Knowledge Modelling and Strategy Engineering in Reconnaissance Blind Chess

Robin Stöhr
Shuai Wang
Zhisheng Huang

VRIJE UNIVERSITEIT AMSTERDAM

01. Introduction to RBC
02. Research questions
03. Literature review
04. Knowledge in RBC
05. Analysis of different strategies
06. Discussions/Conclusions

Table of contents

01.

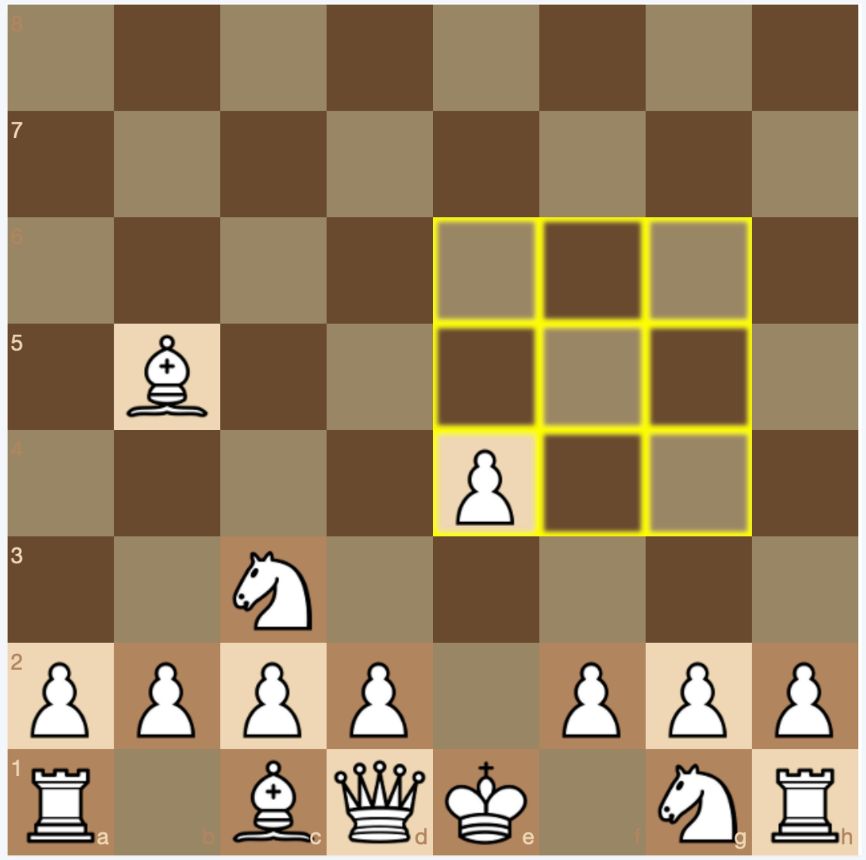
Introduction to RBC

What is Reconnaissance Blind Chess?

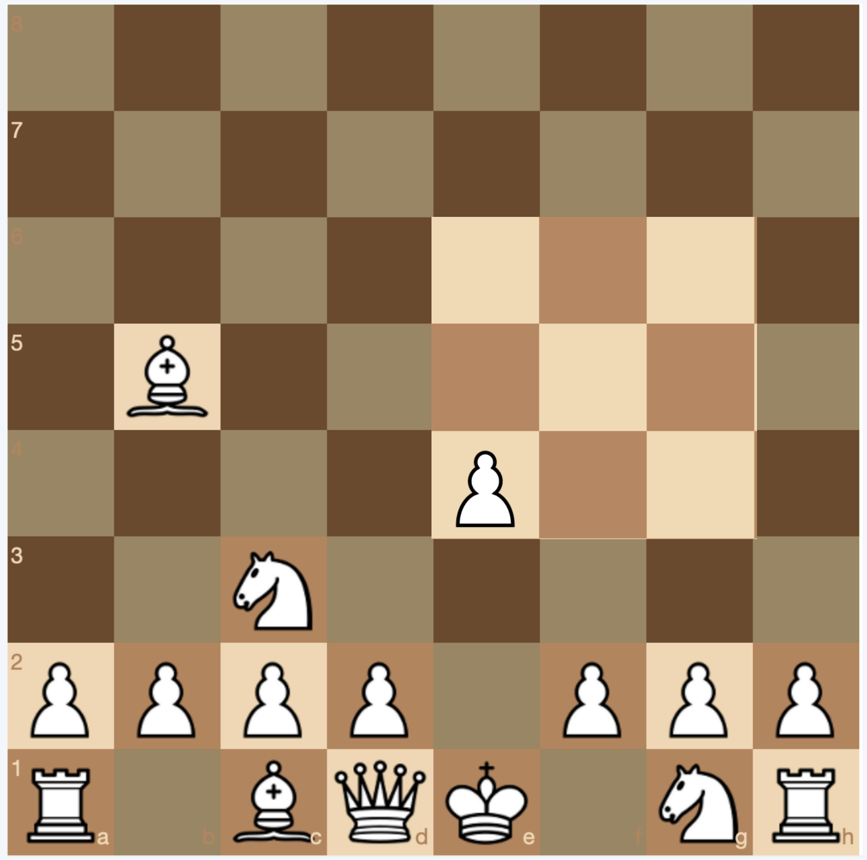
“Reconnaissance Blind Chess (RBC) is a chess variant designed for new research in artificial intelligence (AI). RBC includes imperfect information, long-term strategy, explicit observations, and almost no common knowledge. These features appear in real-world scenarios, and challenge even state of the art algorithms.“

Key mechanics

“Fog of War”



Sense action



Next legal move counts

Complexity

Classical Chess	10^{43}
Reconnaissance Blind Chess	10^{139}
Go	10^{170}

02.

Research Questions

Research Questions

RQ 1

Can the estimation of the opponents knowledge contribute to improving one’s game performance?

RQ 2

What are the most effective strategies to diminish the game’s inherent uncertainty?

RQ 3

Utilizing the current knowledge with the given uncertainty, what is the most effective move strategy?

03.

Literature Review

Literature Review

Most bots were developed for the NeurIPS (Conference on Neural Information Processing Systems) 2021 and 2022 competition

Rank	User	Rating	Num Ranked Matches
1	StrangeFish2 v264	1724	31735
2	JKU-CODA v296	1577	119
3	Châteaux v1	1569	9410
4	Kevin v26	1504	103
5	SiameseOptimist v19	1482	423
6	ROOKie v13	1481	24790
7	Oracle v1	1459	156953
8	LaQ-Bot v4	1455	121
9	StrangeFish v6	1452	38941
10	hydroblade Human	1450	93

01

StrangeFish

Uses classical Stockfish engine for move creation. Senses based on next move to make. (Johns Hopkins University)

02

JKU-CODA

Based on the history of the game. Uses Neural Nets and self-play learning for both move and sense. (Johannes Kepler University, Linz)

03

Chateaux

Constructs a belief state using an unweighted particle filter. Information of sampled states is converted into bit boards and those are used as input for a Neural Net. (Google)

NeurIPS 2021 RBC competition

Bot	Elo	Elo rank	Tracks all board states	Tracks opponent infosets	Tracks full game state	Models opponent moves	Models opponent senses	Uses a chess engine	Senses to minimize states
Fianchetto	1759	1	●			●		●	
StrangeFish2	1662	2	●					●	
penumbra	1584	3	●	●		●	●		
Kevin	1544	4	●	●		●	●	●	
Oracle	1503	5	●					●	●
Gnash	1454	6	●	●		●		●	●
Marmot	1315	7	●	●		●	●	●	
DynamicEntropy	1299	8	●					●	
Frampt	1208	10	●						
GarrisonNRL	1140	11				●		●	●
trout	1127	12						●	
callumcanavan	1066	13							
attacker	1049	14							
URChIn	854	15						●	●
random	753	17							

Bot	Elo	Elo rank	Median # States	# States Rank	# States Δ Rank	Engine Move Agreement	Move Agree. Rank	Move Agree. Δ Rank
Fianchetto	1759	1	9	2	+1	72%	2	+1
StrangeFish2	1662	2	15	5	+3	59%	4	+2
penumbra	1584	3	24	10	+7	43%	7	+4
Kevin	1544	4	13	3	-1	67%	3	-1
Oracle	1503	5	13	4	-1	73%	1	-4
Gnash	1454	6	18	7	+1	58%	5	-1
Marmot	1315	7	20	9	+2	28%	11	+4
DynamicEntropy	1299	8	16	6	-2	39%	9	+1
wbernar5	1219	9	5	1	-8	28%	12	+3
Frampt	1208	10	19	8	-2	25%	13	+3
GarrisonNRL	1140	11	61	11	0	44%	6	-5
trout	1127	12	3243	14	+2	36%	10	-2
callumcanavan	1066	13	7158	15	+2	8%	16	+3
attacker	1049	14	>1M	17	+3	4%	17	+3
URChIn	854	15	124	12	-3	39%	8	-7
armandli	777	16	204	13	-3	15%	14	-2
random	753	17	68263	16	-1	8%	15	-2
ai_games_cvi	288	18	-	-	-	-	-	-

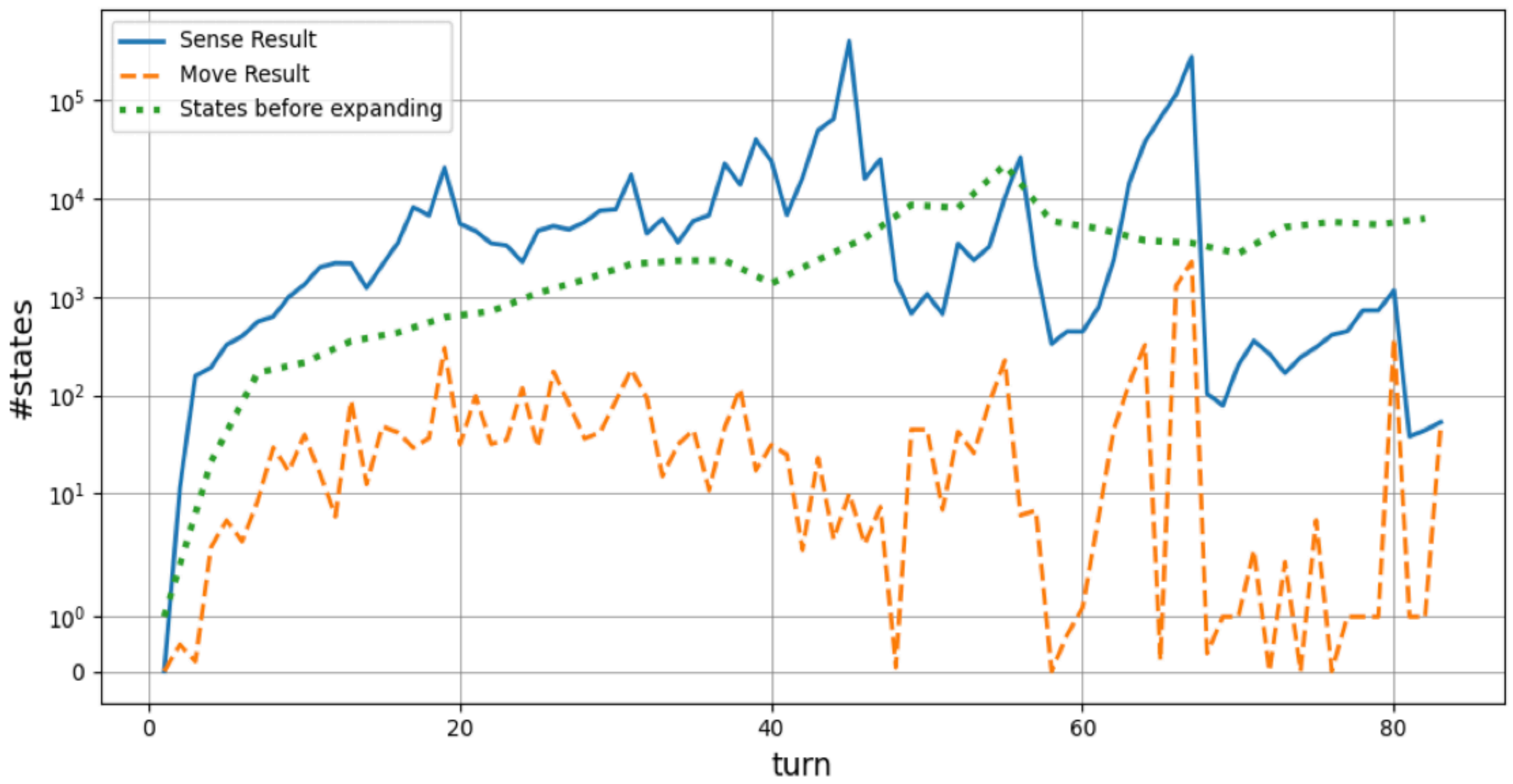
04.

Knowledge in RBC

Knowledge modelling in RBC

Three ways of obtaining knowledge

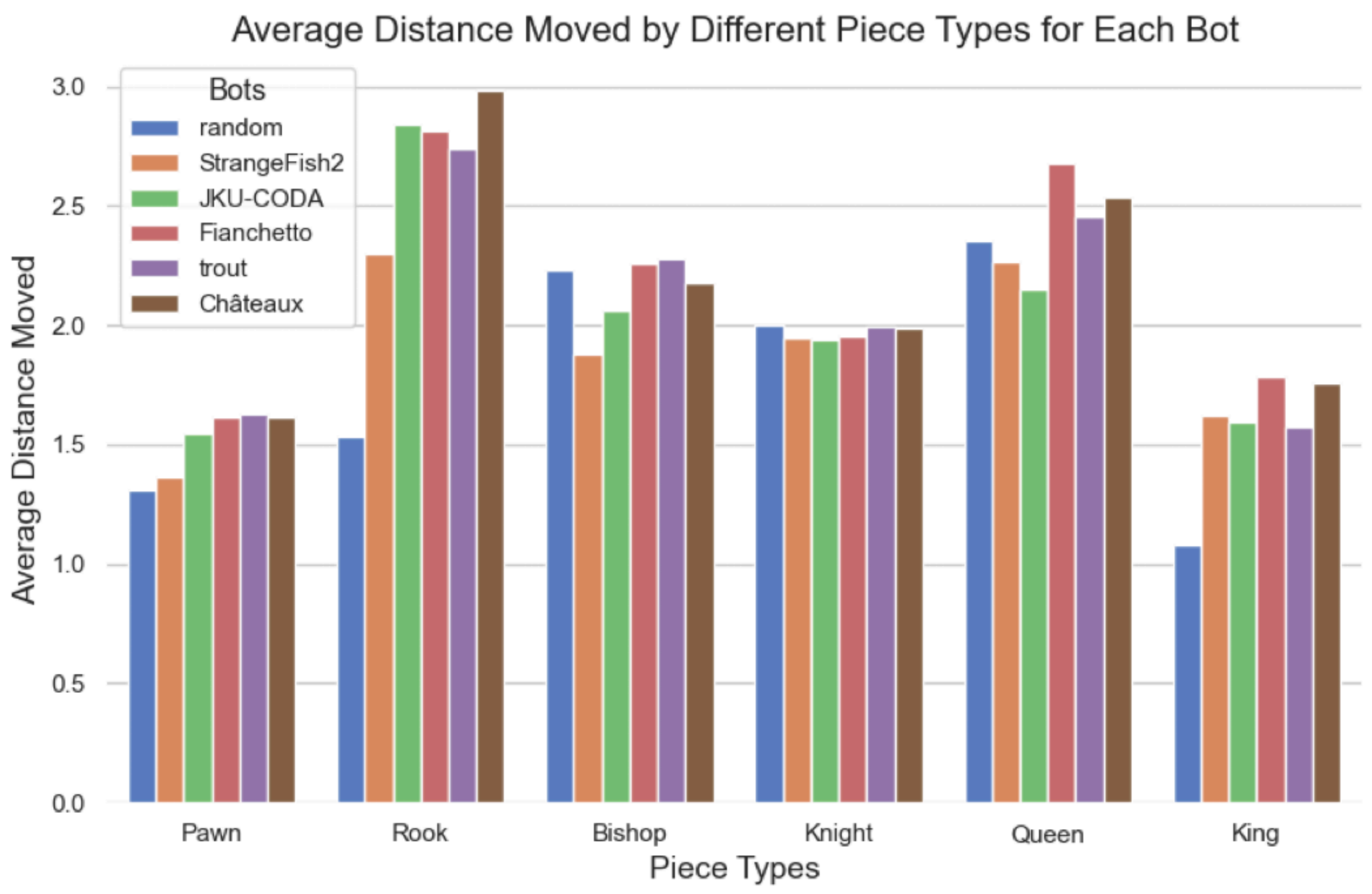
1. Sense result
2. Move result
3. Opponents move result



*Results taken from roughly 500 historic games

Gaining insights from piece movement characteristics

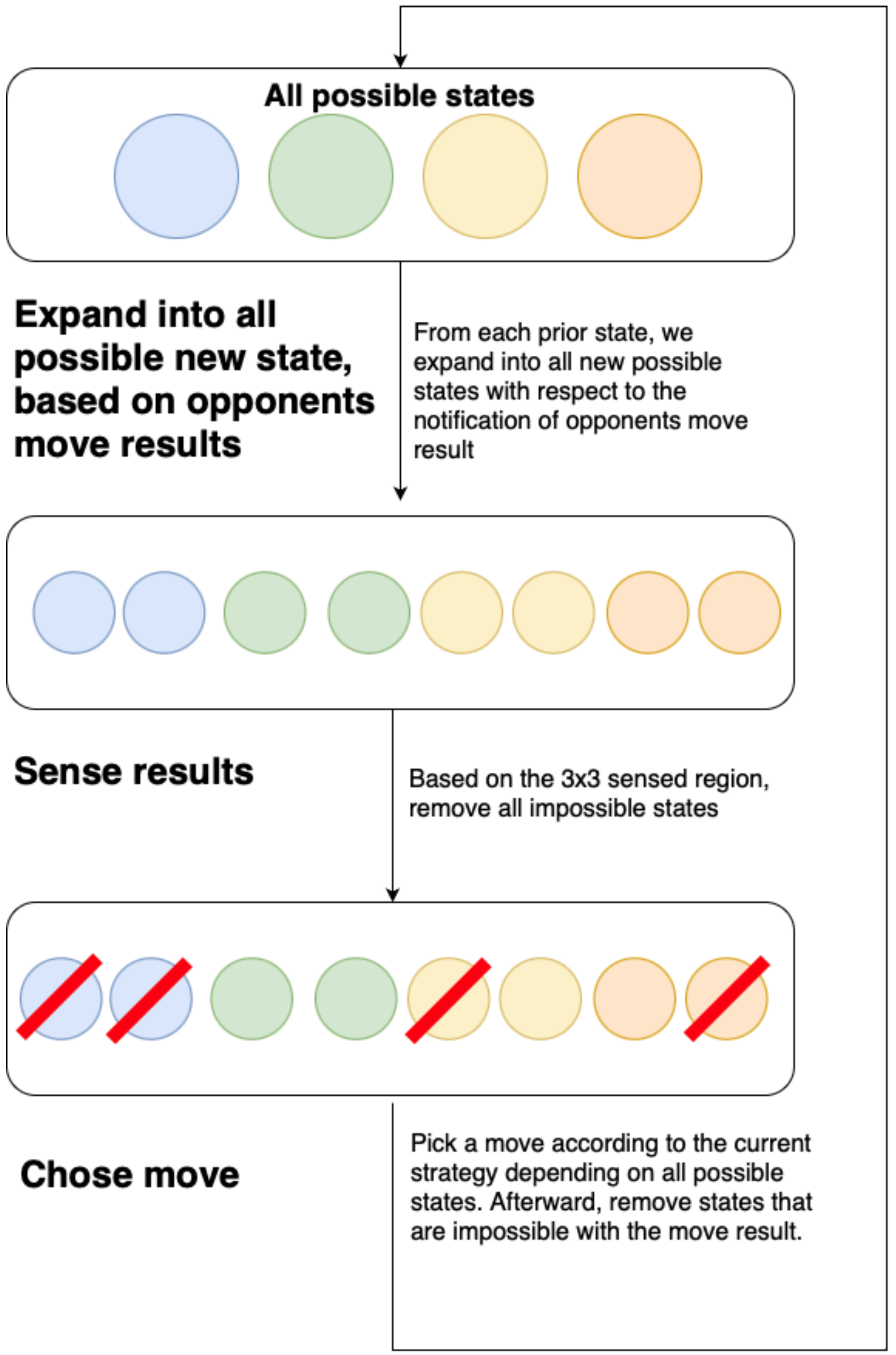
Distance as a measure of aggressiveness



*Results taken from historic games

Uncertain pawn scout move

Knowledge modeling flow



05.

Analysis of different strategies

Sense strategies

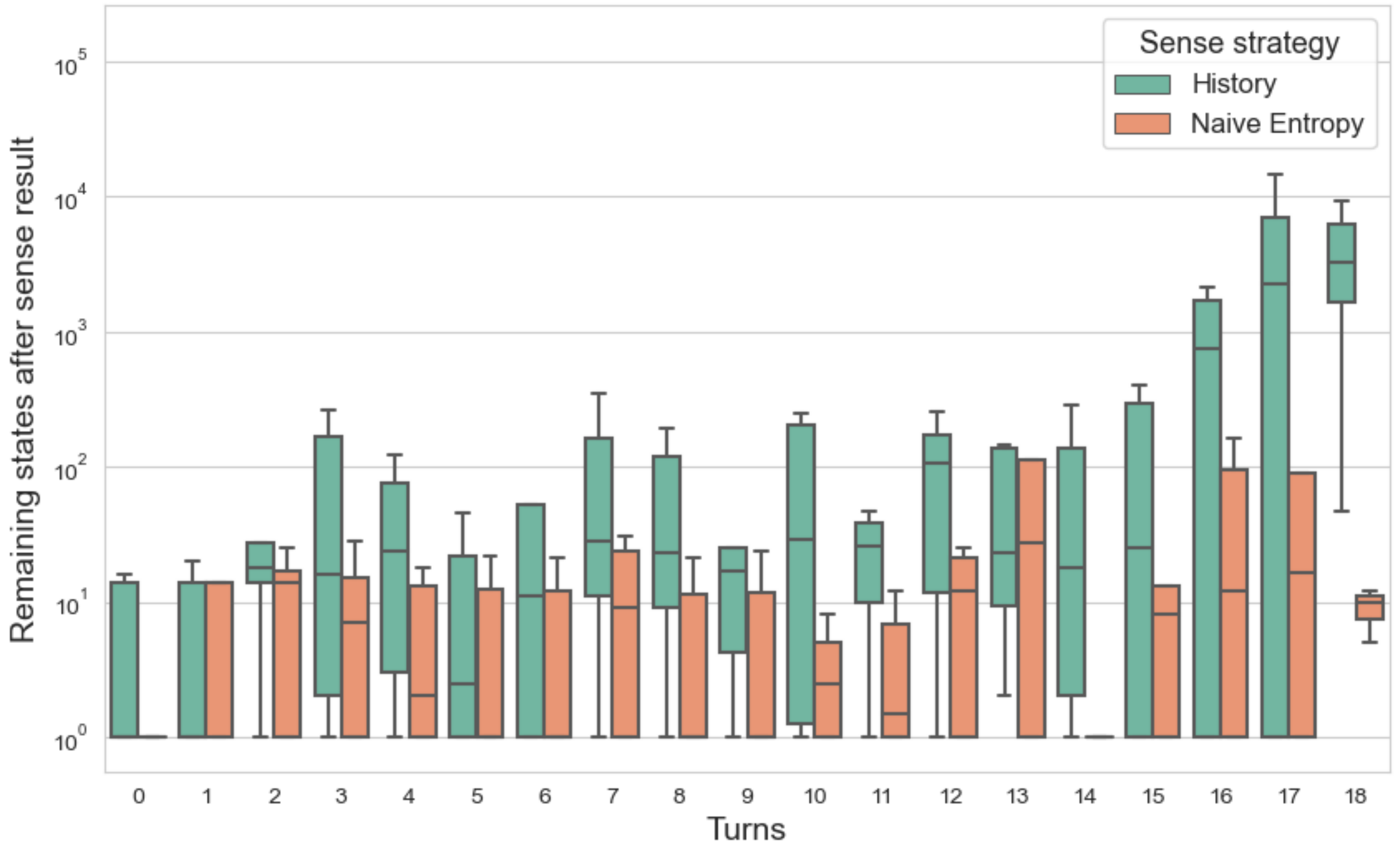
01. Naive Entropy Sense
02. Adapted Entropy Sense
03. Opponents move weights sense
04. Entropy with most likely states

Naive Entropy Sense

Find highest 3x3 square using Shannon entropy formula

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

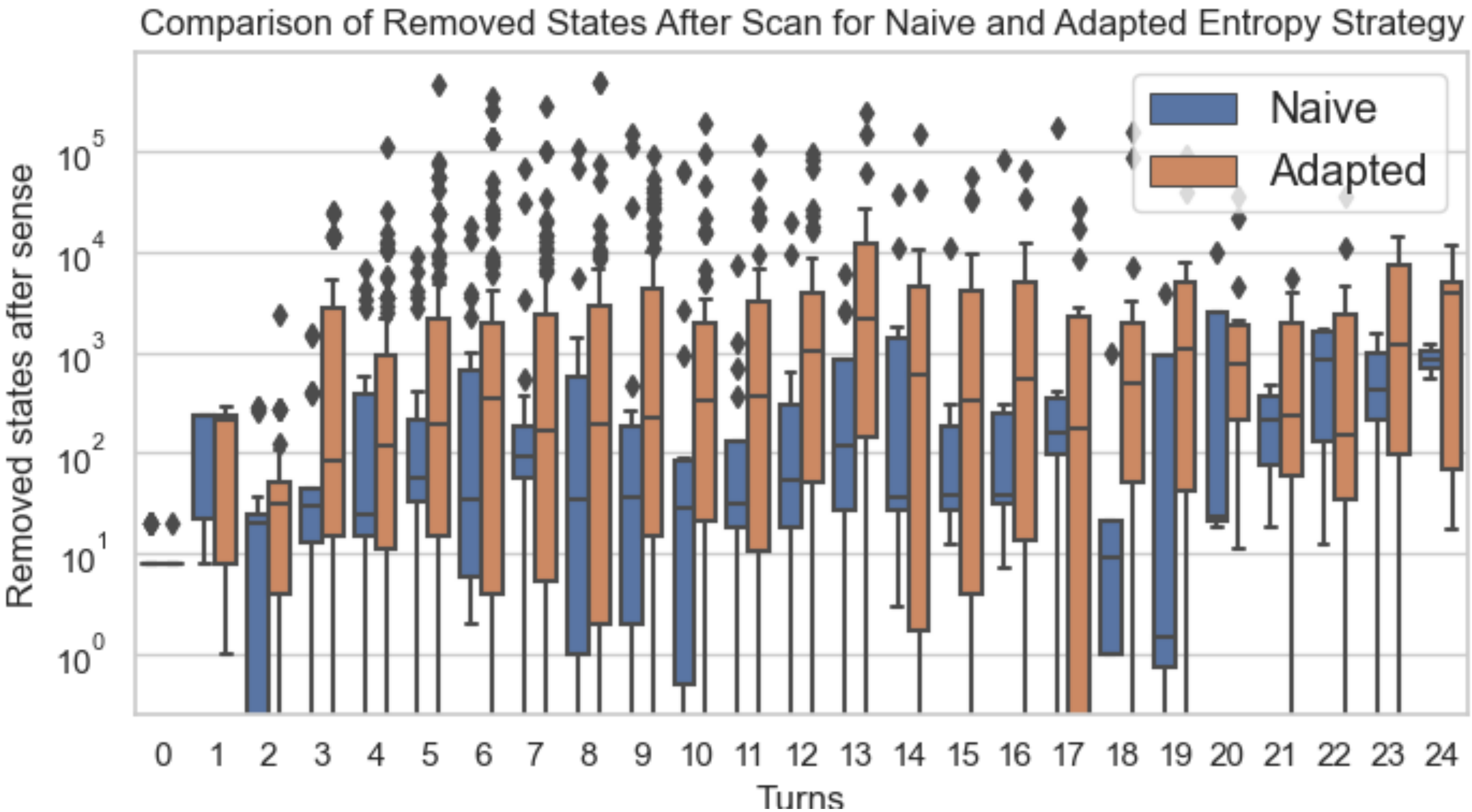
Naive Entropy Sense Performance



Adapted Entropy Sense

Added manual heuristic to the naive entropy:

- King safety
- Time factor decay
- Piece weights



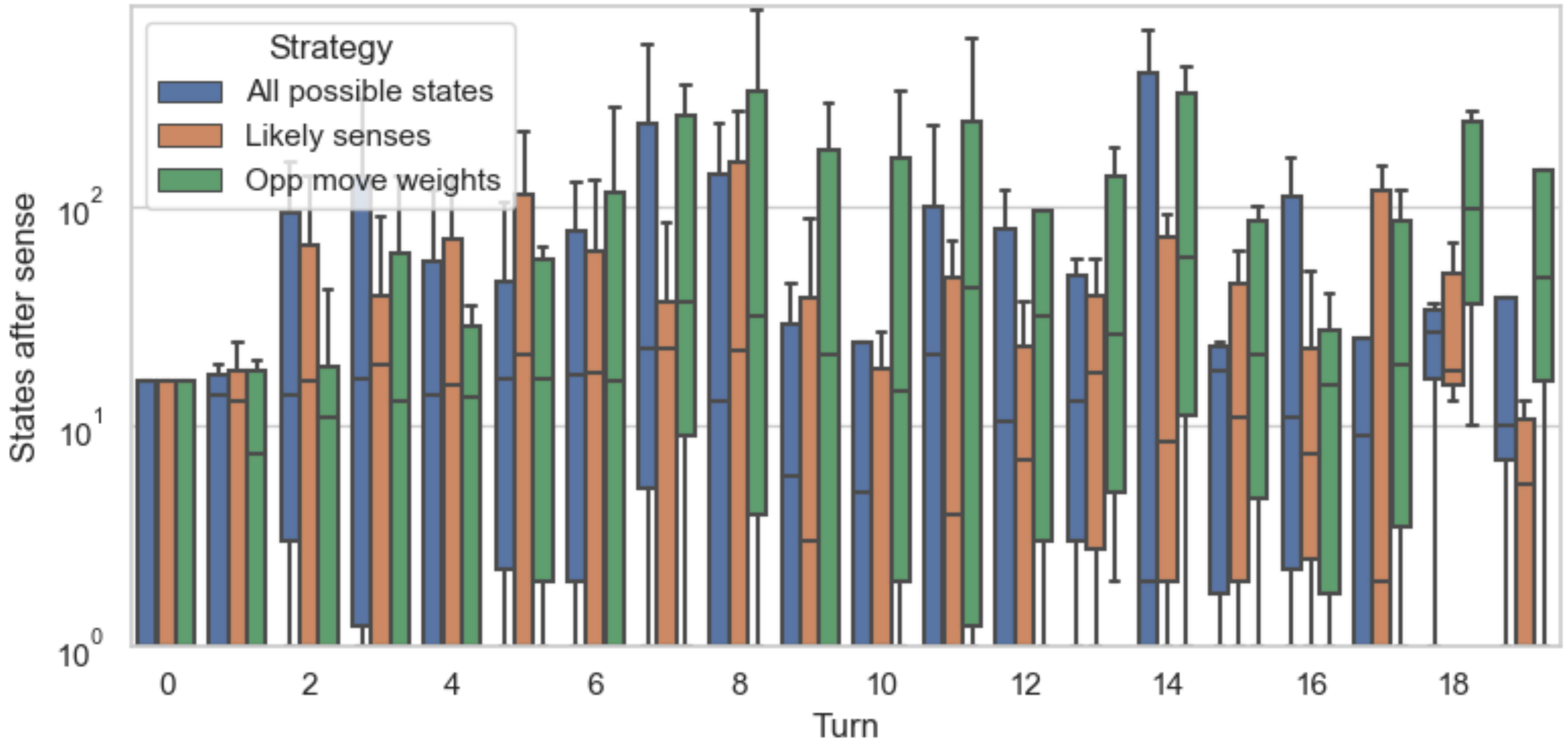
Opponents move weights

- Assign scores for all moves for all boards.
- Scores are added ‘to’ and ‘from’ squares.
- Sense highest score area.

Likely states entropy sense

- For every board state, expand into all possible states and calculate which states are the most likely by taking the n best moves.
- Apply the adapted entropy sense based on the likely boards, in contrast to all boards.

Comparison



Strategy
All possible states
Likely senses
Opp move weights
States after sense
10^0
10^1
10^2
0
2
4
6
8
10
12
14
16
18
Turn

Move strategies

01. Baseline classical chess engine (SunFish)
02. Baseline + Theory of Mind
03. Neural Network approach (Lc0)

Baseline

- Assign moves scores to all boards using a classical chess engine
- Weigh those moves with the ‘extremeness’ of the boards.
- Take move with highest score.

Baseline + ToM

- Using a graph representation for the complete game history.
- Replay the game for every leaf node from the opponents point of view
- Take an average of the possible boards from the opponents point of view for all board states, as the new scores
- Continue in the same way

Neural Network based approach

- Utilizing Leela chess zero
- A single forward pass provides scores for the board evaluation AND for all possible moves

  -> We are able to compute wary higher quantities
- Indexed based penalty system for moves

$$\mathrm{pv} = 4 \cdot \log(idx + 1)$$

Move strategies comparison

	Scorca (Sunfish)	Scorca (Sunfish+ToM)	Trout	StrangeFish(v2)
Sunfish	-	-	60%	-
Sunfish + ToM	70%	-	70%	-
Leela Chess	100%	100%	90%	70%

06.

Discussion + Conclusion

Coming back to the research questions:

RQ1: We used measures to estimate the opponents knowledge of the field and incorporated that in the decision making of our classical agent, resulting in better moves.

RQ2: We proposed a strategy to have an entropy based scanning in combination with a Neural Network based chess engine which resulted in the best performance in eliminating uncertainty.

RQ3: We evaluated the performance of different strategies and found a Neural Network based approach to lead to the best results. Combination with ToM further increases this, but is not feasible due to time constraints.

Questions