Exploring the No-3-in-Line Problem with Monte Carlo Tree Search

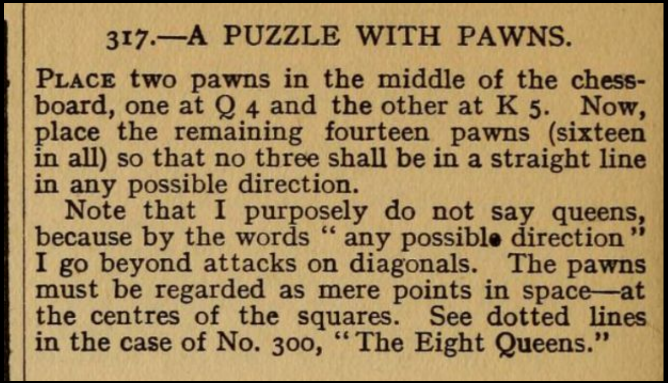
Luoning Zhang

Faculty Mentor: Prof. Nathan Kaplan
Graduate Student Mentors: Xu Zhuang, Tianhao Wang

Department of Mathematics, School of Physical Sciences, UC Irvine

2025 Fall Project Pitch Challenge

What is the No-3-in-Line Problem?



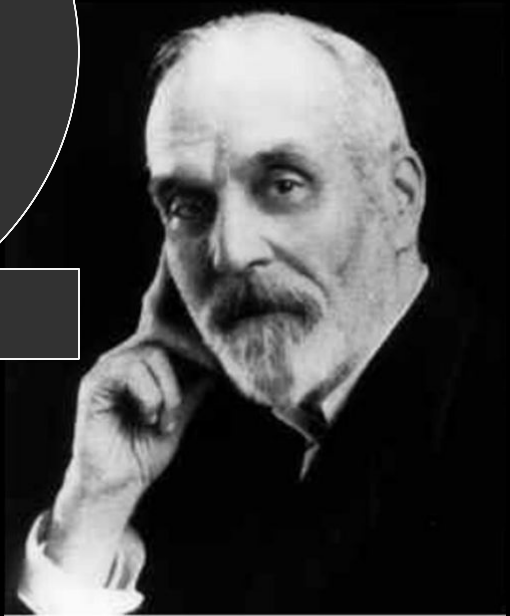
317.—A PUZZLE WITH PAWNS.

PLACE two pawns in the middle of the chess-board, one at Q 4 and the other at K 5. Now, place the remaining fourteen pawns (sixteen in all) so that no three shall be in a straight line in any possible direction.

Note that I purposely do not say queens, because by the words "any possible direction" I go beyond attacks on diagonals. The pawns must be regarded as mere points in space—at the centres of the squares. See dotted lines in the case of No. 300, "The Eight Queens."

Dudeney, 1917

What is the largest number of points we can place on an n x n grid, no three of which lie on a line?



Play it yourself!

Three
Points
Collinear

Play it yourself!

Number of points: 5

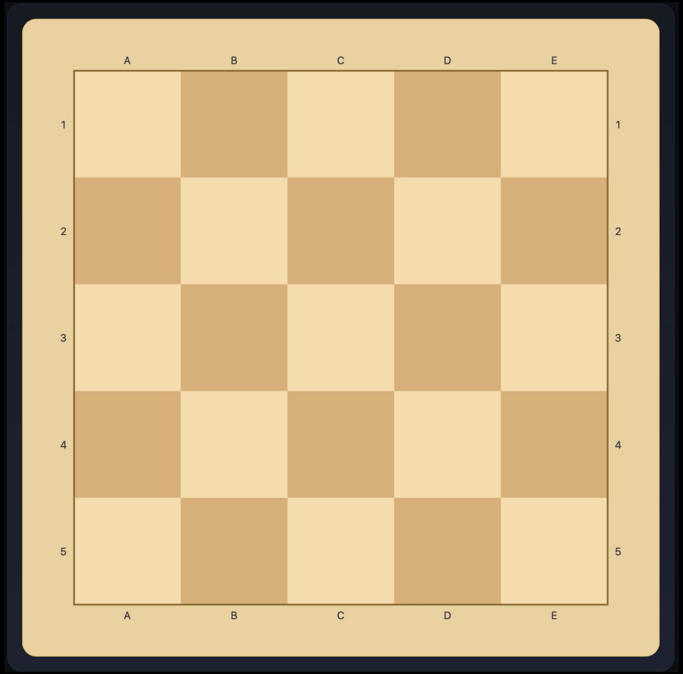
Play it yourself!

Number of points: 6

Play it yourself!

Why is this such a difficult problem, one that has puzzled mathematicians for over 100 years?

Challenge: Try to find 10 points on this 5×5 grid such that no three are collinear.



Theorem (Hall, Jackson, Sudbery, Wild, 1975):
For any ε>0 and all n that are large enough, we can find at least $(1.5-\varepsilon)n$ points on an $n \times n$ grid, no three in a line.

Theorem (Flammenkamp, 1998):
For all $n \leq 46$ and $n = 48, 50, 52$, we can find $2n$ points on an $n \times n$ grid, no three in a line.

Theorem (Hall, Jackson, Sudbery, Wild, 1975):
For any ε>0 and all n that are large enough, we can find at least (1.5-ε)n points on an n × n grid, no three in a line.

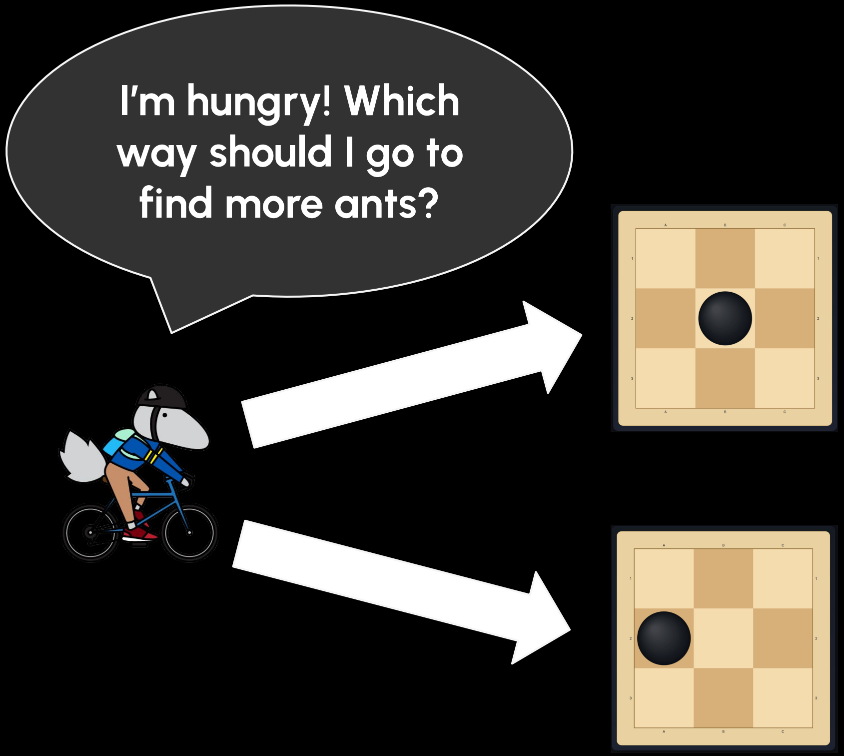
Theorem (Flammenkamp, 1998):
For all n ≤46 and n = 48,50,52, we can find 2n points on an n × n grid, no three in a line.

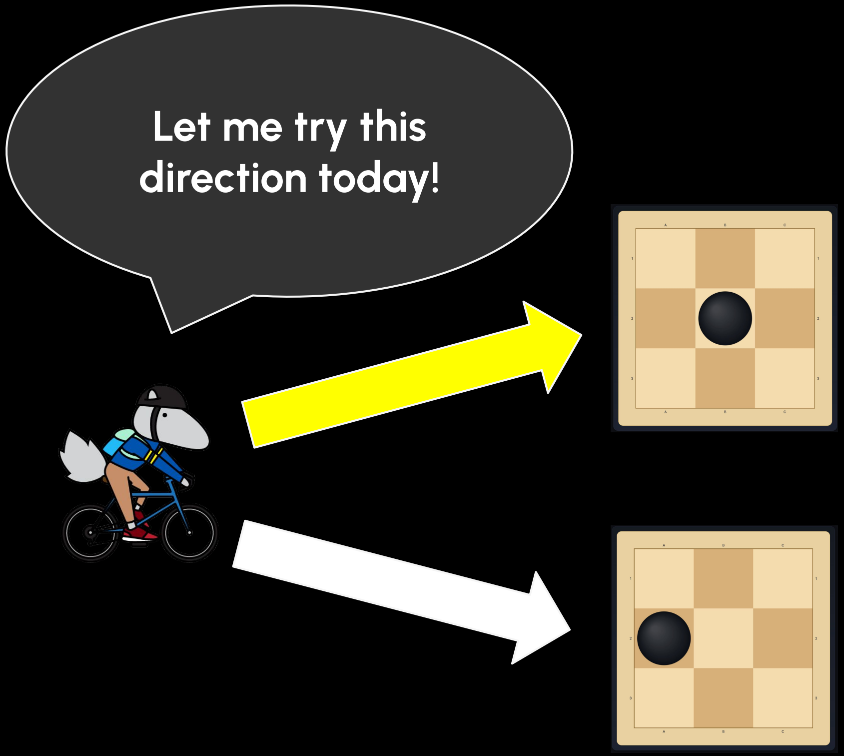
For n=47, 49, 51 and n>52, we do not know the largest number of points on an n x n grid with no three in line.

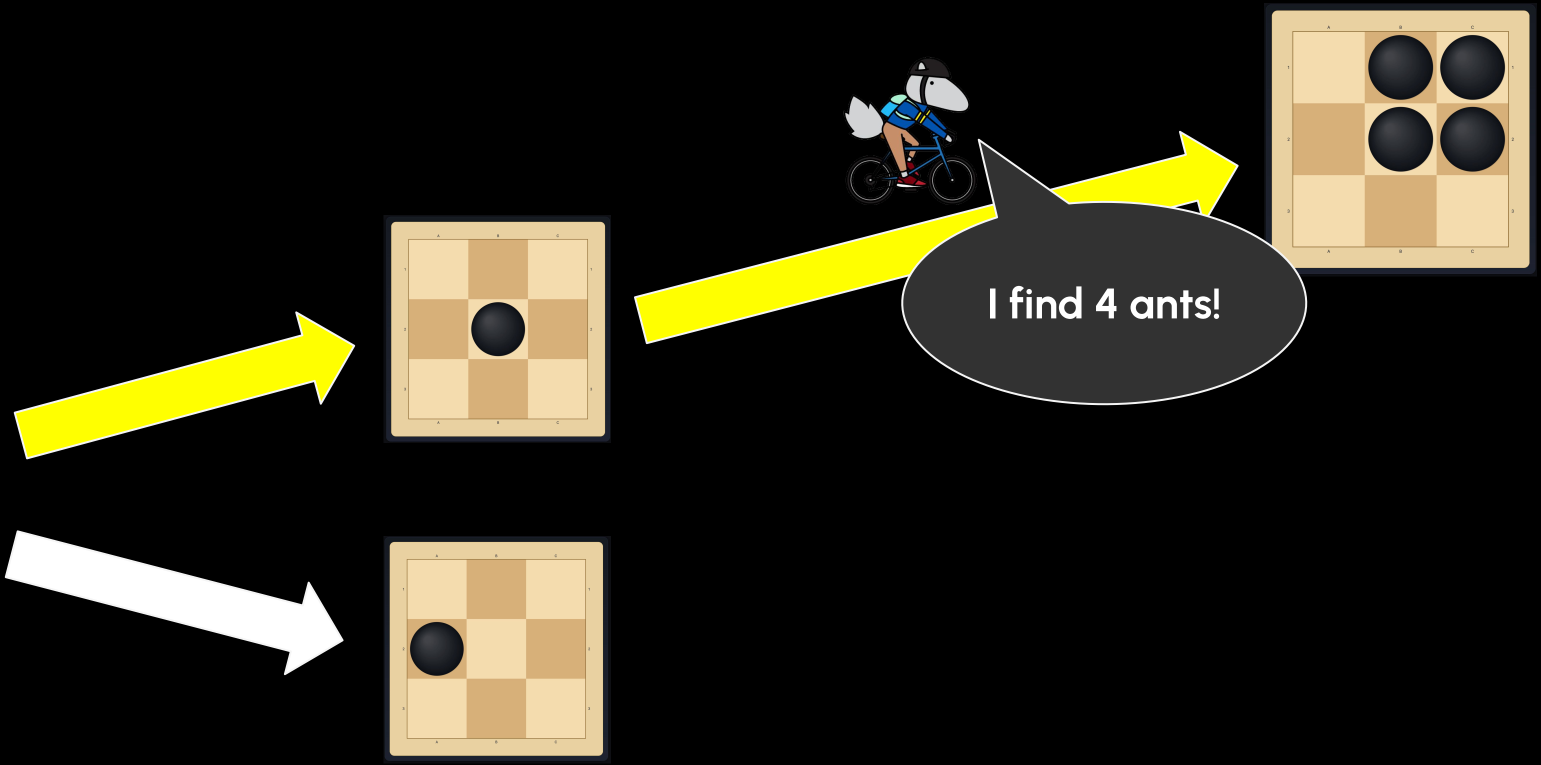
Can we use computer algorithms to find more than 1.5n points in these “big grids”?

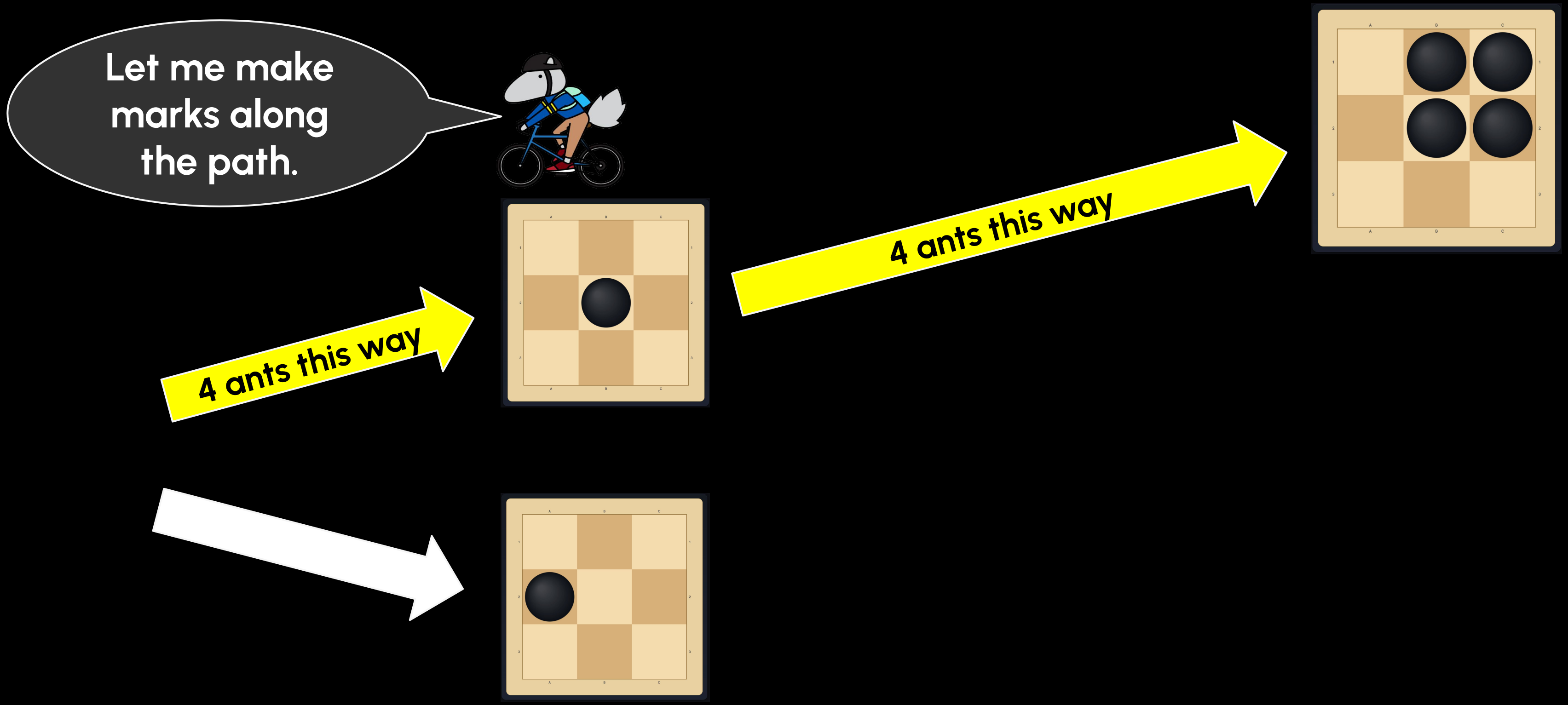
We adapt the Monte Carlo Tree Search (MCTS).

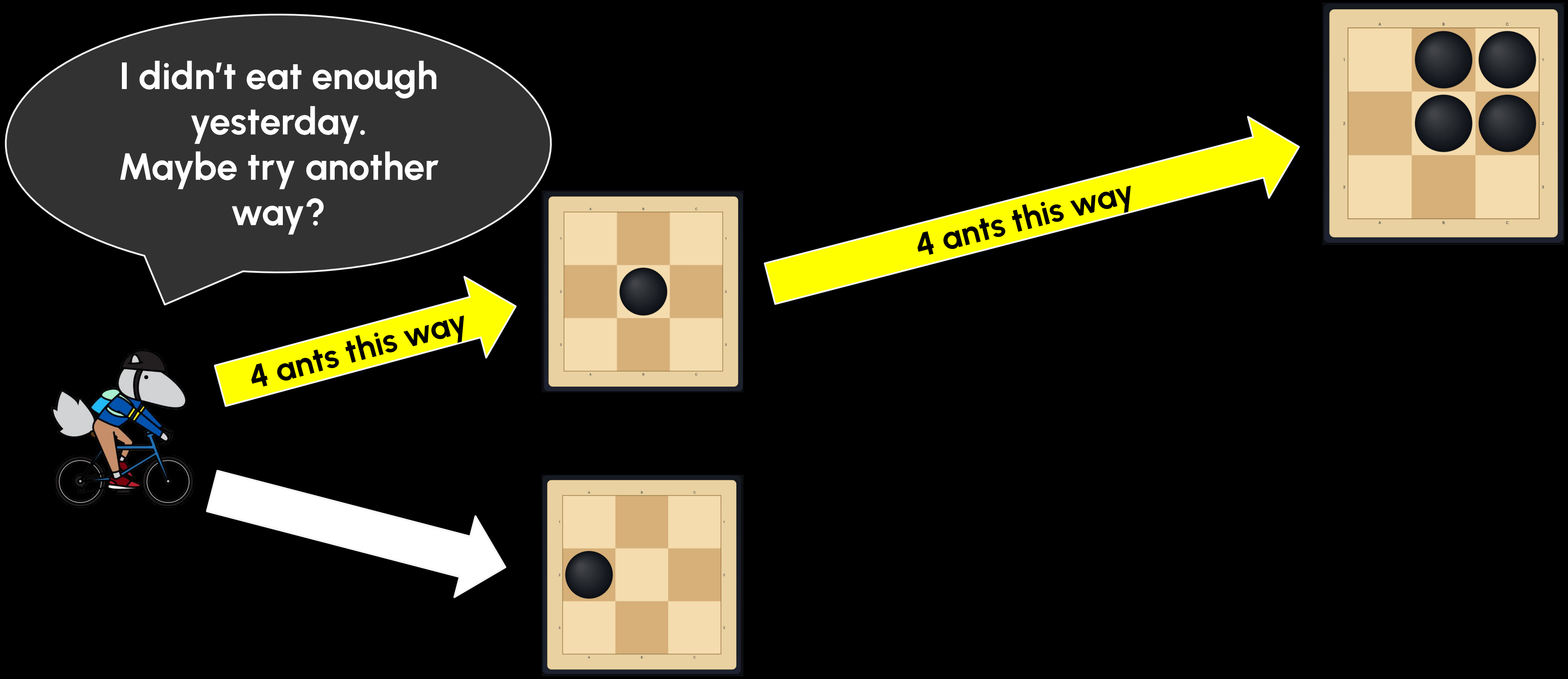
- What is MCTS?

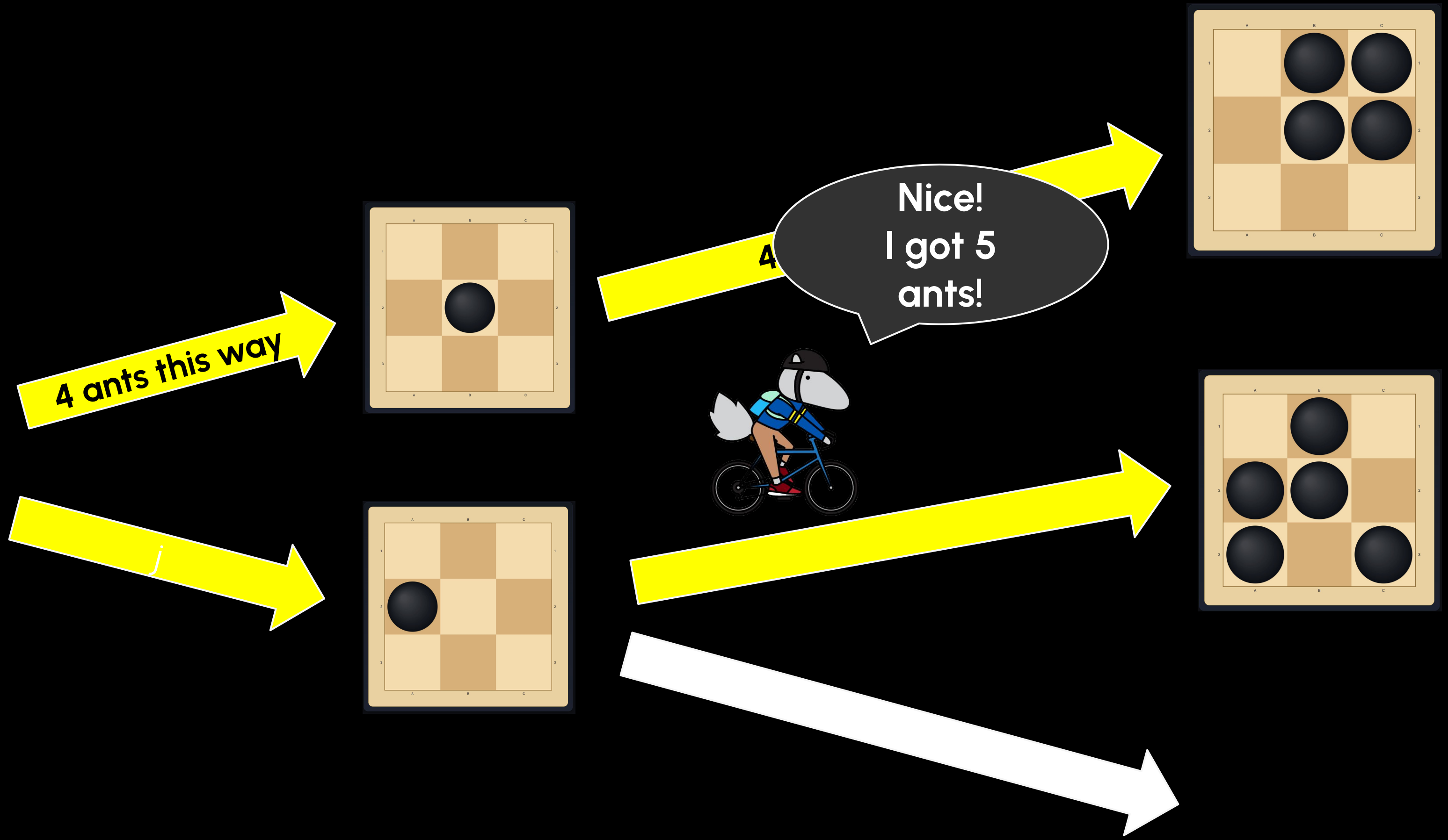

I'm hungry! Which way should I go to find more ants?

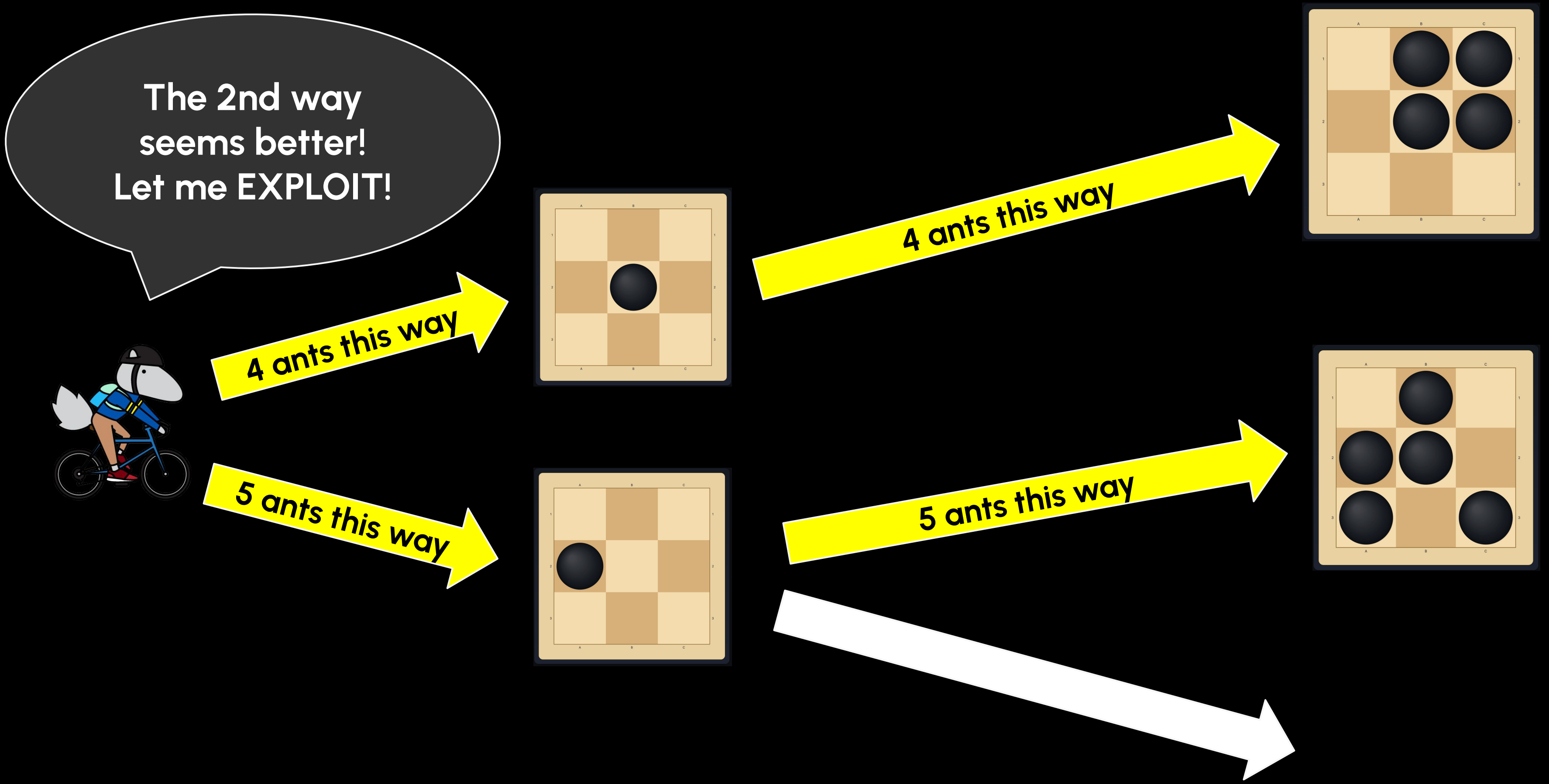

Let me try this direction today!

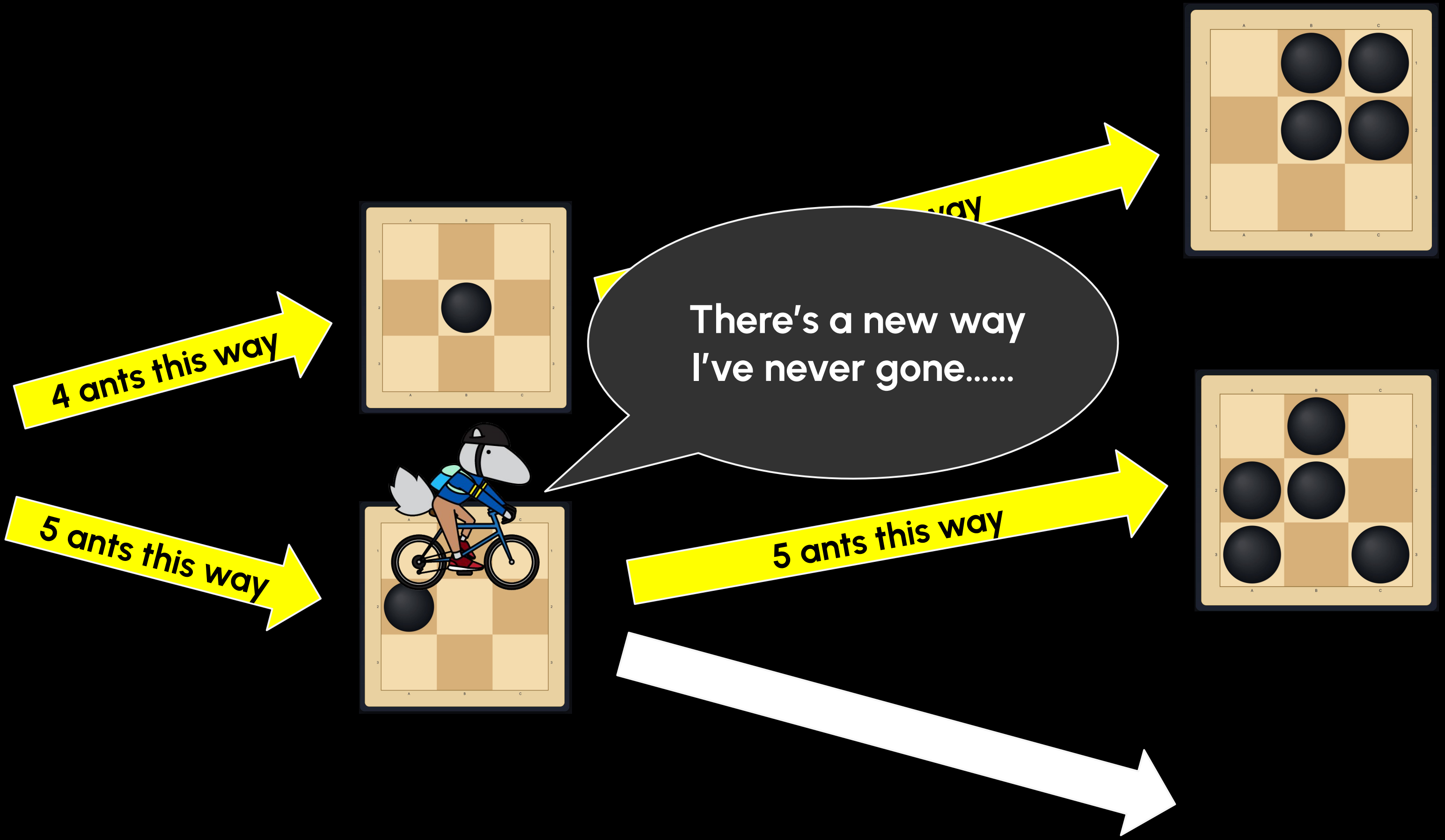

I find 4 ants!

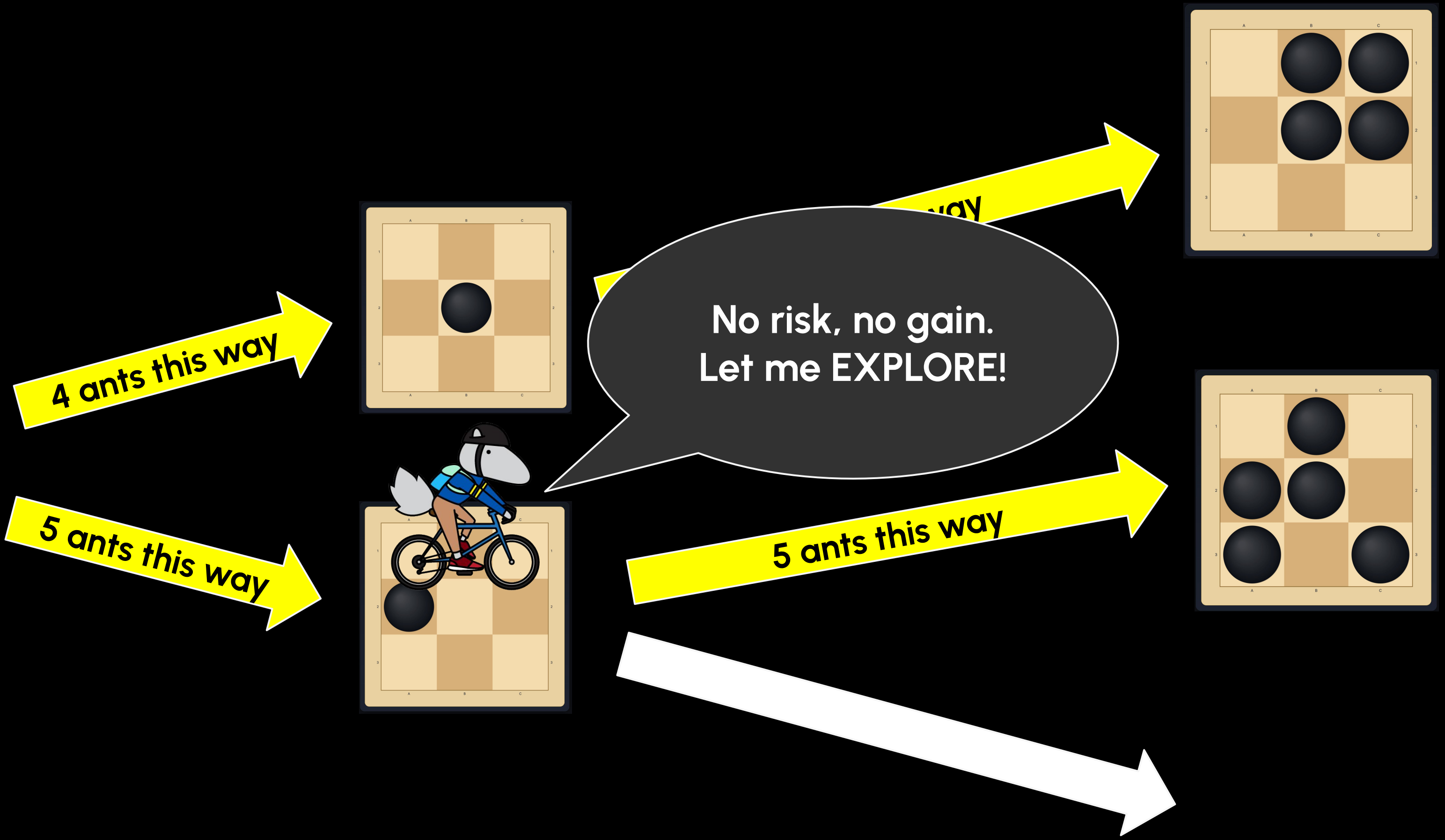

Let me make marks along the path.
4 ants this way
4 ants this way

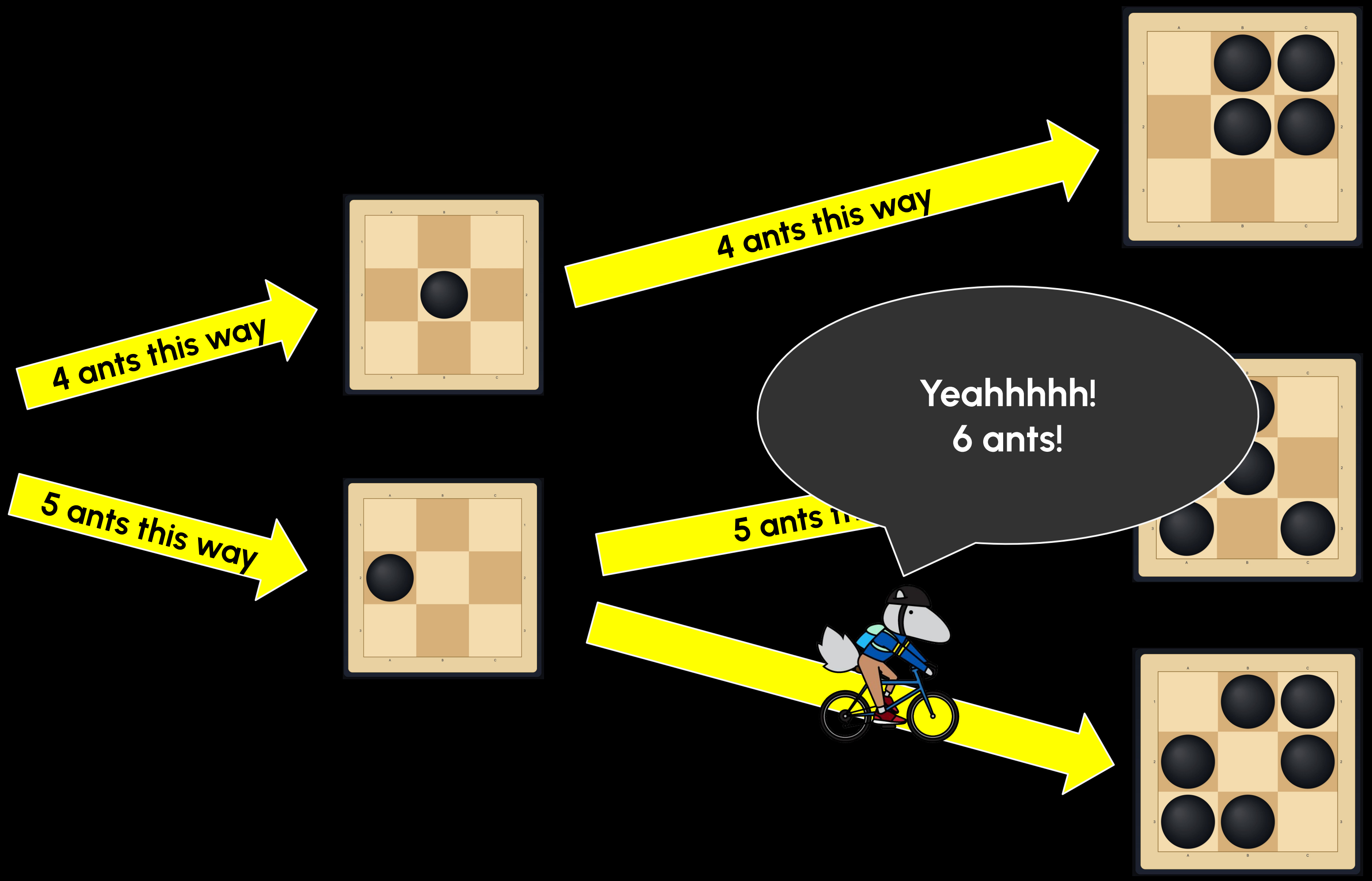

I didn't eat enough yesterday.
Maybe try another way?
4 ants this way
4 ants this way


4 ants this way
4
Nice!
I got 5
ants!


The 2nd way
seems better!
Let me EXPLOIT!
4 ants this way
5 ants this way
4 ants this way
5 ants this way


4 ants this way
5 ants this way
There's a new way
I've never gone……
5 ants this way


4 ants this way
5 ants this way
No risk, no gain.
Let me EXPLORE!
5 ants this way

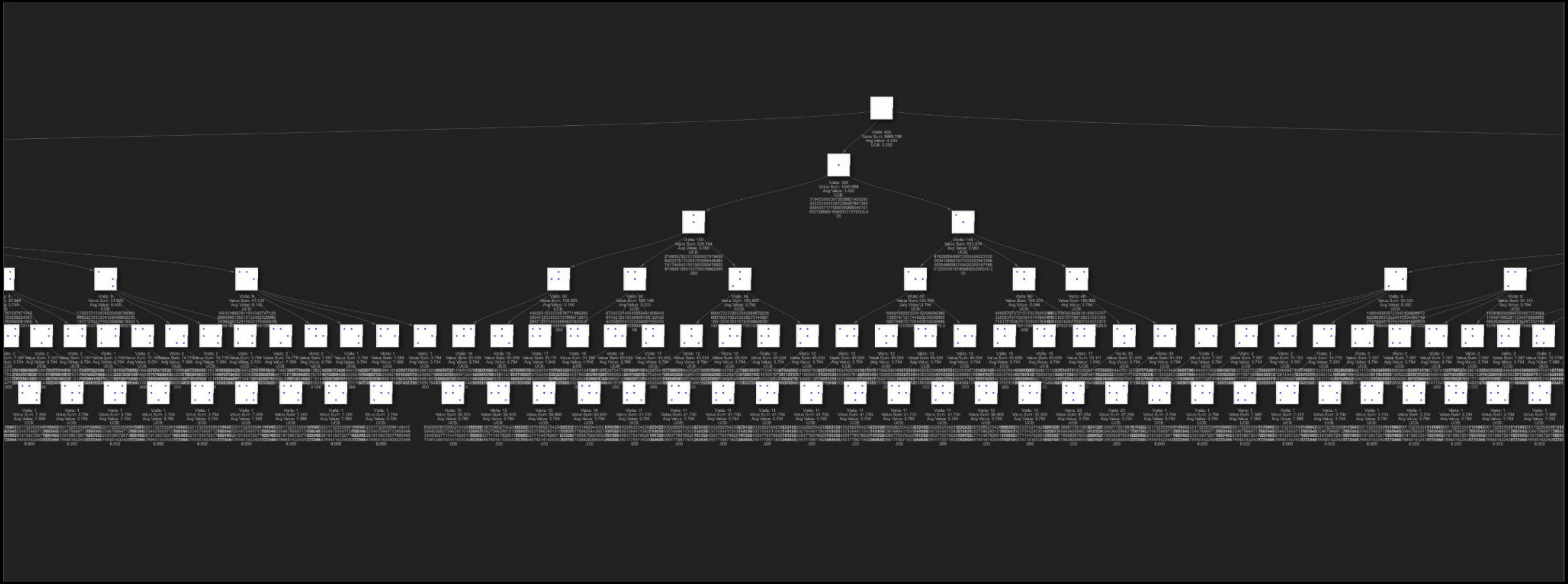

4 ants this way
5 ants this way
4 ants this way
5 ants th
Yeahhhhh!
6 ants!

Optimized Reward Function:

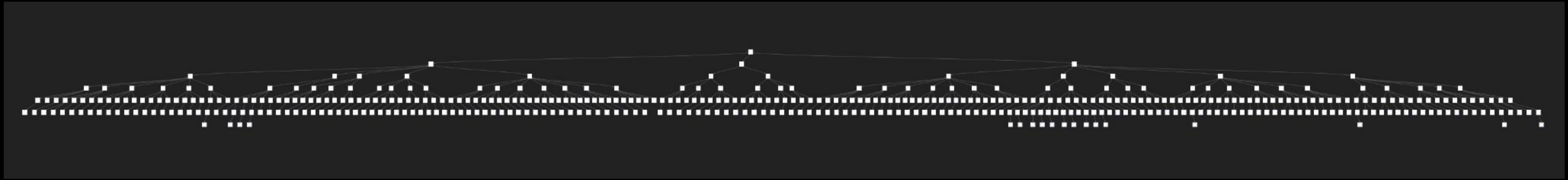
$$\text{Value}(s) = e^{2 \cdot \frac{number\_of\_points - n}{n}}$$

Q: Why is the balance between exploration and exploitation so important?

- If we always explore, the tree will be like this (443 nodes):



The tree is actually extremely large if we zoom out……

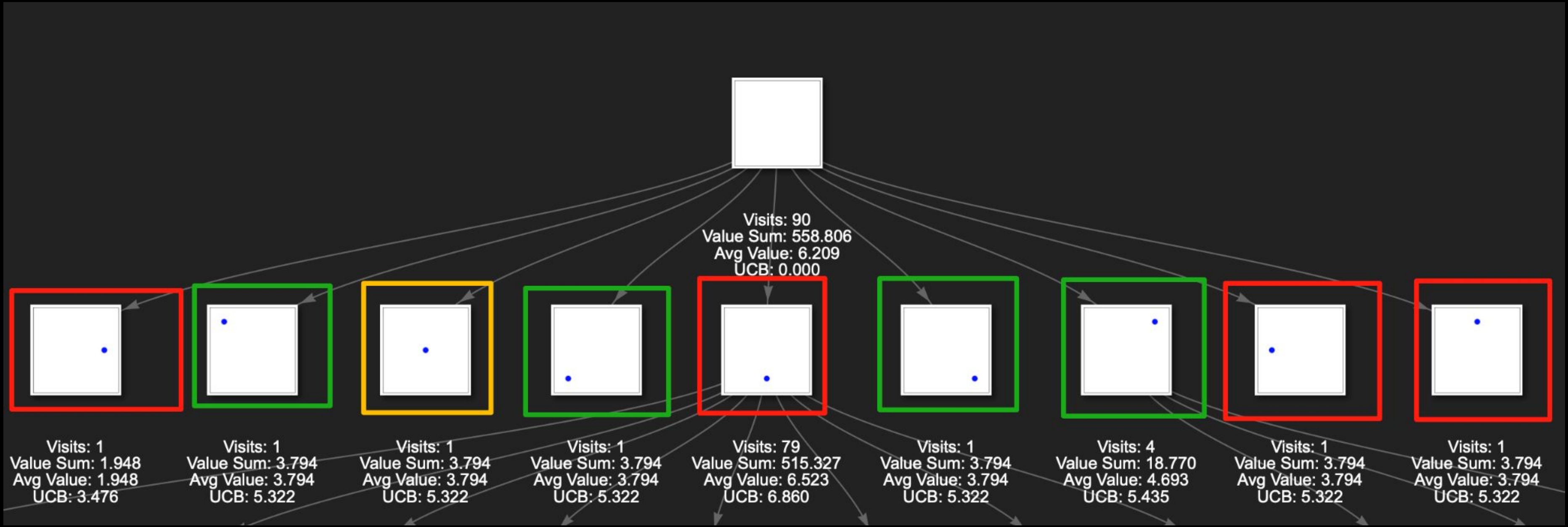


Optimized Upper Confidence Bound (UCB) for Trees with Exploration Decay

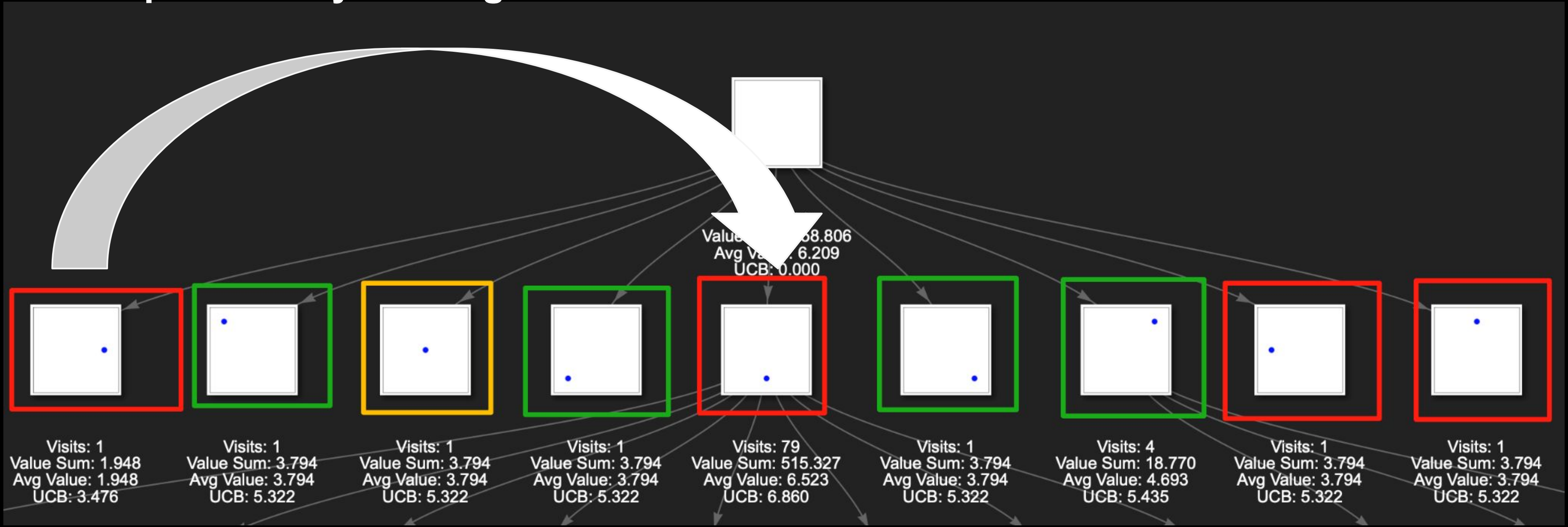
$$\text{UCB}(child, iter) = \frac{\text{value\_sum}_{child}}{\text{visit\_count}_{child}} + \left[C \cdot \text{exploration\_decay}\left(\frac{iter}{num\_searches}\right)\right] \cdot \sqrt{\frac{\ln(\text{visit\_count}_{parent})}{\text{visit\_count}_{child}}}$$

Where

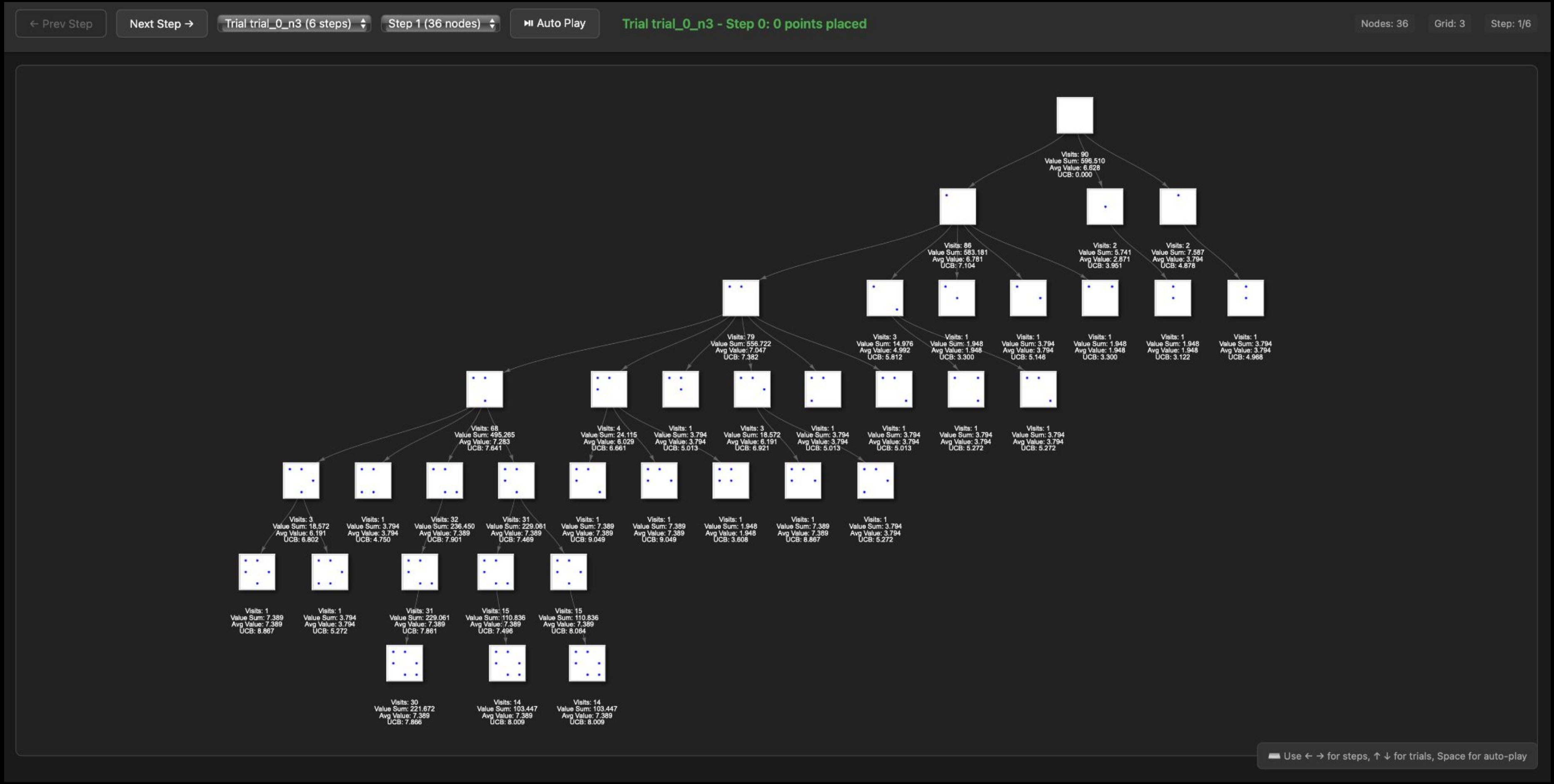
$$\text{exploration\_decay}(x) = 1 - 0.85\sqrt{x}$$


Visits: 90
Value Sum: 558.806
Avg Value: 6.209
UCB: 0.000
Visits: 1
Value Sum: 1.948
Avg Value: 1.948
UCB: 3.476
Visits: 1
Value Sum: 3.794
Avg Value: 3.794
UCB: 5.322
Visits: 1
Value Sum: 3.794
Avg Value: 3.794
UCB: 5.322
Visits: 1
Value Sum: 3.794
Avg Value: 3.794
UCB: 5.322
Visits: 79
Value Sum: 515.327
Avg Value: 6.523
UCB: 6.860
Visits: 1
Value Sum: 3.794
Avg Value: 3.794
UCB: 5.322
Visits: 4
Value Sum: 18.770
Avg Value: 4.693
UCB: 5.435
Visits: 1
Value Sum: 3.794
Avg Value: 3.794
UCB: 5.322
Visits: 1
Value Sum: 3.794
Avg Value: 3.794
UCB: 5.322

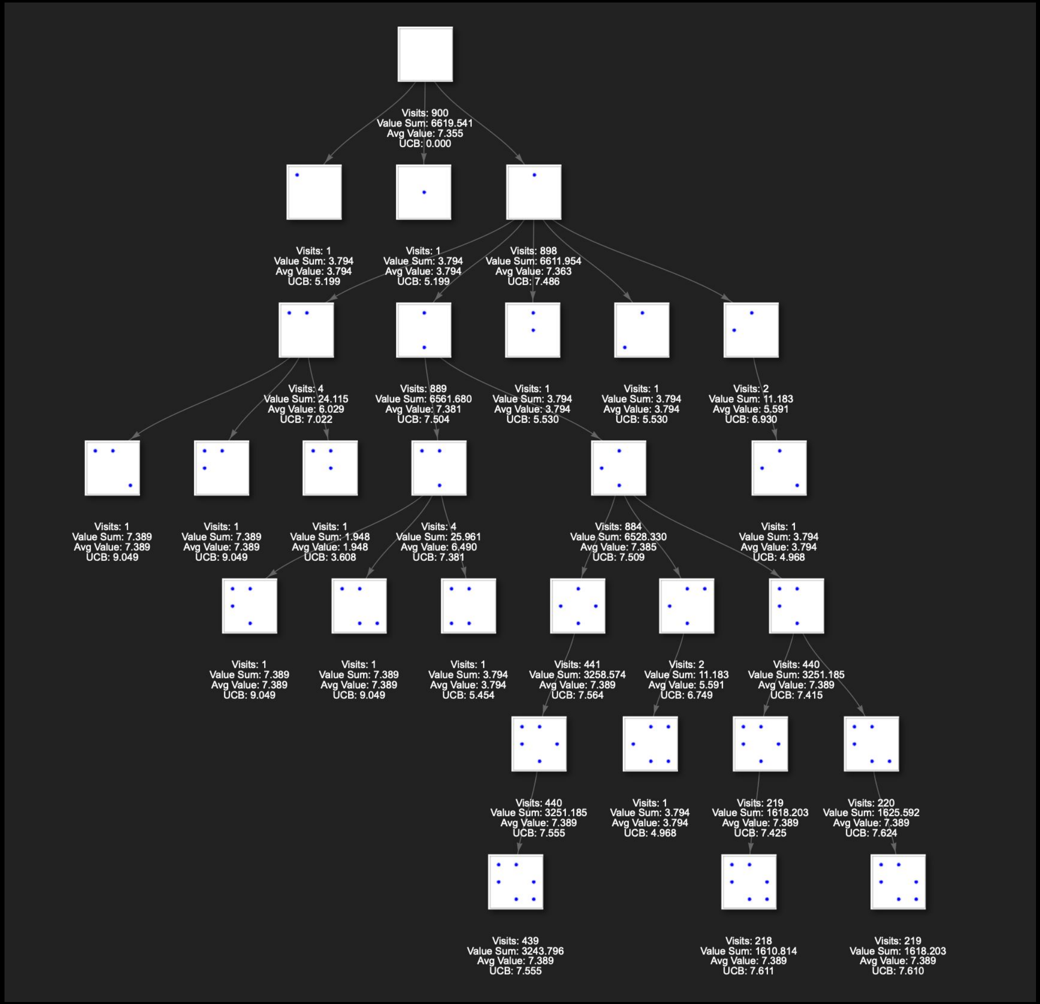
Equivalent by rotating 90° clockwise



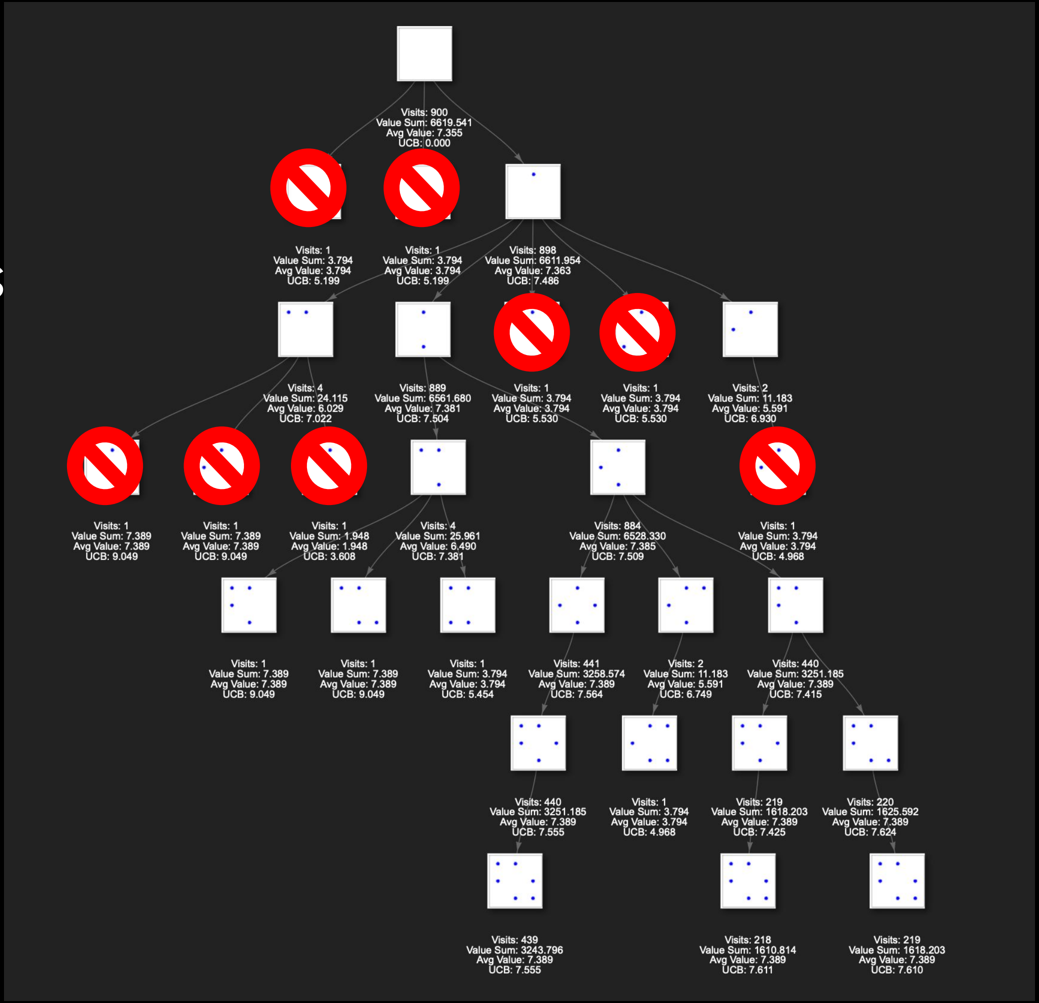
76 -> 36 nodes by symmetric action space restriction



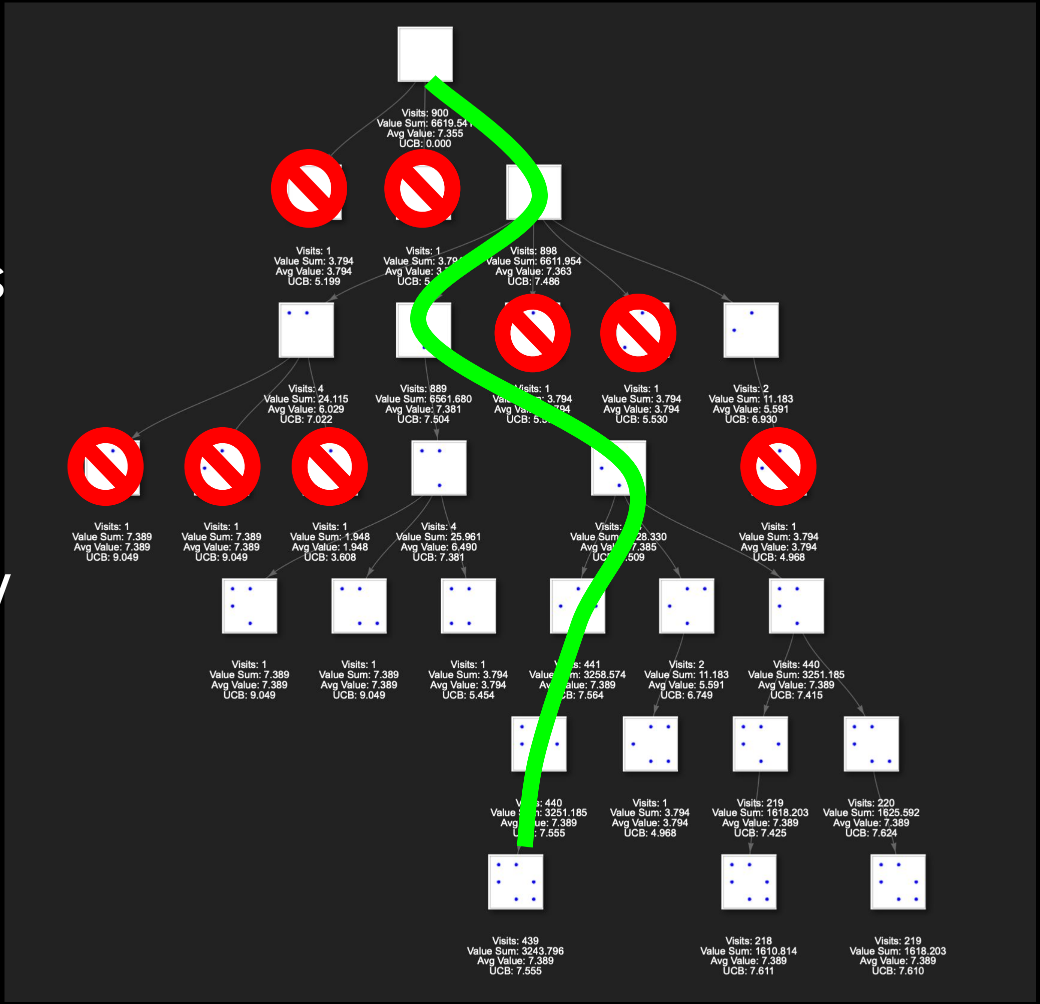
- With optimized exploration-exploitation balance,



- With optimized exploration-exploitation balance,
- it cleverly gives up actions that do not seem to lead to good outcomes



- With optimized exploration-exploitation balance,
- it cleverly gives up actions that do not seem to lead to good outcomes
- and outputs an optimal solution with a tree of only 28 nodes:



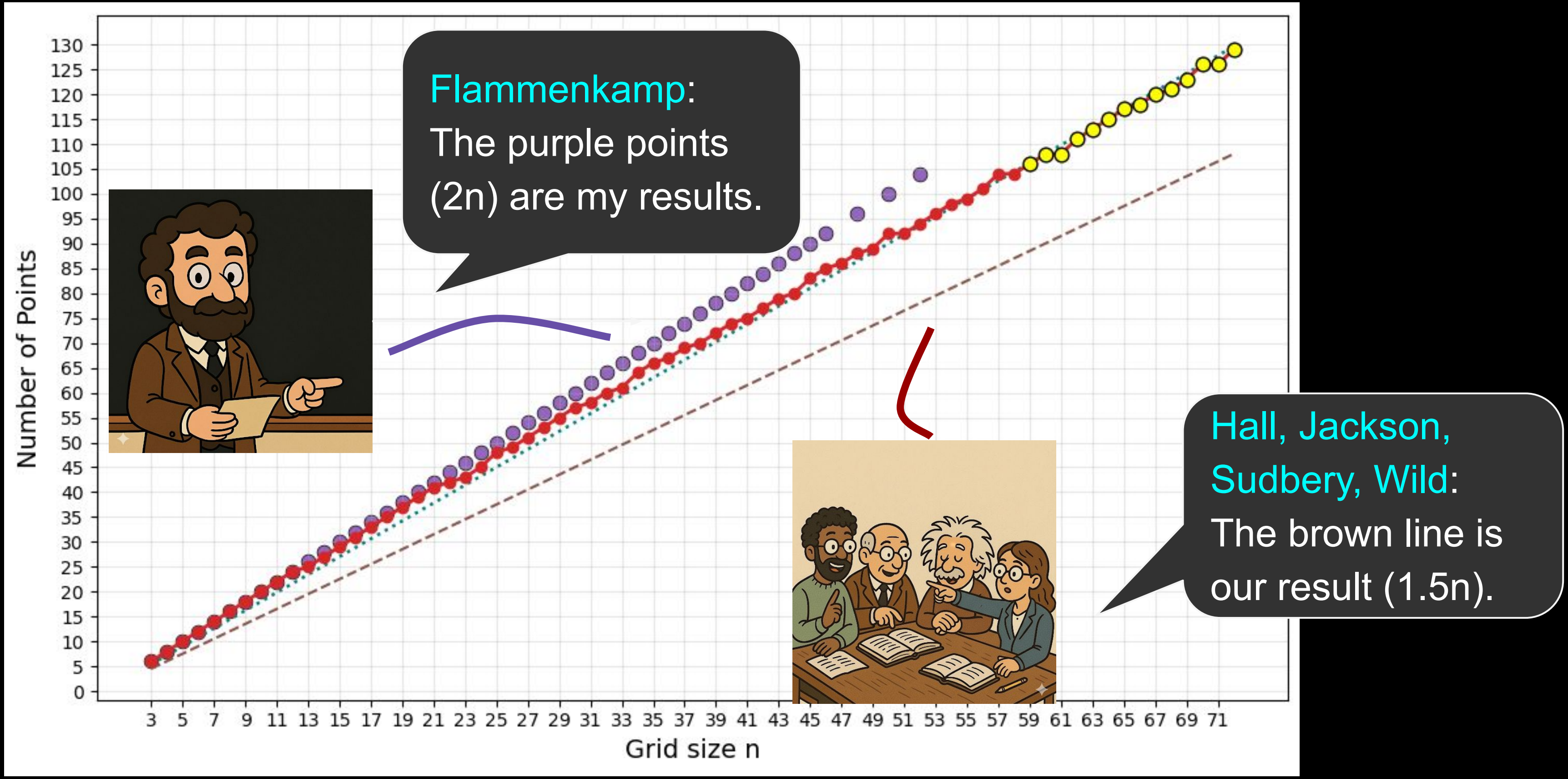
Algorithm performance optimization:

~2000x faster compared with early version

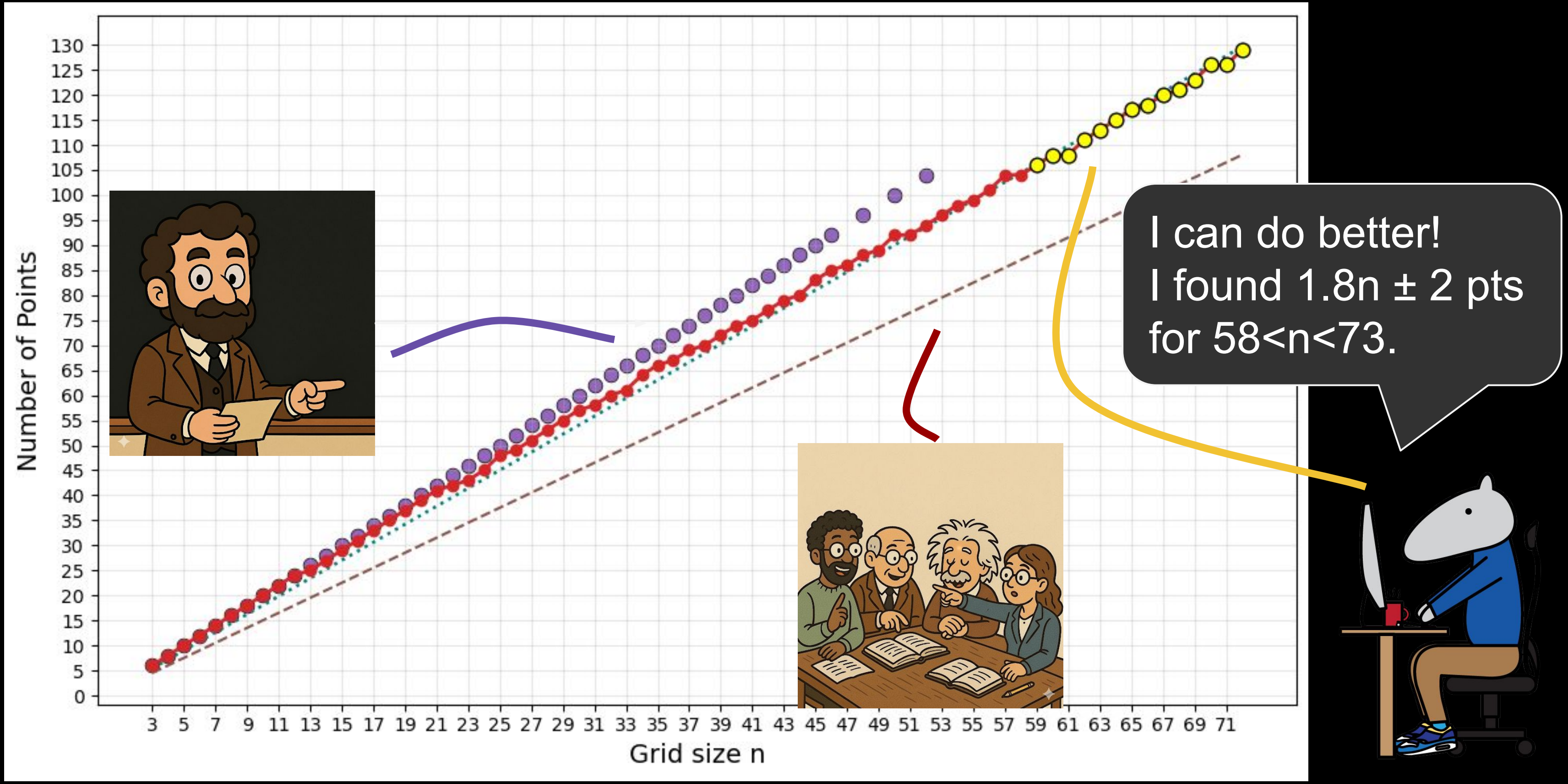
```
evaluate(args)
20.3s

--------------------------
Number of points: 0
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
4%  147/4000 [00:20<09:13, 6.97it/s]
```

```
=== Starting experiments for grid size n=20 ===
Trial 1/1 for n=20... ----------------------------
Number of points: 0
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
100%|████████████████████████████████| 4000/4000 [00:00<00:00, 15386.80it/s]
```

Results for largest number of points:

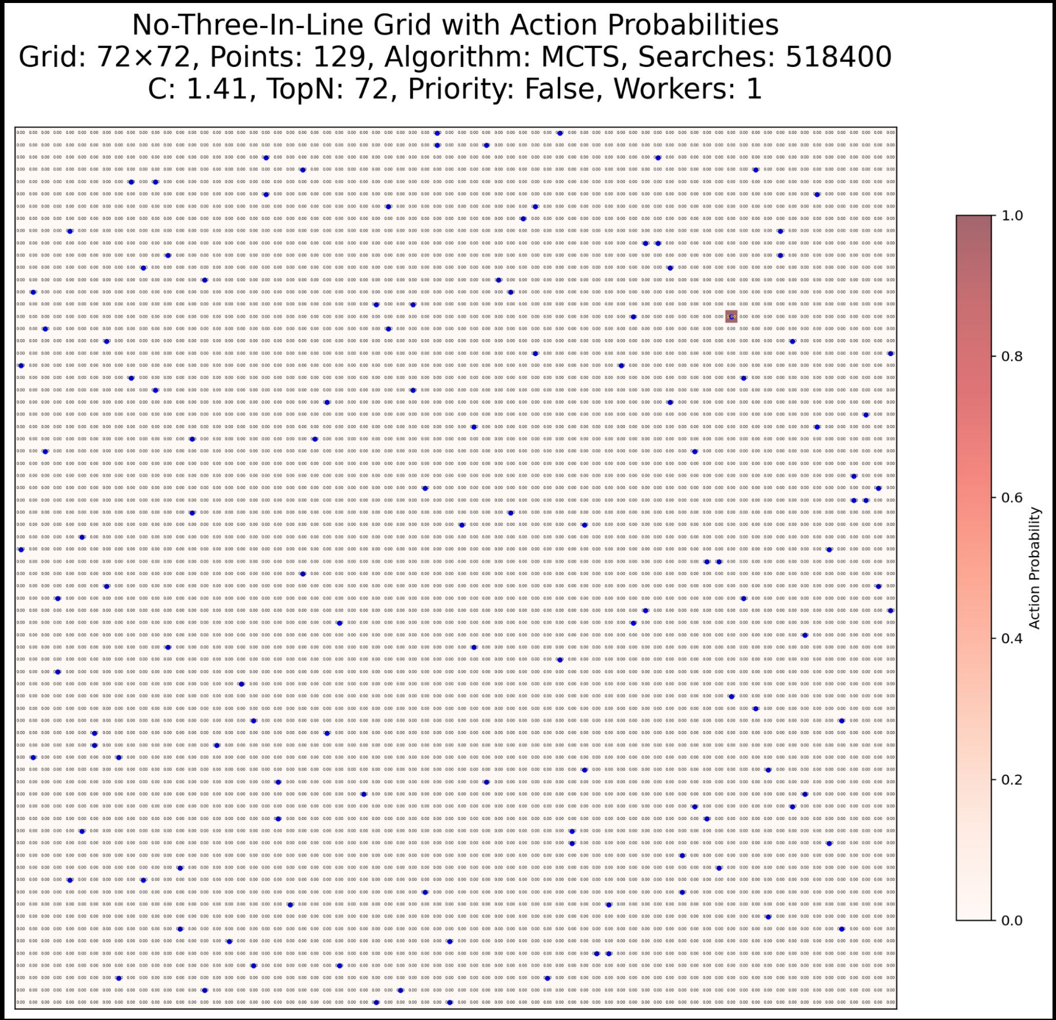


The yellow points surpassed previous best results:



Example:

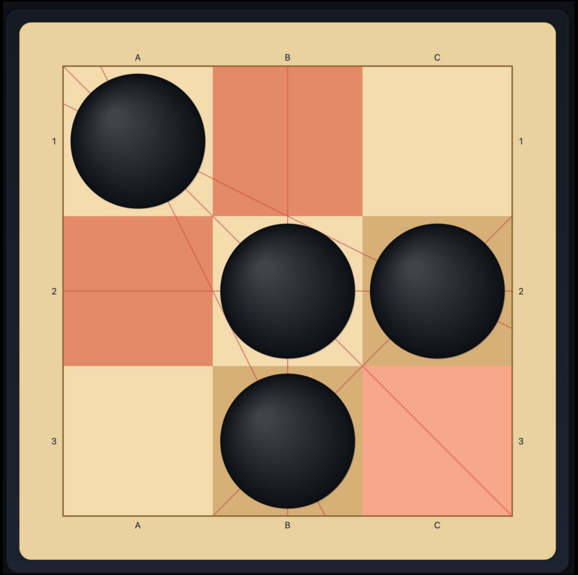
129 no-three-in-line points on a **72 x 72** grid



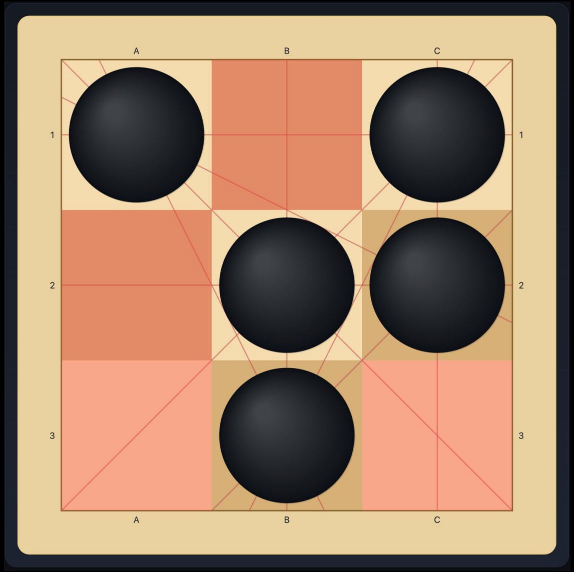
Another way to phrase this problem
-What is a “Complete Set”?

Short informal definition:
Putting any more point into the set of points would cause 3 in a line.

Not "complete"



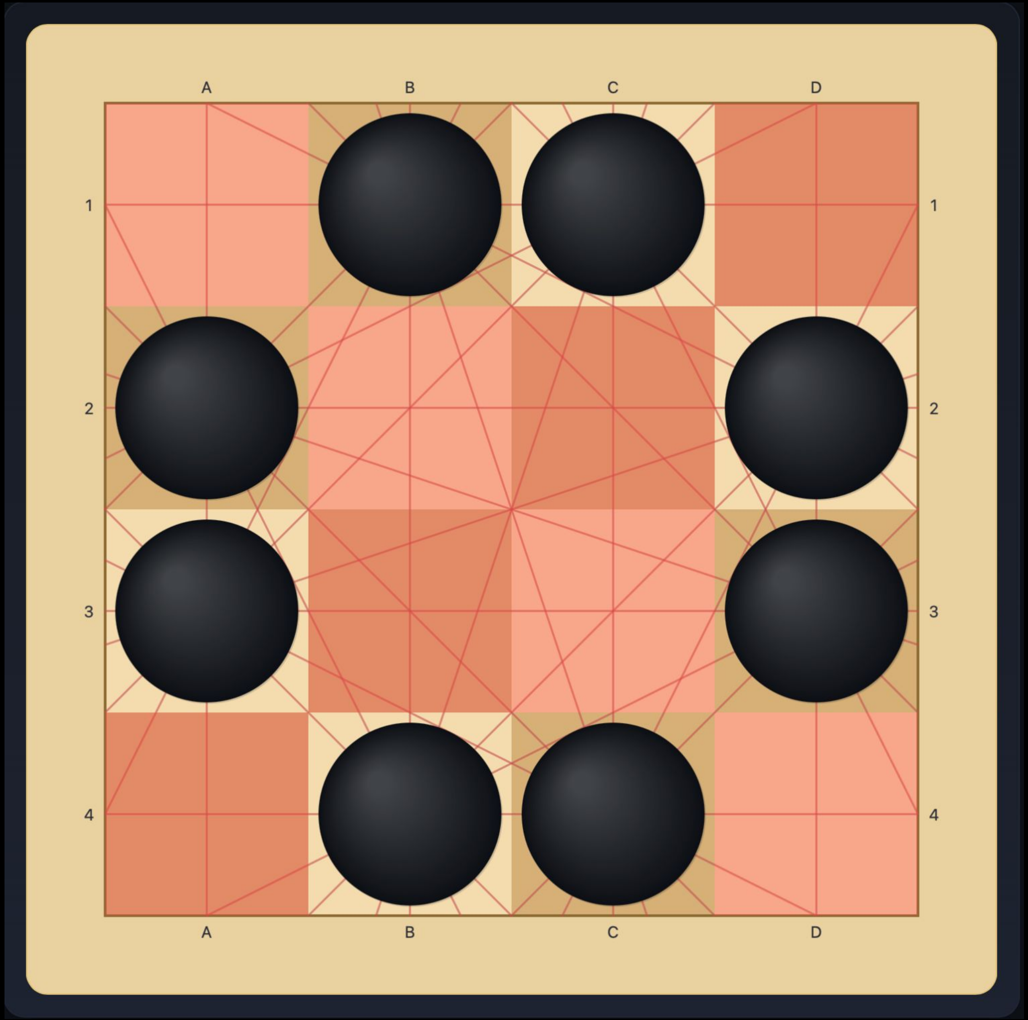
"complete"



Try Yourself!

Example:

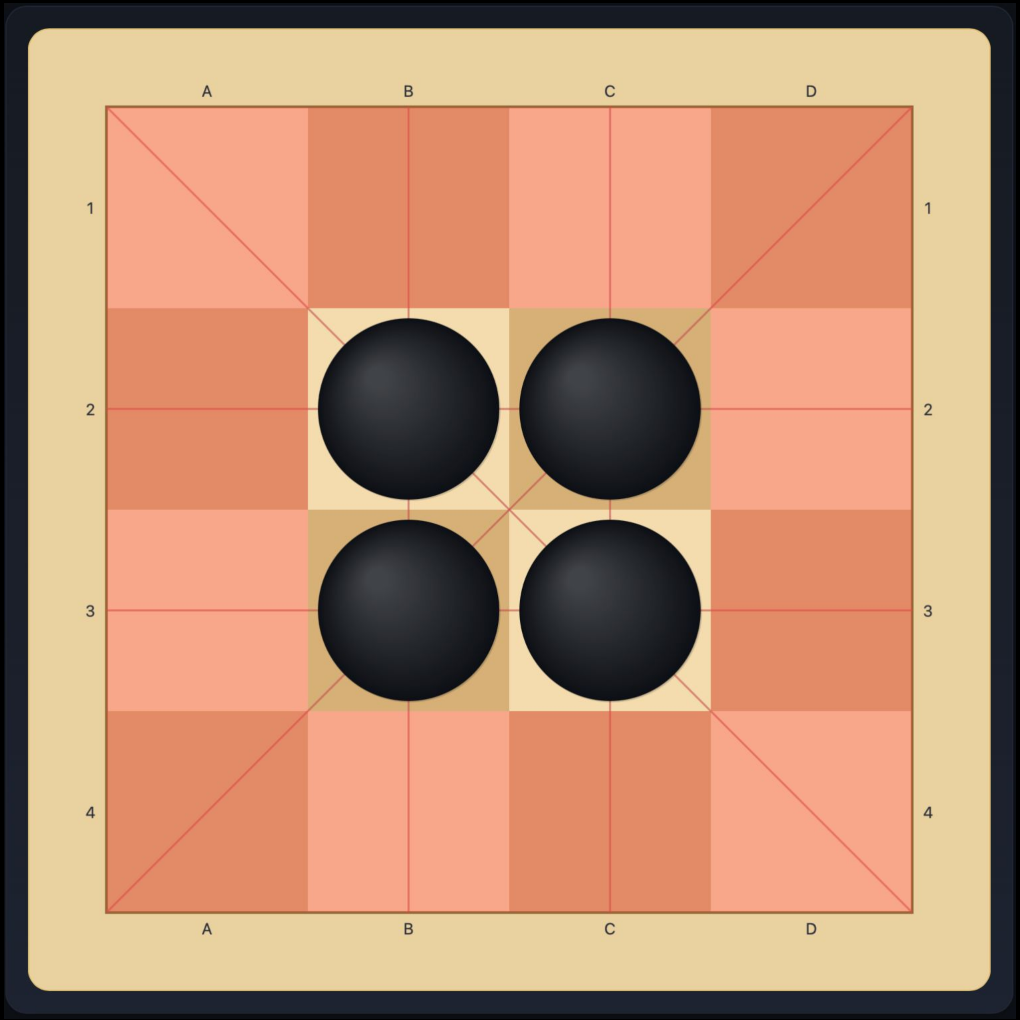
Largest
Complete
Set
In 4 by 4



Try Yourself!

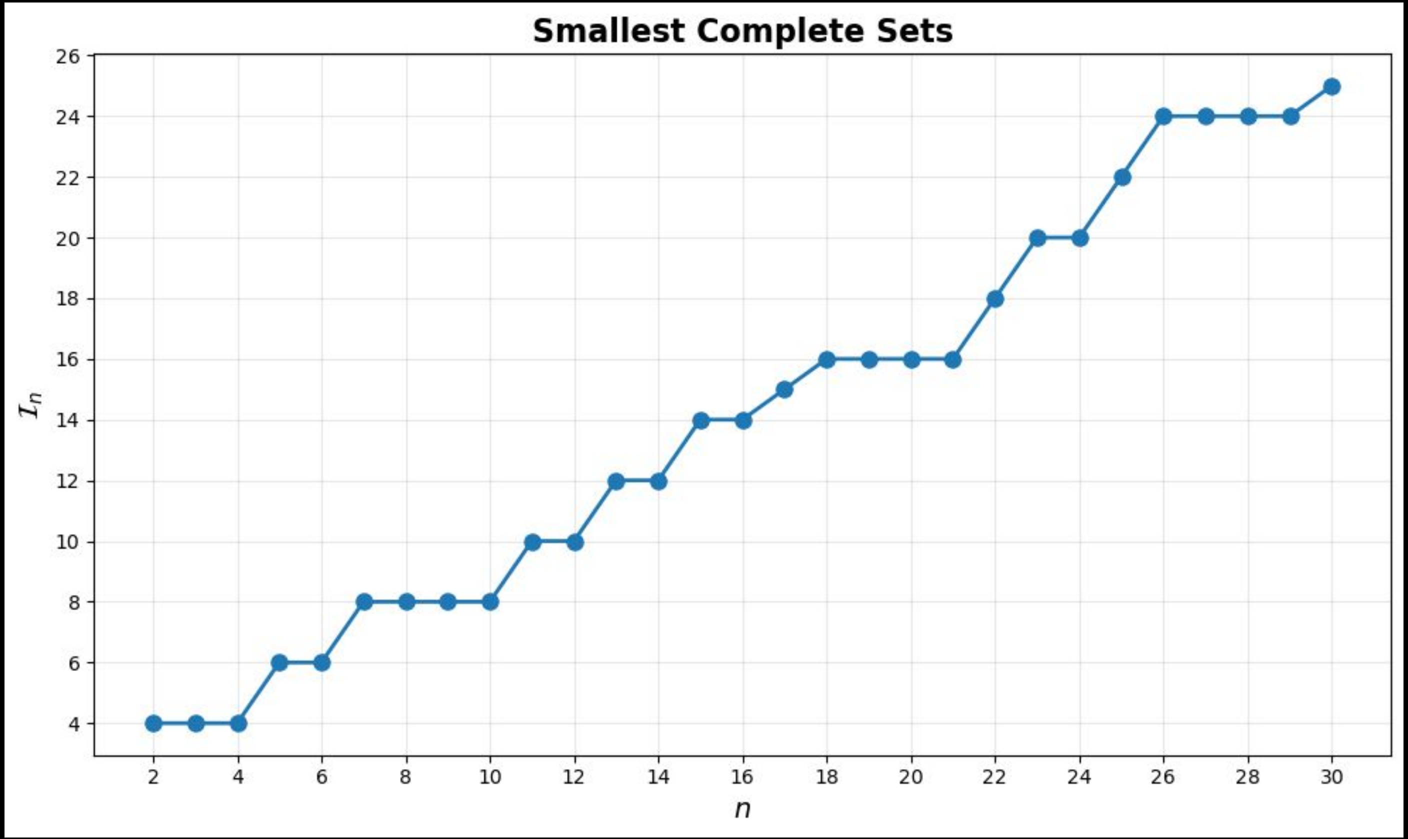
Example:

Smallest
Complete
Set
In 4 by 4



Try Yourself!

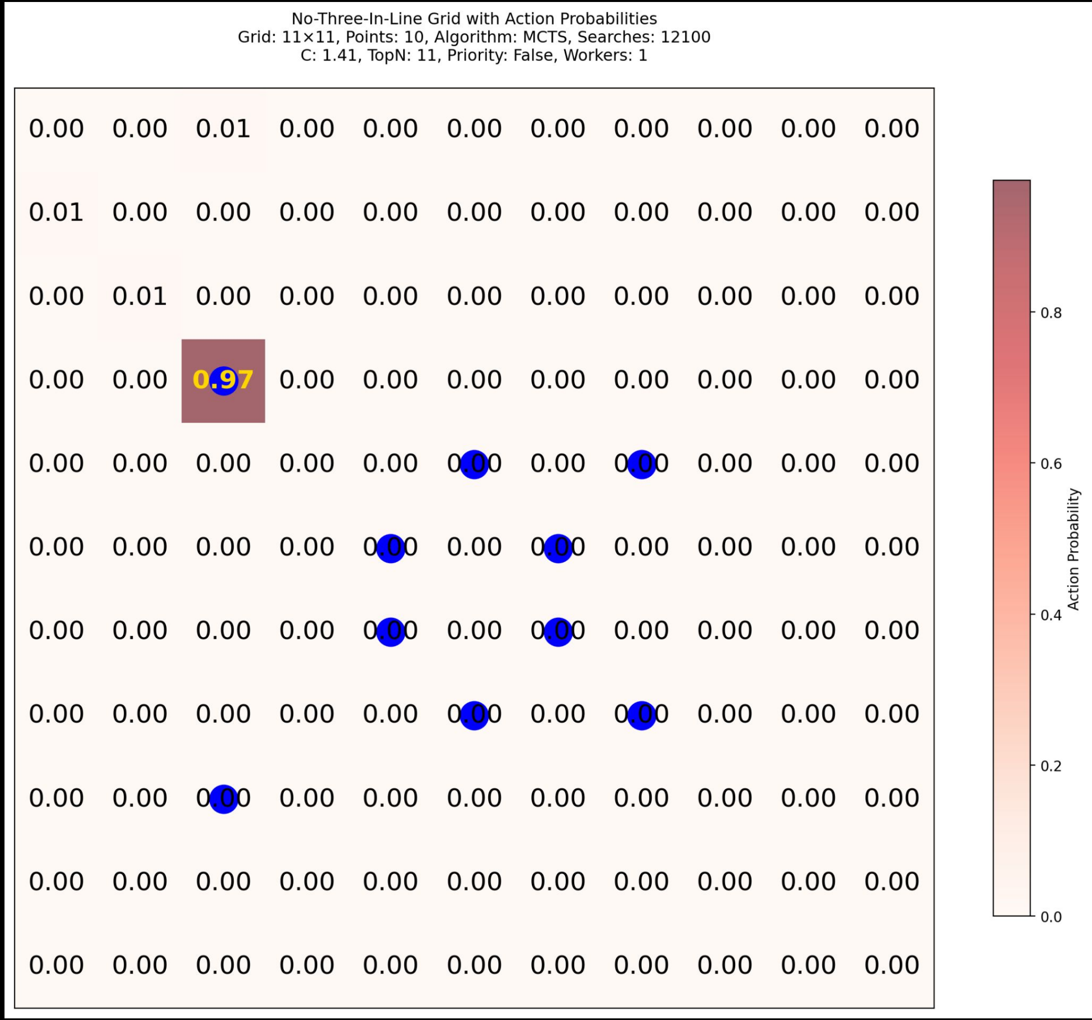
SOTA Results for Smallest Complete Set (Aichholzer et al.):

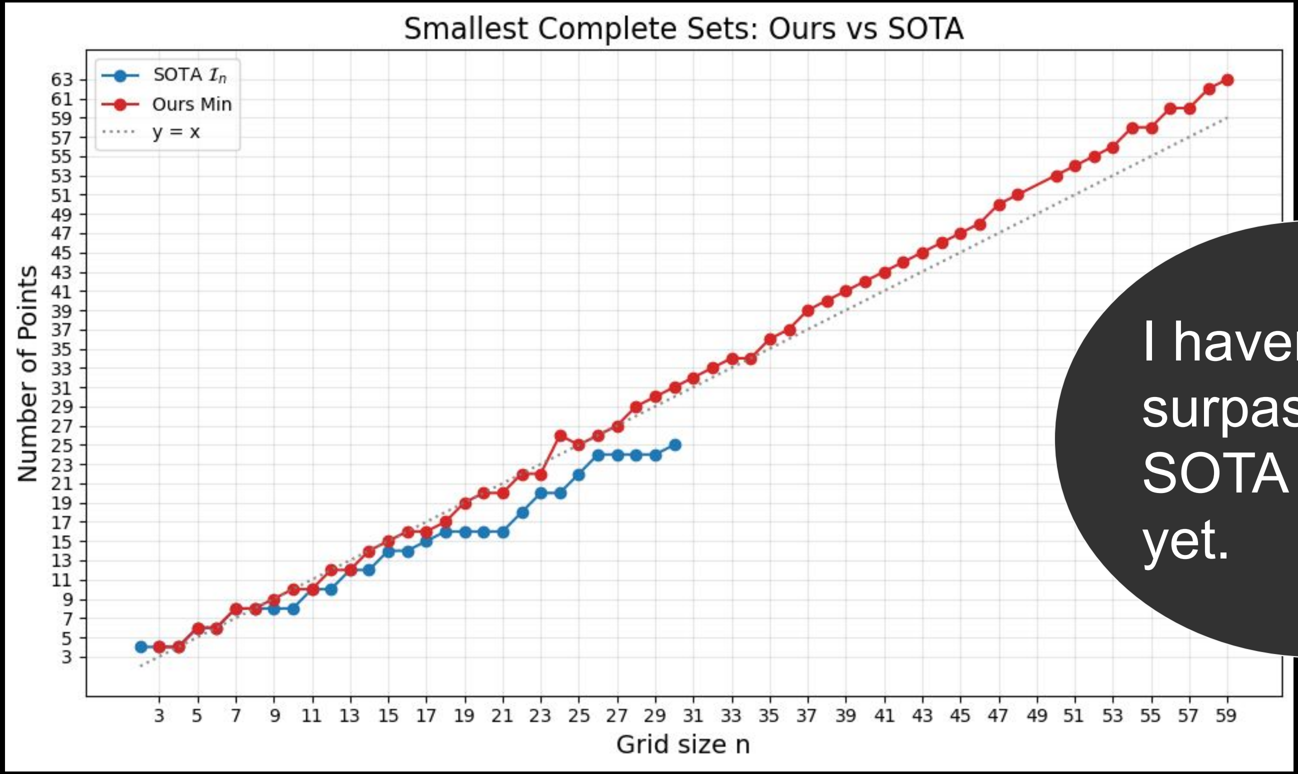


We just "negate" the Reward Function:

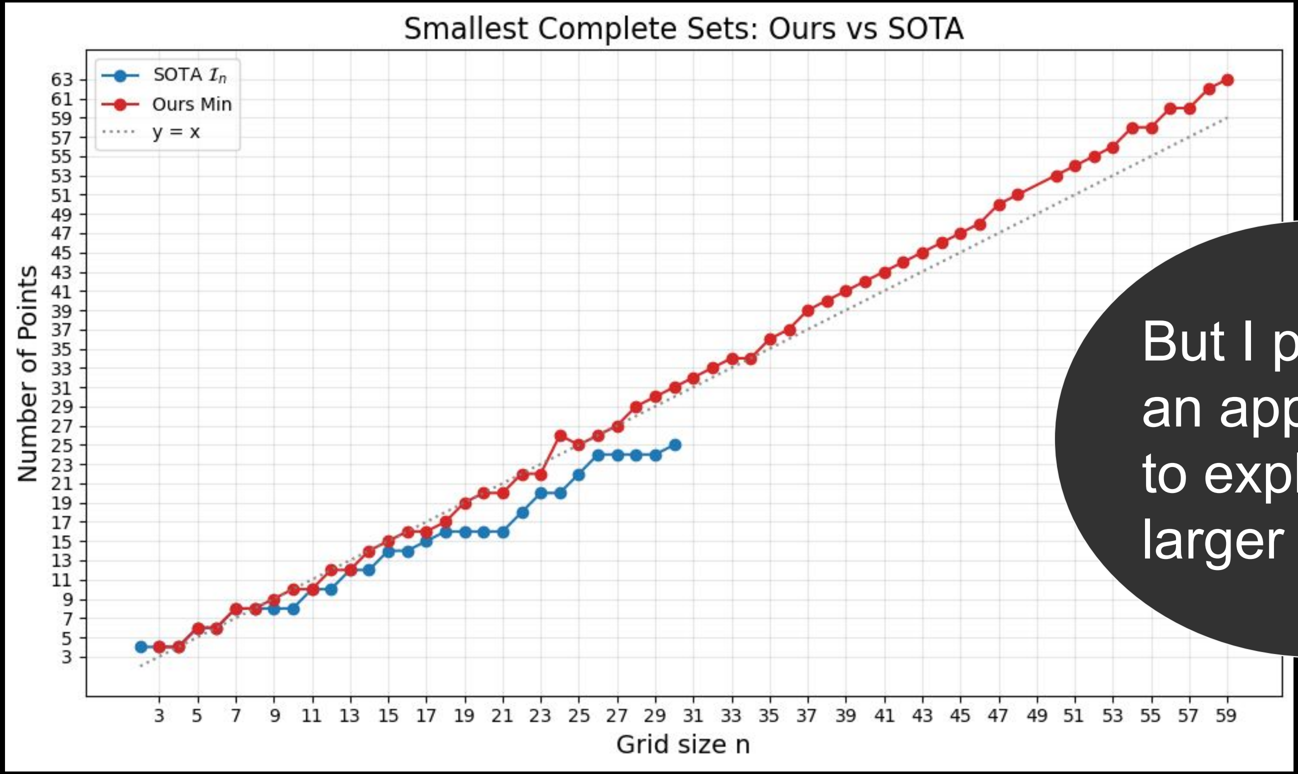
$$\text{Value}(s) = e^{2 \cdot \frac{number\_of\_points - n}{n}}$$

$$\text{Value}(s) = e^{2 \cdot \frac{n - number\_of\_points}{n}}$$


No-Three-In-Line Grid with Action Probabilities
Grid: 11×11, Points: 10, Algorithm: MCTS, Searches: 12100
C: 1.41, TopN: 11, Priority: False, Workers: 1
0.00 0.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.97 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
Action Probability
0.0
0.2
0.4
0.6
0.8



I haven’t surpassed the SOTA result yet.



But I propose an approach to explore larger grids.

Future Work

1. Reward Function Optimization
2. Single Tree Search
3. Graph Search-Merge Similar States
4. Incorporating Neural Networks
5. LLM-based Code Optimization (e.g. Alpha Evolve)

Thank you

Play it yourself!

Challenge:

Question of 100+ years.
Largest number of points on a 47x47 grid is still unknown.
Maybe you can do it!

Reference

Aichholzer, Oswin, et al. "Geometric Dominating Sets - a Minimum Version of the No-Three-In-Line Problem." *Computational Geometry*, vol. 108, Elsevier, June 2022, p. 101913, https://doi.org/10.1016/j.comgeo.2022.101913.

AlphaProof and AlphaGeometry teams. "AI Achieves Silver-Medal Standard Solving International Mathematical Olympiad Problems." *Google DeepMind*, 14 May 2024, deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/.

Guy, Richard K., and Patrick A. Kelly. "The No-Three-In-Line Problem." *Canadian Mathematical Bulletin*, vol. 11, 4, 1968, pp. 527–31, https://doi.org/10.4153/CMB-1968-062-3.

Hall, R. R., et al. "Some Advances in the No-Three-In-Line Problem." *Journal of Combinatorial Theory, Series A*, vol. 18, no. 3, May 1975, pp. 336–41, https://doi.org/10.1016/00973165(75)900436.

Henry Ernest Dudeney. *Amusements in Mathematics*. Outlook Verlag, 2020.

Kaplan, Nathan. *Dudeney's No-Three-In-Line Problem (Slides)*. 21 Oct. 2020.

Lan, Li-Cheng, et al. "Multiple Policy Value Monte Carlo Tree Search." *ArXiv.org*, 2019, arxiv.org/abs/1905.13521. Accessed 14 Aug. 2025.

Leurent, Edouard, et al. "Monte-Carlo Graph Search: The Value of Merging Similar States Odalric-Ambrym Maillard." *Proceedings of Machine Learning Research*, vol. 129, 2020, pp. 577–92, proceedings.mlr.press/v129/leurent20a/leurent20a.pdf. Accessed 14 Aug. 2025.

Novikov, Alexander, et al. "AlphaEvolve: A Coding Agent for Scientific and Algorithmic Discovery." *ArXiv.org*, 2025, arxiv.org/abs/2506.13131.

Romera-Paredes, Bernardino, et al. "Mathematical Discoveries from Program Search with Large Language Models." *Nature*, vol. 625, no. 7995, 2024, pp. 468–75, https://doi.org/10.1038/s41586023069246.

Silver, David, et al. "Mastering the Game of Go without Human Knowledge." *Nature*, vol. 550, no. 7676, Oct. 2017, pp. 354–59, https://doi.org/10.1038/nature24270.

Vaswani, Ashish, et al. *Attention Is All You Need*. Edited by I Guyon et al., vol. 30, Curran Associates, Inc., 2017, proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Image sources: UCI Student Affairs graphics, Wikipedia, Chat GPT and Gemini Image Generation