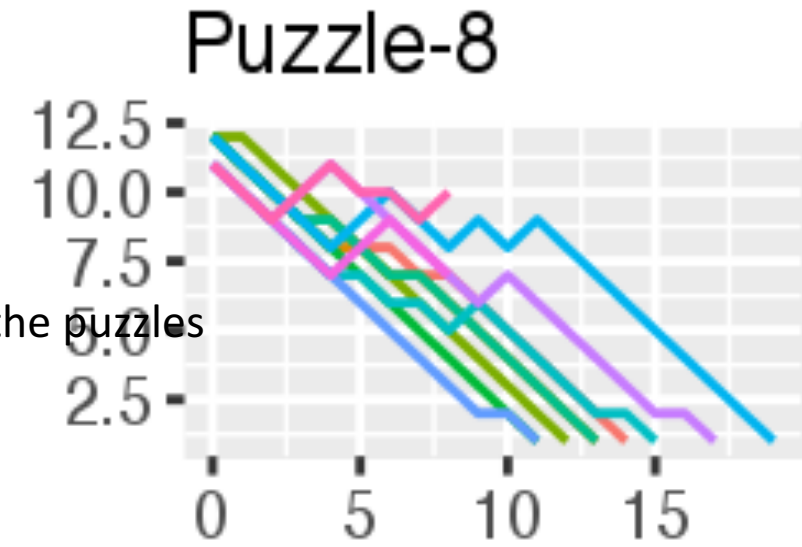


Instance 20 / prb29414

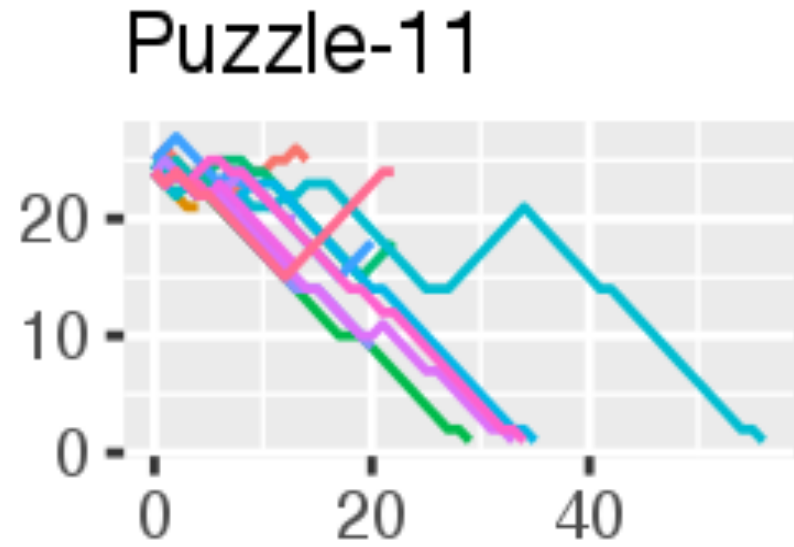


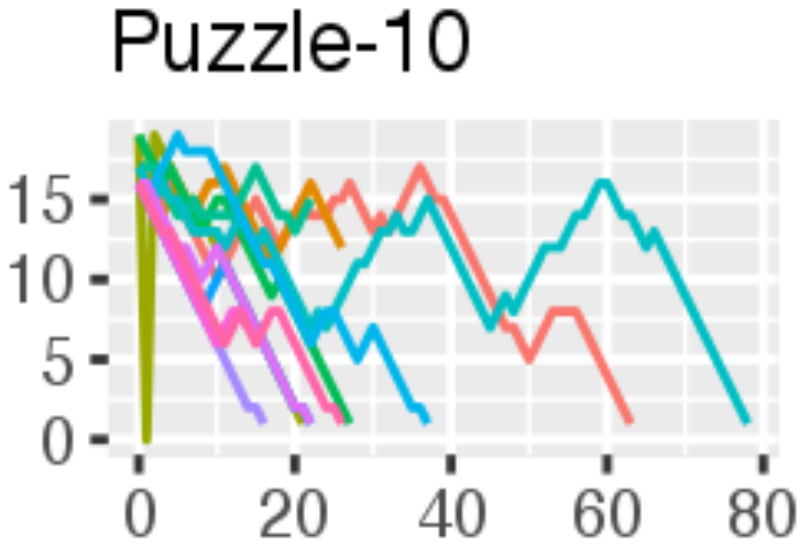
Instance 24/ prb3217

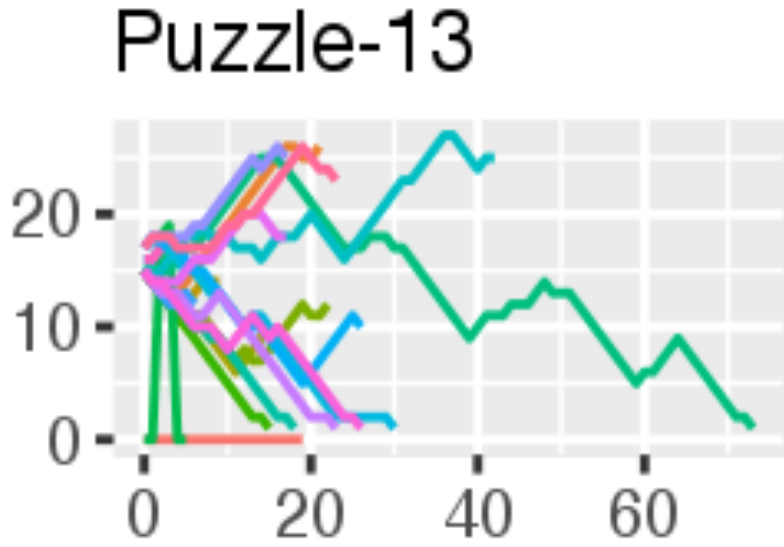


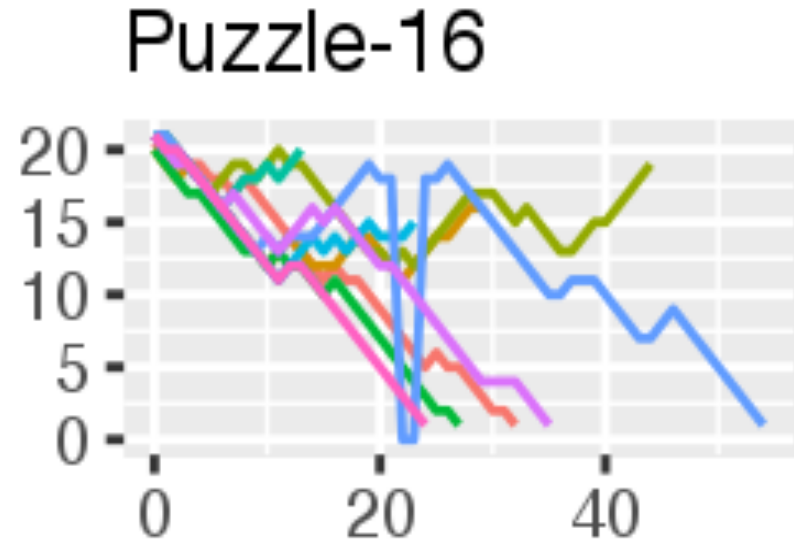


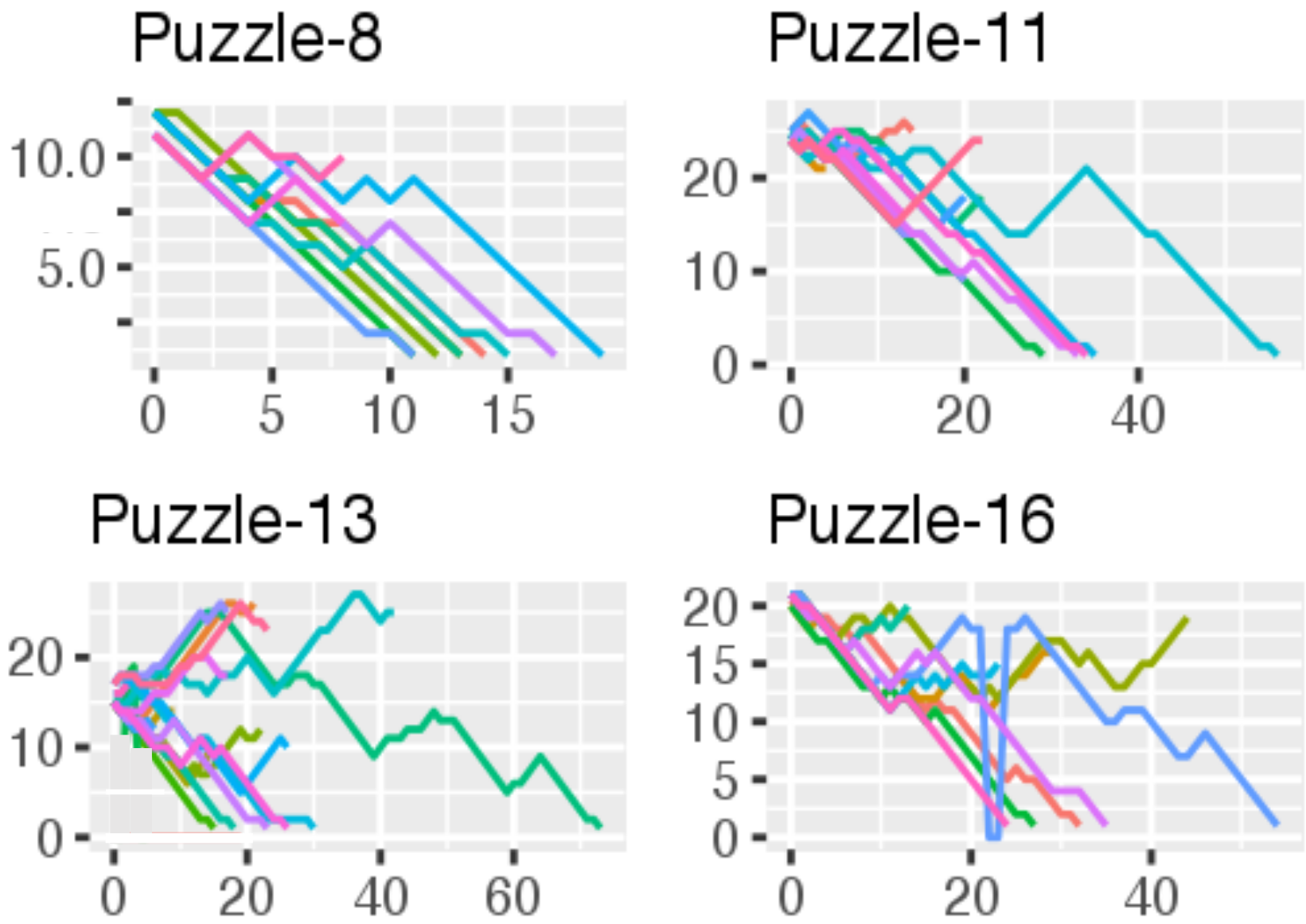
Show only two with the puzzles





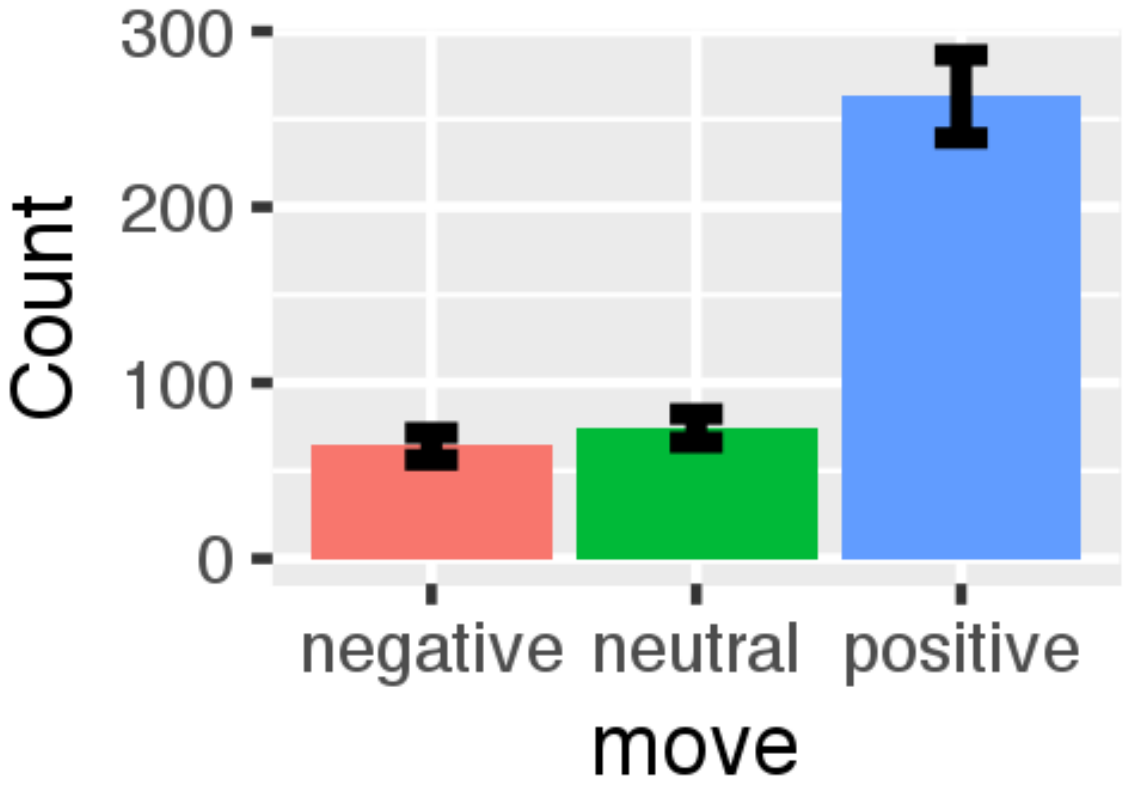




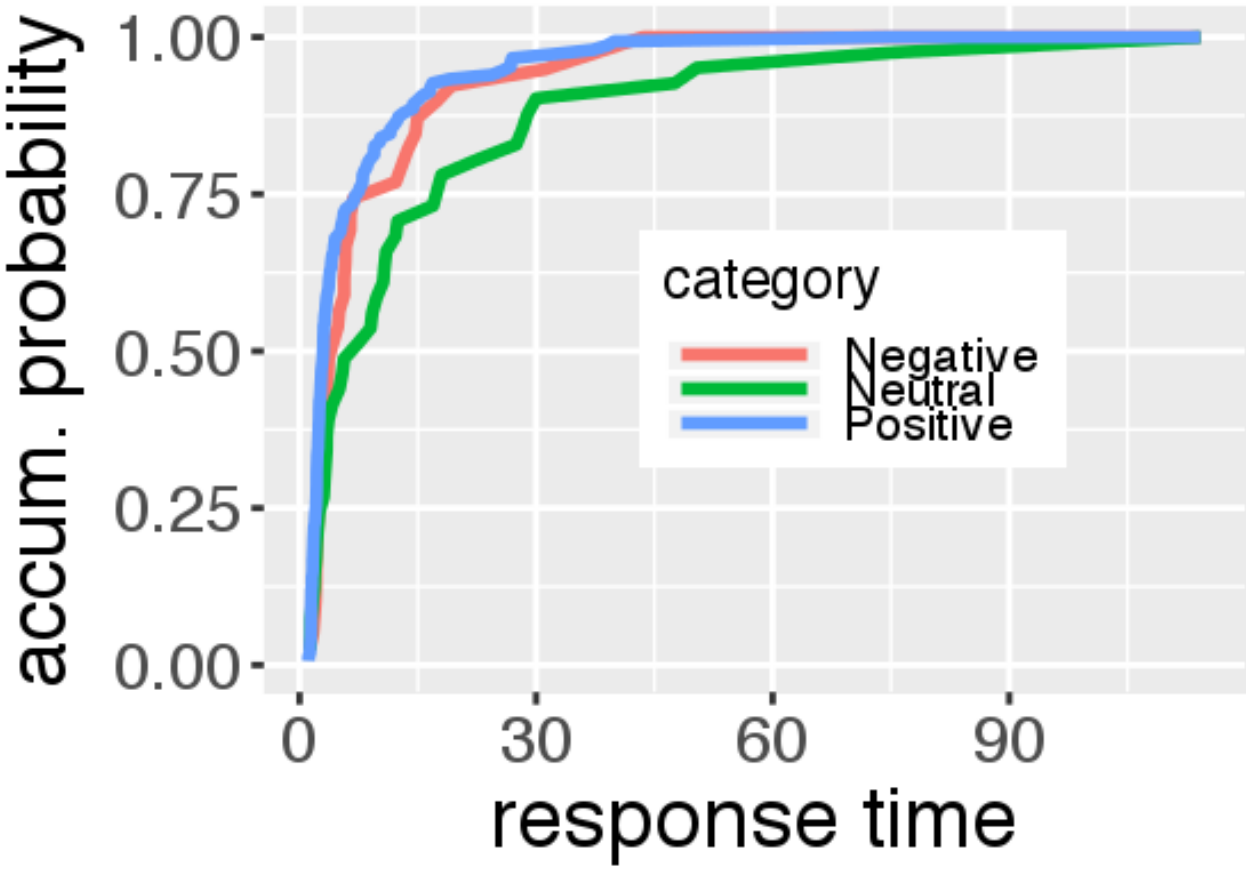

Puzzle-8
10.0
5.0
0
5
10
15
Puzzle-11
20
10
0
0
20
40
Puzzle-13
20
10
0
0
20
40
60
Puzzle-16
20
15
10
5
0
0
20
40

Move categories

- **Positive moves** – get the subject closer to the goal
- **Neutral moves** – Gets the subject in the distance
- **Negative moves** – Gets the subject far away from the goal



Response Times



What Strategy did you use?

"I figured out which car was blocking the way and then what I had to do in order to fix it, starting backwrads from the exit"

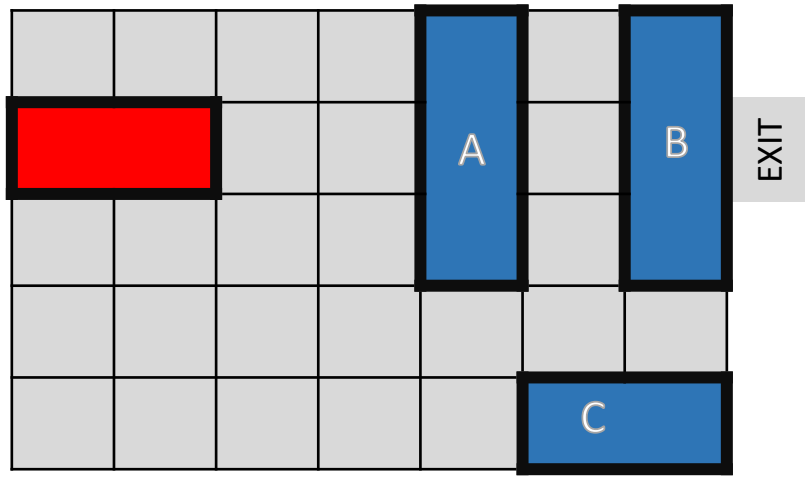
"I find the car that MUST be relocated in a certain way"

"I focus on removing one car at a time"

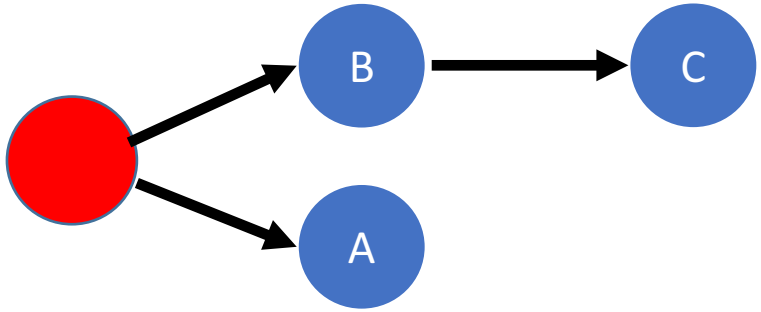
"I had no clue what I was doing."

Do people solve rushhour using Macro-Actions?

- An abstraction of the action space.
- "Clamping" together a few actions to achieve a "sub-goal".
- Macro-Action Graph (MAG)
  - Every node is a "blocked" car
  - An edge ***A->B*** means ***move car B to "unblock" A***.

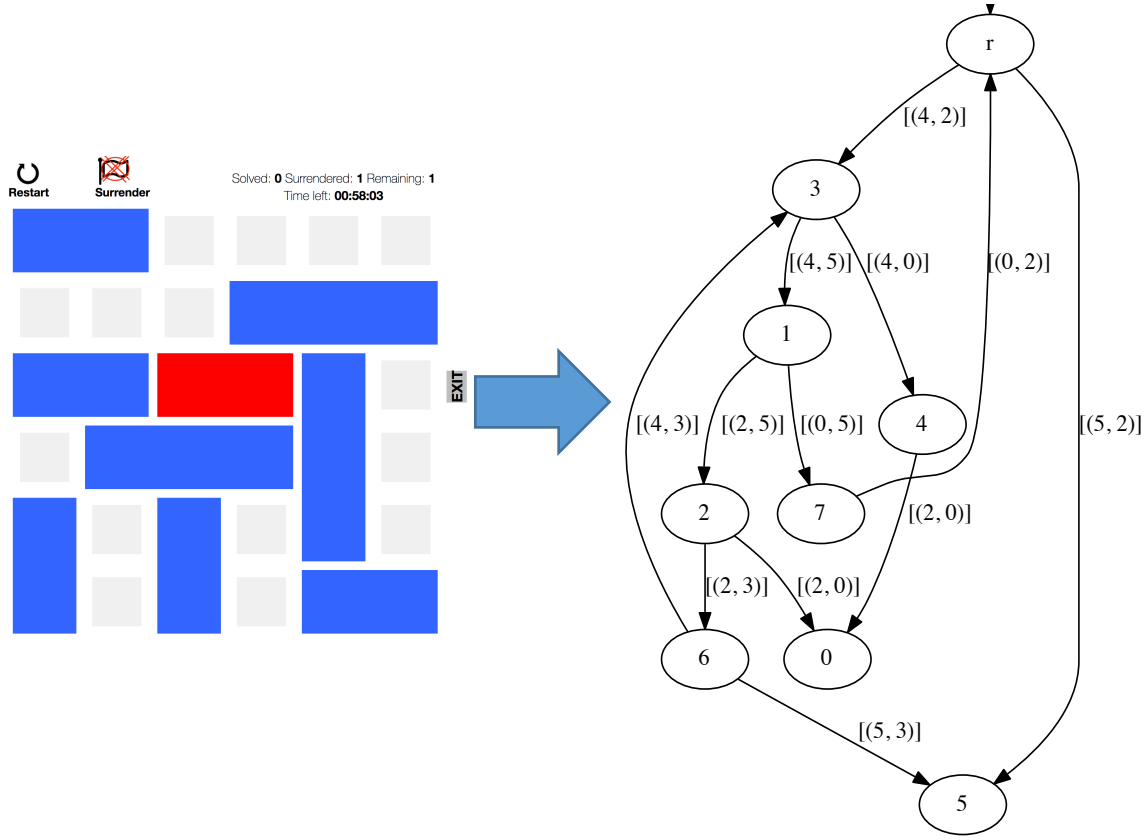


Plan: "Move car *C* to 'unblock' car *B*. Then move cars *B* and *A* to `unblock` the red car."



Analyzing Macro Action Graphs

- What features correlate with difficulty?
  - Number of nodes
  - Number of edges
  - Number of strongly connected components
- How a MAG translates into a plan?
- Can a MAG be used as a heuristic?
- Many More



Summary

- Artificial Intelligence and human behavior are linked together
  - But humans and algorithms are also very different


© Warneken & Tomasgifs.com

Summary

- Artificial Intelligence and human behavior are linked together
  - But humans and algorithms are very different
- There is not much research about how people come up with plans
  - Chess research only deals with expert players (Chase and Simon (1976), Adrien De Groot (1964))
  - Some work on the 15-Tile Puzzle – (Pizlo, Z. and Li, Z., 2005.)
- Some research fields deals with using human data to augment algorithms
  - ***Imitation Learning*** (Schaal, S., 1999) – learning to act based on an expert databases
  - ***Inverse Reinforcement Learning****(*Abbeel, Ng., 2004*)* – learning the reward function
- Bidirectional propagation of knowledge and inspiration between the study of human behavior and search algorithms.

- Thank you!