Reinforcement Learning for finding large sets in extremal combinatorics

Karan Srivastava | Specialty Exam

Research supported in part by NSF Award DMS-2023239

Under supervision of Jordan Ellenberg (PhD Advisor) and Amy Cochran (IFDS Mentor)

Motivation

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › articles › article

Article | Open Access | Published: 05 October 2022

Discovering faster matrix multiplication algorithms with reinforcement learning

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis & Pushmeet Kohli

Nature **610**, 47–53 (2022) | Cite this article

521k Accesses | **45** Citations | **3708** Altmetric | Metrics

Abstract

Improving the efficiency of algorithms for fundamental computations can have a widespread impact, as it can affect the overall speed of a large amount of computations. Matrix multiplication is one such primitive task, occurring in many systems—from neural networks to scientific computing routines. The automatic discovery of algorithms using machine learning offers the prospect of reaching beyond human intuition and outperforming the current best human-designed algorithms. However, automating the algorithm discovery procedure is intricate, as the space of possible algorithms is enormous. Here we report a deep reinforcement learning approach based on AlphaZero[1] for discovering efficient and provably correct algorithms for the multiplication of arbitrary matrices. Our agent, AlphaTensor, is trained to play a single-player game where the objective is finding tensor decompositions within a finite factor space. AlphaTensor discovered algorithms that outperform the state-of-the-art complexity for many matrix sizes. Particularly relevant is the case of 4×4 matrices in a finite field, where AlphaTensor's algorithm improves on Strassen's two-level algorithm for the

Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

1 Introduction

Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

Recent work shown that ML techniques can be useful for generating examples in math.

Motivation


nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > articles > article

Article | Open Access | Published: 05 October 2022

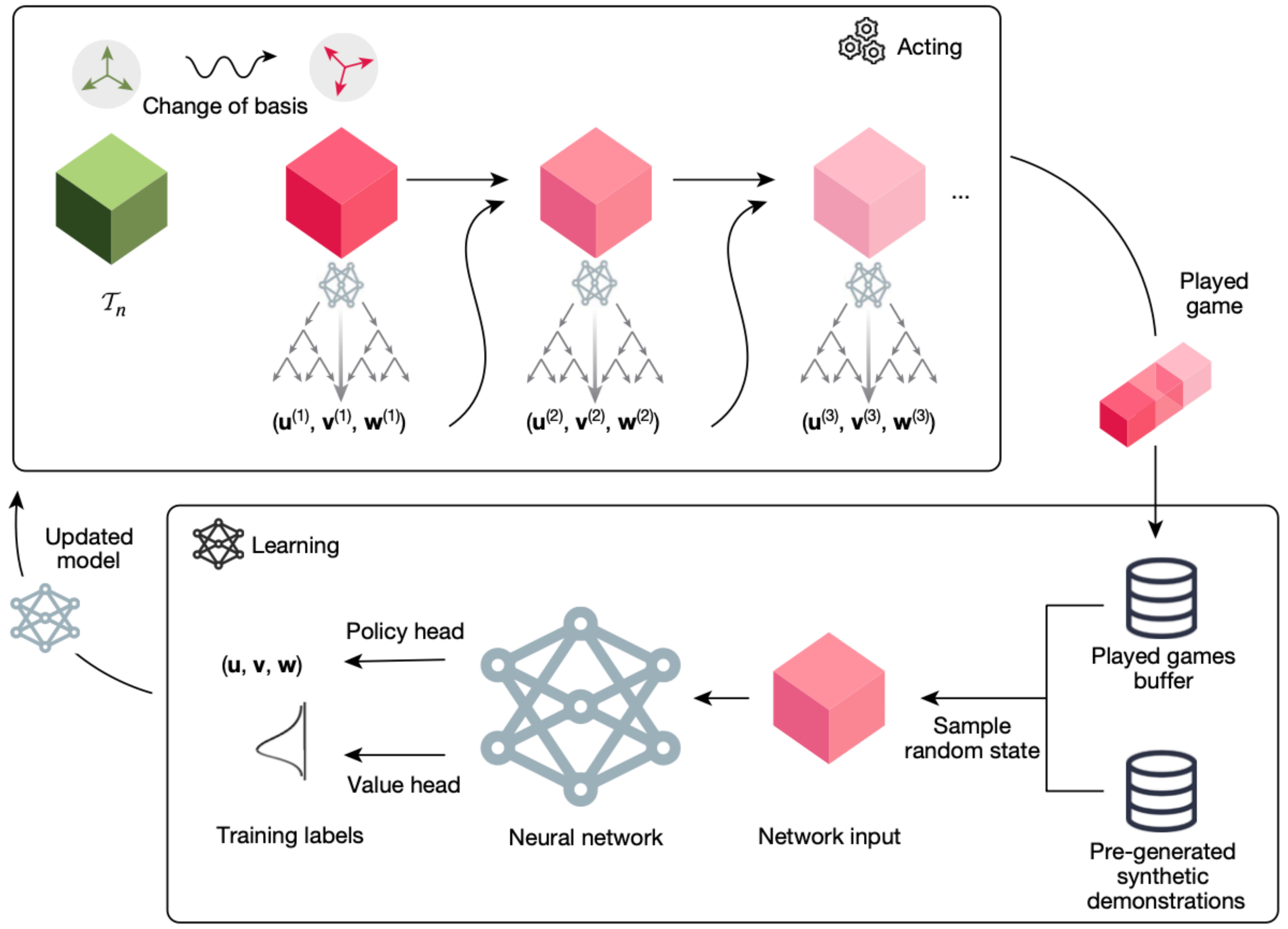
Discovering faster matrix multiplication algorithms with reinforcement learning

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis & Pushmeet Kohli

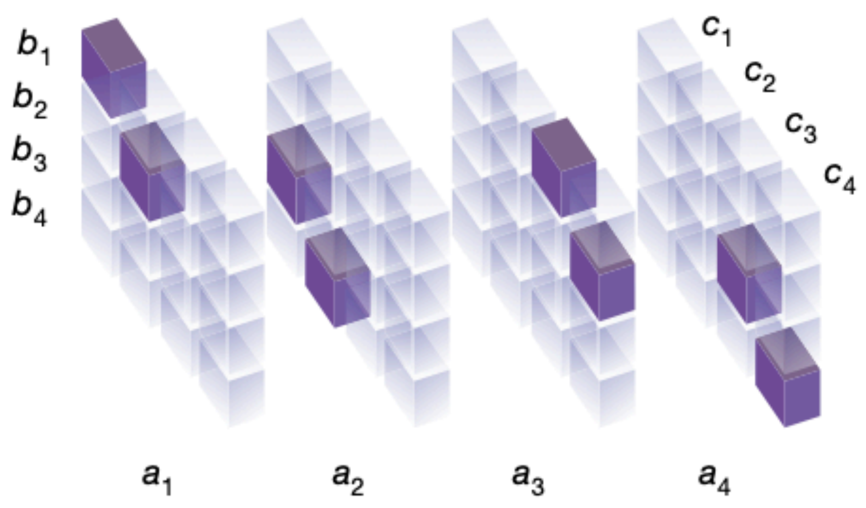
Nature 610, 47–53 (2022) | Cite this article

Abstract

Improving the efficiency of algorithms for fundamental computations can have a widespread impact, as it can affect the overall speed of a large amount of computations. Matrix multiplication is one such primitive task, occurring in many systems—from neural networks to scientific computing routines. The automatic discovery of algorithms using machine learning offers the prospect of reaching beyond human intuition and outperforming the current best human-designed algorithms. However, automating the algorithm discovery procedure is intricate, as the space of possible algorithms is enormous. Here we report a deep reinforcement learning approach based on AlphaZero[1] for discovering efficient and provably correct algorithms for the multiplication of arbitrary matrices. Our agent, AlphaTensor, is trained to play a single-player game where the objective is finding tensor decompositions within a finite factor space. AlphaTensor discovered algorithms that outperform the state-of-the-art complexity for many matrix sizes. Particularly relevant is the case of 4 × 4 matrices in a finite field, where AlphaTensor's algorithm improves on Strassen's two-level algorithm for the



$$\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix}$$



$$m_1 = (a_1 + a_4)(b_1 + b_4)$$
$$m_2 = (a_3 + a_4)\, b_1$$
$$m_3 = a_1\, (b_2 - b_4)$$
$$m_4 = a_4\, (b_3 - b_1)$$
$$m_5 = (a_1 + a_2)\, b_4$$
$$m_6 = (a_3 - a_1)(b_1 + b_2)$$
$$m_7 = (a_2 - a_4)(b_3 + b_4)$$
$$c_1 = m_1 + m_4 - m_5 + m_7$$
$$c_2 = m_3 + m_5$$
$$c_3 = m_2 + m_4$$
$$c_4 = m_1 - m_2 + m_3 + m_6$$

Can we leverage ML algorithms to find difficult examples in math?
What can we learn from the examples that machines find?

Motivation

Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

1 Introduction

Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

One idea: Find counterexamples

Motivation

Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

1 Introduction

Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

One idea: Find counterexamples

Example Conjecture 1

For any graph G with n vertices, we have,

$$\lambda_1(G) + \mu(G) \geq \sqrt{n-1} + 1$$

Motivation

Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

1 Introduction

Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

One idea: Find counterexamples

Example Conjecture 1

For any graph G with n vertices, we have,

$$\lambda_1(G) + \mu(G) \geq \sqrt{n-1} + 1$$

Previous known counterexample: 600 nodes

Motivation

Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

1 Introduction

Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

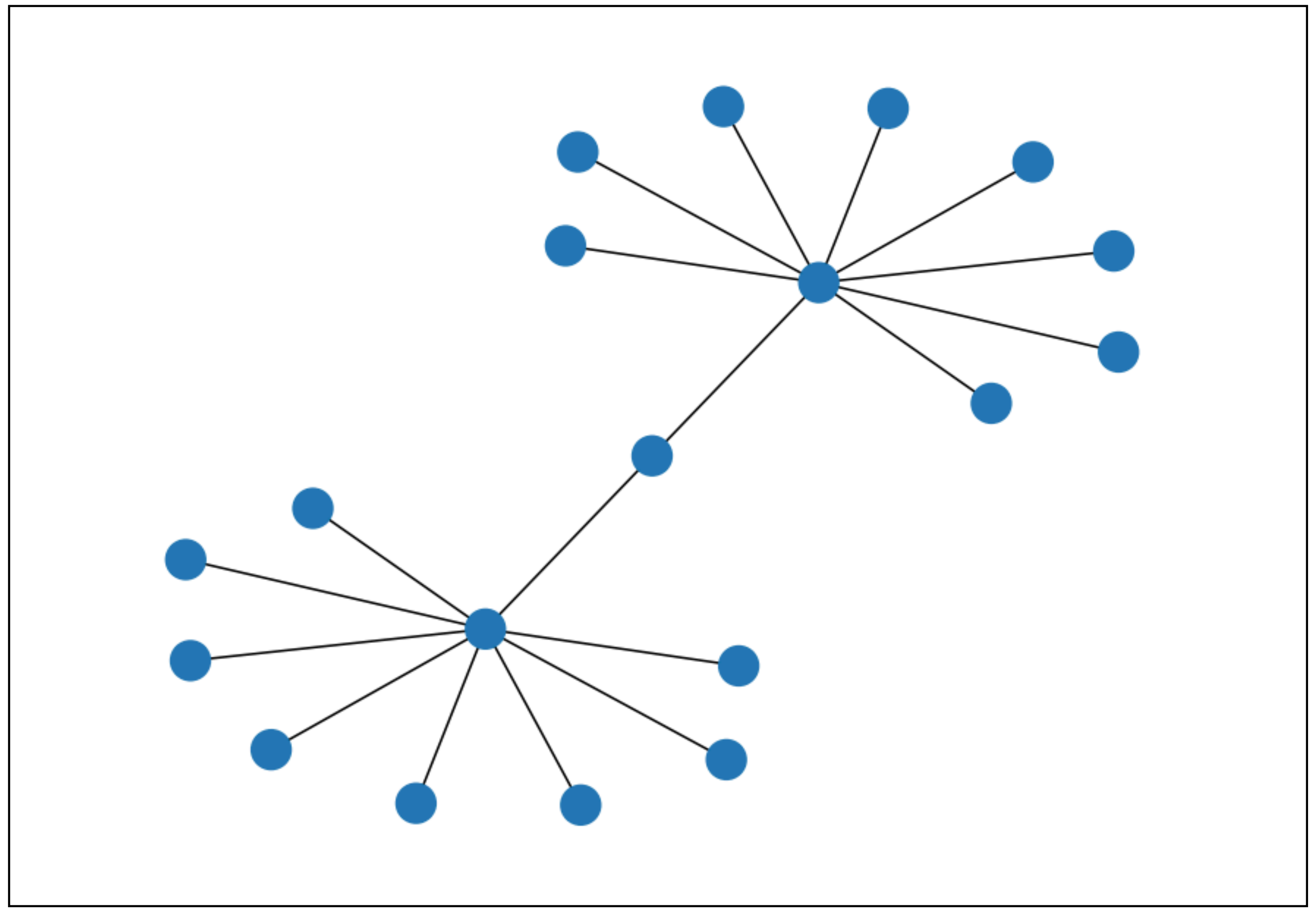
One idea: Find counterexamples

Example Conjecture 1

For any graph G with n vertices, we have,

$$\lambda_1(G) + \mu(G) \geq \sqrt{n-1} + 1$$

Previous known counterexample: 600 nodes



Algorithm Background

Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

1 Introduction

Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

One idea: Find counterexamples

Example Conjecture 2

For any graph G, $K_4(G) + K_4(\bar{G})$ is asymptotically minimized by random graphs.

Algorithm Background

Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

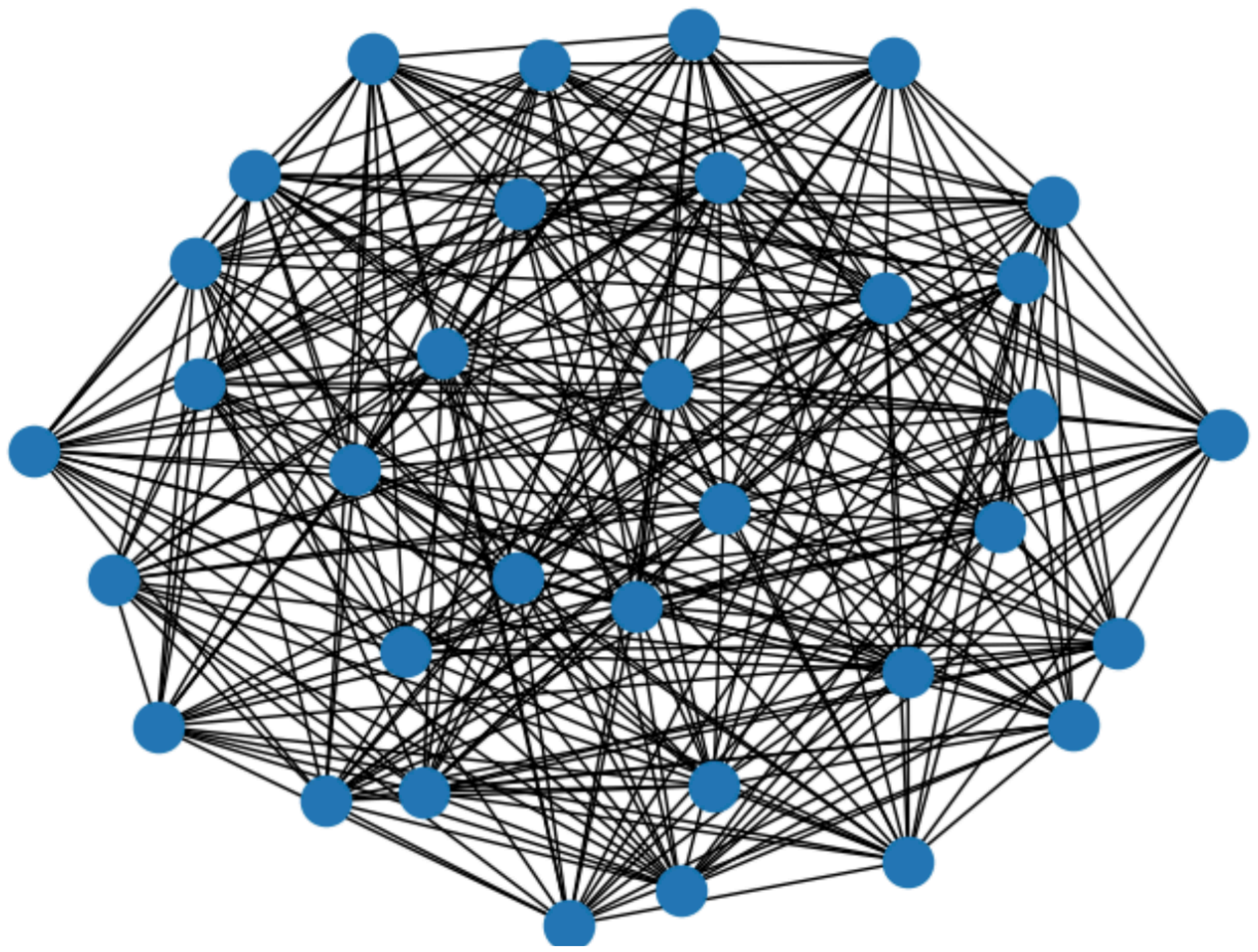
1 Introduction

Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

One idea: Find counterexamples

Example Conjecture 2

For any graph G, $K_4(G) + K_4(\bar{G})$ is asymptotically minimized by random graphs.



Algorithm Background

Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

1 Introduction

Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

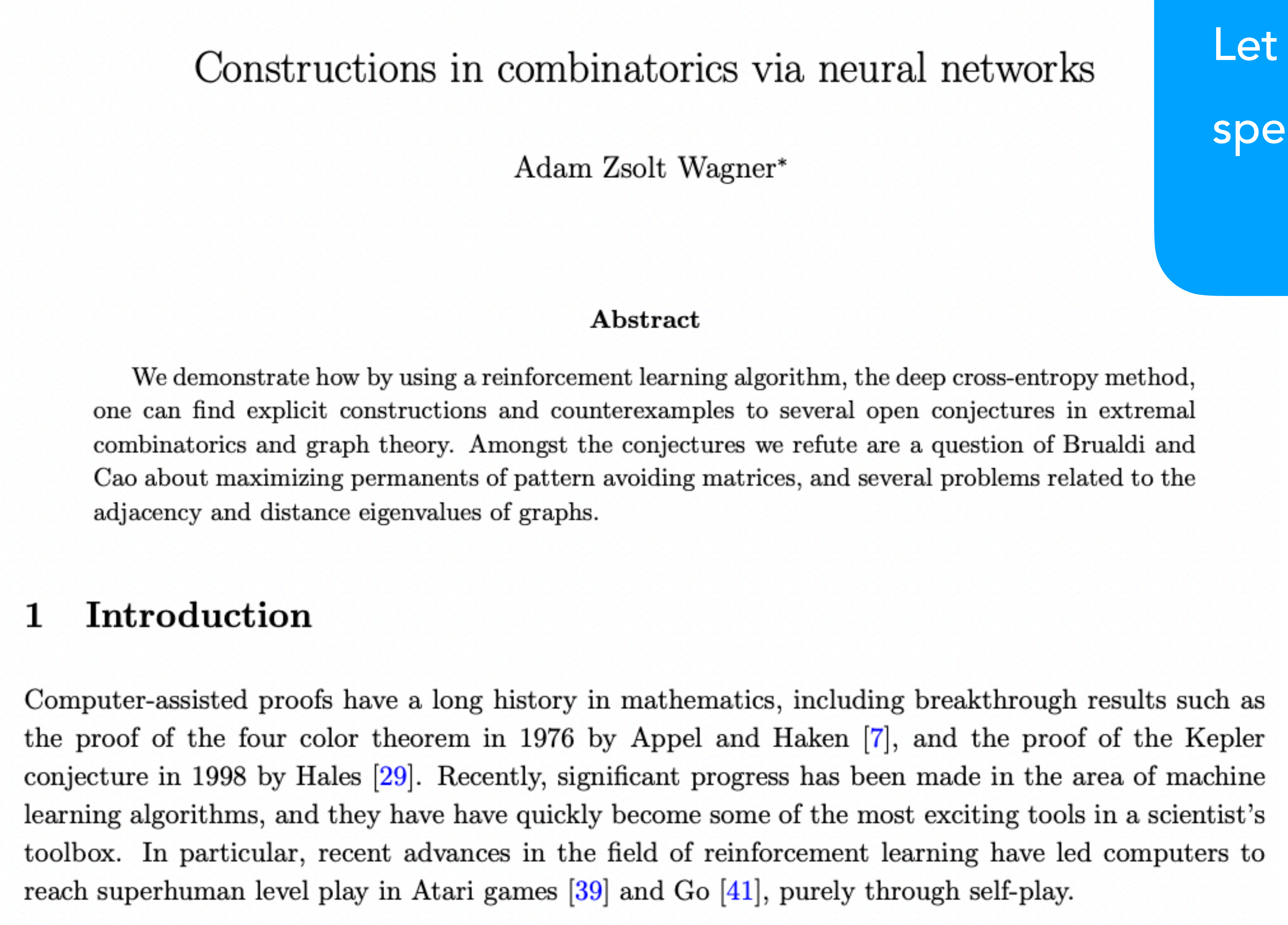
Example Conjecture 3

Let G be a graph with diameter D, proximity π, and distance spectrum $\partial_1 \geq \ldots \geq \partial_n$, then

$$\pi + \partial_{\lfloor \frac{2D}{3} \rfloor} > 0$$

One idea: Find counterexamples

Algorithm Background



Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

1 Introduction

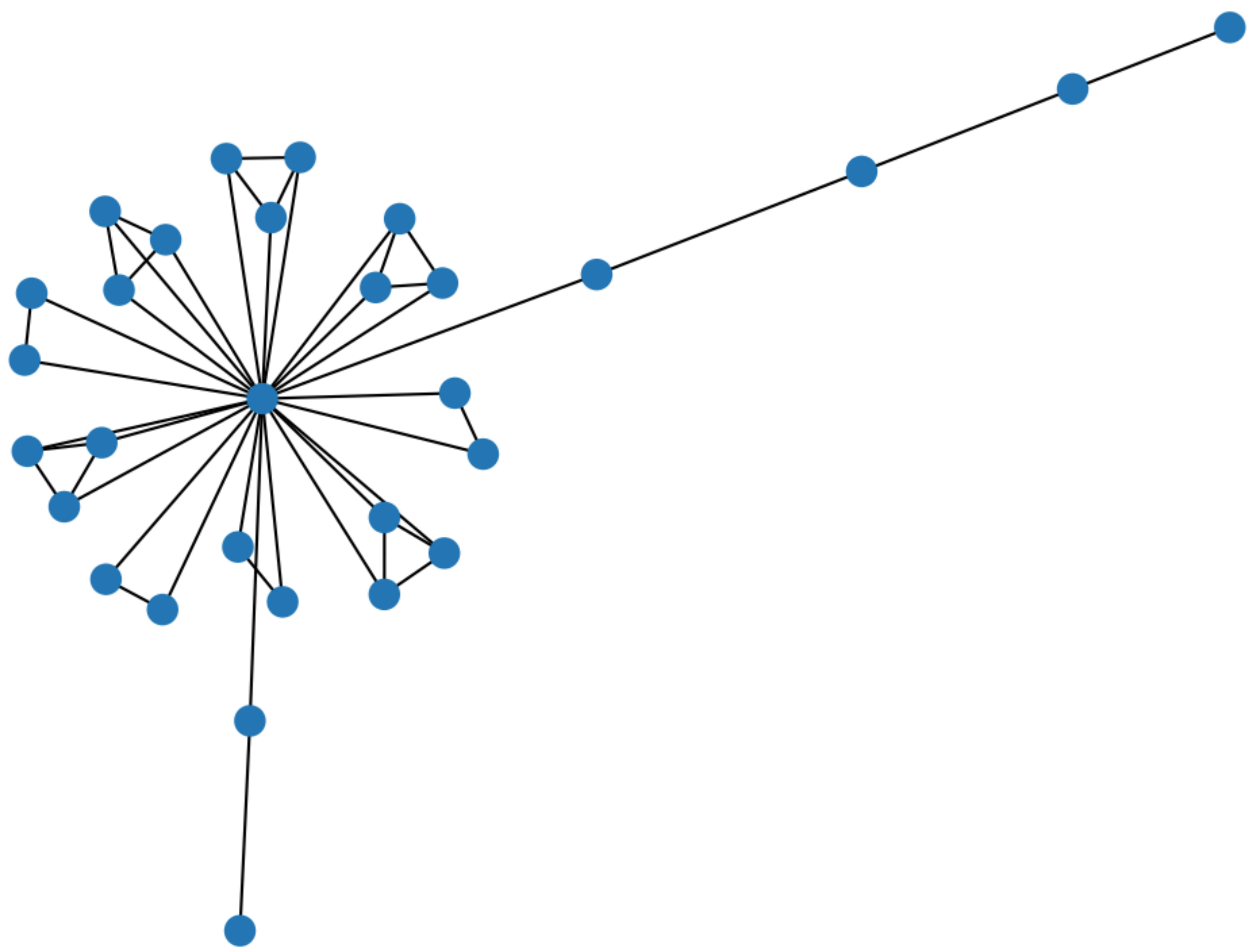
Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

One idea: Find counterexamples

Example Conjecture 3

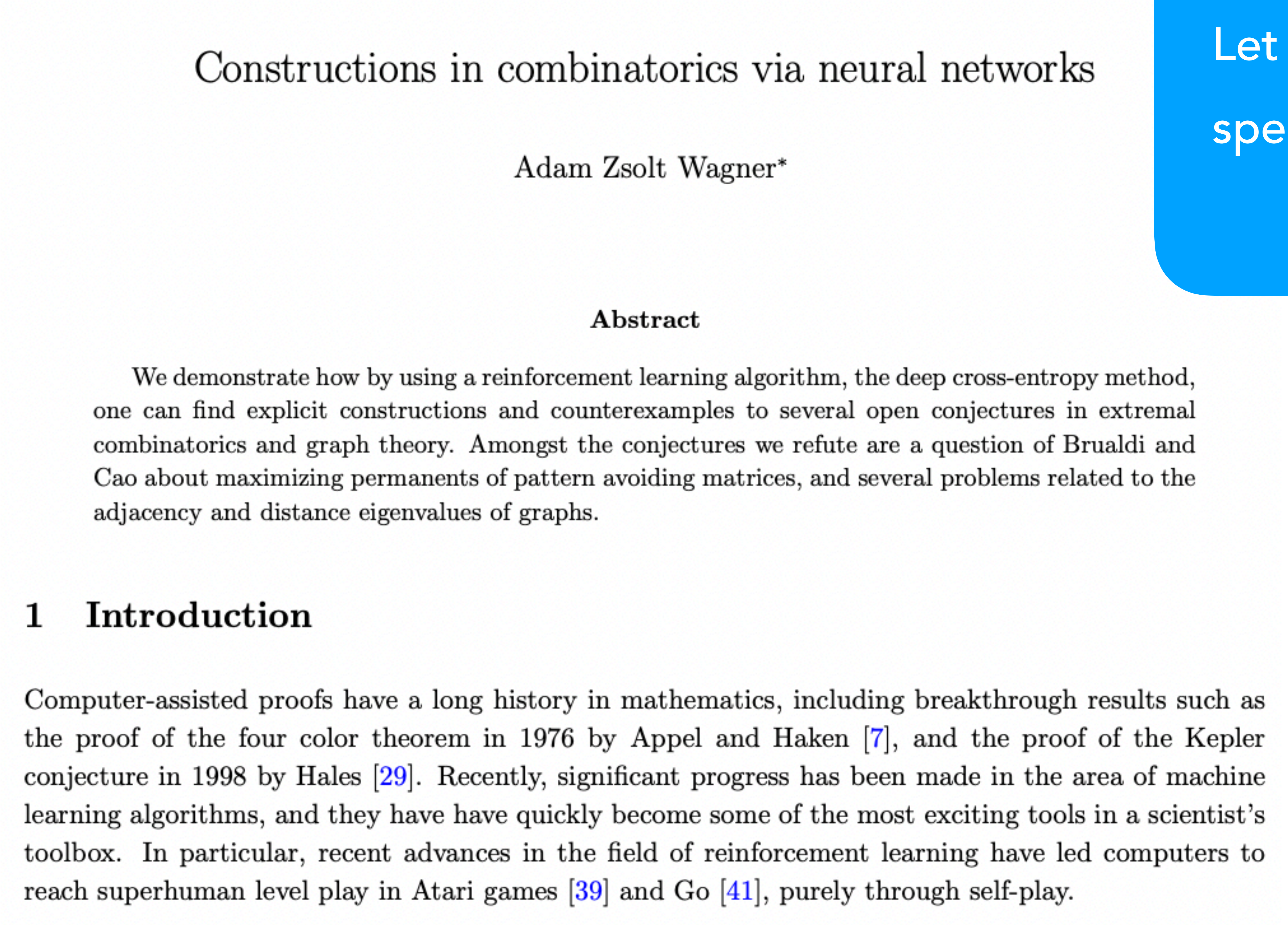
Let G be a graph with diameter D, proximity π, and distance spectrum $\partial_1 \geq \ldots \geq \partial_n$, then

$$\pi + \partial_{\lfloor \frac{2D}{3} \rfloor} > 0$$



Not a counterexample…..

Algorithm Overview



Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

1 Introduction

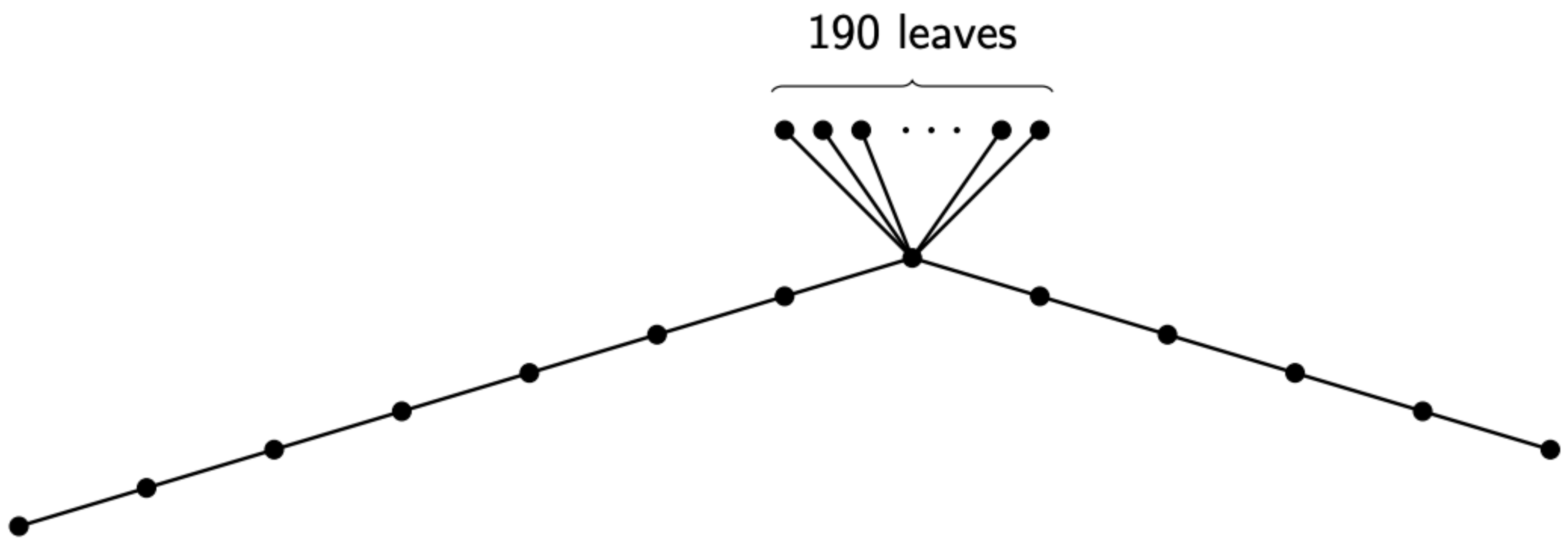
Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

One idea: Find counterexamples

Example Conjecture 3

Let G be a graph with diameter D, proximity π, and distance spectrum $\partial_1 \geq \ldots \geq \partial_n$, then

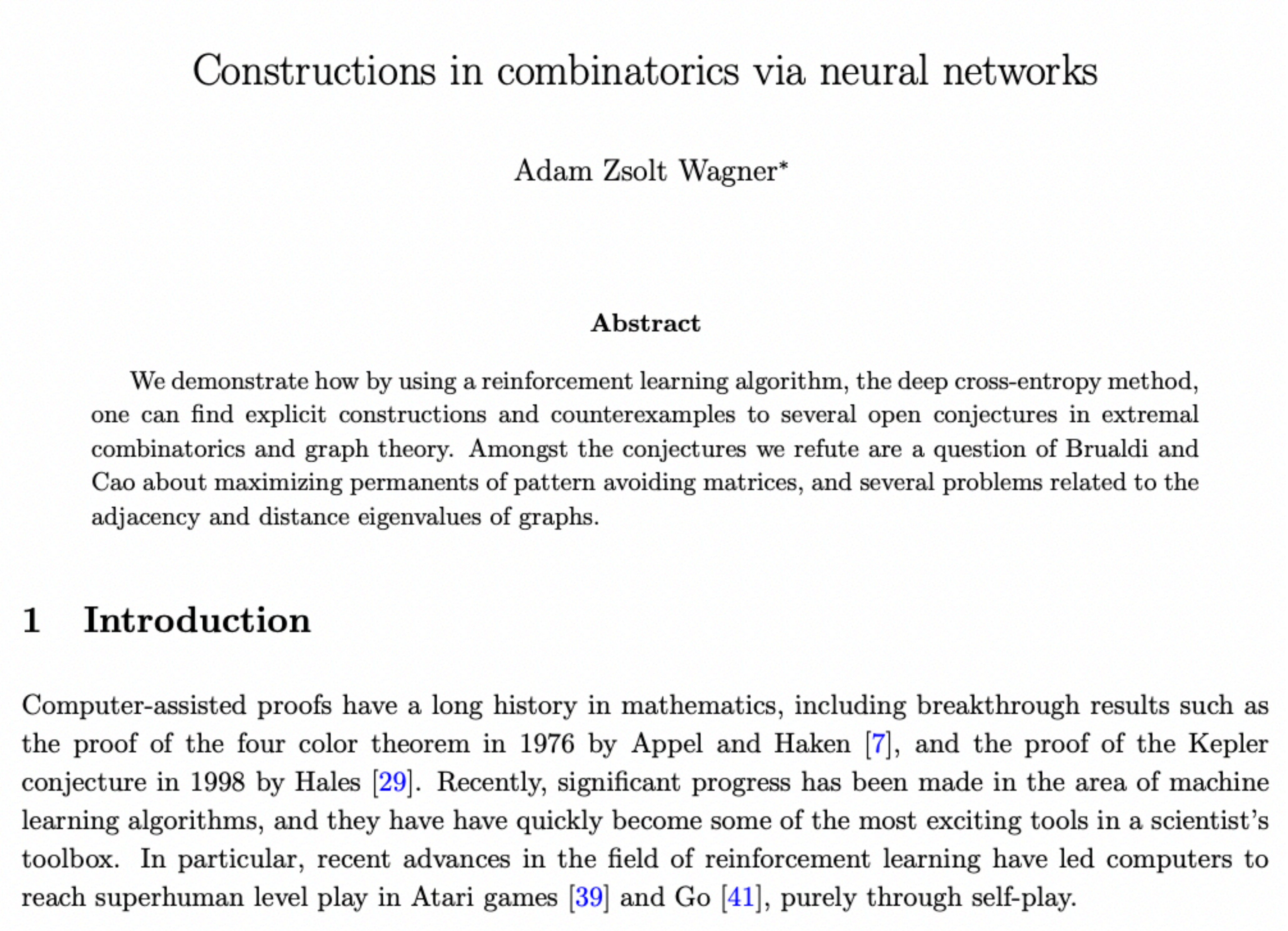
$$\pi + \partial_{\lfloor \frac{2D}{3} \rfloor} > 0$$



Not a counterexample….. but it leads to one

Algorithm Background

Adapted from [Wagner, 2021]:



Constructions in combinatorics via neural networks

Adam Zsolt Wagner*

Abstract

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

1 Introduction

Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning algorithms, and they have have quickly become some of the most exciting tools in a scientist's toolbox. In particular, recent advances in the field of reinforcement learning have led computers to reach superhuman level play in Atari games [39] and Go [41], purely through self-play.

One idea: Find counterexamples

Immediate Counterexample

Not a Counterexample and / or not insightful

Almost a Counterexample
But was able to extend to counterexample

Motivation

- In extremal combinatorics, we’re interested in finding examples of large sets with conditions holding

Motivation

- In extremal combinatorics, we're interested in finding examples of large sets with conditions holding
- Can we teach a neural network to generate these large sets with no prior knowledge? (Of course, to do it well, we need heuristics)

Motivation

- In extremal combinatorics, we're interested in finding examples of large sets with conditions holding
- Can we teach a neural network to generate these large sets with no prior knowledge? (Of course, to do it well, we need heuristics)
- Can we (as humans) learn heuristic rules from these constructions and use them to find better examples, gain insights into the properties of these sets, and prove theorems?

RL

Reinforcement Learning: Learning Decisions to Maximize Reward

- An agent plays a game many times

RL

Reinforcement Learning: Learning Decisions to Maximize Reward

- An agent plays a game many times
- It knows the current state, the current actions it can make, and the resulting state.

RL

Reinforcement Learning: Learning Decisions to Maximize Reward

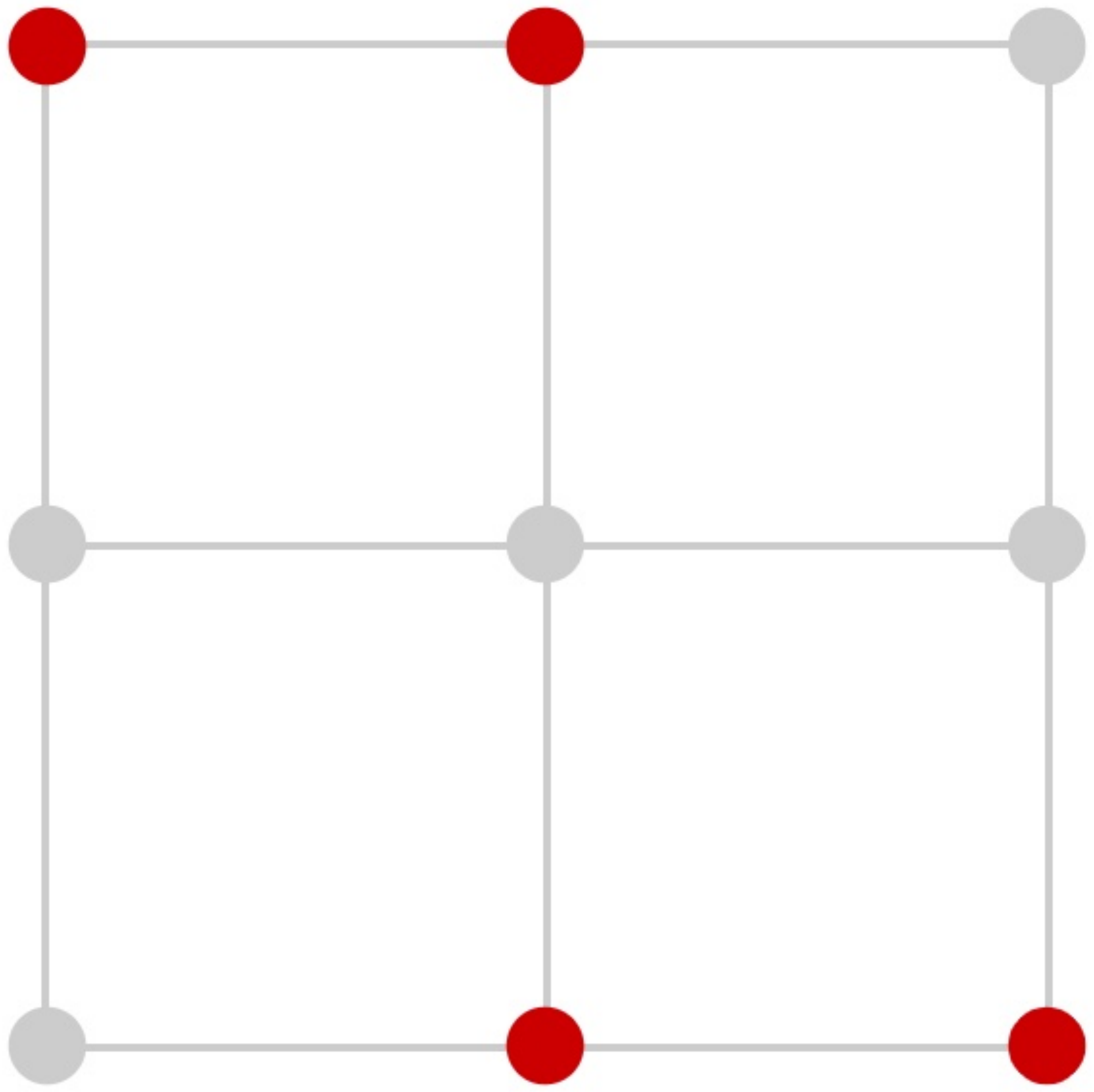
- An agent plays a game many times
- It knows the current state, the current actions it can make, and the resulting state.
- It does not know how good or bad each state is. It does know the reward it gains at the end of the game

RL

Reinforcement Learning: Learning Decisions to Maximize Reward

- An agent plays a game many times
- It knows the current state, the current actions it can make, and the resulting state.
- It does not know how good or bad each state is. It does know the reward it gains at the end of the game
- Through many games, tries to maximize rewards

Problem 1



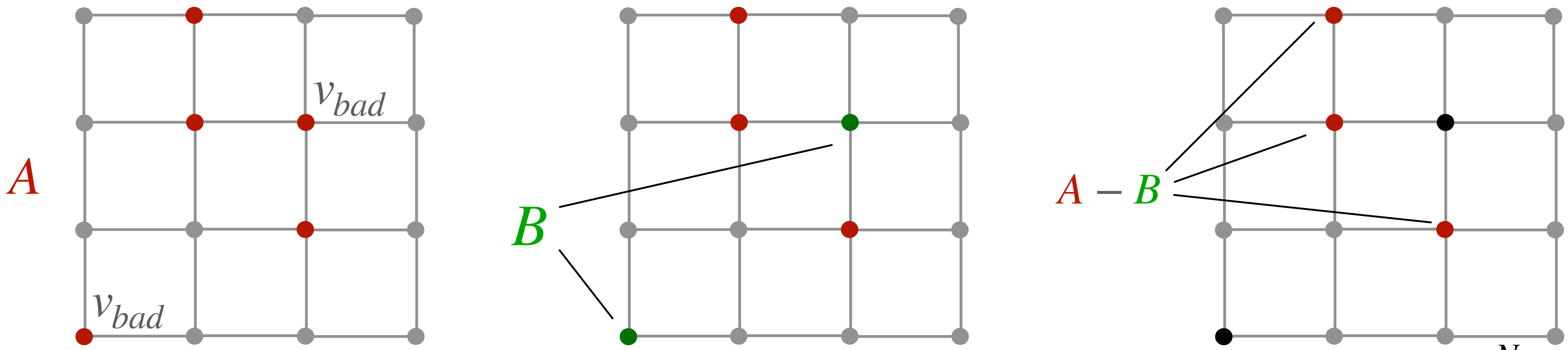
Given an n x n finite integer lattice, what's the size of the largest subset such that no three points form an isosceles triangle?

Problem 1

What do we know?

Lower Bound

$$\epsilon' \frac{N}{\sqrt{\log N}} \leq |\text{Largest Set}|$$



$P(v \text{ is bad}) \leq Cp^3N^2 \log N$

$\mathbb{E}(|B|) \leq Cp^3N^4 \log N$

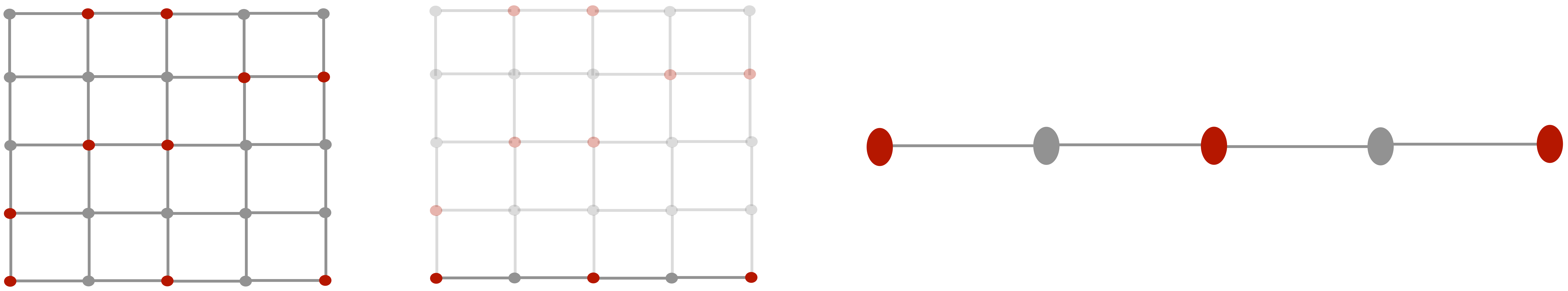
$\mathbb{E}(|A - B|) \geq \epsilon' \frac{N}{\sqrt{\log N}}$

Problem 1

What do we know?

Upper Bound

$$|\text{Largest Set}| \leq \exp(-c(\log N)^{\frac{1}{9}})N^2$$



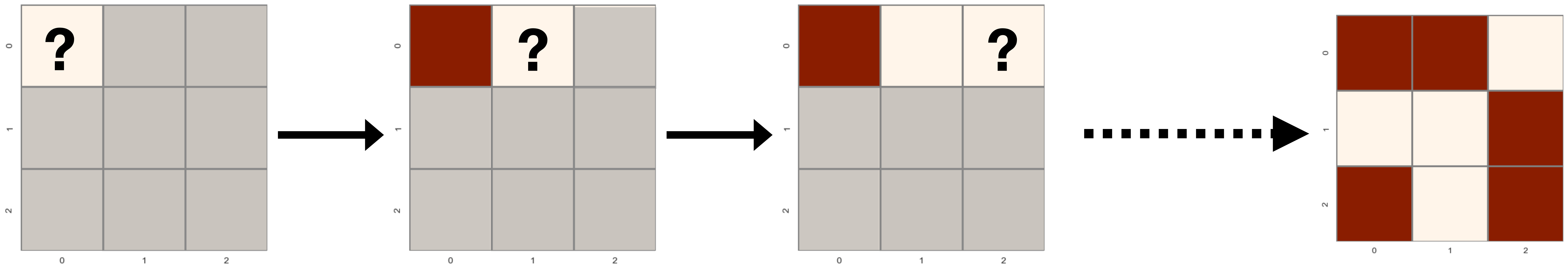
Problem 1

Final Bounds

$$\epsilon' \frac{N}{\sqrt{\log N}} \leq |\text{Largest Set}| \leq \exp(-c(\log N)^{\frac{1}{9}})N^2$$

Problem 1

Game Setup: Binary action



Model: Neural Network

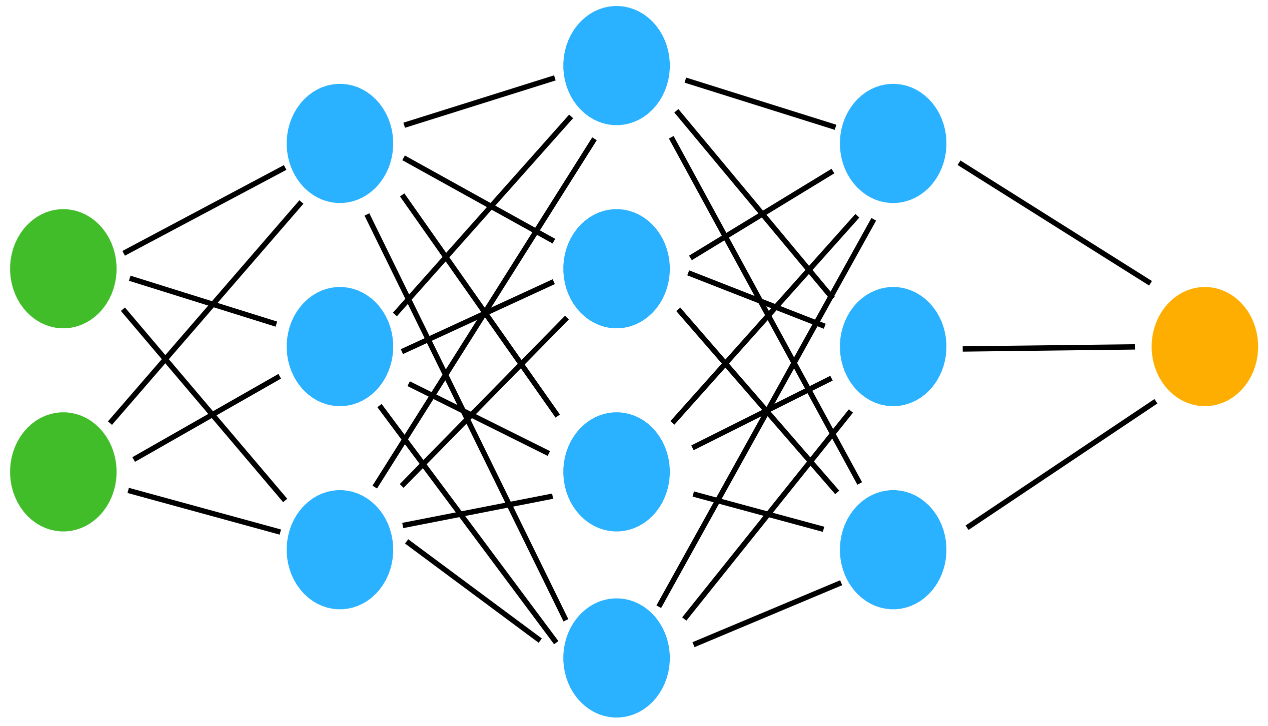
<u>Feed Forward Step</u>

3 Hidden Layers

(128, 64, 4)

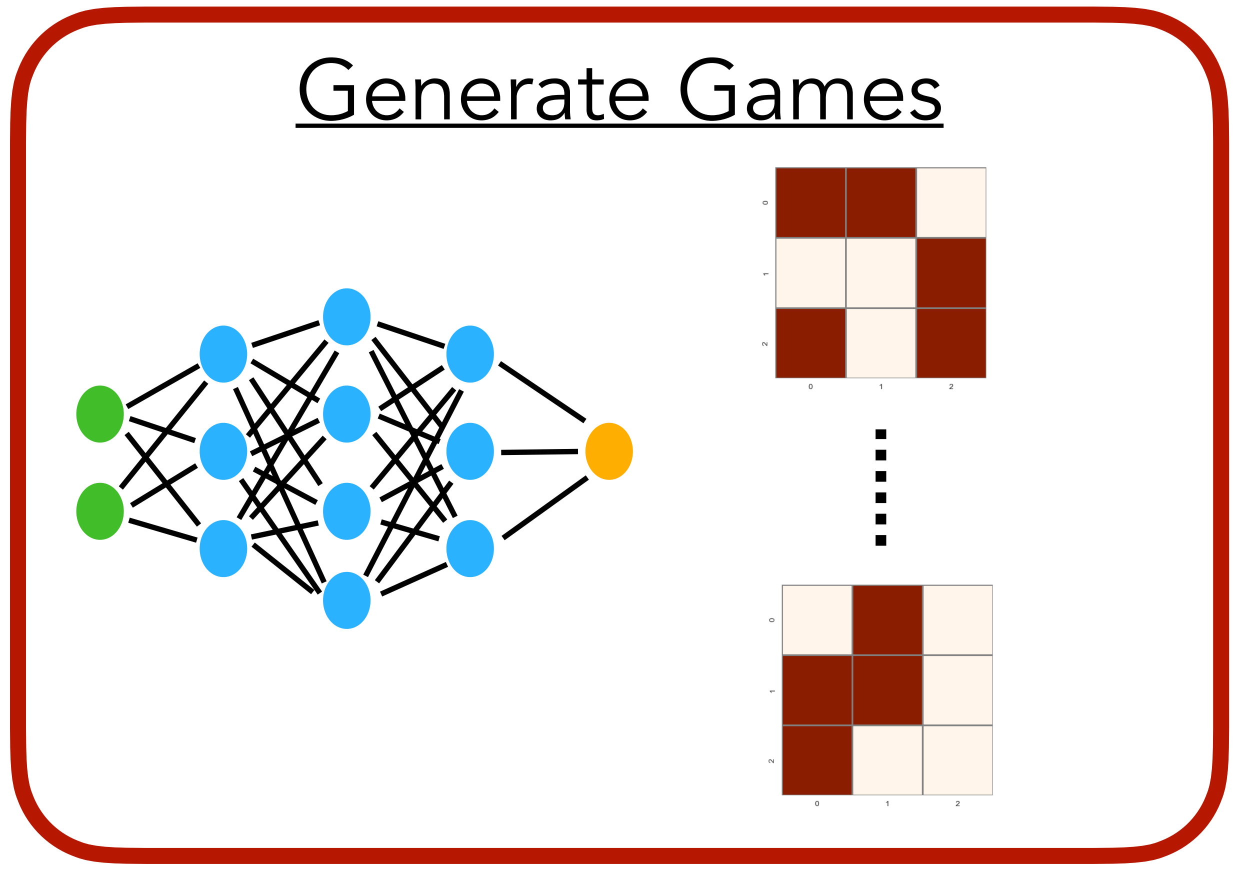
Relu Hidden Activation

Sigmoid Output Activation



Problem 1

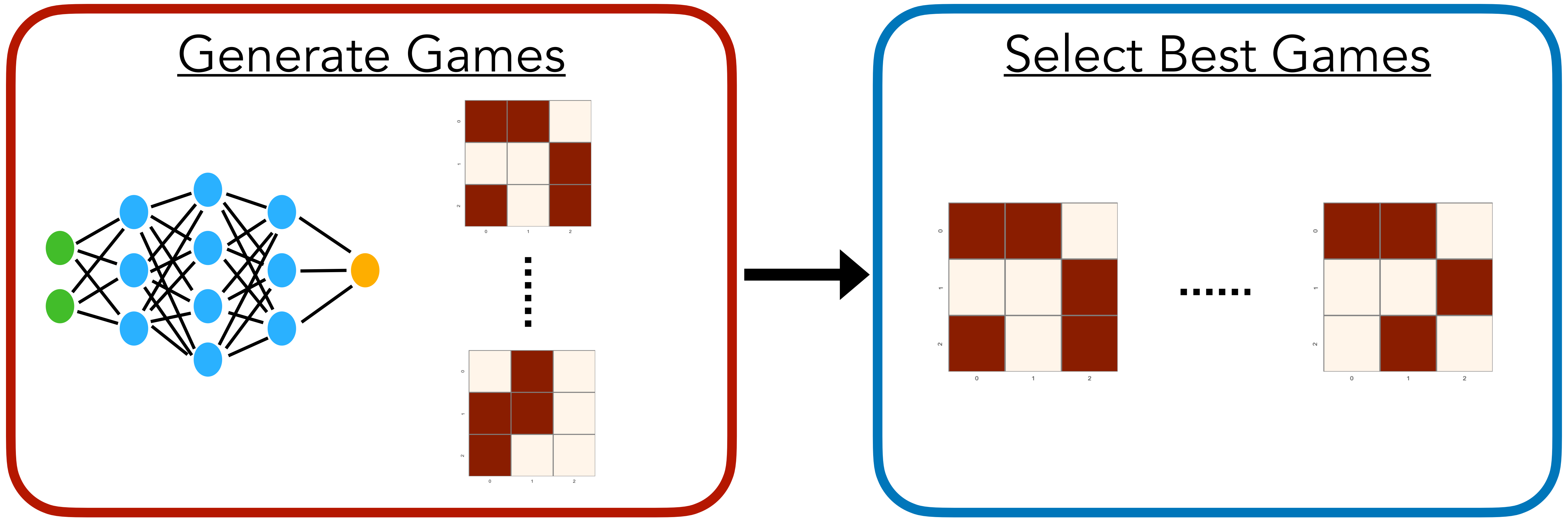
Training Paradigm



Reward the agent for each game: $s(\cdot) = -$ (# of isosceles Δ's) $+ \lambda \cdot$ (# of points)

Problem 1

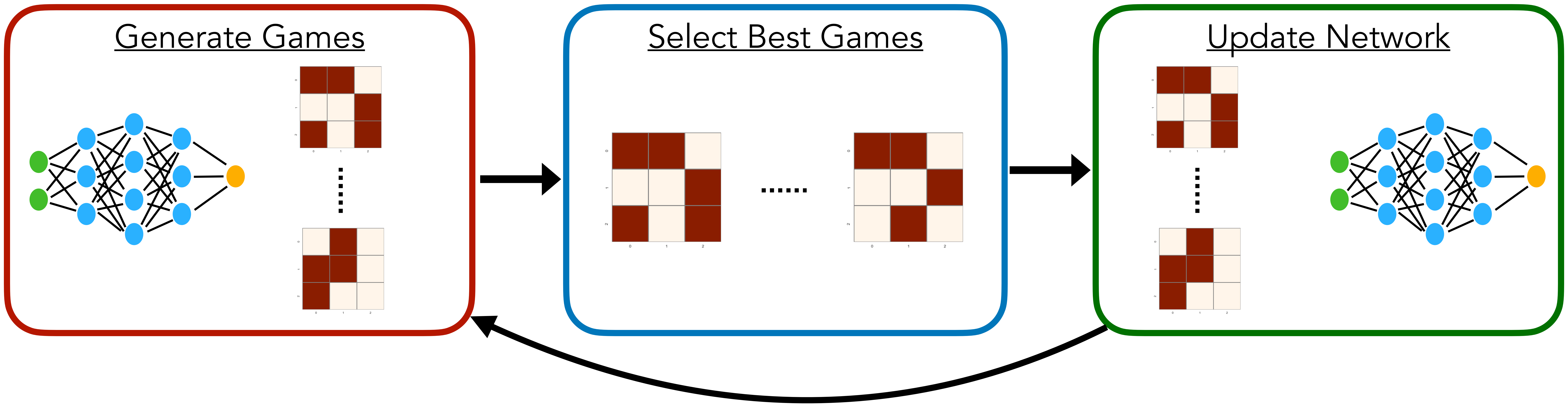
Training Paradigm



Reward the agent for each game: $s(\cdot) = -$ (# of isosceles Δ's) $+ \lambda \cdot$ (# of points)

Problem 1

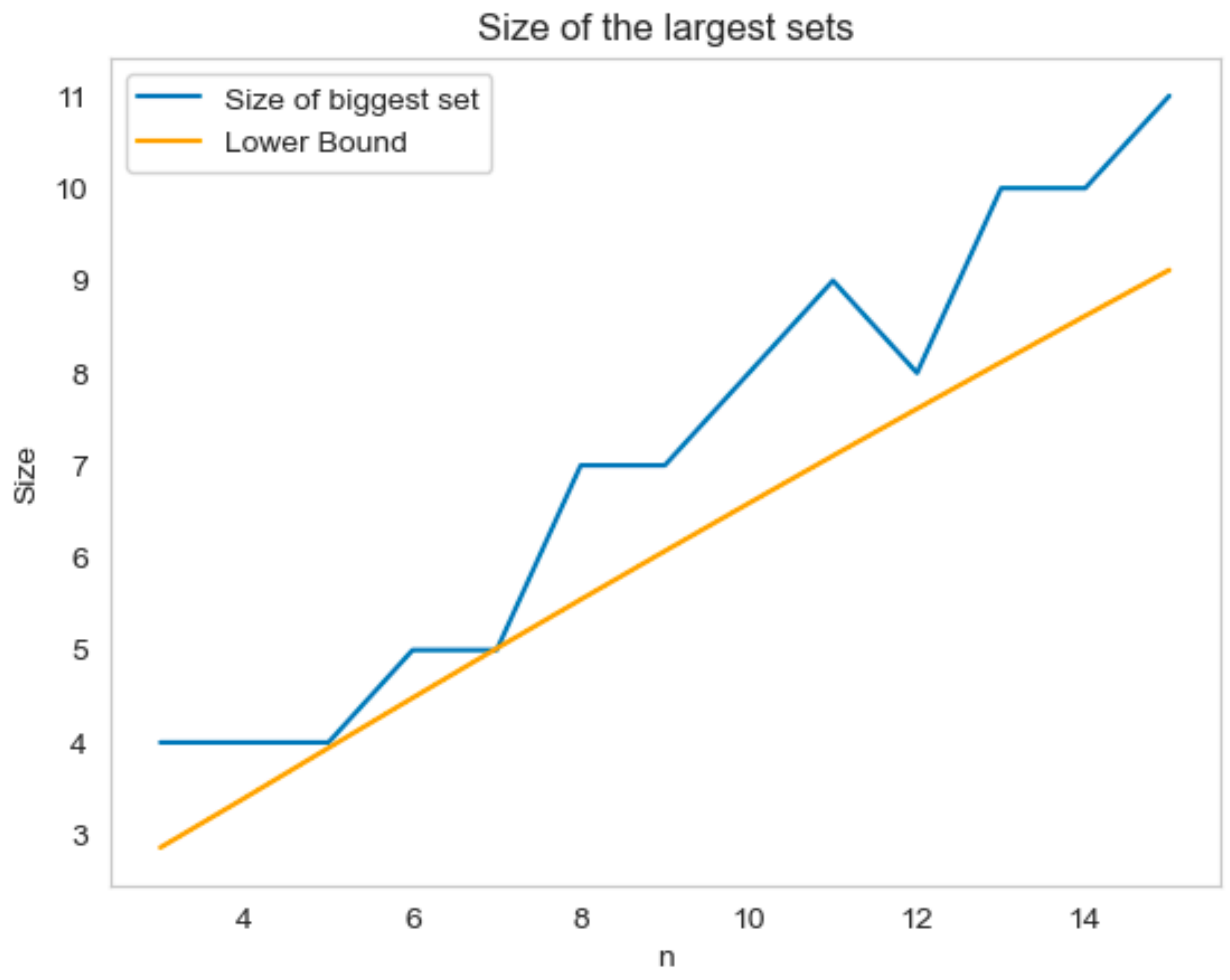
Training Paradigm



Reward the agent for each game: $s(\cdot) = -$ (# of isosceles Δ's) $+ \lambda \cdot$ (# of points)

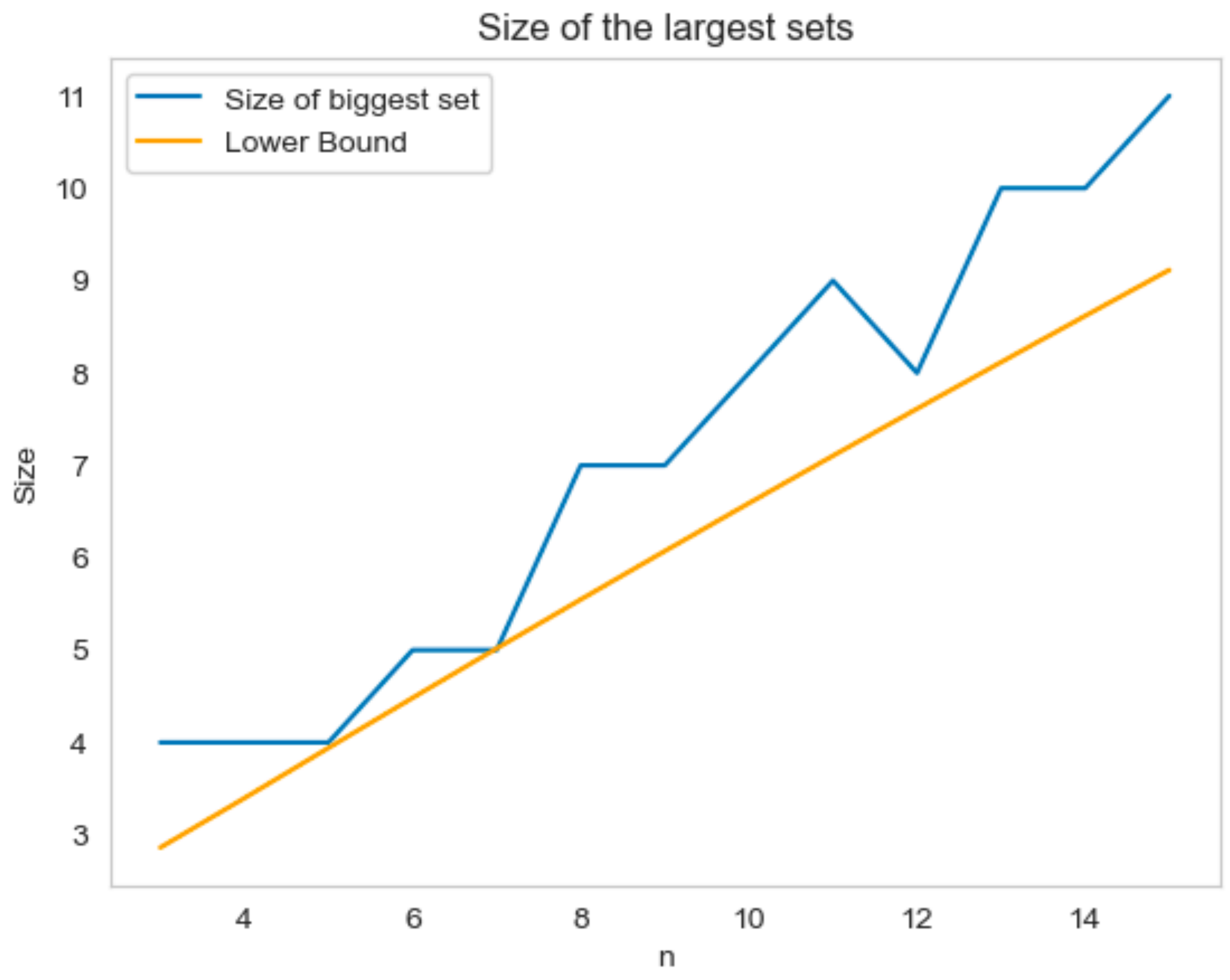
Results

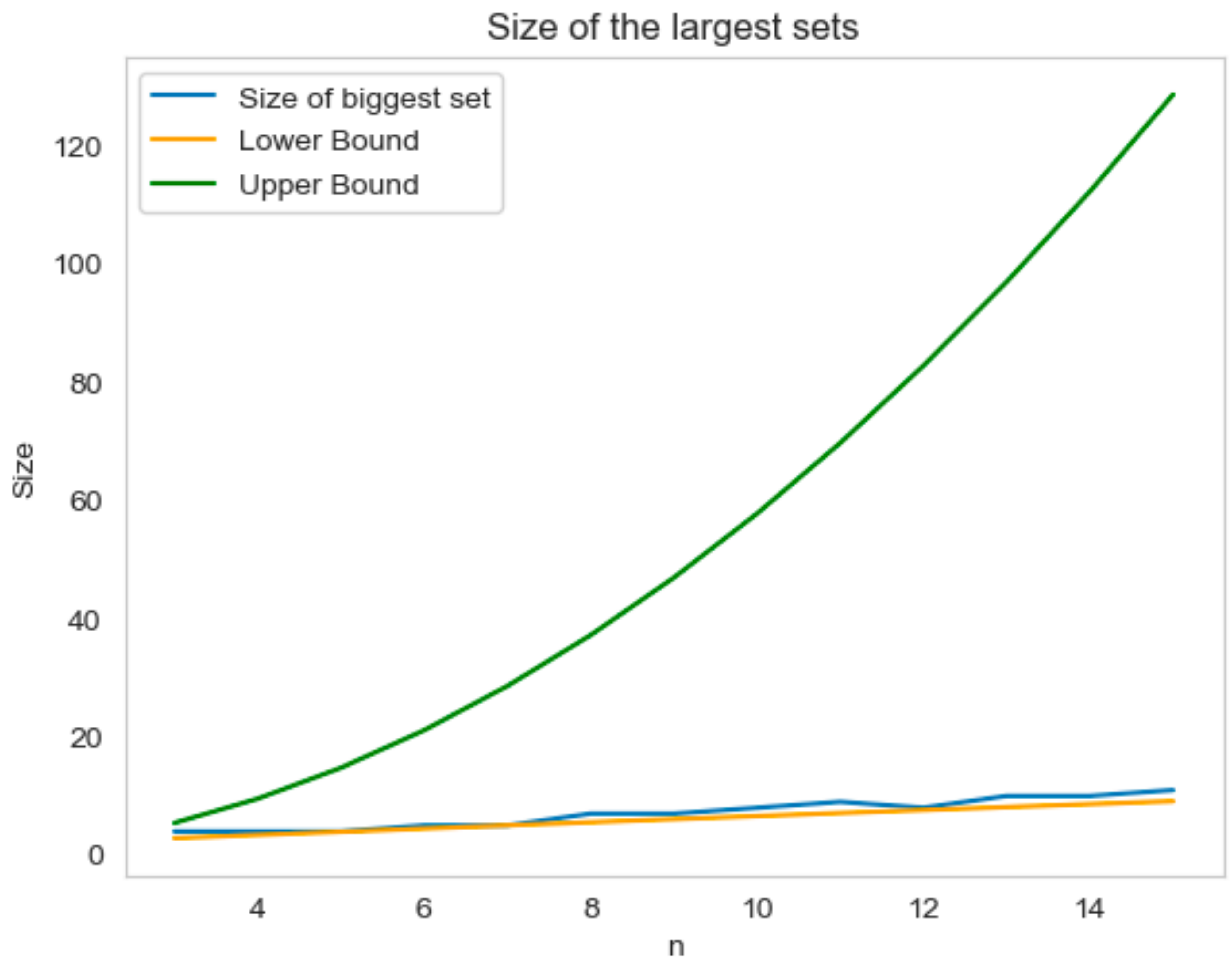
….Not that great



Results

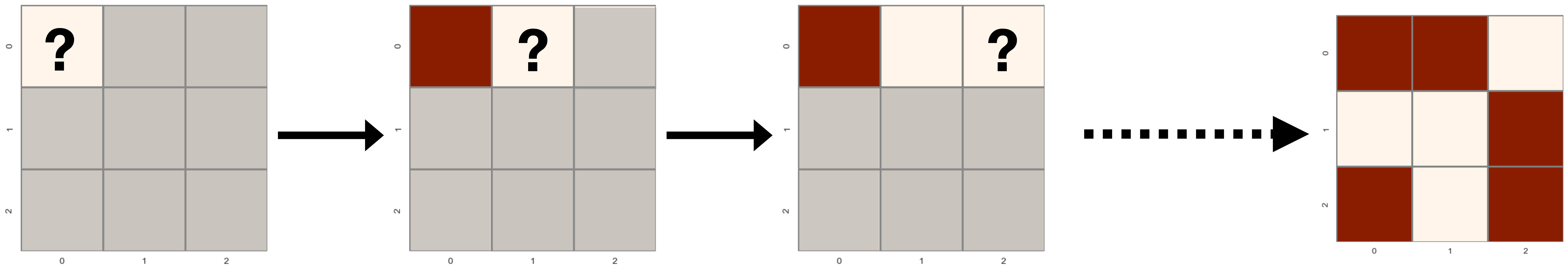
....Not that great





Problem 1

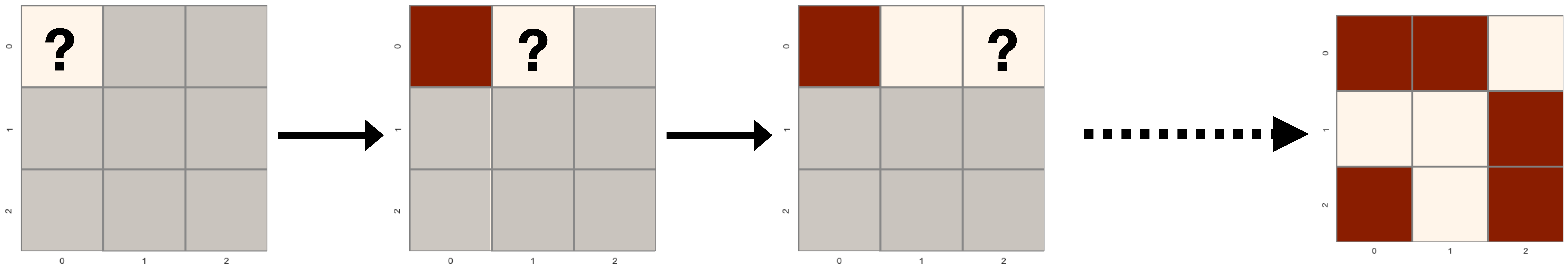
Game setup



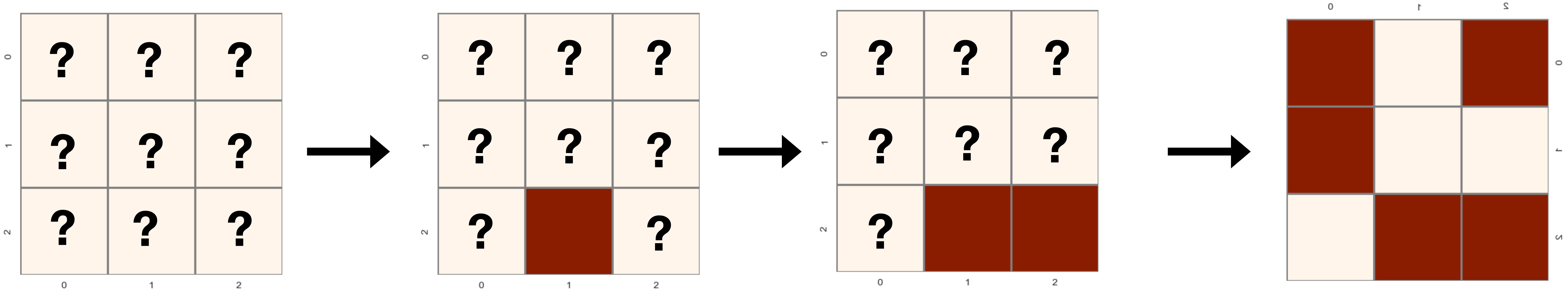
Model only ‘sees’ extremely local information. How can we set it up?

Problem 1

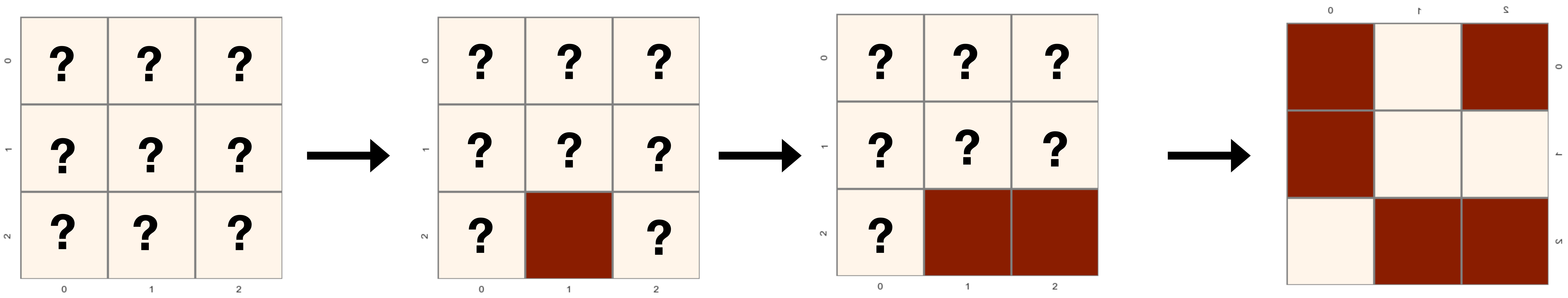
Game setup



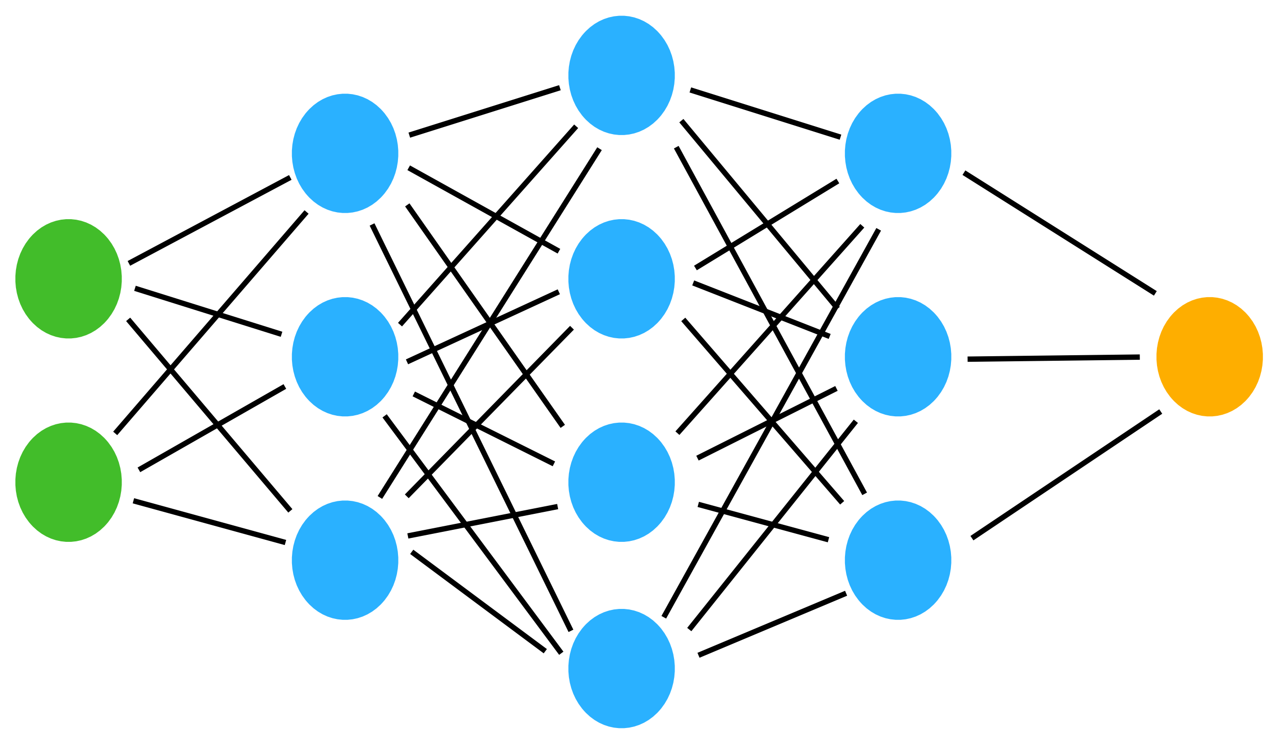
Model only ‘sees’ extremely local information. How can we set it up?



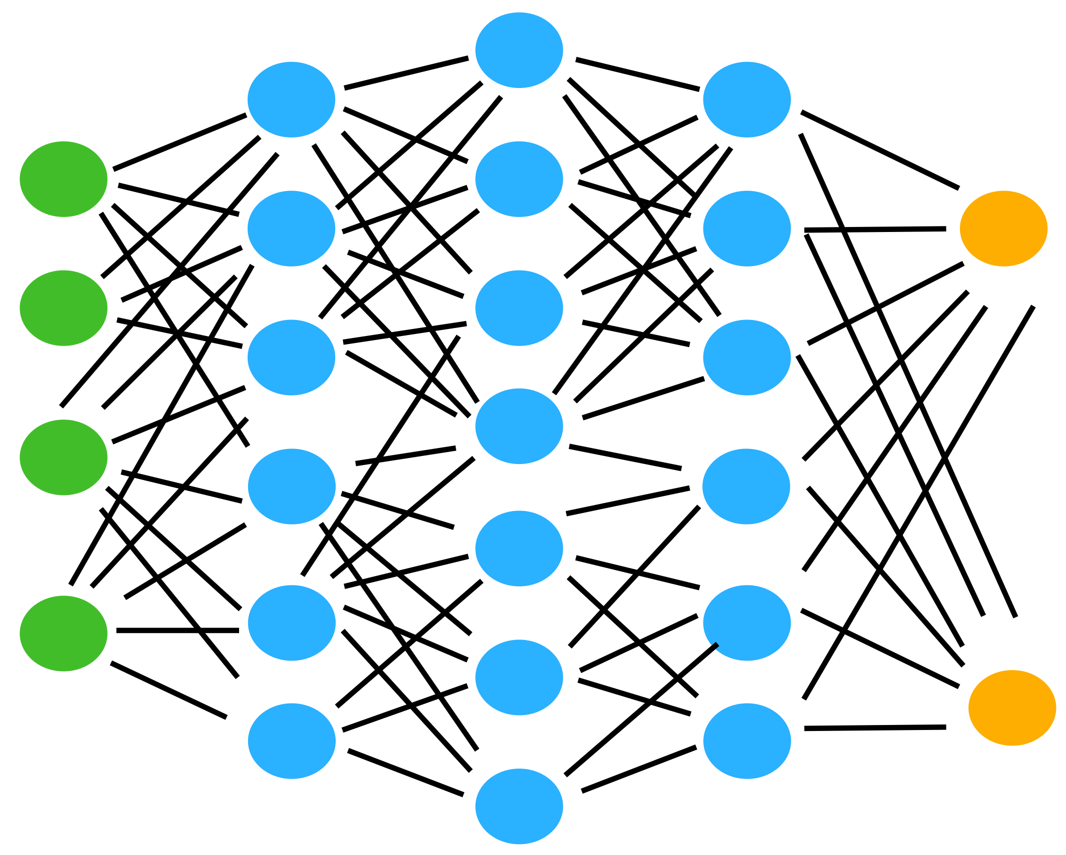
Problem 1



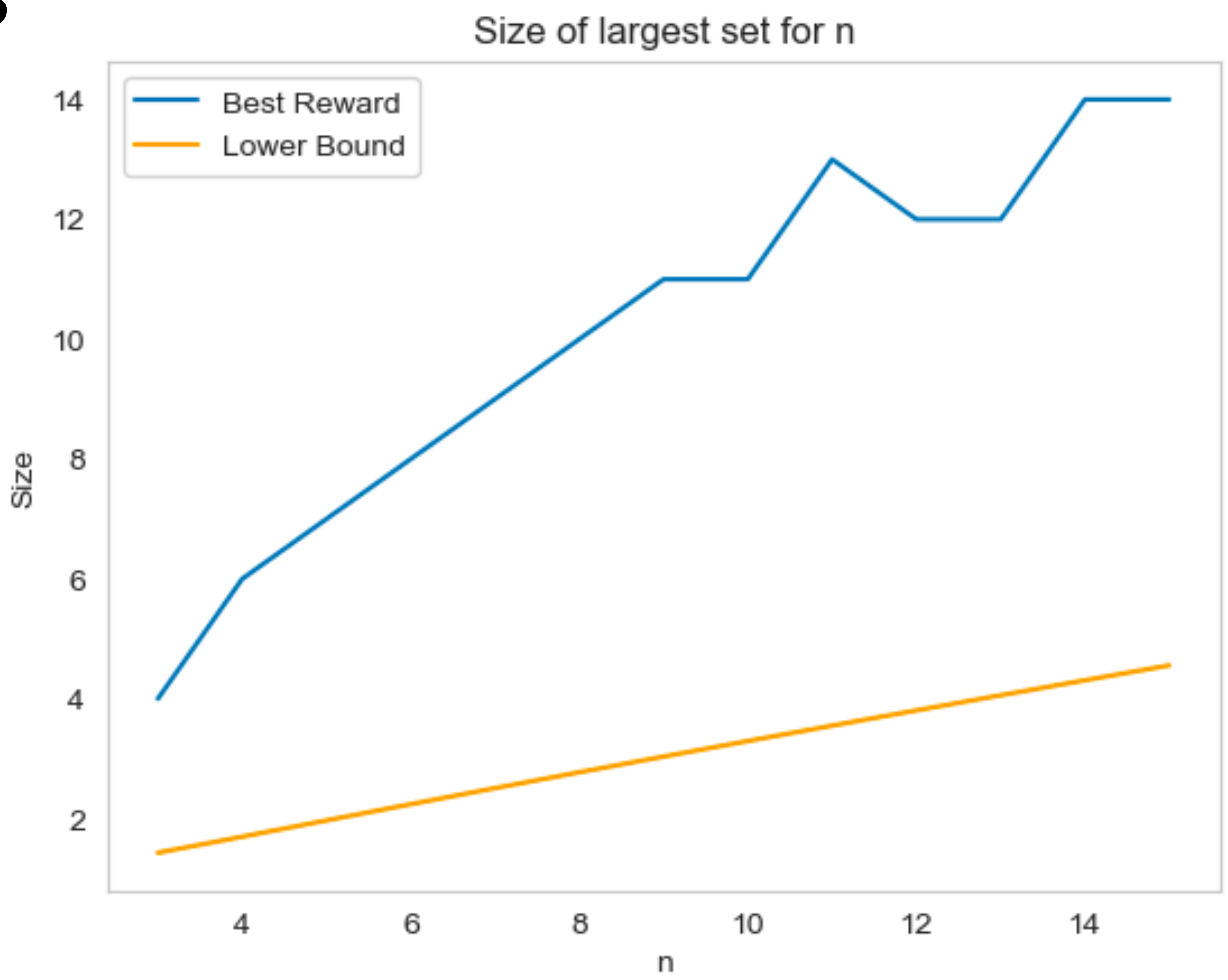
N = 3
3 Hidden Layers
(128, 64, 4)
Relu Hidden Layers
Sigmoid Output



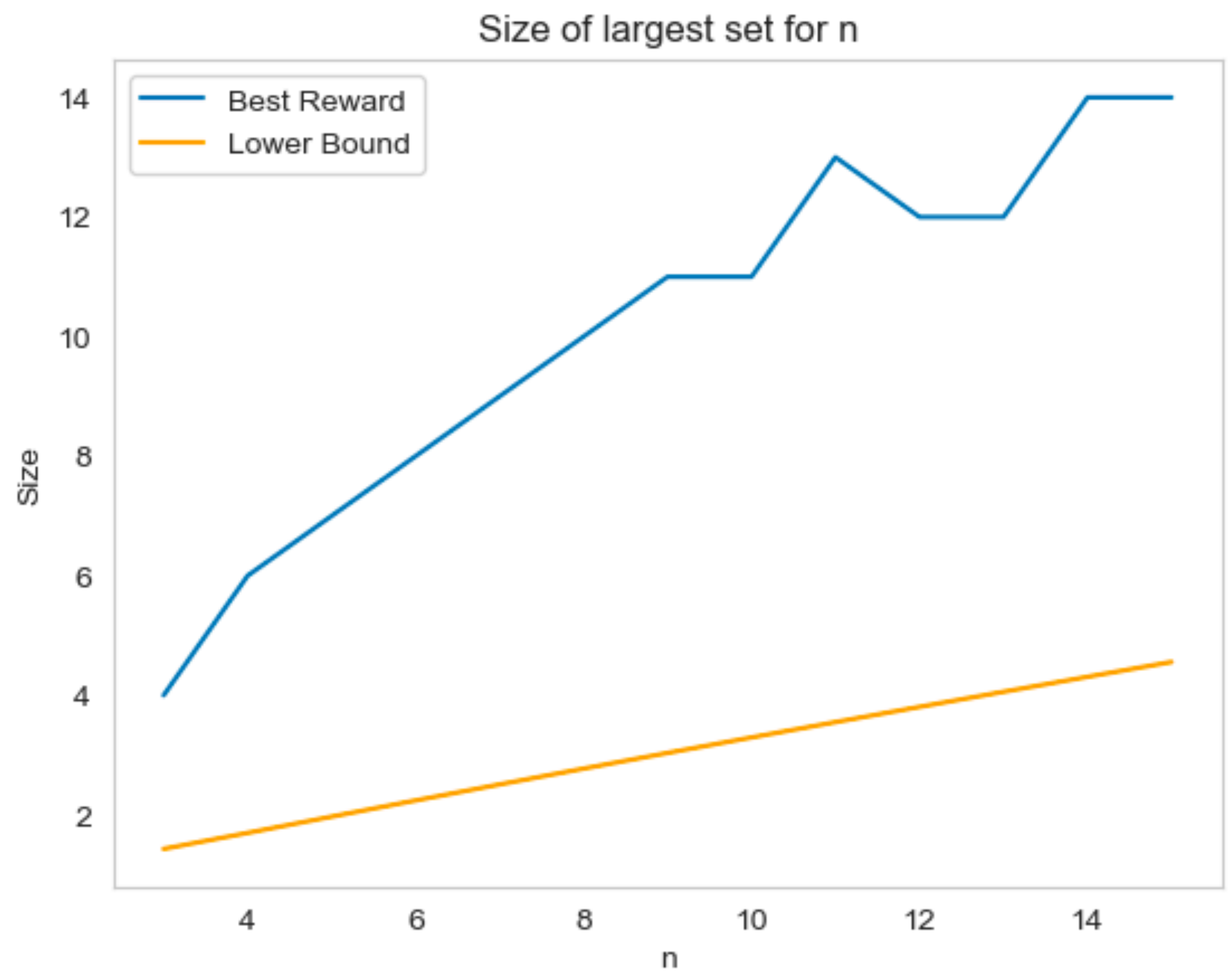
N = 6
3 Hidden Layers
(256, 128, 8)
Relu Hidden Layers
Softmax Output



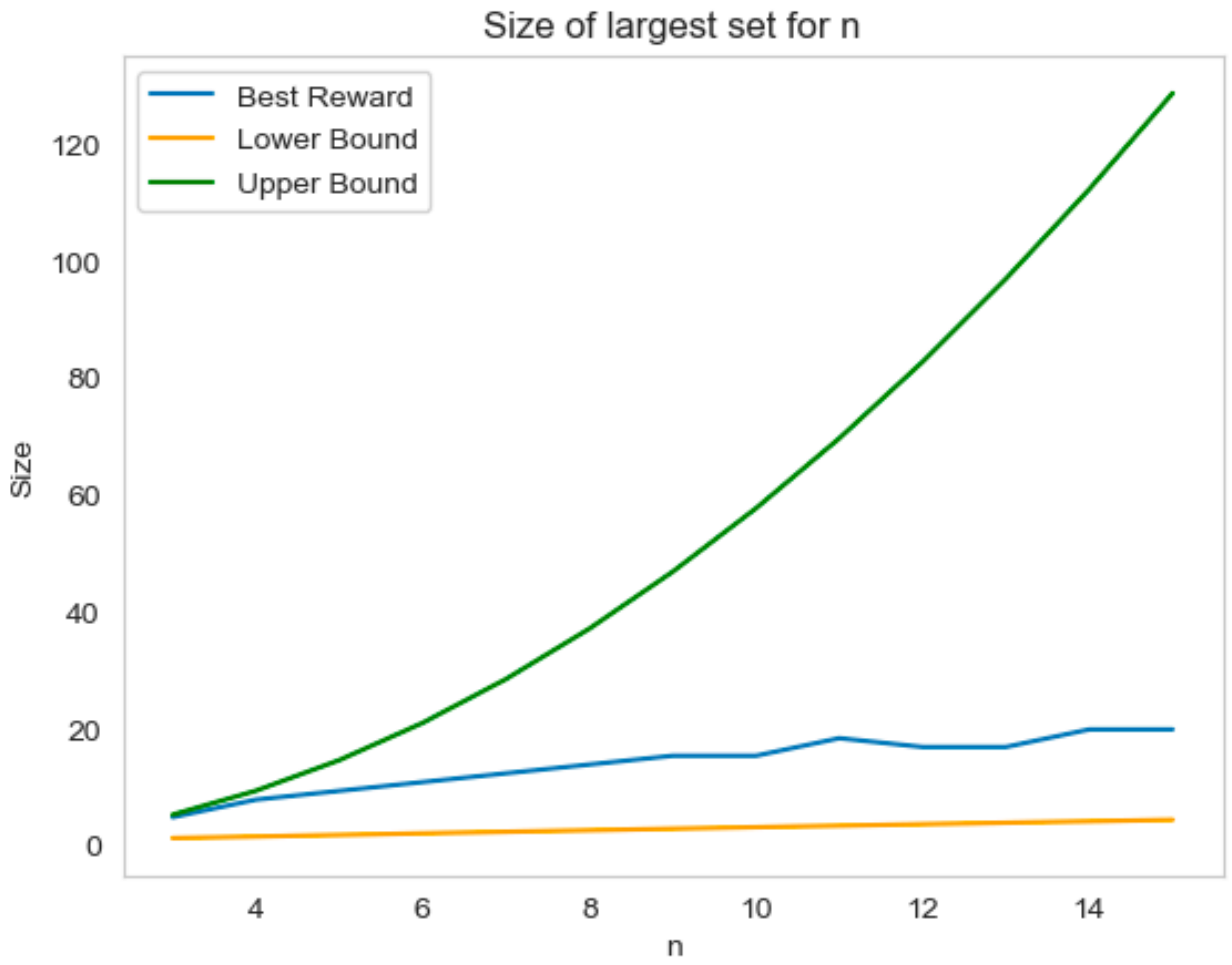
Results


Size of largest set for n
Best Reward
Lower Bound
Size
n
14
12
10
8
6
4
2
4
6
8
10
12
14

Results

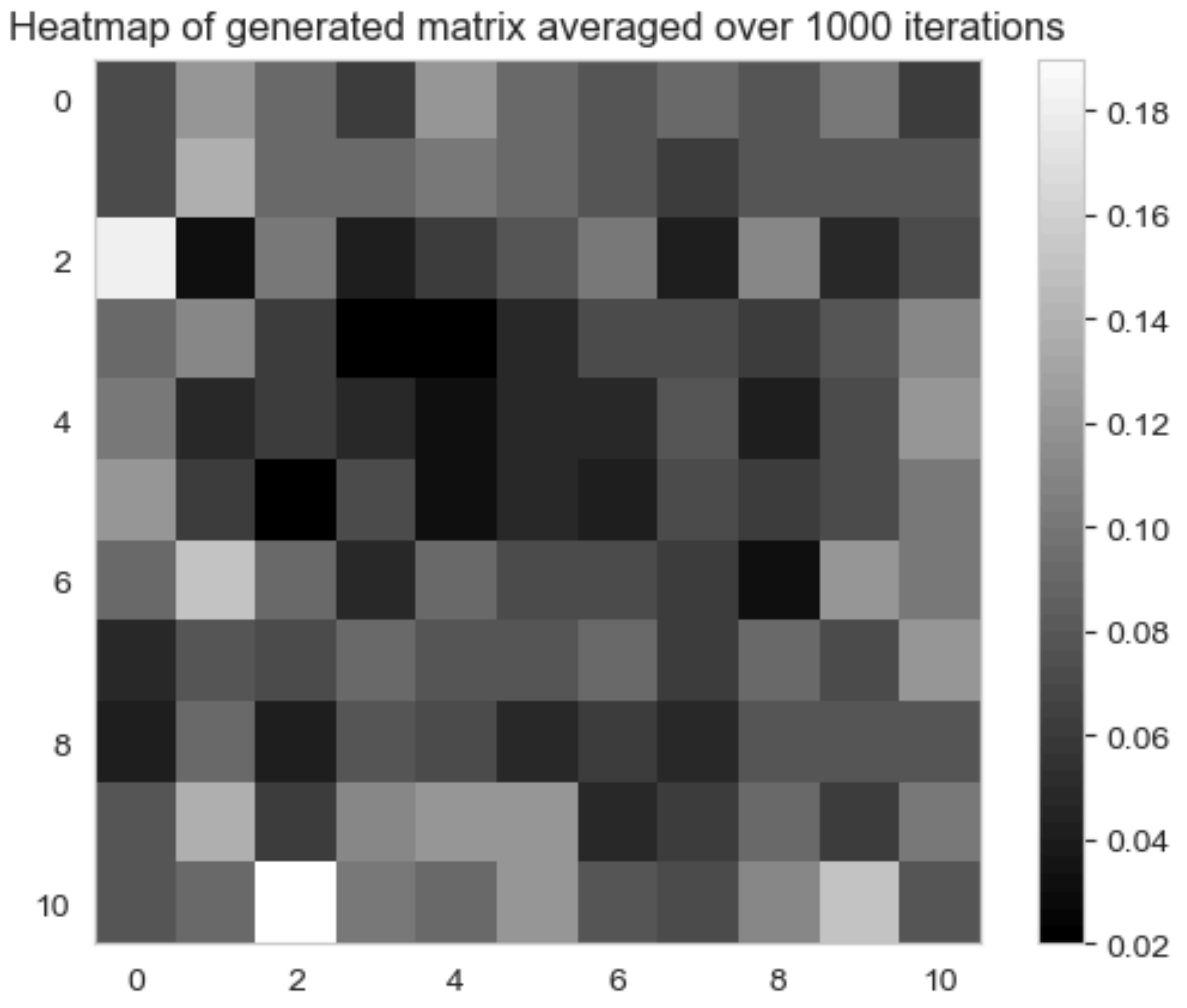


Size of largest set for n
Best Reward
Lower Bound
Size
n
14
12
10
8
6
4
2
4
6
8
10
12
14



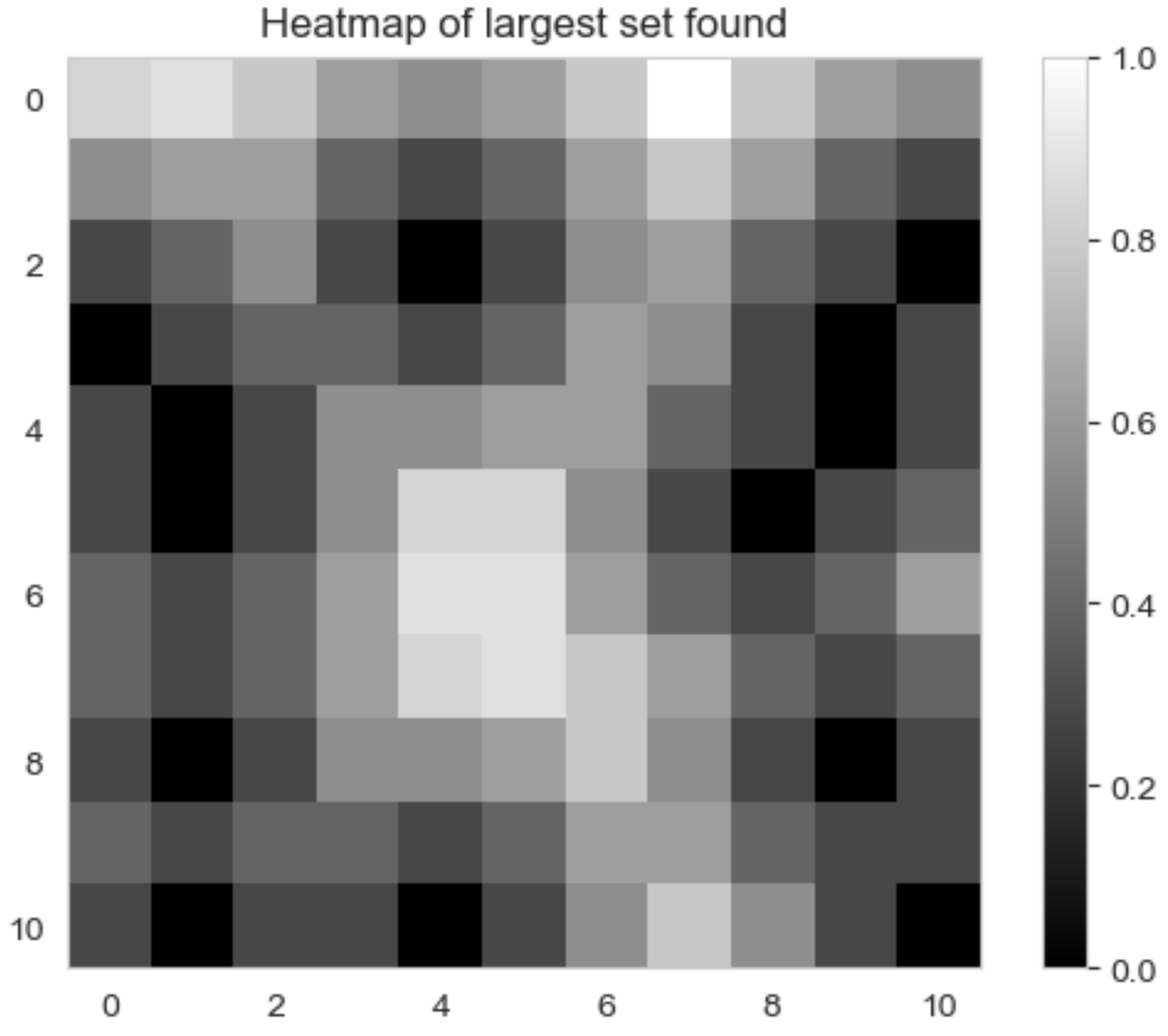
Size of largest set for n
Best Reward
Lower Bound
Upper Bound
Size
n
120
100
80
60
40
20
0
4
6
8
10
12
14

Results



Lower bound method

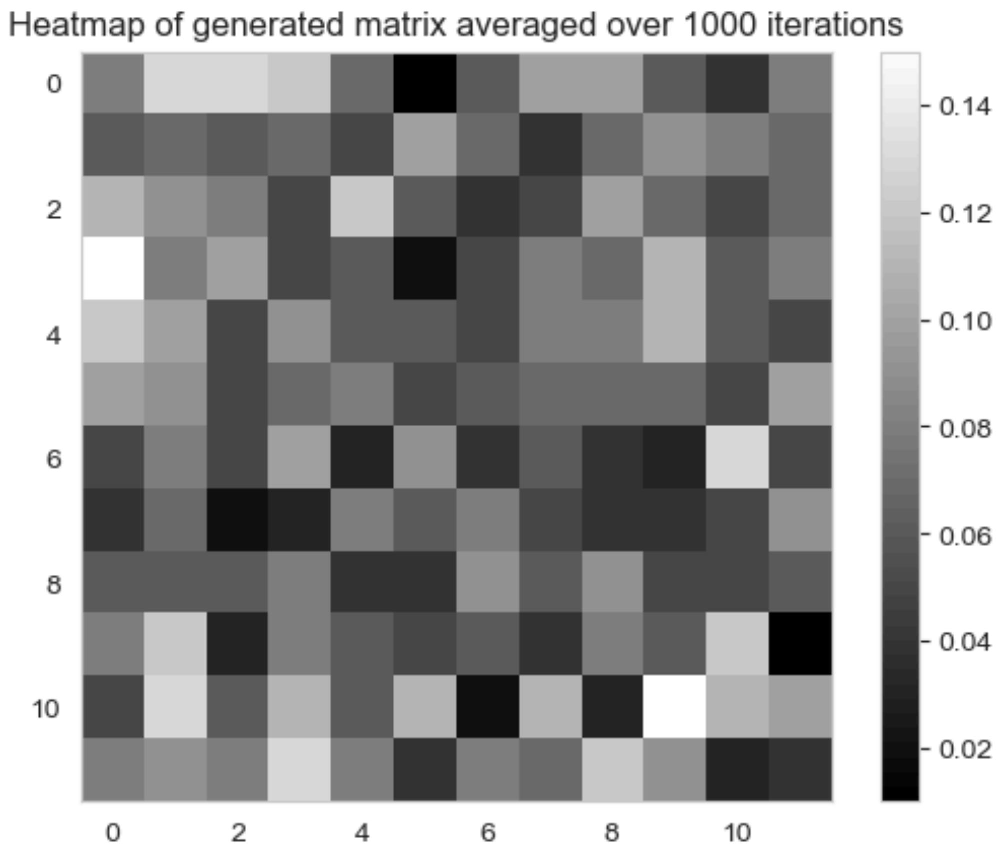
N = 11



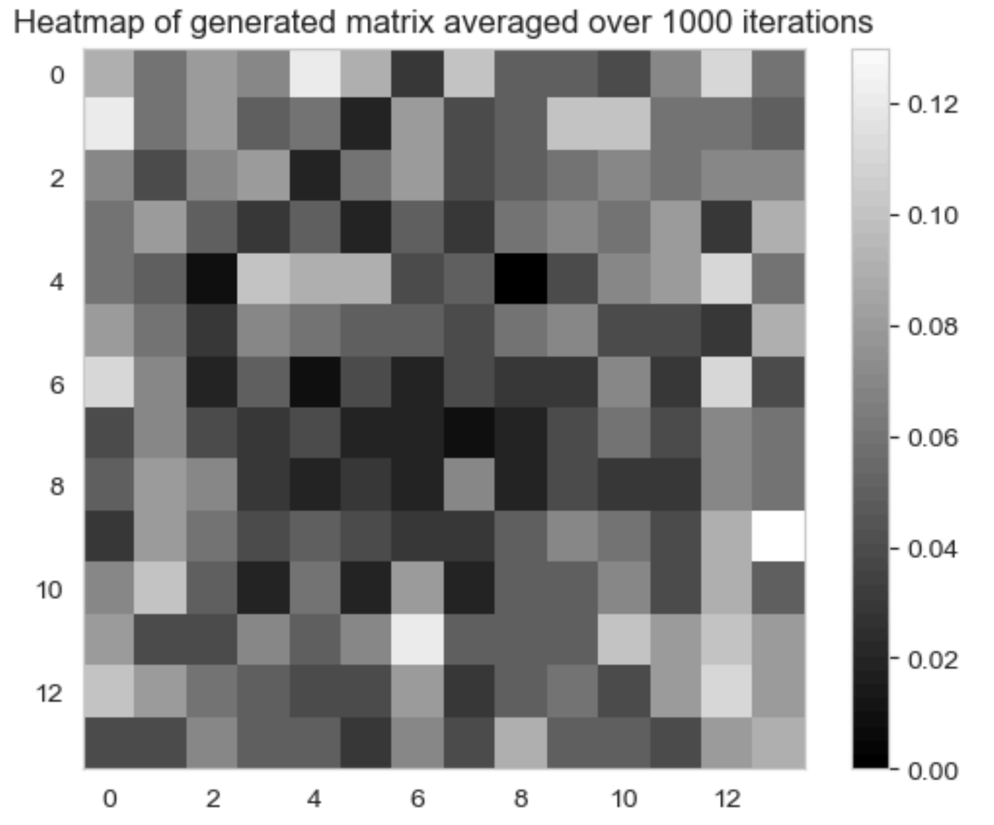
RL generated map

N = 11

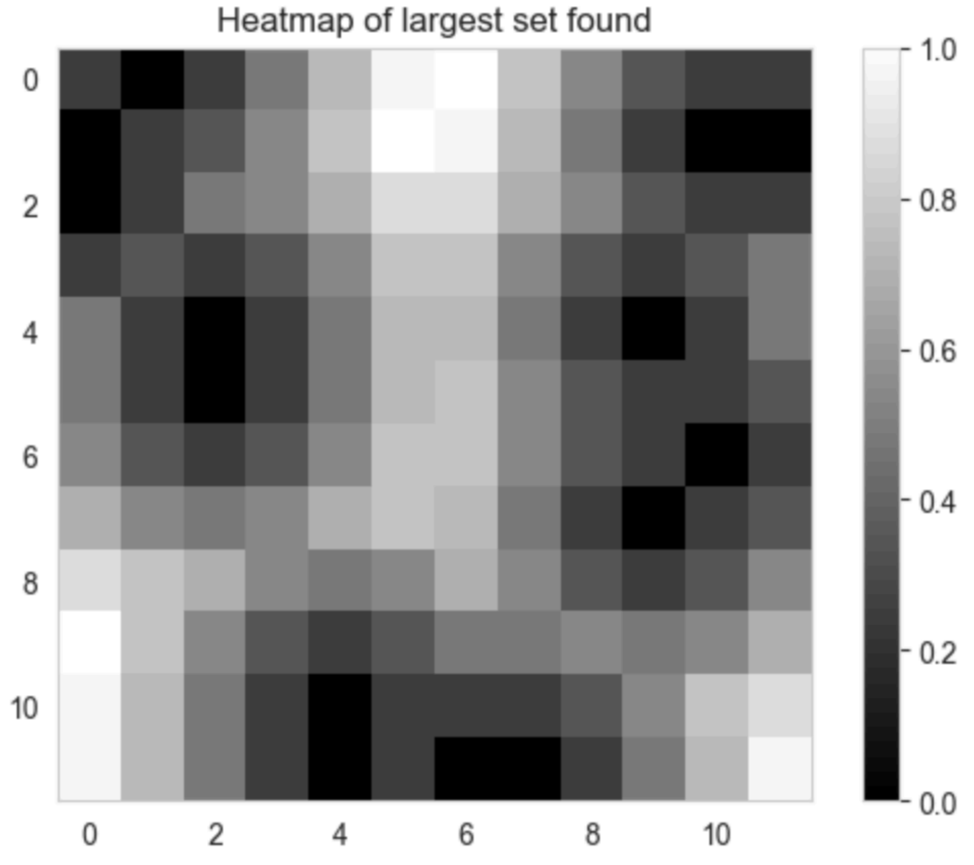
Results



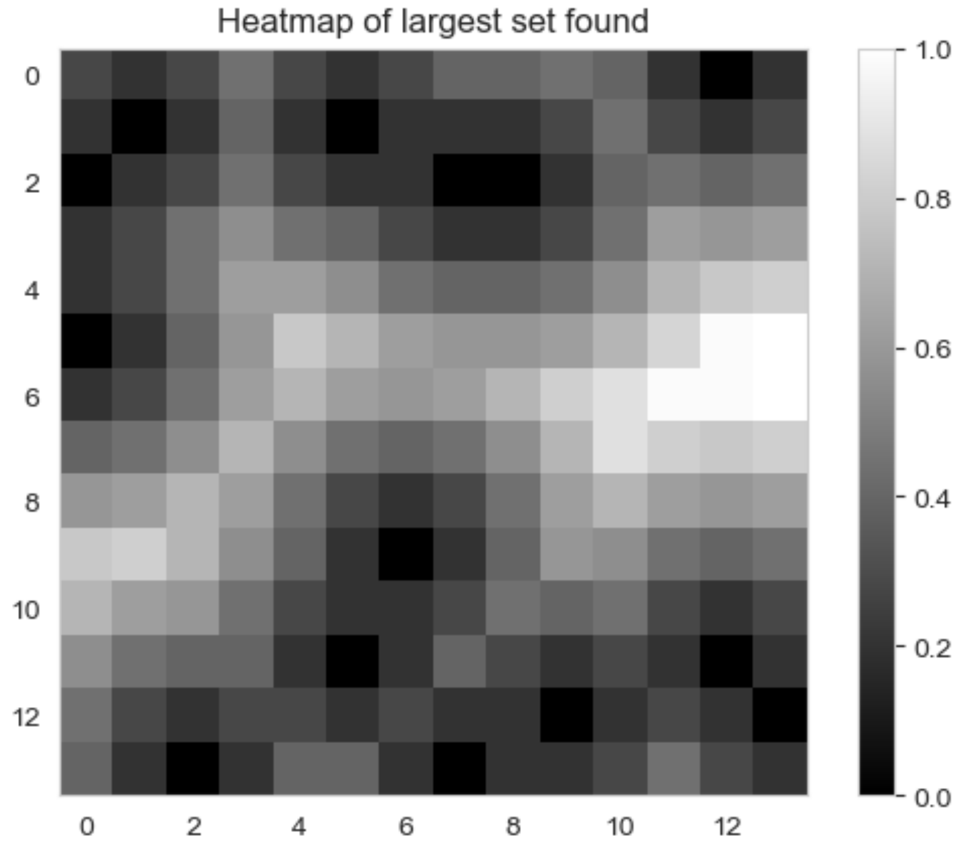
N = 12



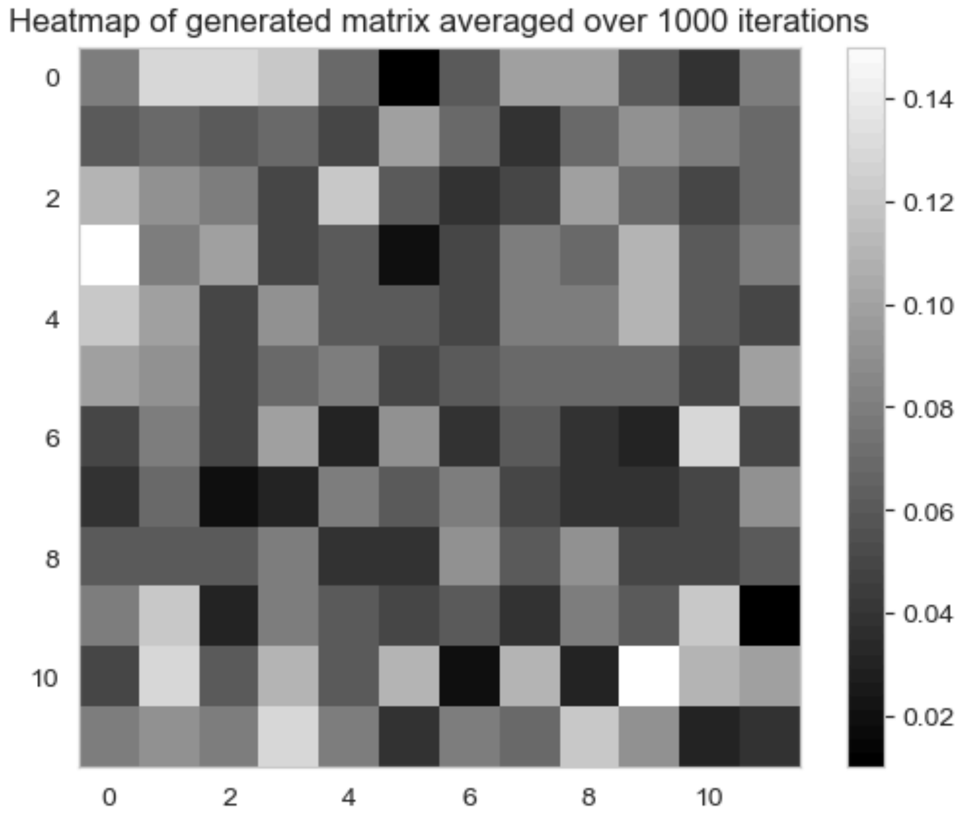
N = 14



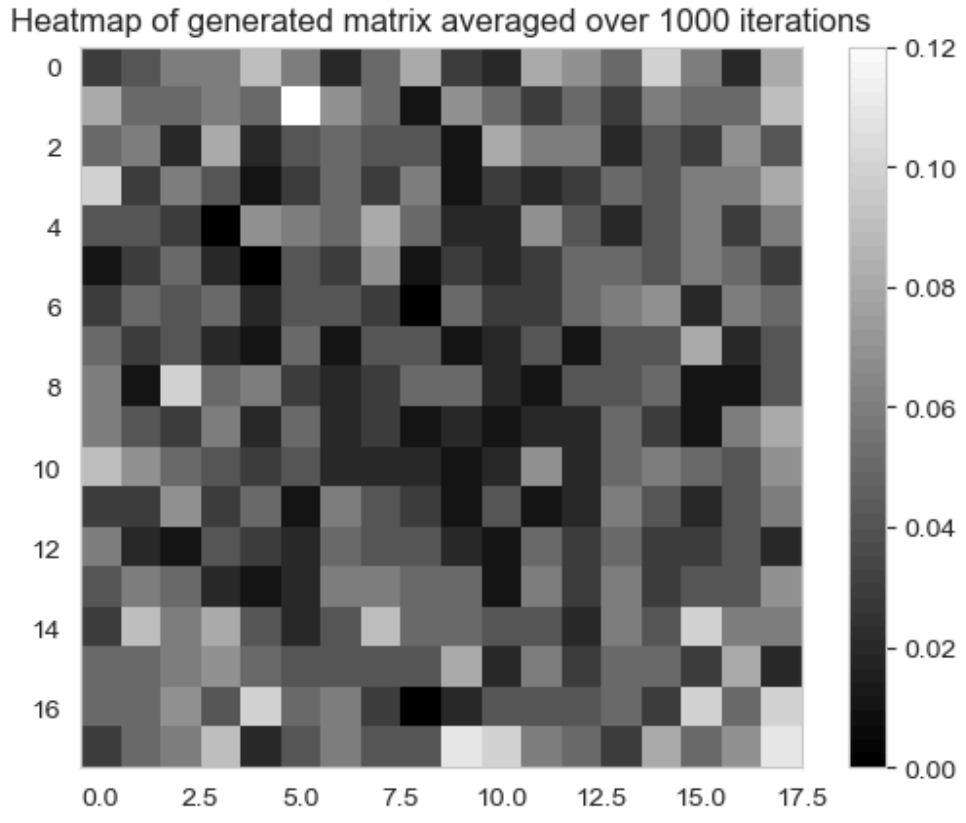
N = 12



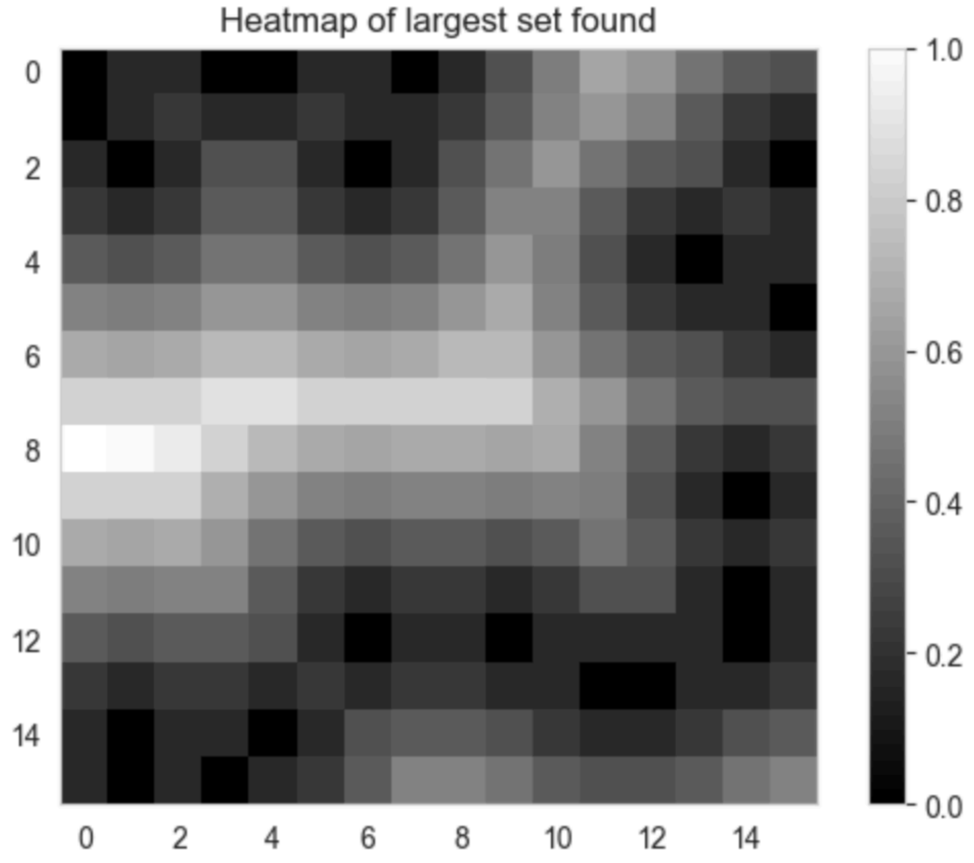
N = 14



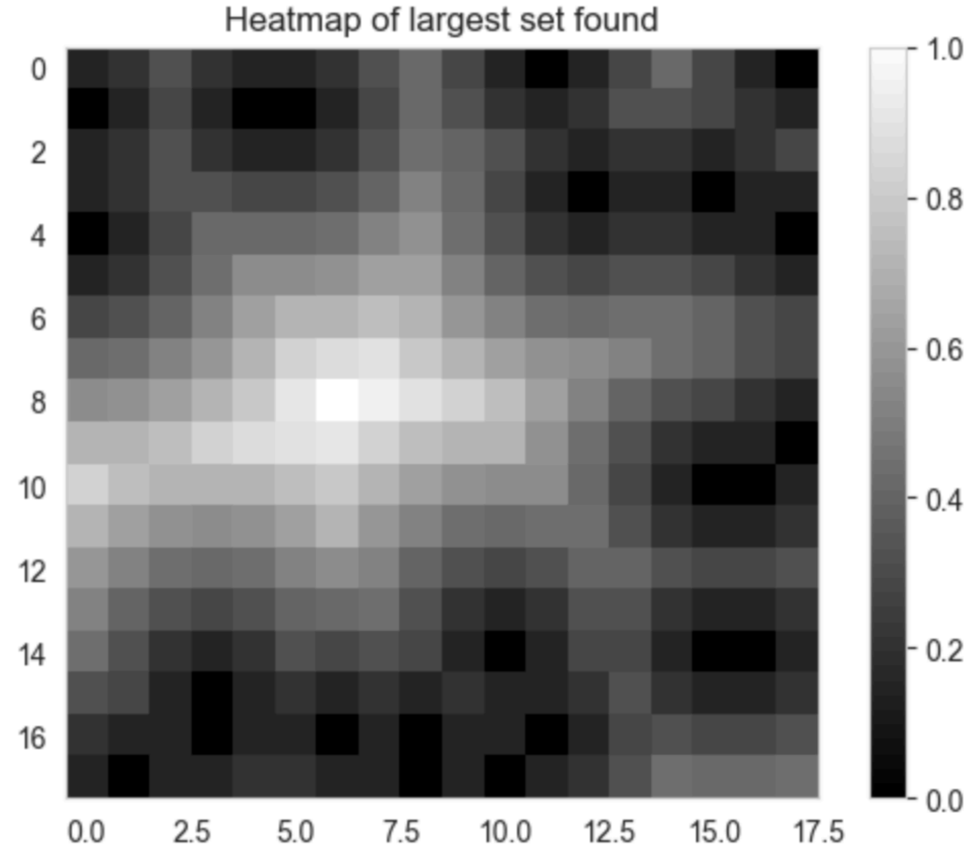
N = 16



N = 18



N = 16



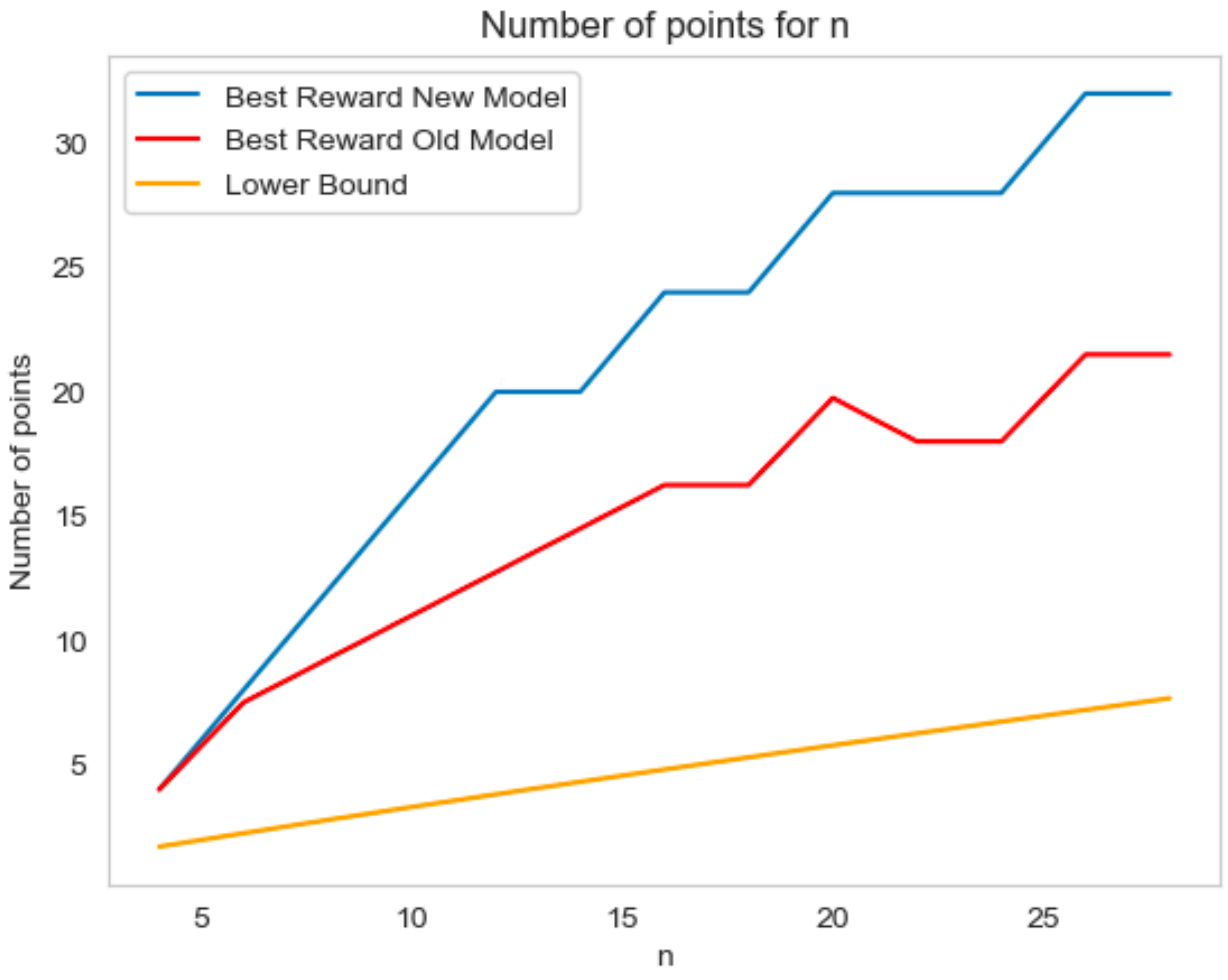
N = 18

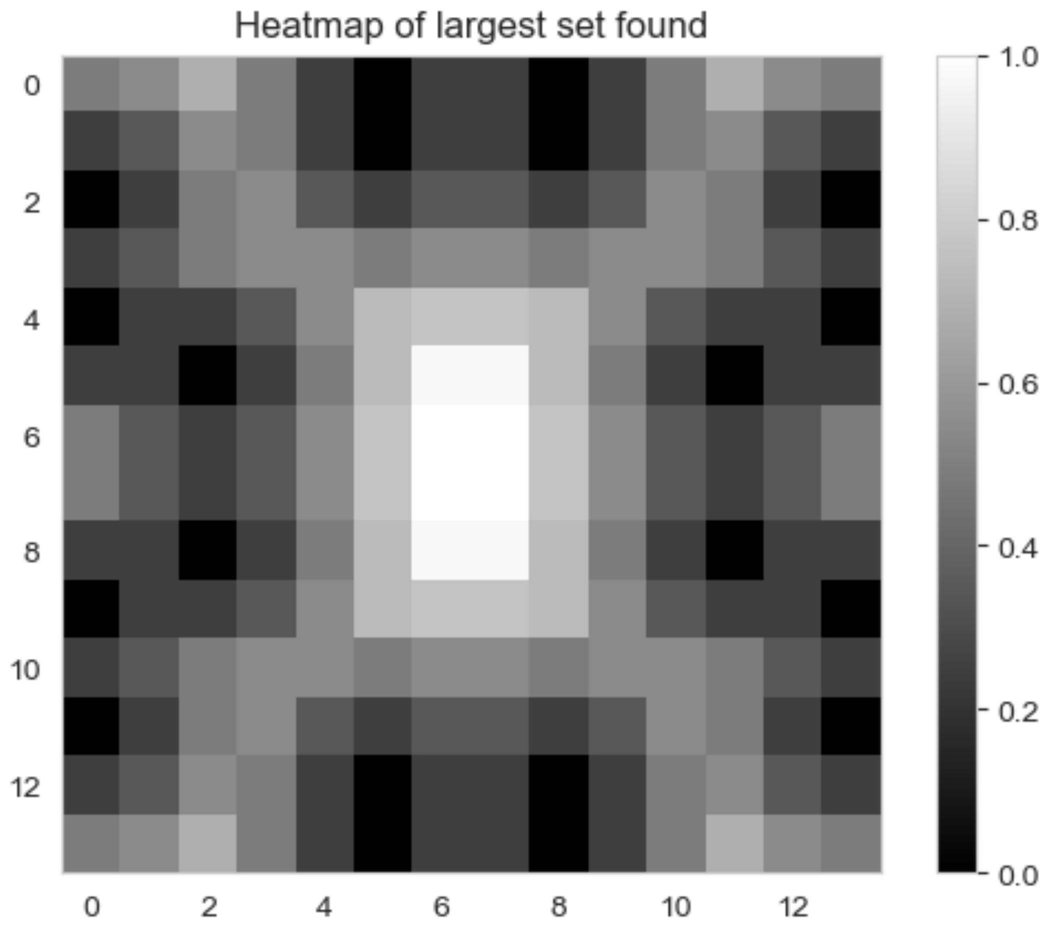
Lower bound method

RL generated map

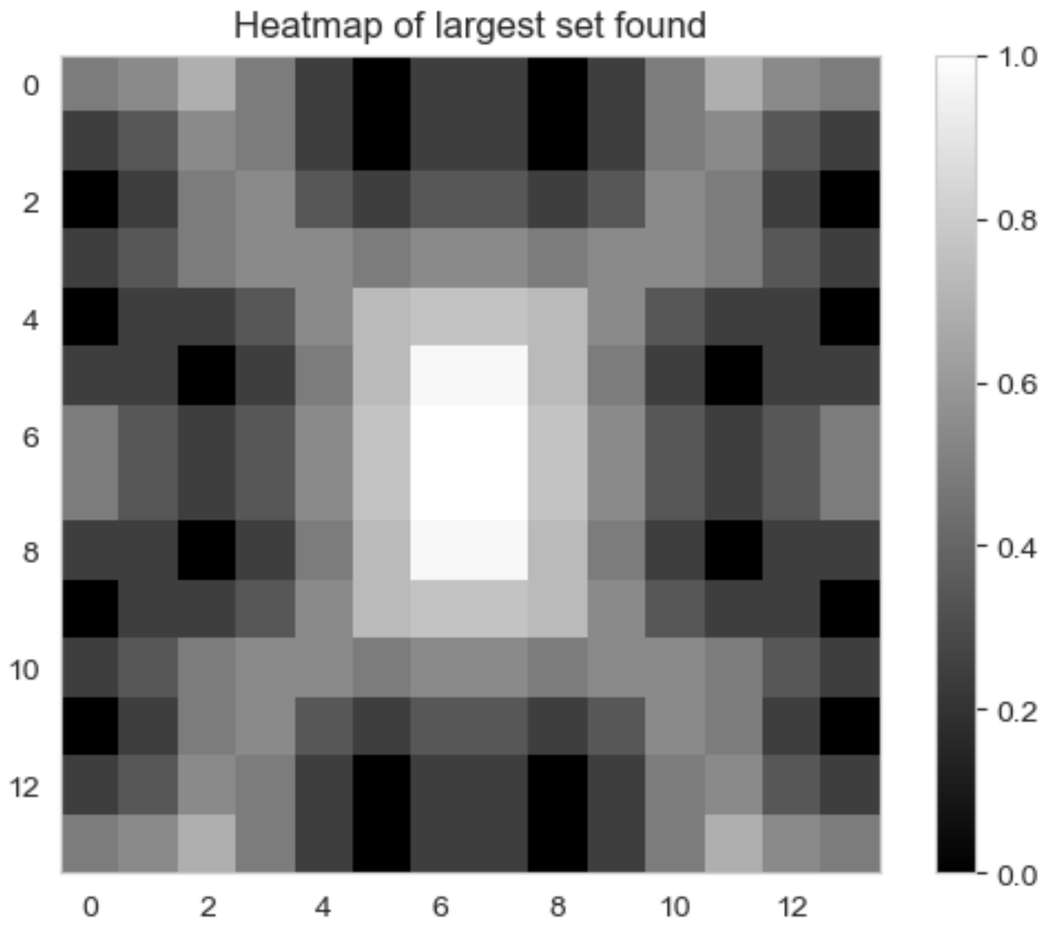
Results

When we reward the model for symmetric generation and higher edge densities.

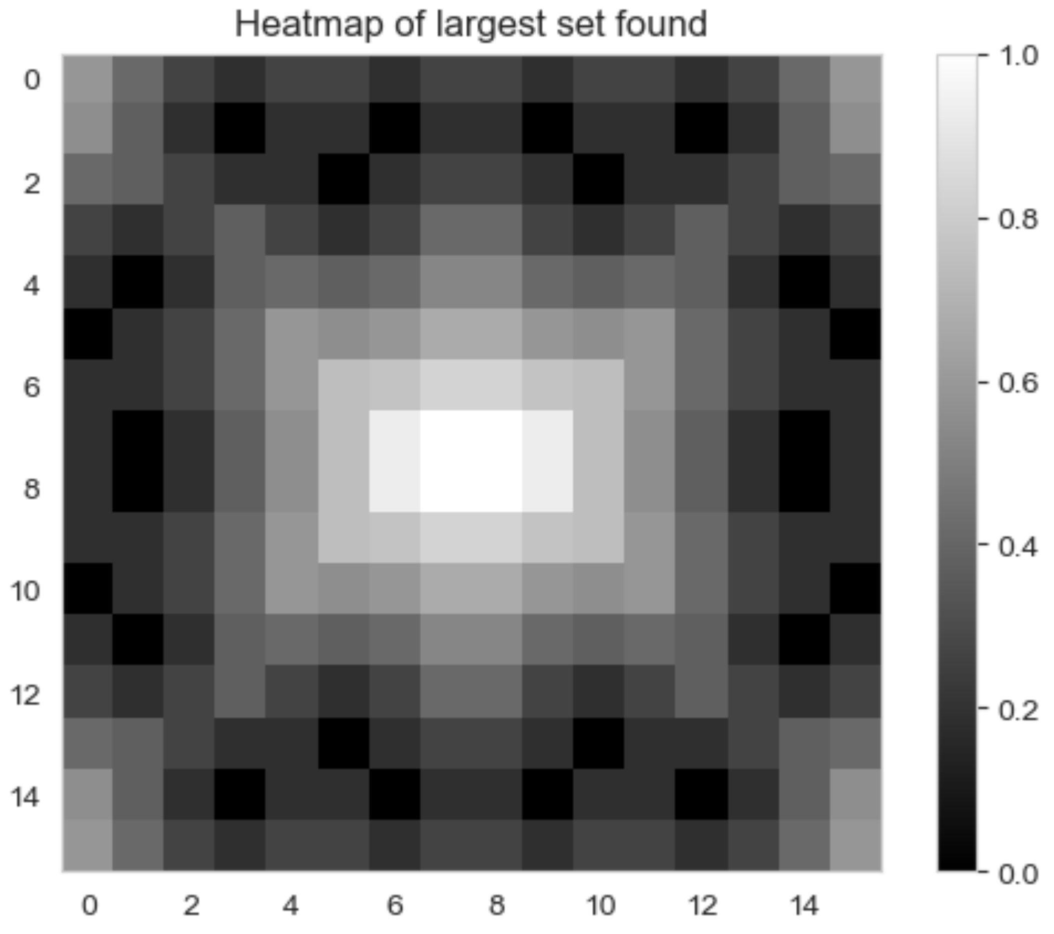




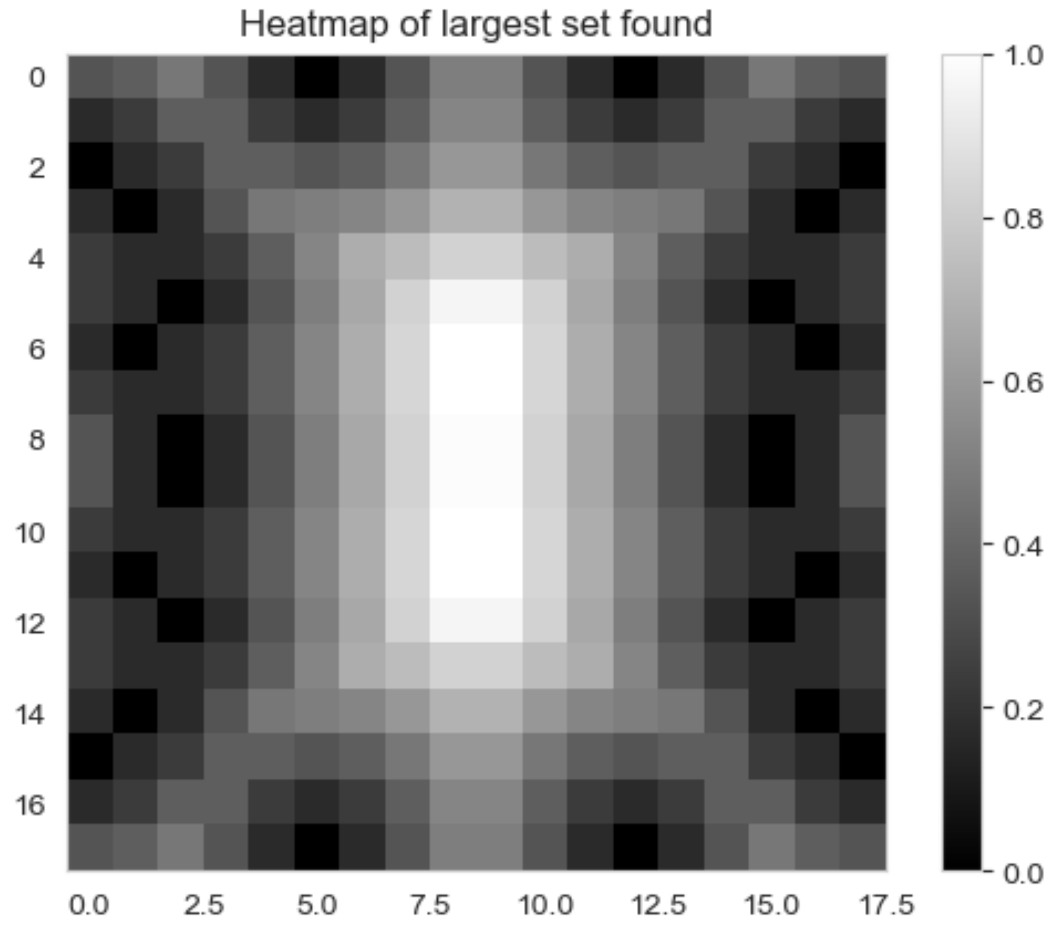
n = 12



n = 14



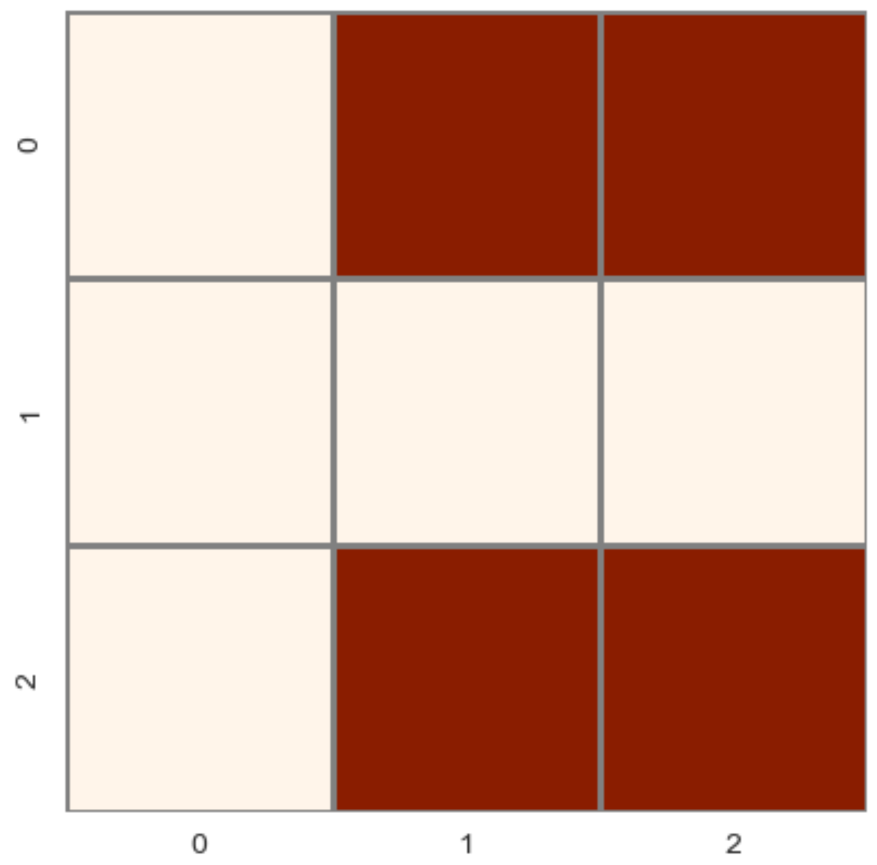
n = 16



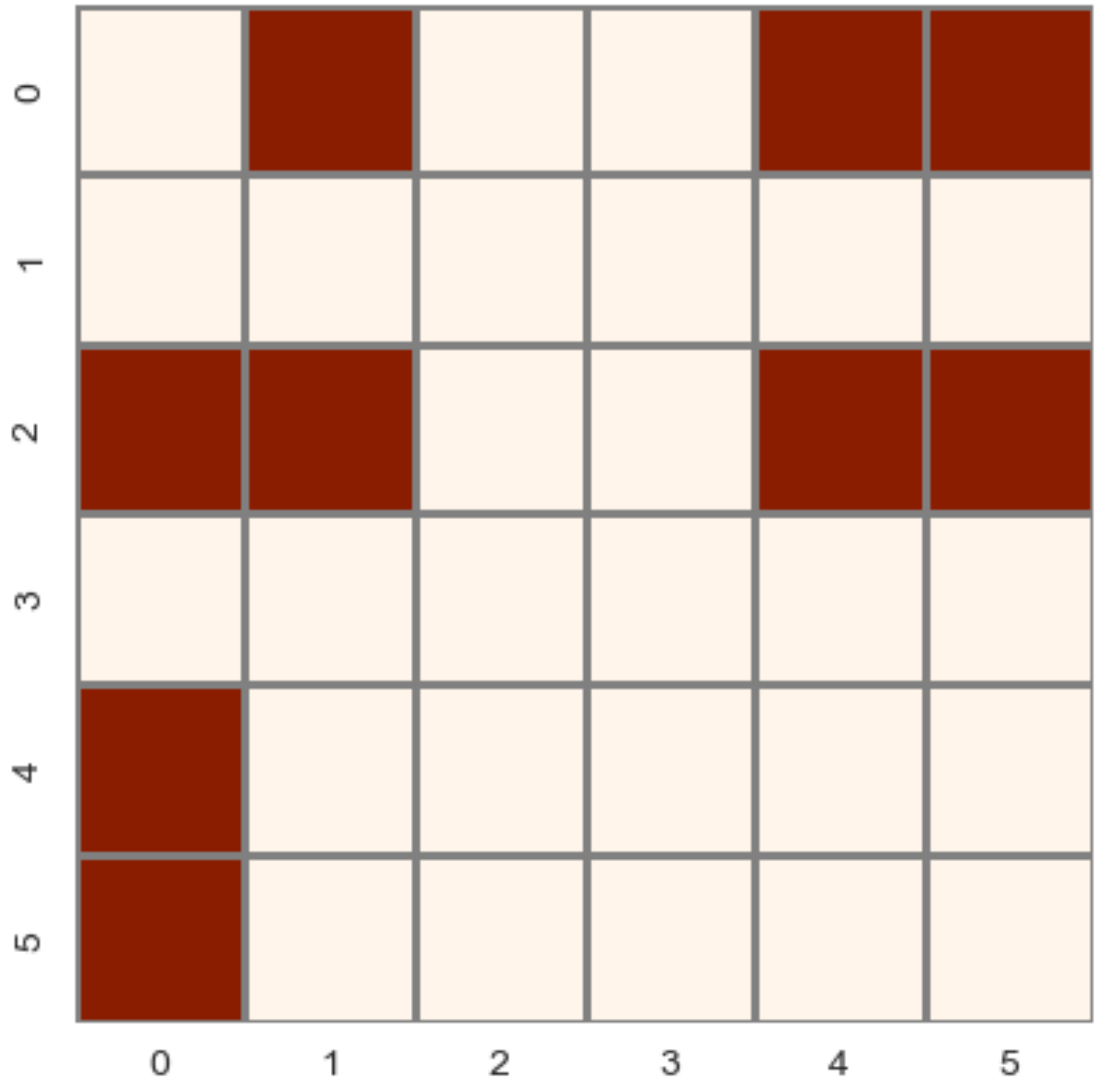
n = 18

Observations

Other observations:



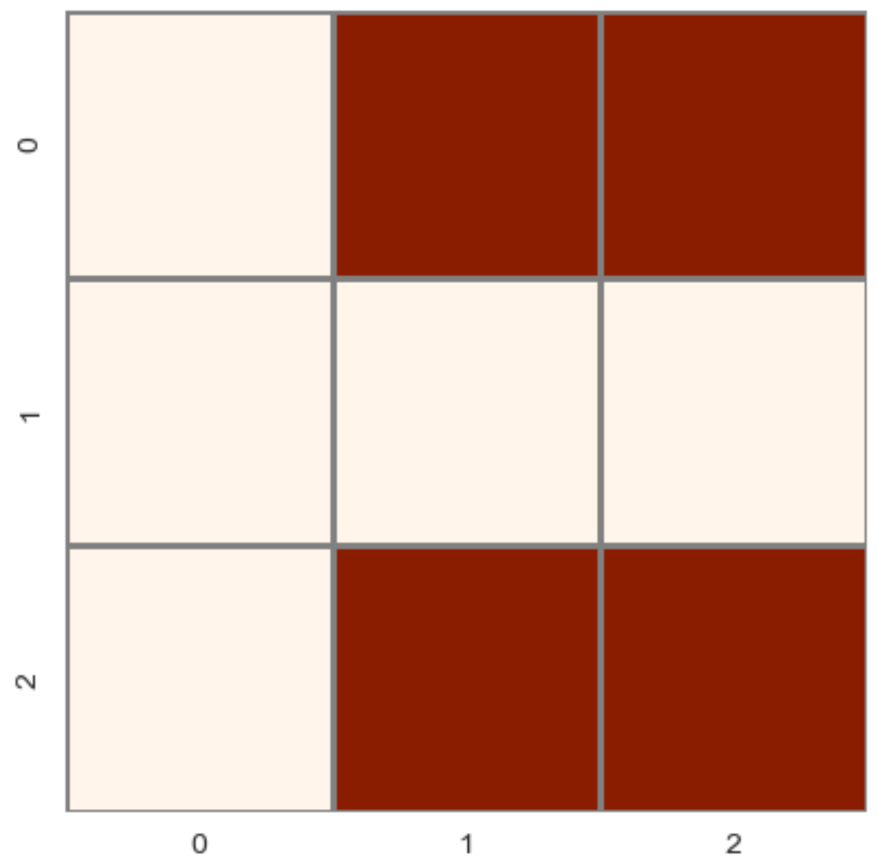
n = 3



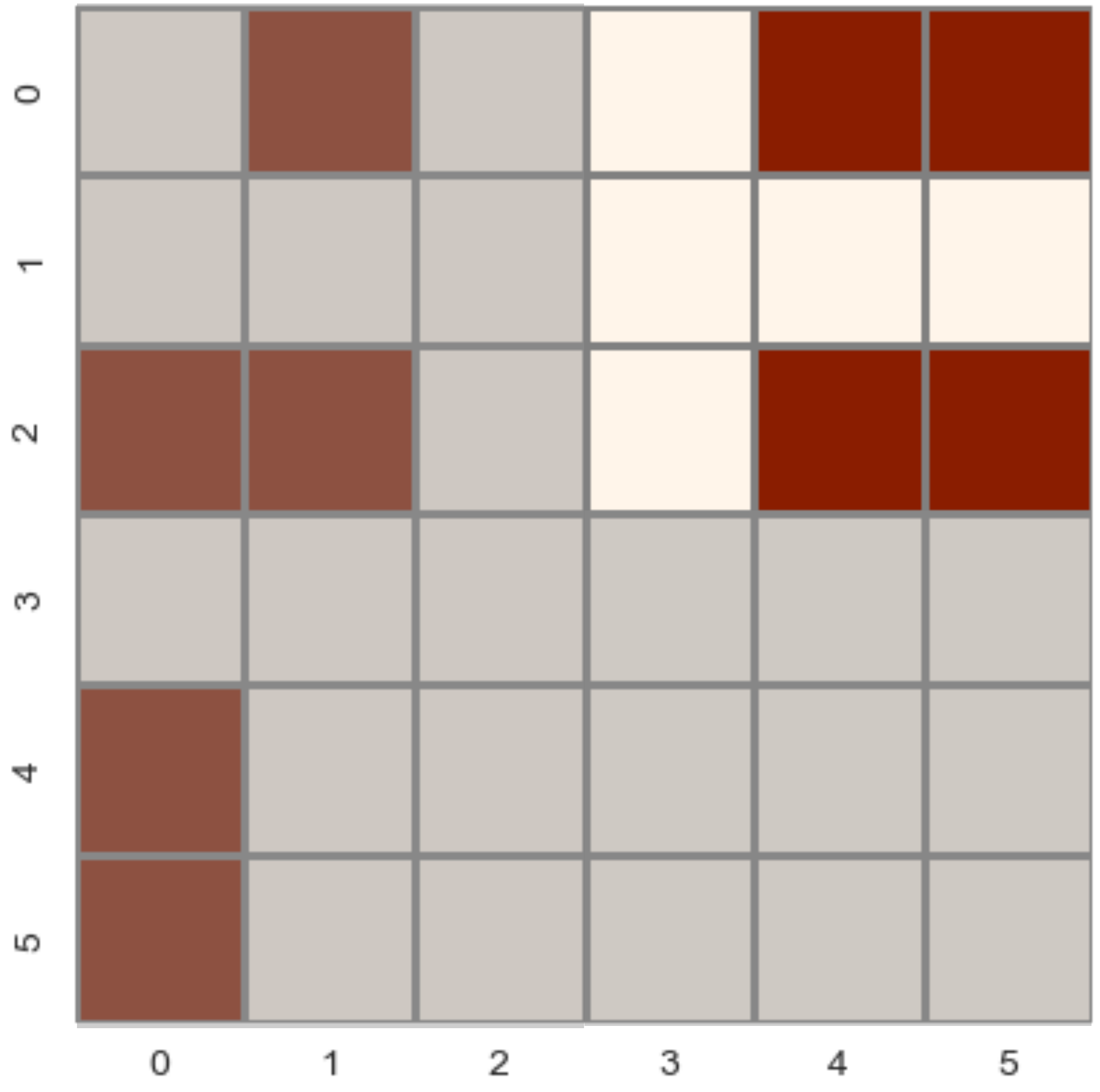
n = 6

Observations

Other observations:



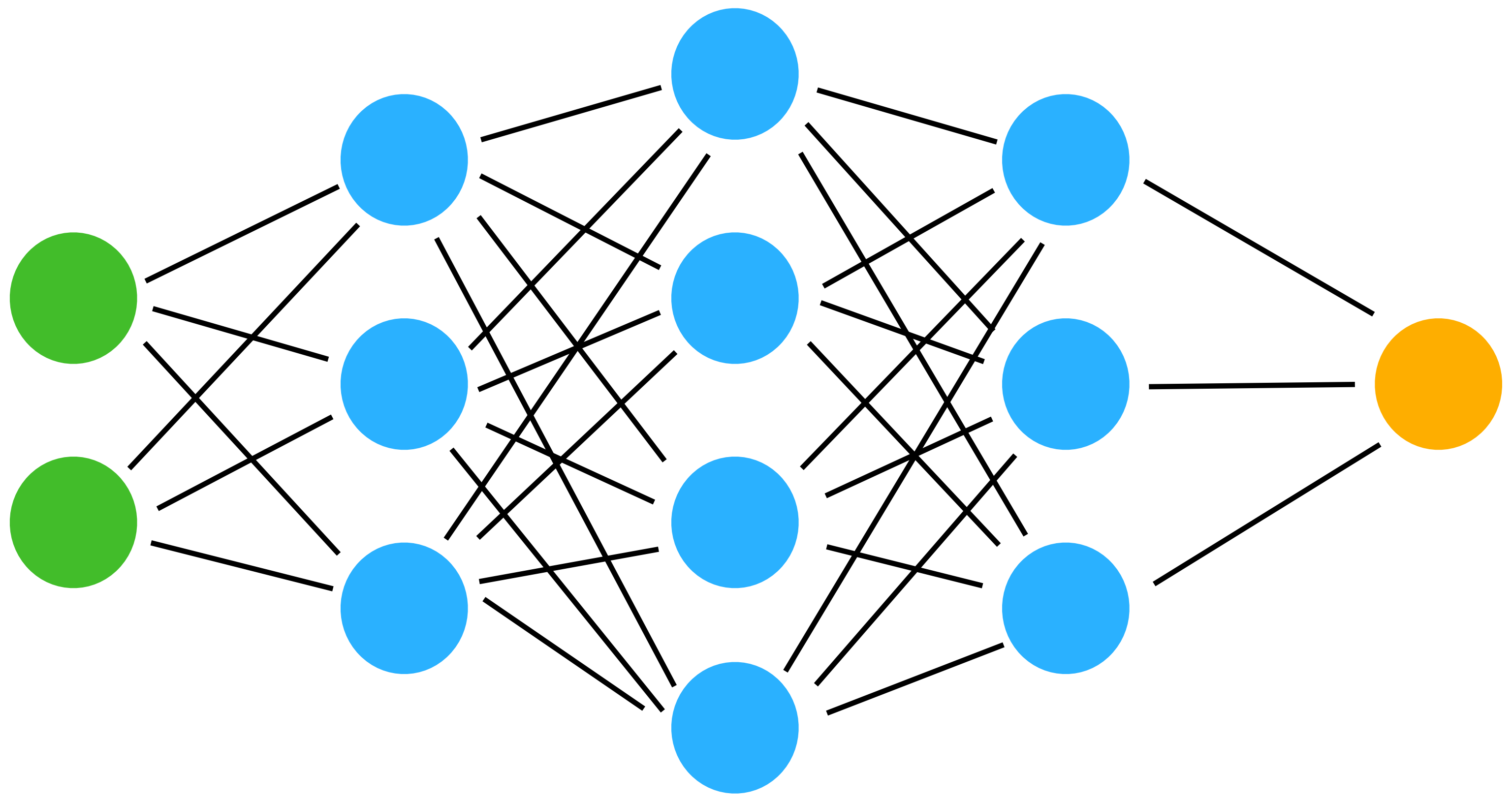
n = 3



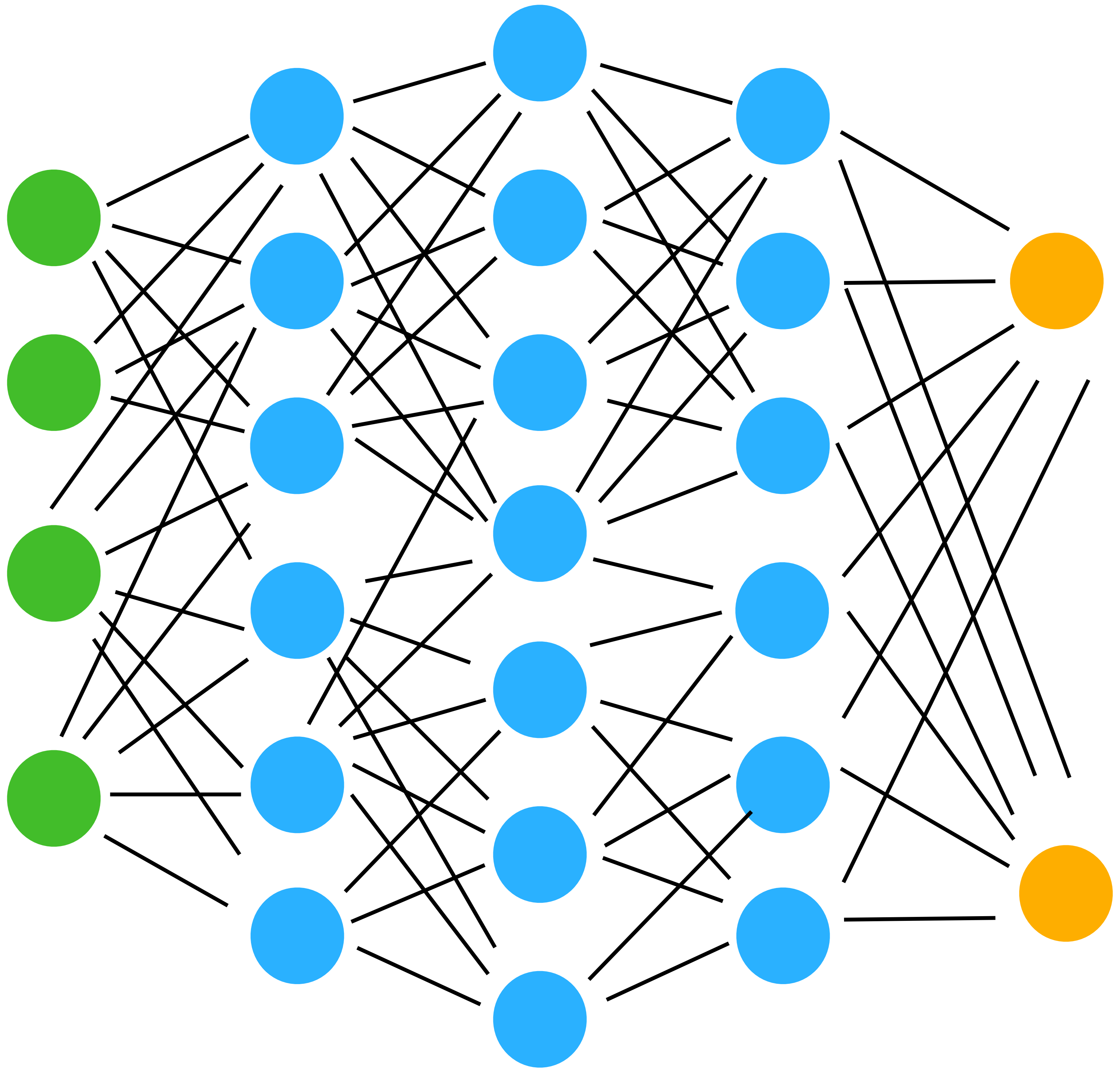
n = 6

Model

Inductive learning Framework



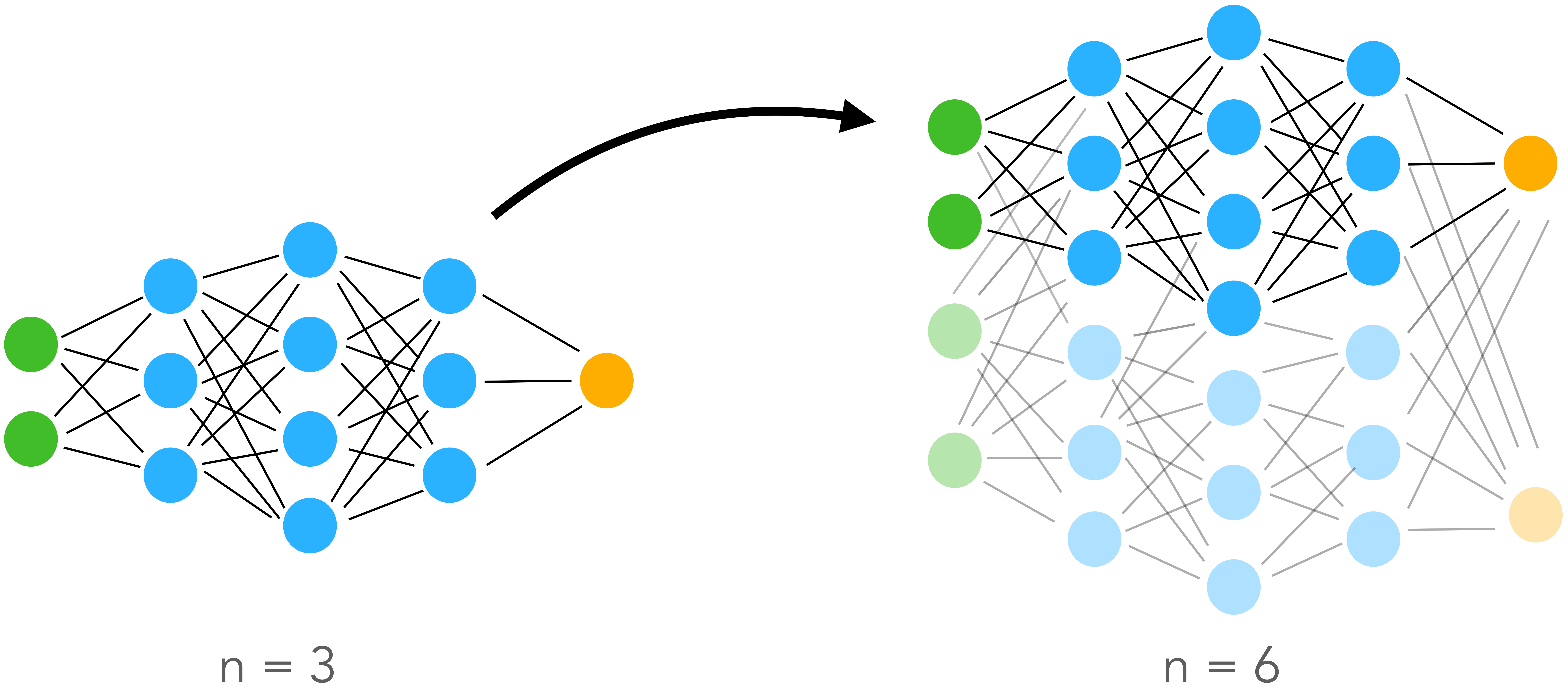
n = 3



n = 6

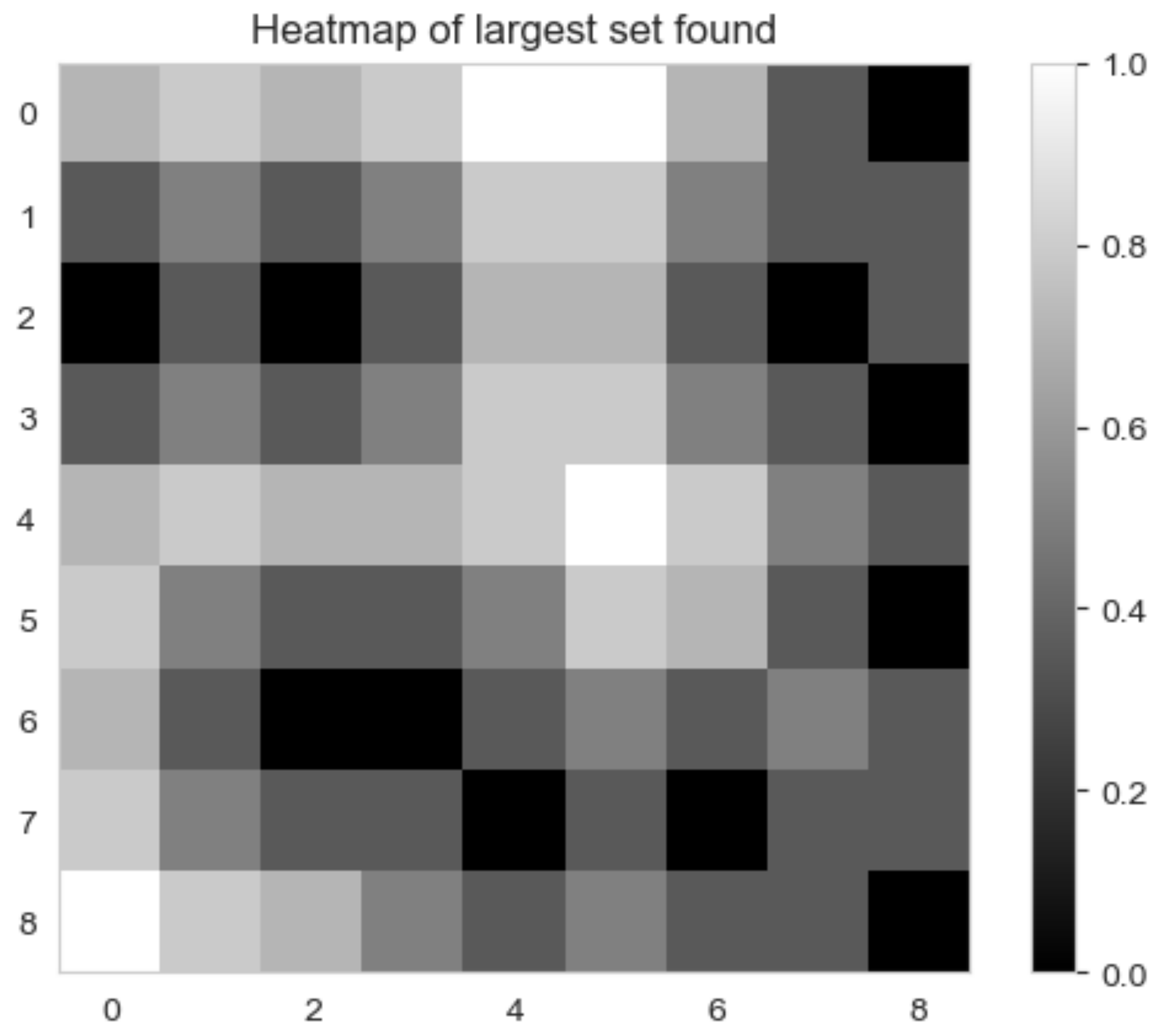
Model

Inductive learning Framework

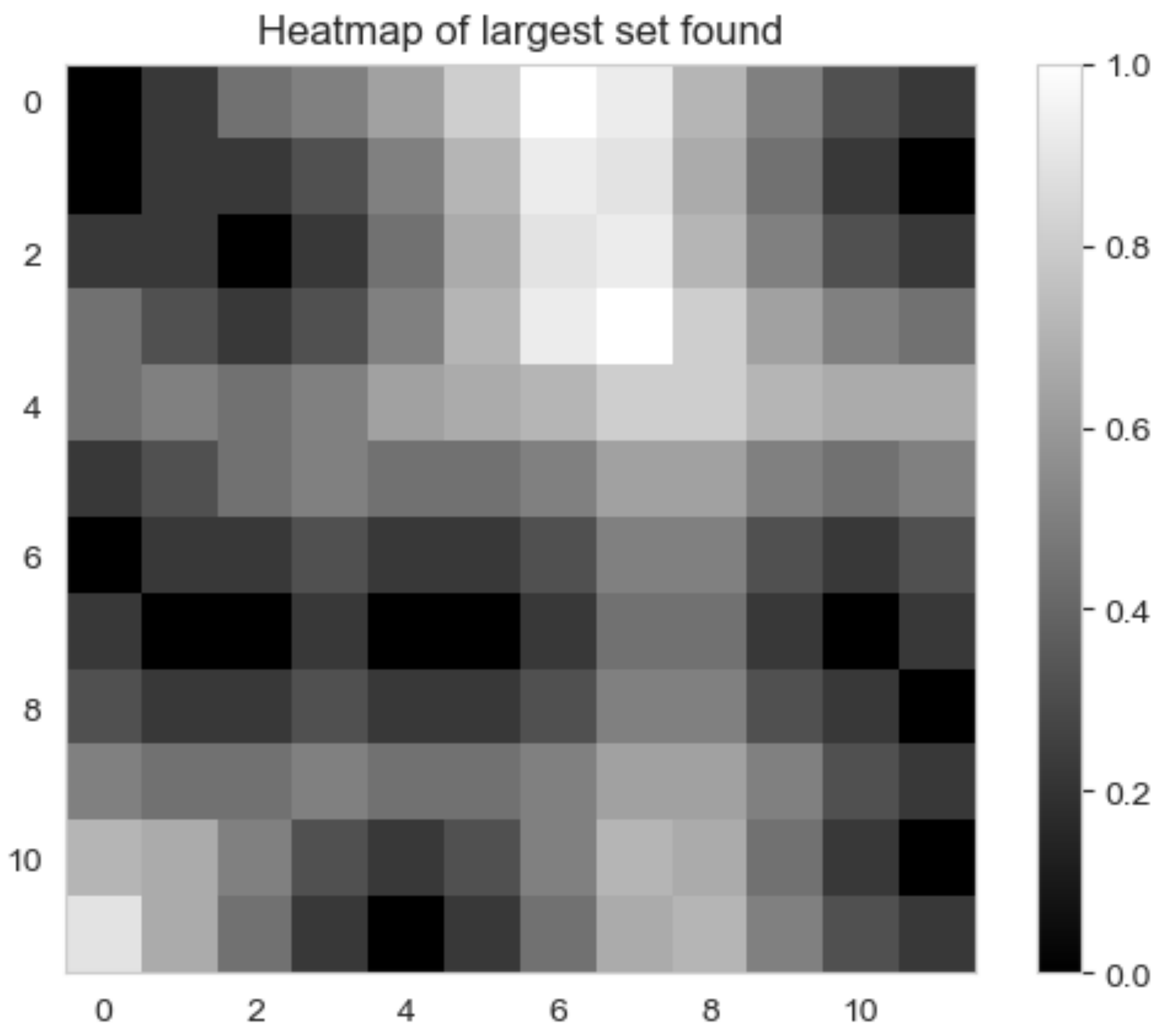


Results (so far)

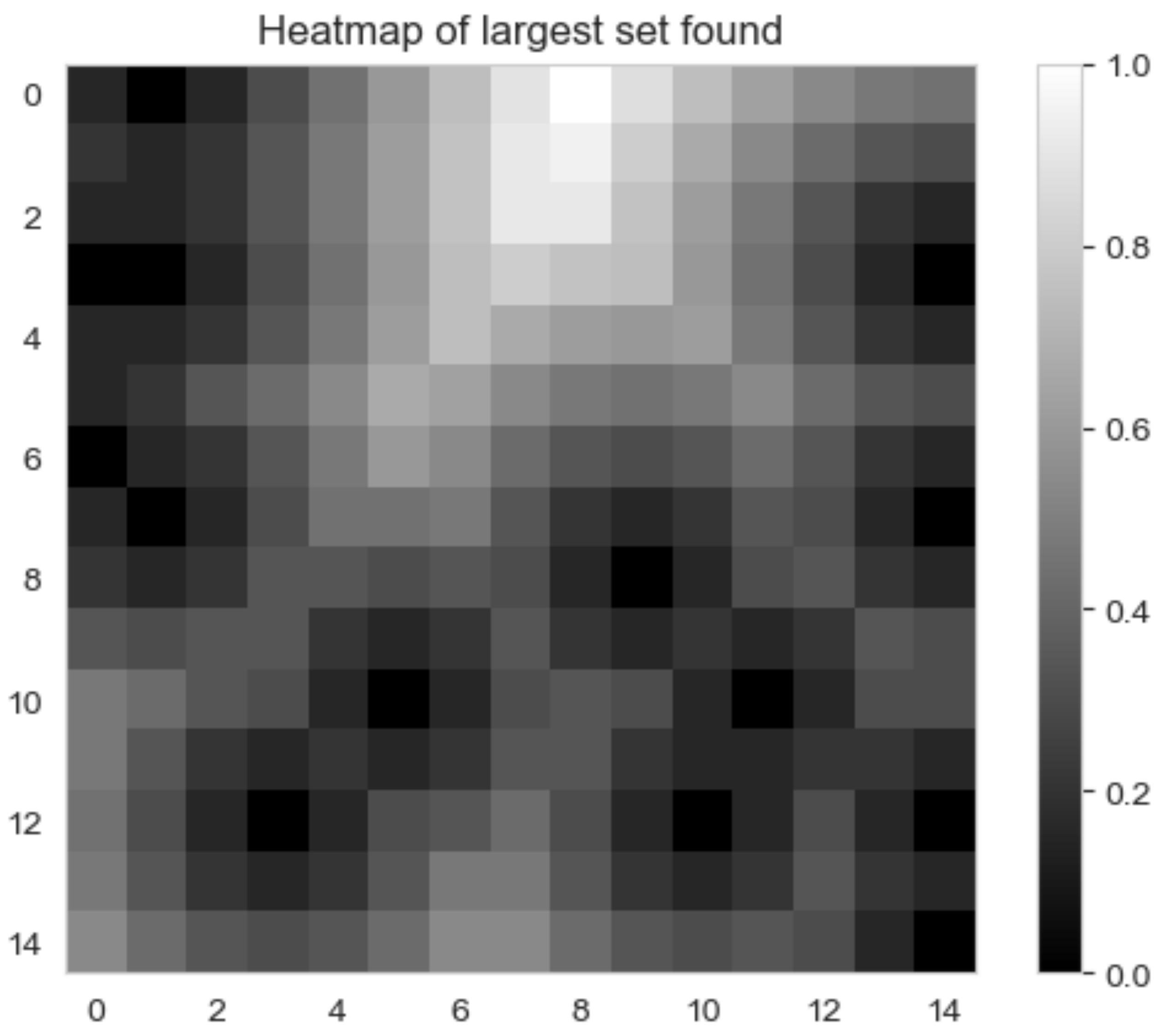
Similar heatmaps



n = 9



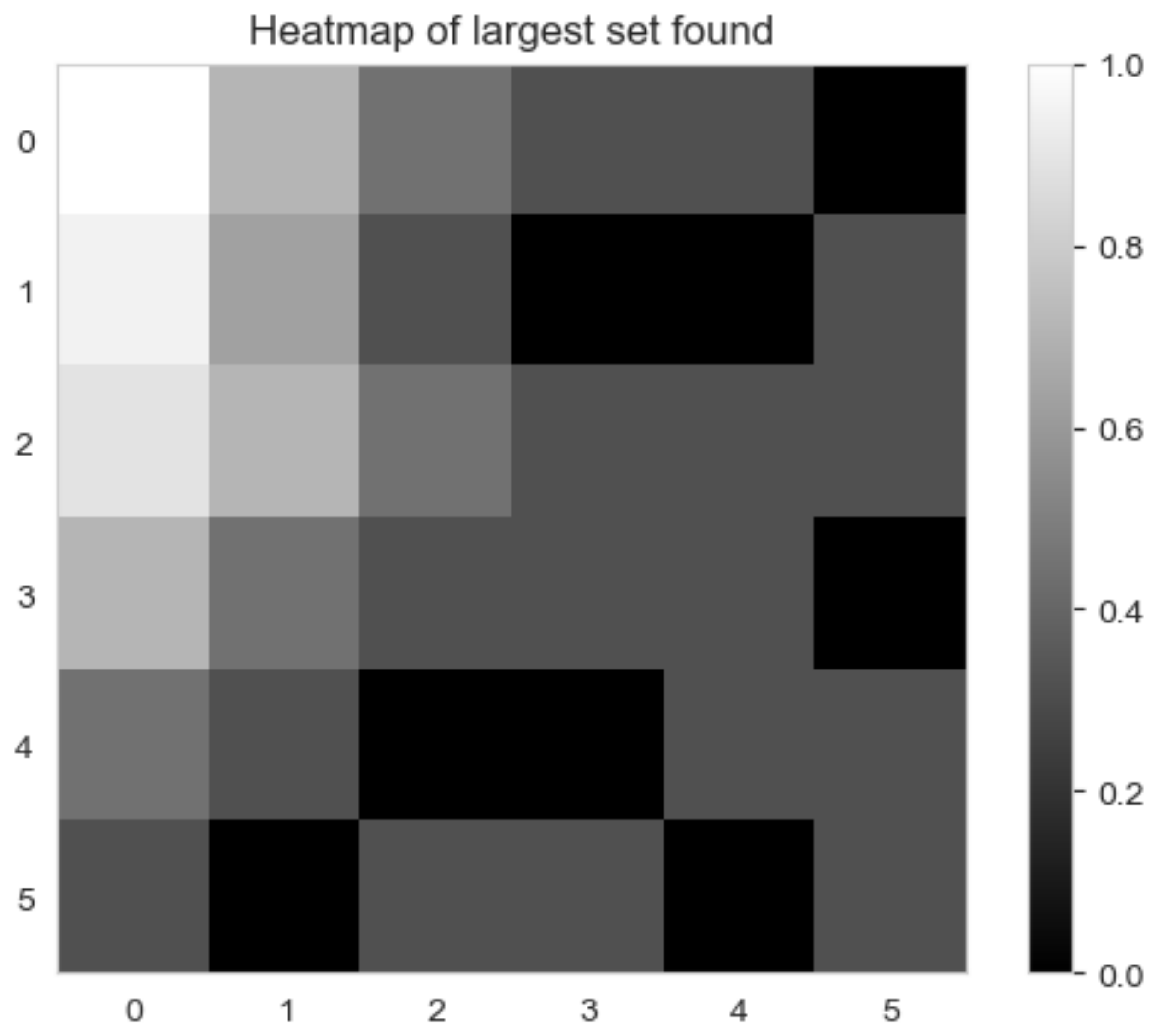
n = 12



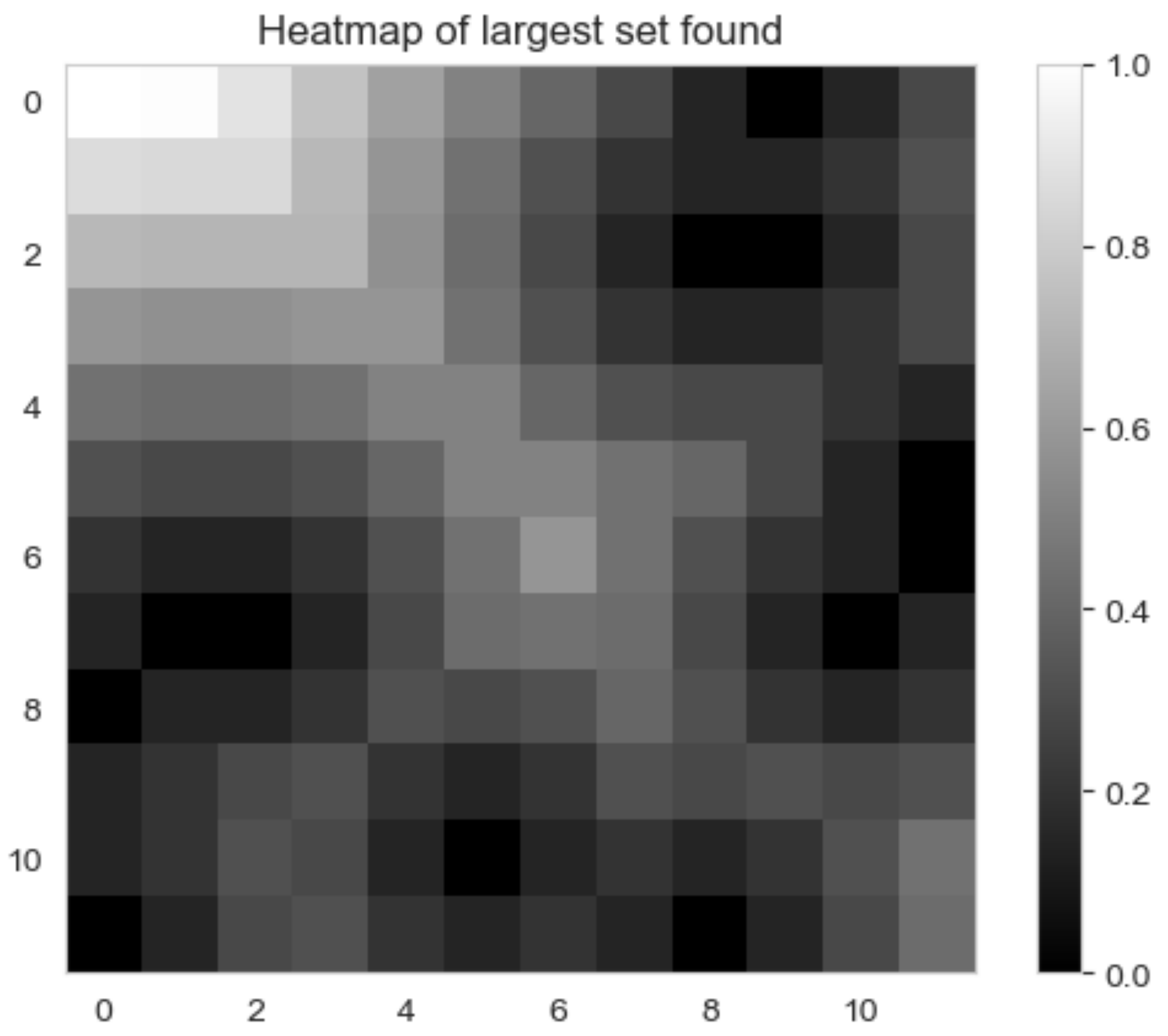
n = 15

Results (so far)

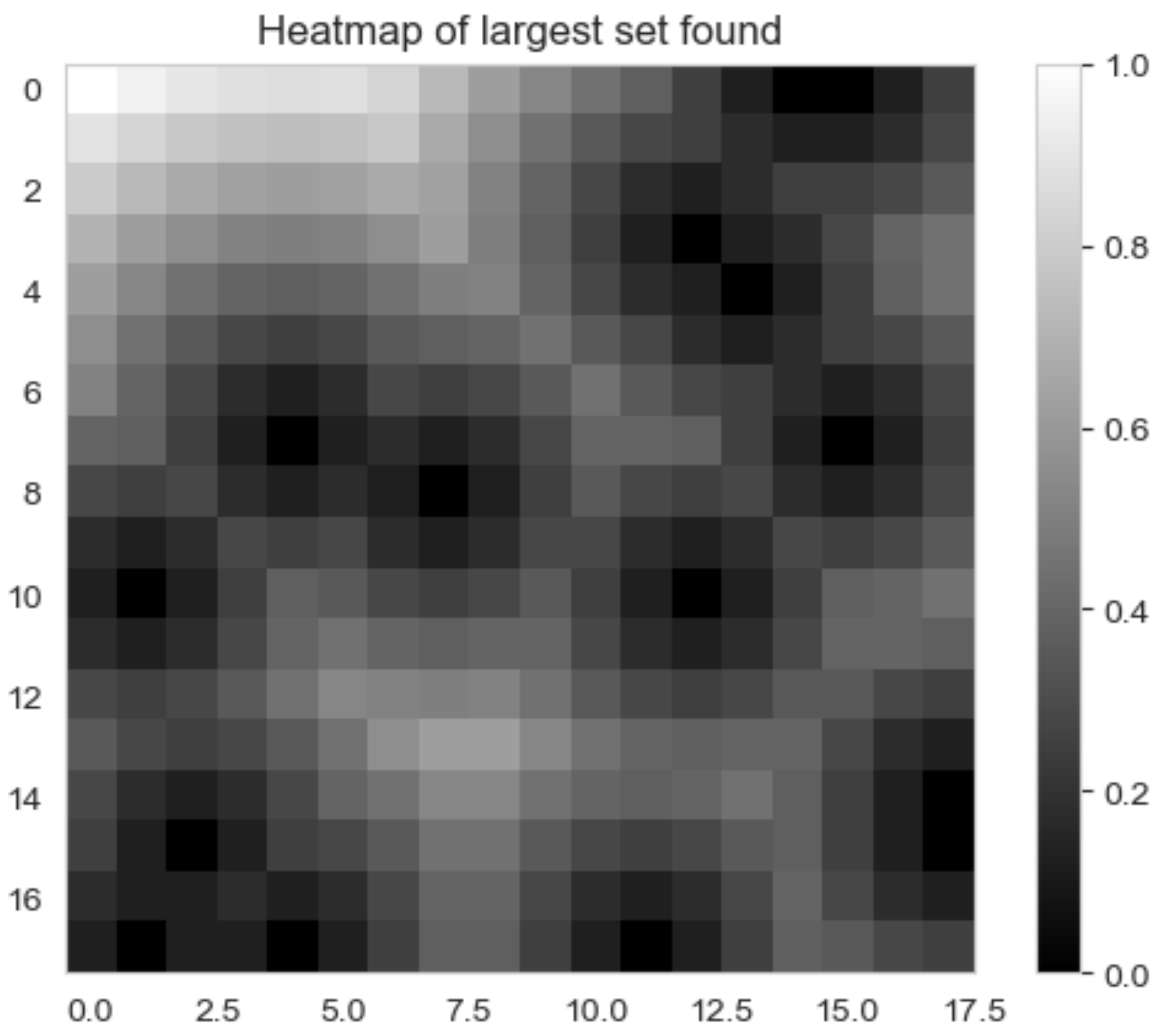
Similar heatmaps



n = 6



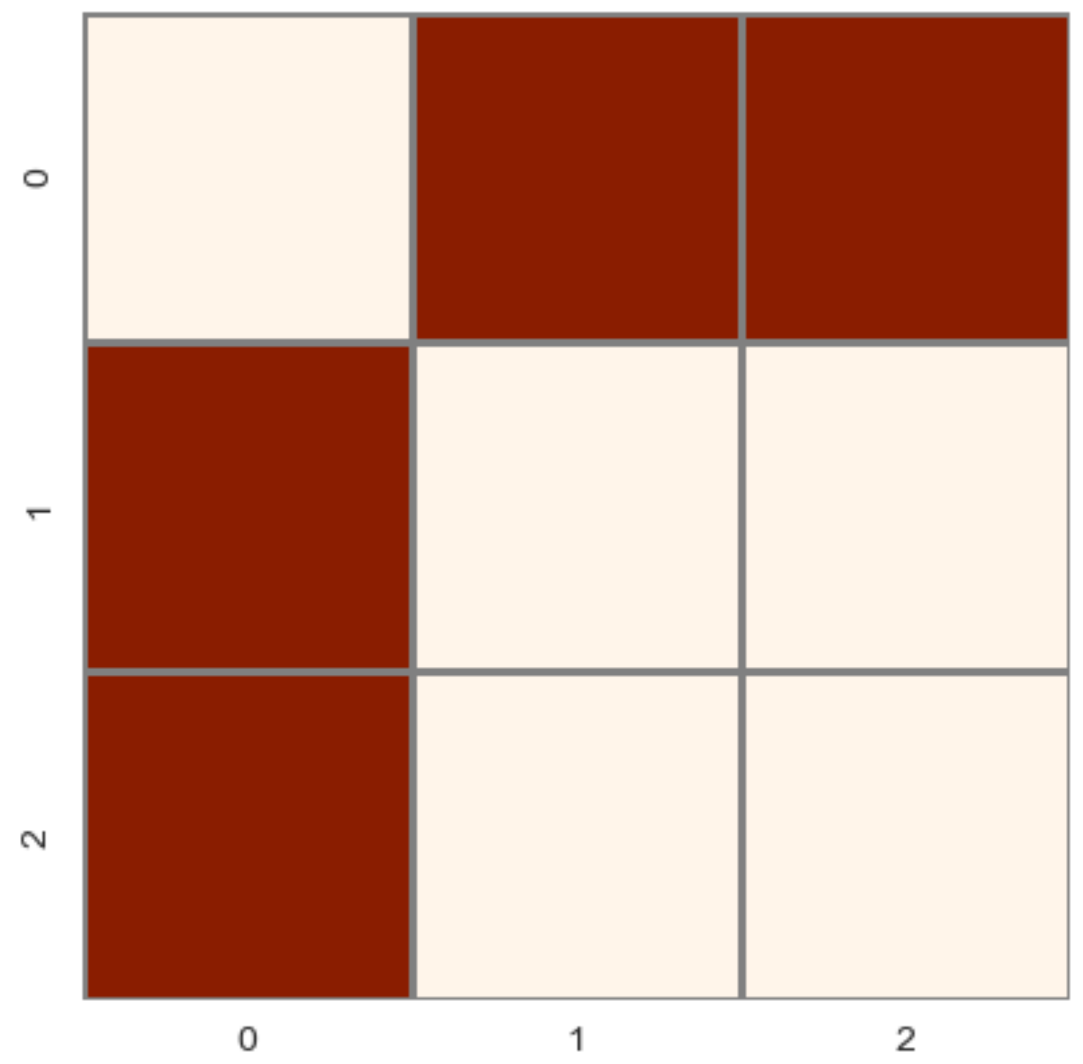
n = 12



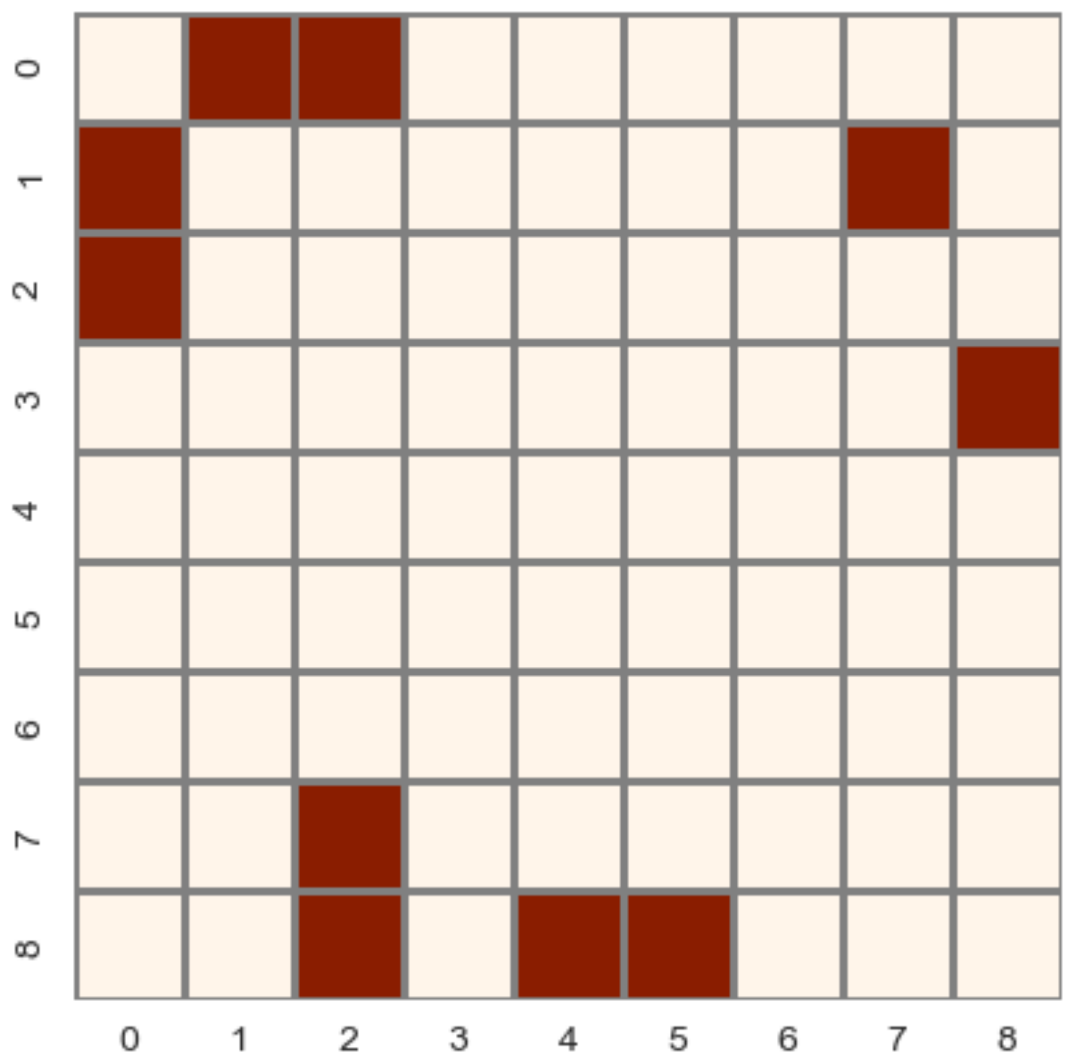
n = 18

Results (so far)

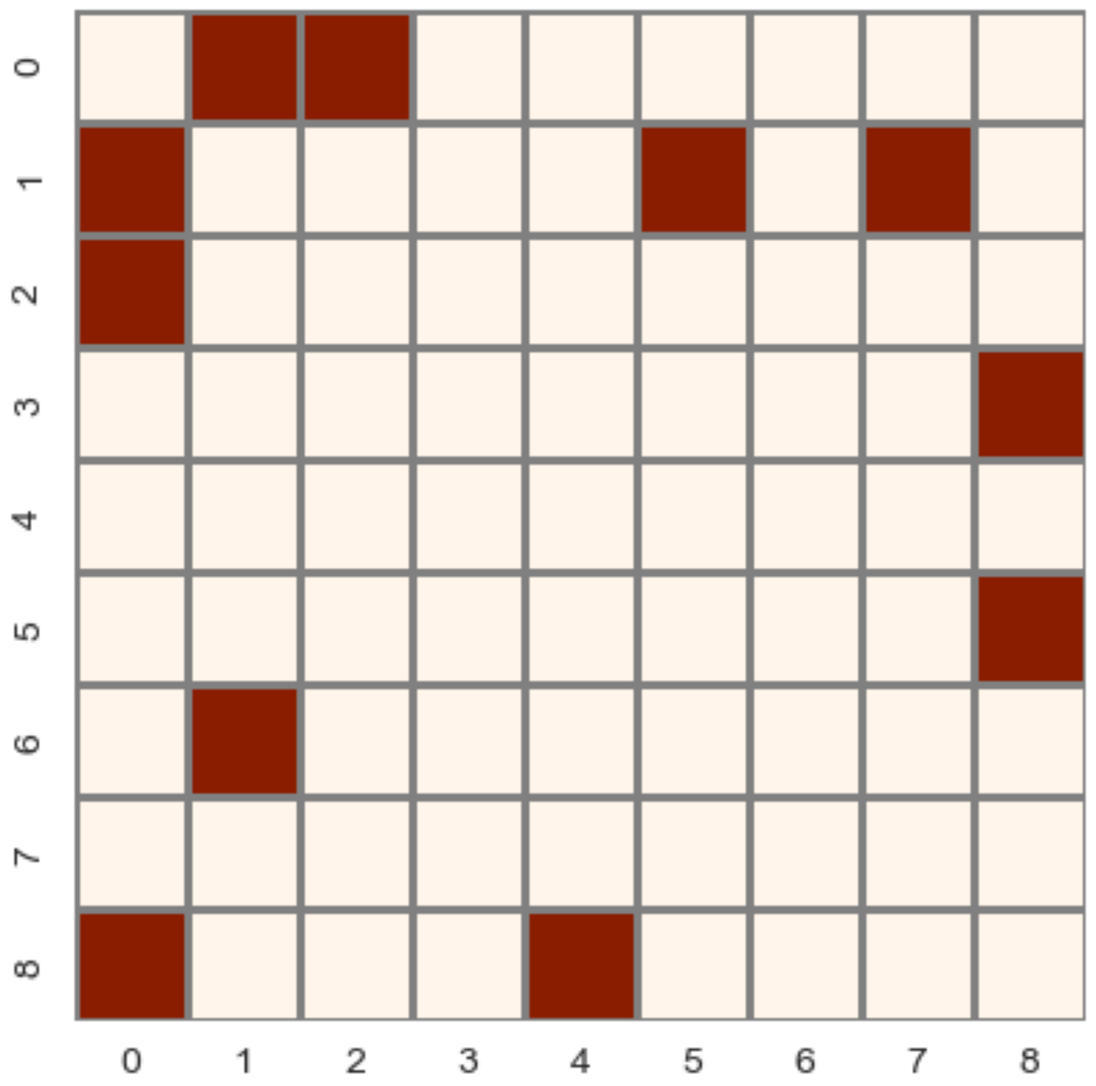
Saw an increased number of repeated patterns for very small n



n = 3



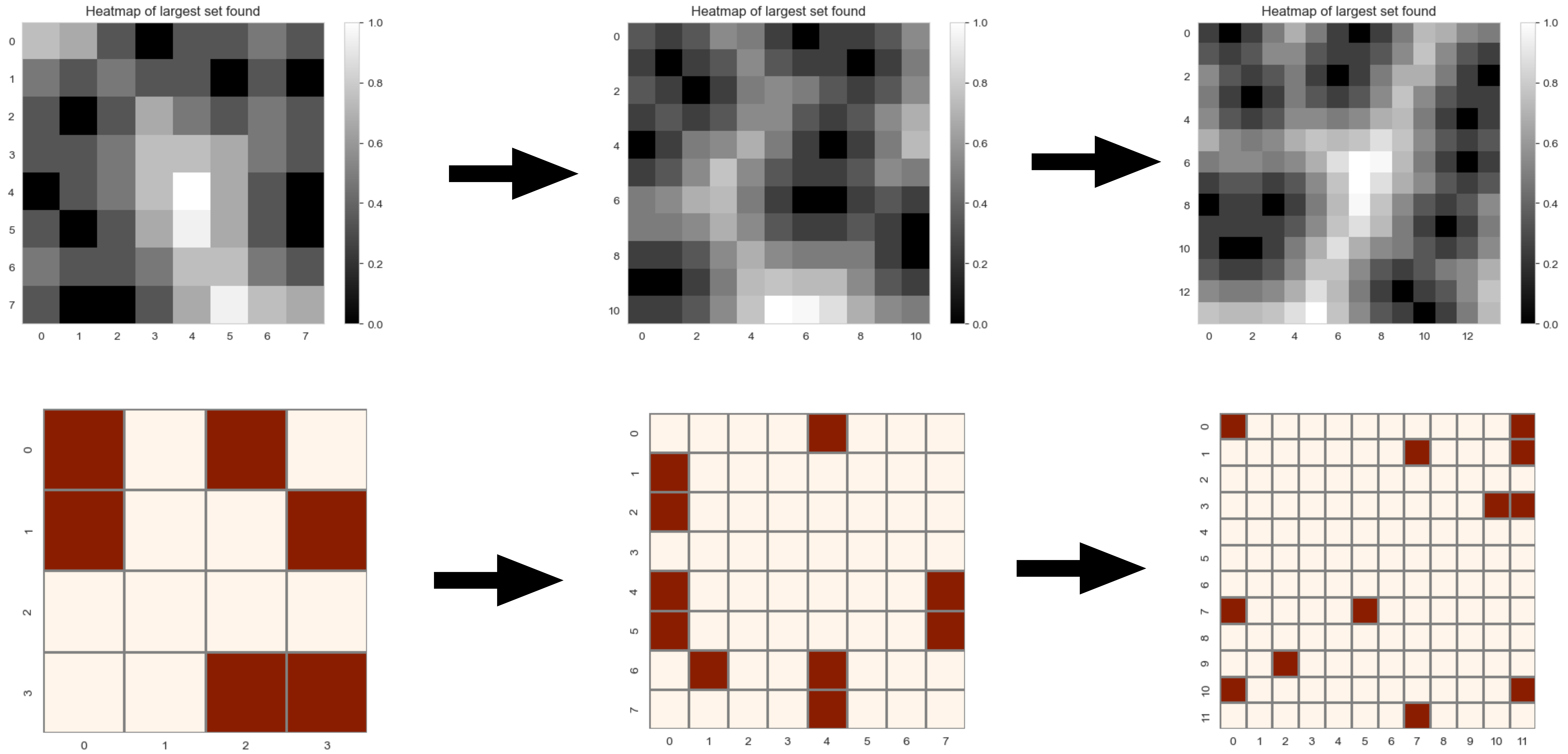
n = 9



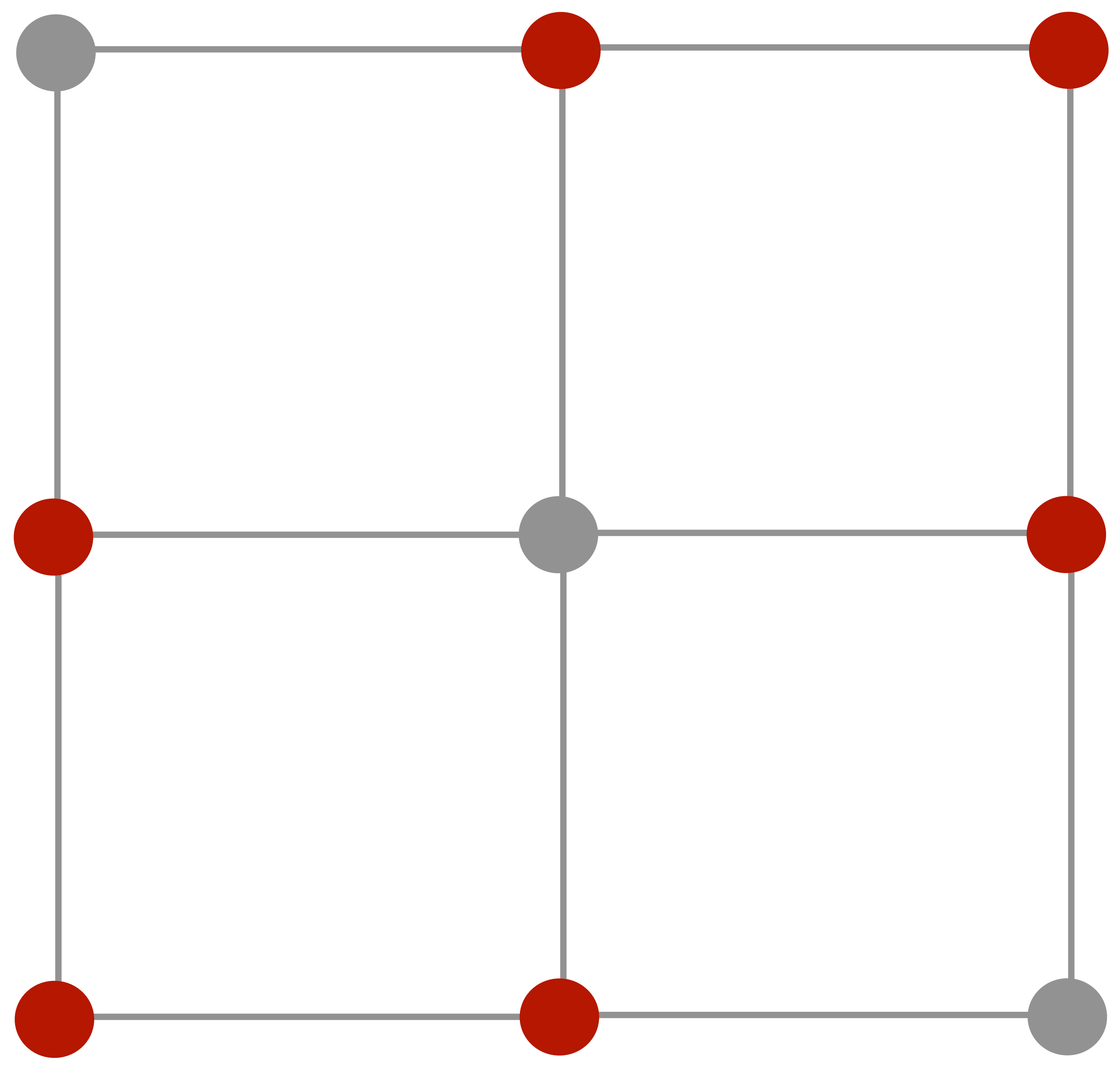
n = 9

Results (so far)

But most of the time, there was no discernible pattern



Problem 2



Given an n x n finite integer lattice, find the largest subset with no 3 colinear points.

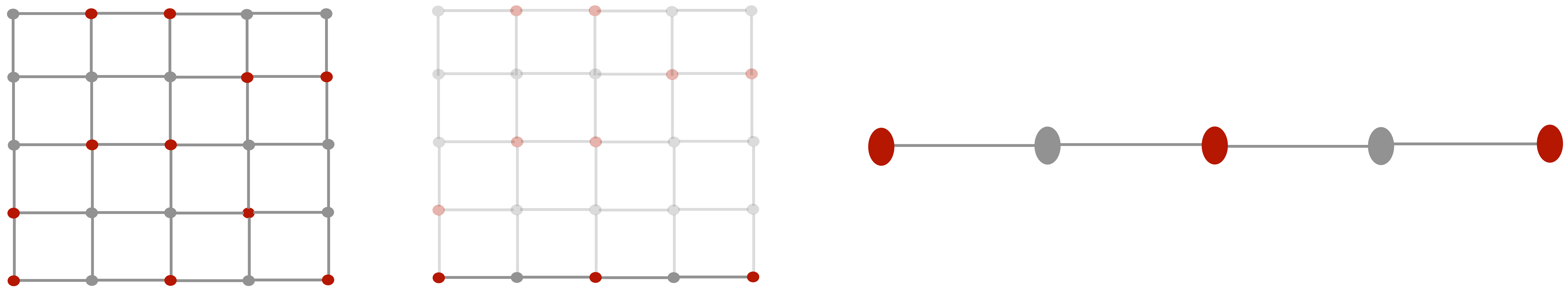
Problem 2

What do we know?

<u>Upper Bound</u>

$$|\text{Largest Set}| \leq 2n$$

See this with pigeonhole.



Problem 2

What do we know?

Lower Bound

$$o(\frac{3}{2}n) < |\text{Largest Set}|$$

Current Bound

$$o(\frac{3}{2}n) < |\text{Largest Set}| < 2n$$

Problem 2

What do we know?

Upper Bound

$$|\text{Largest Set}| \leq 2n$$

Turns out this bound is tight for $n \leq 46$

Problem 2

Open questions:

Main Conjecture:

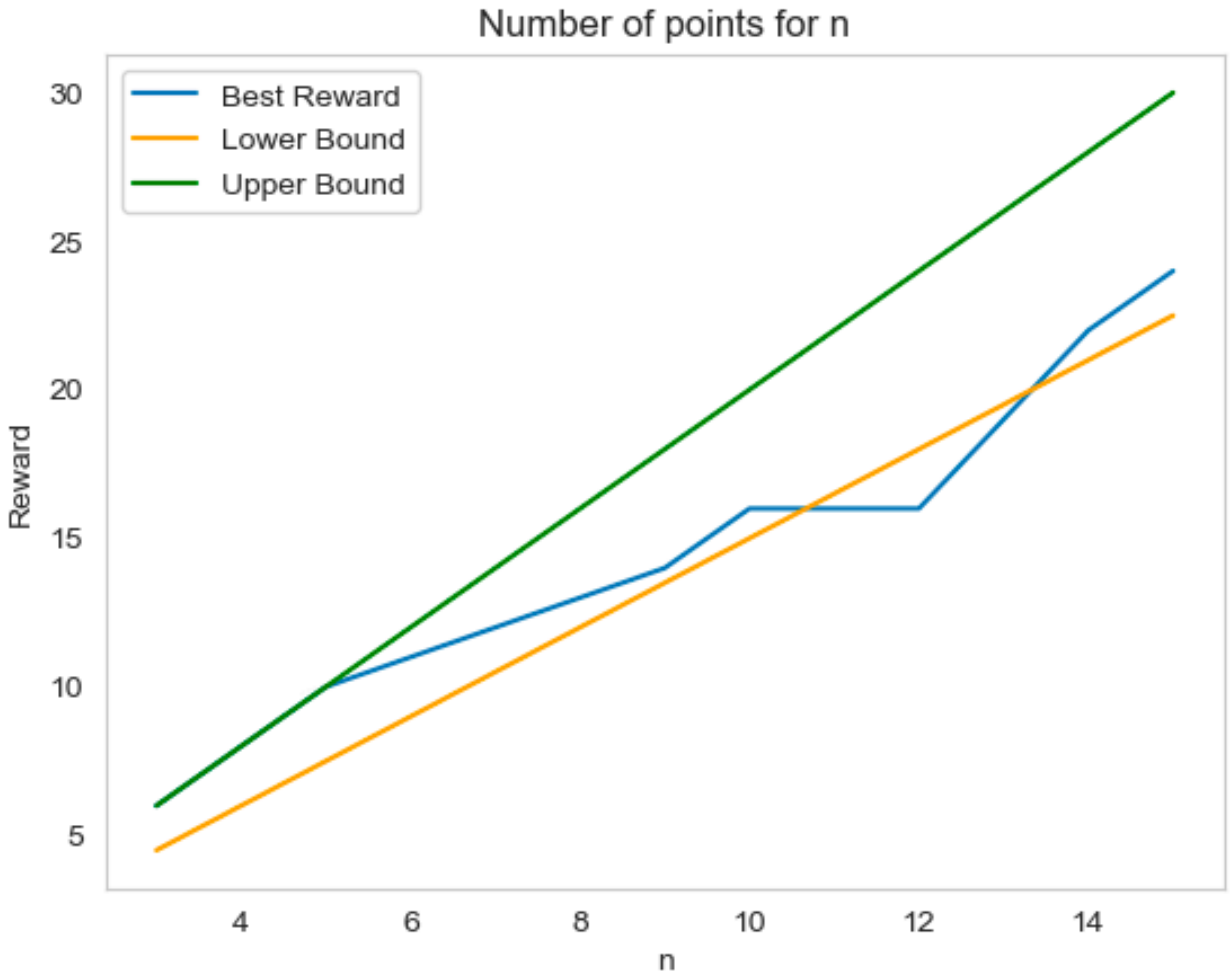
For $n > 46$, $|\text{largest set}| < 1.814n$

Other open questions in a minute!

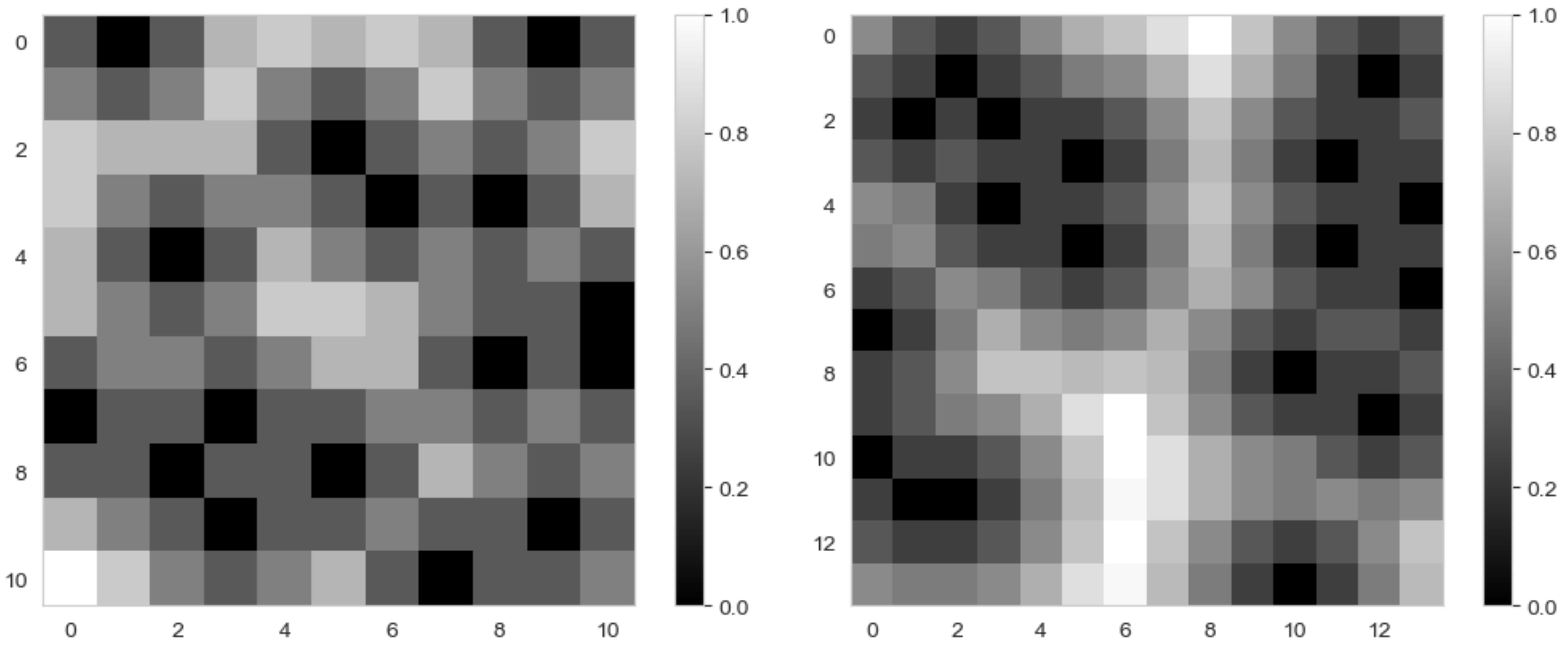
Results so far

<u>Results from base RL model</u>
3 hidden layers
(128, 64, 4)
Relu activation
Softmax output

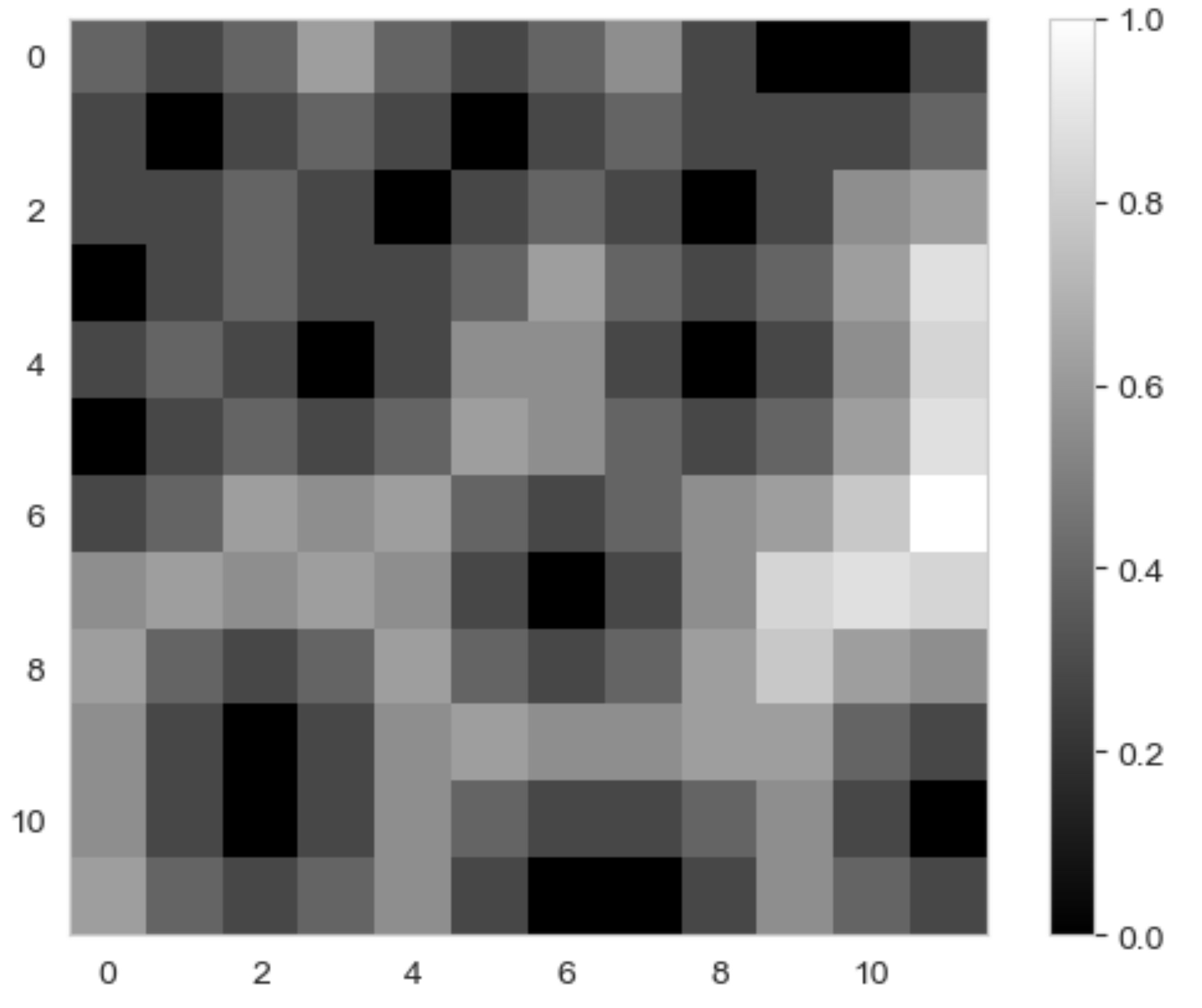
Rewarding more points with no 3 in a line



Results so far



Good generations are clustered away from the origin

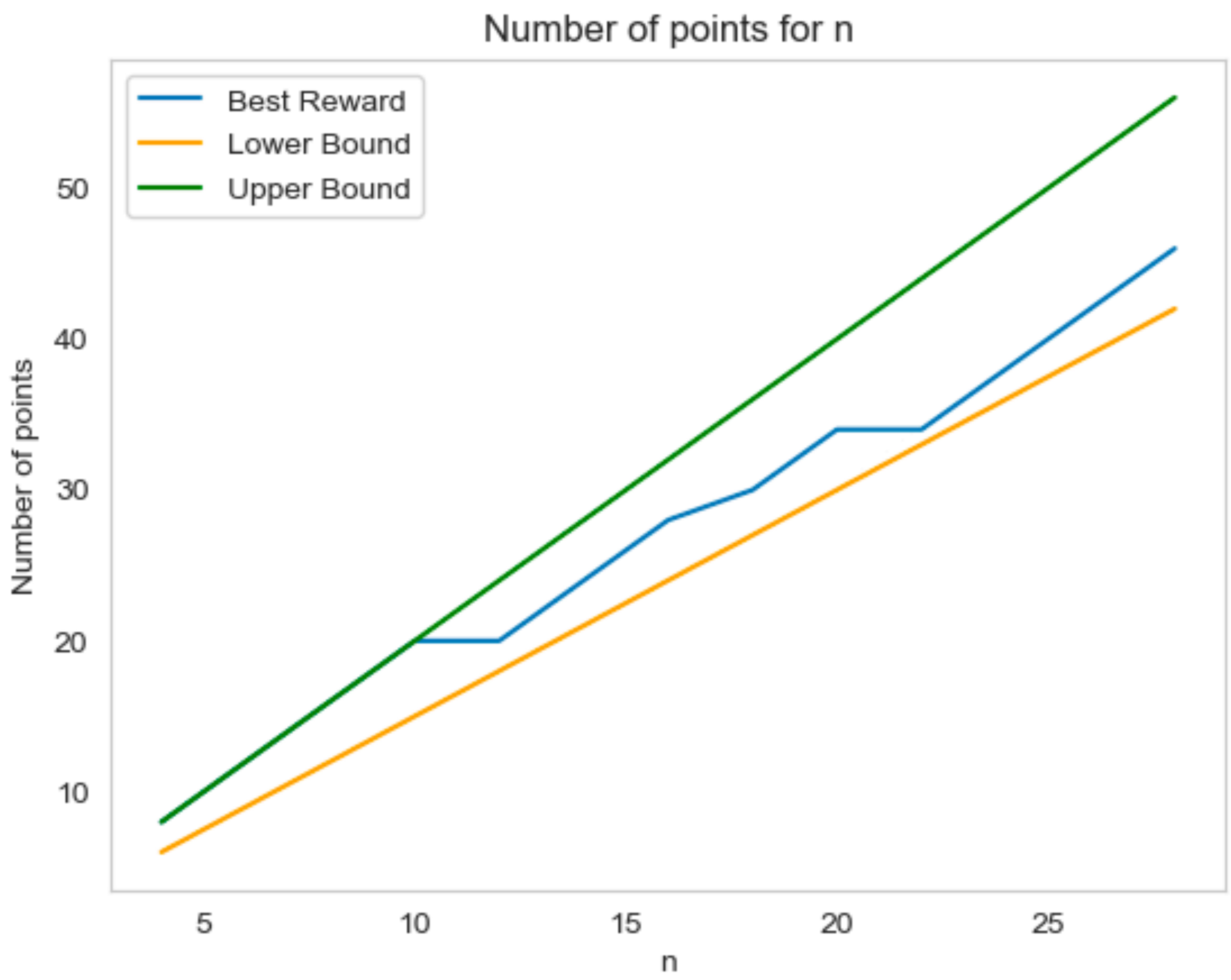


Bad ones have points near it

Results so far

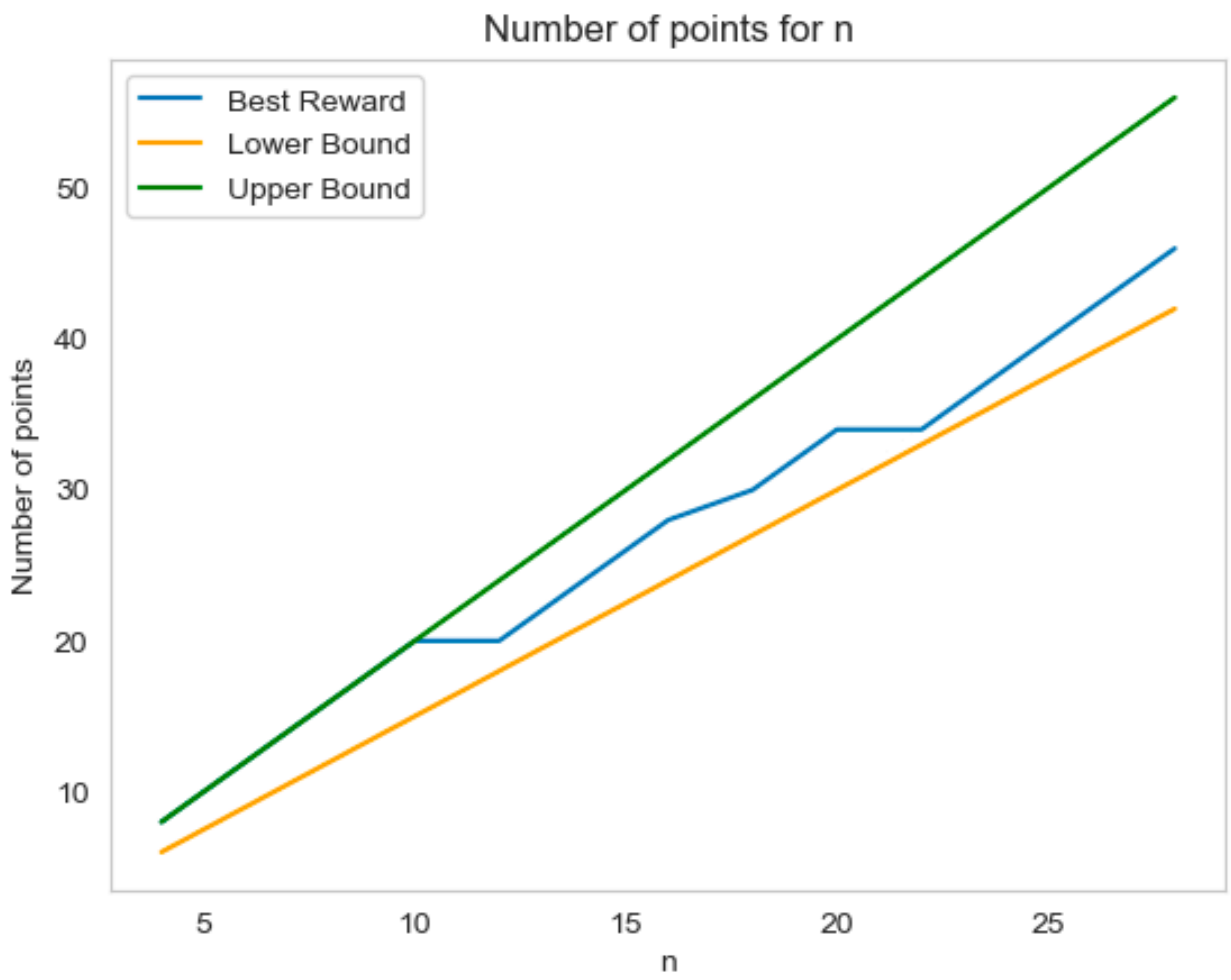
Results from base RL model
3 hidden layers
(128, 64, 4)
Relu activation
Softmax output

Rewarding more points with no 3 in a line with symmetry



Results so far

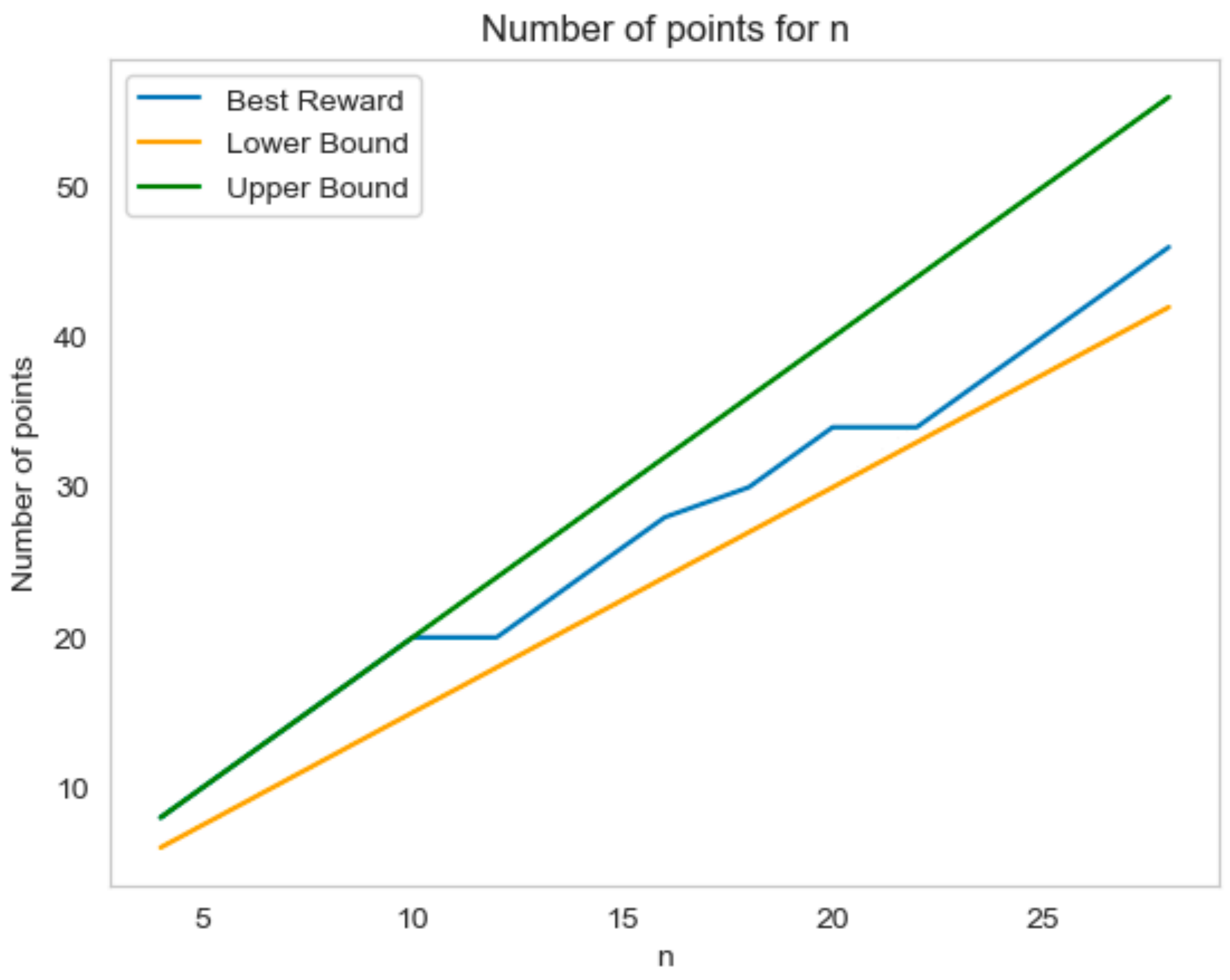
It turns out this is great!



Results so far

It turns out this is great!

- Finding optimal solutions till $n = 10$.

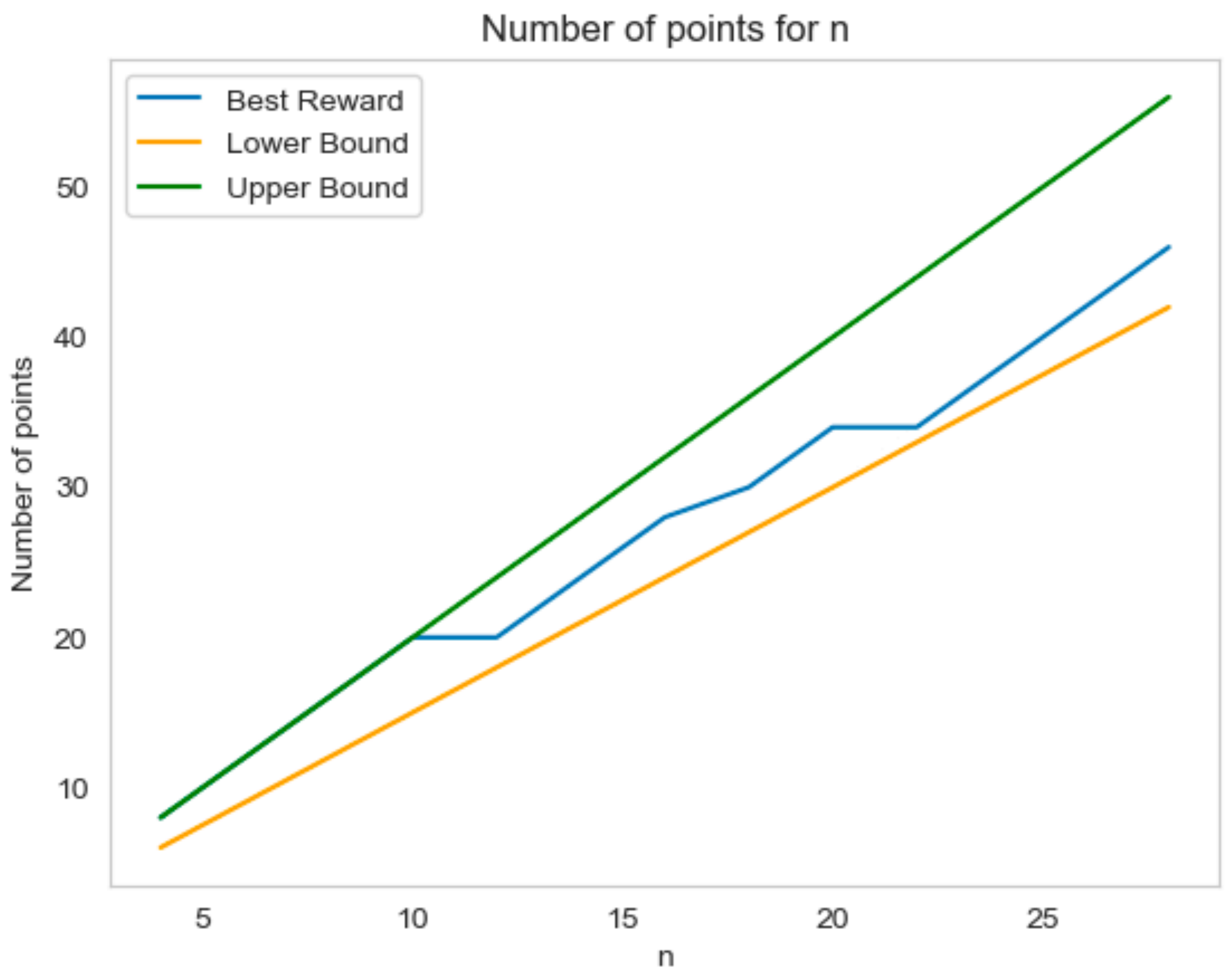


Results so far

It turns out this is great!

- Finding optimal solutions till $n = 10$.

- There are no $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ symmetric solutions for $n > 10$. So the results from $n = 10$ to 24 are the optimal symmetric solutions ($2n - 4$ points)



Further connections

Connections to what we know:

Further connections

Connections to what we know:

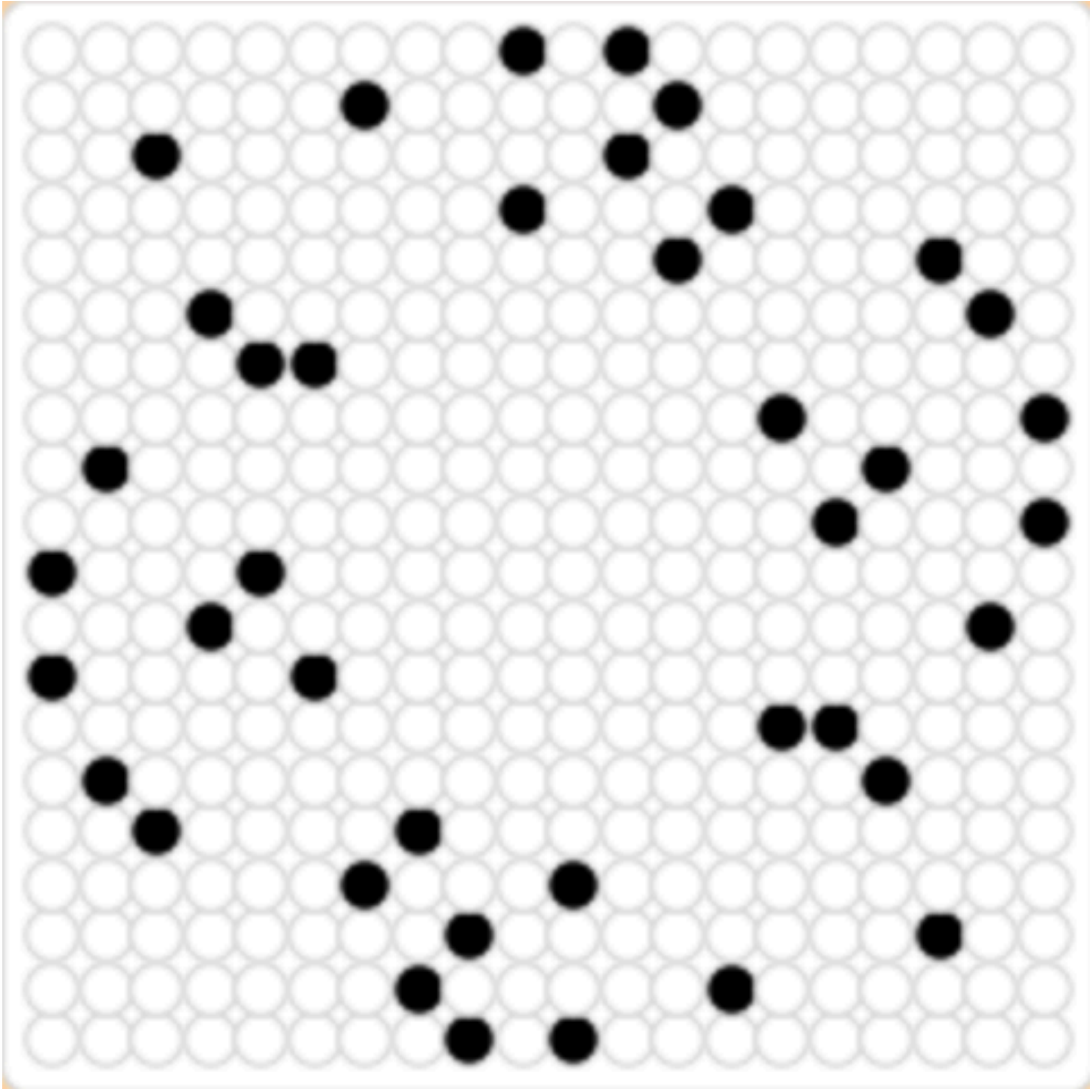
- No symmetric solutions known for more than 10x10 (known until 50x50 grids)

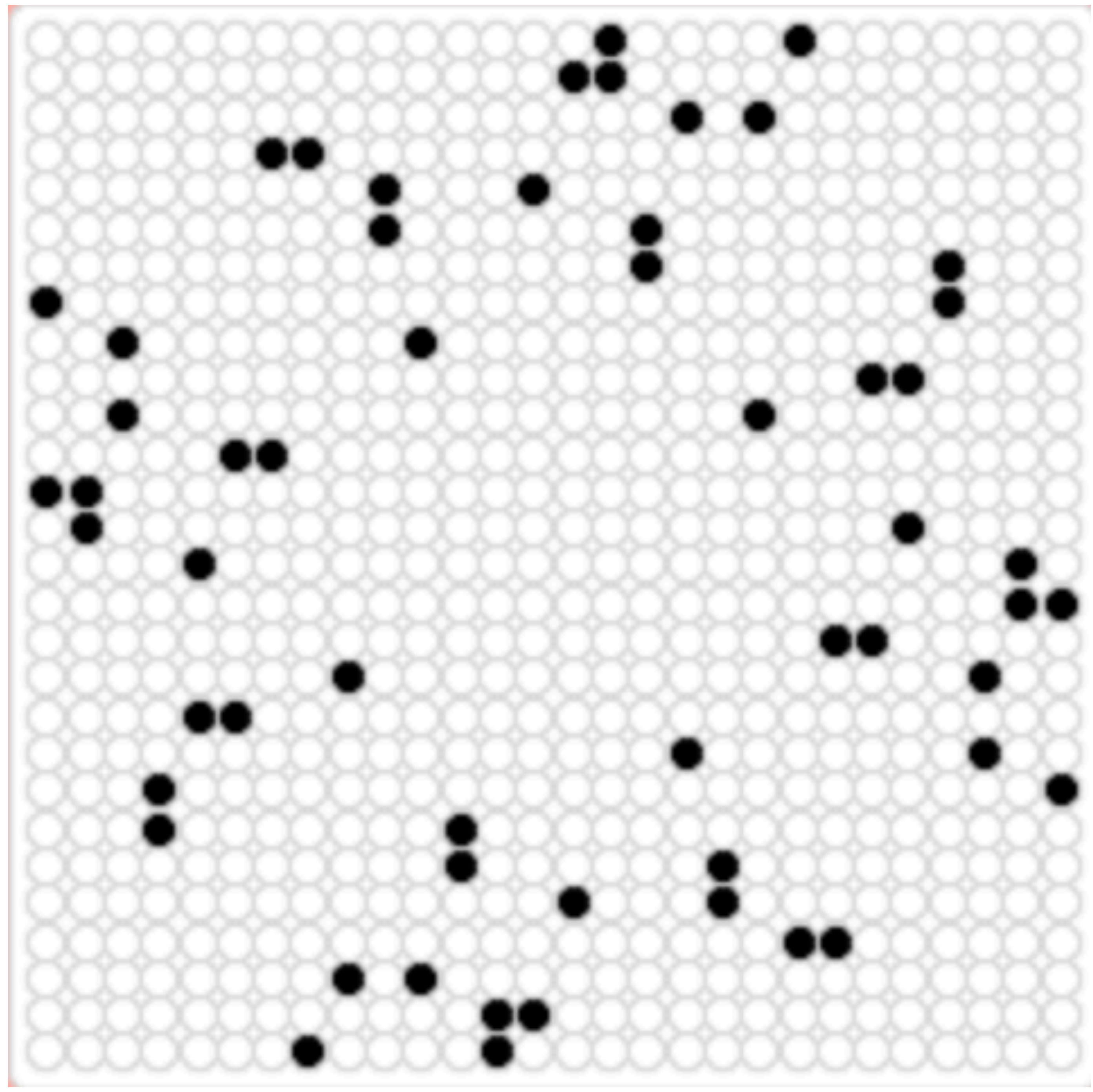
Further connections

Connections to what we know:

- No symmetric solutions known for more than 10x10 (known until 50x50 grids)

- Optimal constructions are very close to being symmetric or just have different symmetries.





Further connections

- Since our model learns symmetric solutions really well, it can probably learn these (currently testing)

Further connections

- Since our model learns symmetric solutions really well, it can probably learn these (currently testing)

- A lot of open conjectures about this problem have to do with symmetries of solutions. Hopefully we can study them with this.

Further connections

- Since our model learns symmetric solutions really well, it can probably learn these (currently testing)

- A lot of open conjectures about this problem have to do with symmetries of solutions. Hopefully we can study them with this.

Some open conjectures:

1. Are there any solutions bigger than 10x10 with full symmetry?

Further connections

- Since our model learns symmetric solutions really well, it can probably learn these (currently testing)

- A lot of open conjectures about this problem have to do with symmetries of solutions. Hopefully we can study them with this.

Some open conjectures:

1. Are there any solutions bigger than 10x10 with full symmetry?

2. Is every solution that has vertical and horizontal lines of symmetry fully symmetric including rotationally symmetric?

Further connections

- Since our model learns symmetric solutions really well, it can probably learn these (currently testing)

- A lot of open conjectures about this problem have to do with symmetries of solutions. Hopefully we can study them with this.

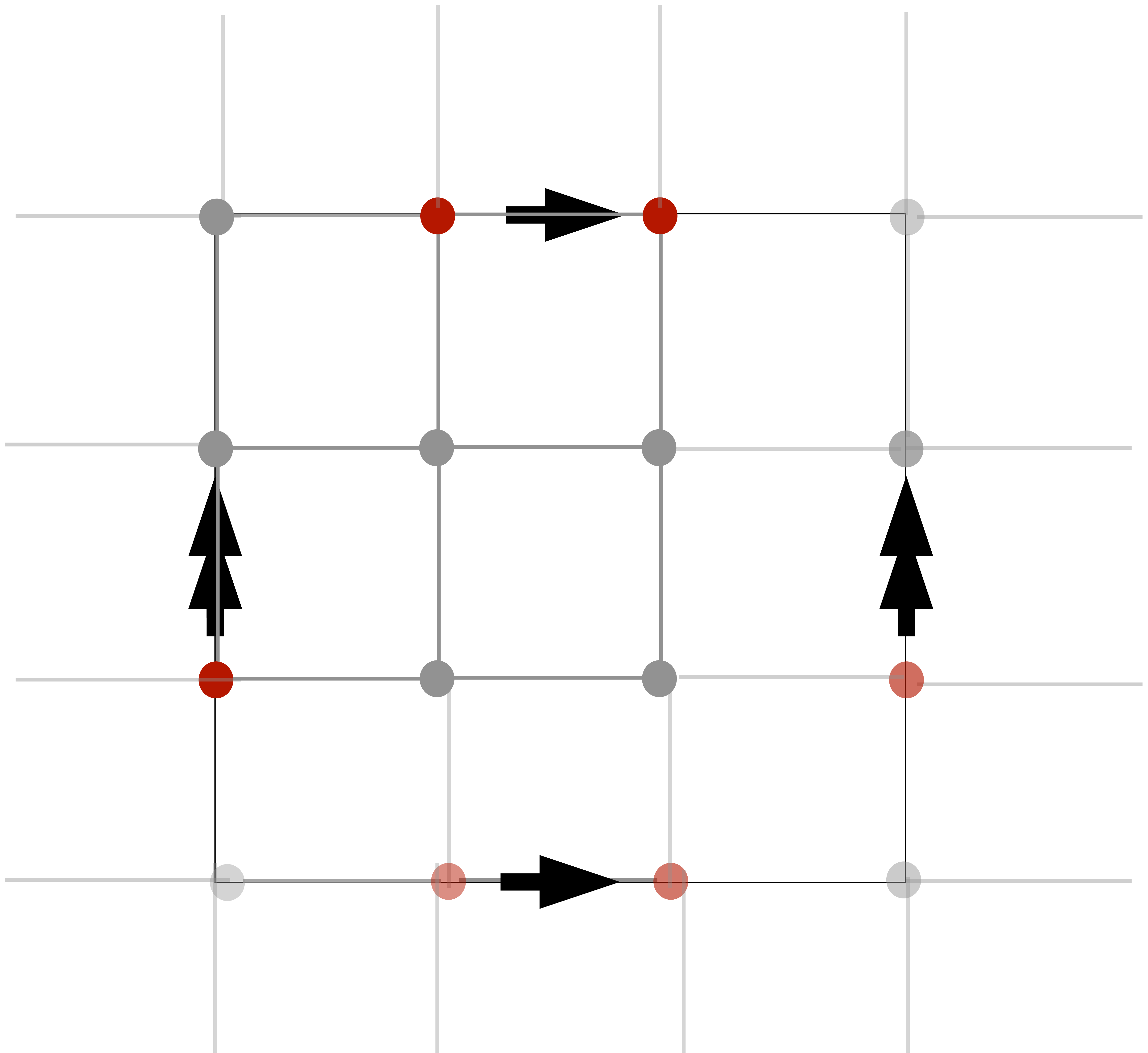
Some open conjectures:

1. Are there any solutions bigger than 10x10 with full symmetry?

2. Is every solution that has vertical and horizontal lines of symmetry fully symmetric including rotationally symmetric?

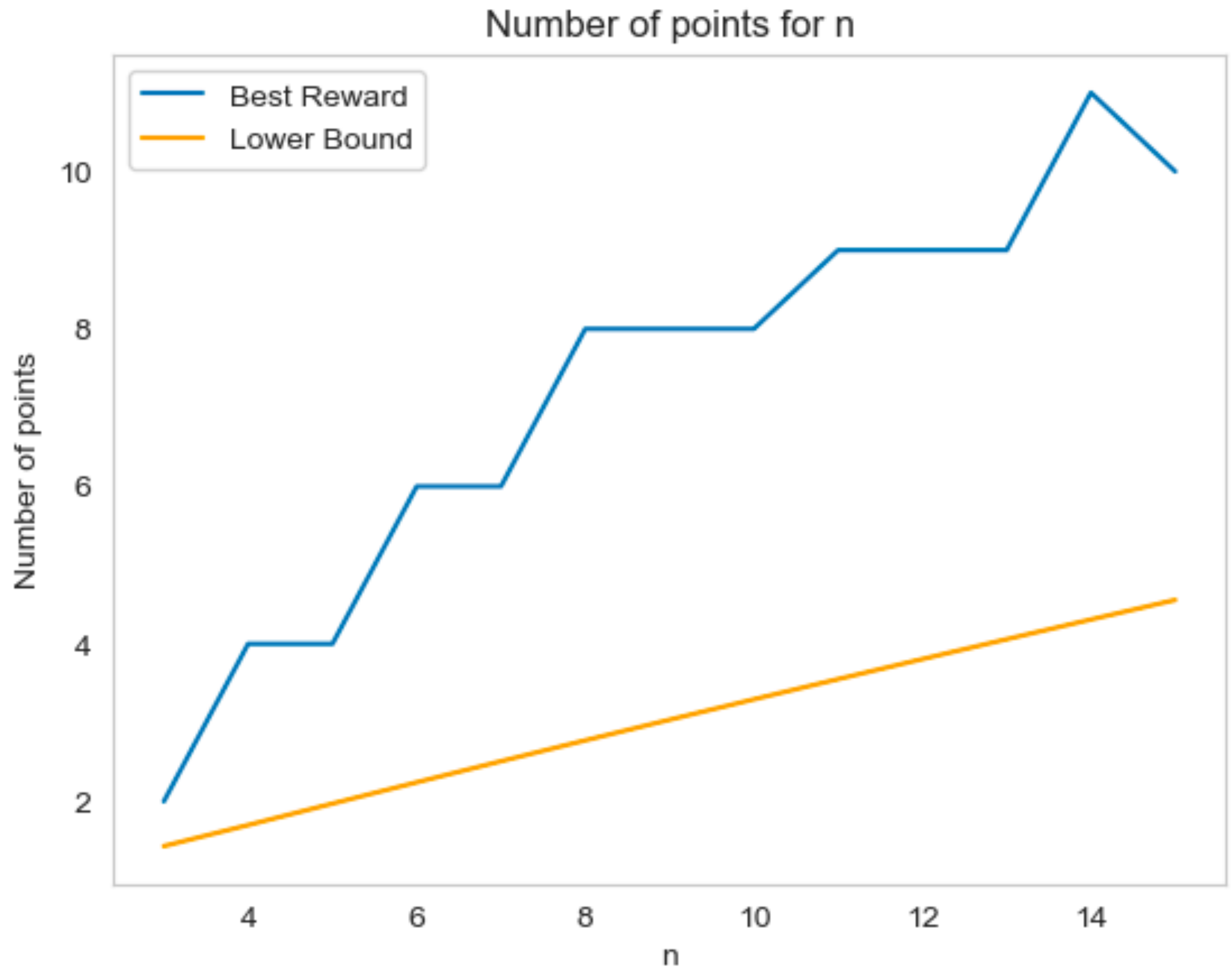
3. Are there any solutions that have no symmetries for a >18x18 board?

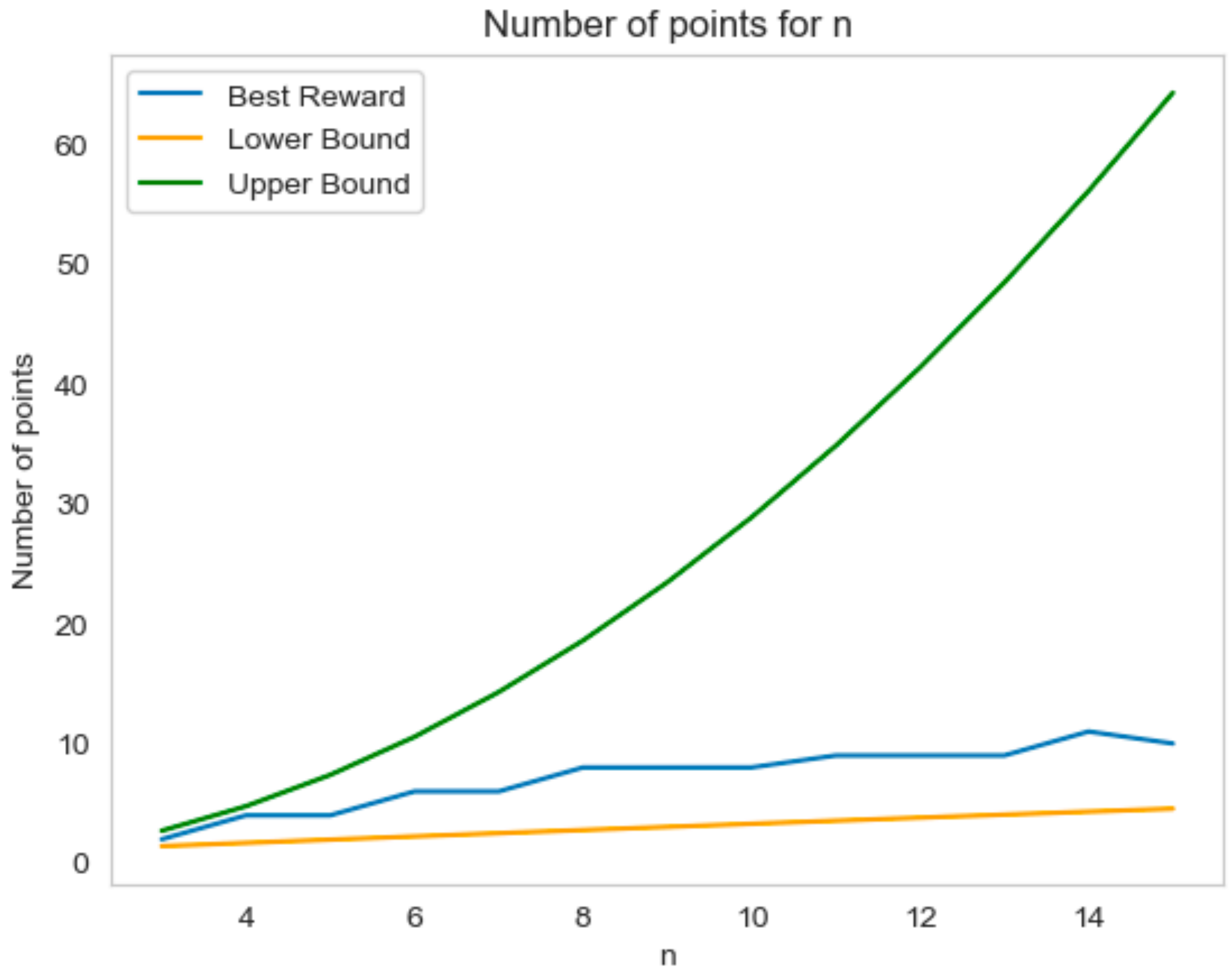
Problem 3



Given an n x n finite integer lattice, what's the size of the largest subset such that no three points form an isosceles triangle?

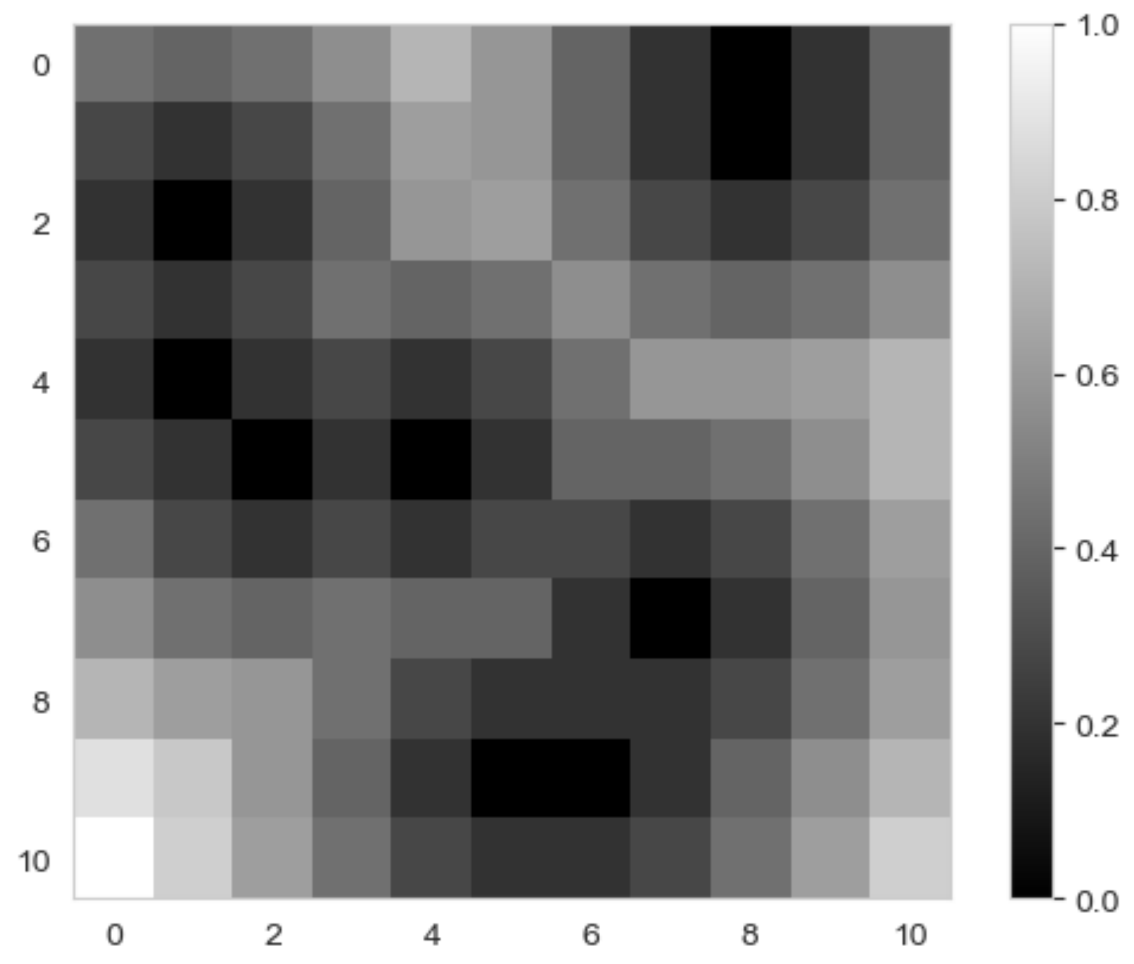
Results so far

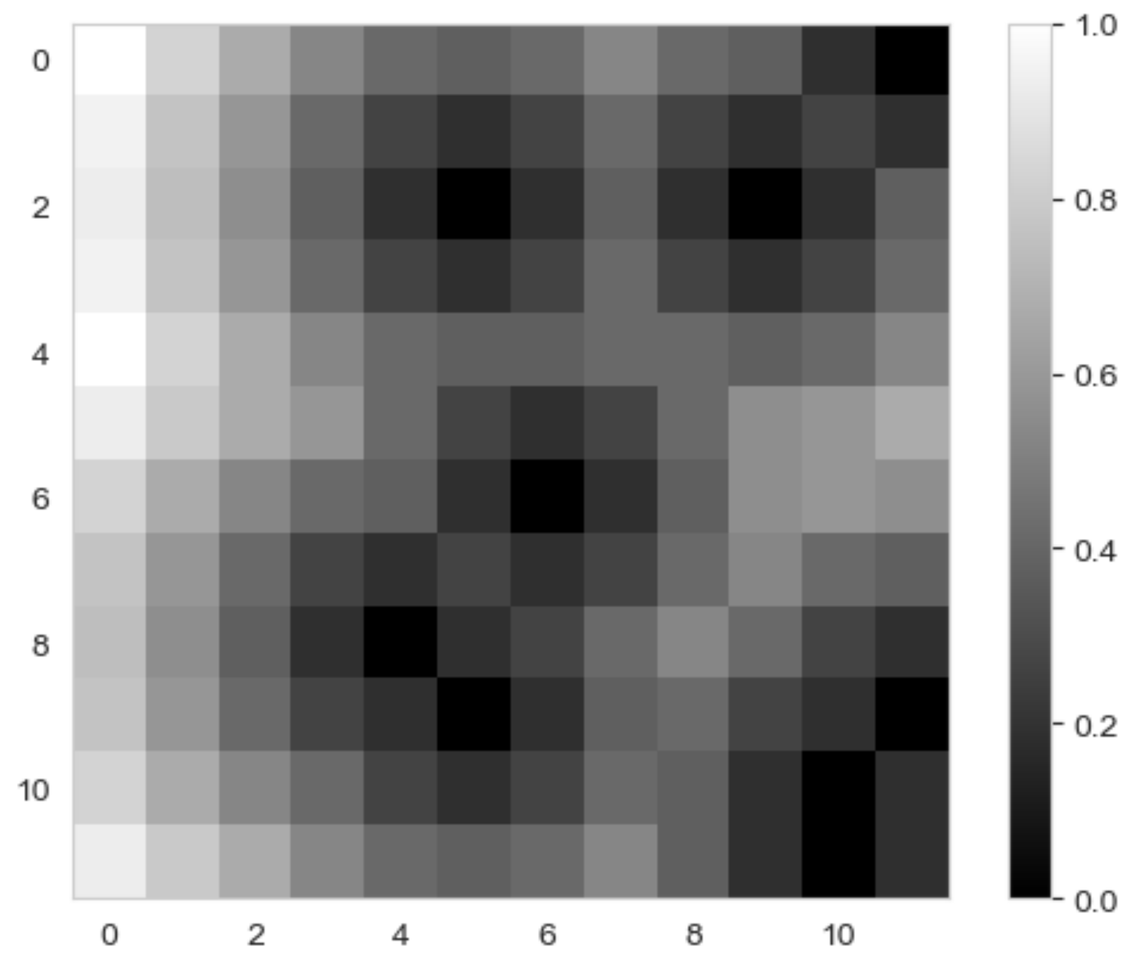




Results so far

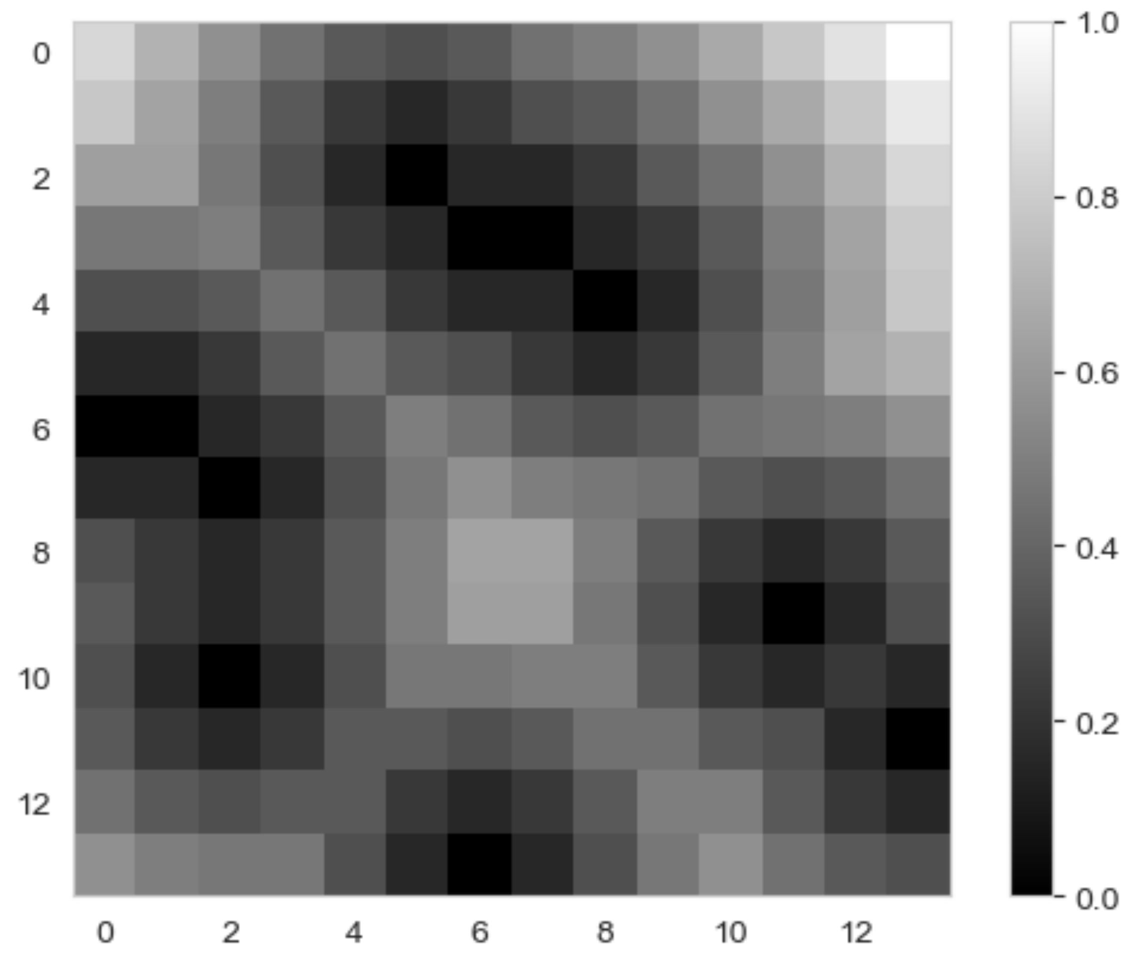
n=10

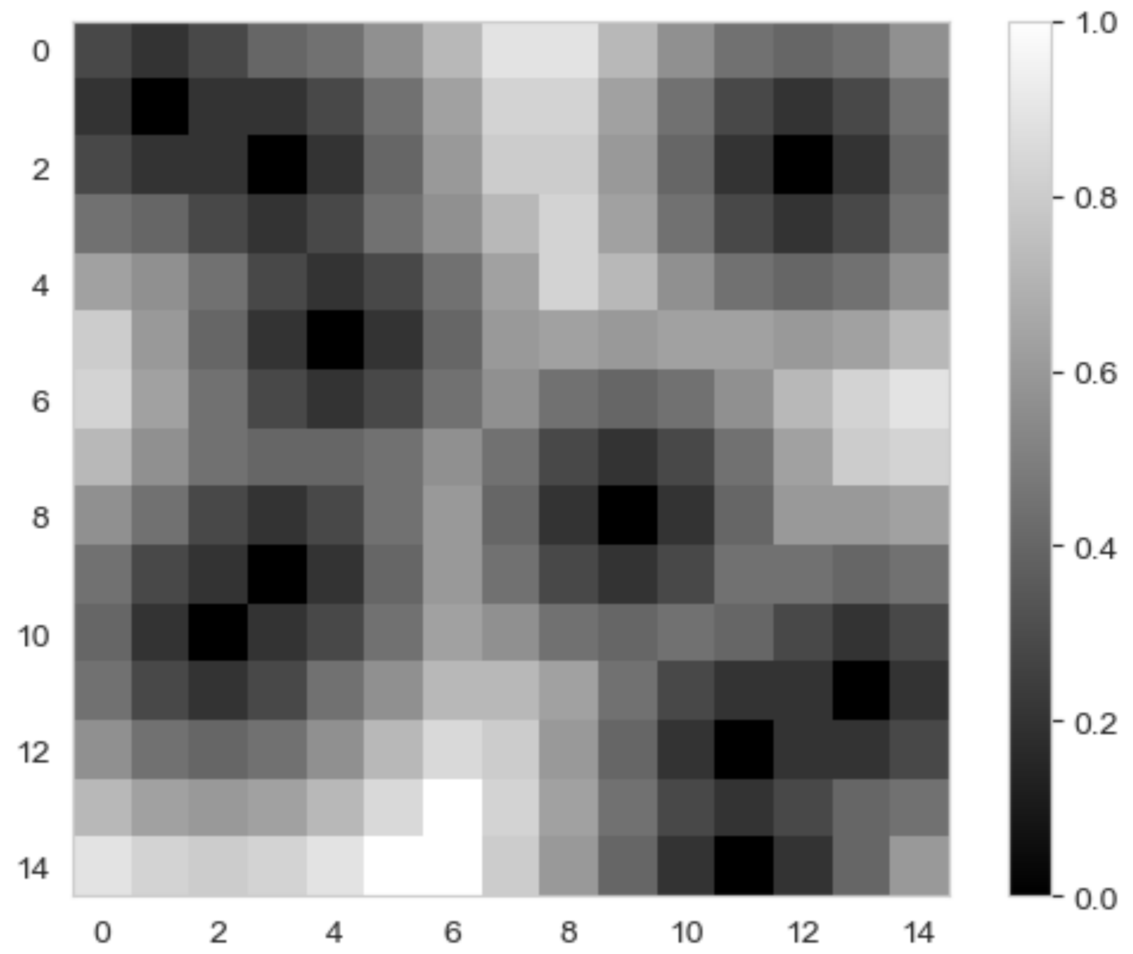




n=11

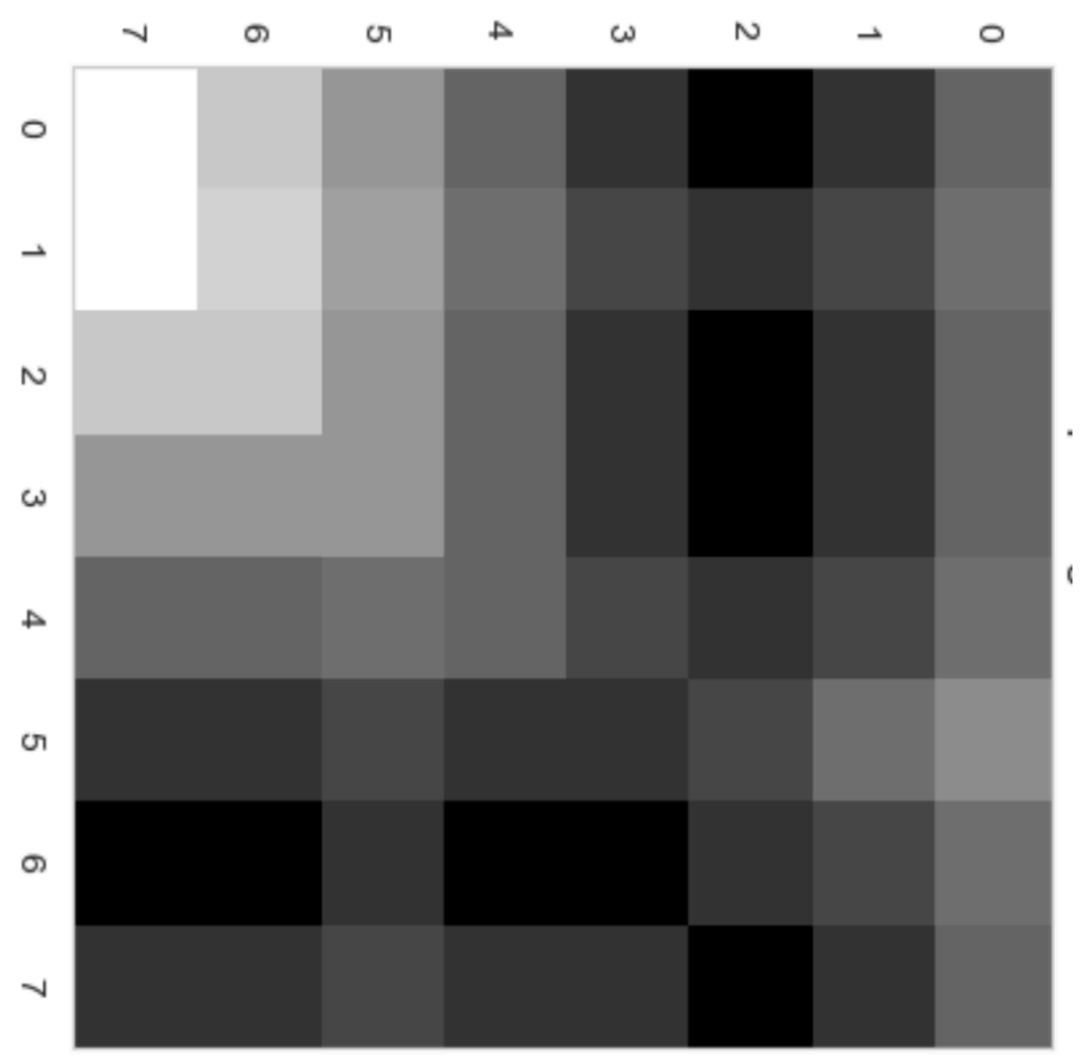
n=13



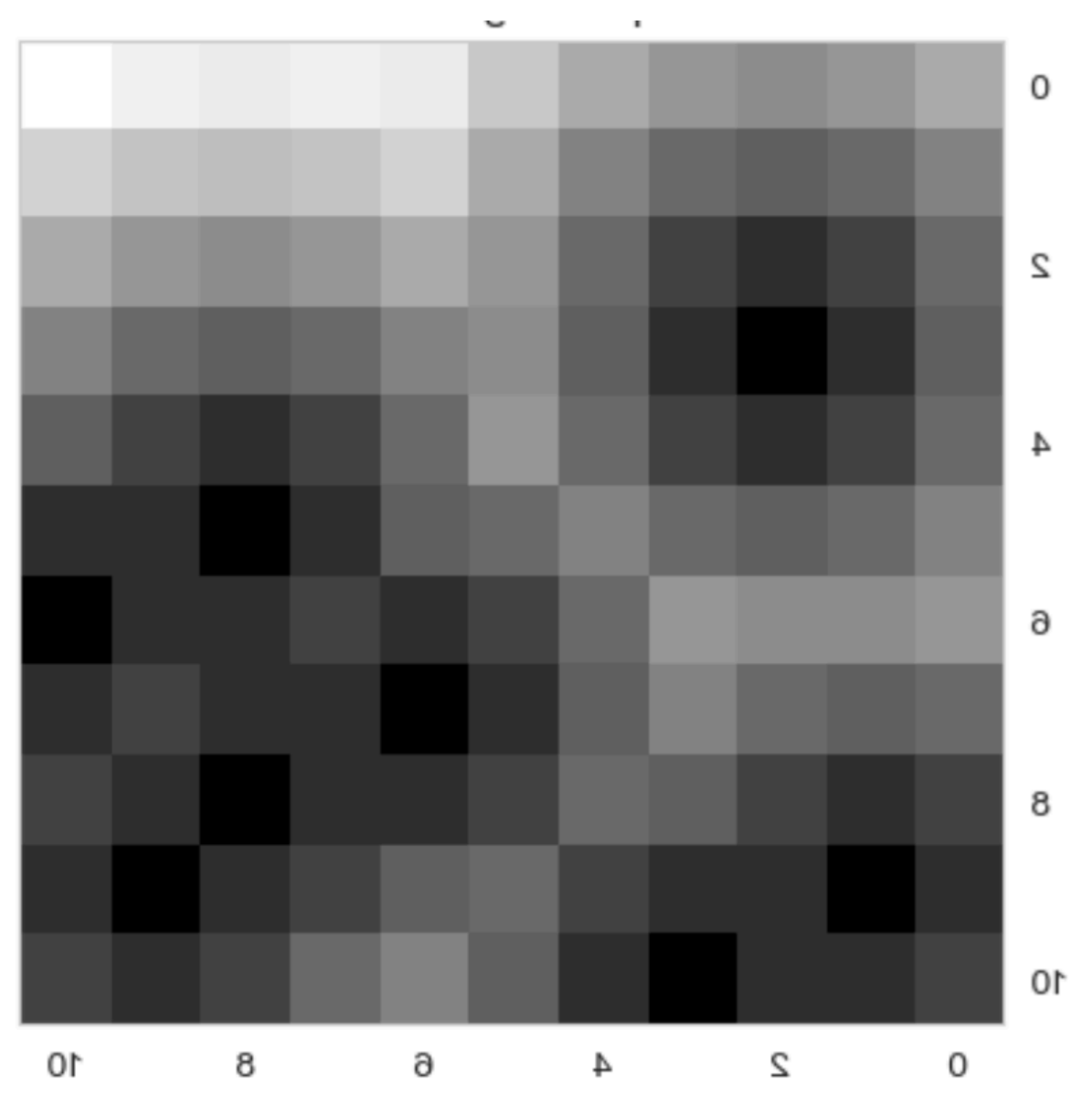


n=14

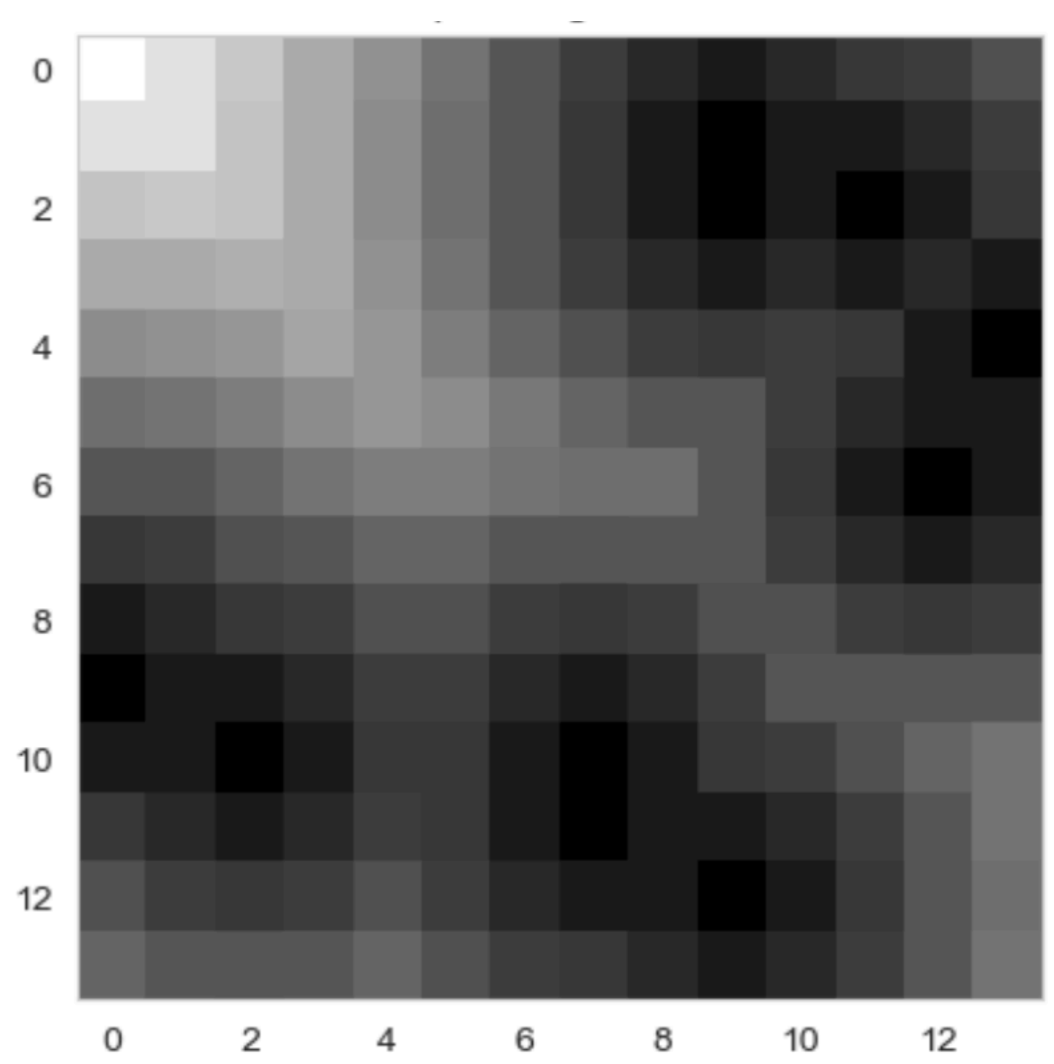
Results so far



n=8



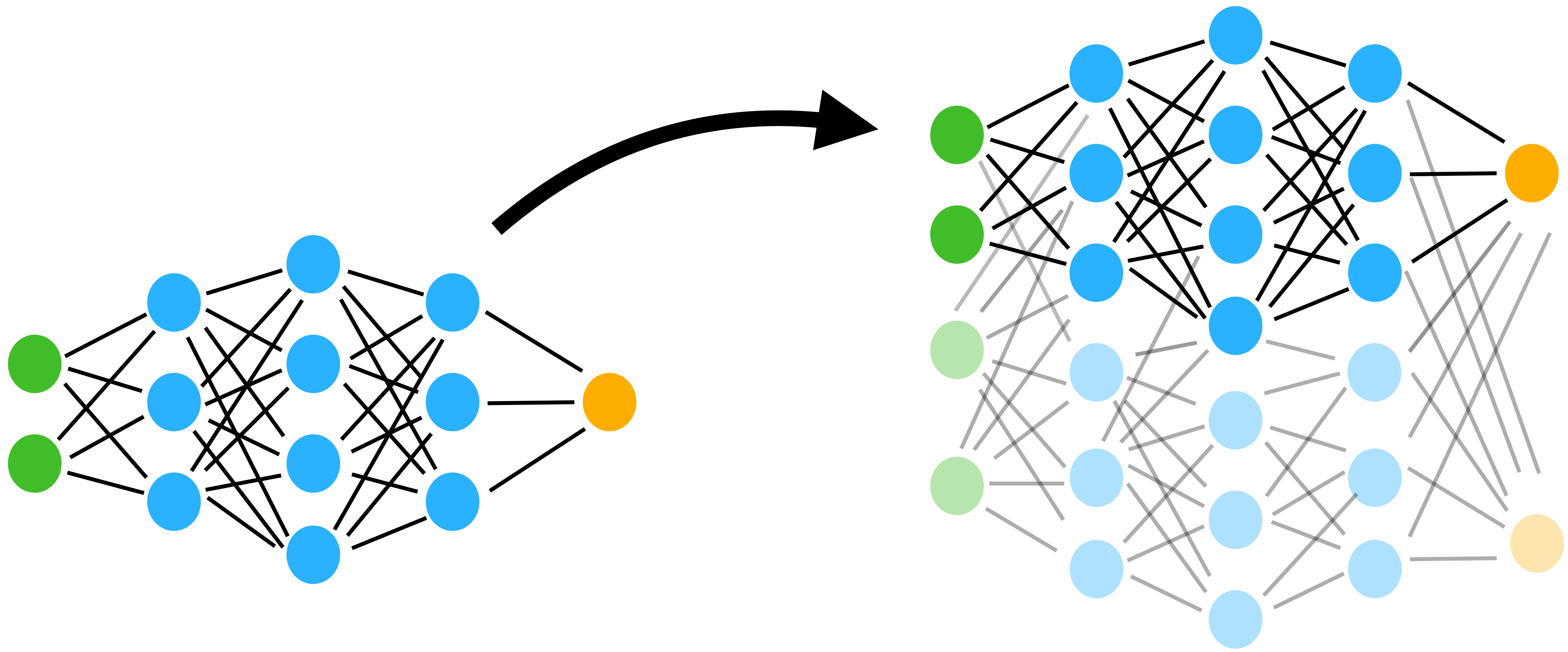
n=11

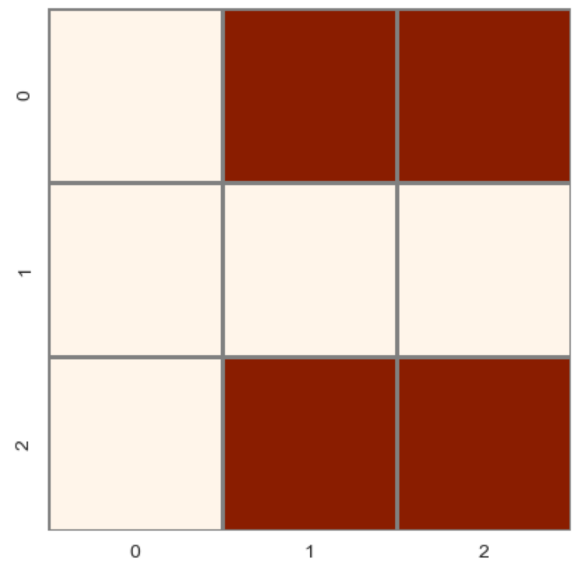


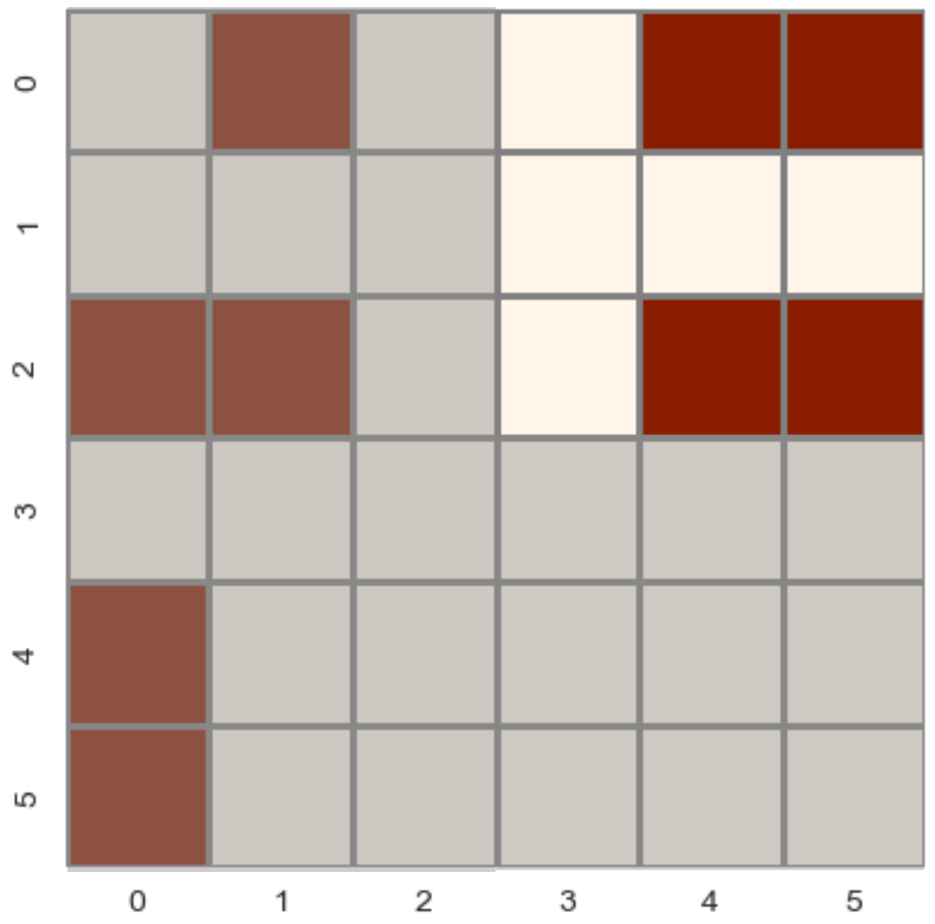
n=14

But still not generally true

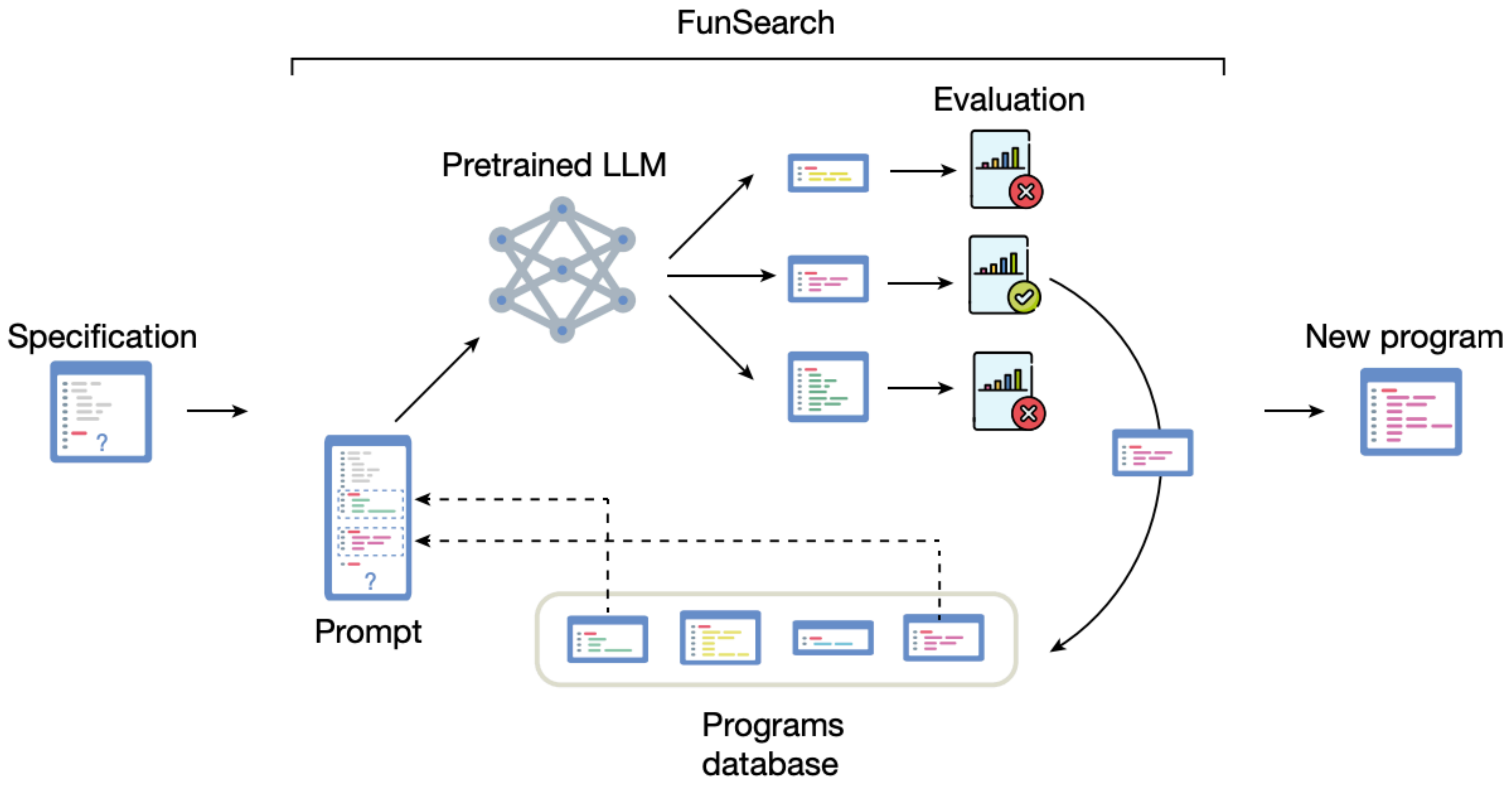
Future Directions







Future Directions



FunSearch

Uses a large language model instead of a classical neural network

Searches space of generating programs instead of examples

Potentially a way to get more interpretable examples

Future Directions

Discovered function that builds the best known independent sets in C_9^n for $n = 3, \ldots, 7$

These independent sets match the best known construction reported by Matthew & Östergård (2016).

```
def priority(el: tuple[int, ...], num_nodes: int, n: int) -> float:
  """Returns the priority with which we want to add `el` to the set."""
  s = 0.
  for i in range(n):
    s += el[i] << i
    s %= num_nodes
  return (2 * el[2] - 4 * el[0] + el[1]) % num_nodes + s
```

Future Directions

Linear Shannon Capacity of Cayley Graphs

Venkatesan Guruswami and Andrii Riazanov
Carnegie Mellon University
Computer Science Department
Pittsburgh, PA 15213
Email: {venkatg, riazanov}@cs.cmu.edu

***Abstract*—The Shannon capacity of a graph is a fundamental quantity in zero-error information theory measuring the rate of growth of independent sets in graph powers. Despite being well-studied, this quantity continues to hold several mysteries. Lovász famously proved that the Shannon capacity of C_5 (the 5-cycle) is at most $\sqrt{5}$ via his theta function. This bound is achieved by a simple linear code over $\mathbb{F}_5$ mapping $x \mapsto 2x$.**

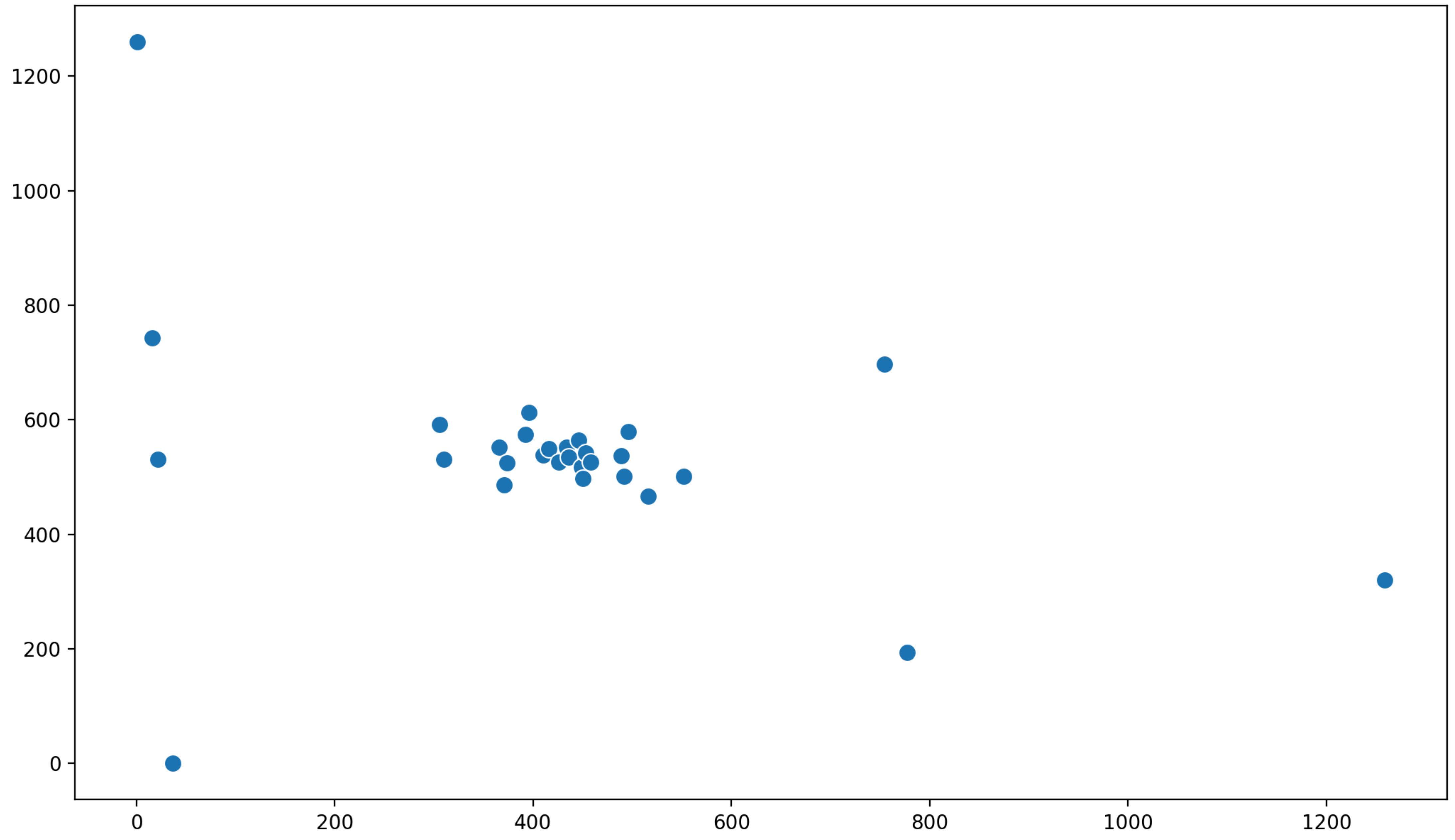
This motivates the notion of *linear Shannon capacity* of graphs, which is the largest rate achievable when restricting oneself to linear codes. We give a simple proof based on the polynomial method that the linear Shannon capacity of C_5 is $\sqrt{5}$. Our method applies more generally to Cayley graphs over the additive group of finite fields $\mathbb{F}_q$, giving an upper bound on the linear Shannon capacity. We compare this bound to the Lovász theta function, showing that they match for self-complementary Cayley graphs (such as C_5), and that the bound is smaller in some cases. We also exhibit a quadratic gap between linear and general Shannon capacity for some graphs.

C_5, the famous work of Lovász that introduced the theta function proved that the Shannon capacity equals $\sqrt{5}$ [2].

In coding theory, $\Theta(G)$ captures the zero-error capacity of the channel with confusion graph G. Specifically, consider a coding channel with input set $V = \{1, 2, \ldots, n\}$, and let the confusion graph G have V as the vertex set. Further, let $(v, u) \in E(G)$ if and only if the letters v and u might be confused in the transmission (i.e. lead to the same output). Clearly, $\alpha(G)$ captures the maximum size of a set of letters that can be communicated in an error-free manner in a single use of the channel. From the definition of the graph power, it follows that $\alpha(G^k)$ represents the largest set of k-letter words (code) that can be communicated in an error-free manner over k uses of the channel. Therefore, $\Theta(G)$ can be interpreted as the maximal effective number of symbols that can be transmitted per use of the channel, amortized over k uses of the channel in the limit of large k.

Example Problems

Can we upper bound the number of points in the real plane
So that no empty convex-6-gons exist?



Convex Geometry

Karan Srivastava
ksrivastava4@wisc.edu