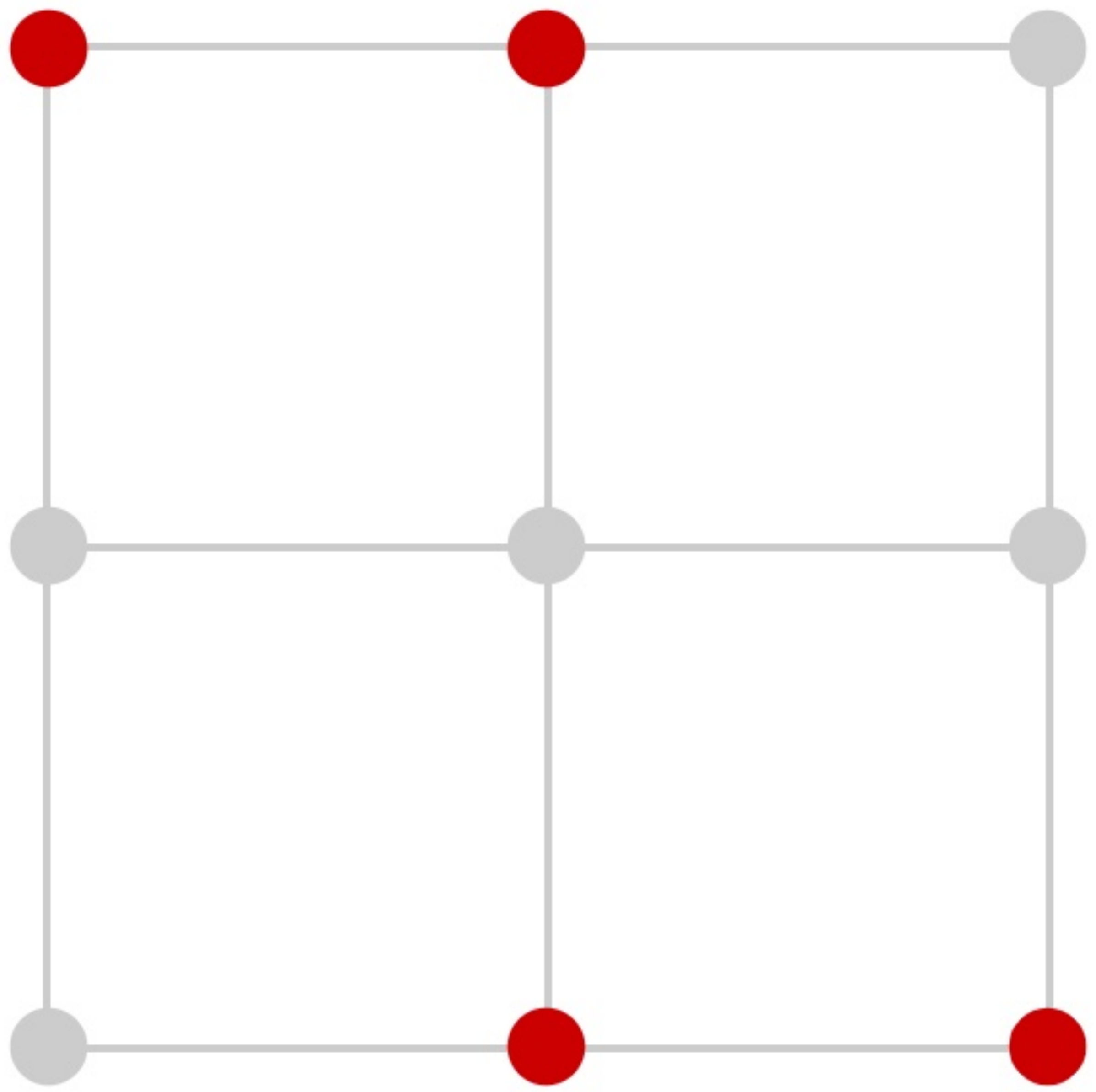
Reinforcement Learning for generating large isosceles-free subsets of an integer lattice

Karan Srivastava | Specialty Exam

Research supported in part by NSF Award DMS-2023239

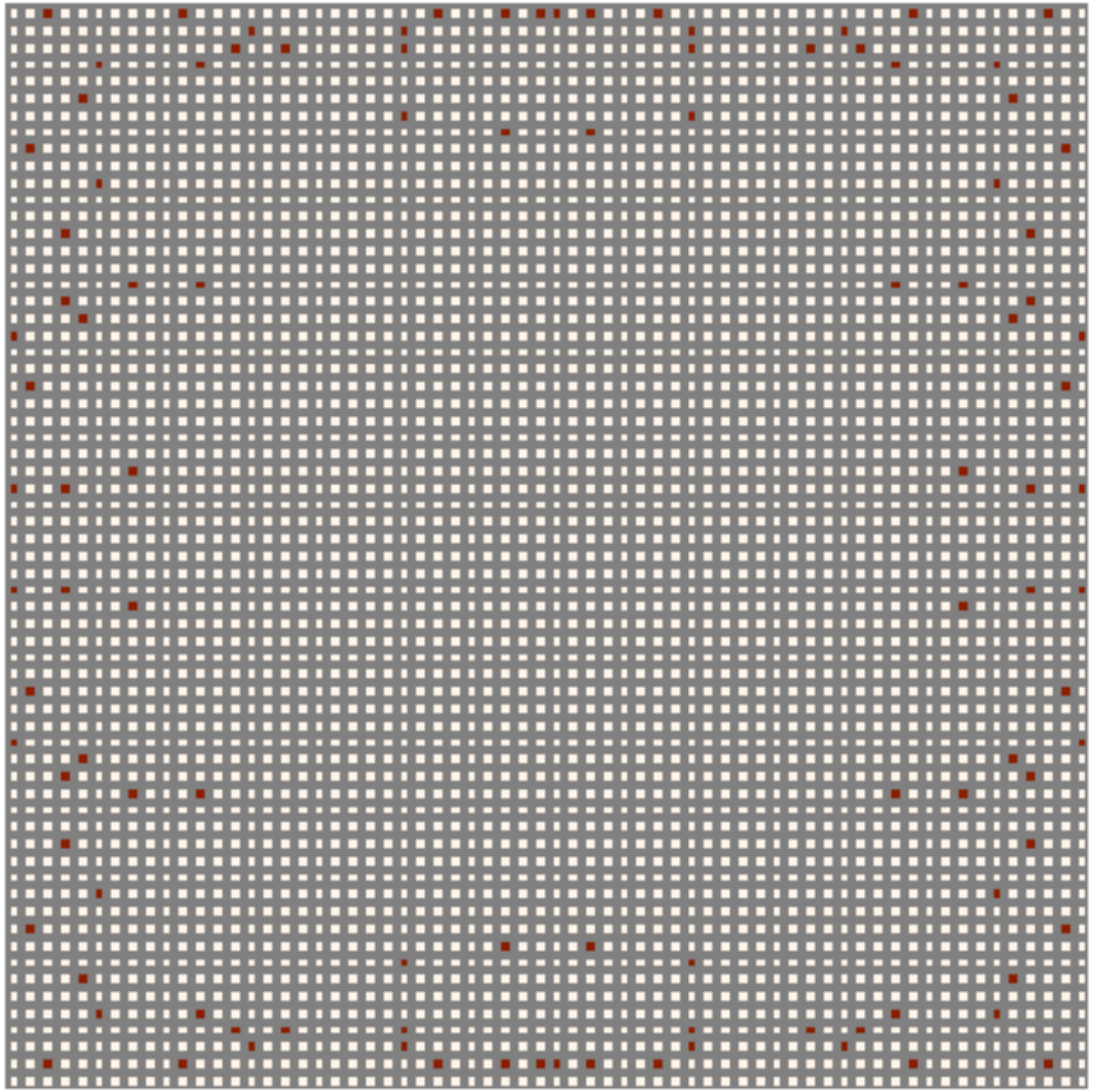
Under supervision of Jordan Ellenberg (PhD Advisor) and Amy Cochran (IFDS Mentor)
Collaboration with Adam Z. Wagner | Tel Aviv University

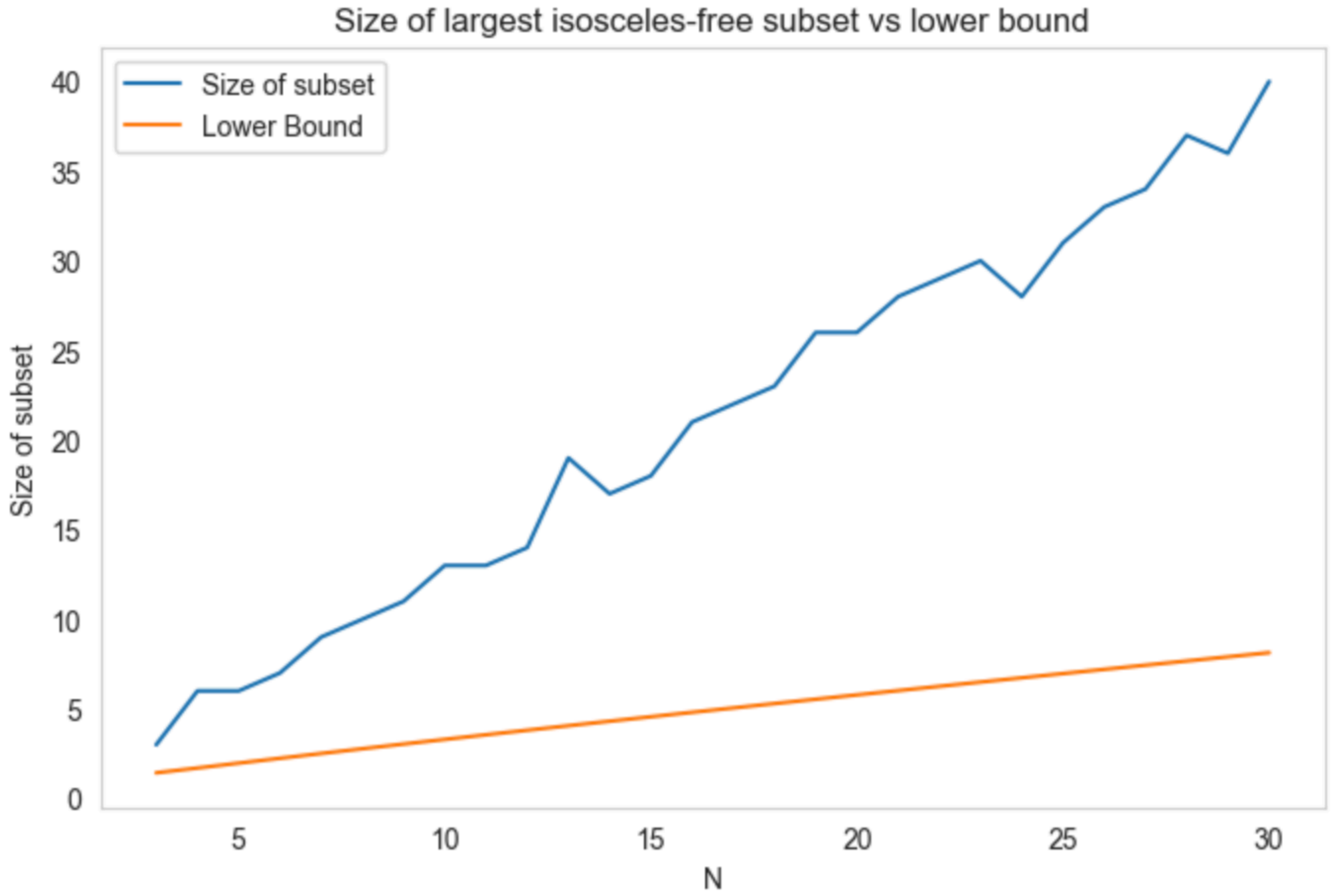
Problem Statement



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 Statement





Aim: To use machine learning to generate best known examples, beat current bounds, explore how we can gain insights.

Overview

Mathematical Motivation and Background

- Motivation: Non Metric Multidimensional Scaling
- Key definitions and propositions
- Known bounds for the problem

How Reinforcement Learning can help

- Reinforcement learning background and main algorithm
- Current results and observations
- Next Steps

Overview

Mathematical Motivation and Background

- Motivation: Non Metric Multidimensional Scaling
- Key definitions and propositions
- Known bounds for the problem

How Reinforcement Learning can help

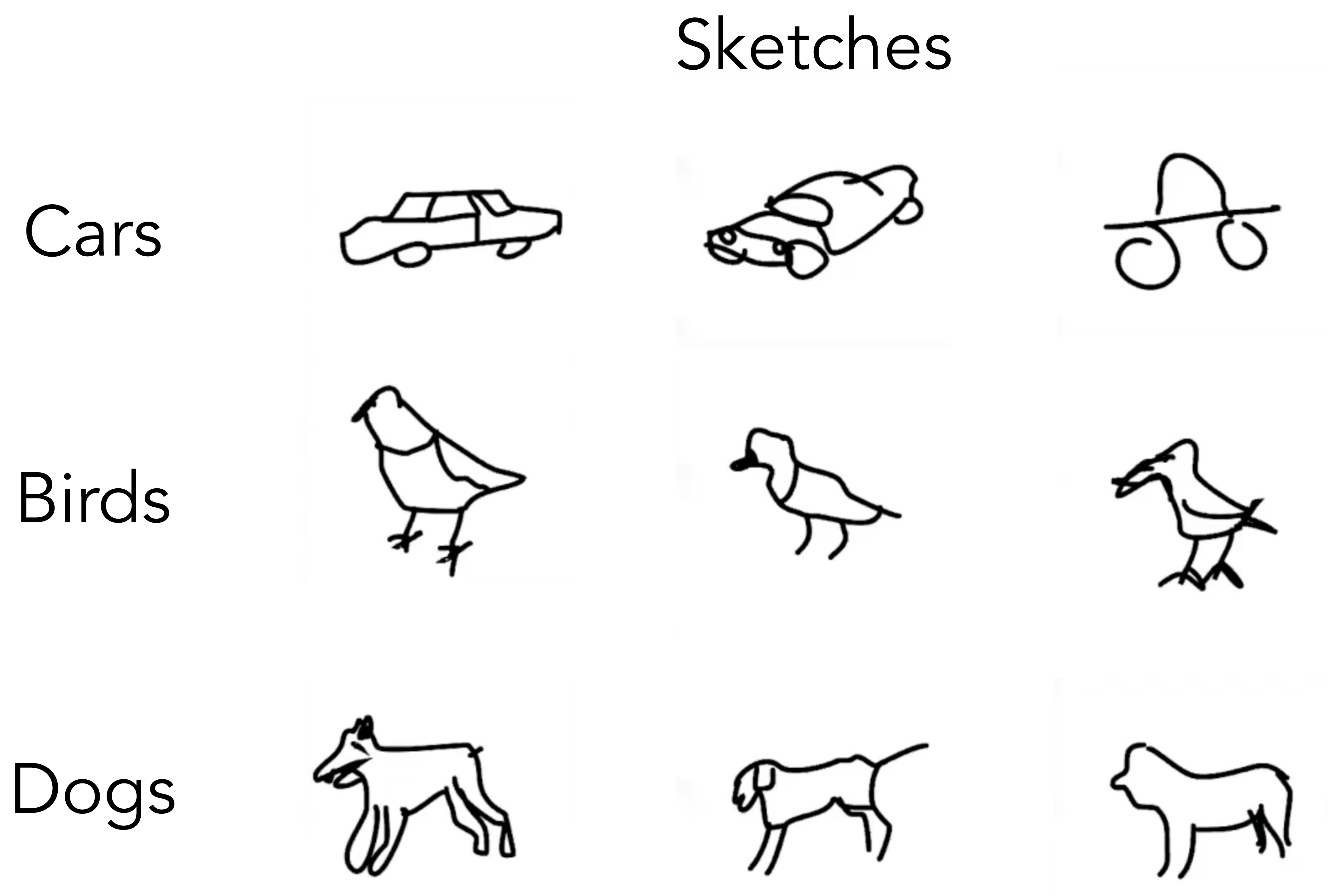
- Reinforcement learning background and main algorithm
- Current results and observations
- Next Steps

Motivation

Psychology - How closely are the representations of concepts in our mind related?

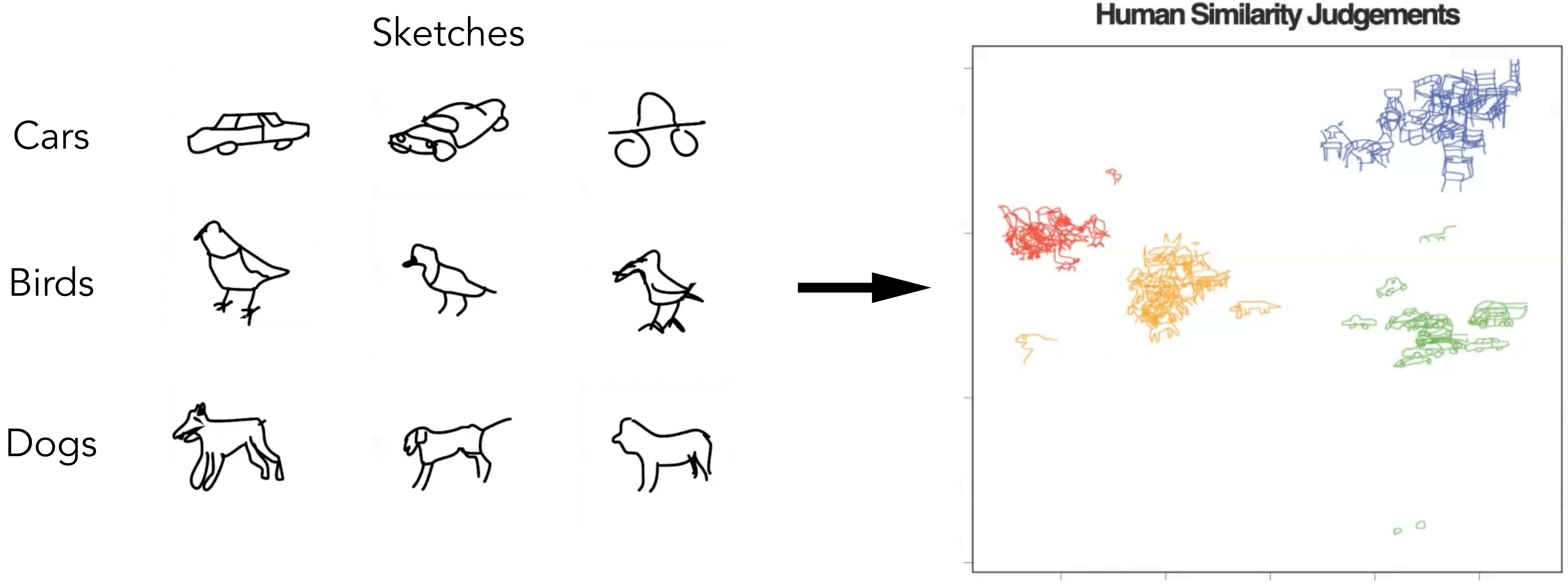
Motivation

Psychology - How closely are the representations of concepts in our mind related?



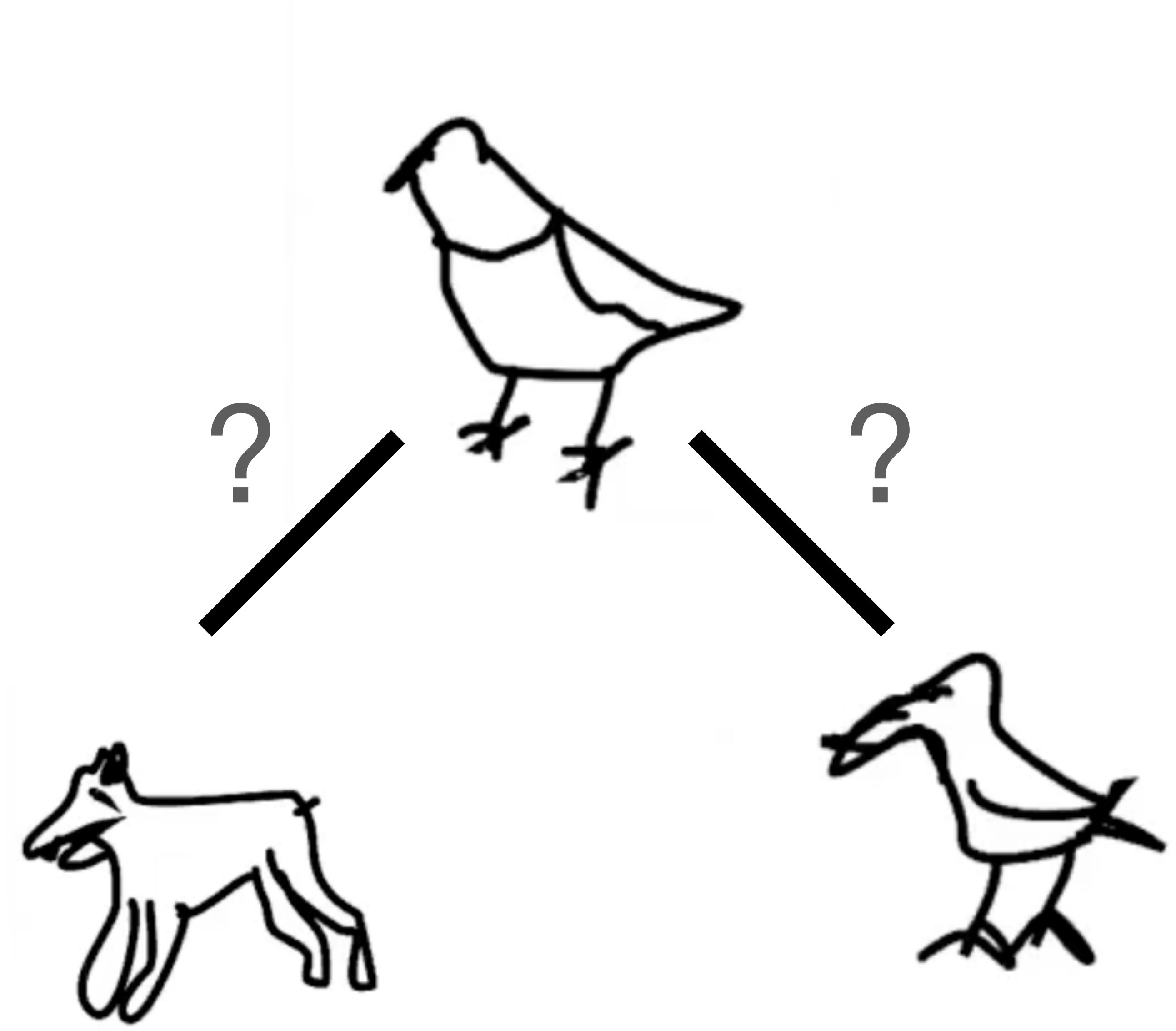
Motivation

Psychology - How closely are the representations of concepts in our mind related?



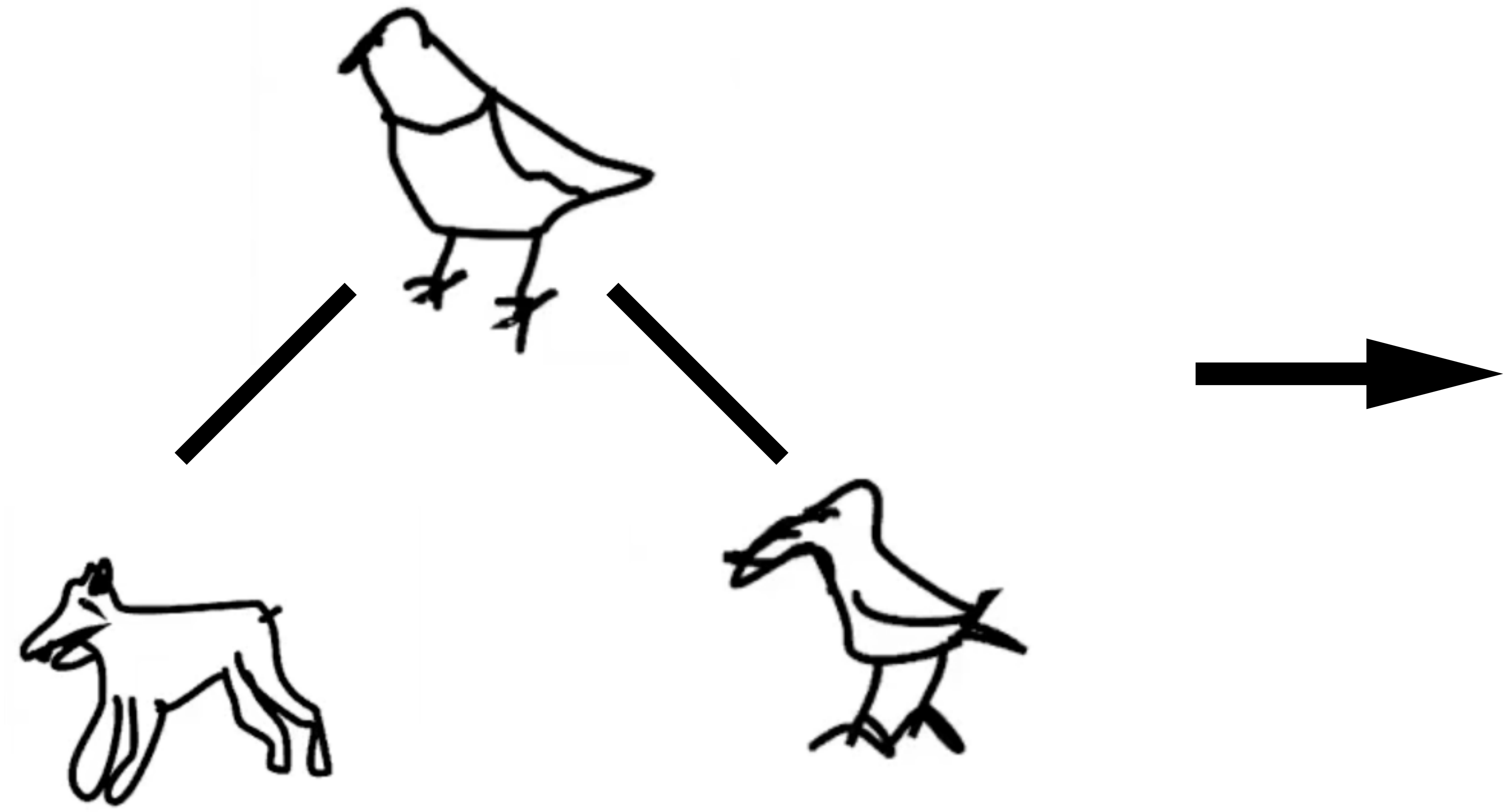
Motivation

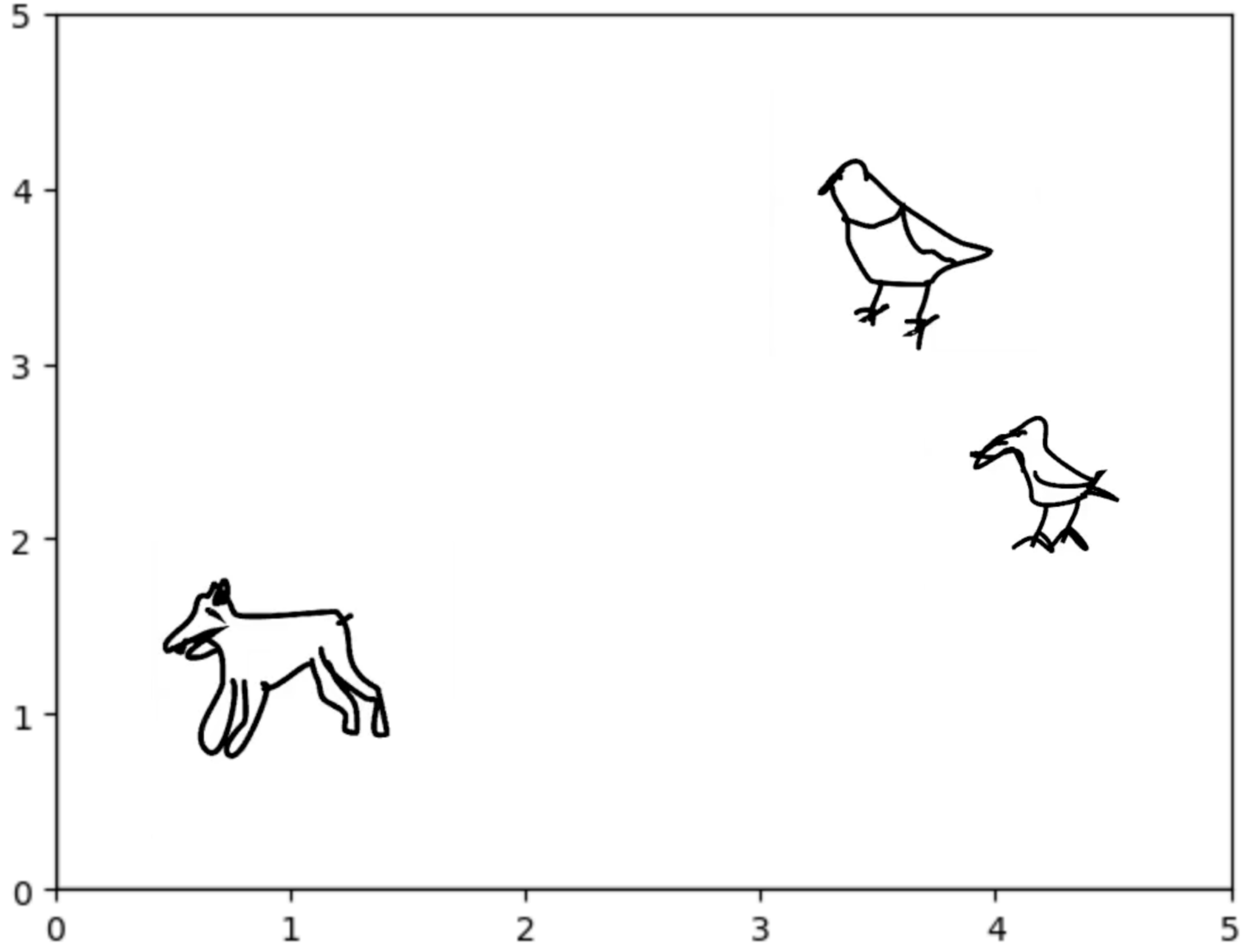
- Framework: Asking questions of the form “is concept A closer to concept B or concept C?”



Motivation

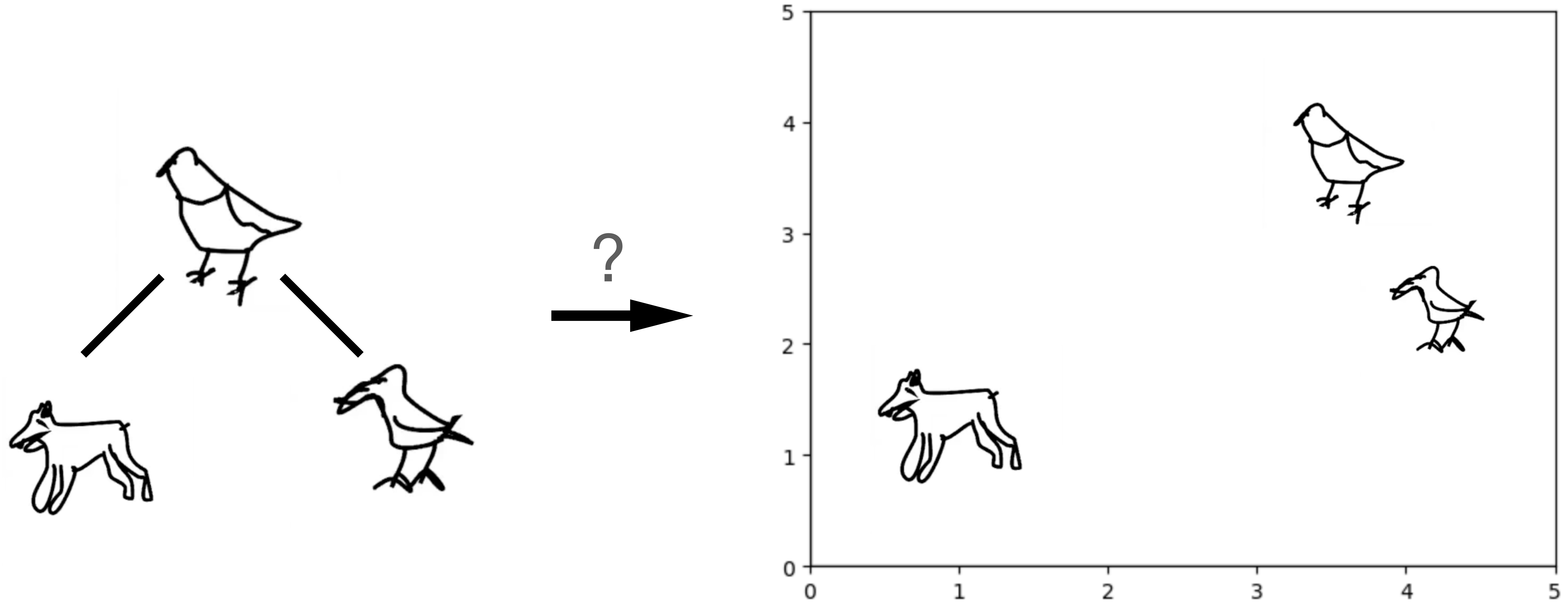
- Framework: Asking questions of the form "is concept A closer to concept B or concept C?"





Motivation

- Framework: Asking questions of the form “is concept A closer to concept B or concept C?”



Math setting: Ordinal Embeddings

Problem of *non-metric multidimensional scaling*:
Given an integer n, a metric space (M, d), and a set Σ of ordered tuples $(i, j, k, l) \in [1...n]^4$, find an embedding

$$[1...n] \mapsto (x_1, \ldots, x_n) \in M$$

such that for each $(x_i, x_j, x_k, x_l) \in \Sigma$,

$$d(x_i, x_j) < d(x_k, x_l)$$

Math setting: Ordinal Embeddings

Problem of *non-metric multidimensional scaling*:
Given an integer n, a metric space (M, d), and a set Σ of ordered tuples $(i, j, k, l) \in [1...n]^4$, find an embedding

$$[1...n] \mapsto (x_1, \ldots, x_n) \in M$$

such that for each $(x_i, x_j, x_k, x_l) \in \Sigma$,

$$d(x_i, x_j) < d(x_k, x_l)$$

We will restrict ourselves to the case where every tuple in Σ is of the form (i, j, i, k). We call constraints of this form *Triplet Comparisons*

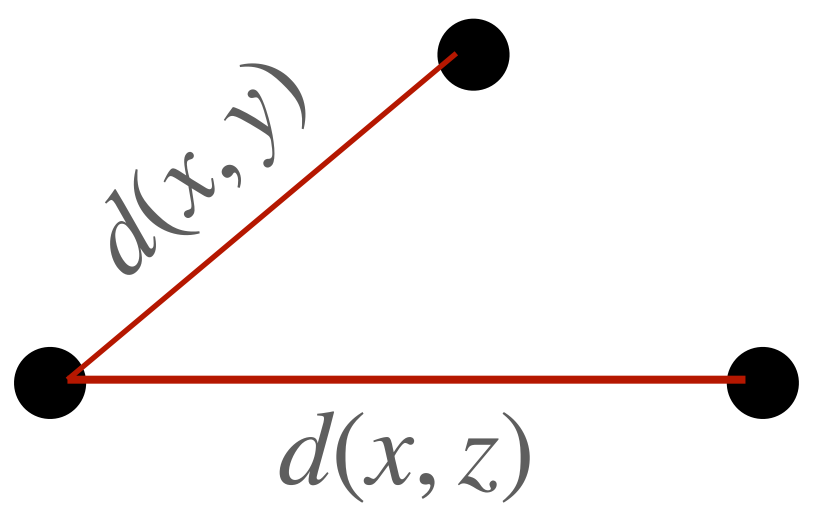
Math setting: non-metric vs classical MDS

Consider $M = \mathbb{R}^d$

Math setting: non-metric vs classical MDS

Consider $M = \mathbb{R}^d$

Classical MDS

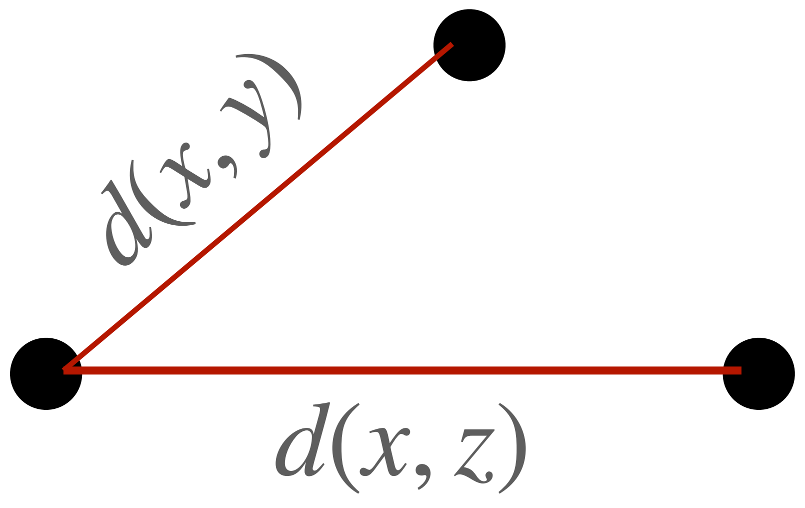


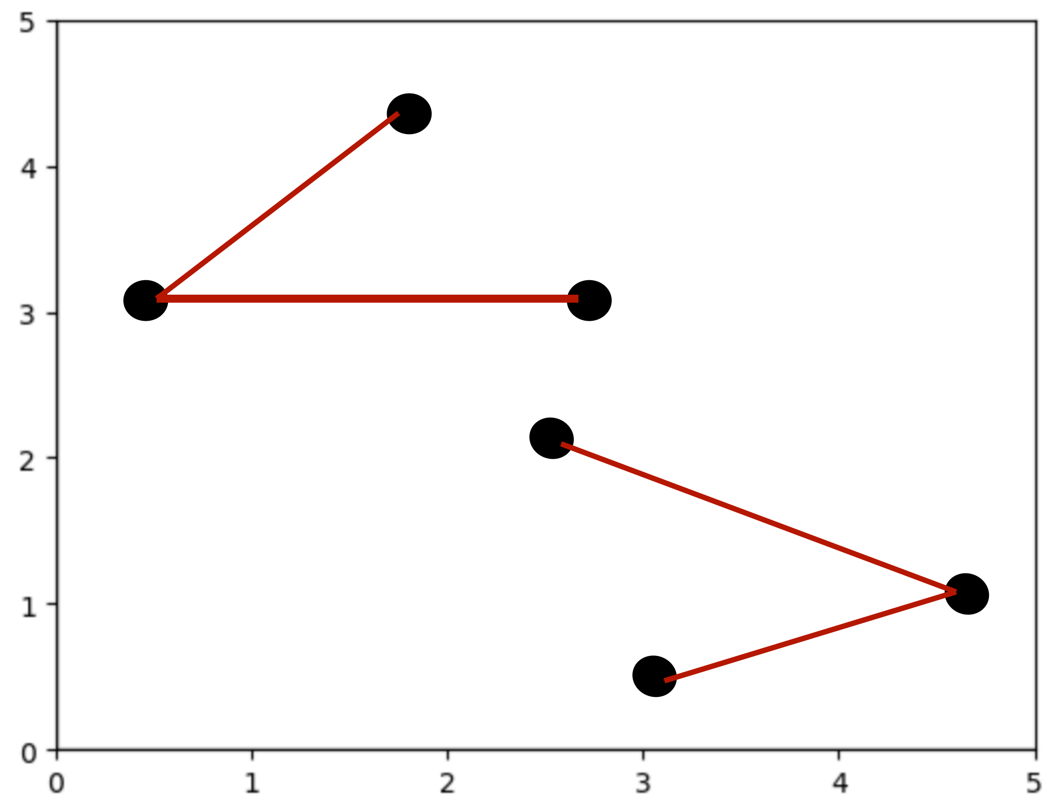
- Have the pairwise distances between the points

Math setting: non-metric vs classical MDS

Consider $M = \mathbb{R}^d$

Classical MDS





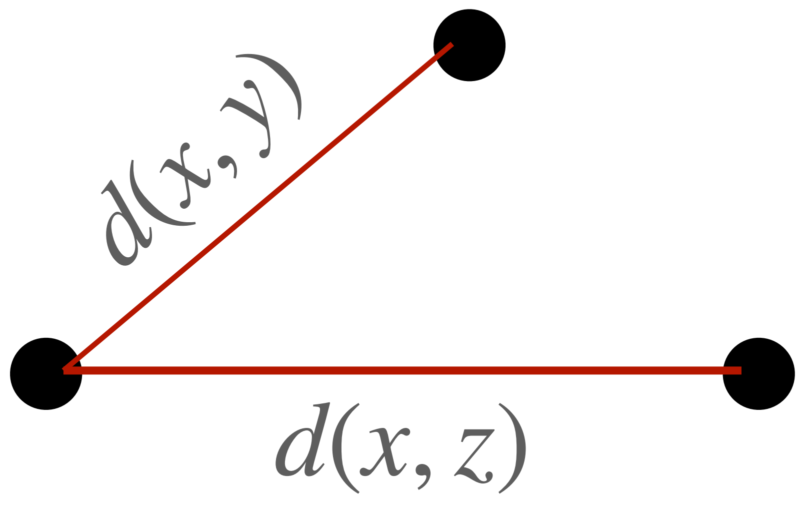
- Have the pairwise distances between the points

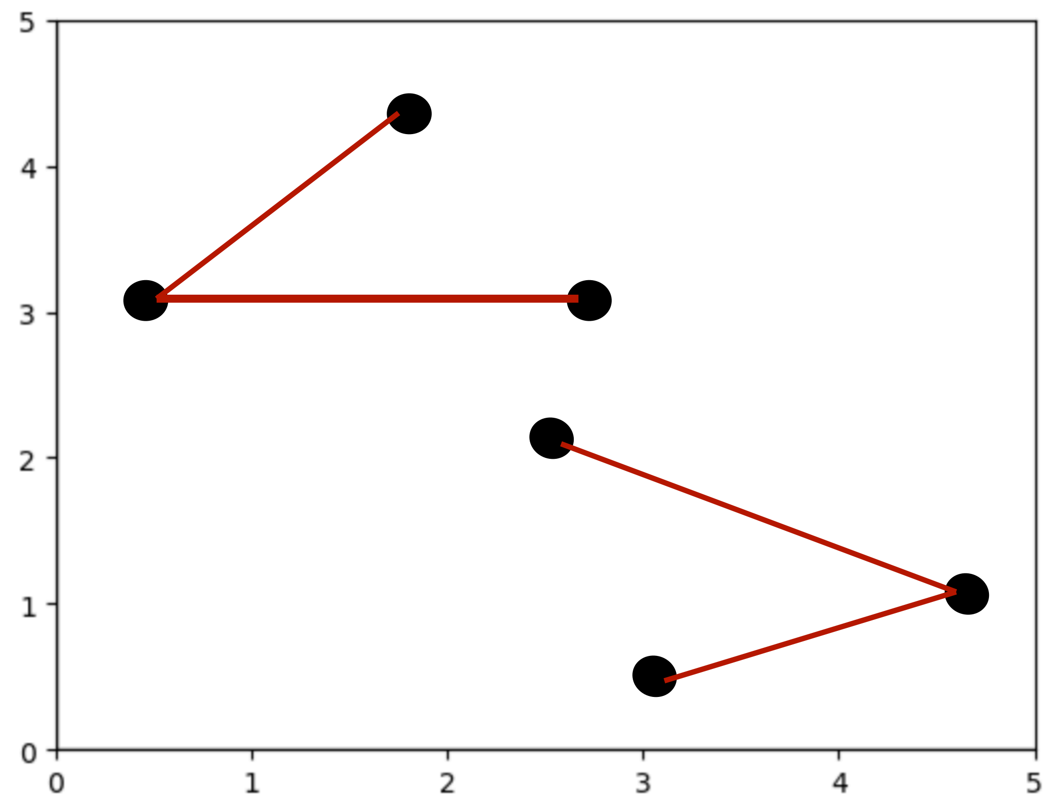
Know: The location of the points up to distance preserving affine linear transformation

Math setting: non-metric vs classical MDS

Consider $M = \mathbb{R}^d$

Classical MDS

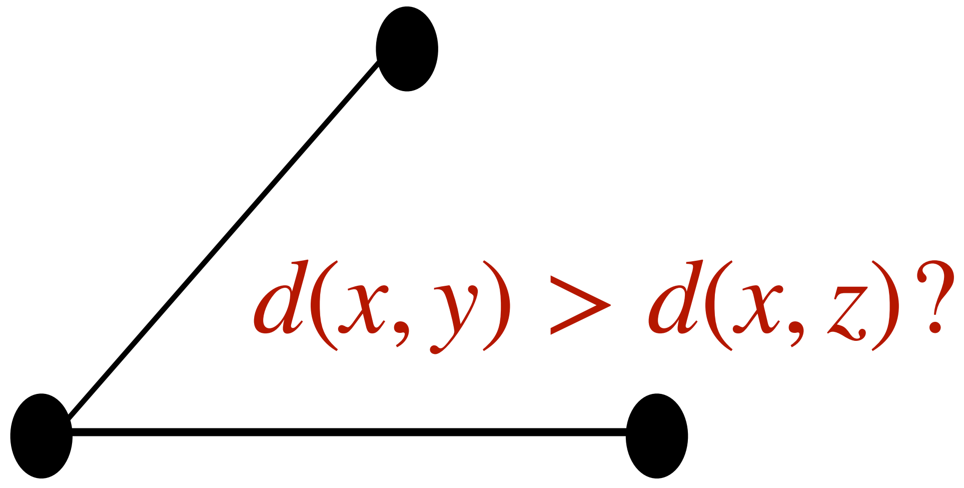




- Have the pairwise distances between the points

Know: The location of the points up to distance preserving affine linear transformation

Non-Metric MDS

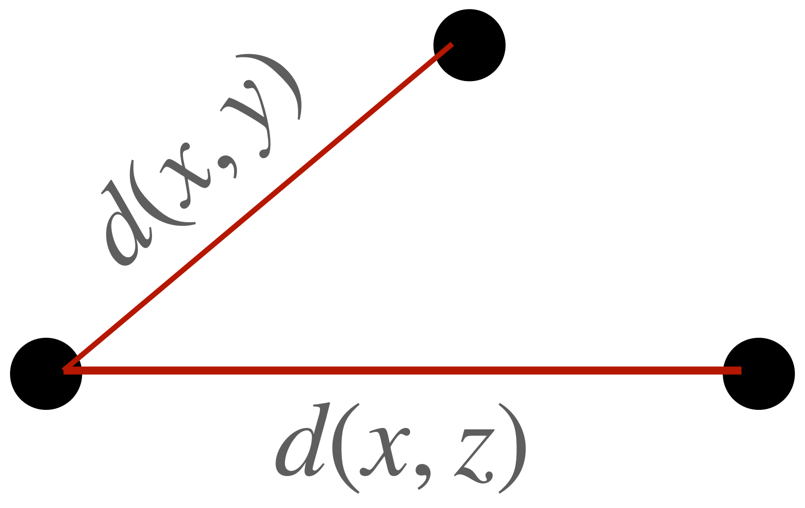


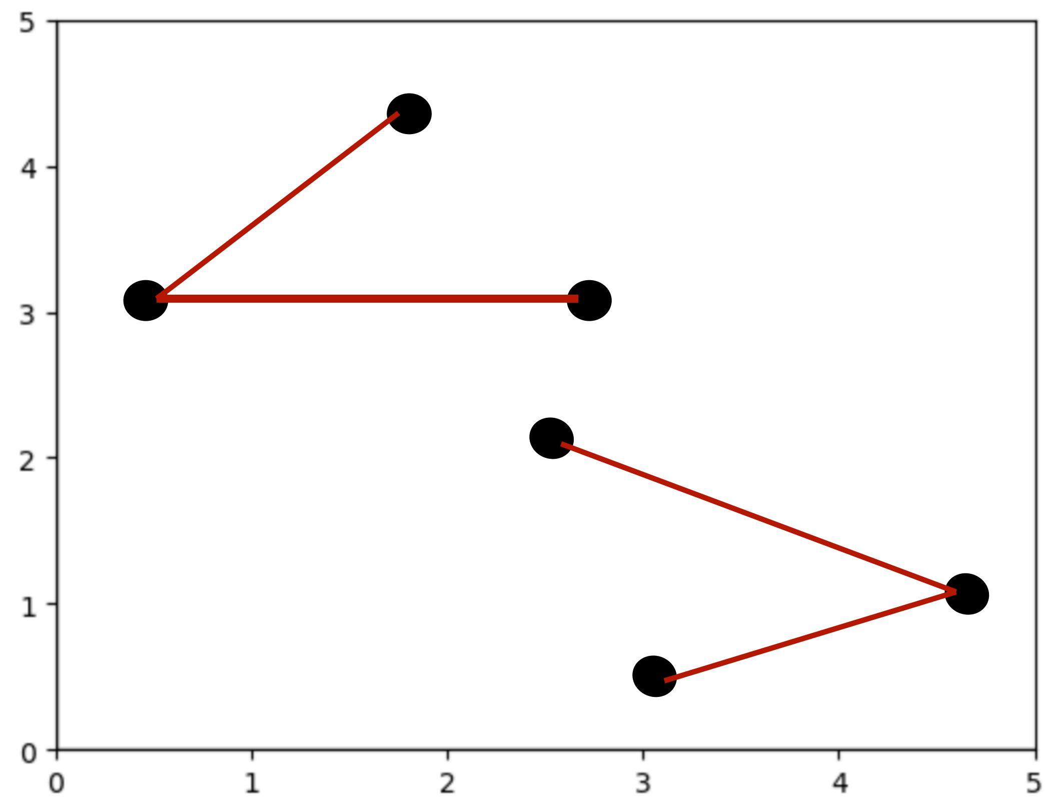
- Have all possible triplet comparisons

Math setting: non-metric vs classical MDS

Consider $M = \mathbb{R}^d$

Classical MDS

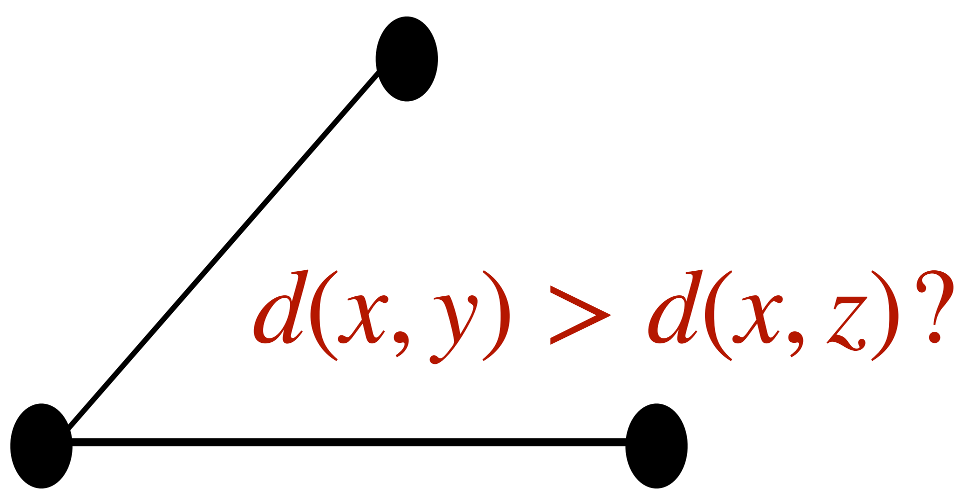


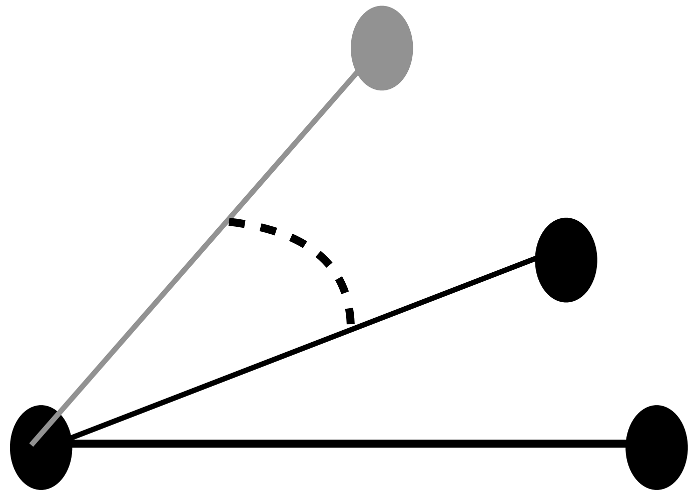


- Have the pairwise distances between the points

Know: The location of the points up to distance preserving affine linear transformation

Non-Metric MDS





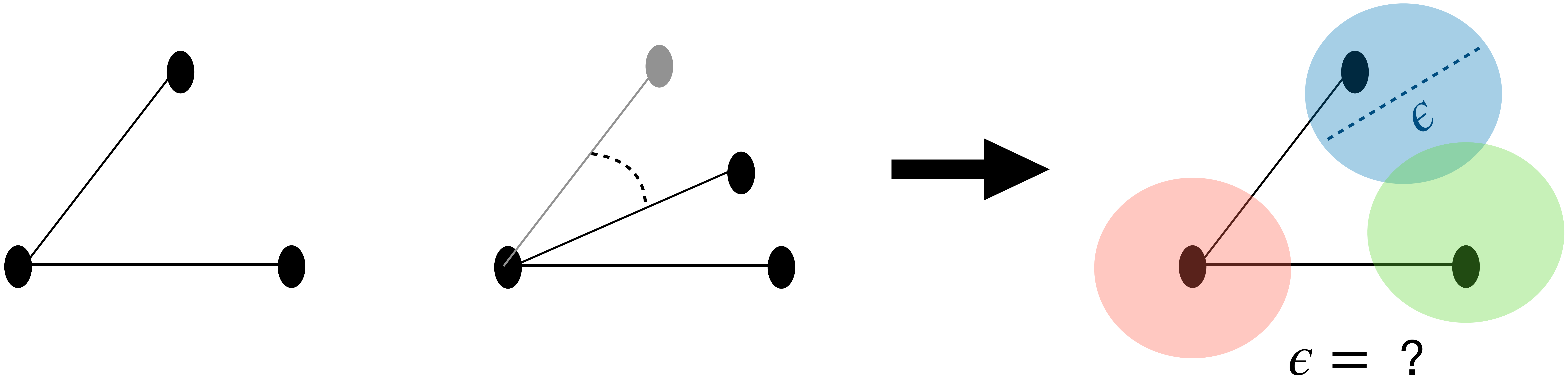
- Have all possible triplet comparisons

If $(x_1, ..., x_n)$ satisfies all constraints in Σ, so does some perturbation of $(x_1, ..., x_n)$

Math setting: Ordinal Embeddings

Question

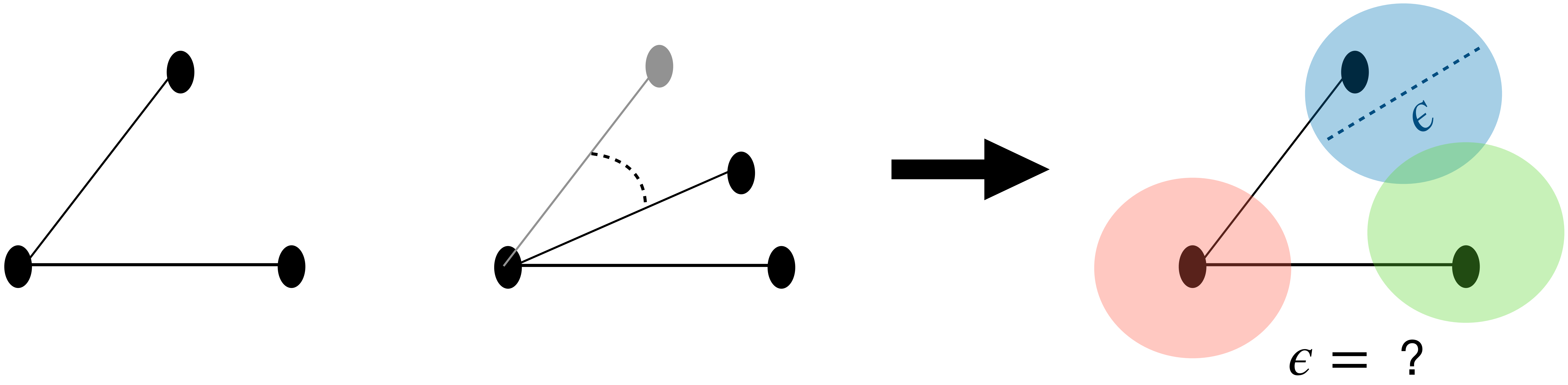
If all the *triplet comparisons* are known, then within what error can we determine $(x_1, \ldots, x_n)$?



Math setting: Ordinal Embeddings

Question

If all the *triplet comparisons* are known, then within what error can we determine $(x_1, \ldots, x_n)$?



Need: A way to compare to points satisfying the same triplet comparisons and establish a metric.

Overview

Mathematical Motivation and Background

- Motivation: Non Metric Multidimensional Scaling
- Key definitions and propositions
- Known bounds for the problem

How Reinforcement Learning can help

- Reinforcement learning background and main algorithm
- Current results and observations
- Next Steps

Math setting: Definitions - Isotonic functions

A function on metric spaces $f : M \to N$ is *weakly isotonic* if for every $m, m', m'' \in M$, we have

$$d_M(m, m') < d_M(m, m'') \quad \text{if and only if} \quad d_N(f(m), f(m')) < d_N(f(m), f(m''))$$

Math setting: Definitions - Isotonic functions

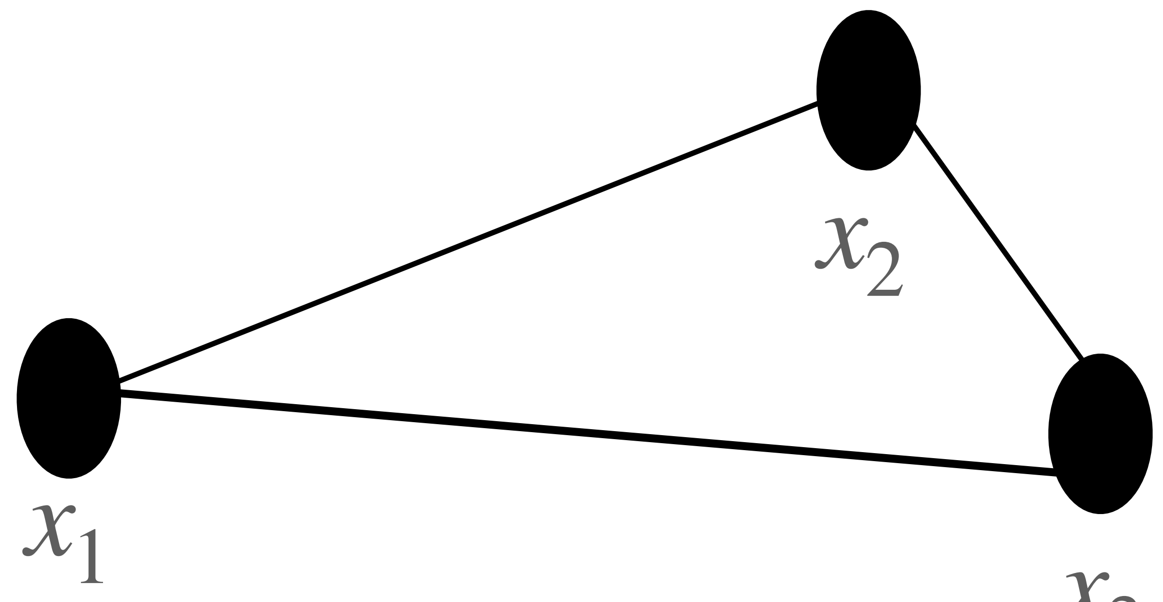
A function on metric spaces $f : M \to N$ is *weakly isotonic* if for every $m, m', m'' \in M$, we have

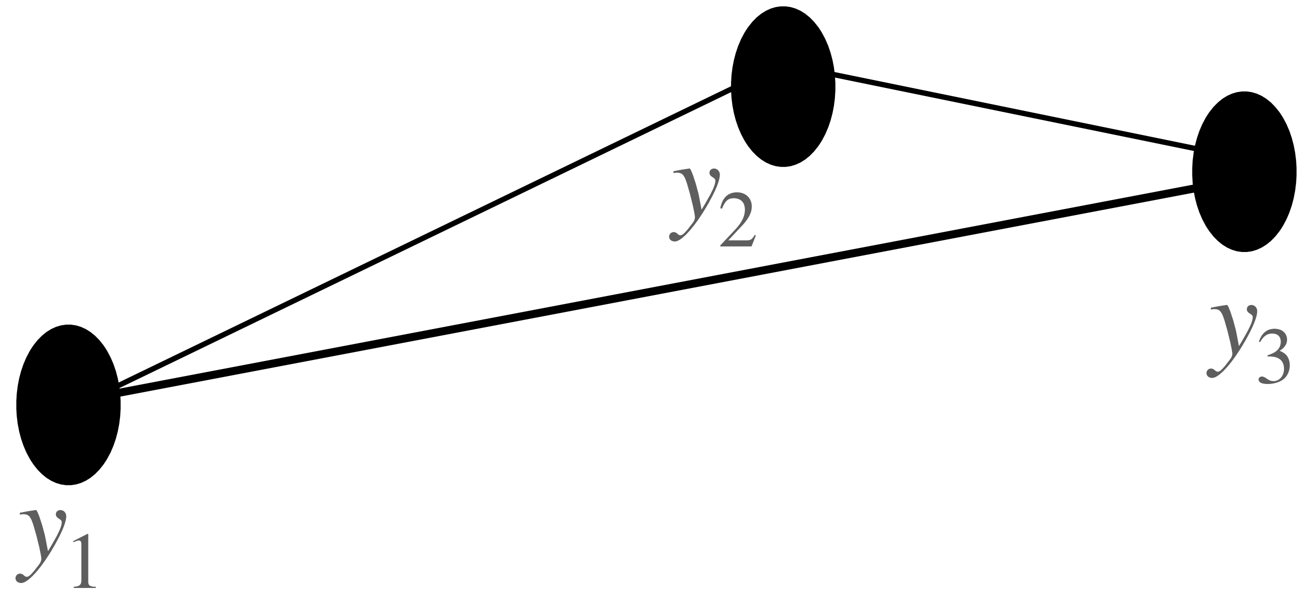
$$d_M(m, m') < d_M(m, m'') \quad \text{if and only if} \quad d_N(f(m), f(m')) < d_N(f(m), f(m''))$$

We say that two n-tuples $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ in an ambient metric space M are *weakly isotonic* if the induced map on the metric spaces

$$\{x_1, \ldots, x_n\} \to \{y_1, \ldots, y_n\}$$

is weakly isotonic.



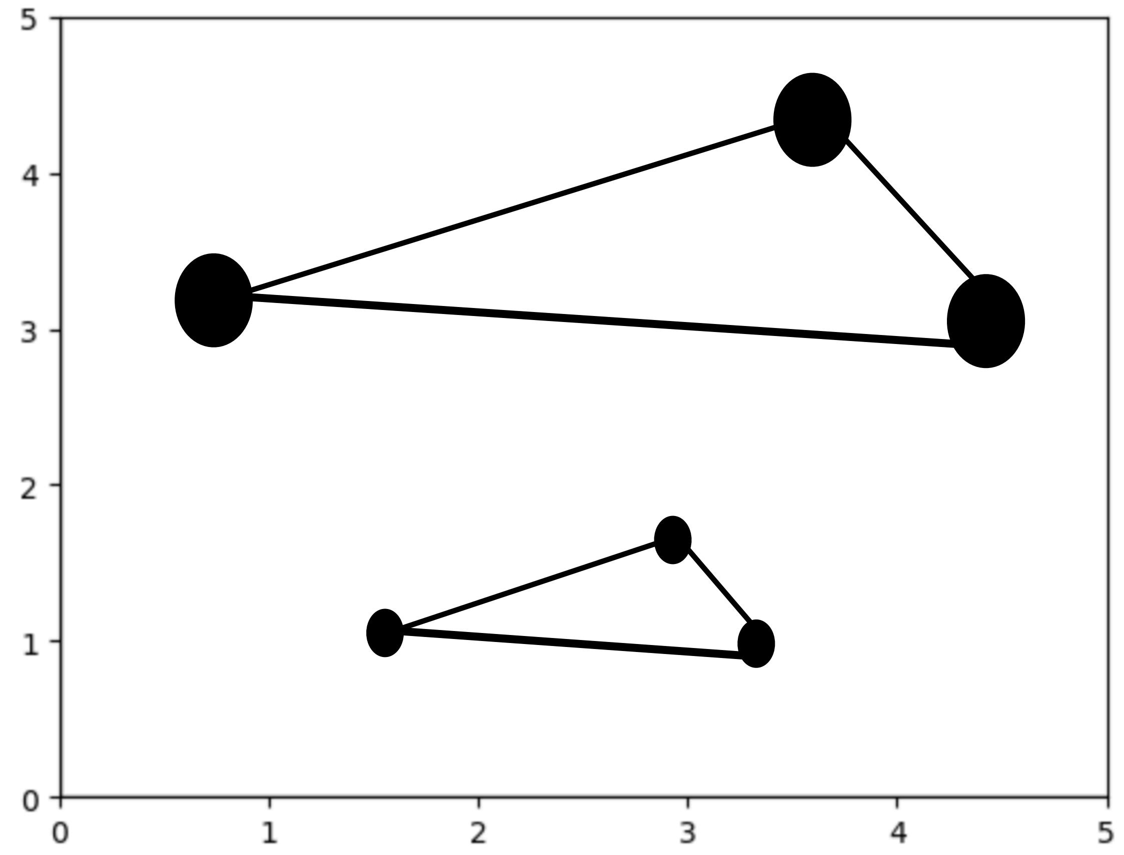


Math setting: Definitions - Distance metrics

For n-tuples $x, y \in M^n$, we denote $d_\infty(x, y) := \max_i d_M(x_i, y_i)$ over $i \in [1,...,n]$

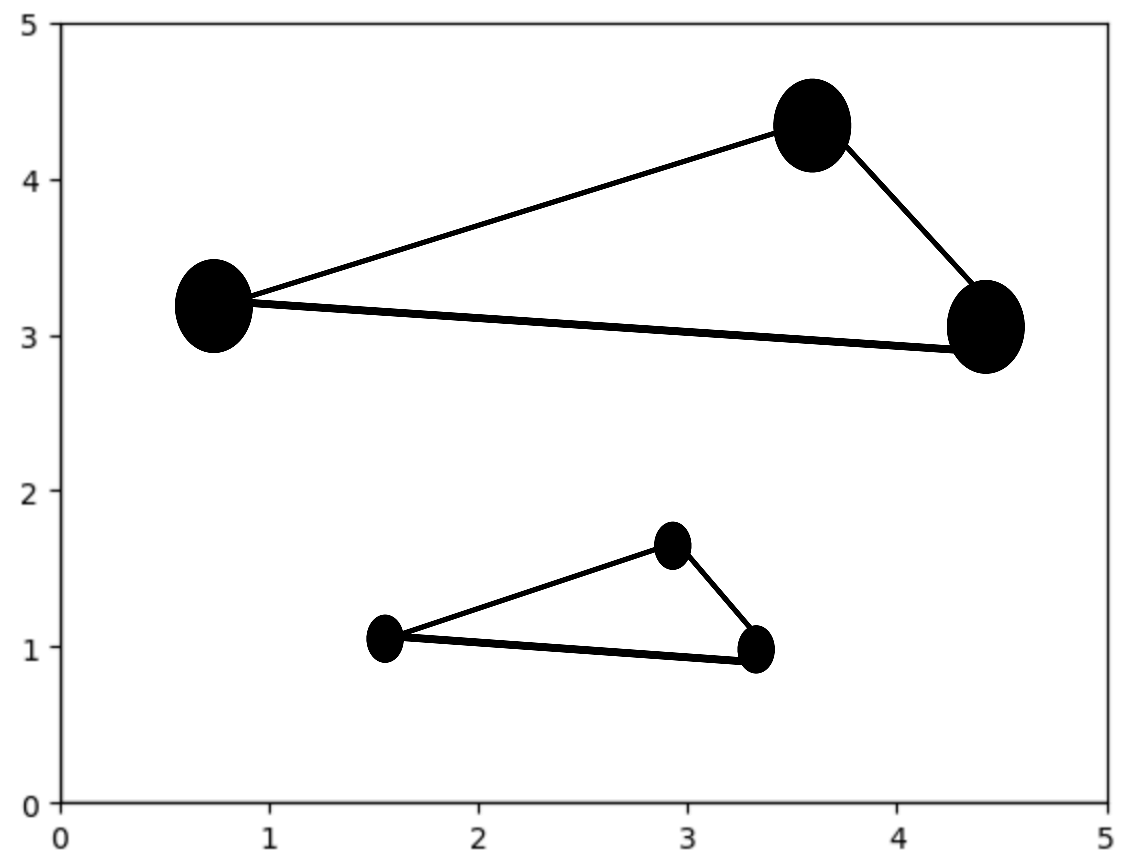
Math setting: Definitions - Distance metrics

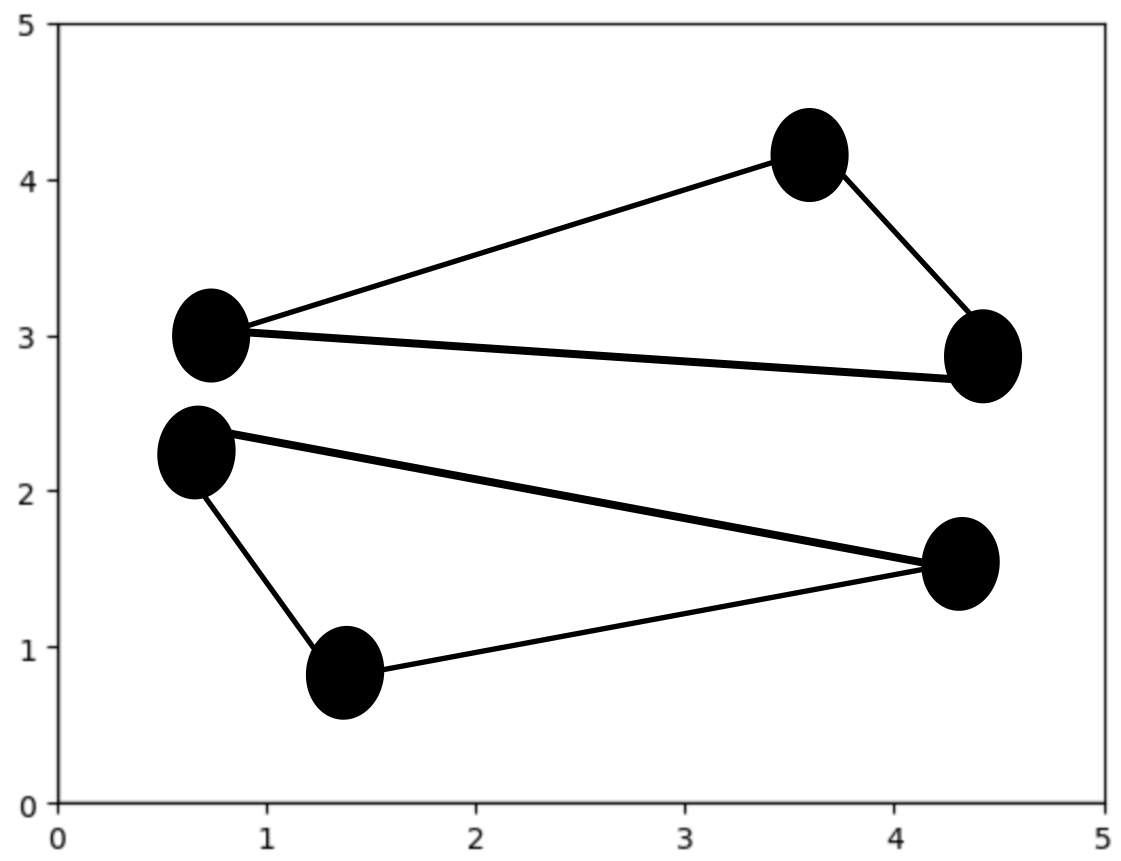
For n-tuples $x, y \in M^n$, we denote $d_\infty(x, y) := \max_i d_M(x_i, y_i)$ over $i \in [1,...,n]$



Math setting: Definitions - Distance metrics

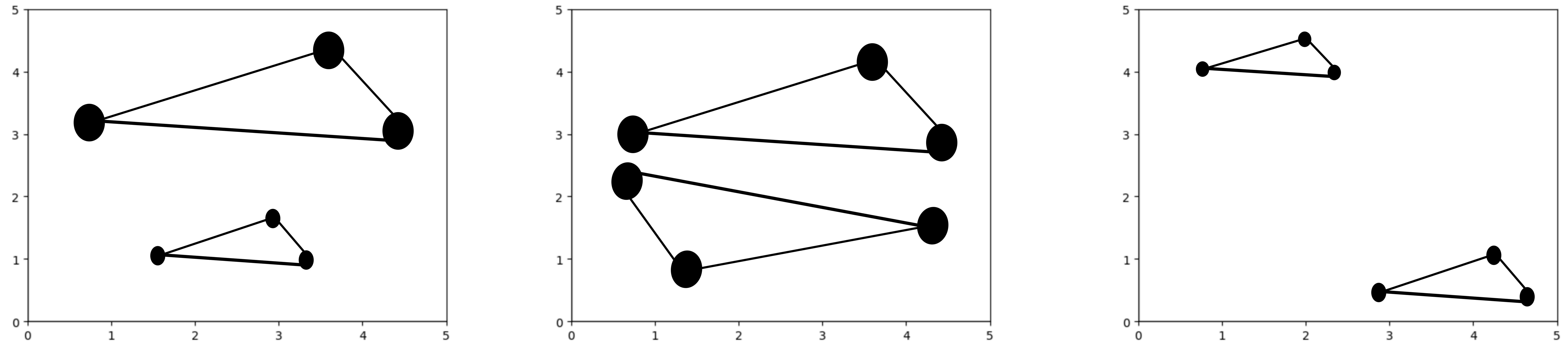
For n-tuples $x, y \in M^n$, we denote $d_\infty(x, y) := \max_i d_M(x_i, y_i)$ over $i \in [1, ..., n]$





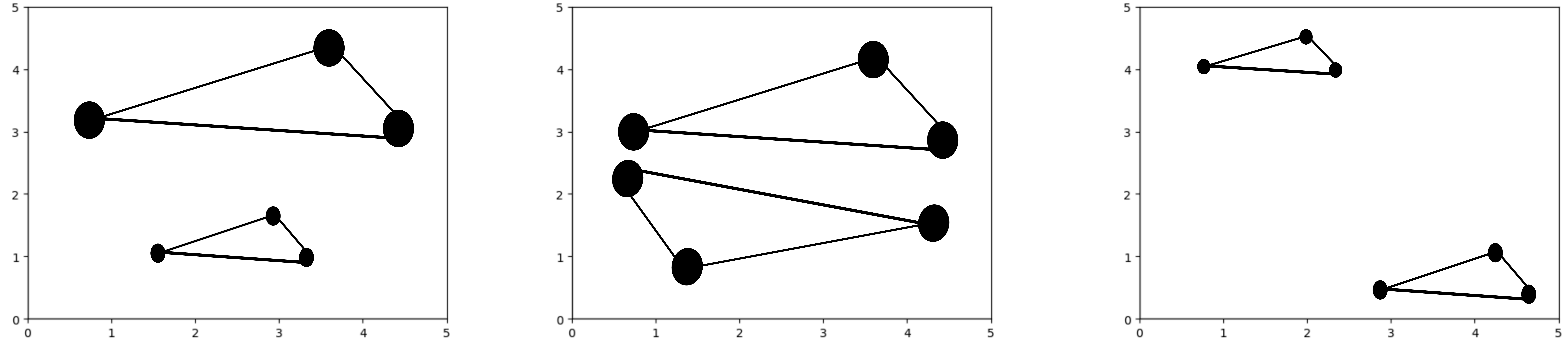
Math setting: Definitions - Distance metrics

For n-tuples $x, y \in M^n$, we denote $d_\infty(x, y) := \max_i d_M(x_i, y_i)$ over $i \in [1,...,n]$



Math setting: Definitions - Distance metrics

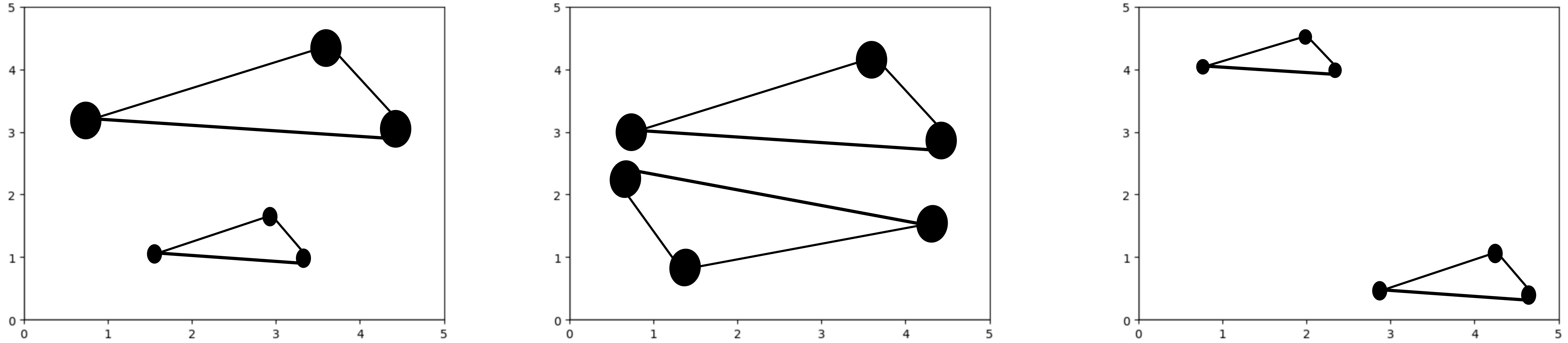
For n-tuples $x, y \in M^n$, we denote $d_\infty(x, y) := \max_i d_M(x_i, y_i)$ over $i \in [1,\ldots,n]$



A *similarity* of a metric space M is a homeomorphism $A : M \to M$ which multiplies all distances by a scalar.

Math setting: Definitions - Distance metrics

For n-tuples $x, y \in M^n$, we denote $d_\infty(x, y) := \max_i d_M(x_i, y_i)$ over $i \in [1,\ldots,n]$

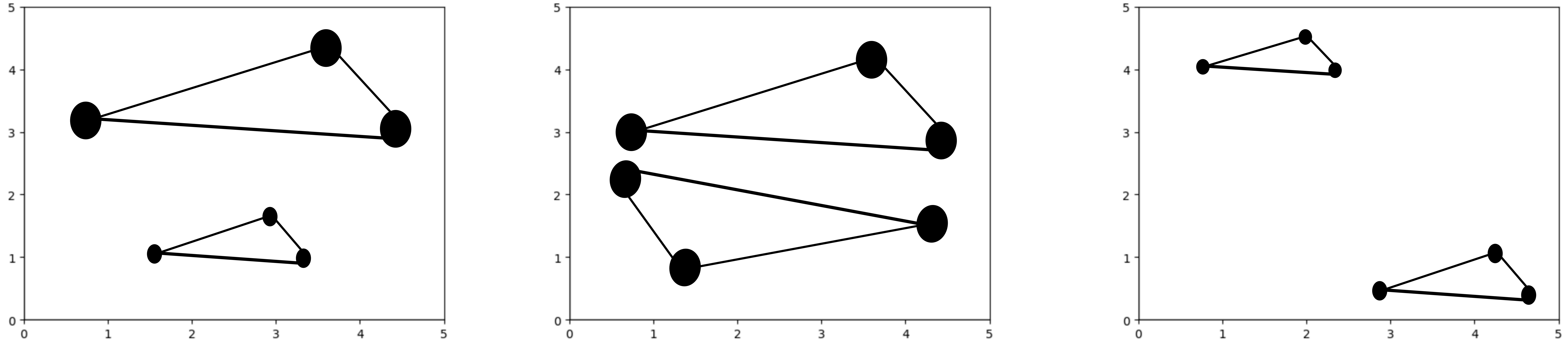


A *similarity* of a metric space M is a homeomorphism $A : M \to M$ which multiplies all distances by a scalar.

Note: if there is a similarity that maps x to y, then x and y are weakly isometric.

Math setting: Definitions - Distance metrics

For n-tuples $x, y \in M^n$, we denote $d_\infty(x, y) := \max_i d_M(x_i, y_i)$ over $i \in [1,\ldots,n]$



A *similarity* of a metric space M is a homeomorphism $A : M \to M$ which multiplies all distances by a scalar.

Note: if there is a similarity that maps x to y, then x and y are weakly isometric.

Final Distance Metric: $\min_A d_\infty(x, Ay)$

Math setting: First proposition

Given a tuple x and a Manifold M, we define the *Hausdorff distance* $\delta_H(x, M)$ between the two as the smallest α such that given any point $m \in M$, there is some i such that

$$d_M(x_i, m) \leq \alpha$$

Math setting: First proposition

Given a tuple x and a Manifold M, we define the *Hausdorff distance* $\delta_H(x, M)$ between the two as the smallest α such that given any point $m \in M$, there is some i such that

$$d_M(x_i, m) \leq \alpha$$

For the following results, we take $M = [0,1]$

Math setting: First proposition

Given a tuple x and a Manifold M, we define the <u>Hausdorff distance</u> $\delta_H(x, M)$ between the two as the smallest α such that given any point $m \in M$, there is some i such that

$$d_M(x_i, m) \leq \alpha$$

For the following results, we take $M = [0,1]$

<u>Proposition [J. Ellenberg, L. Jain, 2019]</u>

For tuples $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \subset [0,1]$, if we have that:

- $\delta_H(x, [0,1]) \leq \alpha$
- x and y are weakly isotonic

Then there exists a similarity A such that $d_\infty(x, Ay) = O_\epsilon(\alpha^{1-\epsilon})$

Math setting: First proposition

Proposition [J. Ellenberg, L. Jain, 2019]

For tuples $x = (x_1, ..., x_n), y = (y_1, ..., y_n) \subset [0,1]$, if we have that:

- $\delta_H(x, [0,1]) \leq \alpha$
- x and y are weakly isotonic

Then there exists a similarity A such that $d_\infty(x, Ay) = O_\epsilon(\alpha^{1-\epsilon})$

The Hausdorff Condition: There are no large gaps

Math setting: First proposition

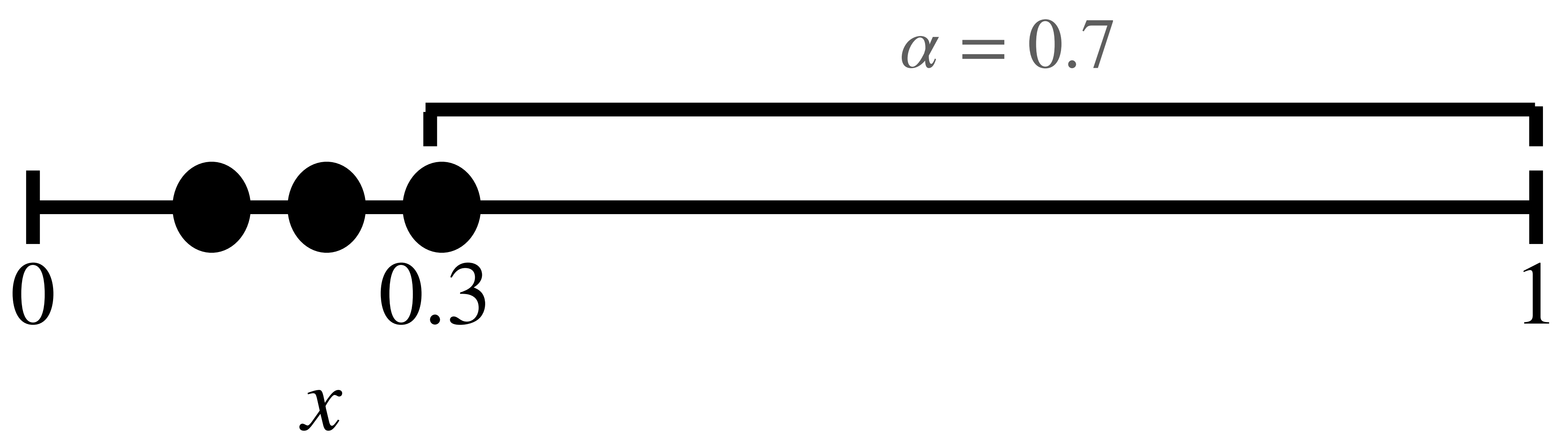
Proposition [J. Ellenberg, L. Jain, 2019]

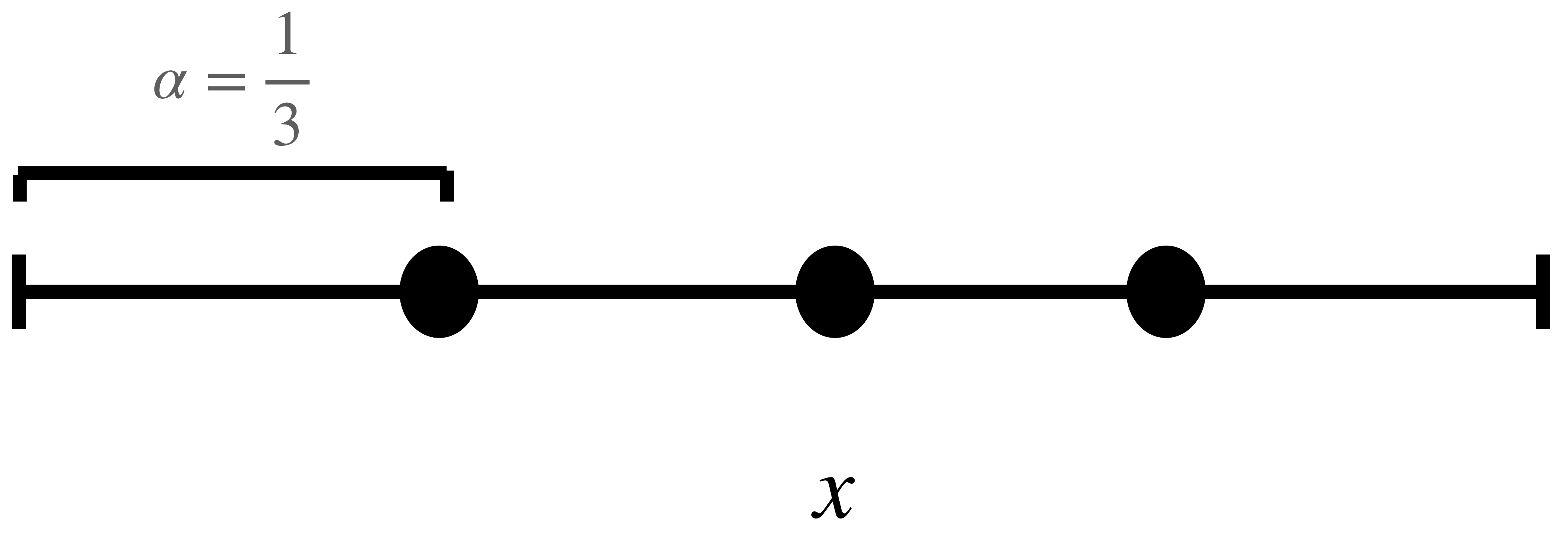
For tuples $x = (x_1, ..., x_n), y = (y_1, ..., y_n) \subset [0,1]$, if we have that:

- $\delta_H(x, [0,1]) \le \alpha$
- x and y are weakly isotonic

Then there exists a similarity A such that $d_\infty(x, Ay) = O_\epsilon(\alpha^{1-\epsilon})$

The Hausdorff Condition: There are no large gaps





Math setting: First proposition

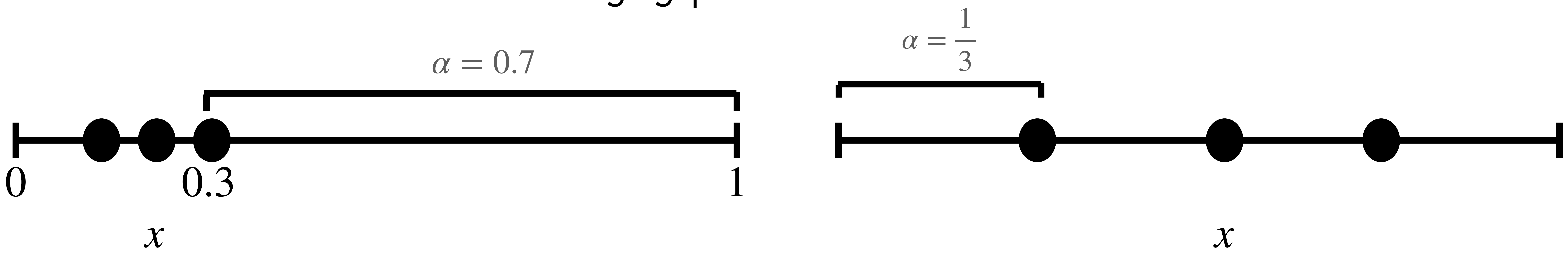
Proposition [J. Ellenberg, L. Jain, 2019]

For tuples $x = (x_1, ..., x_n), y = (y_1, ..., y_n) \subset [0,1]$, if we have that:

- $\delta_H(x, [0,1]) \leq \alpha$
- x and y are weakly isotonic

Then there exists a similarity A such that $d_\infty(x, Ay) = O_\epsilon(\alpha^{1-\epsilon})$

The Hausdorff Condition: There are no large gaps



Question: How optimal is $O_\epsilon(\alpha^{1-\epsilon})$?

Math setting: First proposition

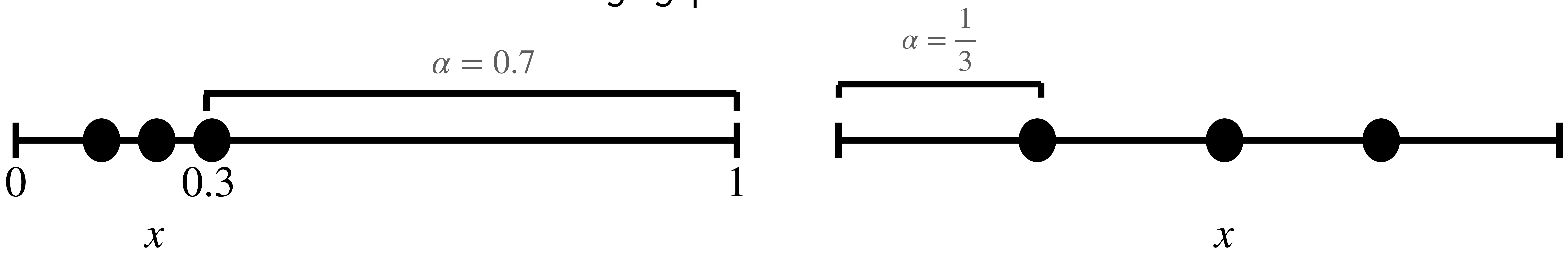
Proposition [J. Ellenberg, L. Jain, 2019]

For tuples $x = (x_1, ..., x_n), y = (y_1, ..., y_n) \subset [0,1]$, if we have that:

- $\delta_H(x, [0,1]) \leq \alpha$
- x and y are weakly isotonic

Then there exists a similarity A such that $d_\infty(x, Ay) = O_\epsilon(\alpha^{1-\epsilon})$

The Hausdorff Condition: There are no large gaps



Question: How optimal is $O_\epsilon(\alpha^{1-\epsilon})$? Turns out, almost optimal!

Math setting: Proposition 2

Proposition [J. Ellenberg, L. Jain, 2019]

For tuples $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \subset [0,1]$, if we have that:

- $\delta_H(x, [0,1]) \leq \alpha$
- x and y are weakly isotonic

Then there exists a similarity A such that $d_\infty(x, Ay) = O_\epsilon(\alpha^{1-\epsilon})$

Proposition 2 [J. Ellenberg, L. Jain, 2019]

For sufficiently small α, there exist tuples $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \subseteq [0,1]$ such that

- $\delta_H(x, [0,1]) \leq \alpha$
- x and y are weakly isotonic

With $d_\infty(x, Ay) = \Omega_\epsilon(\alpha^{1+\epsilon})$ for every similarity A.

Math setting: Proof of Proposition 2

Theorem [Graham, Ron 2006]

For every positive integer k, there exists a subset S of $\mathbb{Z}$ such that

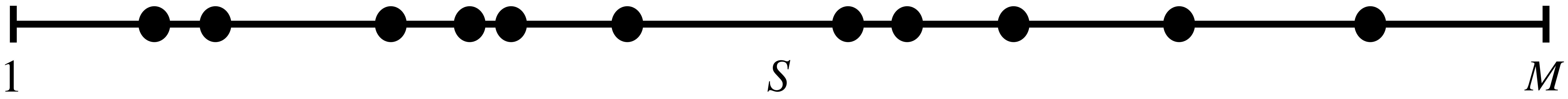
- $S \subset [1, M]$ with $M \geq k^{c \log k}$ for some absolute constant c
- S has no 3 terms in arithmetic progression
- S has no gaps between successive terms of size greater than k

Math setting: Proof of Proposition 2

Theorem [Graham, Ron 2006]

For every positive integer k, there exists a subset S of $\mathbb{Z}$ such that

- $S \subset [1,M]$ with $M \geq k^{c \log k}$ for some absolute constant c
- S has no 3 terms in arithmetic progression
- S has no gaps between successive terms of size greater than k



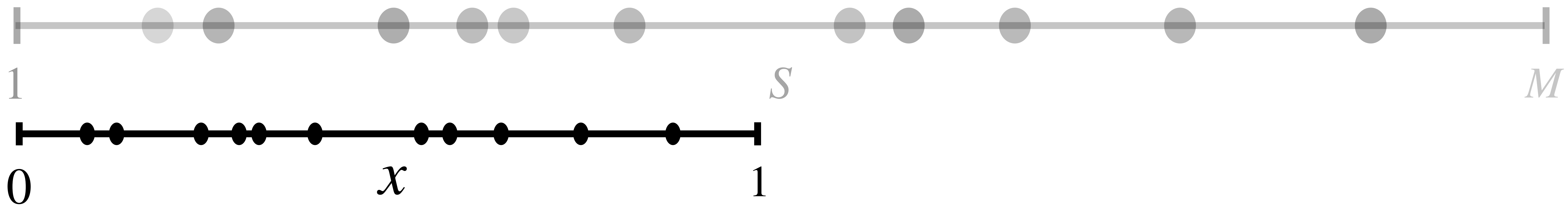
Math setting: Proof of Proposition 2

Theorem [Graham, Ron 2006]

For every positive integer k, there exists a subset S of $\mathbb{Z}$ such that

- $S \subset [1,M]$ with $M \geq k^{c \log k}$ for some absolute constant c
- S has no 3 terms in arithmetic progression
- S has no gaps between successive terms of size greater than k

Take $x = (x_1, \ldots, x_{|S|})$ be the set of points $\{s/M : s \in S\} \subset [0,1]$.



Math setting: Proof of Proposition 2

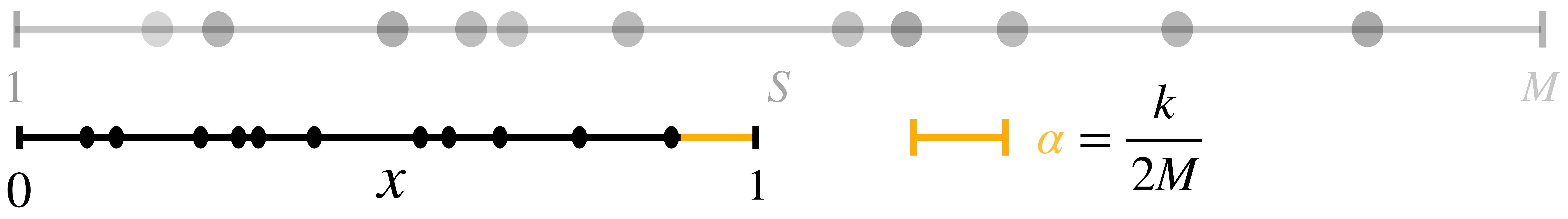
Theorem [Graham, Ron 2006]

For every positive integer k, there exists a subset S of $\mathbb{Z}$ such that

- $S \subset [1, M]$ with $M \geq k^{c \log k}$ for some absolute constant c
- S has no 3 terms in arithmetic progression
- S has no gaps between successive terms of size greater than k

Take $x = (x_1, \ldots, x_{|S|})$ be the set of points $\{s/M : s \in S\} \subset [0,1]$. With $\alpha = k/2M$, we have that

$$\delta_H(x, [0,1]) \leq \alpha$$



Math setting: Proof of Proposition 2

Theorem [Graham, Ron 2006]

For every positive integer k, there exists a subset S of $\mathbb{Z}$ such that

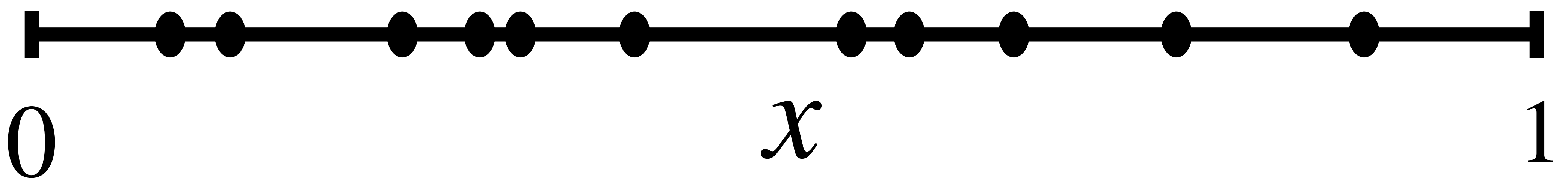
- $S \subset [1,M]$ with $M \geq k^{c \log k}$ for some absolute constant c
- S has no 3 terms in arithmetic progression
- S has no gaps between successive terms of size greater than k

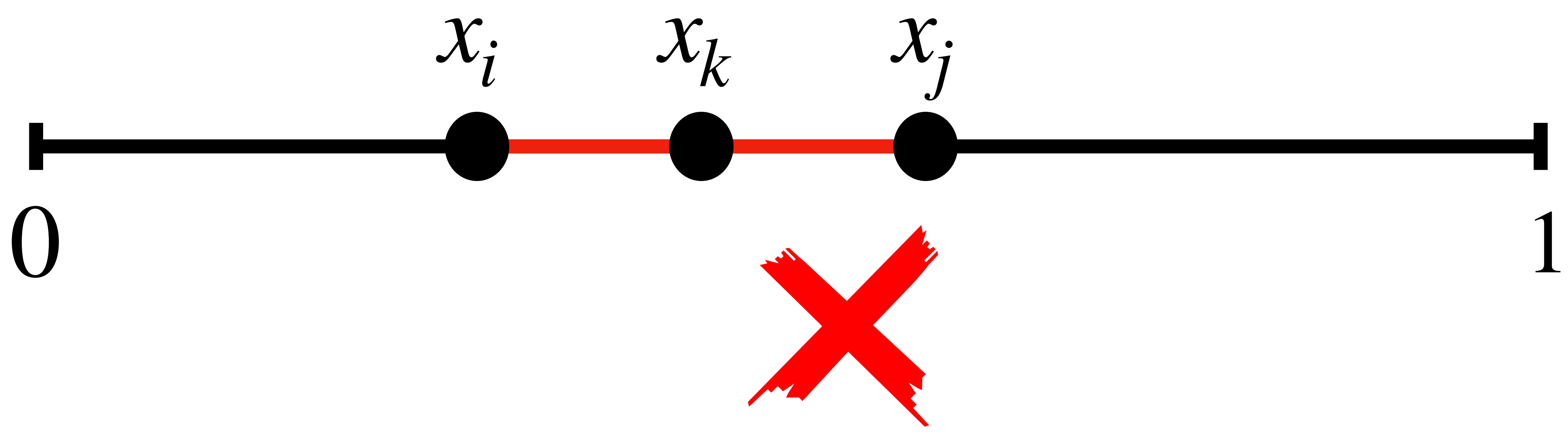
Take $x = (x_1, \ldots, x_{|S|})$ be the set of points $\{s/M : s \in S\} \subset [0,1]$. With $\alpha = k/2M$, we have that

$$\delta_H(x, [0,1]) \leq \alpha$$

Given S has no 3-term arithmetic progression, for each triplet x_i, x_j, x_k we have

$$2x_k - x_i - x_j \neq 0$$





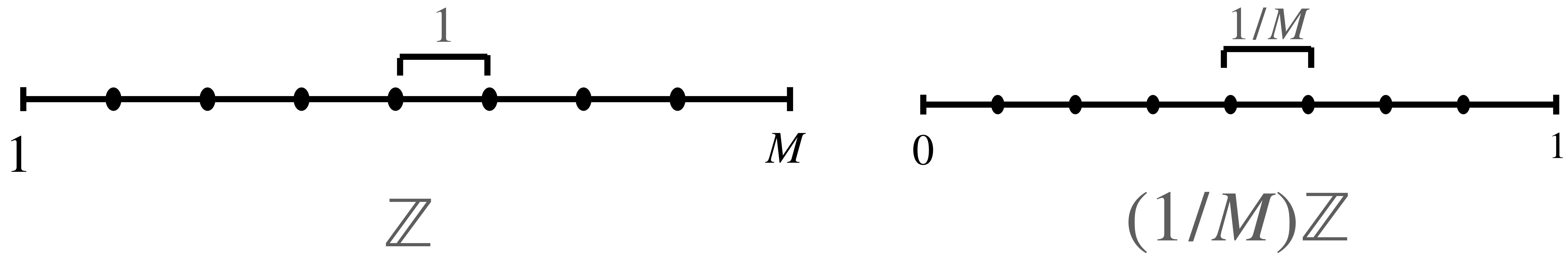
Math setting: Proof of Proposition 2

Given S has no 3-term arithmetic progression, for each triplet x_i, x_j, x_k we have

$$2x_k - x_i - x_j \neq 0$$

Since $x \subset (1/M)\mathbb{Z}$, we get

$$|2x_k - x_i - x_j| \geq \frac{1}{M}$$



Math setting: Proof of Proposition 2

Given S has no 3-term arithmetic progression, for each triplet x_i, x_j, x_k we have

$$2x_k - x_i - x_j \neq 0$$

Since $x \subset (1/M)\mathbb{Z}$, we get

$$|2x_k - x_i - x_j| \geq \frac{1}{M}$$

Therefore, if $y = (y_1, \ldots, y_{|S|})$ satisfies

$$|y_i - x_i| < \frac{1}{2M}$$

for each i, then y is weakly isotonic to x.

Math setting: Proof of Proposition 2

Given S has no 3-term arithmetic progression, for each triplet x_i, x_j, x_k we have

$$2x_k - x_i - x_j \neq 0$$

Since $x \subset (1/M)\mathbb{Z}$, we get

$$|2x_k - x_i - x_j| \geq \frac{1}{M}$$

Therefore, if $y = (y_1, \ldots, y_{|S|})$ satisfies

$$|y_i - x_i| < \frac{1}{2M}$$

for each i, then y is weakly isotonic to x. Note that

$$\alpha = \frac{k}{2M} \leq \frac{k}{2k^{c \log k}} = \frac{1}{2}k^{1-c\log k}$$

which means that $\frac{1}{M} \leq k^{-c\log k}$ is $\Omega(\alpha^{1+\epsilon})$.

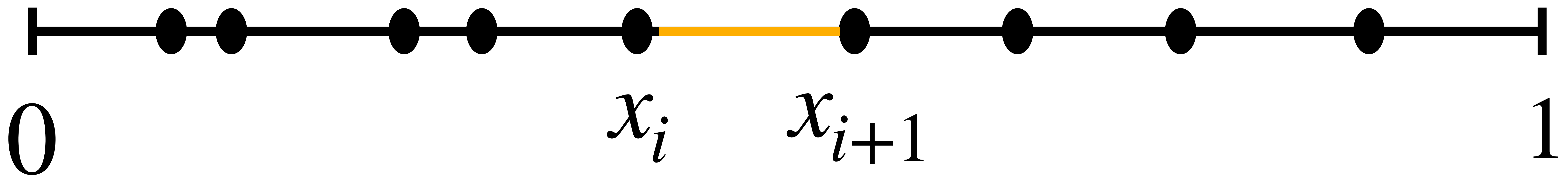
Math setting: Proof of Proposition 2

Need to construct y such that $d_\infty(x, Ay) = \Omega(\alpha^{1+\epsilon})$ for any similarity A.

Math setting: Proof of Proposition 2

Need to construct y such that $d_\infty(x, Ay) = \Omega(\alpha^{1+\epsilon})$ for any similarity A.

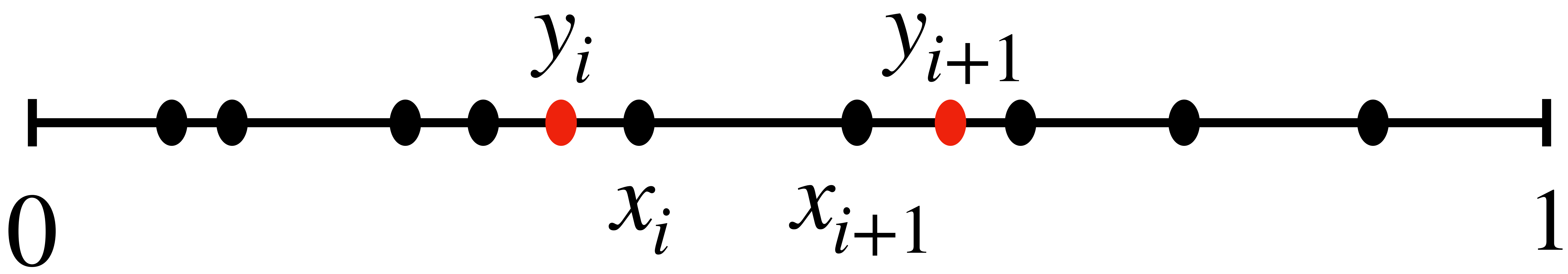
Pick x_i, x_{i+1} such that $|x_i - x_{i+1}|$ are of order α.



Math setting: Proof of Proposition 2

Need to construct y such that $d_\infty(x, Ay) = \Omega(\alpha^{1+\epsilon})$ for any similarity A.

Pick x_i, x_{i+1} such that $|x_i - x_{i+1}|$ are of order α. Then take $y_i = x_i - \beta, y_{i+1} = x_{i+1} + \beta$ for some β on the order $\alpha^{1+\epsilon}$.

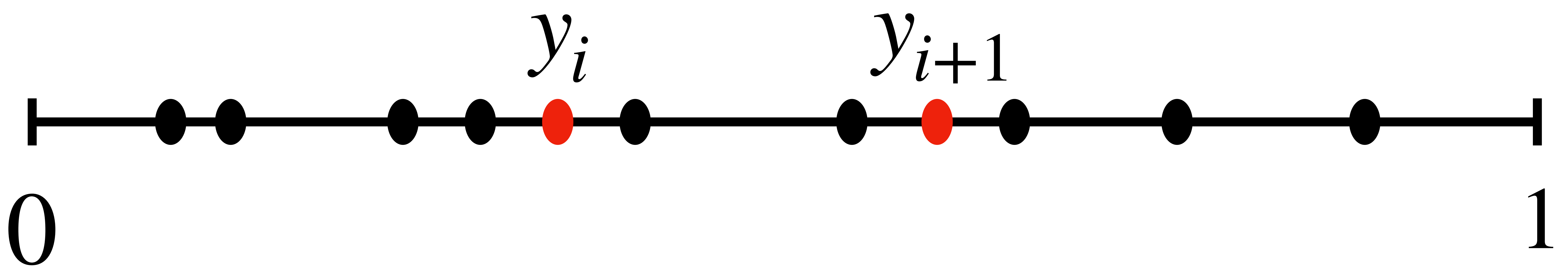


Math setting: Proof of Proposition 2

Need to construct y such that $d_\infty(x, Ay) = \Omega(\alpha^{1+\epsilon})$ for any similarity A.

Pick x_i, x_{i+1} such that $|x_i - x_{i+1}|$ are of order α. Then take $y_i = x_i - \beta, y_{i+1} = x_{i+1} + \beta$ for some β on the order $\alpha^{1+\epsilon}$.

This gives us that $d_\infty(x, y) \geq \beta$.



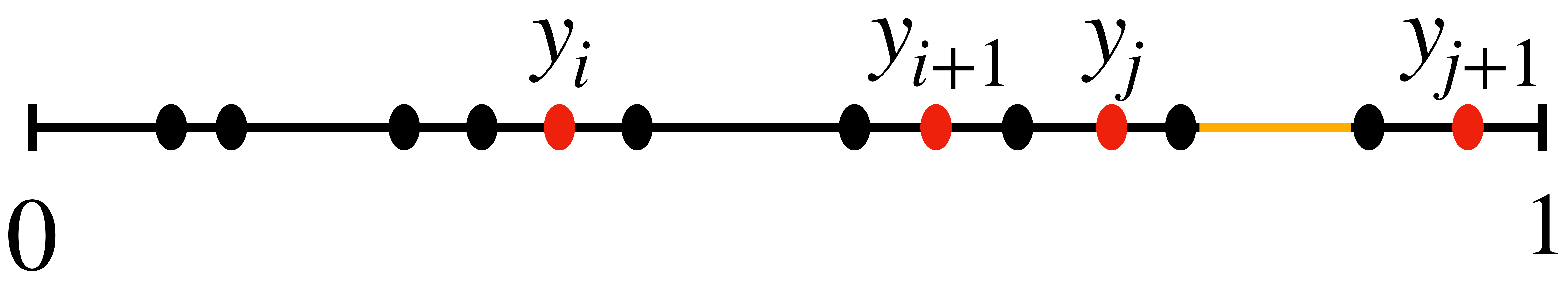
Math setting: Proof of Proposition 2

Need to construct y such that $d_\infty(x, Ay) = \Omega(\alpha^{1+\epsilon})$ for any similarity A.

Pick x_i, x_{i+1} such that $|x_i - x_{i+1}|$ are of order α. Then take $y_i = x_i - \beta, y_{i+1} = x_{i+1} + \beta$ for some β on the order $\alpha^{1+\epsilon}$.

This gives us that $d_\infty(x, y) \geq \beta$.

Pick x_j, x_{j+1} such that $|x_j - x_{j+1}|$ are of order α. Then take $y_j = x_j - \beta, y_{j+1} = x_{j+1} + \beta$.



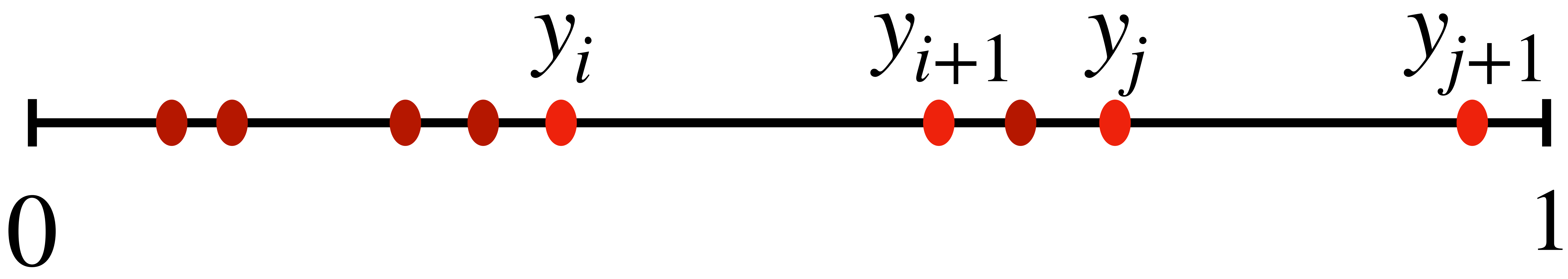
Math setting: Proof of Proposition 2

Need to construct y such that $d_\infty(x, Ay) = \Omega(\alpha^{1+\epsilon})$ for any similarity A.

Pick x_i, x_{i+1} such that $|x_i - x_{i+1}|$ are of order α. Then take $y_i = x_i - \beta, y_{i+1} = x_{i+1} + \beta$ for some β on the order $\alpha^{1+\epsilon}$.

This gives us that $d_\infty(x, y) \geq \beta$.

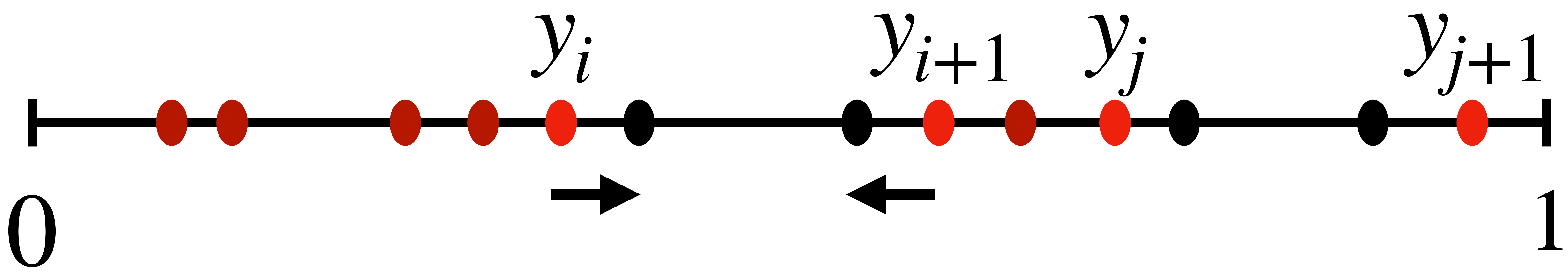
Pick x_j, x_{j+1} such that $|x_j - x_{j+1}|$ are of order α. Then take $y_j = x_j - \beta, y_{j+1} = x_{j+1} + \beta$. For every other $k \neq i, j, i+1, j+1$ we'll take $y_k = x_k$.



Math setting: Proof of Proposition 2

Need to construct y such that $d_\infty(x, Ay) = \Omega(\alpha^{1+\epsilon})$ for any similarity A.

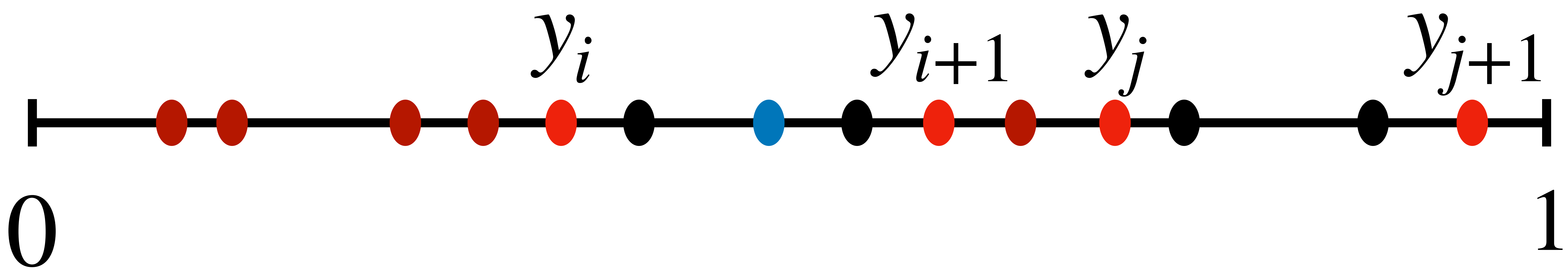
Given a similarity A of $\mathbb{R}$, if we have that $d_\infty(x, Ay) < \beta$, then we would need $Ay_i > y_i$ and $Ay_{i+1} < y_{i+1}$.



Math setting: Proof of Proposition 2

Need to construct y such that $d_\infty(x, Ay) = \Omega(\alpha^{1+\epsilon})$ for any similarity A.

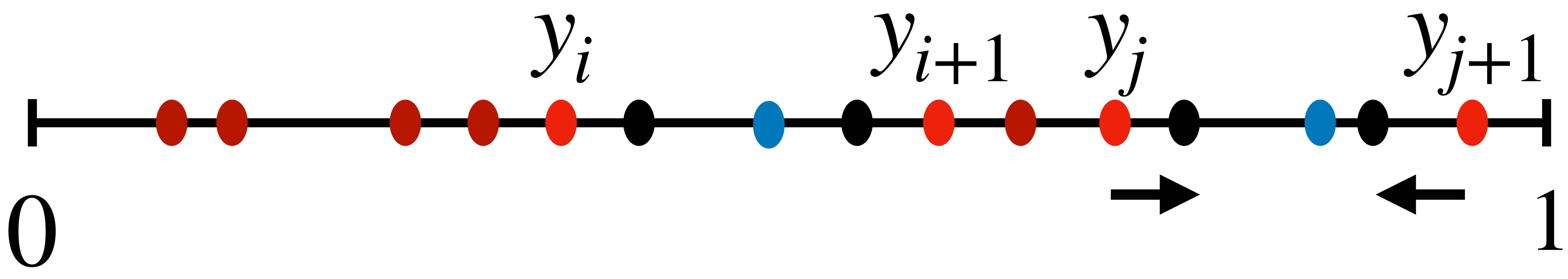
Given a similarity A of $\mathbb{R}$, if we have that $d_\infty(x, Ay) < \beta$, then we would need $Ay_i > y_i$ and $Ay_{i+1} < y_{i+1}$.
So A has a fixed point between x_i and x_{i+1}.



Math setting: Proof of Proposition 2

Need to construct y such that $d_\infty(x, Ay) = \Omega(\alpha^{1+\epsilon})$ for any similarity A.

Given a similarity A of $\mathbb{R}$, if we have that $d_\infty(x, Ay) < \beta$, then we would need $Ay_i > y_i$ and $Ay_{i+1} < y_{i+1}$. So A has a fixed point between x_i and x_{i+1}. But the same is true for y_j and y_{j+1}.



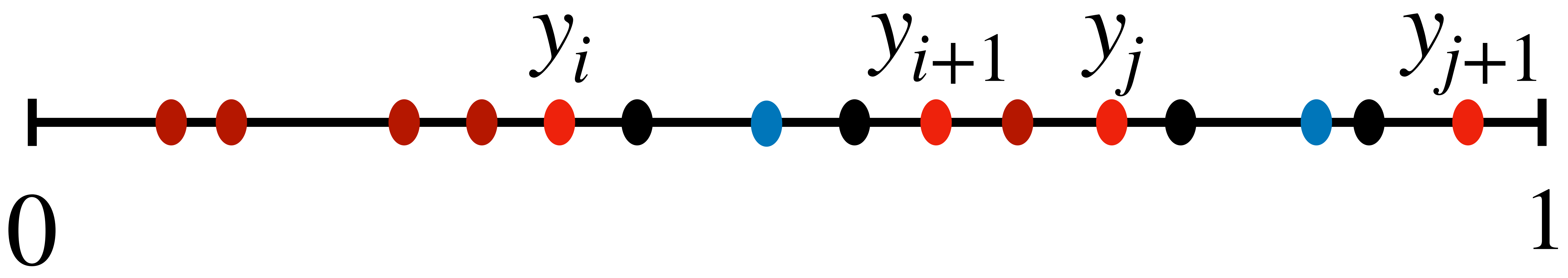
Math setting: Proof of Proposition 2

Need to construct y such that $d_\infty(x, Ay) = \Omega(\alpha^{1+\epsilon})$ for any similarity A.

Given a similarity A of $\mathbb{R}$, if we have that $d_\infty(x, Ay) < \beta$, then we would need $Ay_i > y_i$ and $Ay_{i+1} < y_{i+1}$. So A has a fixed point between x_i and x_{i+1}. But the same is true for y_j and y_{j+1}.

By contradiction, we have that $d_\infty(x, Ay) \geq \beta = \Omega(\alpha^{1+\epsilon})$

□



Math setting: What do we know about higher dimensions?

Math setting: What do we know about higher dimensions?

Theorem [Arias-Castro 2015]

Let U be a bounded, connected, open domain in $\mathbb{R}^d$, x is a tuple such that $\delta_H(x, U) \leq \alpha$, and y is weakly isotonic to x, then for some similarity A of $\mathbb{R}^d$, we have

$$d_\infty(x, Ay) = O(\alpha^{\frac{1}{2}})$$

Math setting: What do we know about higher dimensions?

Theorem [Arias-Castro 2015]

Let U be a bounded, connected, open domain in $\mathbb{R}^d$, x is a tuple such that $\delta_H(x, U) \leq \alpha$, and y is weakly isotonic to x, then for some similarity A of $\mathbb{R}^d$, we have

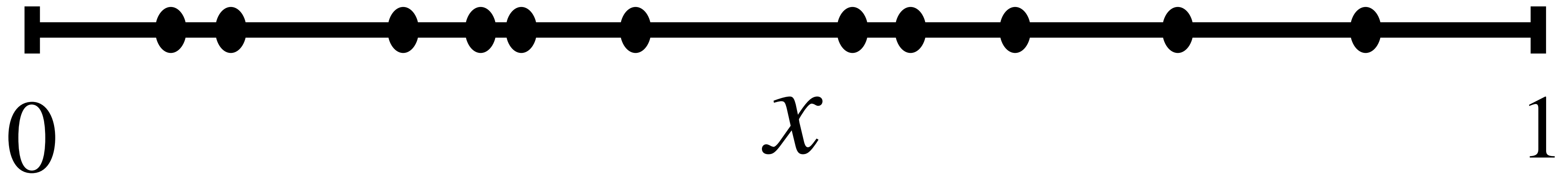
$$d_\infty(x, Ay) = O(\alpha^{\frac{1}{2}})$$

Theorem [J. Ellenberg, L. Jain 2019]

Let $x = (x_1, \ldots, x_n) \subset [0,1]^d$. For $y = (y_1, \ldots, y_n)$ be a subset of $\mathbb{R}^d$ where the y_i are chosen uniformly at random from the Euclidean ball of size $\beta > n^{-1}$ around x_i. Then the probability that y is isotonic to x is bounded above by $\exp(-cn)$ for some constant $c > 0$.

Math setting: What do we know about higher dimensions?

Proposition 2 [J. Ellenberg, L. Jain, 2019]



But what do we need to extend proposition 2 to higher dimensions?

Math setting: Proof of Proposition 2

Theorem [Graham, Ron 2006]

For every positive integer k, there exists a subset S of $\mathbb{Z}$ such that

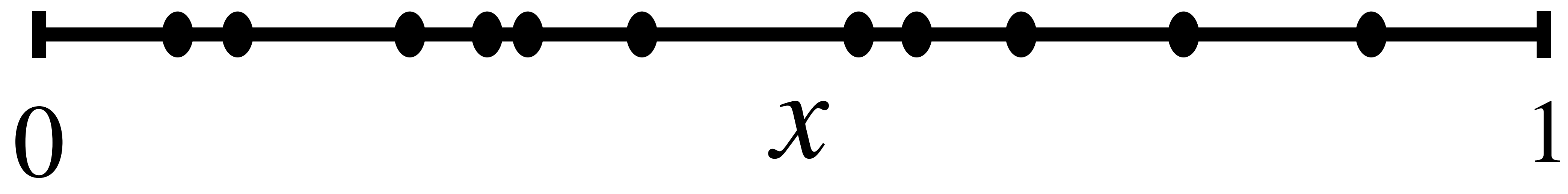
- $S \subset [1,M]$ with $M \geq k^{c \log k}$ for some absolute constant c
- S has no 3 terms in arithmetic progression
- S has no gaps between successive terms of size greater than k

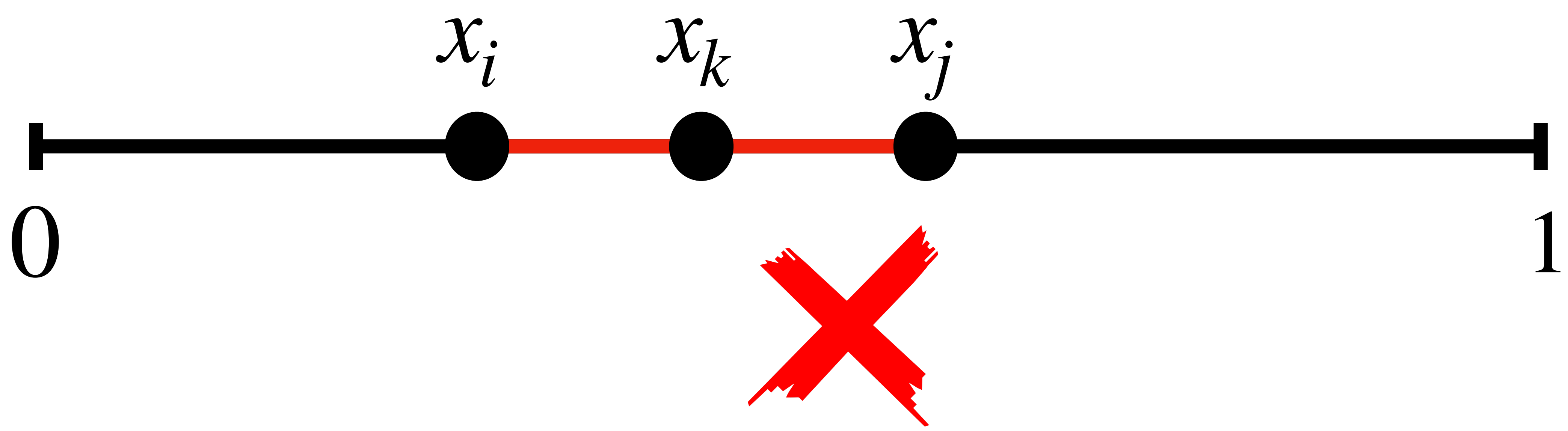
Take $x = (x_1, \ldots, x_{|S|})$ be the set of points $\{s/M : s \in S\} \subset [0,1]$. With $\alpha = k/2M$, we have that

$$\delta_H(x, [0,1]) \leq \alpha$$

Given S has no 3-term arithmetic progression, for each triplet x_i, x_j, x_k we have

$$2x_k - x_i - x_j \neq 0$$





Math setting: Insights from the proof

Theorem [Graham, Ron 2006]

For every positive integer k, there exists a subset S of $\mathbb{Z}$ such that

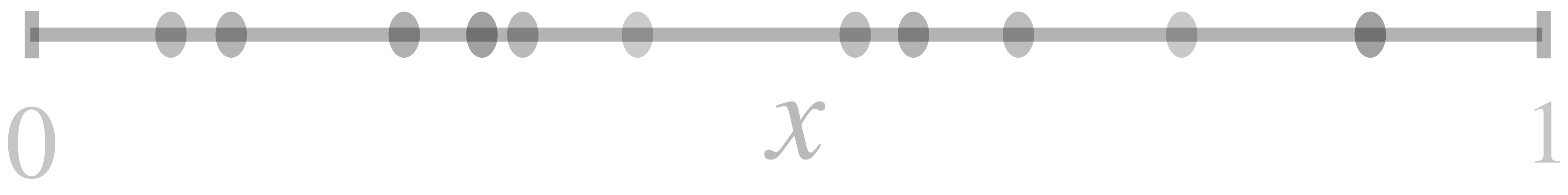
- $S \subset [1,M]$ with $M \geq k^{c \log k}$ for some absolute constant c
- S has no 3 terms in arithmetic progression
- S has no gaps between successive terms of size greater than k

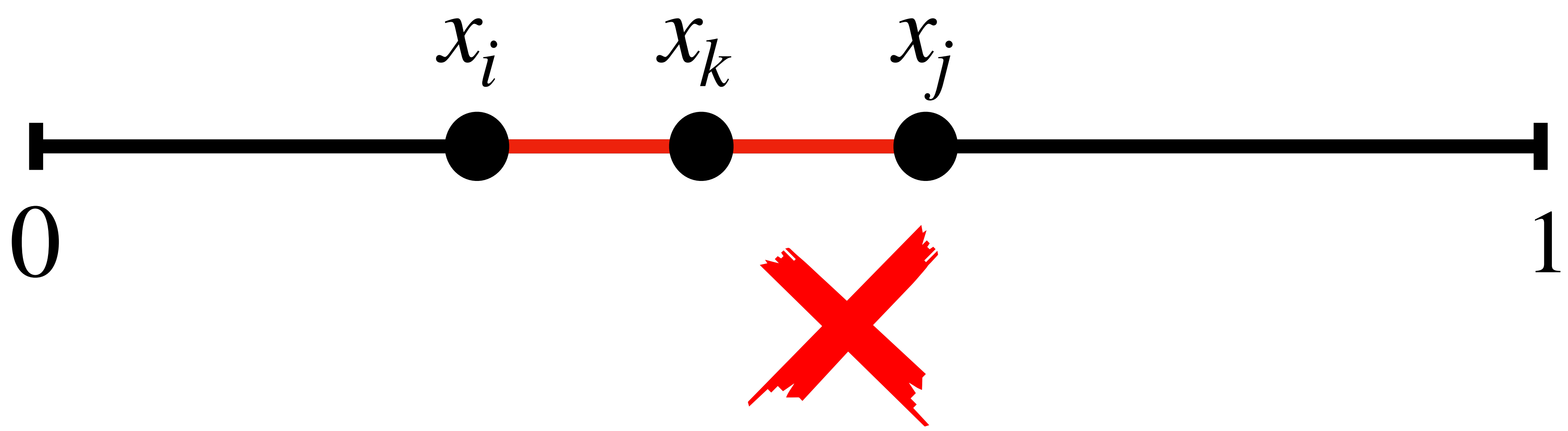
Take $x = (x_1, ..., x_{|S|})$ be the set of points $\{s/M : s \in S\} \subset [0,1]$. With $\alpha = k/2M$, we have that

$$\delta_H(x, [0,1]) \leq \alpha$$

Given S has no 3-term arithmetic progression, for each triplet x_i, x_j, x_k we have

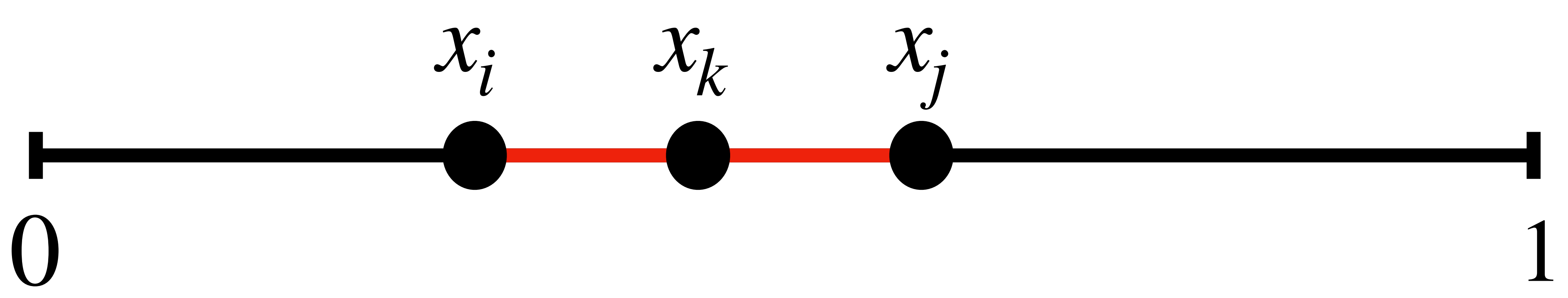
$$2x_k - x_i - x_j \neq 0$$





Math setting: Insights from the proof

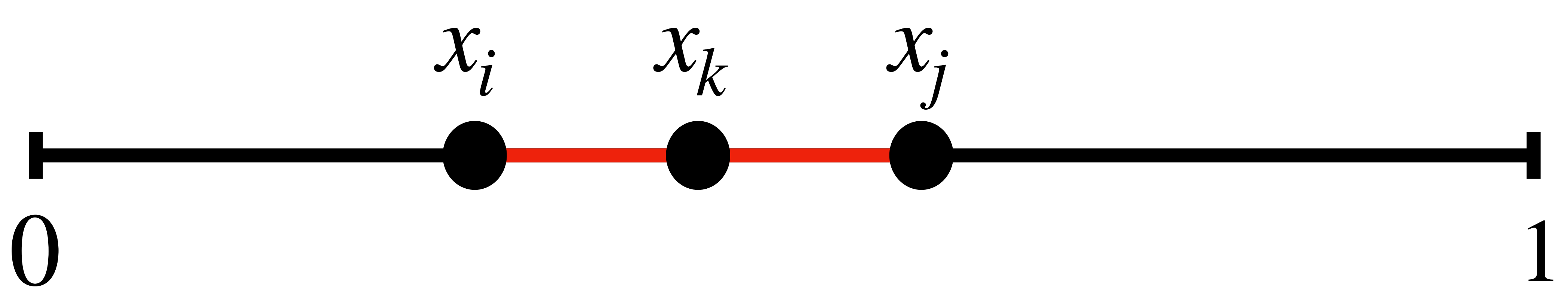
Case: $M = [0,1]$



Needed large set with no 3 term arithmetic progression

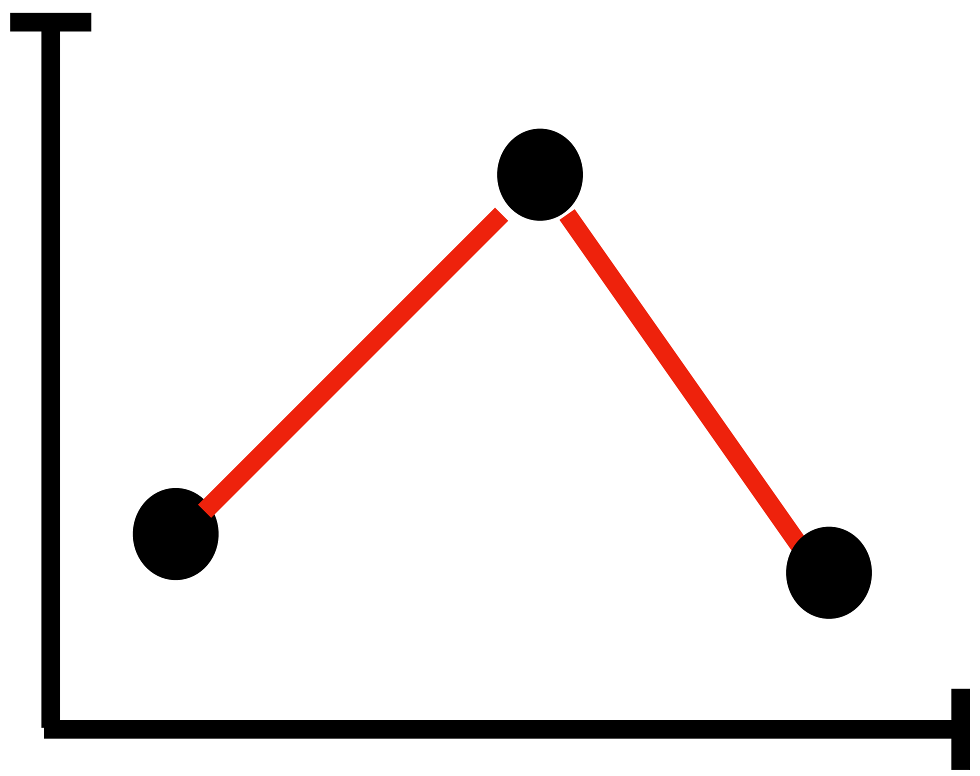
Math setting: Insights from the proof

Case: $M = [0,1]$



Needed large set with no 3 term arithmetic progression

Case: $M = [0,1]^d$



Needed large set with no 3 points forming an isosceles triangle

Question: What's the size of the largest subset S of an NxN integer lattice?

Overview

Mathematical Motivation and Background

- Motivation: Non Metric Multidimensional Scaling
- Key definitions and propositions
- Known bounds for the problem

How Reinforcement Learning can help

- Reinforcement learning background and main algorithm
- Current results and observations
- Next Steps

Current known bounds: Lower Bound

Theorem [A. Wagner 2023]

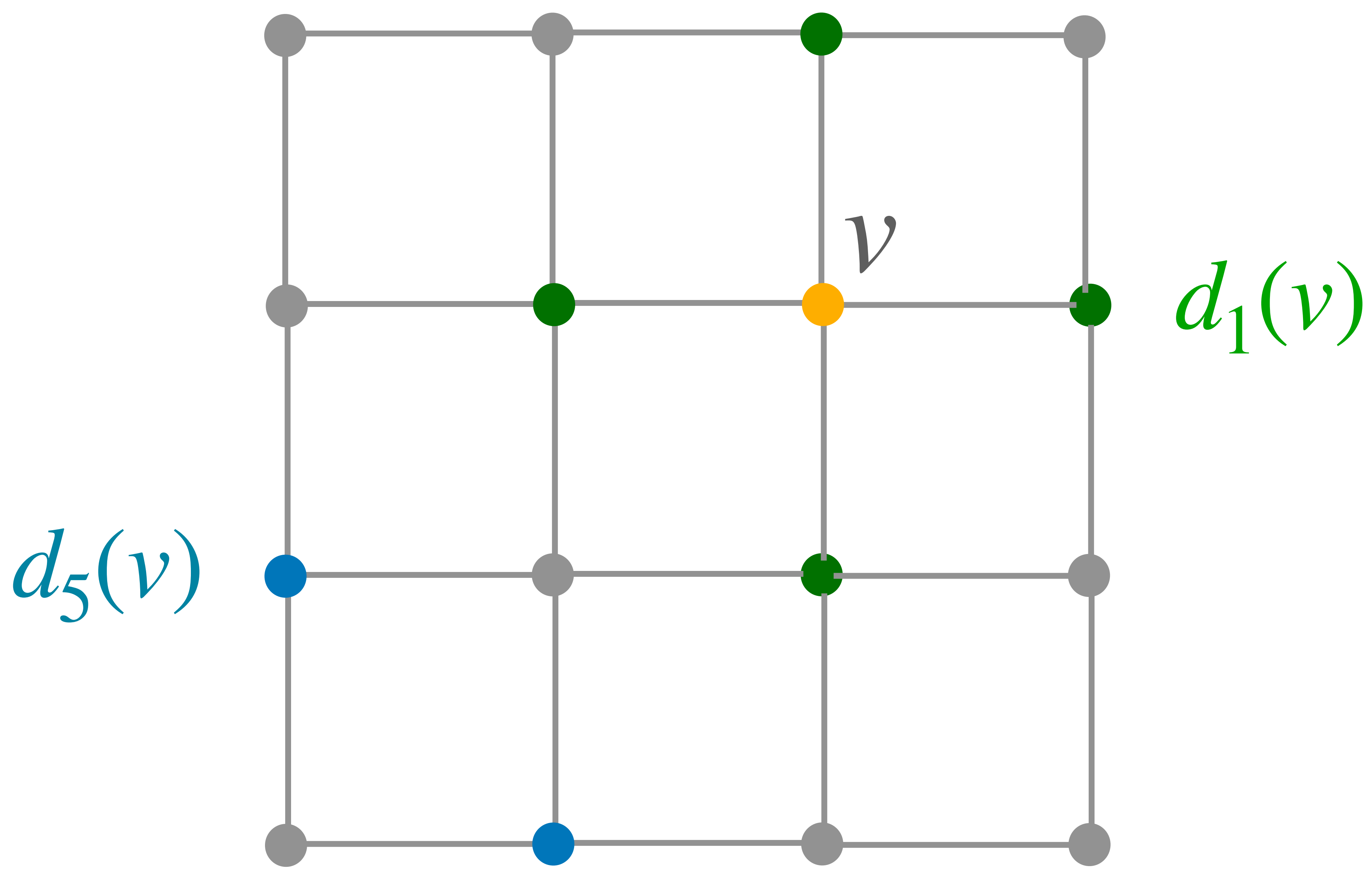
Let S be the largest subset of an NxN lattice that contains no isosceles triangles, then we have that

$$|S| = \Omega(\frac{N}{\sqrt{\log N}})$$

Current known bounds: Lower Bound

Proof:

Let $v \in N$xN grid and $d_k(v)$ is the set of points at a distance k from v.

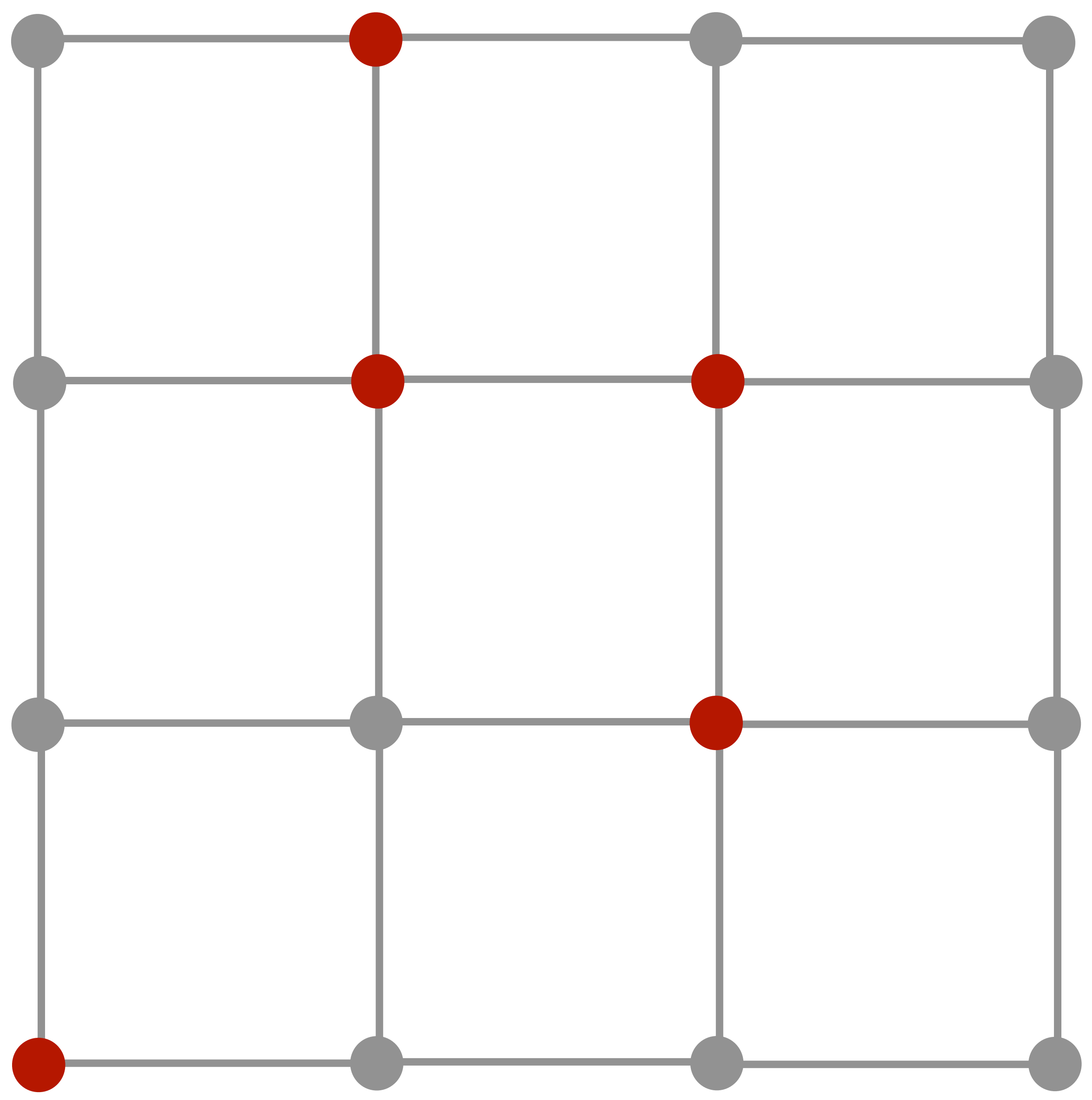


Current known bounds: Lower Bound

Proof:

Let $v \in N$xN grid and $d_k(v)$ is the set of points at a distance k from v.

Let A be a random subset of NxN grid with each point picked with probability p.

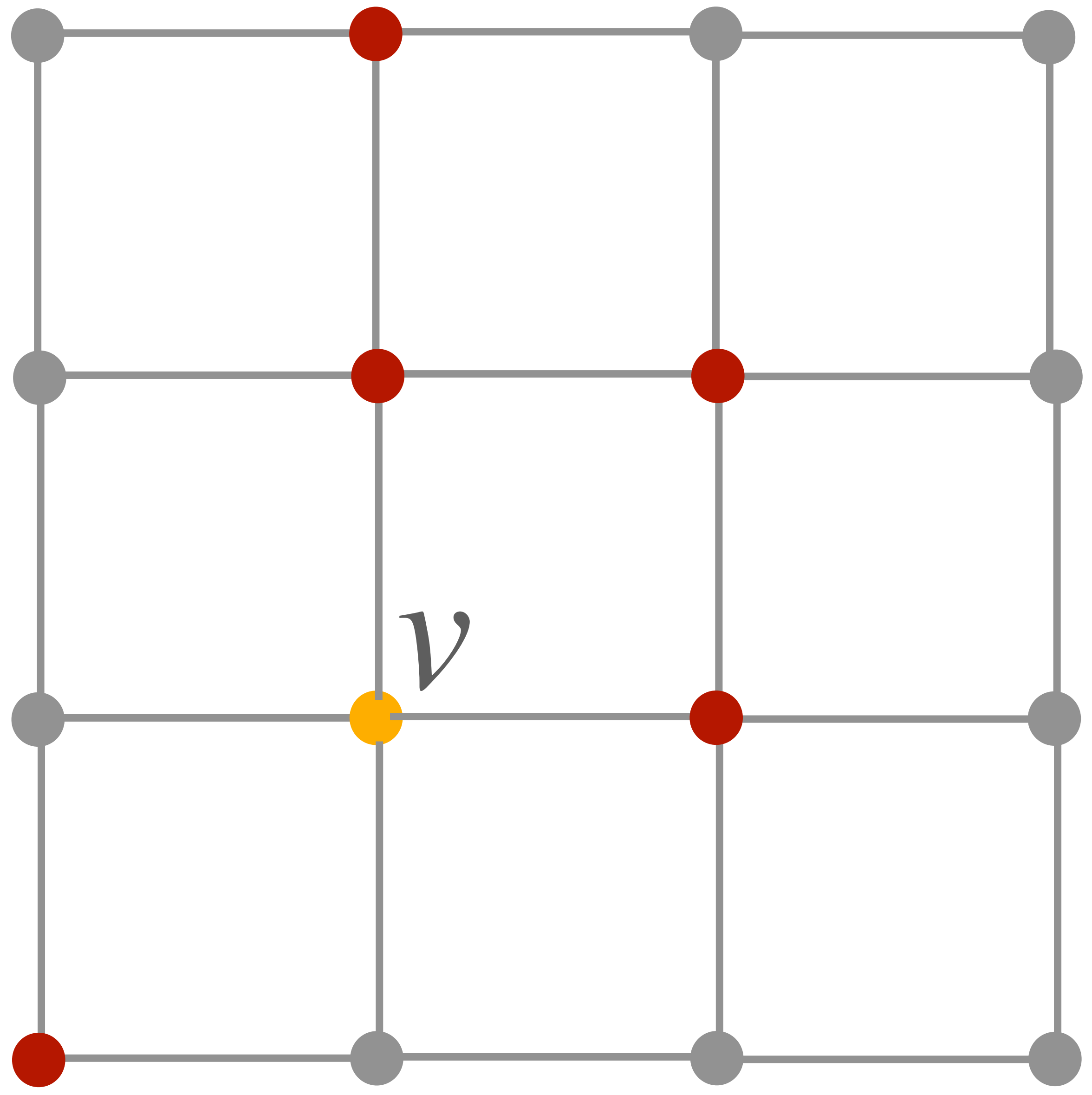


Current known bounds: Lower Bound

Proof:

Let $v \in N$xN grid and $d_k(v)$ is the set of points at a distance k from v.

Let A be a random subset of NxN grid with each point picked with probability p. Then, for a fixed vertex v and integer k,



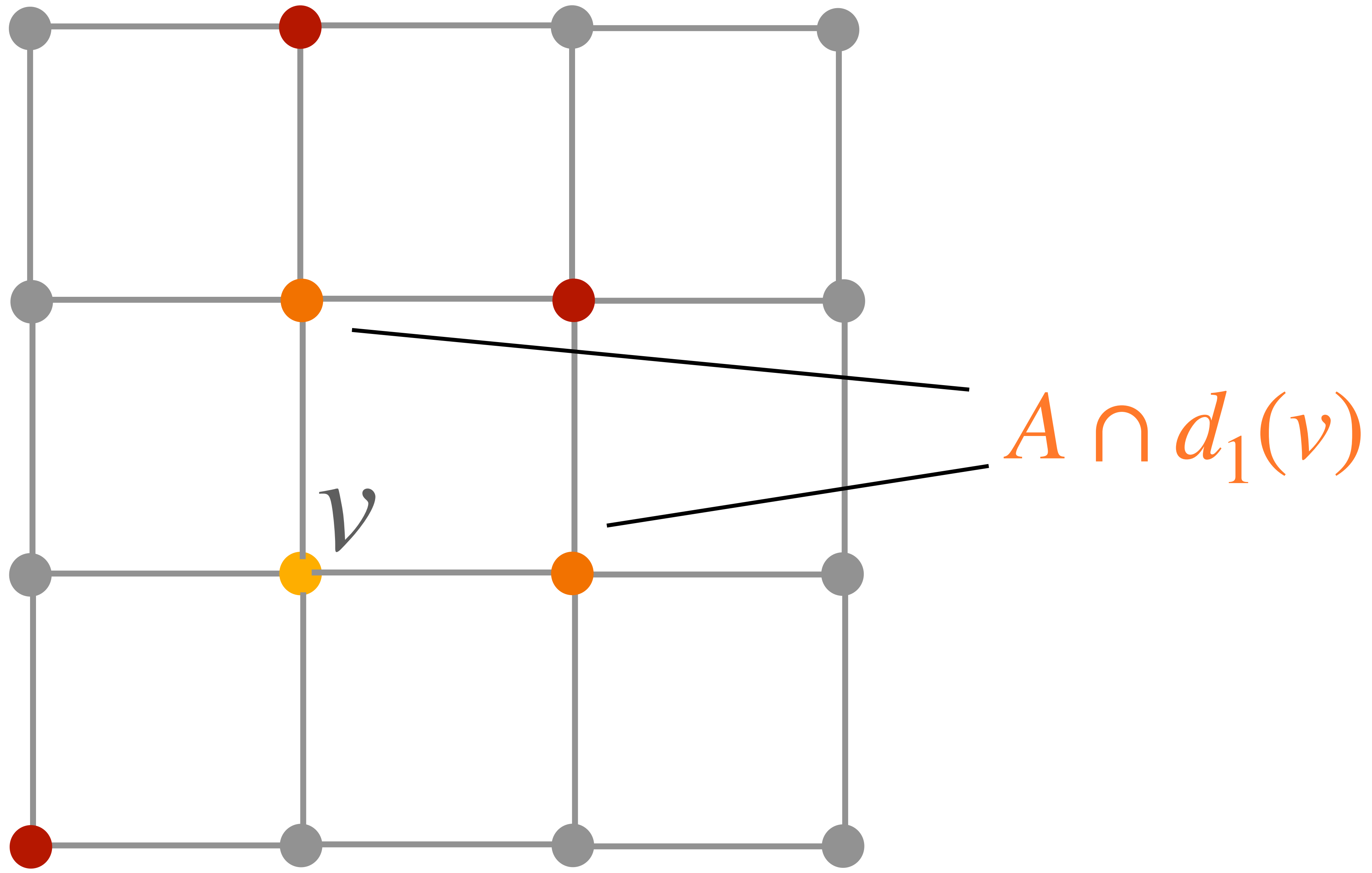
Current known bounds: Lower Bound

Proof:

Let $v \in N\text{x}N$ grid and $d_k(v)$ is the set of points at a distance k from v.

Let A be a random subset of $N\text{x}N$ grid with each point picked with probability p. Then, for a fixed vertex v and integer k,

$$P(|A \cap d_k(v)| \geq 2)$$



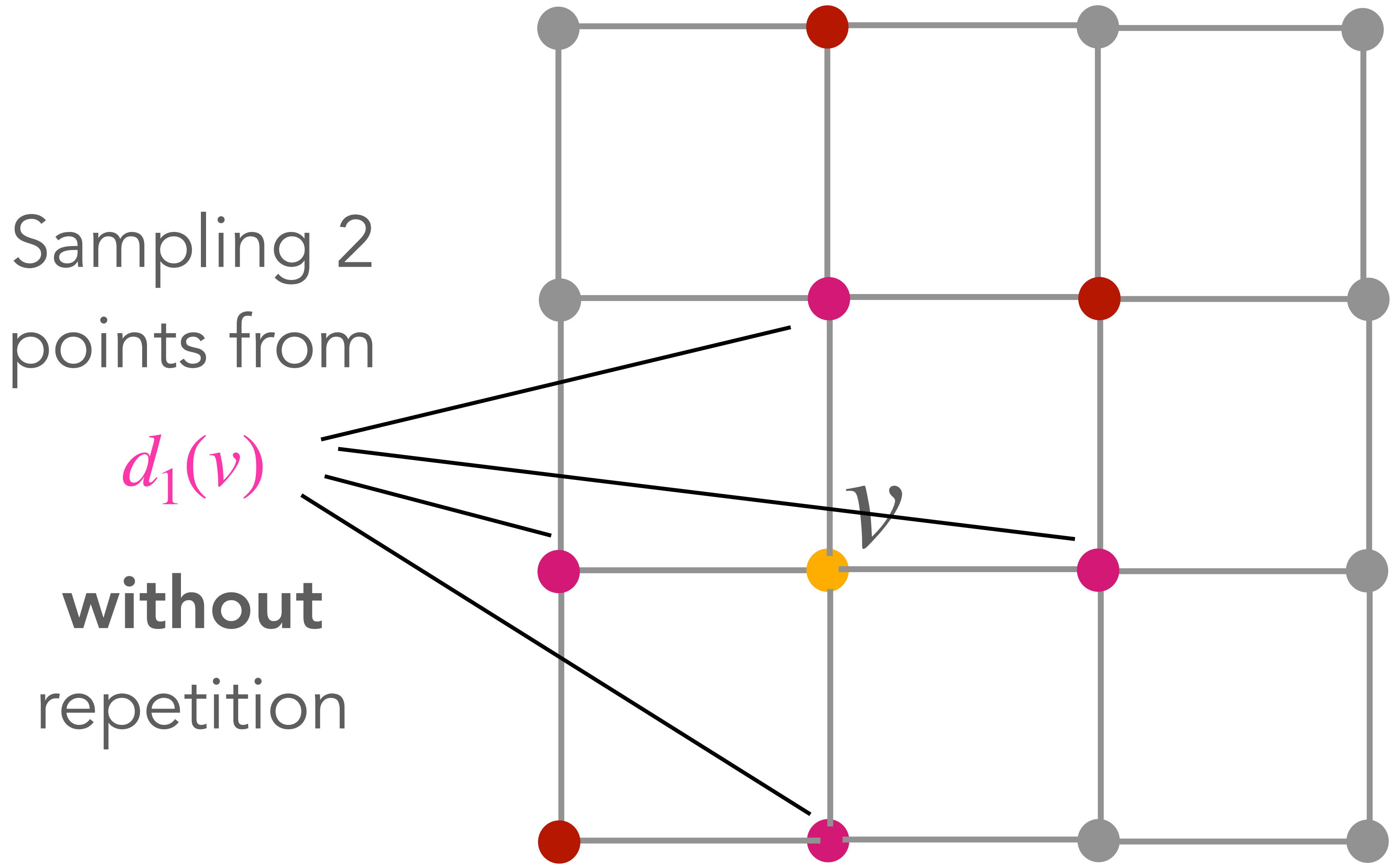
Current known bounds: Lower Bound

Proof:

Let $v \in N$xN grid and $d_k(v)$ is the set of points at a distance k from v.

Let A be a random subset of NxN grid with each point picked with probability p. Then, for a fixed vertex v and integer k,

$$P(|A \cap d_k(v)| \geq 2) \leq \binom{|d_k(v)|}{2} p^2$$



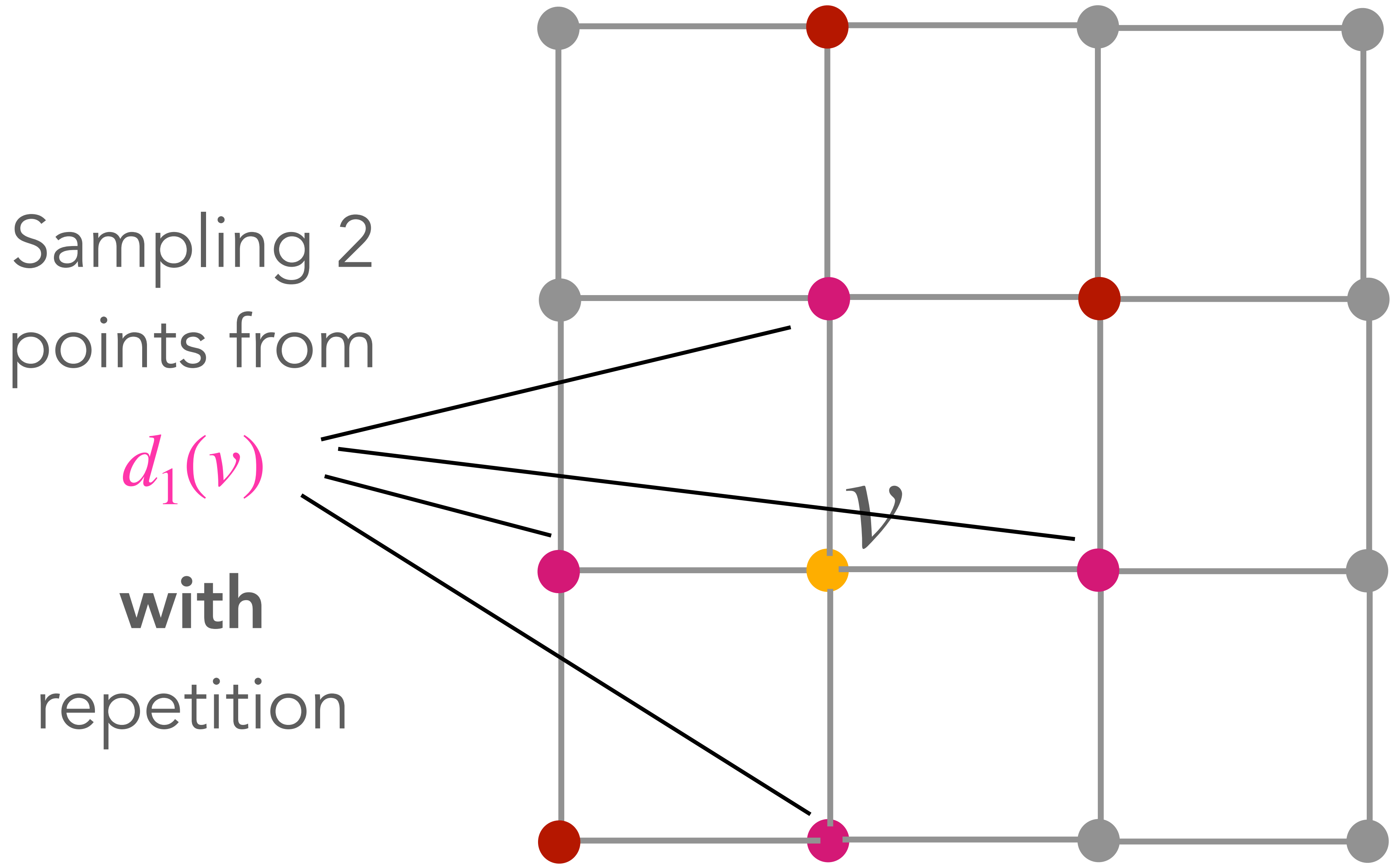
Current known bounds: Lower Bound

Proof:

Let $v \in N$xN grid and $d_k(v)$ is the set of points at a distance k from v.

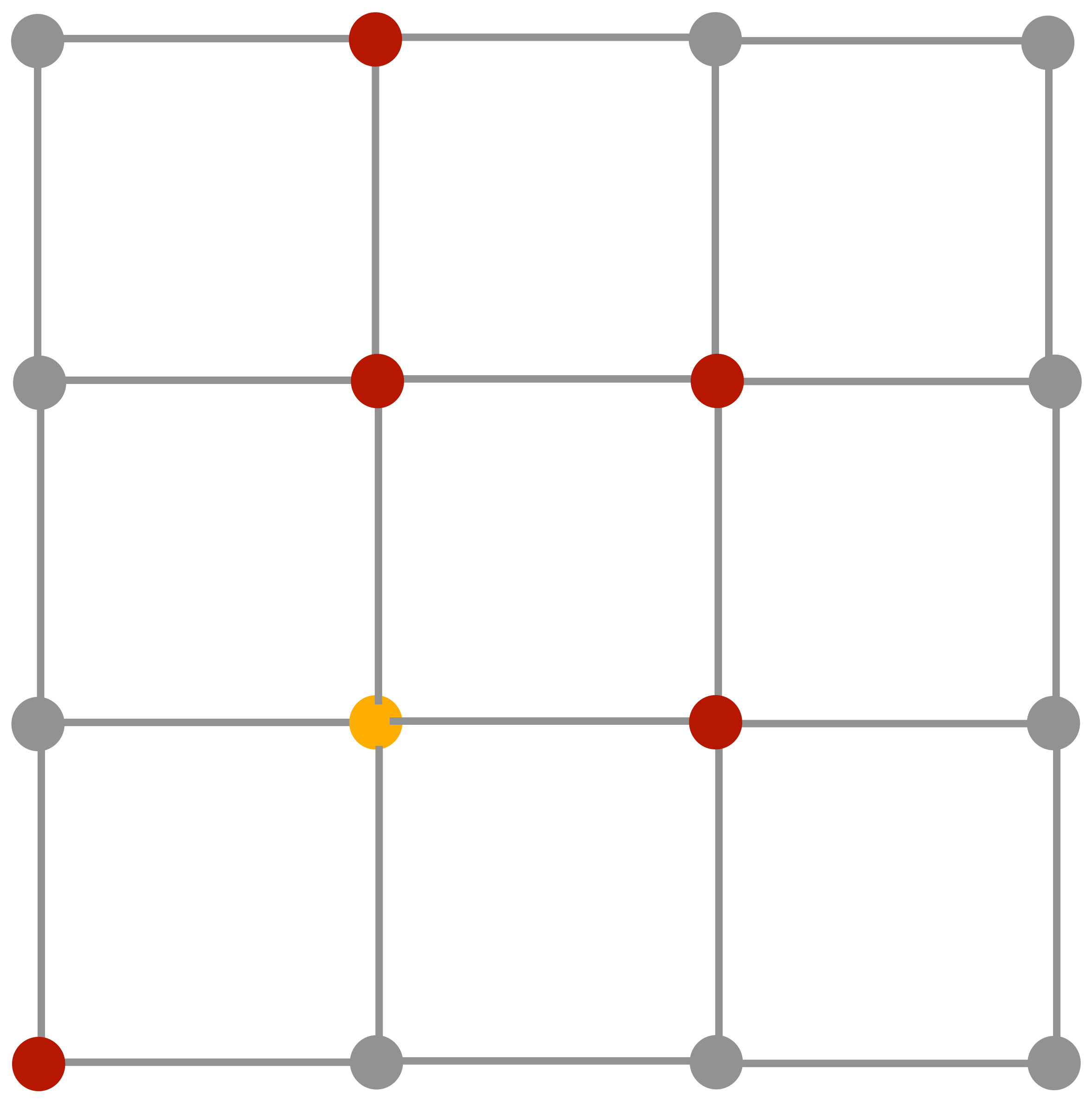
Let A be a random subset of NxN grid with each point picked with probability p. Then, for a fixed vertex v and integer k,

$$P(|A \cap d_k(v)| \geq 2) \leq \binom{|d_k(v)|}{2} p^2 \leq |d_k(v)|^2 p^2$$



Current known bounds: Lower Bound

$$P(|A \cap d_k(v)| \geq 2) \leq \binom{|d_k(v)|}{2} p^2 \leq |d_k(v)|^2 p^2$$

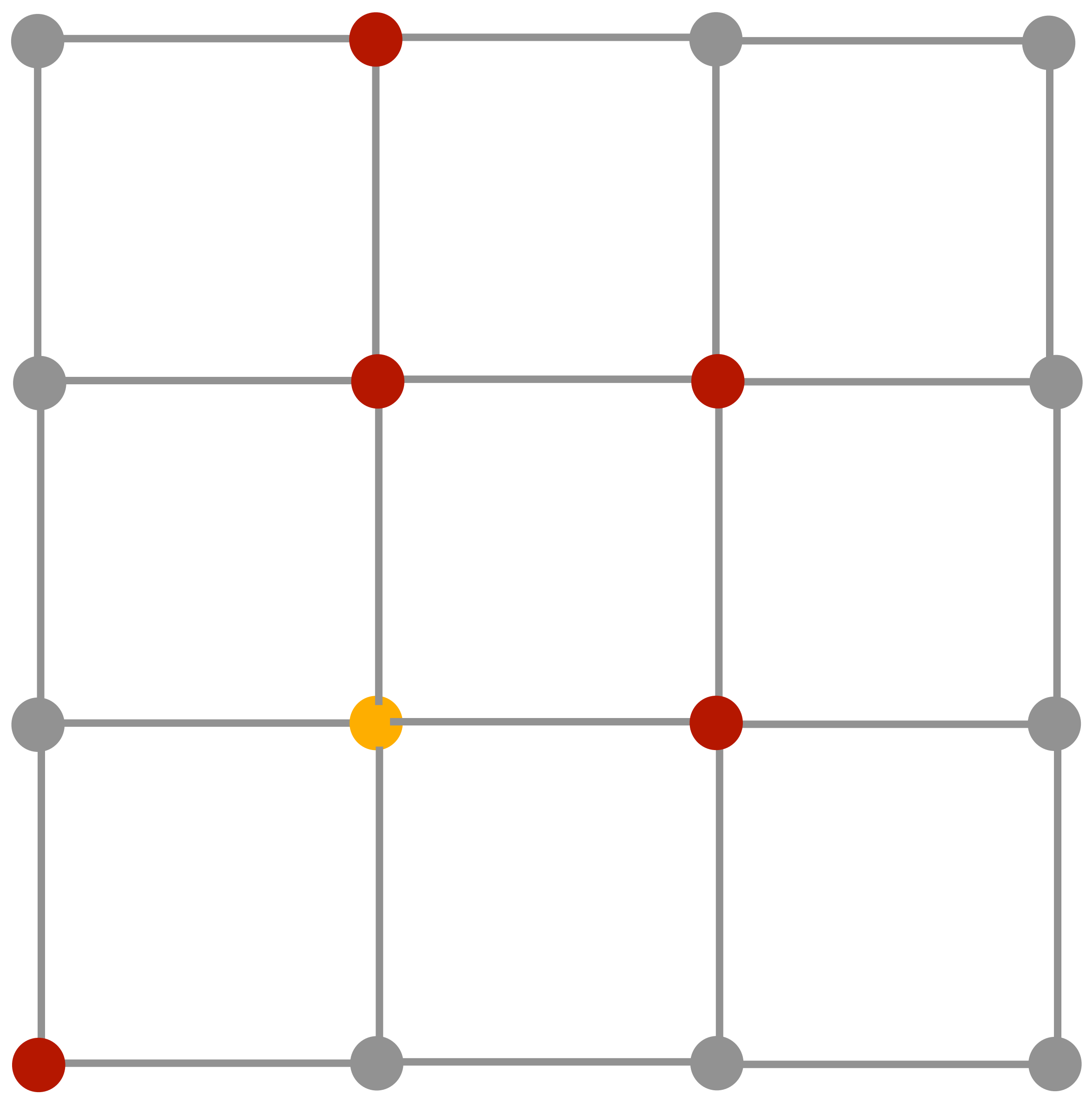


Current known bounds: Lower Bound

$$P(|A \cap d_k(v)| \geq 2) \leq \binom{|d_k(v)|}{2} p^2 \leq |d_k(v)|^2 p^2$$

Fact: $|d_k(v)|$ is bounded above by $r_2(k)$, the sum of squares function. So we get that for **fixed** v, k:

$$P(|A \cap d_k(v)| \geq 2) \leq r_2(k)^2 p^2$$



Current known bounds: Lower Bound

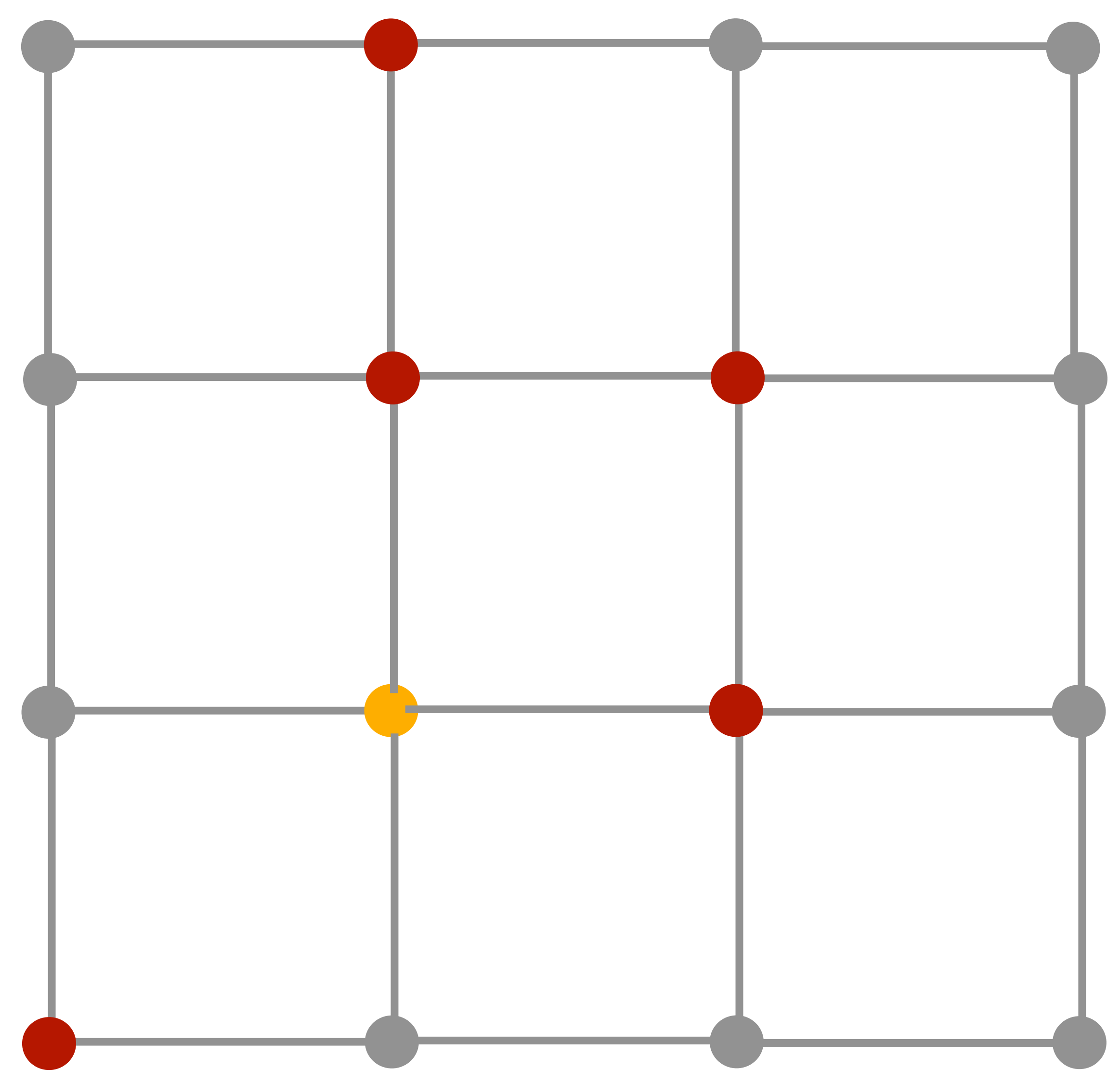
$$P(|A \cap d_k(v)| \geq 2) \leq \binom{|d_k(v)|}{2} p^2 \leq |d_k(v)|^2 p^2$$

Fact: $|d_k(v)|$ is bounded above by $r_2(k)$, the sum of squares function. So we get that for **fixed** v, k:

$$P(|A \cap d_k(v)| \geq 2) \leq r_2(k)^2 p^2$$

Now we just fix v. Then,

$$P(\exists k : |A \cap d_k(v)| \geq 2)$$



Current known bounds: Lower Bound

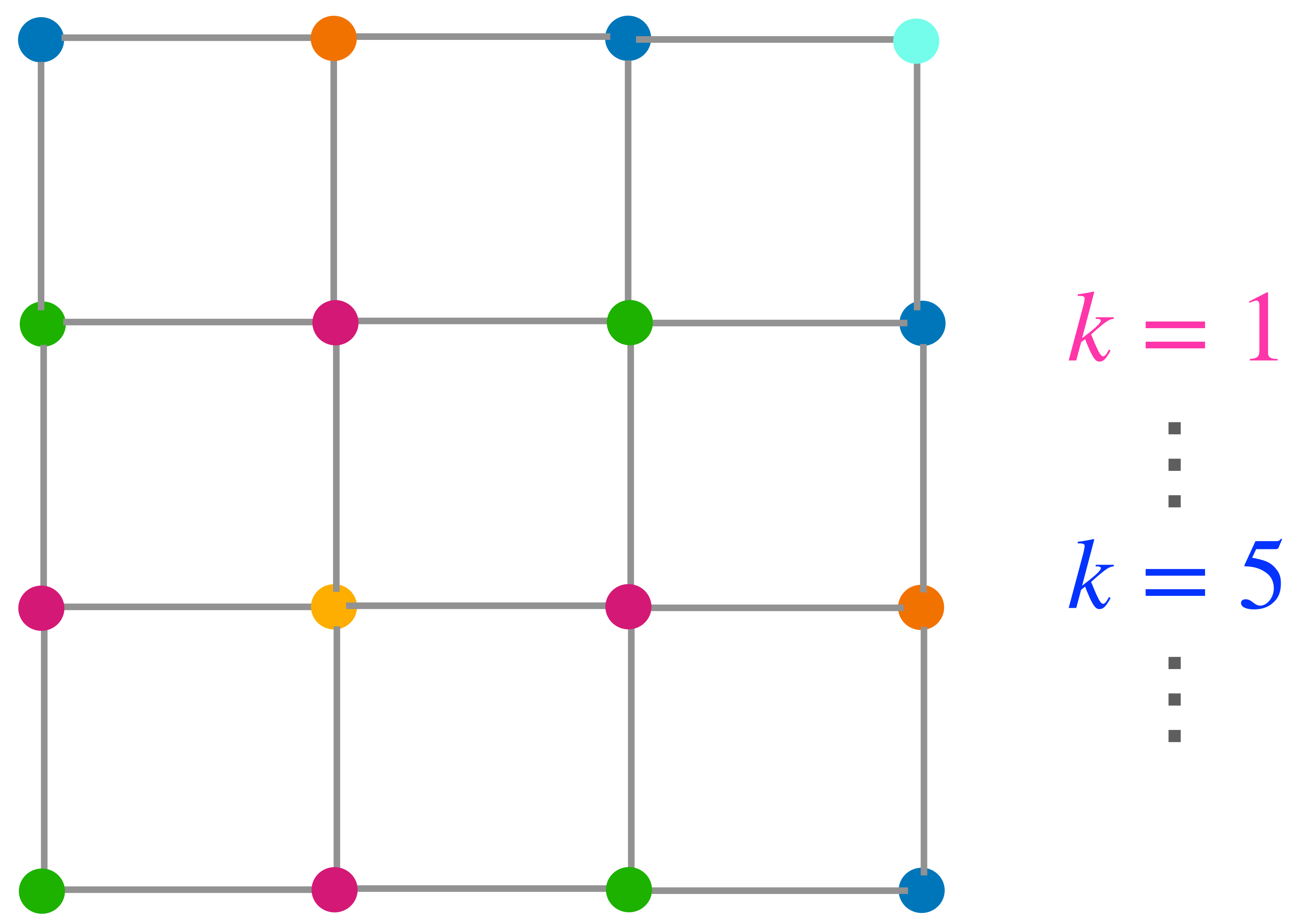
$$P(|A \cap d_k(v)| \geq 2) \leq \binom{|d_k(v)|}{2} p^2 \leq |d_k(v)|^2 p^2$$

Fact: $|d_k(v)|$ is bounded above by $r_2(k)$, the sum of squares function. So we get that for **fixed** v, k:

$$P(|A \cap d_k(v)| \geq 2) \leq r_2(k)^2 p^2$$

Now we just fix v. Then,

$$P(\exists k : |A \cap d_k(v)| \geq 2) \leq \sum_{k=1}^{2N^2} P(|A \cap d_k(v)| \geq 2)$$



Current known bounds: Lower Bound

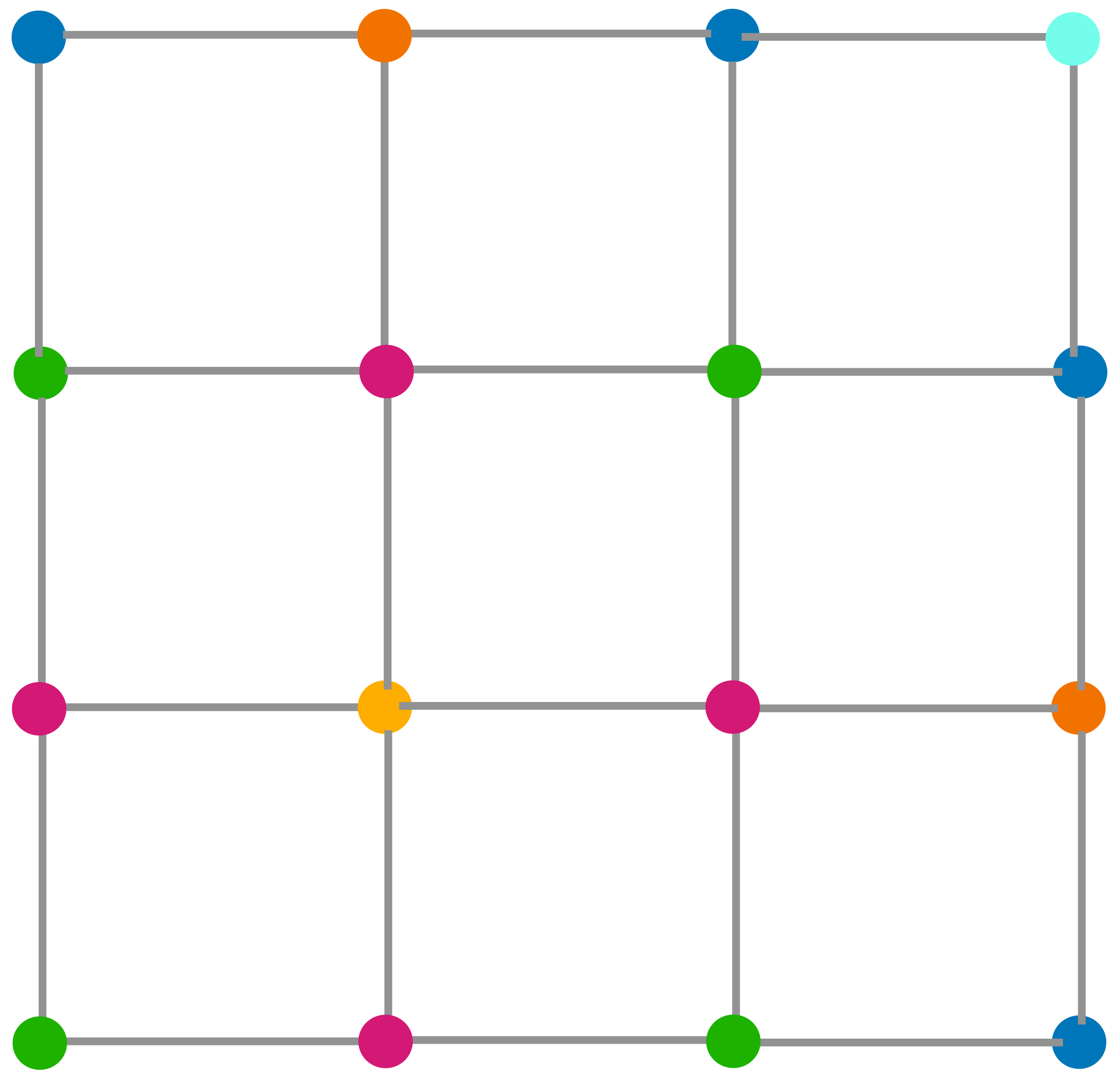
$$P(|A \cap d_k(v)| \geq 2) \leq \binom{|d_k(v)|}{2} p^2 \leq |d_k(v)|^2 p^2$$

Fact: $|d_k(v)|$ is bounded above by $r_2(k)$, the sum of squares function. So we get that for **fixed** v, k:

$$P(|A \cap d_k(v)| \geq 2) \leq r_2(k)^2 p^2$$

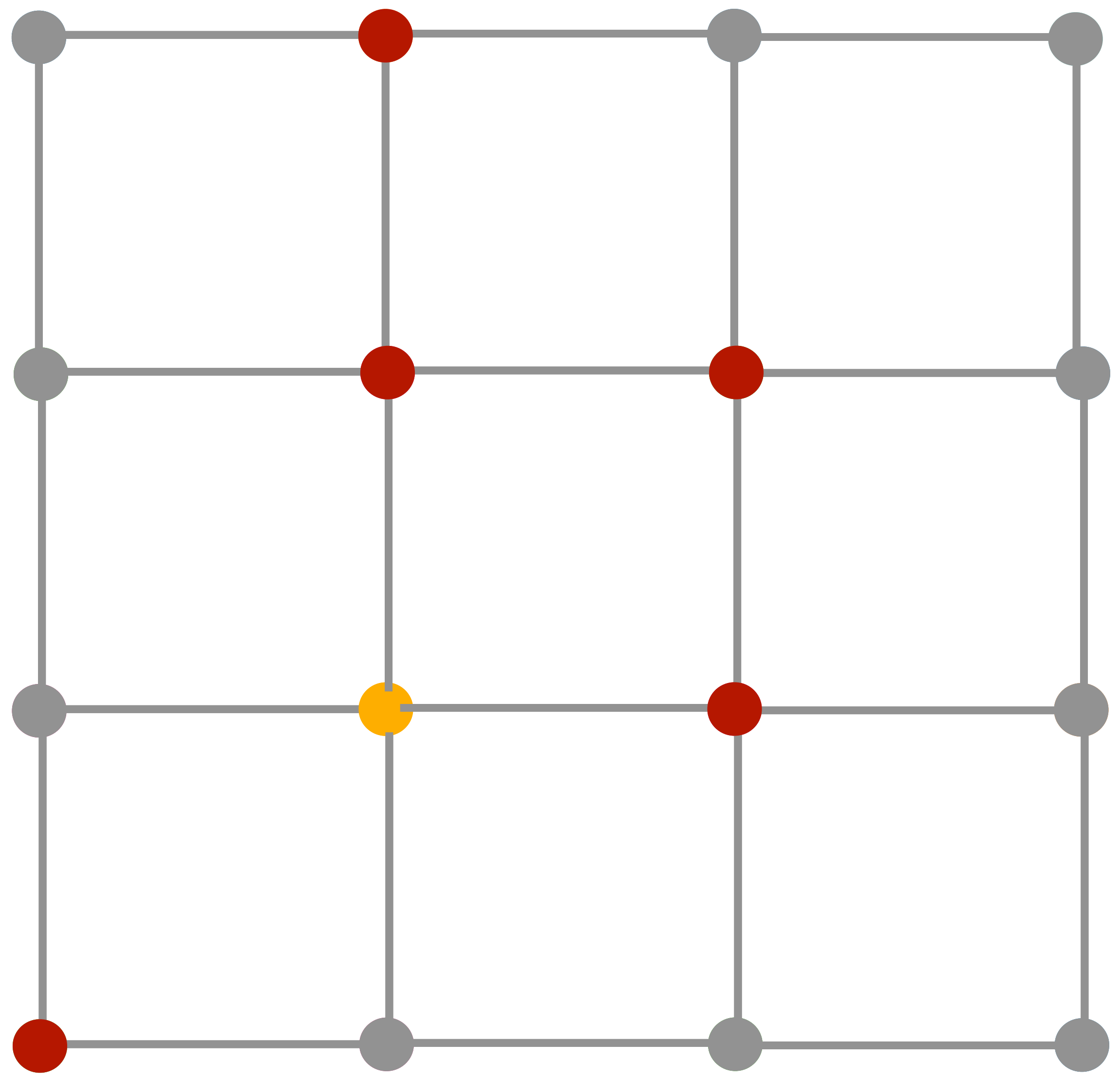
Now we just fix v. Then,

$$P(\exists k : |A \cap d_k(v)| \geq 2) \leq \sum_{k=1}^{2N^2} P(|A \cap d_k(v)| \geq 2) \leq p^2 \sum_{k=1}^{2N^2} r_2(k)^2$$



Current known bounds: Lower Bound

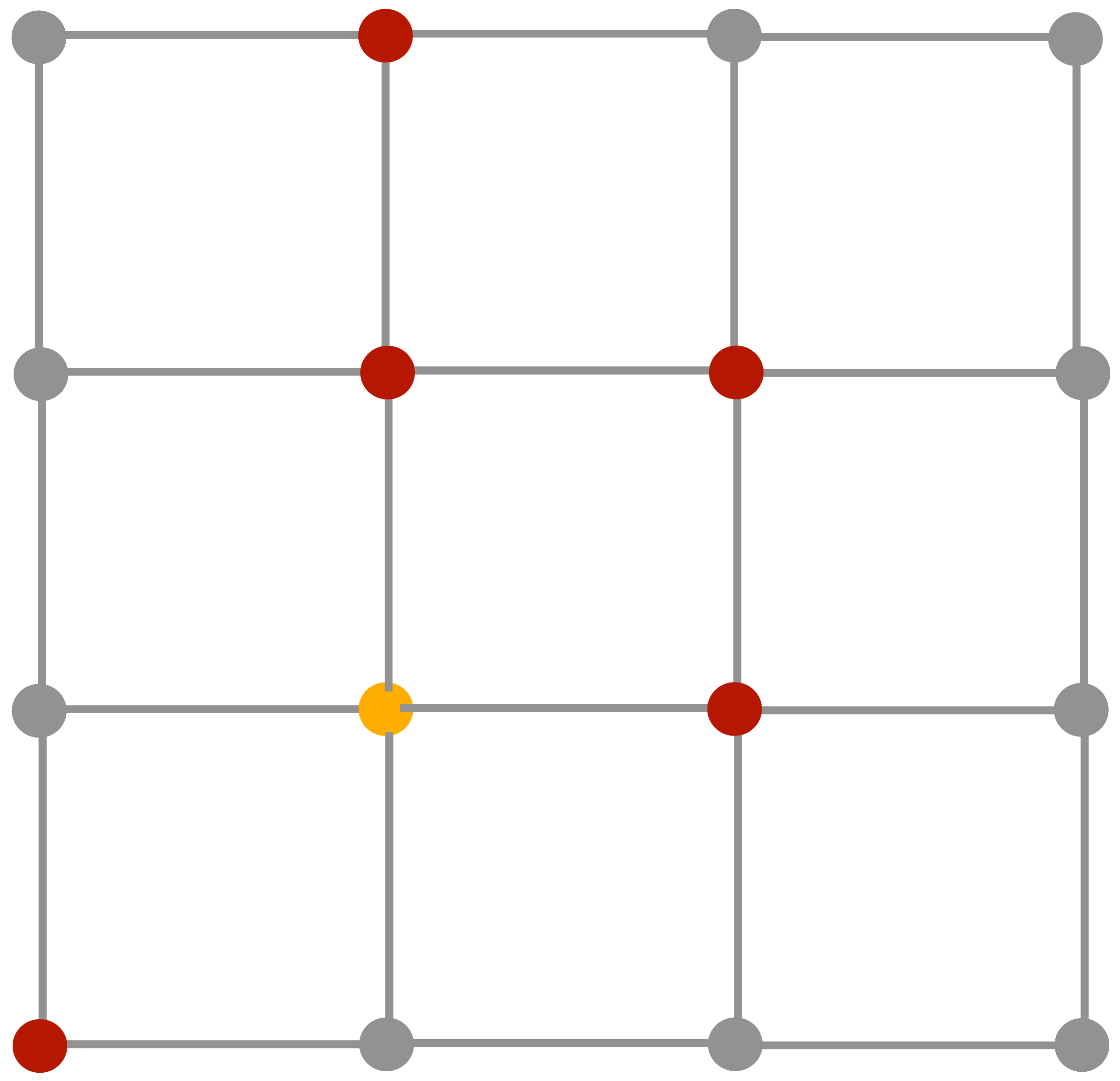
$$P(\exists k : |A \cap d_k(v)| \geq 2) \leq p^2 \sum_{k=1}^{2N^2} r_2(k)^2$$



Current known bounds: Lower Bound

$$P(\exists k : |A \cap d_k(v)| \geq 2) \leq p^2 \sum_{k=1}^{2N^2} r_2(k)^2$$

Fact: $\sum_{k \leq N} r_2(k)^m \leq C_m N (\log N)^{2^{m-1}-1}$ for a constant C_m



Current known bounds: Lower Bound

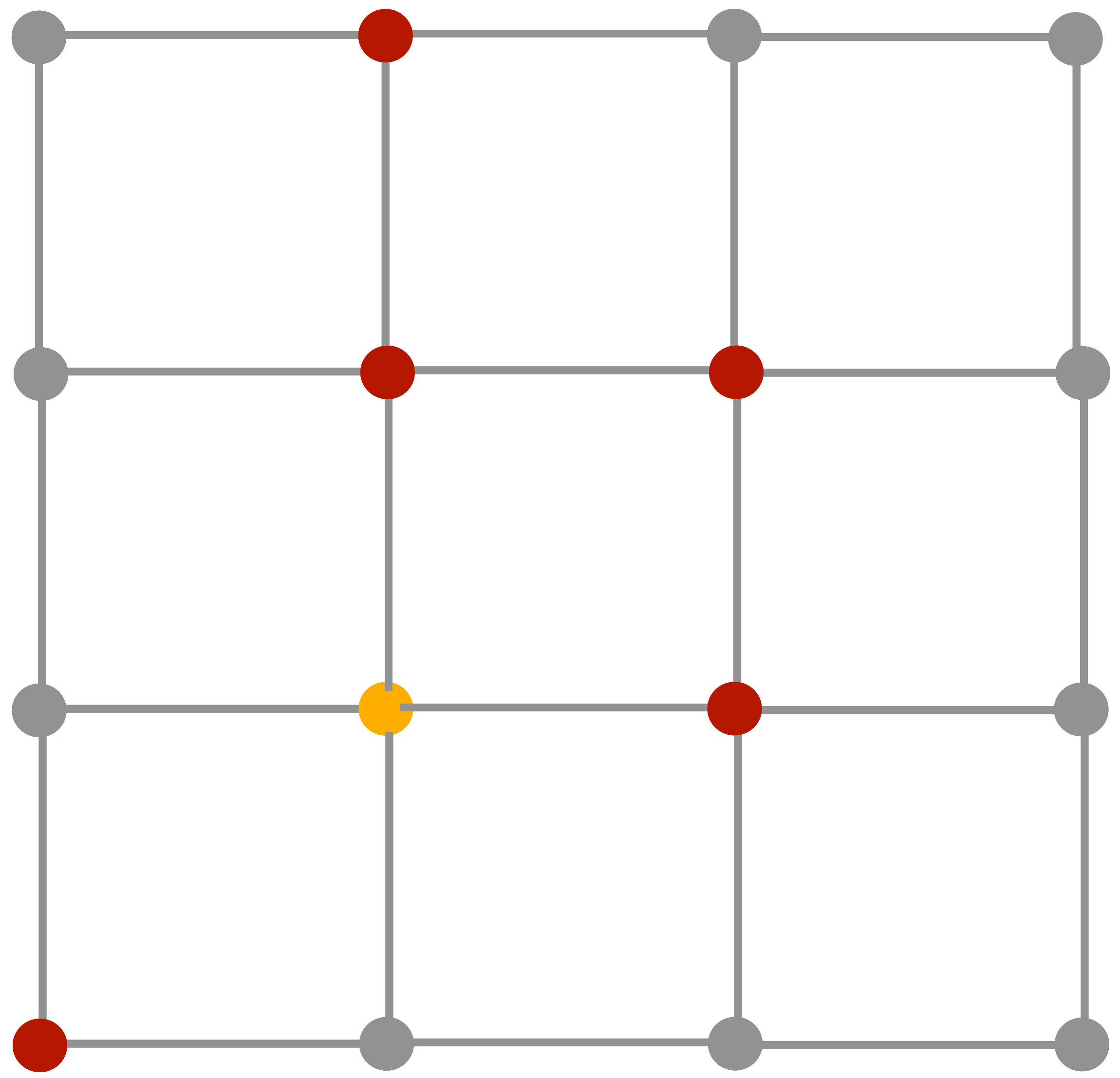
$$P(\exists k : |A \cap d_k(v)| \geq 2) \leq p^2 \sum_{k=1}^{2N^2} r_2(k)^2$$

Fact: $\sum_{k \leq N} r_2(k)^m \leq C_m N(\log N)^{2^{m-1}-1}$ for a constant C_m

So we get for **fixed** v,

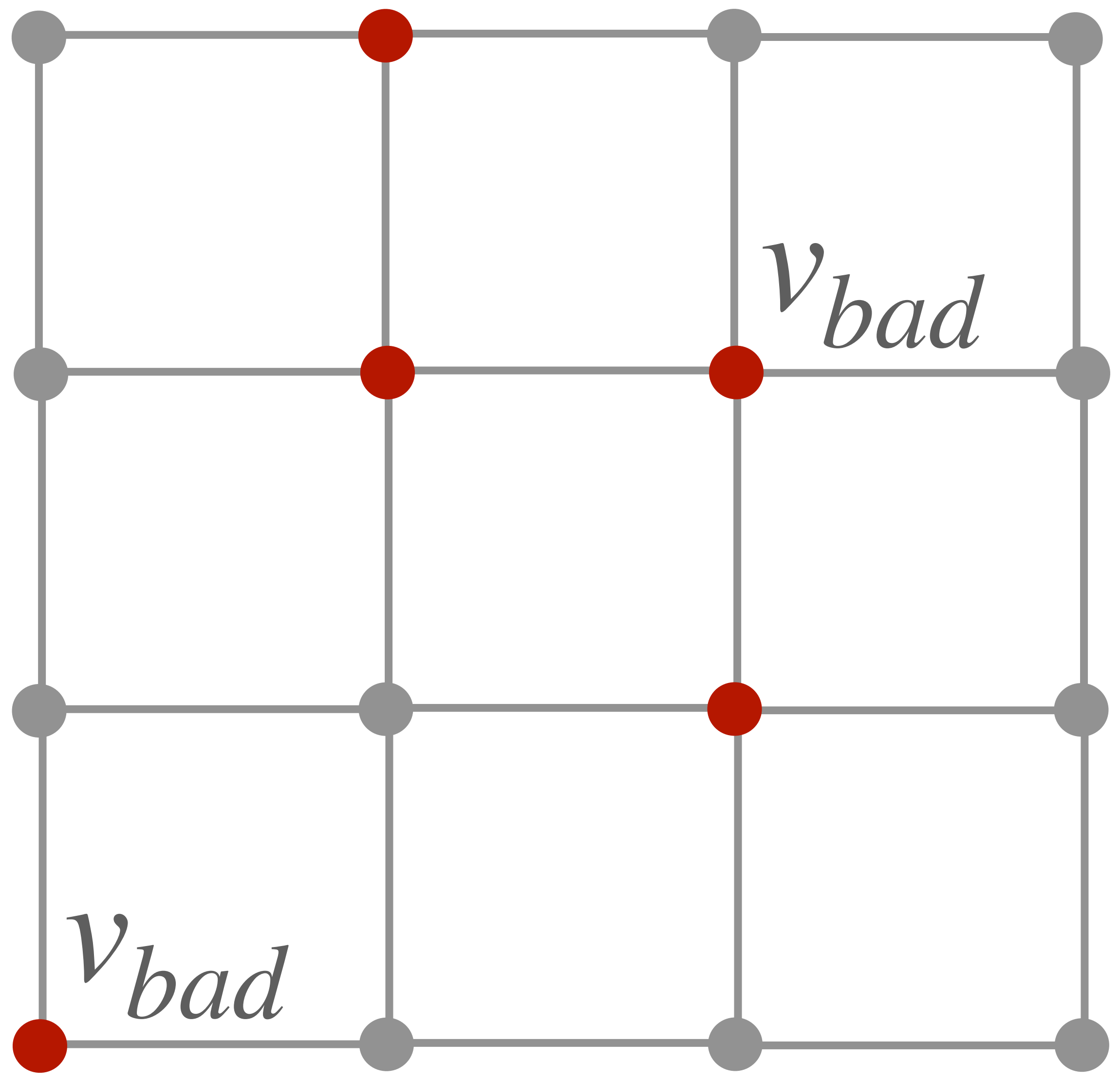
$$P(\exists k : |A \cap d_k(v)| \geq 2) \leq Cp^2N^2 \log N$$

For a constant C.



Current known bounds: Lower Bound

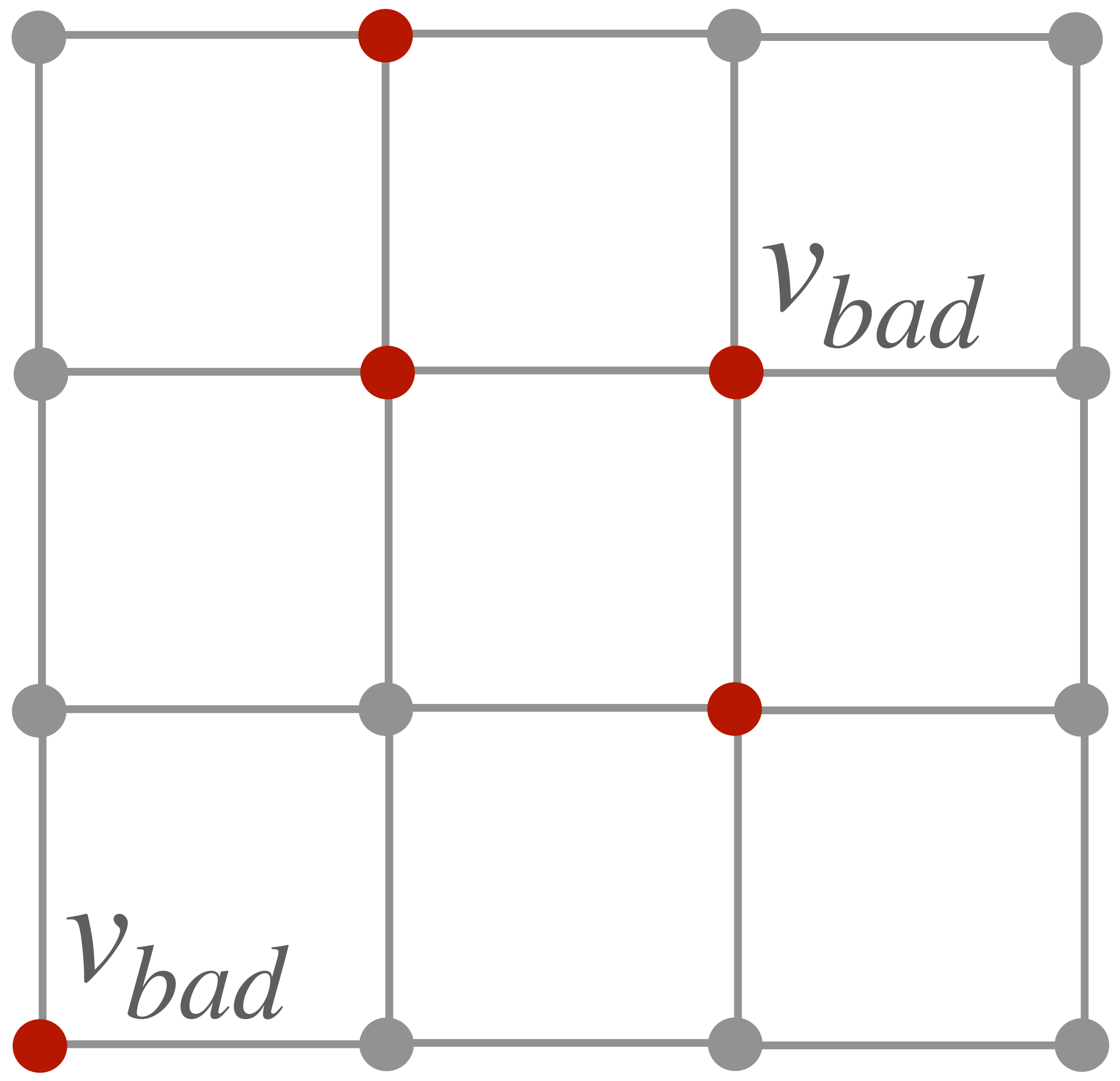
Call a vertex v <u>*bad*</u> if it is in A and there are two points in A at the same distance from v.



Current known bounds: Lower Bound

Call a vertex v *bad* if it is in A and there are two points in A at the same distance from v. Then, we have

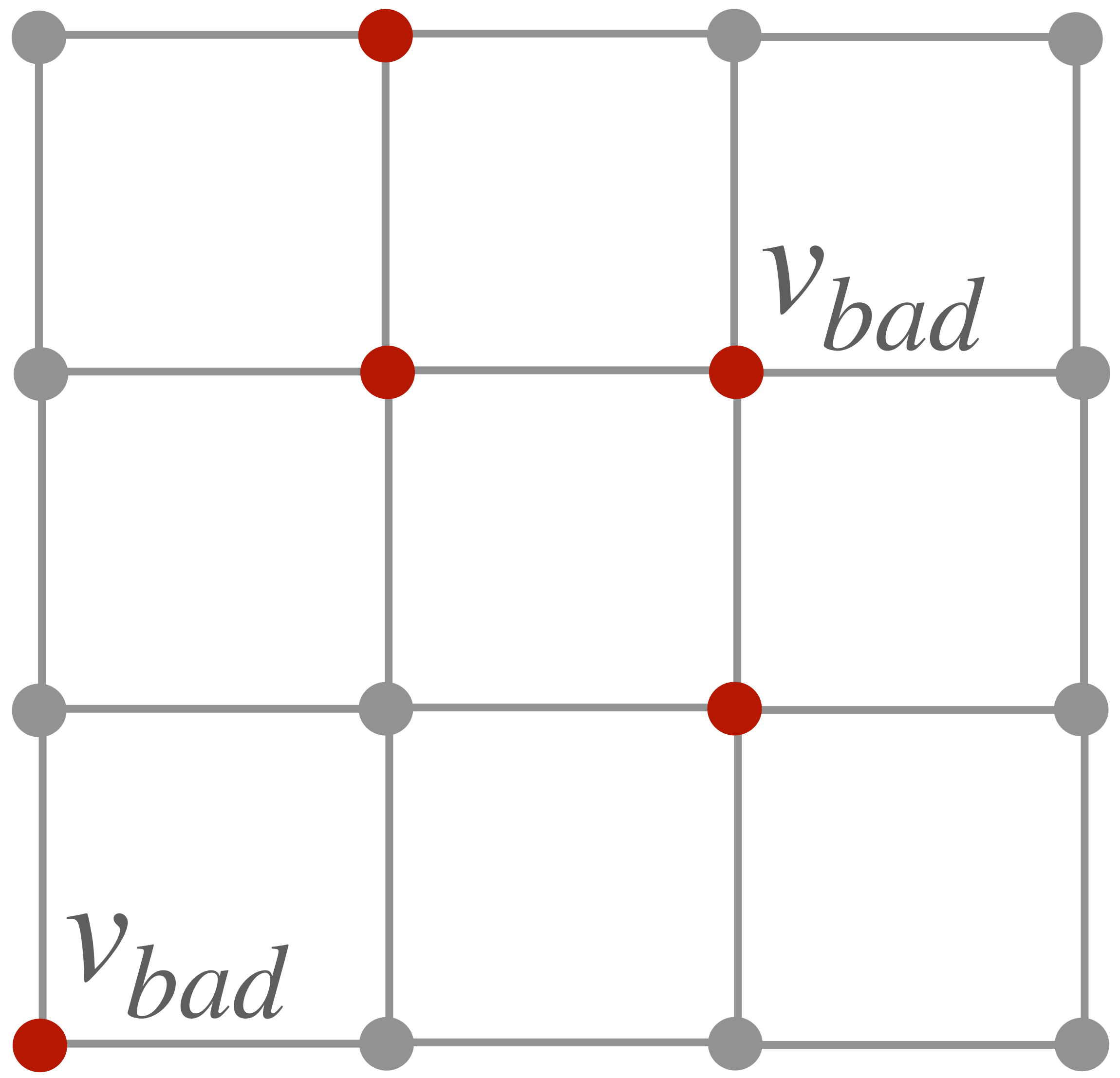
$$P(v \text{ is bad}) = p \cdot P(\exists k : |A \cap d_k(v)| \geq 2)$$



Current known bounds: Lower Bound

Call a vertex v *bad* if it is in A and there are two points in A at the same distance from v. Then, we have

$$P(v \text{ is bad}) = p \cdot P(\exists k : |A \cap d_k(v)| \geq 2) \leq Cp^3N^2 \log N$$



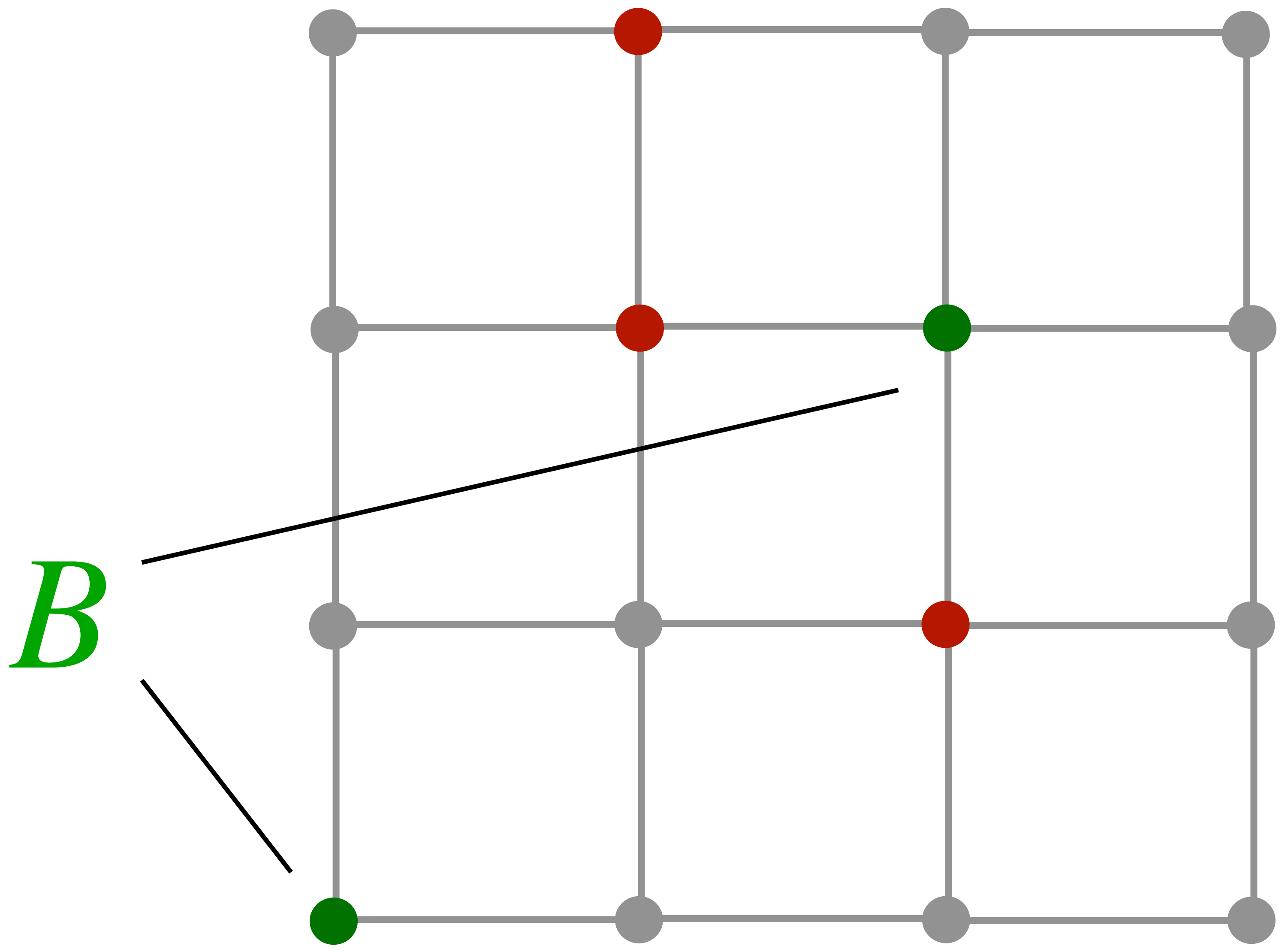
Current known bounds: Lower Bound

Call a vertex v *bad* if it is in A and there are two points in A at the same distance from v. Then, we have

$$P(v \text{ is bad}) = p \cdot P(\exists k : |A \cap d_k(v)| \geq 2) \leq Cp^3N^2 \log N$$

Let B be the set of bad vertices. Then,

$$\mathbb{E}(|B|) = \sum_{v \in \text{grid}} P(v \text{ is bad}) \leq Cp^3N^4 \log N$$



Current known bounds: Lower Bound

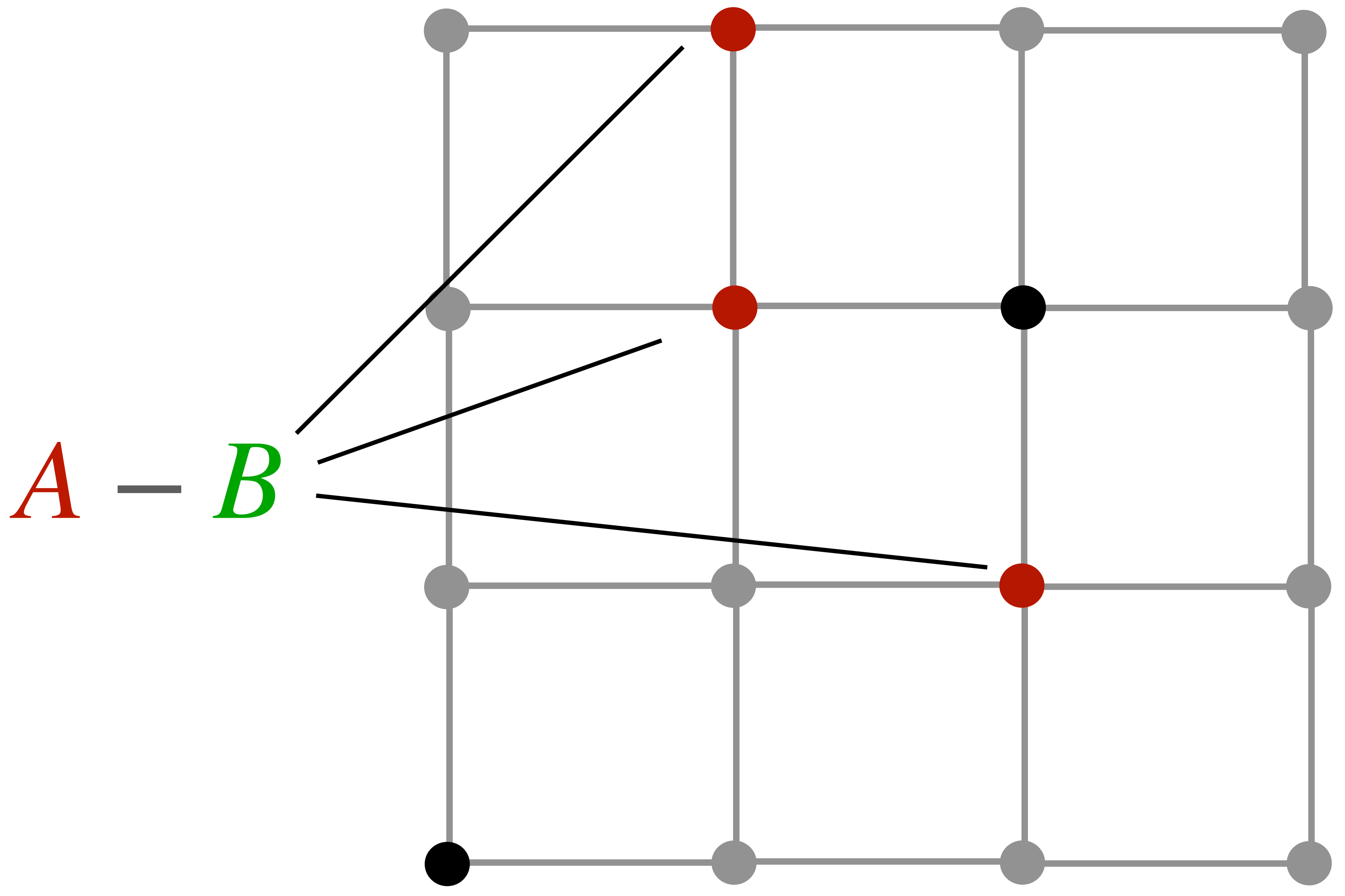
Call a vertex v <u>bad</u> if it is in A and there are two points in A at the same distance from v. Then, we have

$$P(v \text{ is bad}) = p \cdot P(\exists k : |A \cap d_k(v)| \geq 2) \leq Cp^3N^2 \log N$$

Let B be the set of bad vertices. Then,

$$\mathbb{E}(|B|) = \sum_{v \in \text{grid}} P(v \text{ is bad}) \leq Cp^3N^4 \log N$$

Then, $\mathbb{E}(|A - B|) = \mathbb{E}(|A|) - \mathbb{E}(|B|) \geq N^2p - Cp^3N^4 \log N$



Current known bounds: Lower Bound

Call a vertex v _bad_ if it is in A and there are two points in A at the same distance from v. Then, we have

$$P(v \text{ is bad}) = p \cdot P(\exists k : |A \cap d_k(v)| \geq 2) \leq Cp^3N^2 \log N$$

Let B be the set of bad vertices. Then,

$$\mathbb{E}(|B|) = \sum_{v \in \text{grid}} P(v \text{ is bad}) \leq Cp^3N^4 \log N$$

Then, $\mathbb{E}(|A - B|) = \mathbb{E}(|A|) - \mathbb{E}(|B|) \geq N^2p - Cp^3N^4 \log N$

Taking $p = \dfrac{\epsilon}{N\sqrt{\log N}}$ where ϵ is a small enough constant only depending on C, we get

$$\mathbb{E}(|A - B|) \geq \frac{N}{\sqrt{\log N}}(\epsilon - C\epsilon^3) \geq \epsilon' \frac{N}{\sqrt{\log N}}$$

for an absolute constant ϵ'. $\square$

Current known bounds: Upper Bound

Roth's Theorem [Roth, 1953]

Let $r([N])$ be the size of the largest subset of $[1,...,N]$ that contain no 3-term arithmetic progressions. Then,

$$r([N]) = O(\frac{N}{\log \log N})$$

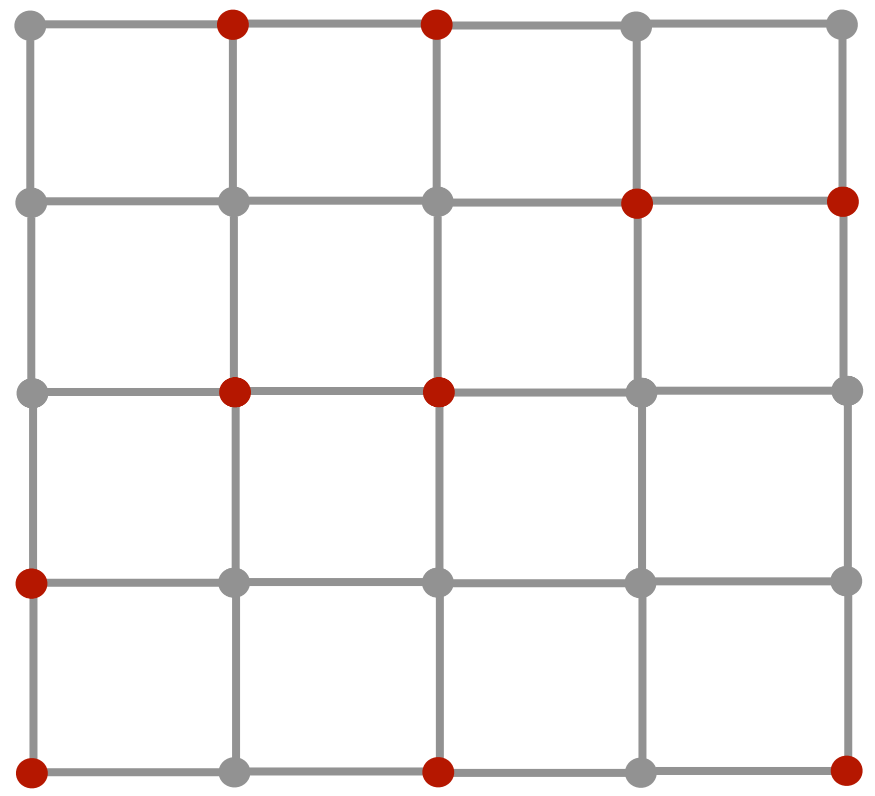
Current known bounds: Upper Bound

Roth's Theorem [Roth, 1953]

Let $r([N])$ be the size of the largest subset of $[1,\ldots,N]$ that contain no 3-term arithmetic progressions. Then,

$$r([N]) = O\left(\frac{N}{\log\log N}\right)$$

Given a subset S of the NxN grid of density greater than $O\left(\left(\frac{N}{\log\log N}\right)^2\right)$



Current known bounds: Upper Bound

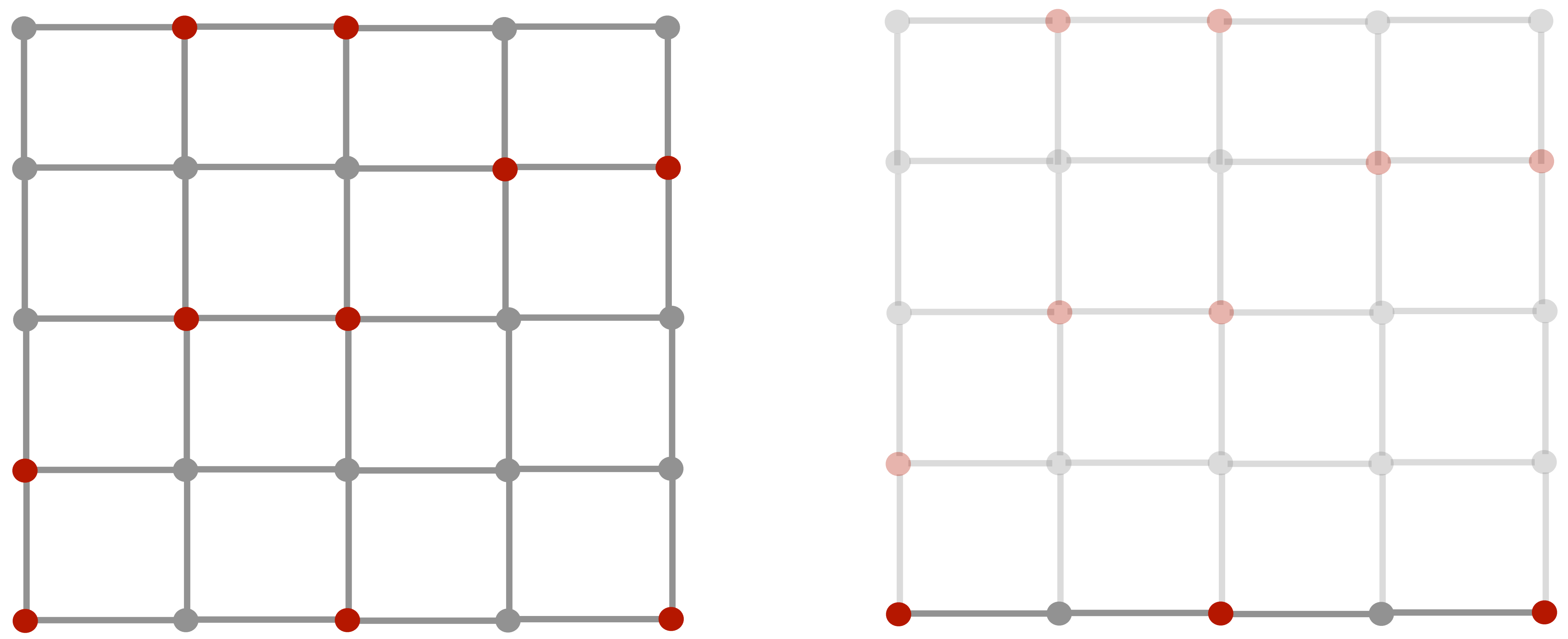
Roth's Theorem [Roth, 1953]

Let $r([N])$ be the size of the largest subset of $[1,\ldots,N]$ that contain no 3-term arithmetic progressions. Then,

$$r([N]) = O(\frac{N}{\log\log N})$$

Given a subset S of the NxN grid of density greater than $O((\frac{N}{\log\log N})^2)$

Then, for some j, we have that the jth row of S has density greater than $O(\frac{N}{\log\log N})$.



Current known bounds: Upper Bound

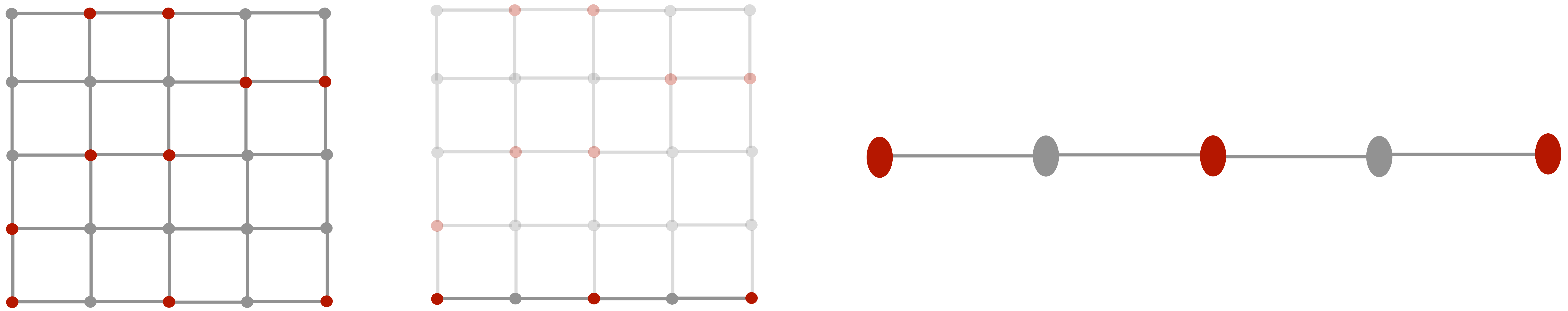
Roth's Theorem [Roth, 1953]

Let $r([N])$ be the size of the largest subset of $[1,\ldots,N]$ that contain no 3-term arithmetic progressions. Then,

$$r([N]) = O(\frac{N}{\log \log N})$$

Given a subset S of the NxN grid of density greater than $O((\frac{N}{\log \log N})^2)$

Then, for some j, we have that the jth row of S has density greater than $O(\frac{N}{\log \log N})$. By Roth's Theorem, S_j contains a 3-term arithmetic progression, i.e. an isosceles triangle.



Current known bounds: Upper Bound

Theorem [Kelley, Meka, 2023]

Let $r([N])$ be the size of the largest subset of $[1,\ldots,N]$ that contain no 3-term arithmetic progressions. Then,

$$r([N]) \leq 2^{-O((\log N)^c)\cdot N}$$

Theorem [Bloom, Sisask, 2023]

Let $r([N])$ be the size of the largest subset of $[1,\ldots,N]$ that contain no 3-term arithmetic progressions. Then,

$$r([N]) \leq \exp(-c(\log N)^{\frac{1}{9}})N$$

Current known bounds: Upper Bound

Theorem [Kelley, Meka, 2023]

Let $r([N])$ be the size of the largest subset of $[1,\ldots,N]$ that contain no 3-term arithmetic progressions. Then,

$$r([N]) \leq 2^{-O((\log N)^c)\cdot N}$$

Theorem [Bloom, Sisask, 2023]

Let $r([N])$ be the size of the largest subset of $[1,\ldots,N]$ that contain no 3-term arithmetic progressions. Then,

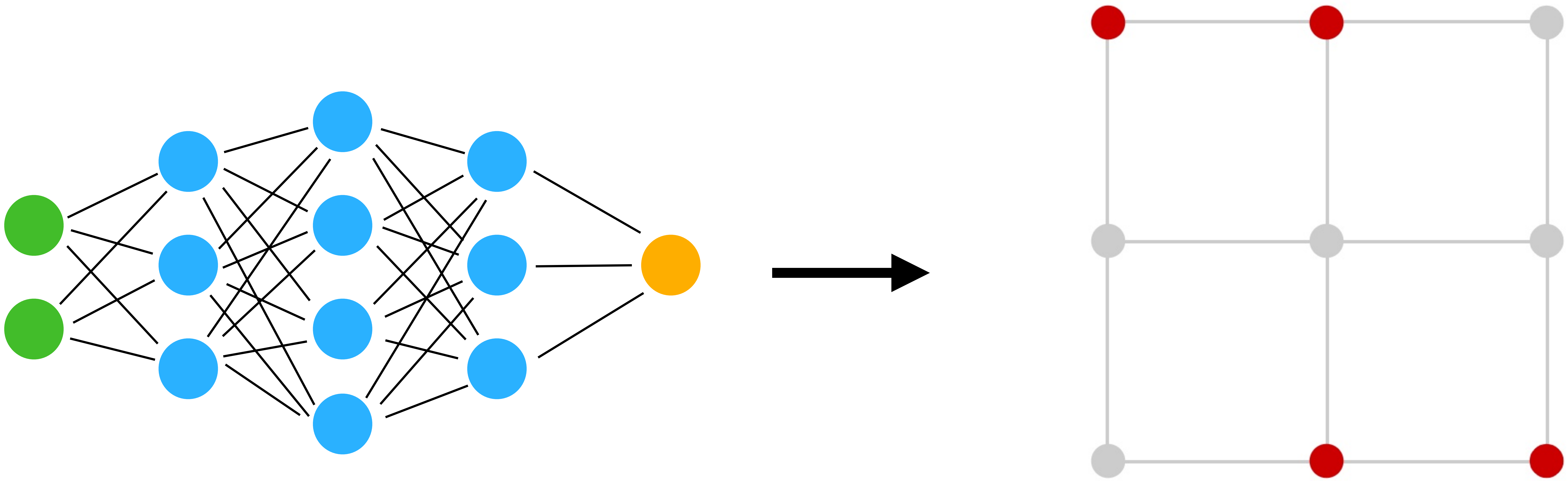
$$r([N]) \leq \exp(-c(\log N)^{\frac{1}{9}})N$$

Final Bounds

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

Aim

- Computationally generate large isosceles free subsets of the integer lattice.



Overview

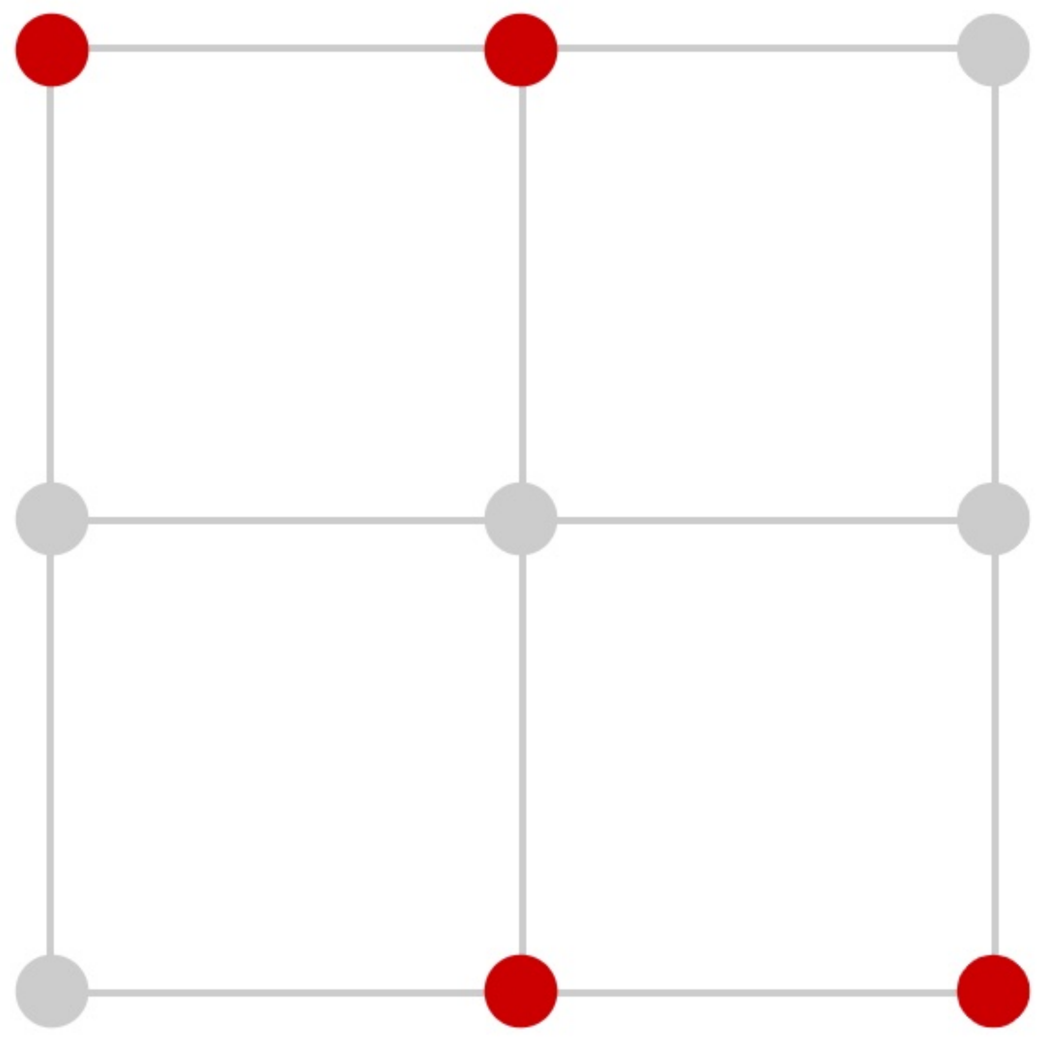
Mathematical Motivation and Background

- Motivation: Non Metric Multidimensional Scaling
- Key definitions and propositions
- Known bounds for the problem

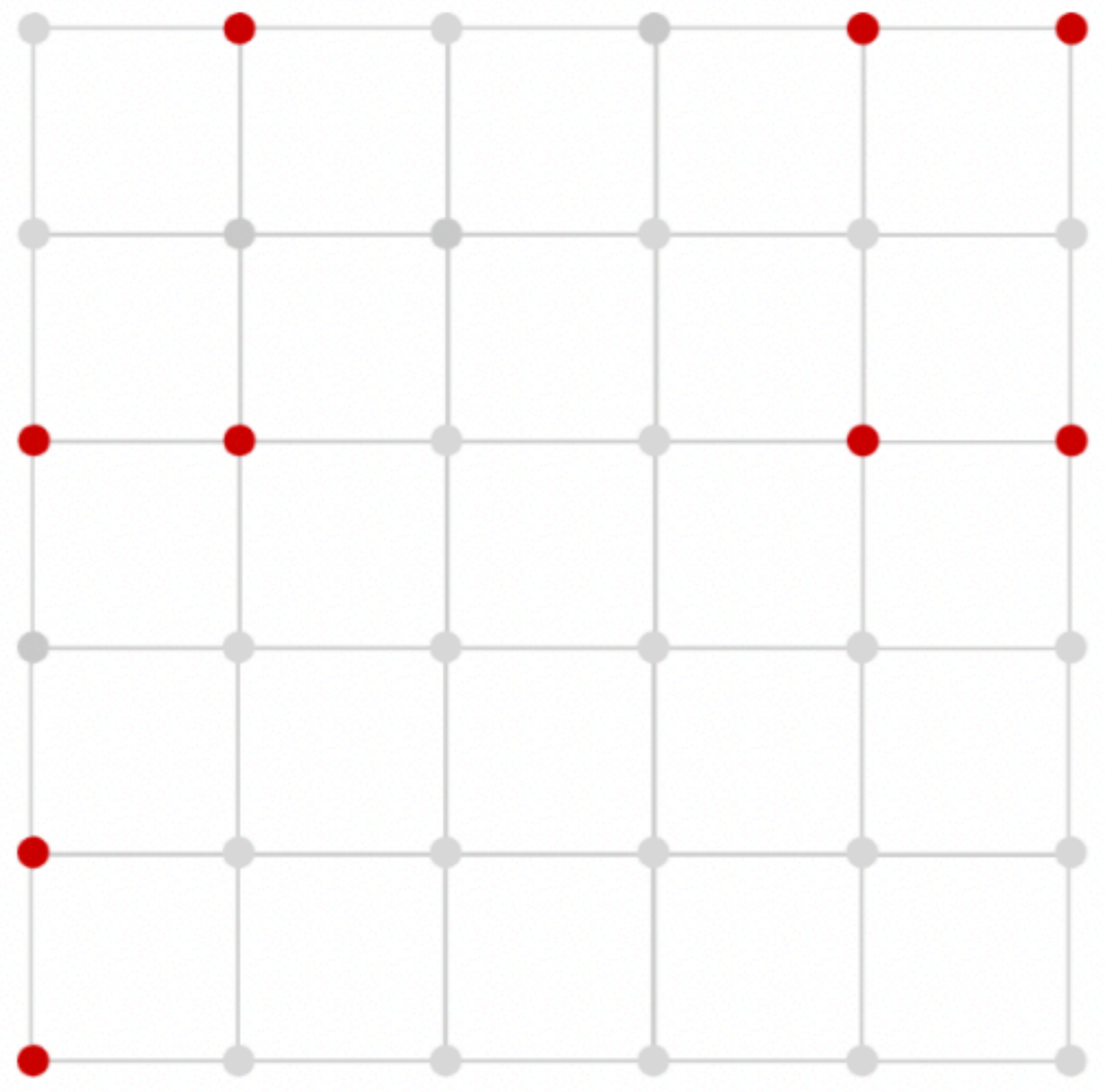
How Reinforcement Learning can help

- Reinforcement learning background and main algorithm
- Current results and observations
- Next Steps

Our Problem



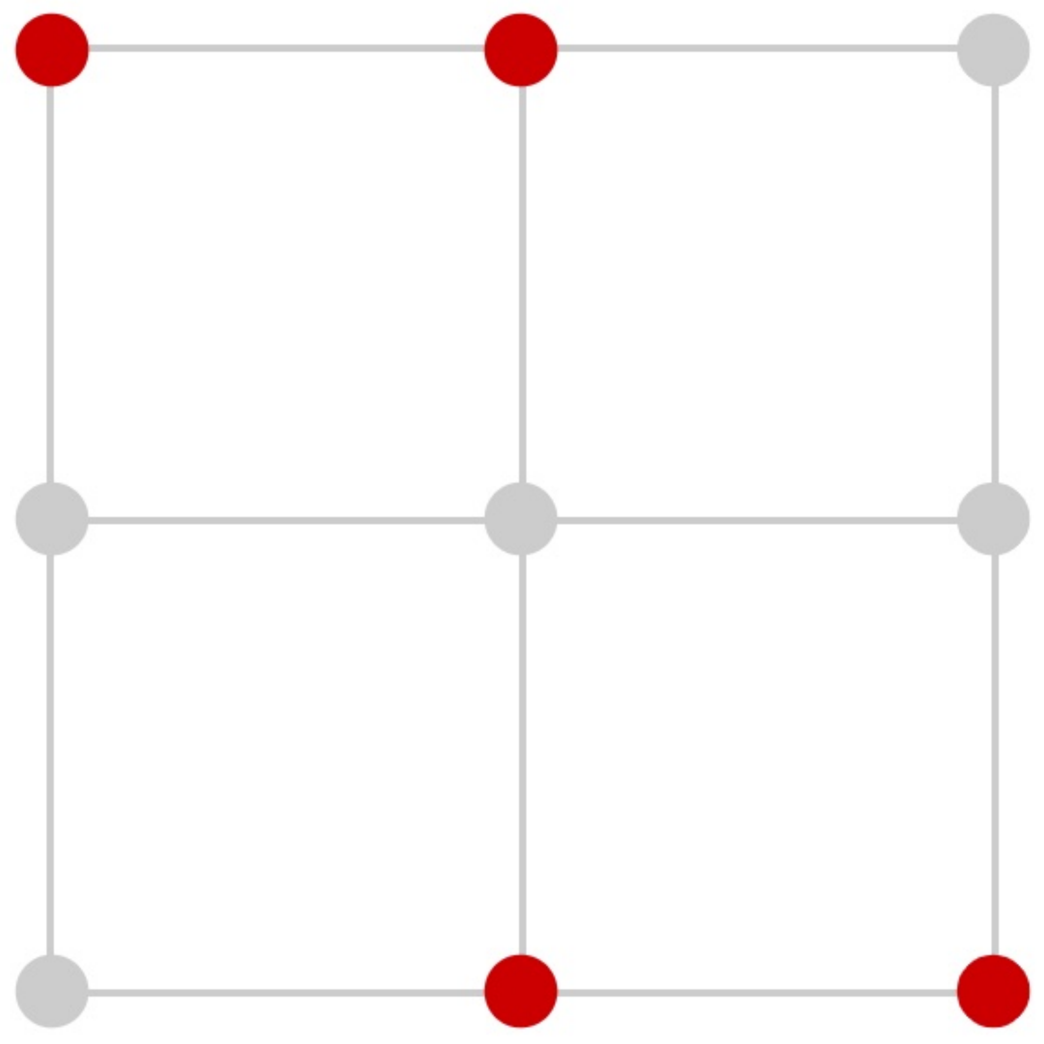
N = 3



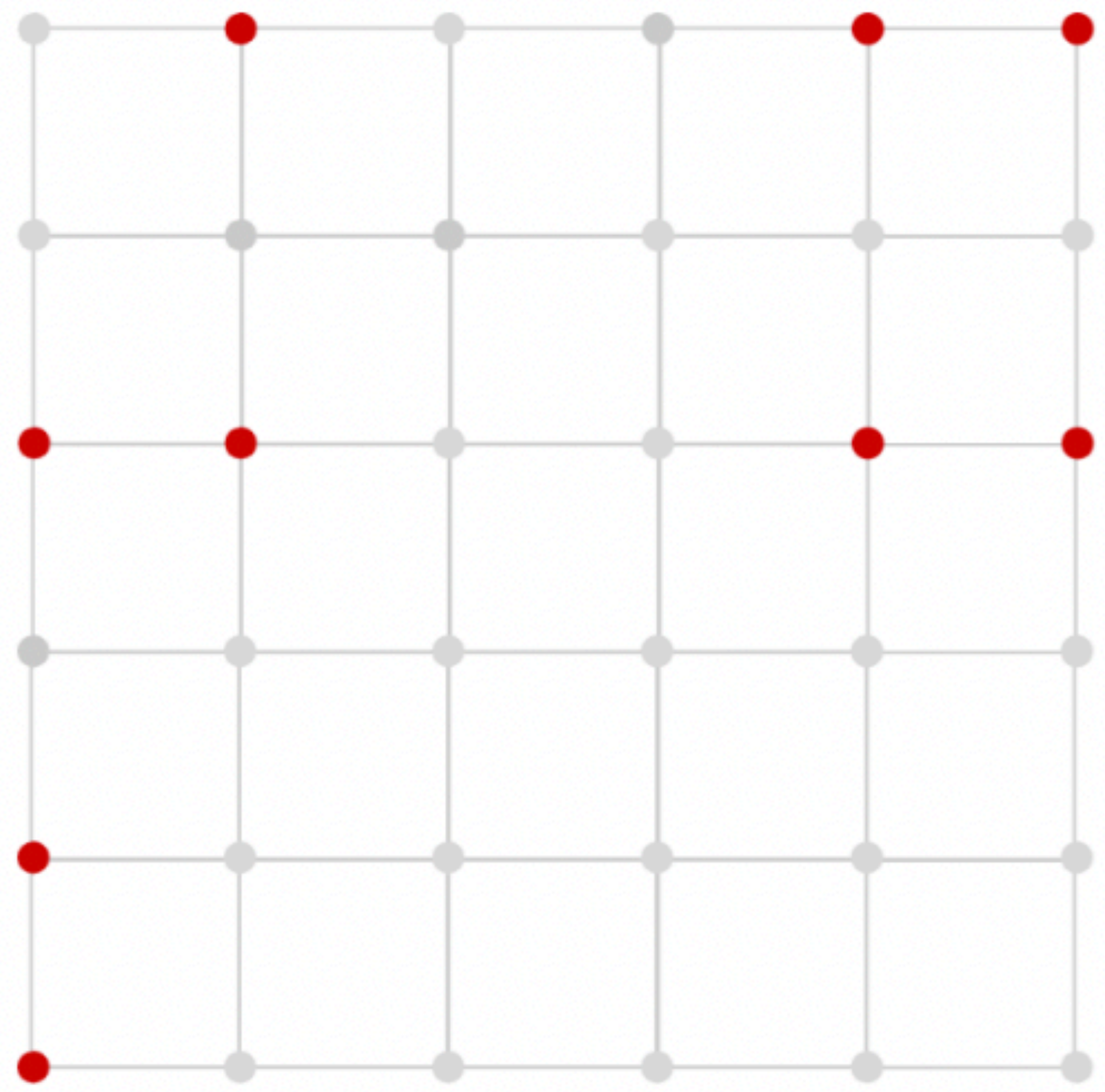
N = 6

$$O(2^{N^2})$$

Our Problem



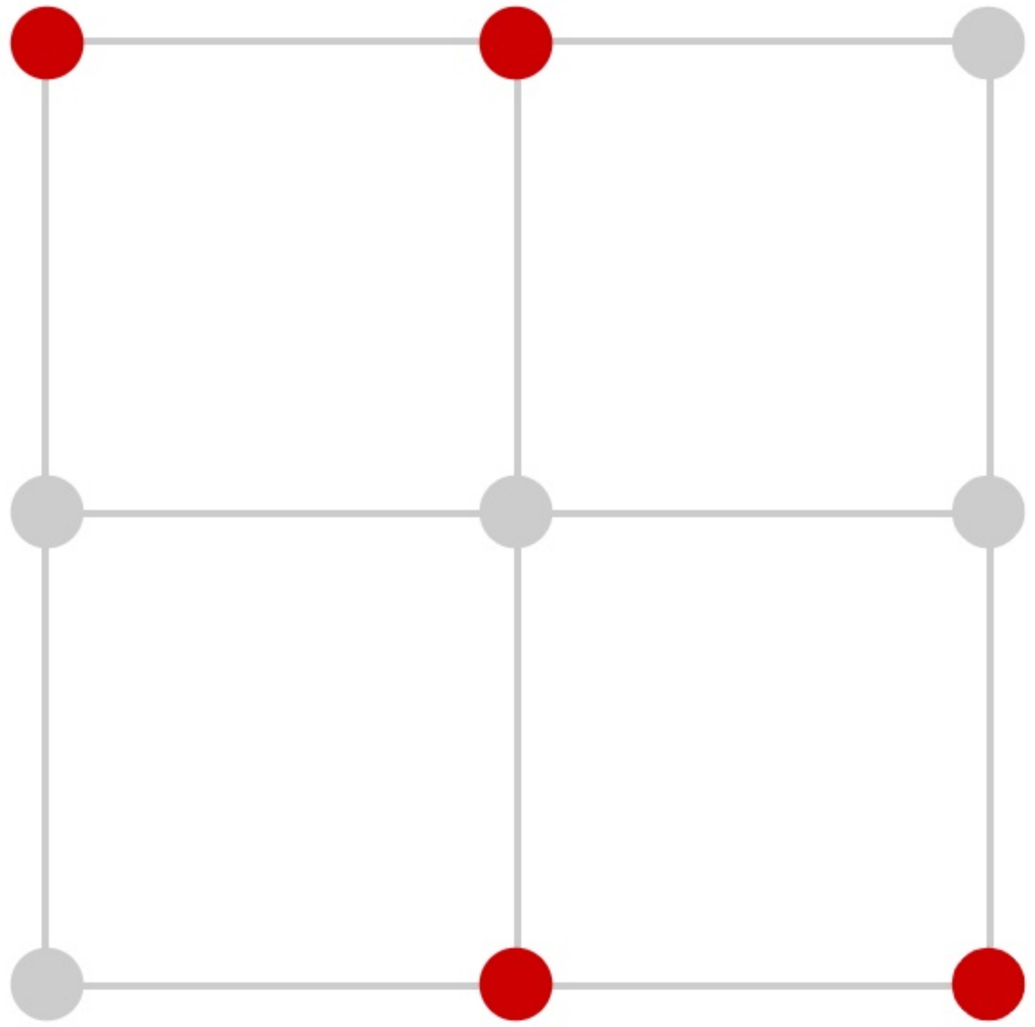
N = 3



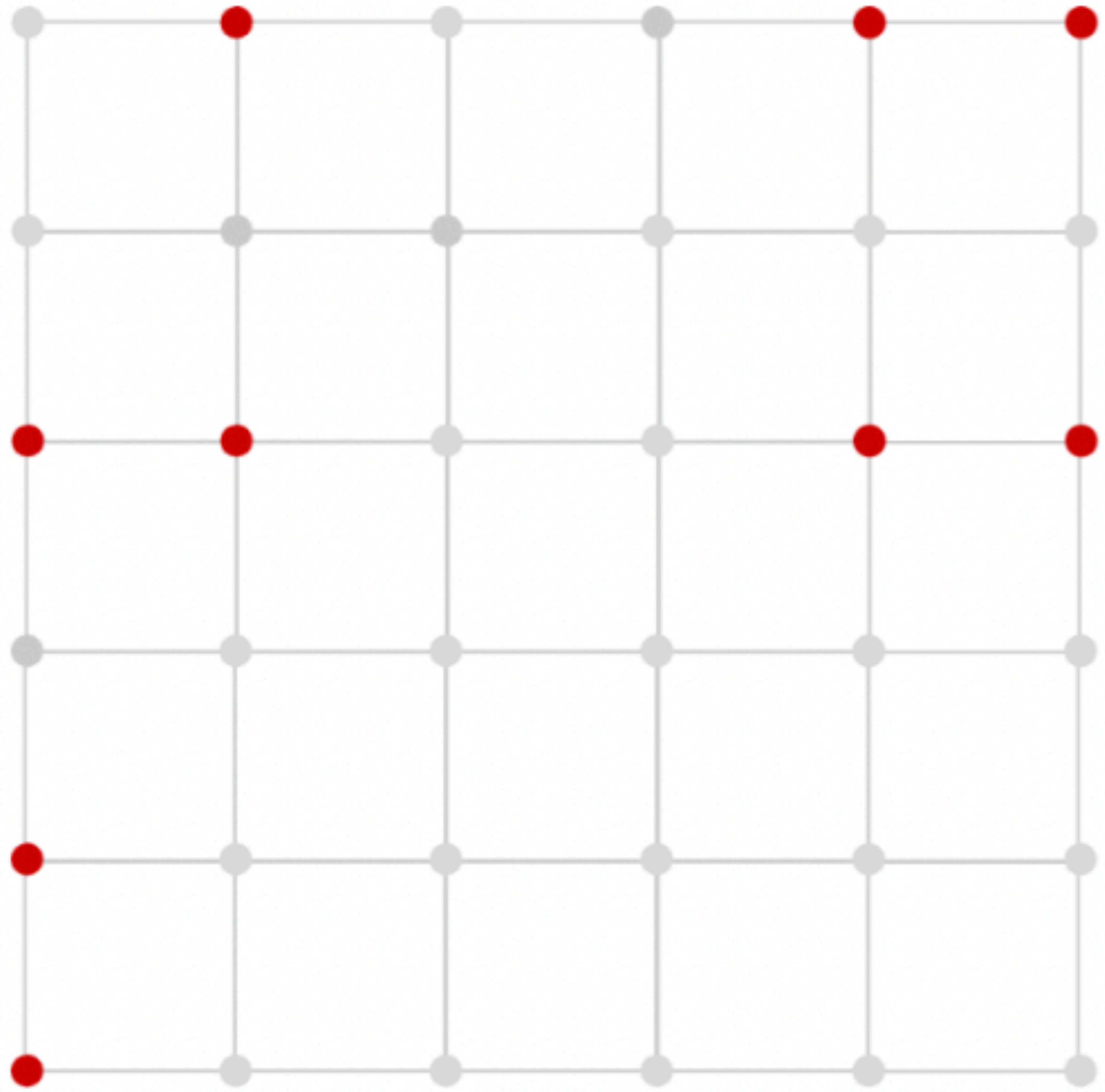
N = 6

$$O(2^{N^2})$$ → **Brute Force**

Our Problem



N = 3

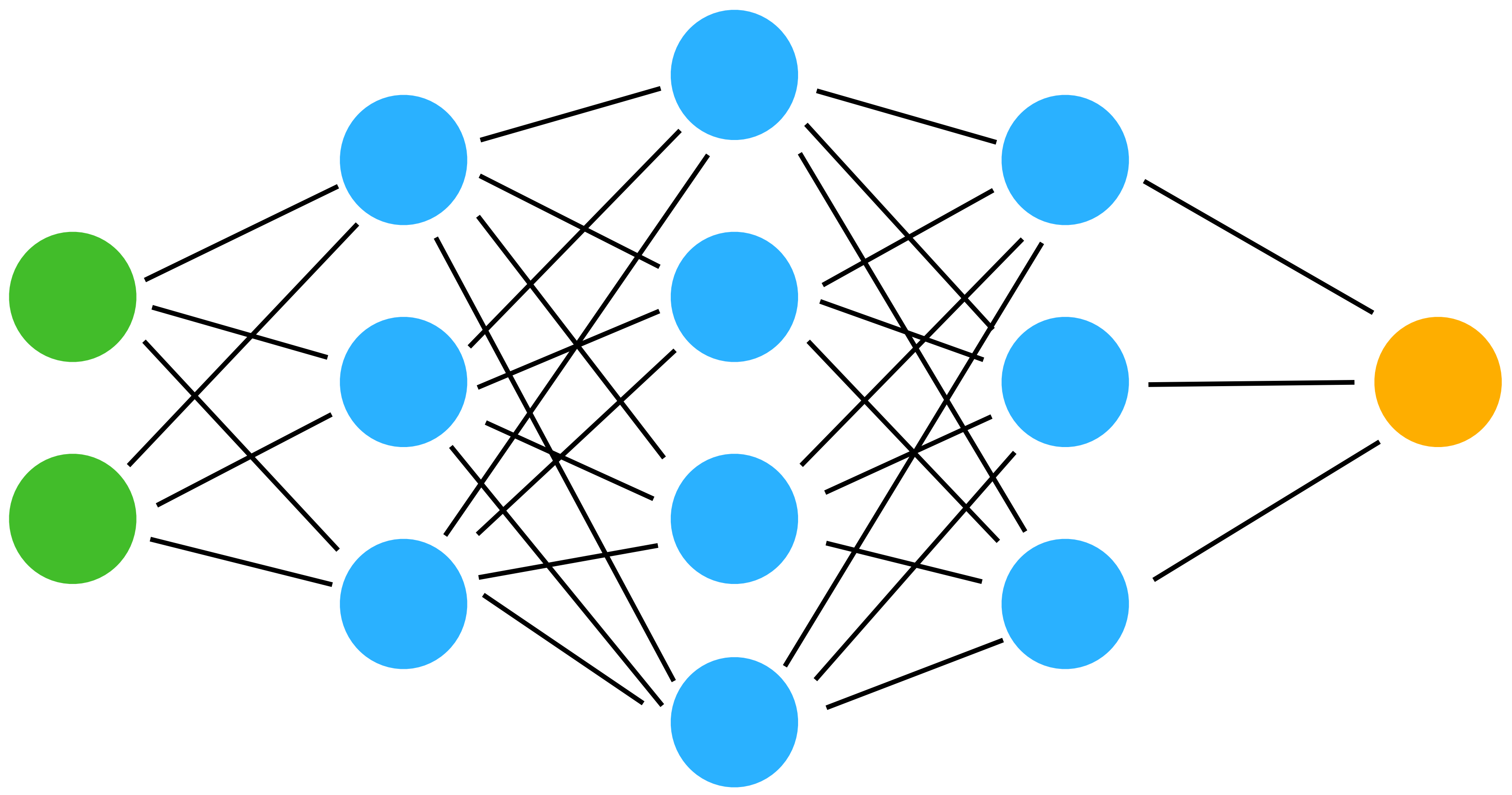


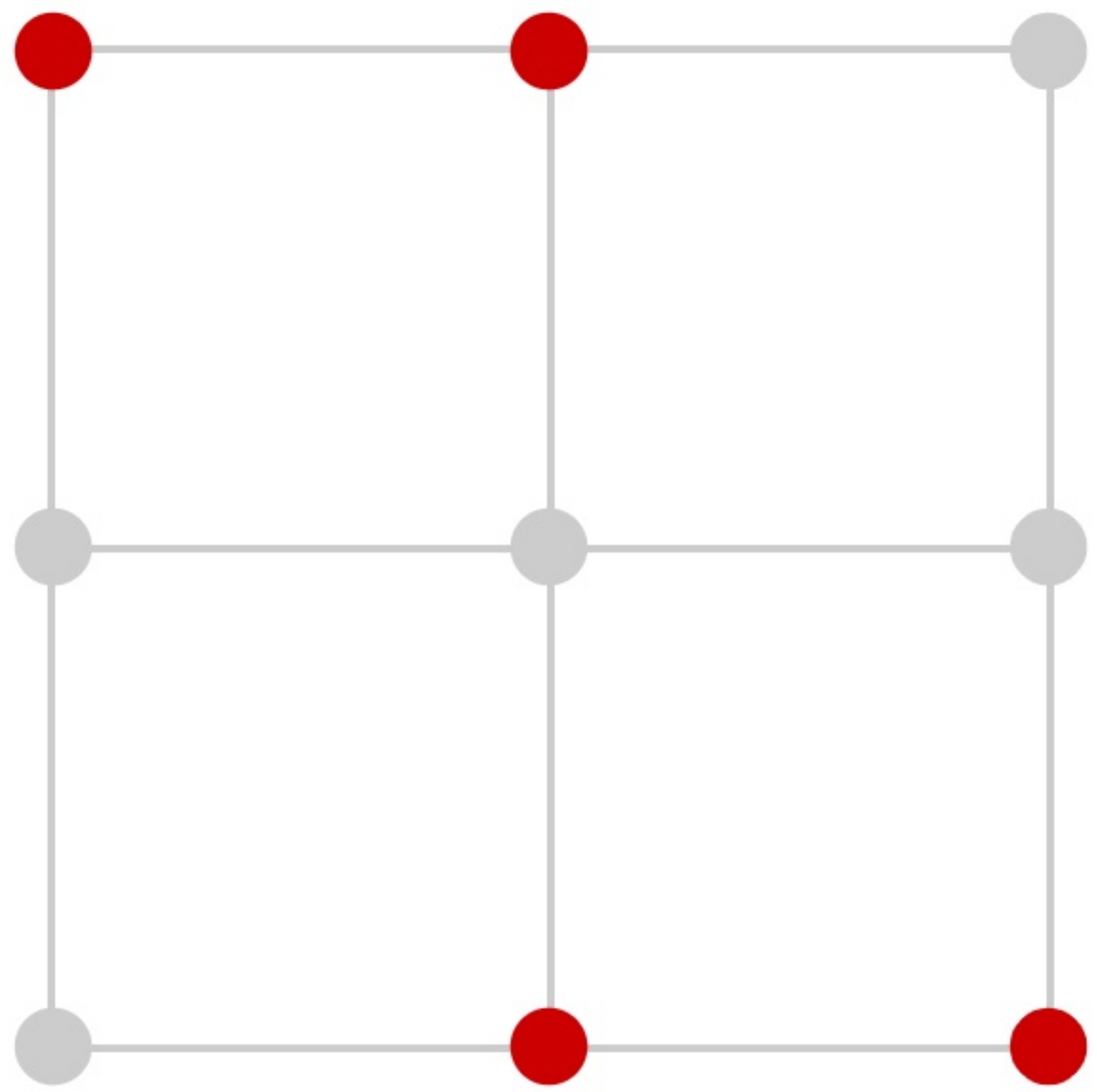
N = 6

$$O(2^{N^2})$$

Aim

- Computationally generate large isosceles free subsets of the integer lattice using reinforcement learning





Some Vocabulary

Reinforcement Learning: Learning Decisions to Maximize Reward

Some Vocabulary

Reinforcement Learning: Learning Decisions to Maximize Reward

- An agent plays a game many times

Some Vocabulary

Reinforcement Learning: Learning Decisions to Maximize Reward

- An agent plays a game many times
- It knows the rules of the game but nothing else

Some Vocabulary

Reinforcement Learning: Learning Decisions to Maximize Reward

- An agent plays a game many times
- It knows the rules of the game but nothing else
- Takes actions without knowing best move

Some Vocabulary

Reinforcement Learning: Learning Decisions to Maximize Reward

- An agent plays a game many times
- It knows the rules of the game but nothing else
- Takes actions without knowing best move
- Rewarded at the end of each game

Some Vocabulary

Reinforcement Learning: Learning Decisions to Maximize Reward

- An agent plays a game many times
- It knows the rules of the game but nothing else
- Takes actions without knowing best move
- Rewarded at the end of each game
- Through many games, tries to maximize rewards

Some Vocabulary

Reinforcement Learning: Learning Decisions to Maximize Reward

- An agent plays a game many times
- It knows the rules of the game but nothing else
- Takes actions without knowing best move
- Rewarded at the end of each game
- Through many games, tries to maximize rewards

Previous works in machine learning applied to math

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > articles > article

Article | Open Access | Published: 01 December 2021

Advancing mathematics by guiding human intuition with AI

Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis & Pushmeet Kohli

Nature **600**, 70–74 (2021) | Cite this article

Abstract

The practice of mathematics involves discovering patterns and using these to formulate and prove conjectures, resulting in theorems. Since the 1960s, mathematicians have used computers to assist in the discovery of patterns and formulation of conjectures[1], most famously in the Birch and Swinnerton-Dyer conjecture[2], a Millennium Prize Problem[3]. Here we provide examples of new fundamental results in pure mathematics that have been discovered with the assistance of machine learning—demonstrating a method by which machine learning can aid mathematicians in discovering new conjectures and theorems. We propose a process of using machine learning to discover potential patterns and relations between mathematical objects, understanding them with attribution techniques and using these observations to guide intuition and propose conjectures. We outline this machine-

Discovering connections between algebraic and geometric knot-invariants with supervised learning

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

Finding faster matrix multiplication algorithms using reinforcement learning

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > articles > article

Article | Open access | Published: 14 December 2023

Mathematical discoveries from program search with large language models

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli & Alhussein Fawzi

Nature **625**, 468–475 (2024) | Cite this article

Abstract

Large language models (LLMs) have demonstrated tremendous capabilities in solving complex tasks, from quantitative reasoning to understanding natural language. However, LLMs sometimes suffer from confabulations (or hallucinations), which can result in them making plausible but incorrect statements[1,2]. This hinders the use of current large models in scientific discovery. Here we introduce FunSearch (short for searching in the function space), an evolutionary procedure based on pairing a pretrained LLM with a systematic evaluator. We demonstrate the effectiveness of this approach to surpass the best-known results in important problems, pushing the boundary of existing LLM-based approaches[3]. Applying FunSearch to a central problem in extremal combinatorics—the cap set problem—we discover new constructions of large cap sets going beyond the best-known ones, both in finite

Discovering largest known capsets using large language models to search space of programs

Previous works in machine learning applied to math

nature
Explore content ⌄ About the journal ⌄ Publish with us ⌄
nature > articles > article
Article | Open Access | Published: 01 December 2021

Advancing mathematics by guiding human intuition with AI

Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis & Pushmeet Kohli

Nature 600, 70–74 (2021) | Cite this article

Abstract

The practice of mathematics involves discovering patterns and using these to formulate and prove conjectures, resulting in theorems. Since the 1960s, mathematicians have used computers to assist in the discovery of patterns and formulation of conjectures[1], most famously in the Birch and Swinnerton-Dyer conjecture[2], a Millennium Prize Problem[3]. Here we provide examples of new fundamental results in pure mathematics that have been discovered with the assistance of machine learning—demonstrating a method by which machine learning can aid mathematicians in discovering new conjectures and theorems. We propose a process of using machine learning to discover potential patterns and relations between mathematical objects, understanding them with attribution techniques and using these observations to guide intuition and propose conjectures. We outline this machine-

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

nature
Explore content ⌄ About the journal ⌄ Publish with us ⌄
nature > articles > article
Article | Open access | Published: 14 December 2023

Mathematical discoveries from program search with large language models

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli & Alhussein Fawzi

Nature 625, 468–475 (2024) | Cite this article

Abstract

Large language models (LLMs) have demonstrated tremendous capabilities in solving complex tasks, from quantitative reasoning to understanding natural language. However, LLMs sometimes suffer from confabulations (or hallucinations), which can result in them making plausible but incorrect statements[1,2]. This hinders the use of current large models in scientific discovery. Here we introduce FunSearch (short for searching in the function space), an evolutionary procedure based on pairing a pretrained LLM with a systematic evaluator. We demonstrate the effectiveness of this approach to surpass the best-known results in important problems, pushing the boundary of existing LLM-based approaches[3]. Applying FunSearch to a central problem in extremal combinatorics—the cap set problem—we discover new constructions of large cap sets going beyond the best-known ones, both in finite

Discovering connections between algebraic and geometric knot-invariants with supervised learning

Finding faster matrix multiplication algorithms using reinforcement learning

Discovering largest known caplets using large language models to search space of programs

Moral: Machine learning can be good at coming up with good examples

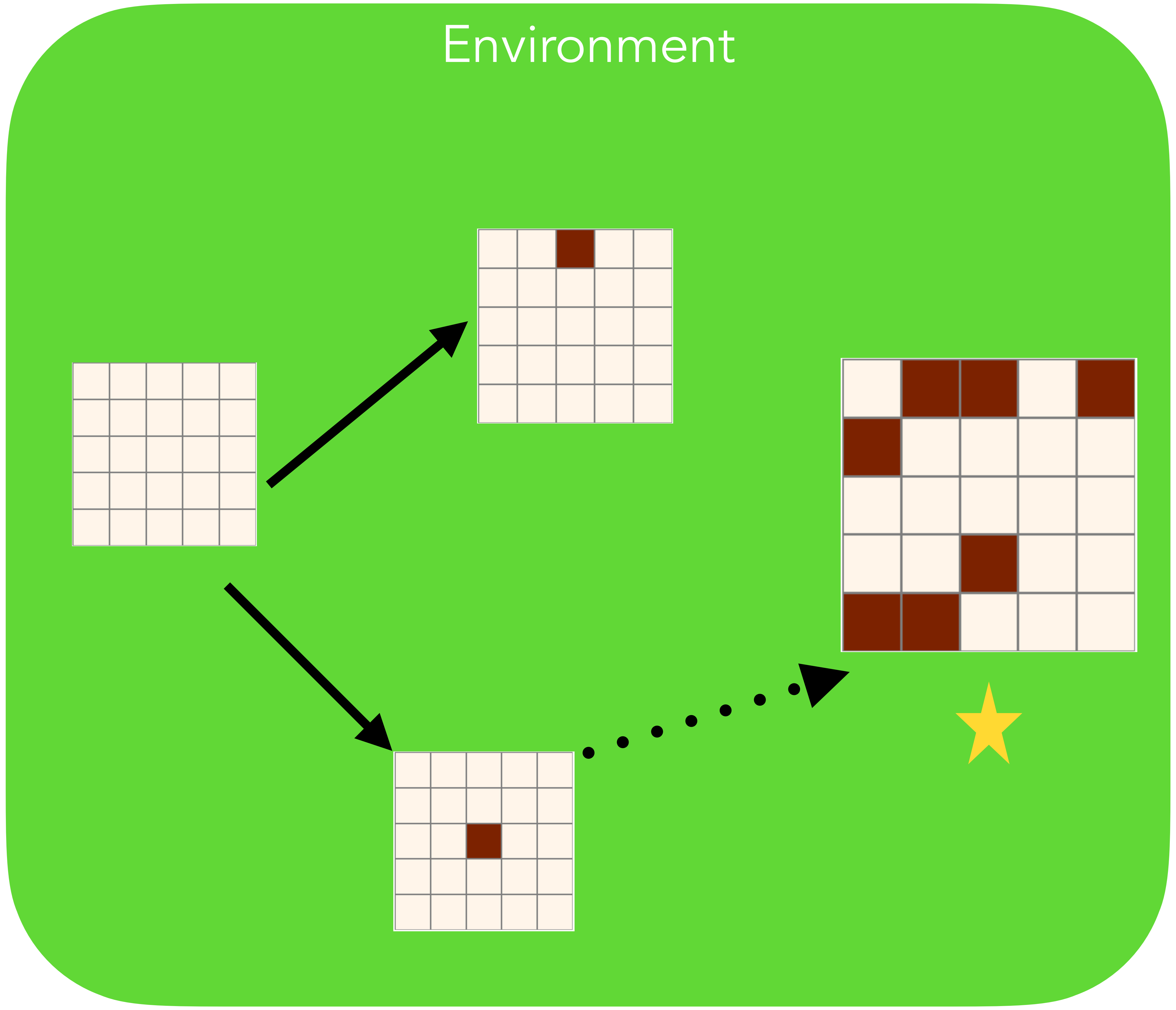
RL Background

What do we need?

RL Background

What do we need?

1. How do we gamify the problem?

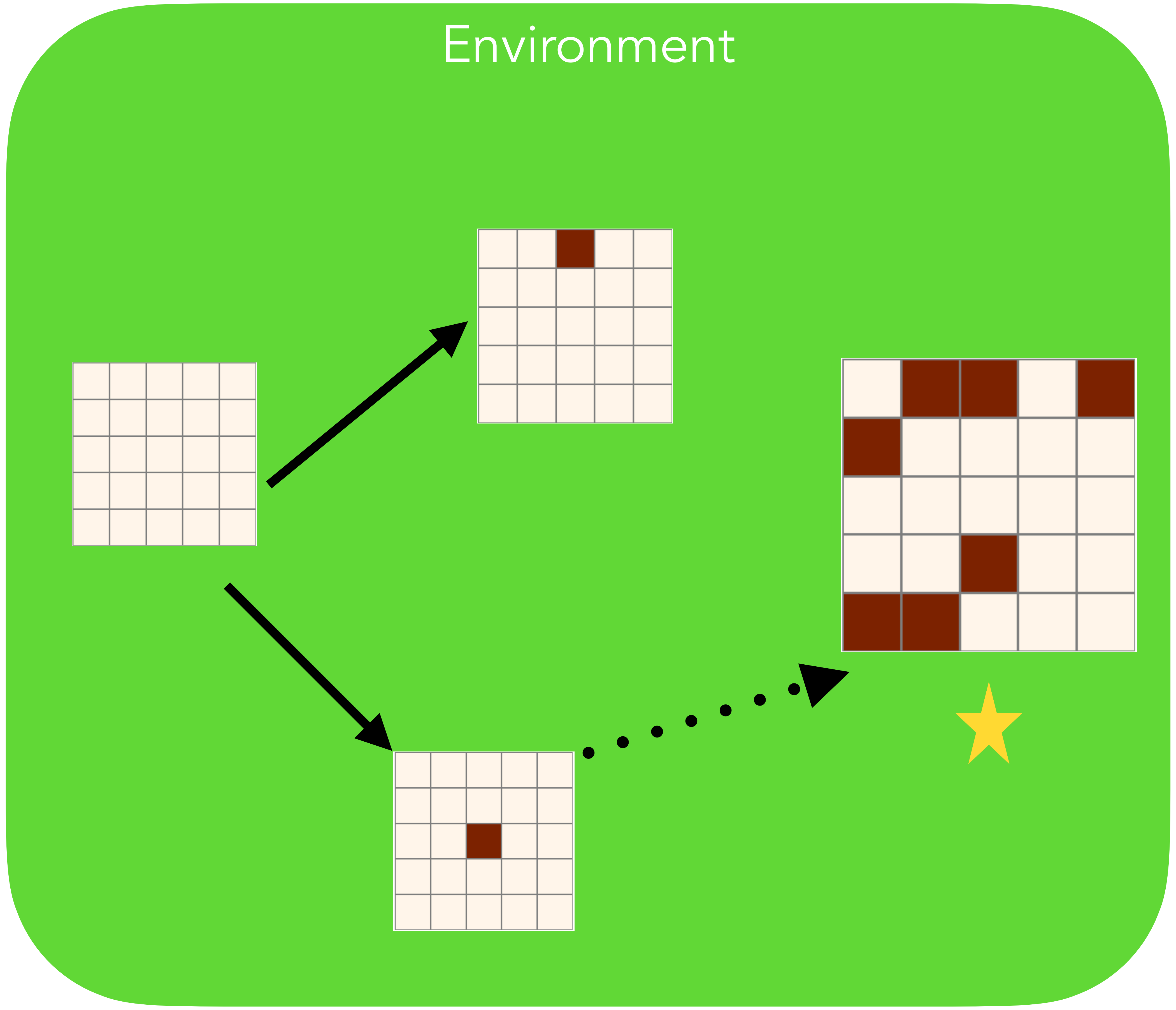


RL Background

What do we need?

1. How do we gamify the problem?

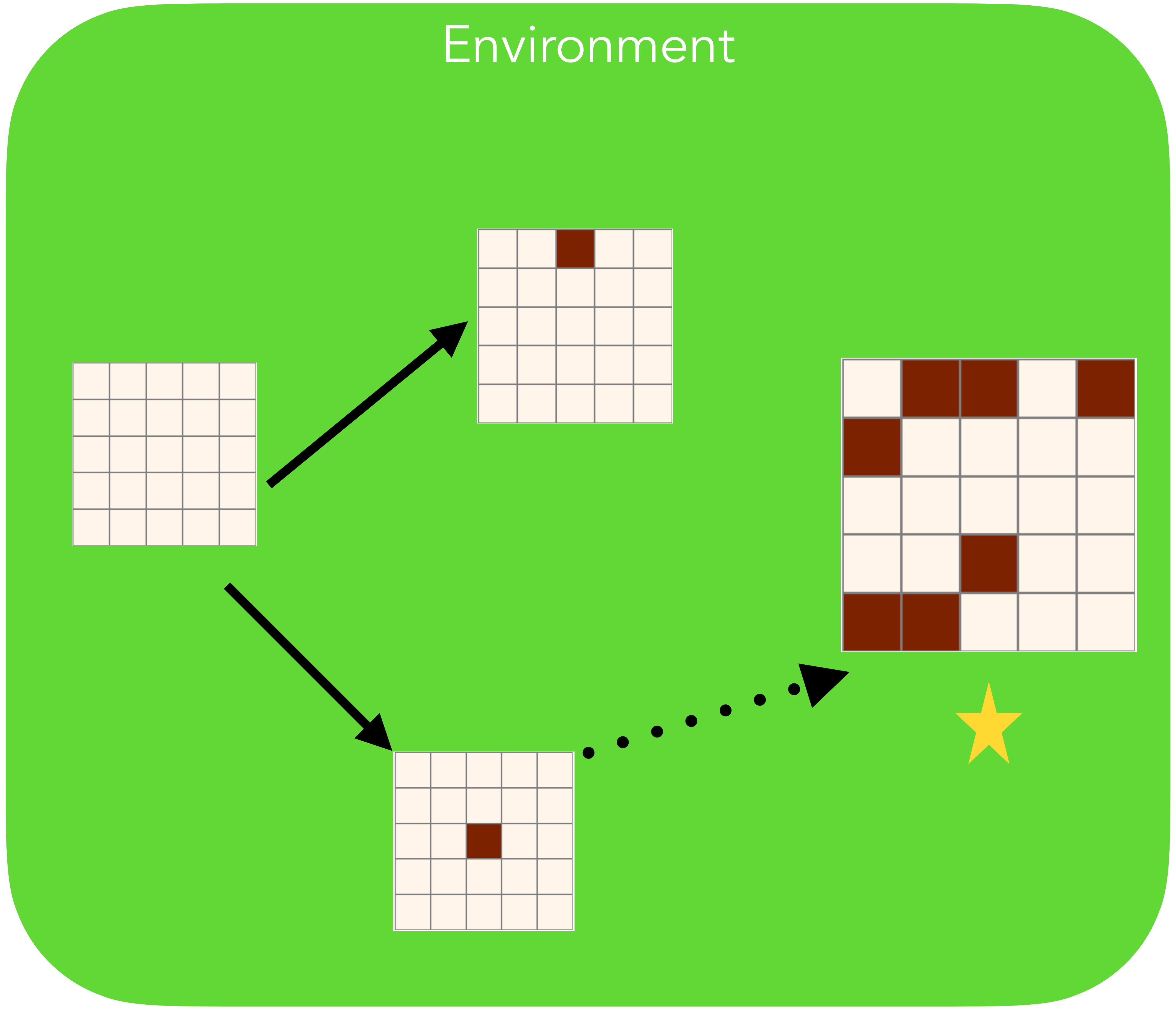
2. What kind of model to use?



RL Background

What do we need?

1. How do we gamify the problem?
2. What kind of model to use?
3. What is the reward function?



RL Background

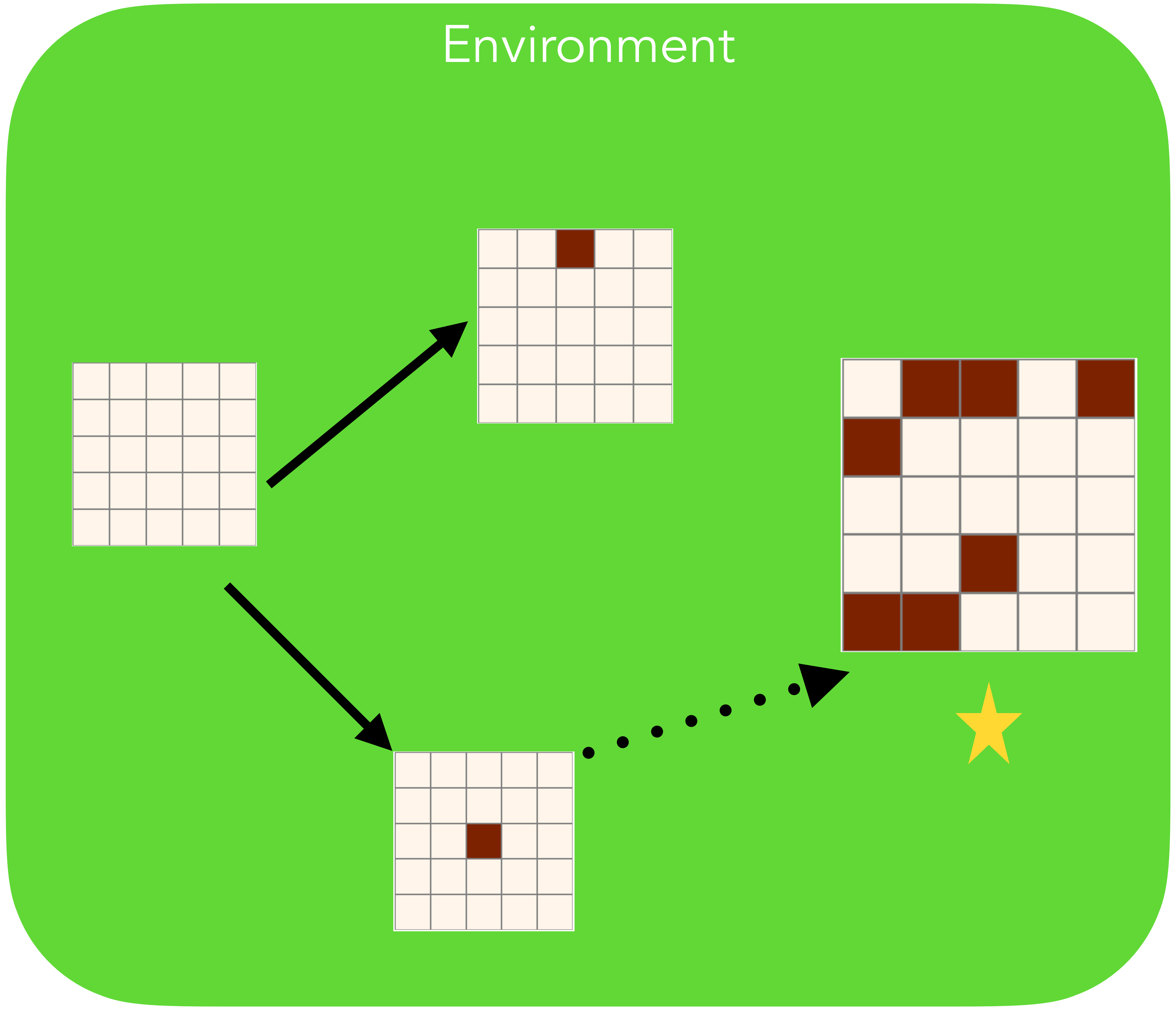
What do we need?

1. How do we gamify the problem?

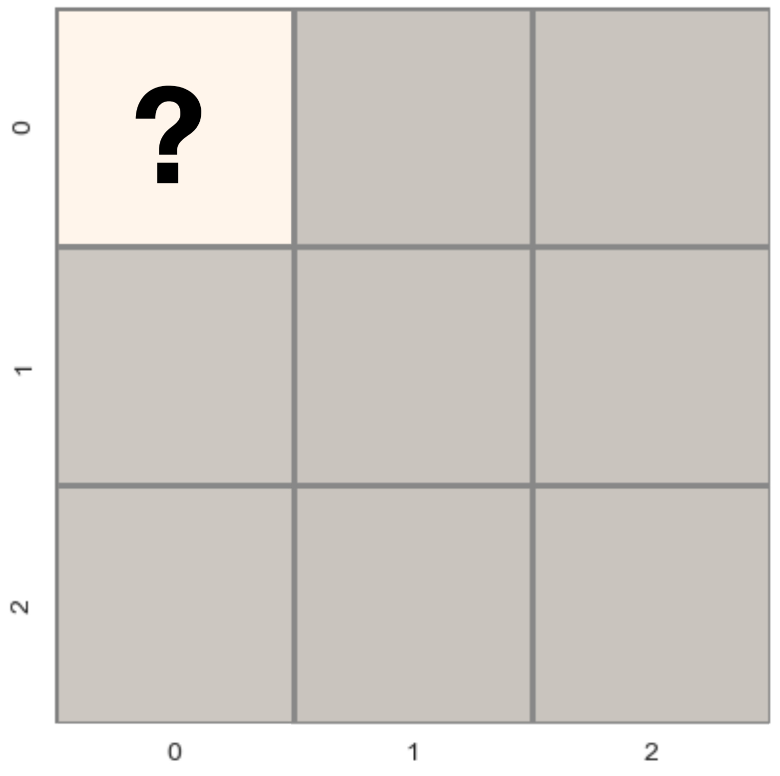
2. What kind of model to use?

3. What is the reward function?

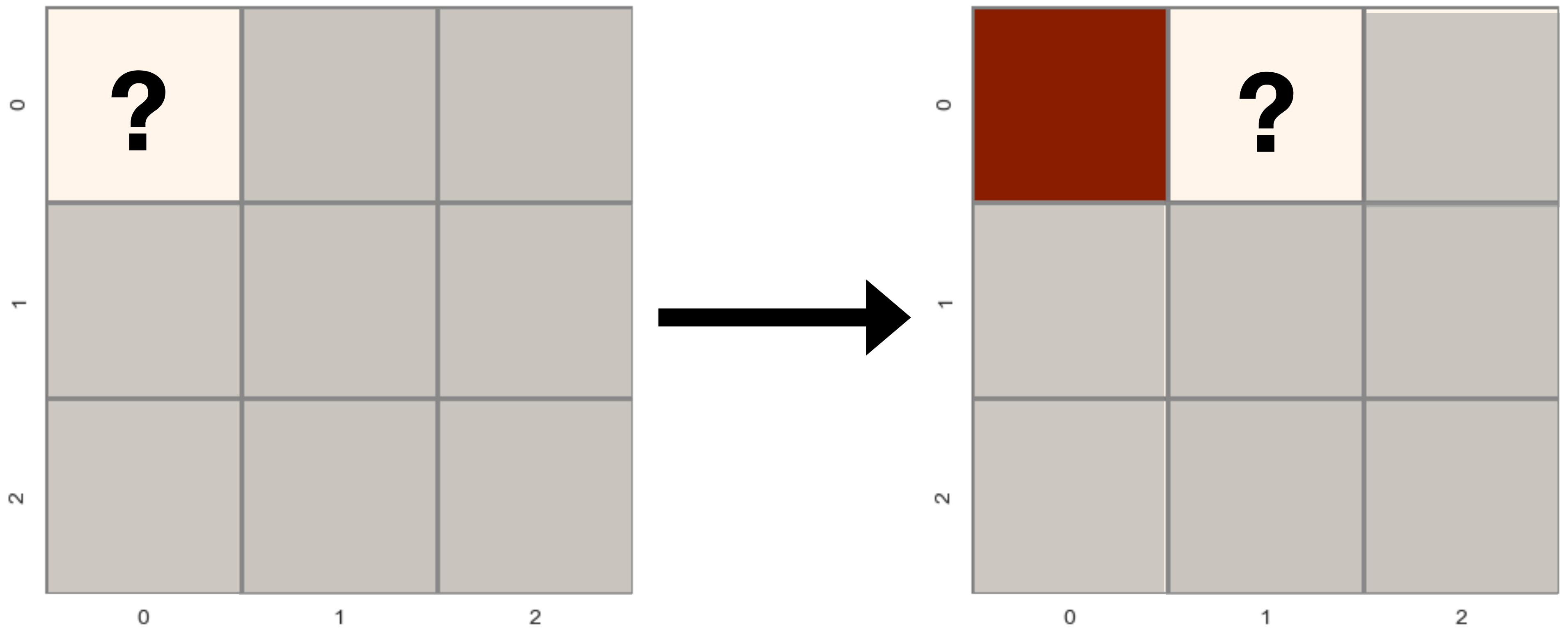
We start with no heuristic information



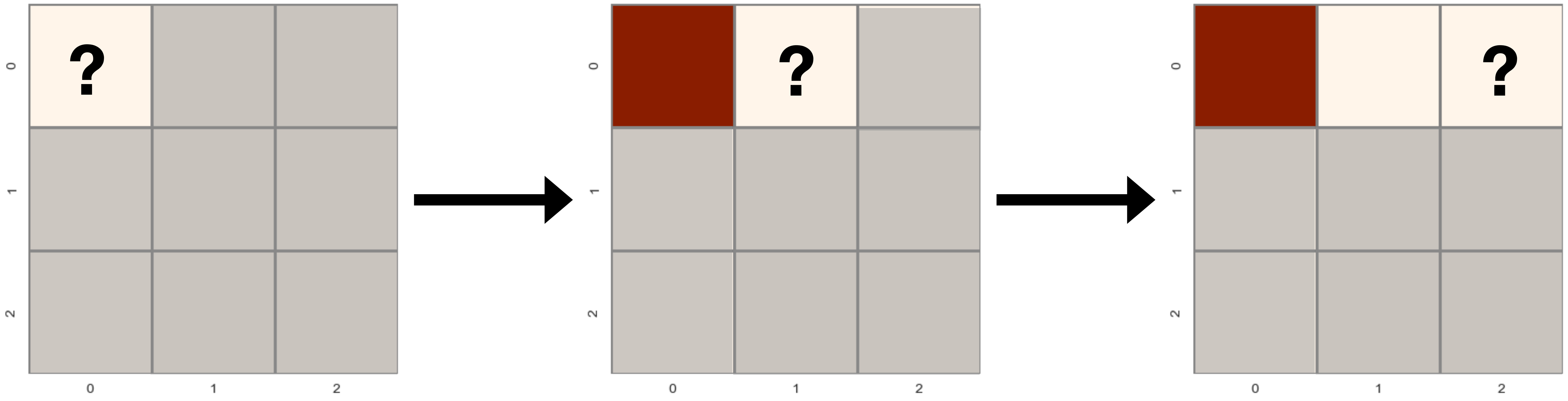
Algorithm Overview - Game setup



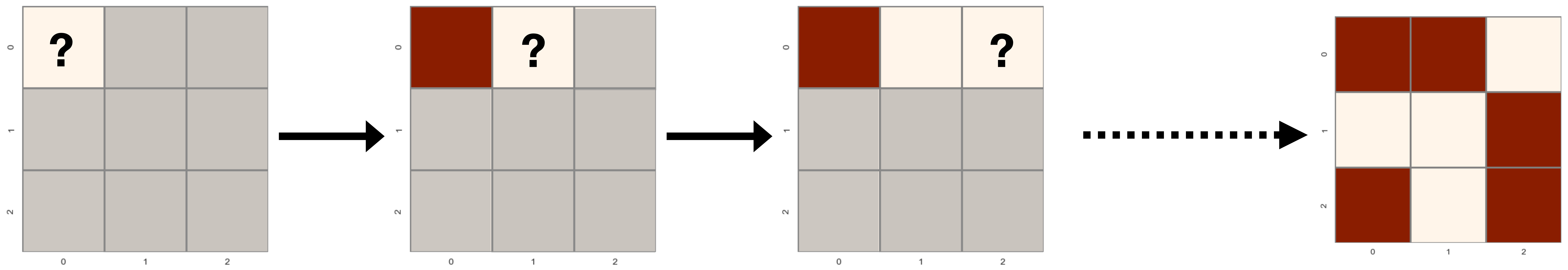
Algorithm Overview - Game setup



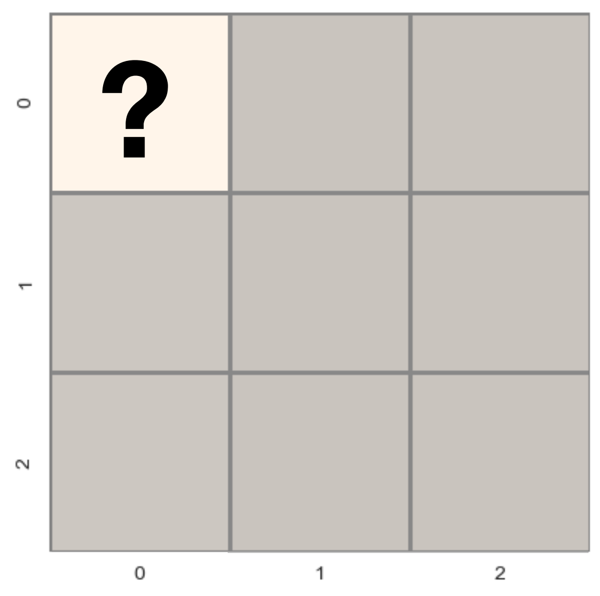
Algorithm Overview - Game setup



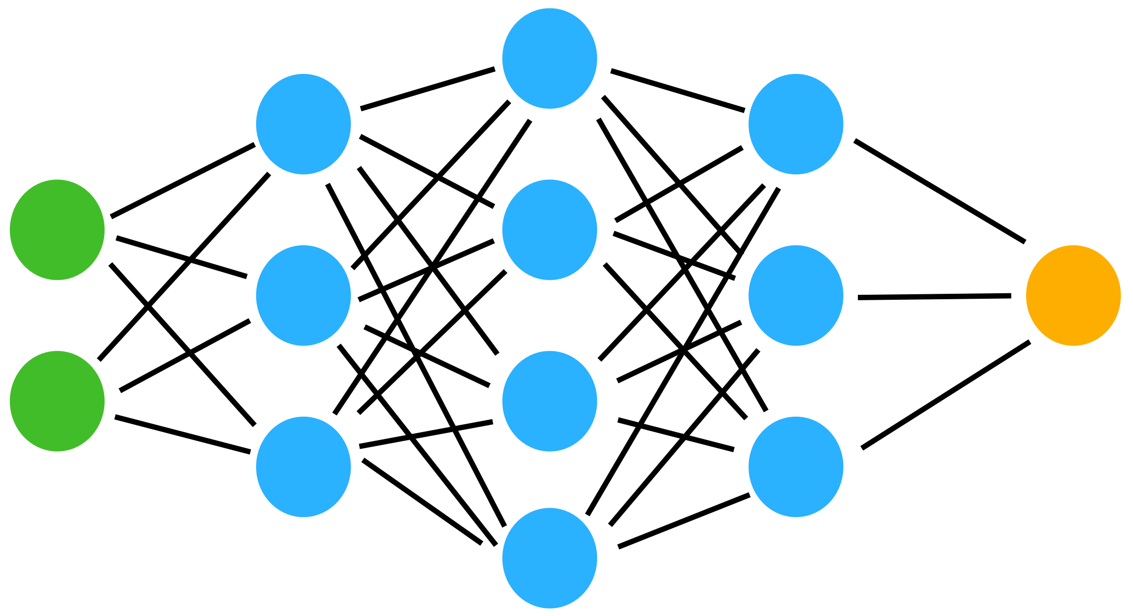
Algorithm Overview - Game setup



Algorithm Overview - Generation

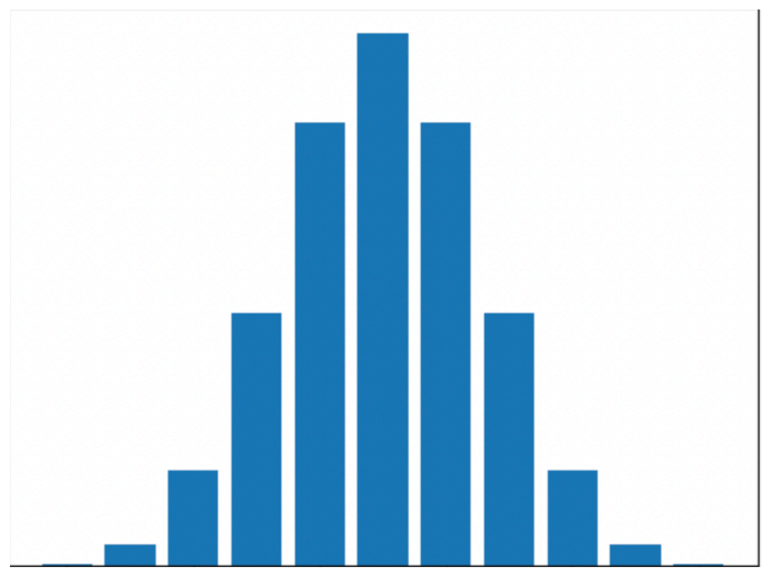


Input
State: 9 x 1 Vector
Position: 9 x 1 Vector

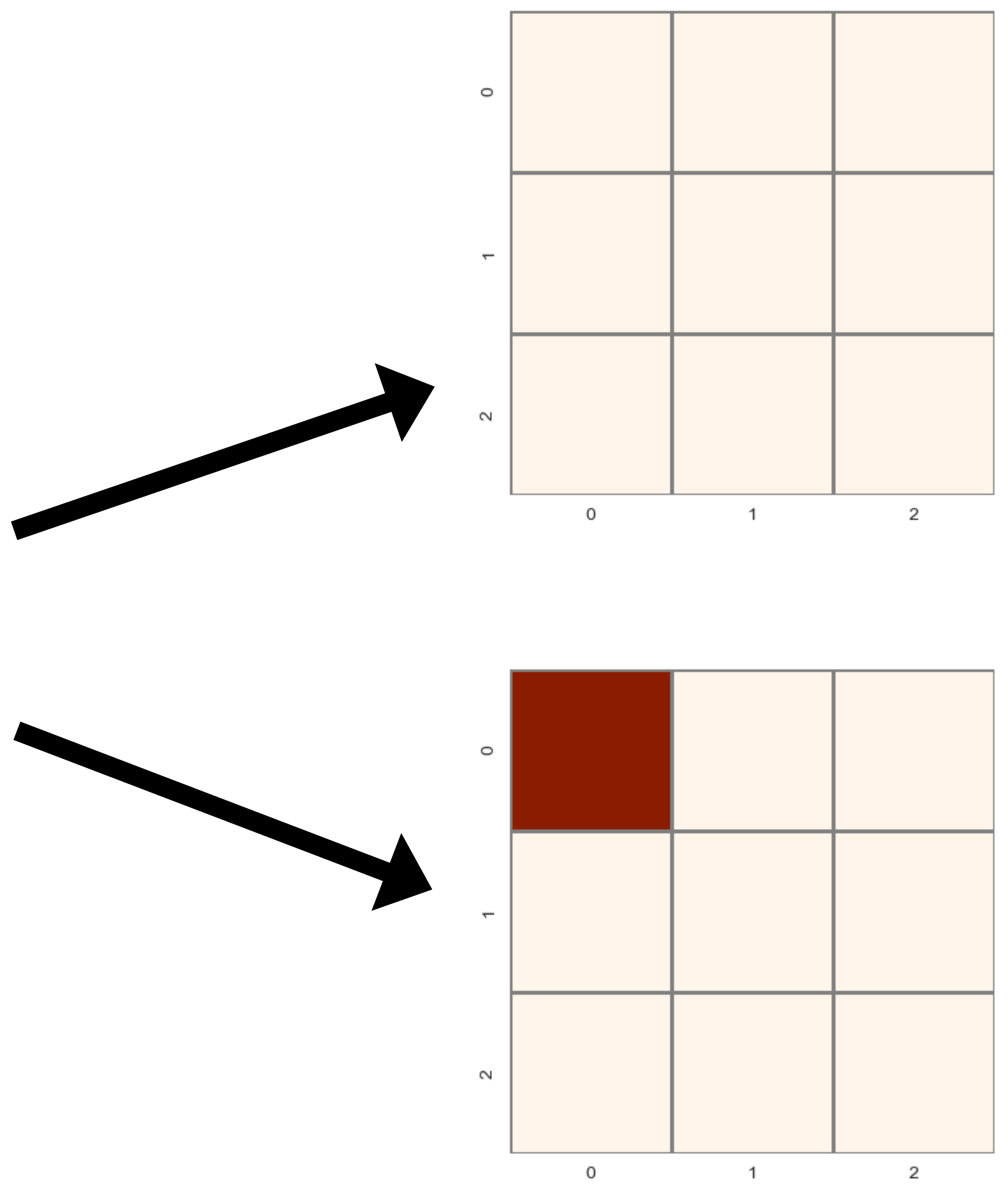


Feed Forward Step
3 Hidden Layers
(128, 64, 4)
Relu Hidden Activation
Sigmoid Output Activation

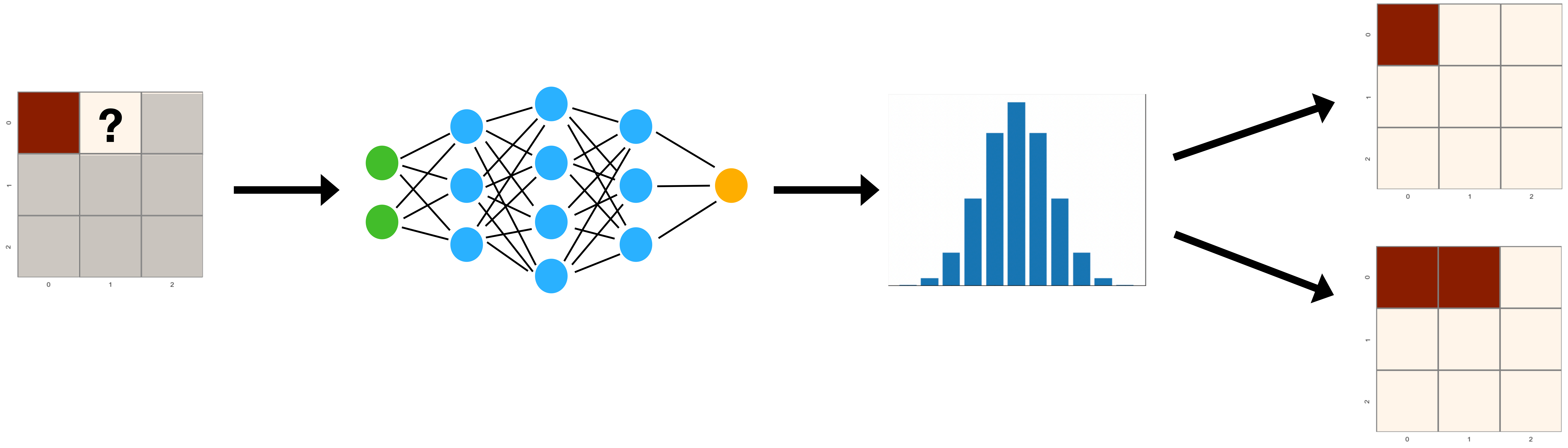
Note: NO TRAINING (yet)



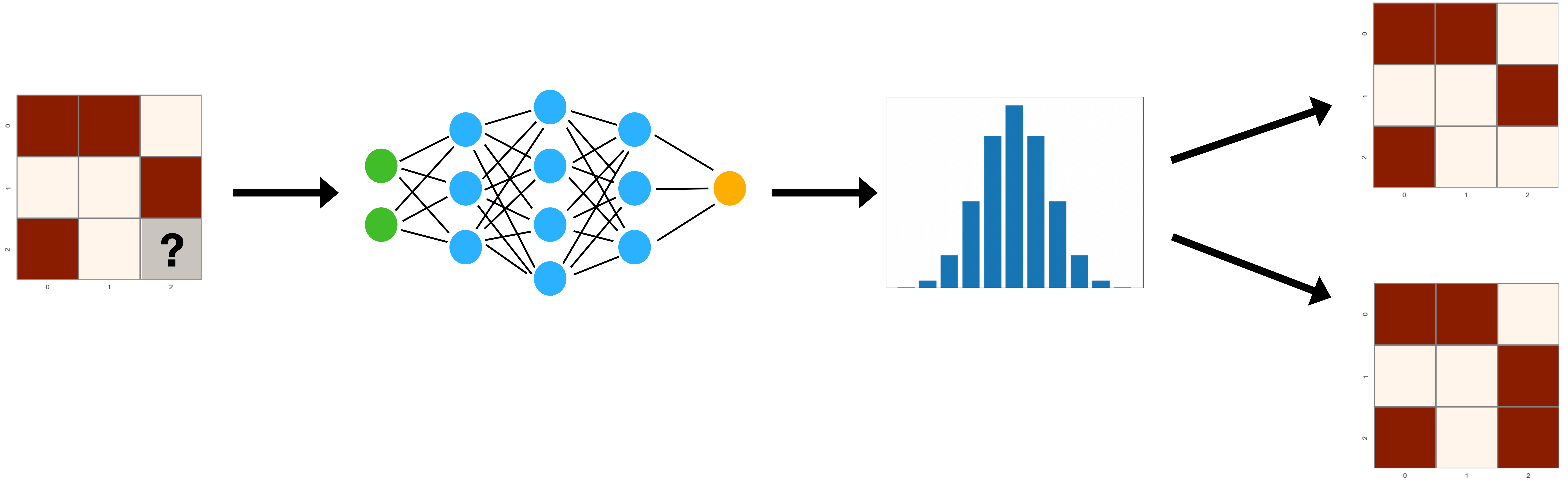
Output
Probability Distribution
(Binomial)



Algorithm Overview - Generation

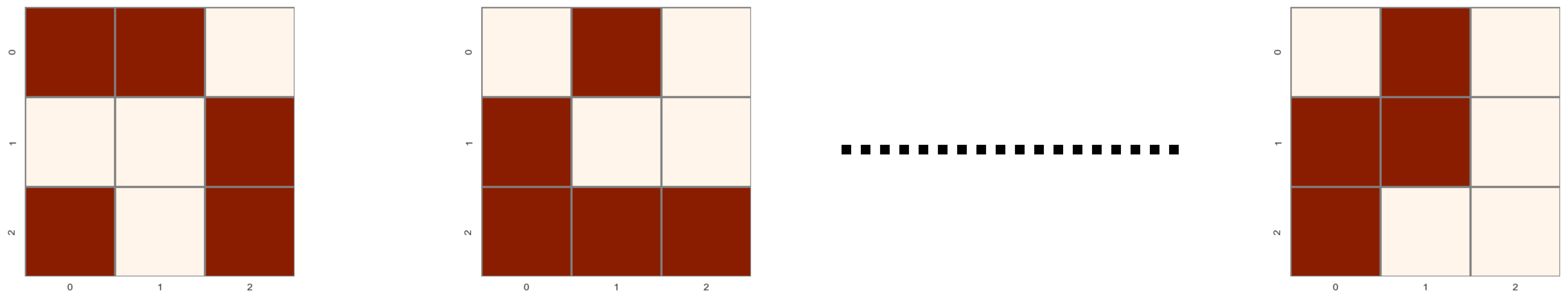


Algorithm Overview - Generation



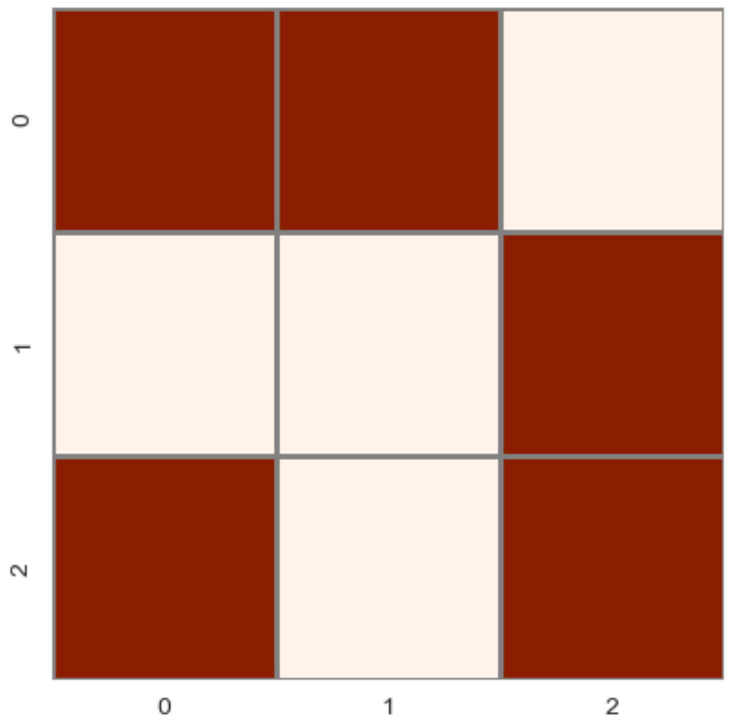
Algorithm Overview - Scoring

Generate lots of games (~2000)

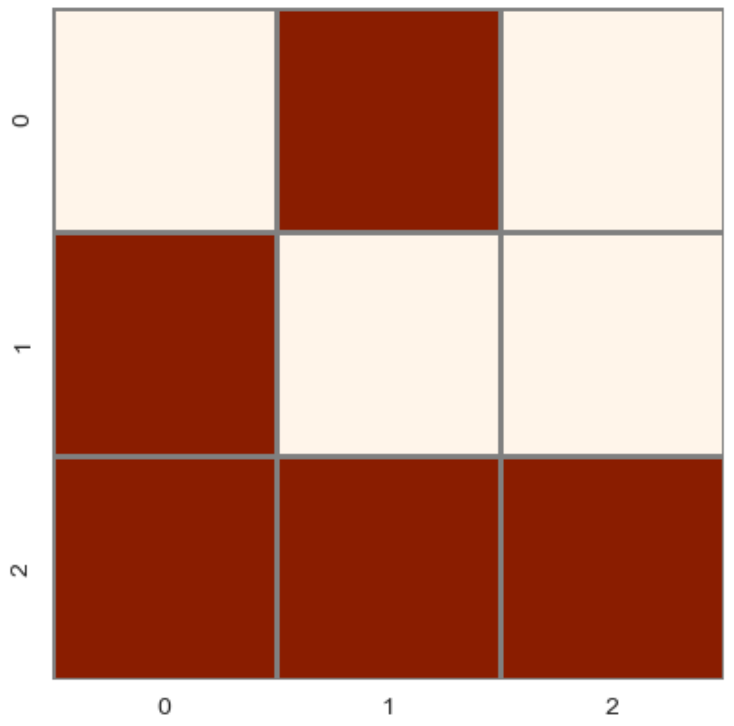


Algorithm Overview - Scoring

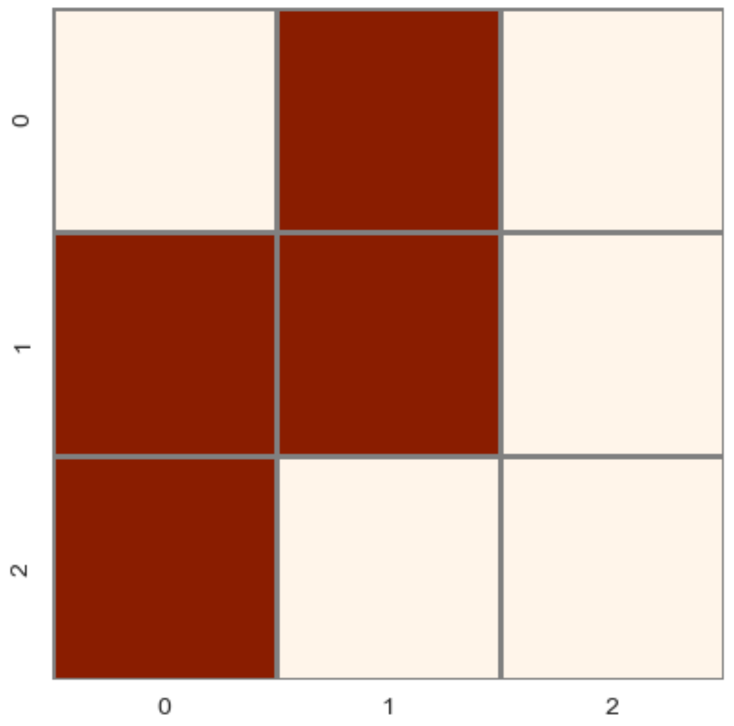
Generate lots of games (~2000)



Score = -0.5



Score = -3.5

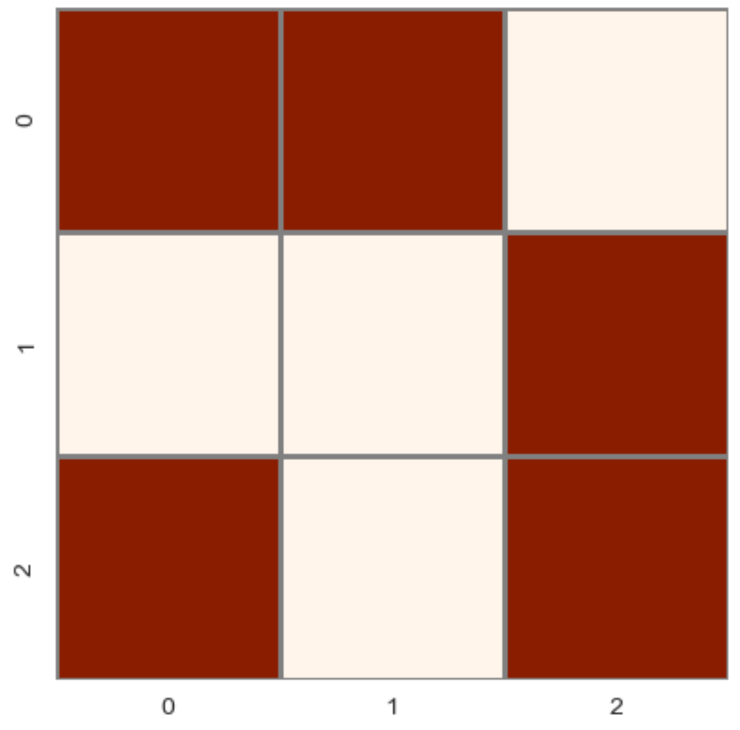


Score = -2

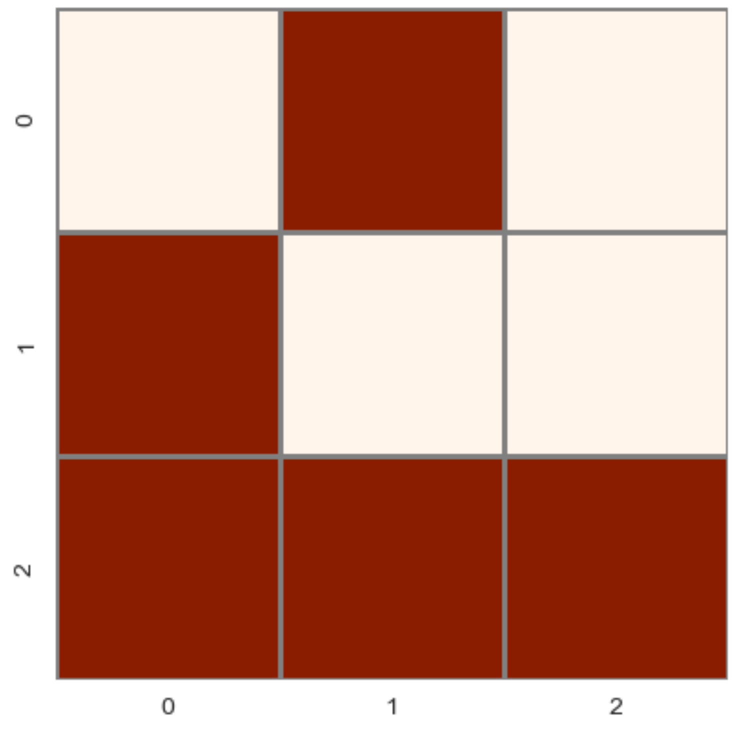
$$s(\cdot) = -\text{(\# of isosceles }\Delta\text{'s)}$$

Algorithm Overview - Scoring

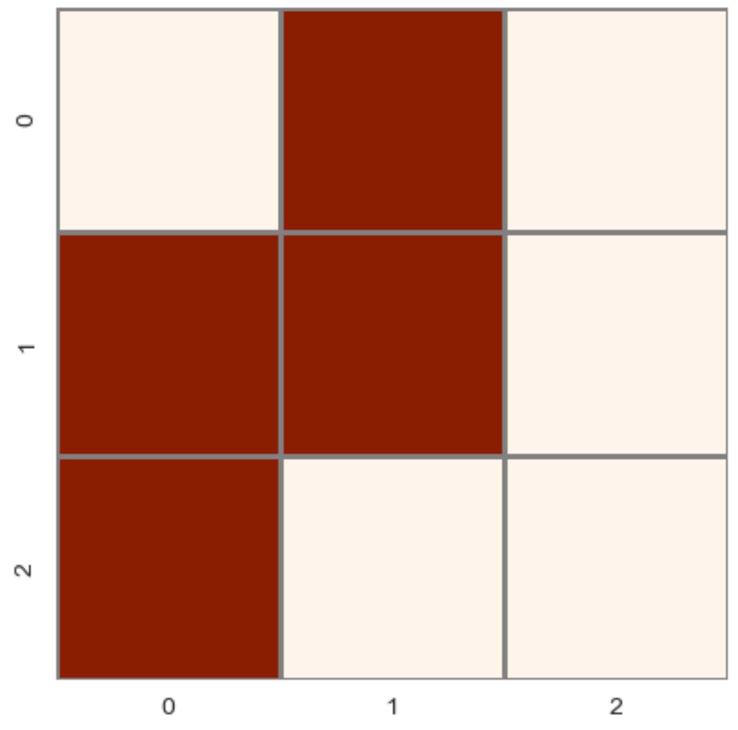
Generate lots of games (~2000)



Score = -0.5



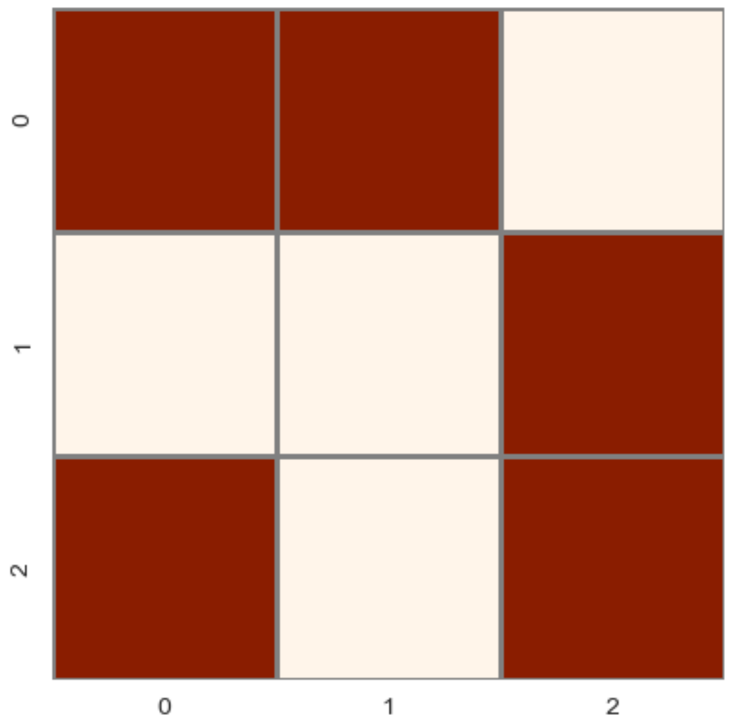
Score = -3.5



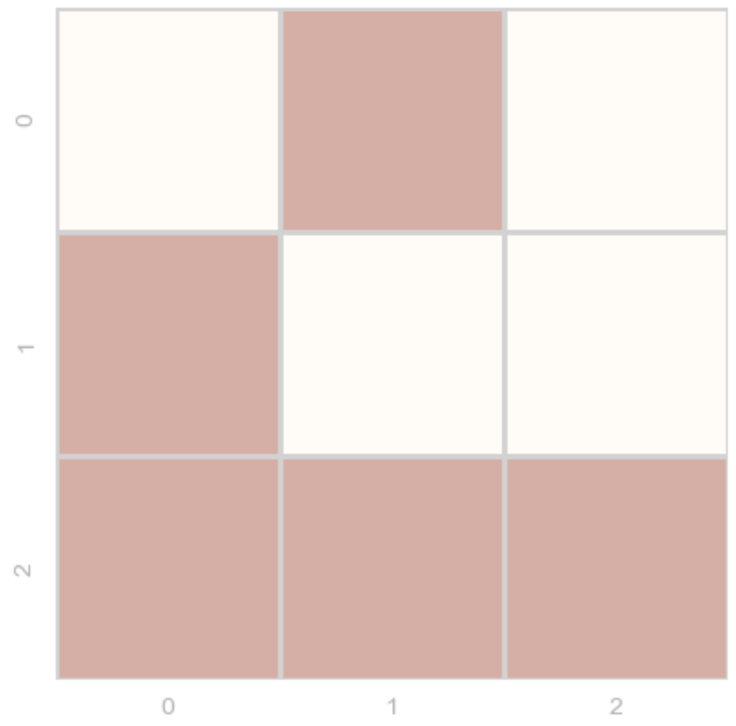
Score = -2

$$s(\cdot) = -\text{(\# of isosceles } \Delta\text{'s)} + \lambda \cdot \text{(\# of points)}$$

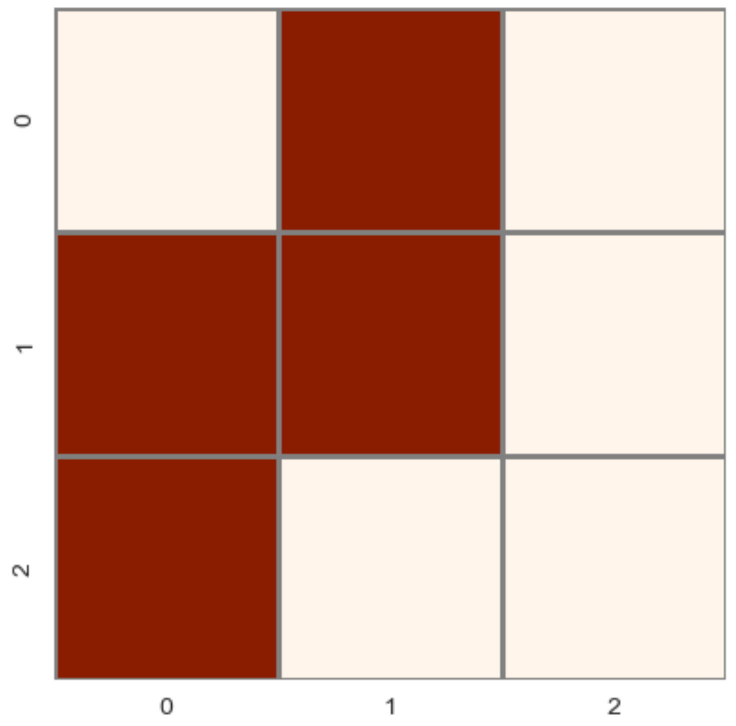
Algorithm Overview - Select Best



Score = -0.5

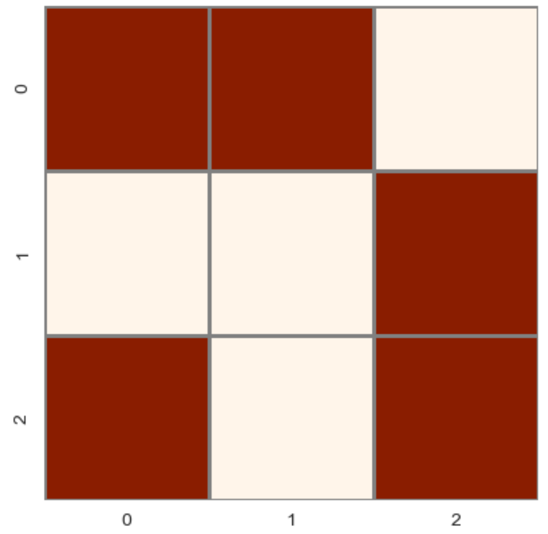


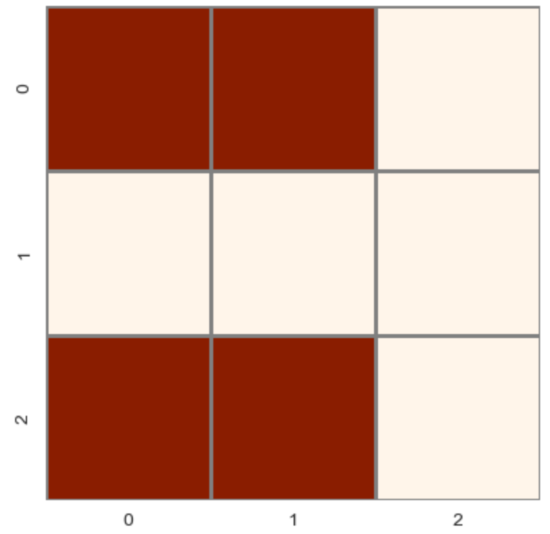
Score = -3.5

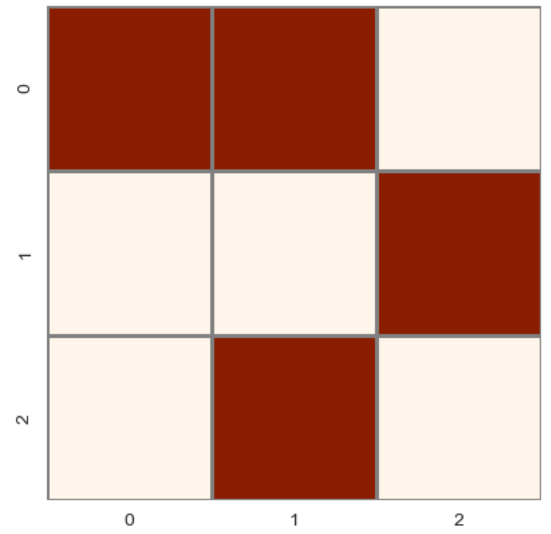


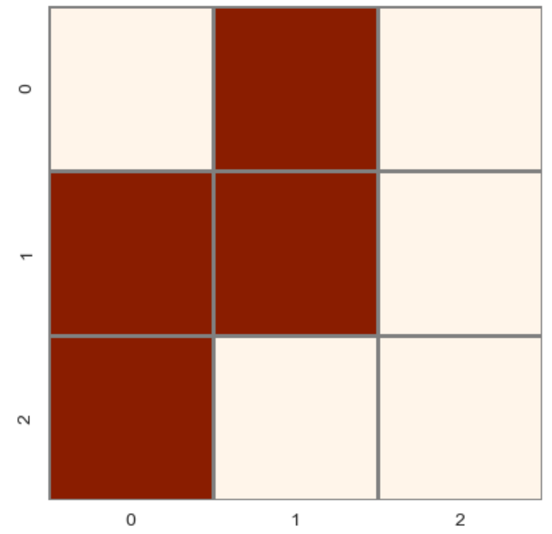
Score = -2

Best Games: Top k percent (Usually ~200 games, i.e. k = 10)



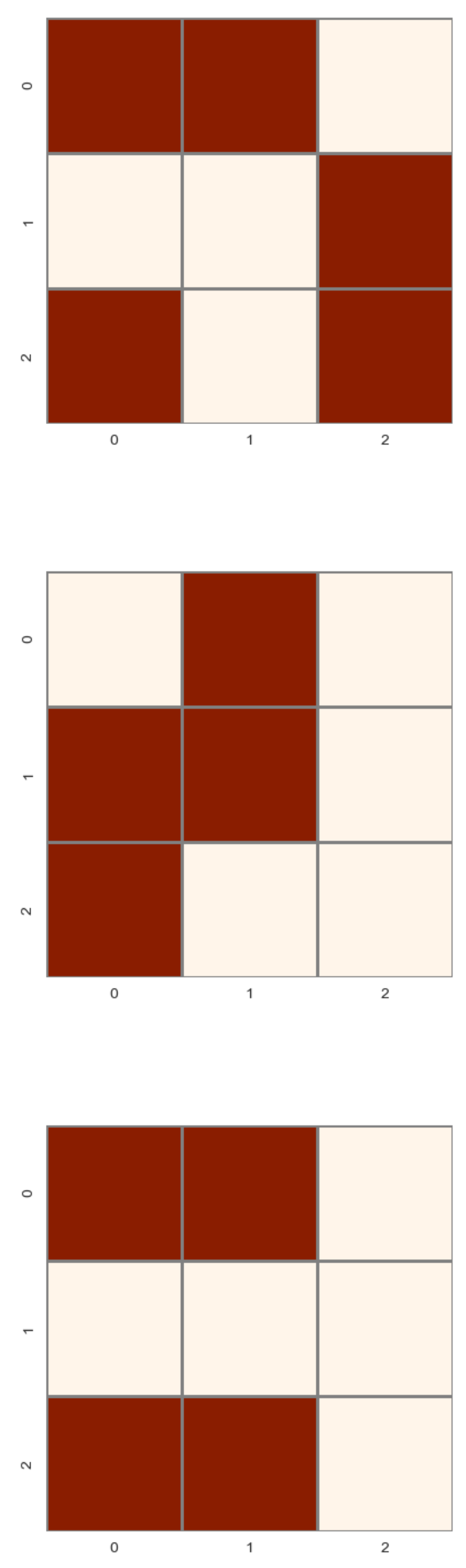


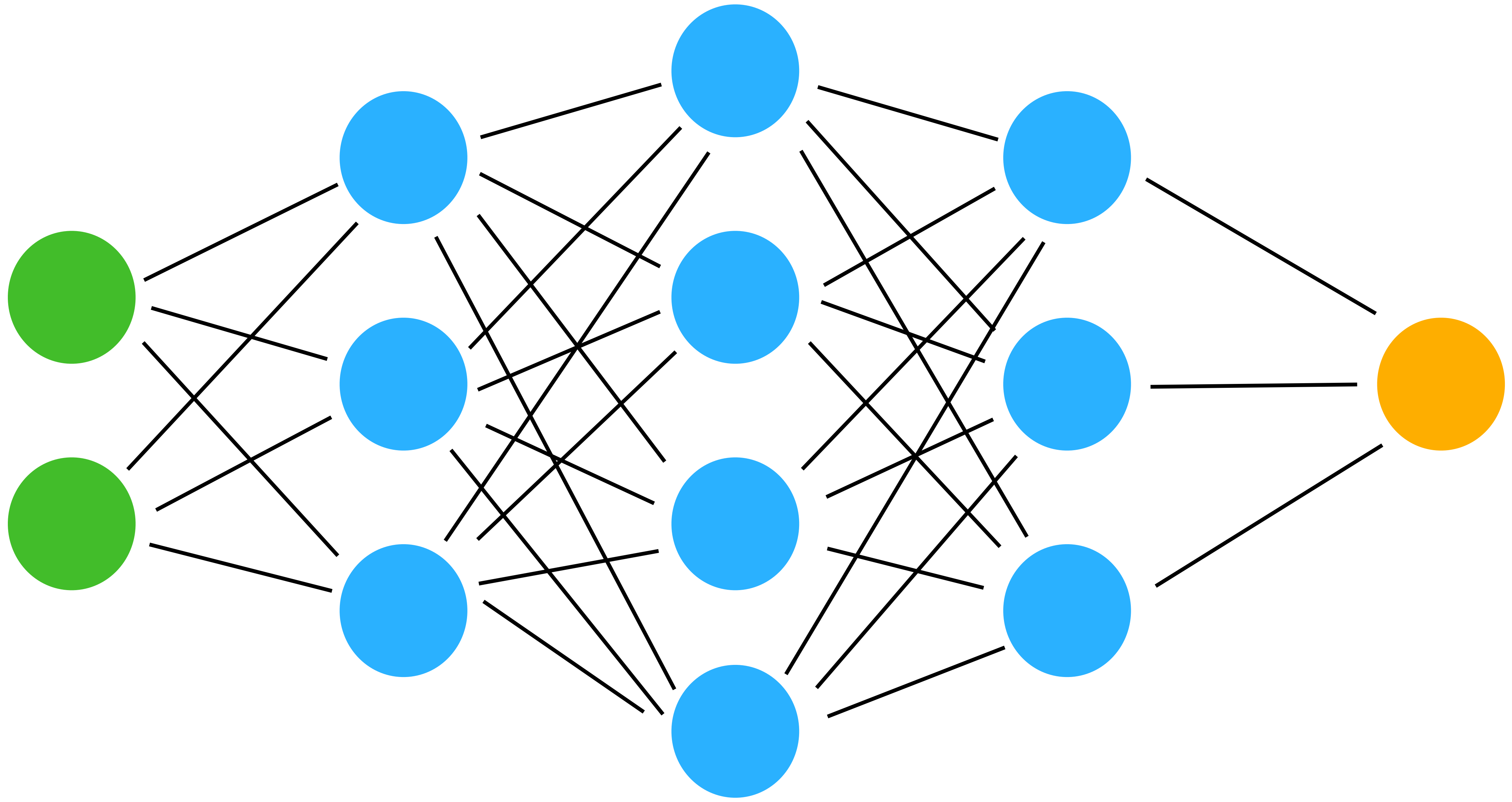




Algorithm Overview - Training Network

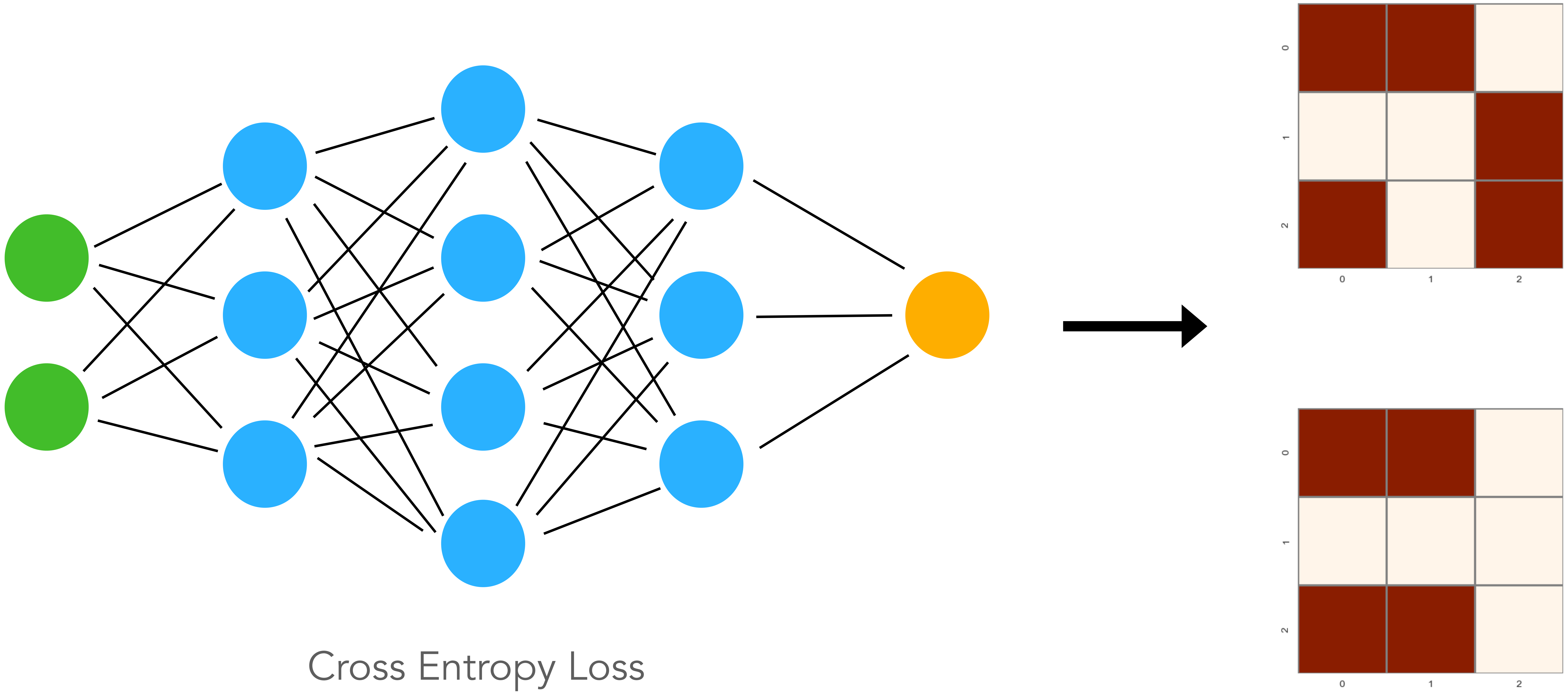
Best Games





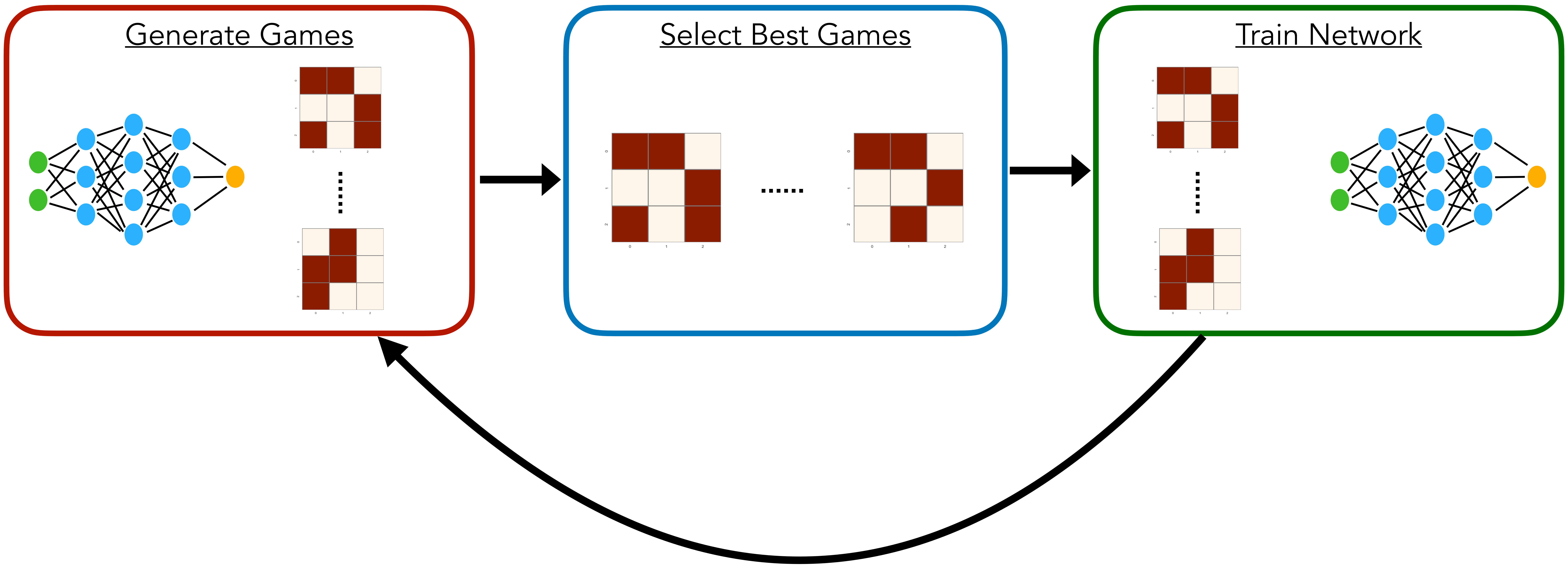
Cross Entropy Loss
Adam Optimizer

Algorithm Overview - Back to Generation



Cross Entropy Loss
Adam Optimizer

Algorithm Overview - Summary



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.

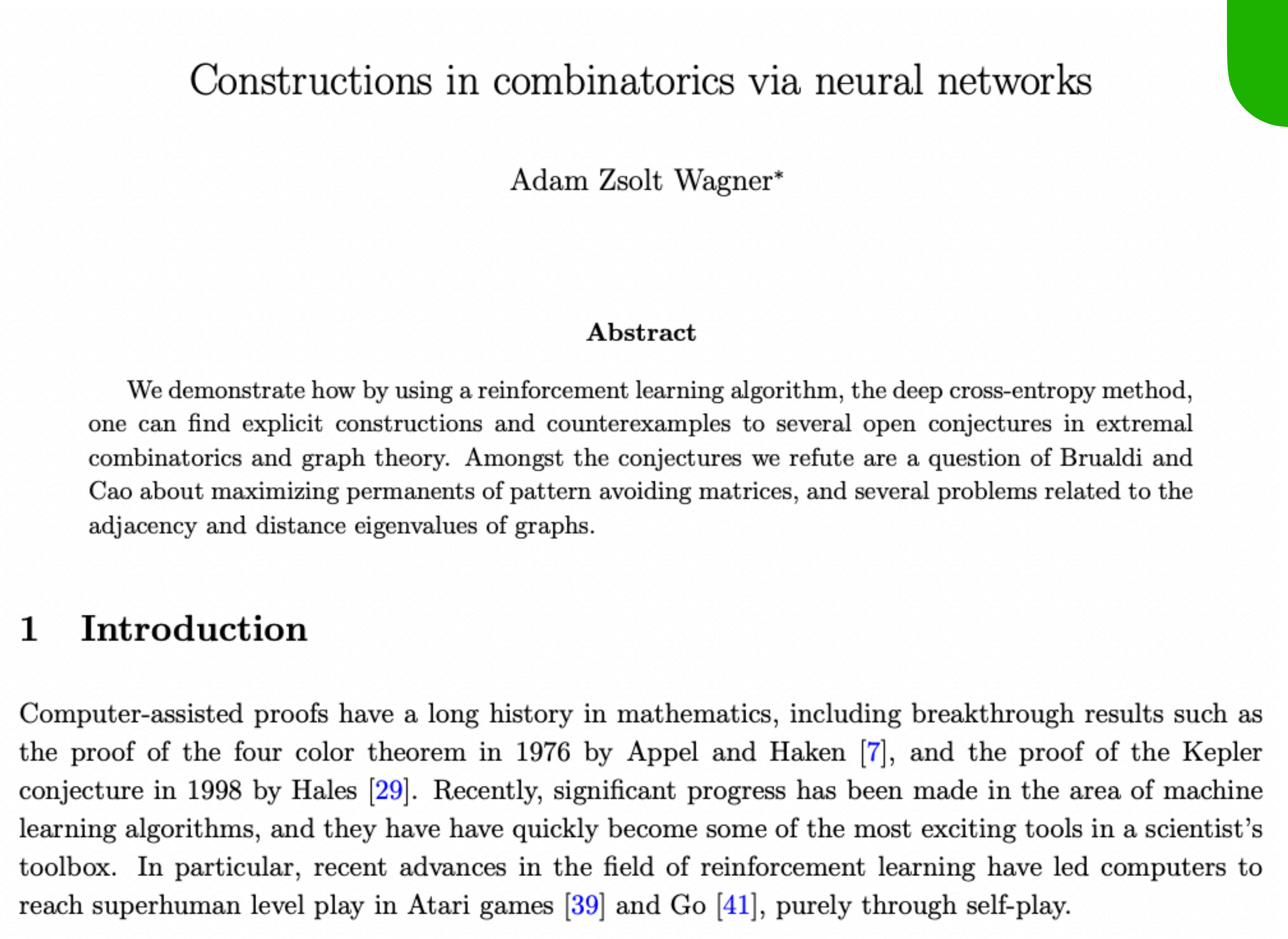
1 Introduction

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.

Aim: Use this algorithm to generate counterexamples to conjectures in combinatorics

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.

Aim: Use this algorithm to generate counterexamples to conjectures in combinatorics

<u>Example Conjecture 1</u>

For any graph G with n vertices, we have,

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

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

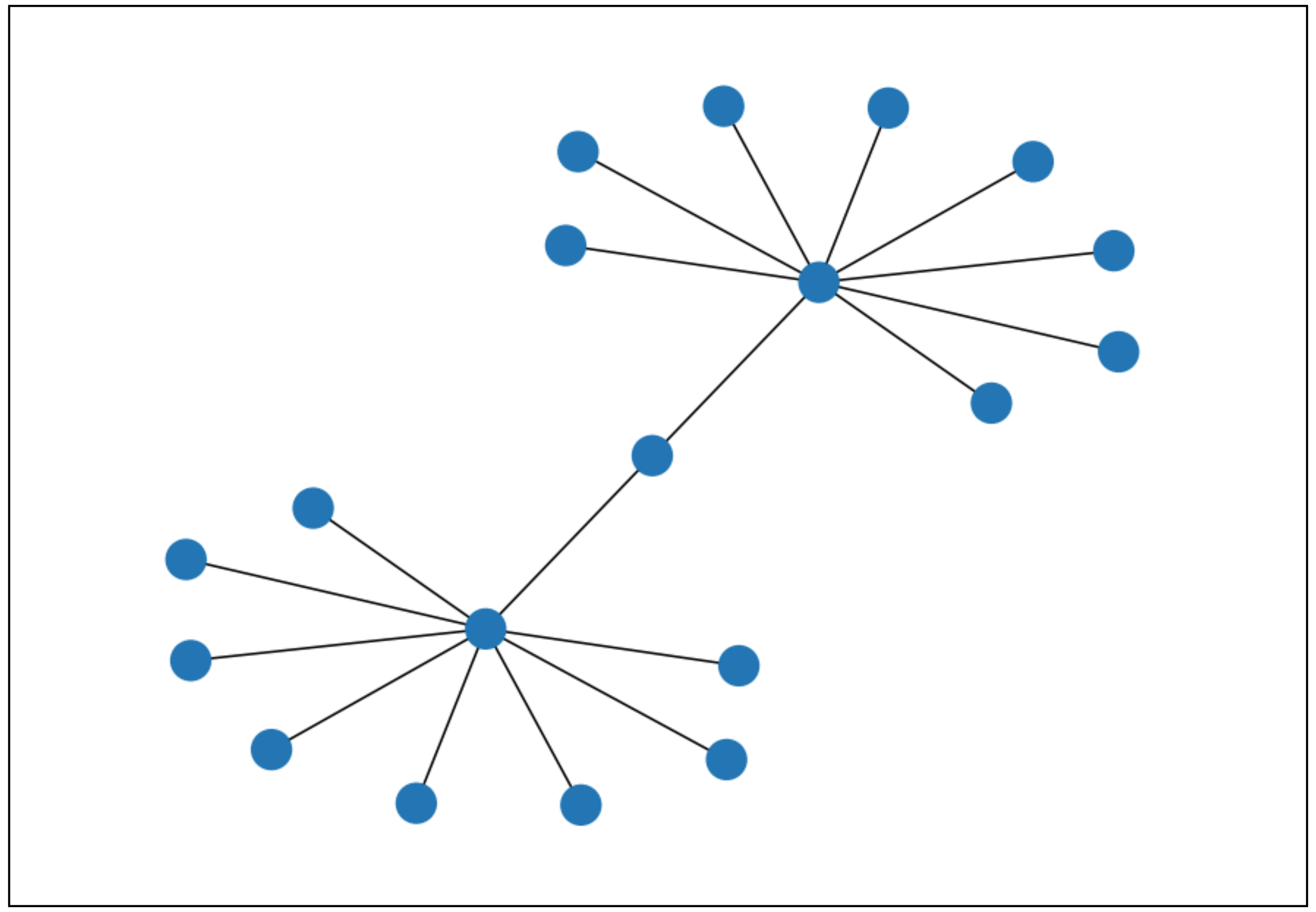
Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning 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.

Aim: Use this algorithm to generate counterexamples to conjectures in combinatorics

Example Conjecture 1

For any graph G with n vertices, we have,

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



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.

Aim: Use this algorithm to generate counterexamples to conjectures in combinatorics

Example Conjecture 2

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

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.

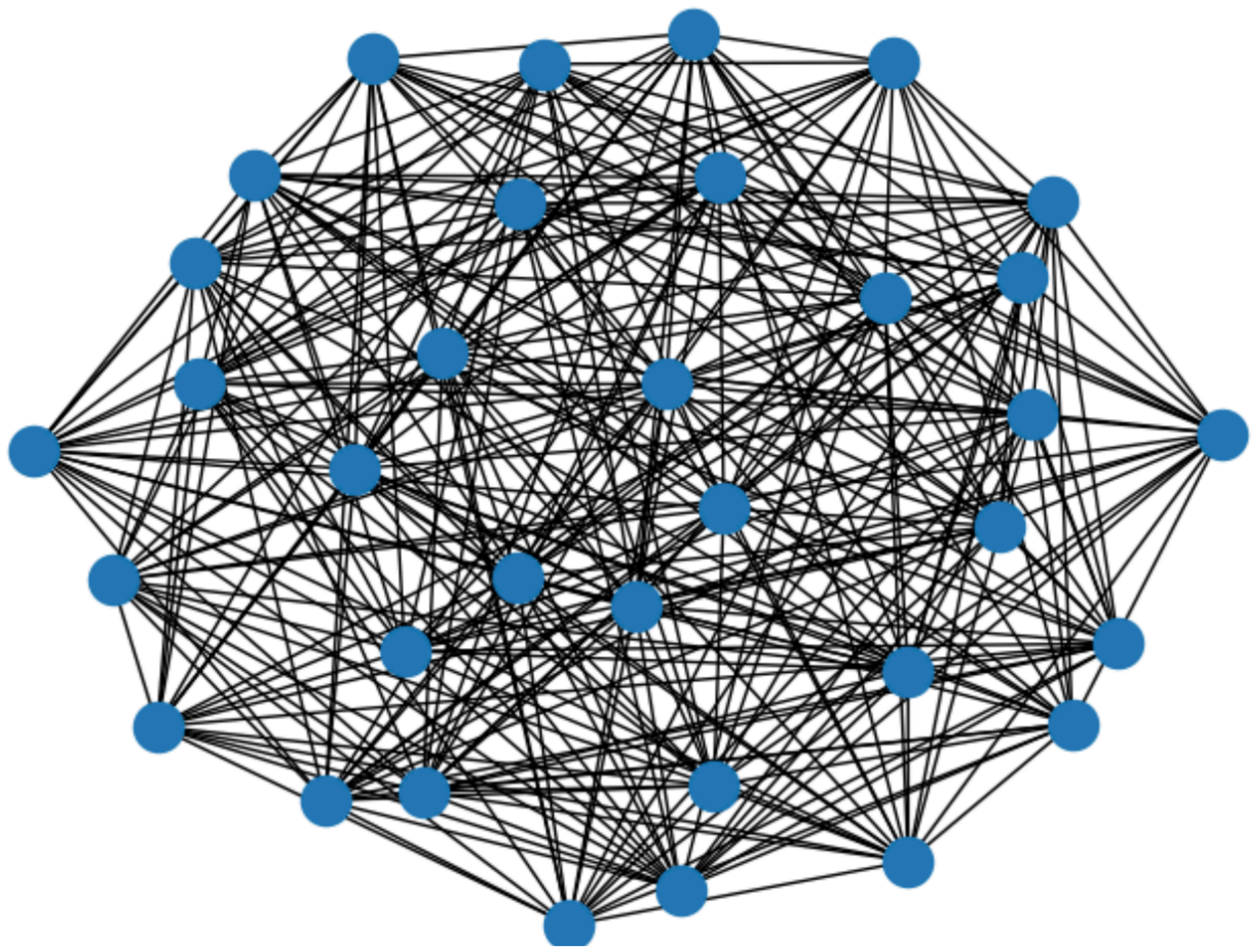
1 Introduction

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.

Aim: Use this algorithm to generate counterexamples to conjectures in combinatorics

Example Conjecture 2

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



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.

Aim: Use this algorithm to generate counterexamples to conjectures in combinatorics

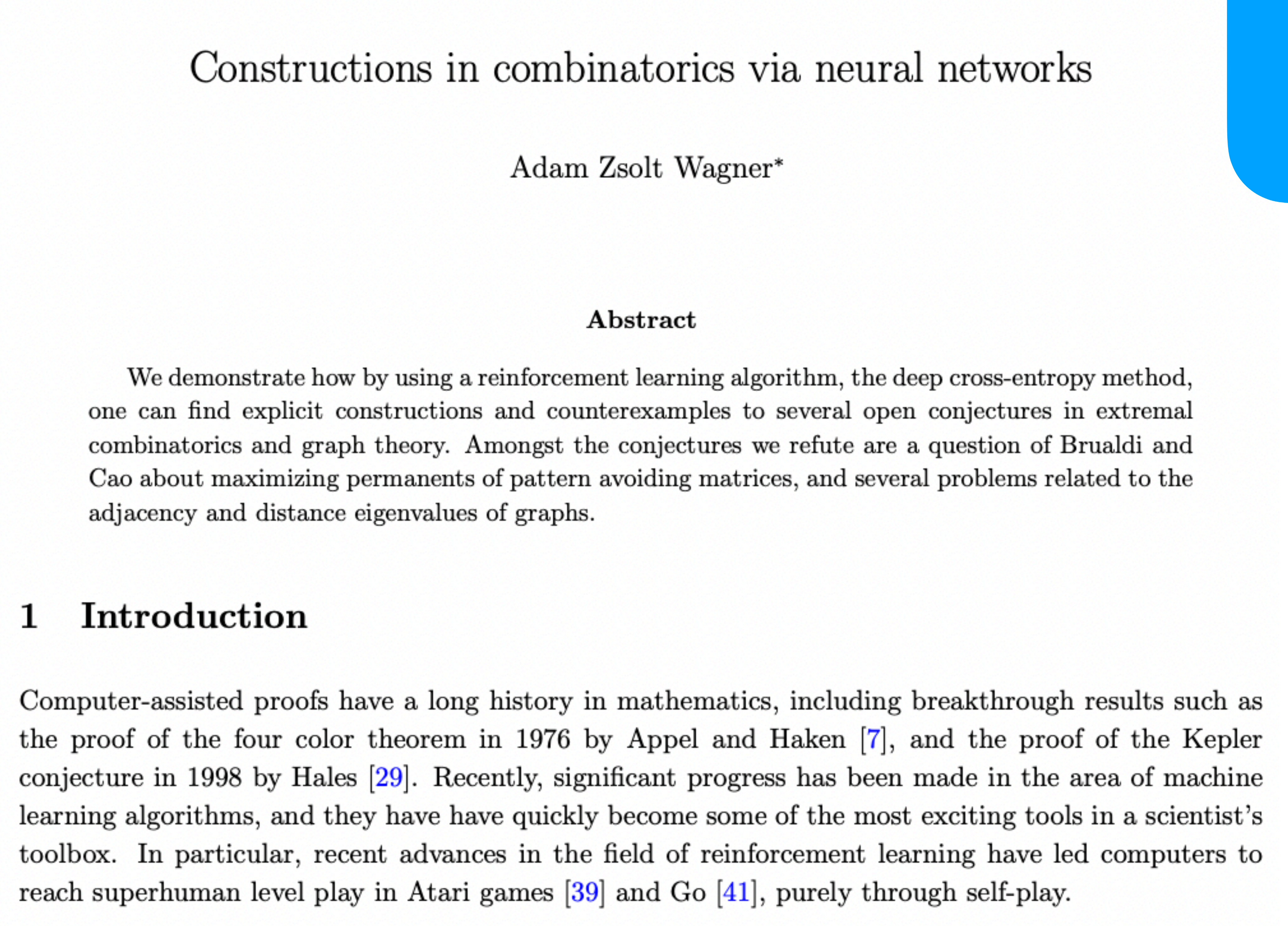
Example Conjecture 3

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

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

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

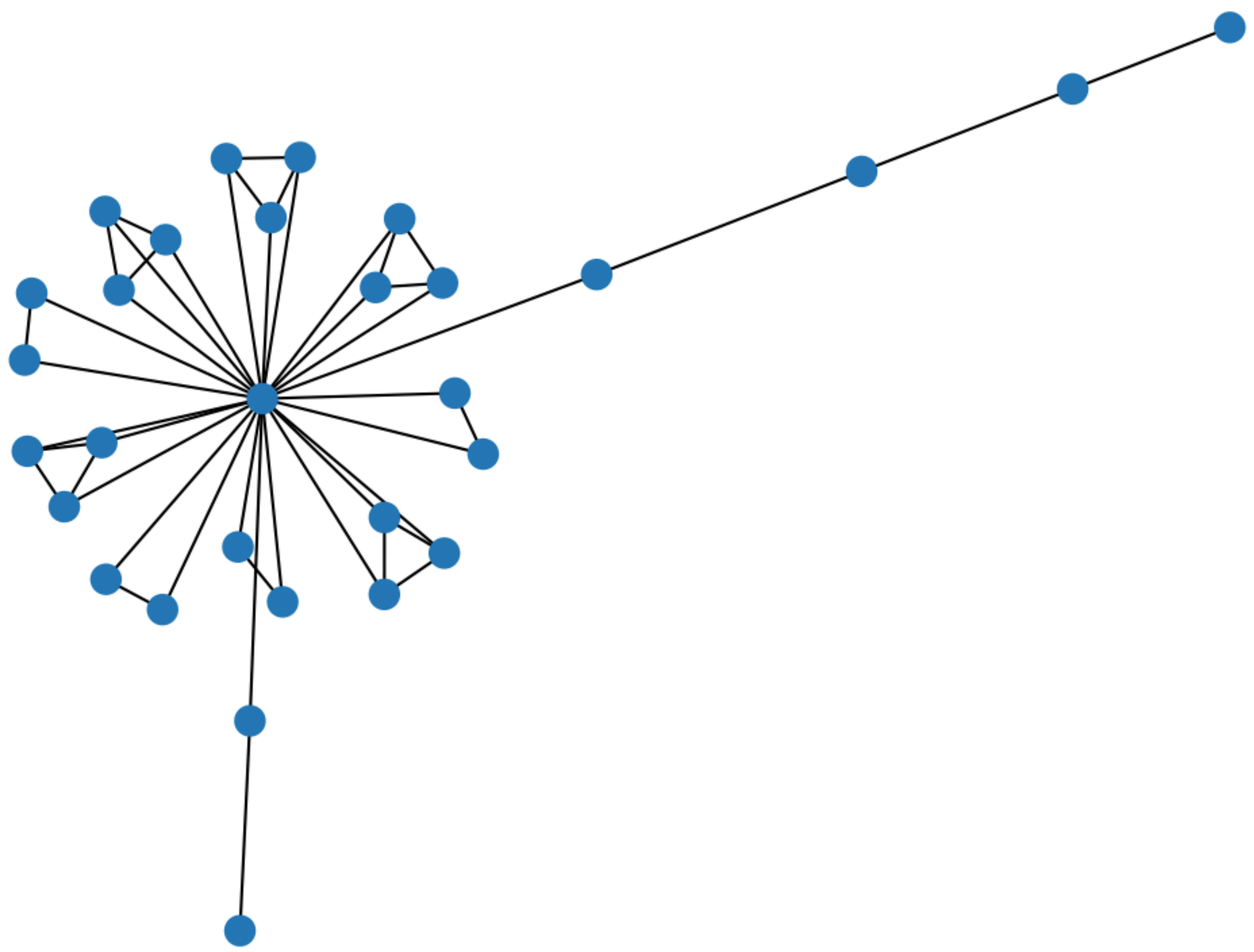
Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning 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.

Aim: Use this algorithm to generate counterexamples to conjectures in combinatorics

<u>Example Conjecture 3</u>

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

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

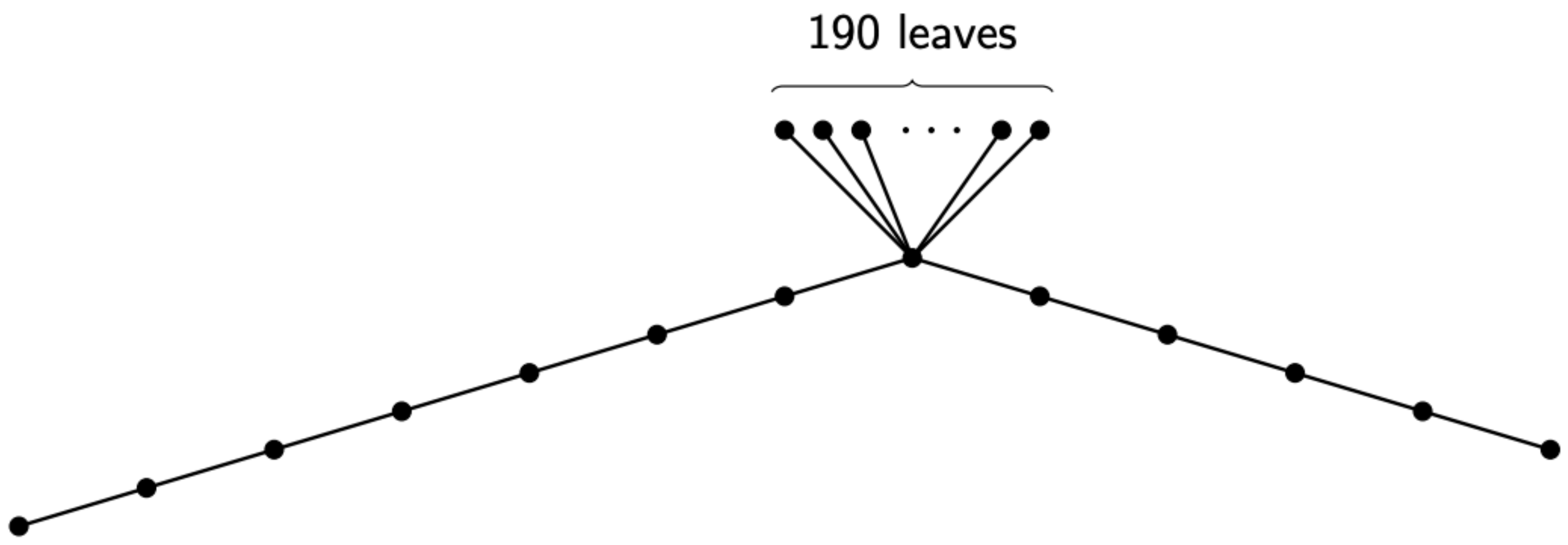
Computer-assisted proofs have a long history in mathematics, including breakthrough results such as the proof of the four color theorem in 1976 by Appel and Haken [7], and the proof of the Kepler conjecture in 1998 by Hales [29]. Recently, significant progress has been made in the area of machine learning 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.

Aim: Use this algorithm to generate counterexamples to conjectures in combinatorics

<u>Example Conjecture 3</u>

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.

Aim: Use this algorithm to generate counterexamples to conjectures in combinatorics

Immediate Counterexample

Not a Counterexample and / or not insightful

Almost a Counterexample
But was able to extend to counterexample

Overview

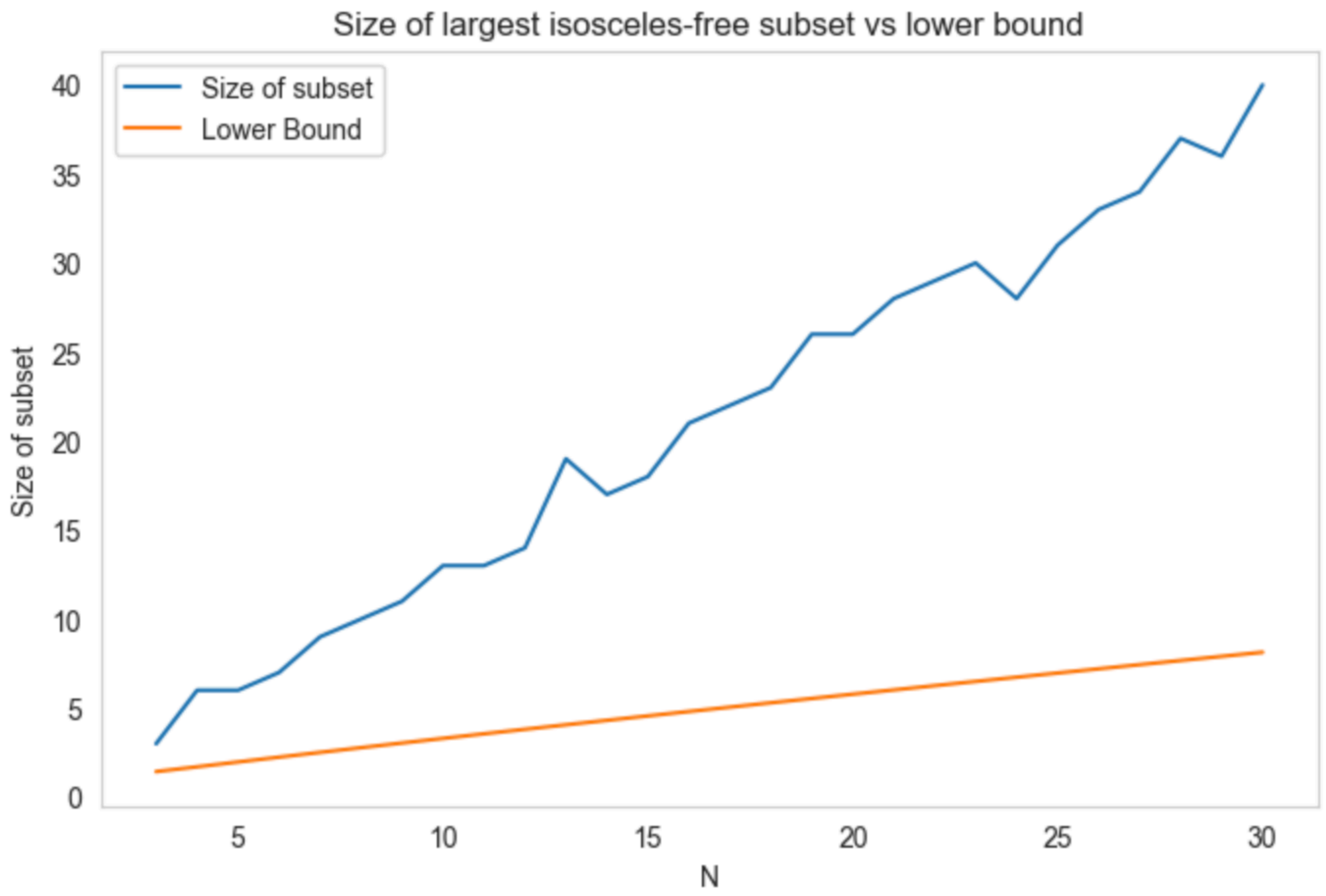
Mathematical Motivation and Background

- Motivation: Non Metric Multidimensional Scaling
- Key definitions and propositions
- Known bounds for the problem

How Reinforcement Learning can help

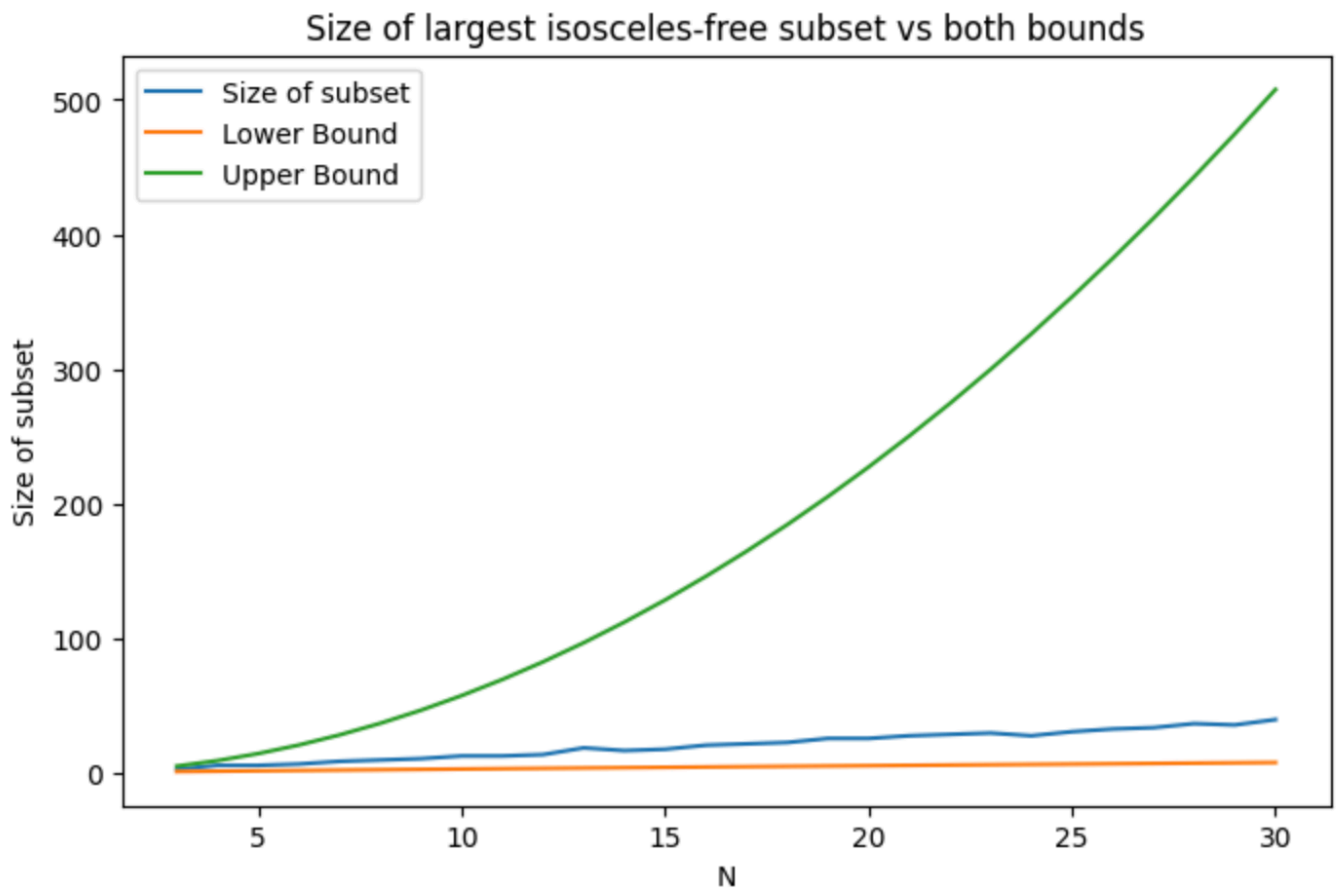
- Reinforcement learning background and main algorithm
- Current results and observations
- Next Steps

Results



Evidence that we can do much better than the current lower bound

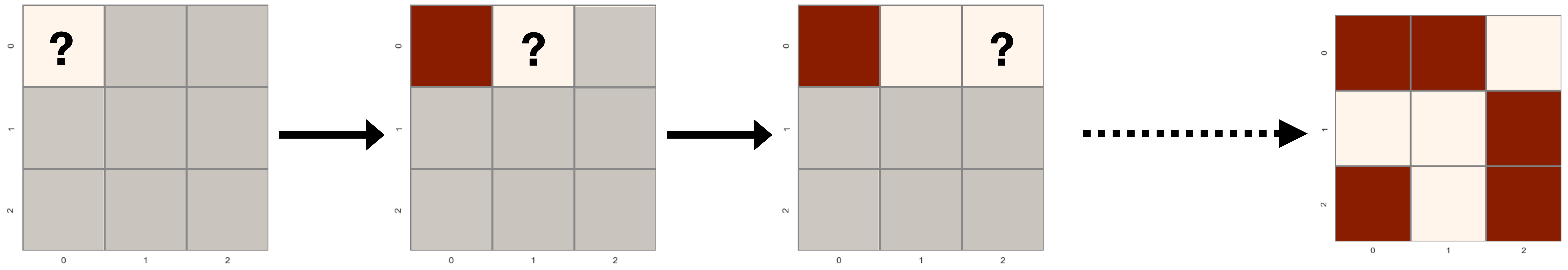
Results



Evidence also shows that we don’t talk about upper bounds…
(Room for improvement exists)

Things we've thought about along the way

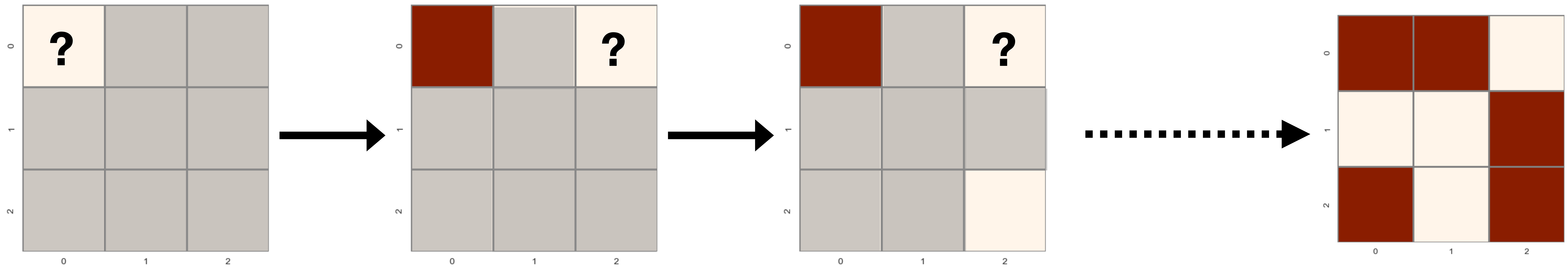
Does the order of how you input the points matter?



Things we've thought about along the way

Does the order of how you input the points matter?

- Turns out no.

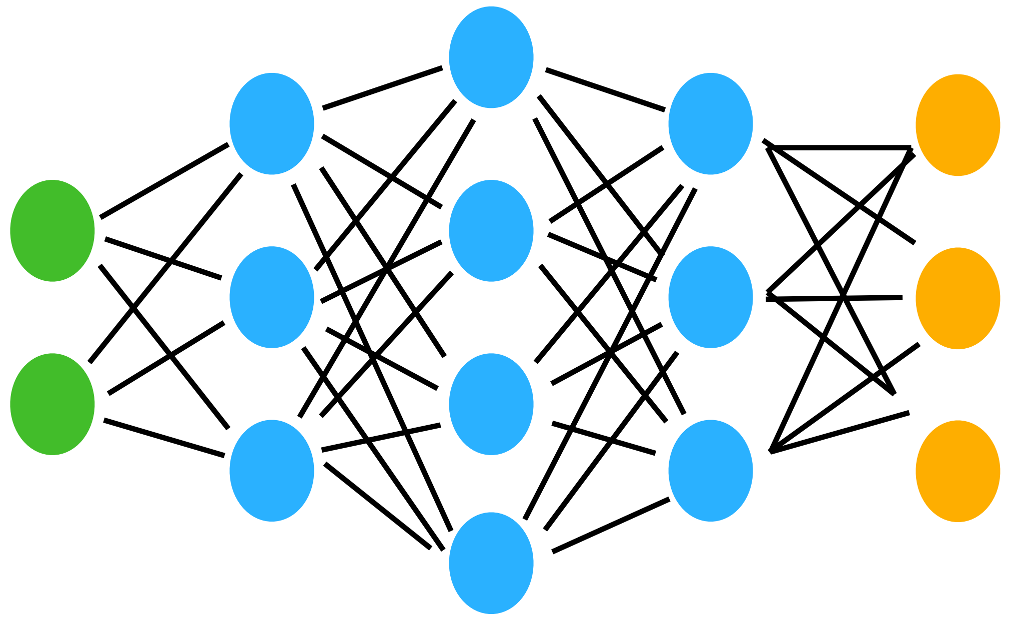


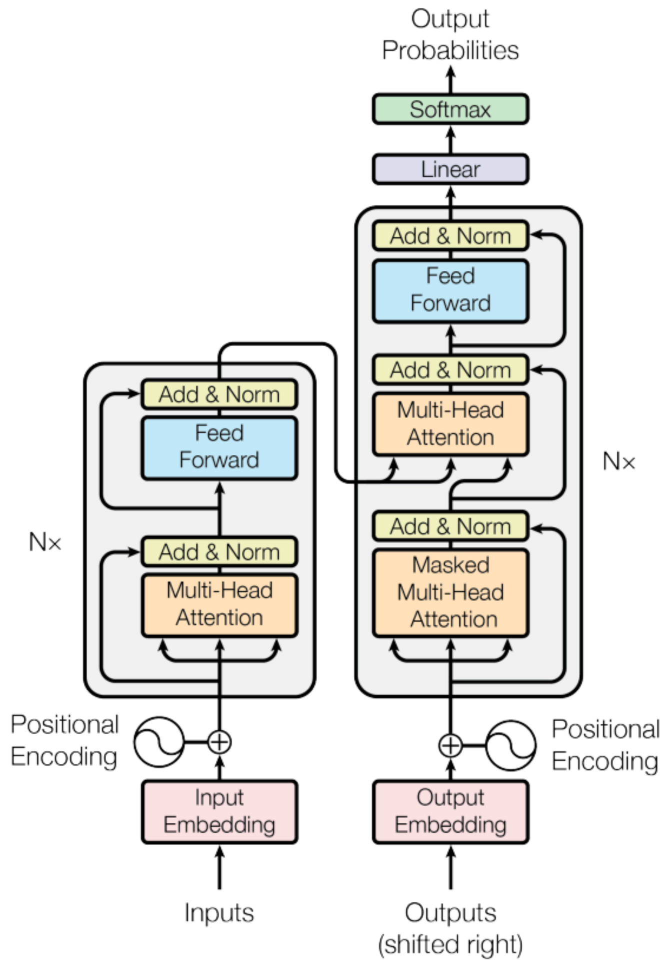
Things we've thought about along the way

Does the order of how you input the points matter?

- Turns out no.

What would happen if we used different model architectures





Transformers

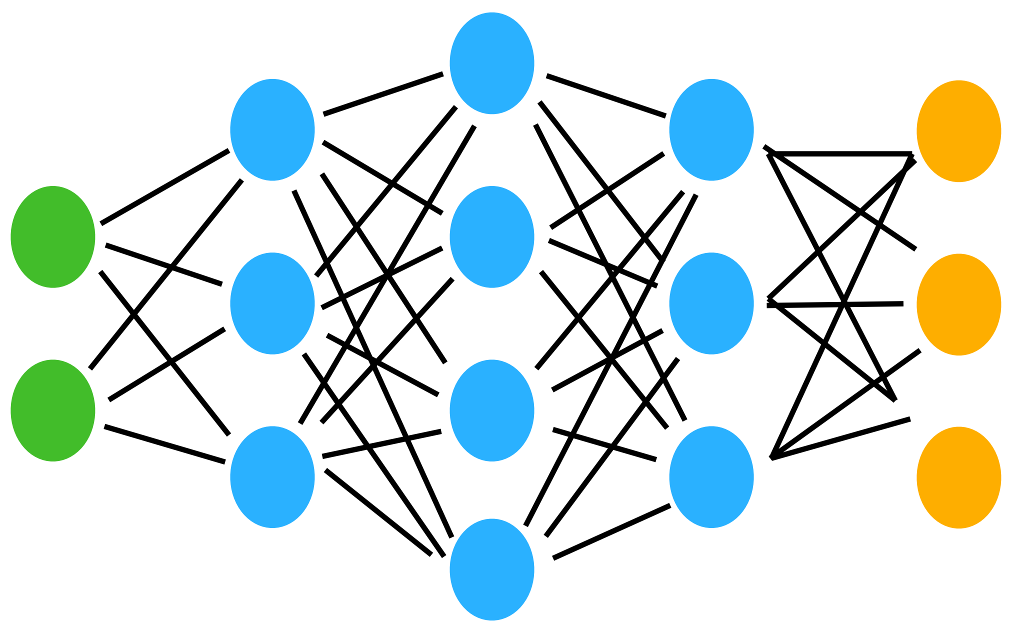
Things we've thought about along the way

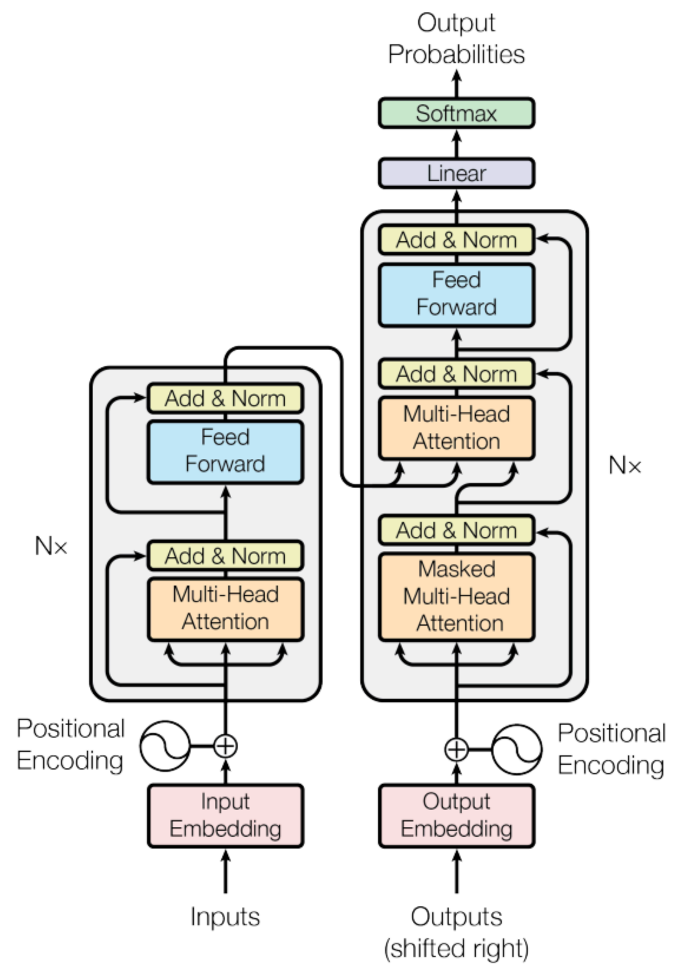
Does the order of how you input the points matter?

- Turns out no.

What would happen if we used different model architectures?

- Does change performance, we will see an example later





Transformers

Things we've thought about along the way

Does the order of how you input the points matter?

- Turns out no.

What would happen if we used different model architectures?

- Does change performance, we will see an example later

What kind of heuristic information can we add?

Things we've thought about along the way

Does the order of how you input the points matter?

- Turns out no.

What would happen if we used different model architectures?

- Does change performance, we will see an example later

What kind of heuristic information can we add?

- Best boards include patterns like symmetries, fewer dominos (adjacent points), and more points closer to the edge

Results

With no heuristics:

For large boards
(e.g. 64 x 64)

Found largest
known generations

64 x 64: 108 Points

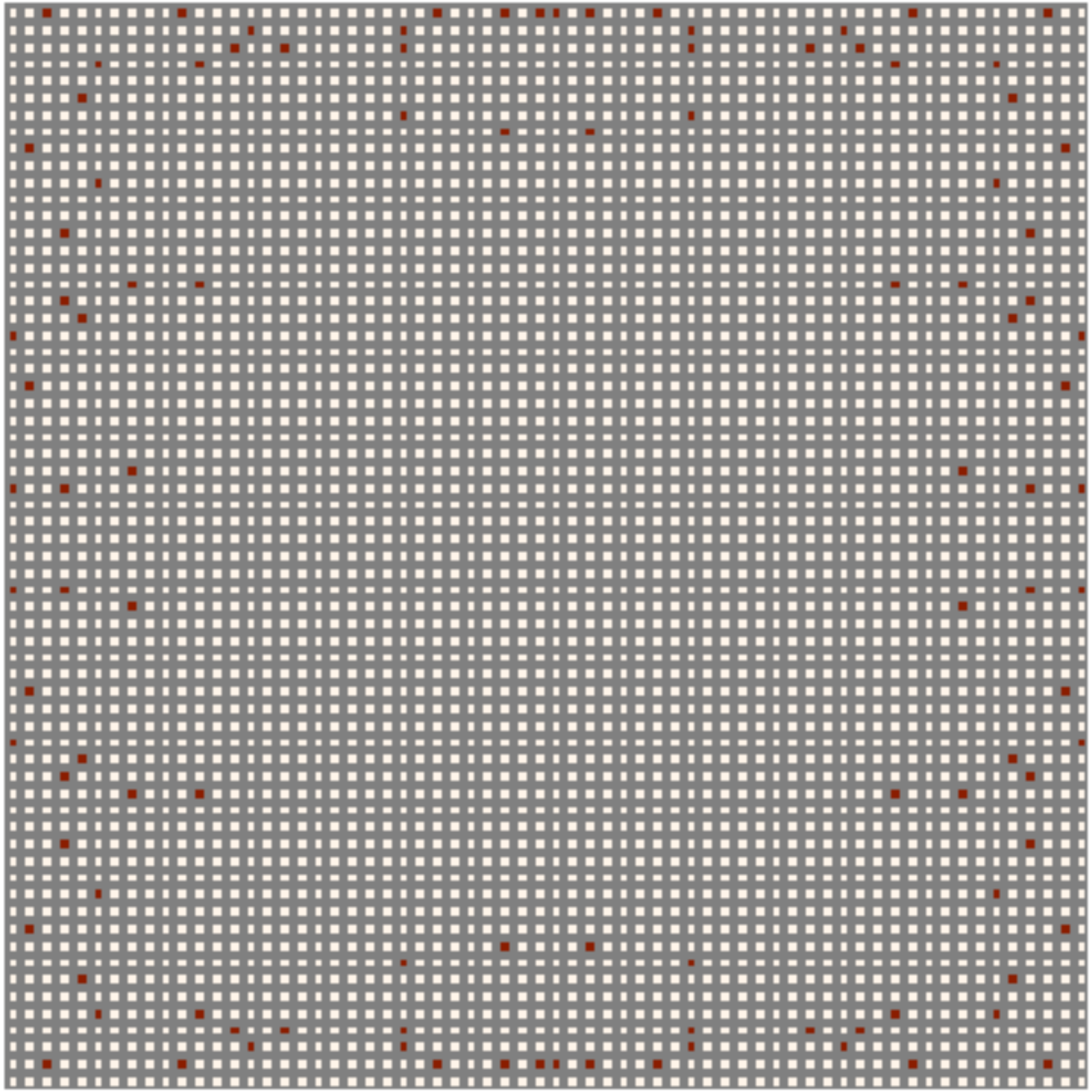


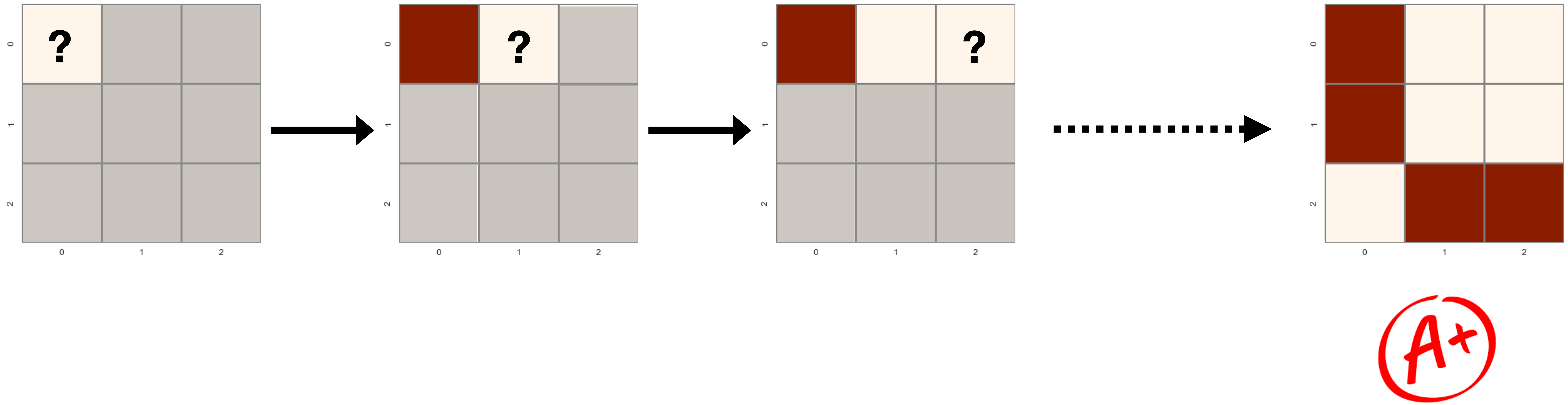
Image and Generation by Adam Z. Wagner

What makes this difficult?

What makes this difficult?

Credit Assignment Problem: Which decision made the most difference?

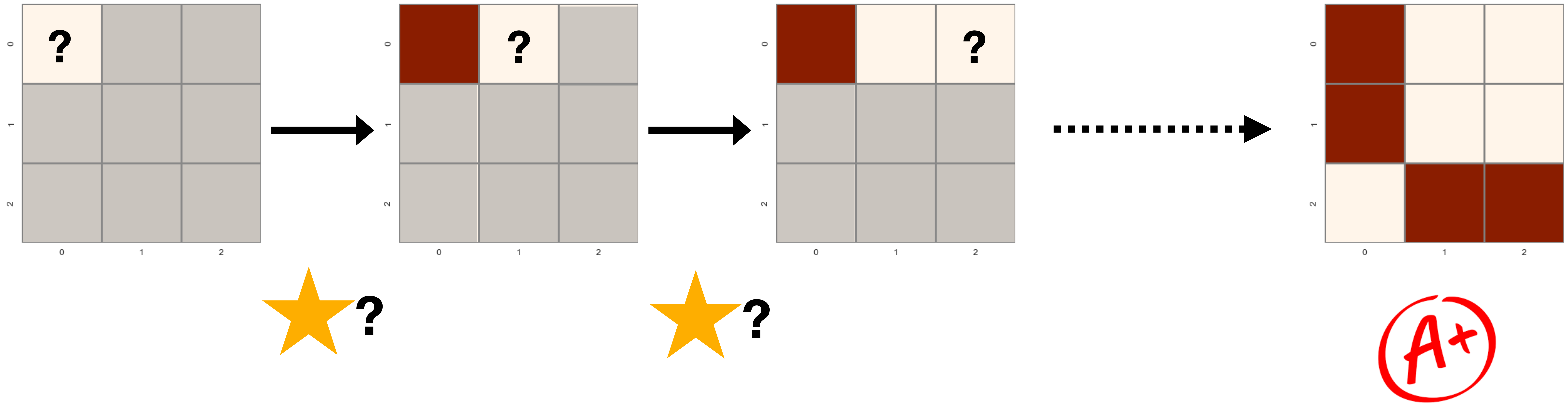
Sparse rewards: We reward the agent at the end of the game



What makes this difficult?

Credit Assignment Problem: Which decision made the most difference?

Sparse rewards: We reward the agent at the end of the game

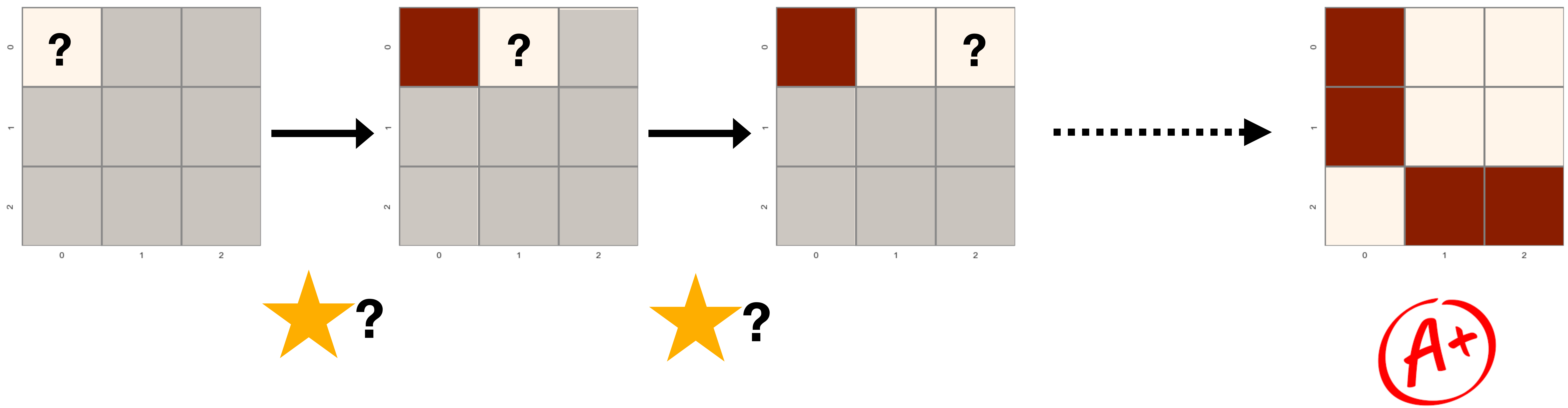


What makes this difficult?

Credit Assignment Problem: Which decision made the most difference?

Sparse rewards: We reward the agent at the end of the game

Reward function design.



Overview

Mathematical Motivation and Background

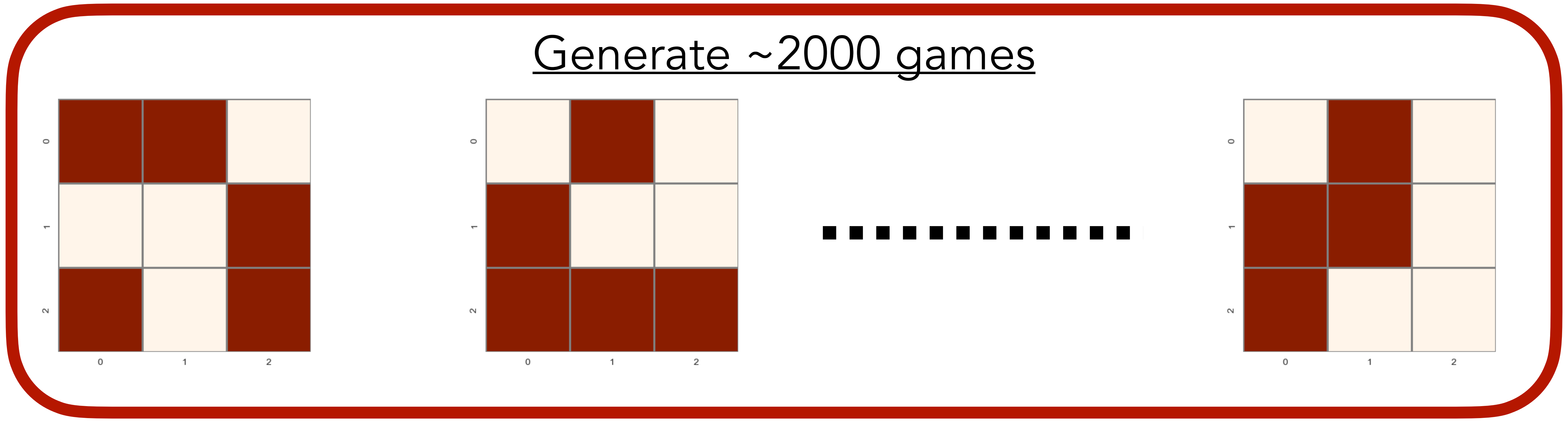
- Motivation: Non Metric Multidimensional Scaling
- Key definitions and propositions
- Known bounds for the problem

How Reinforcement Learning can help

- Reinforcement learning background and main algorithm
- Current results and observations
- Next Steps

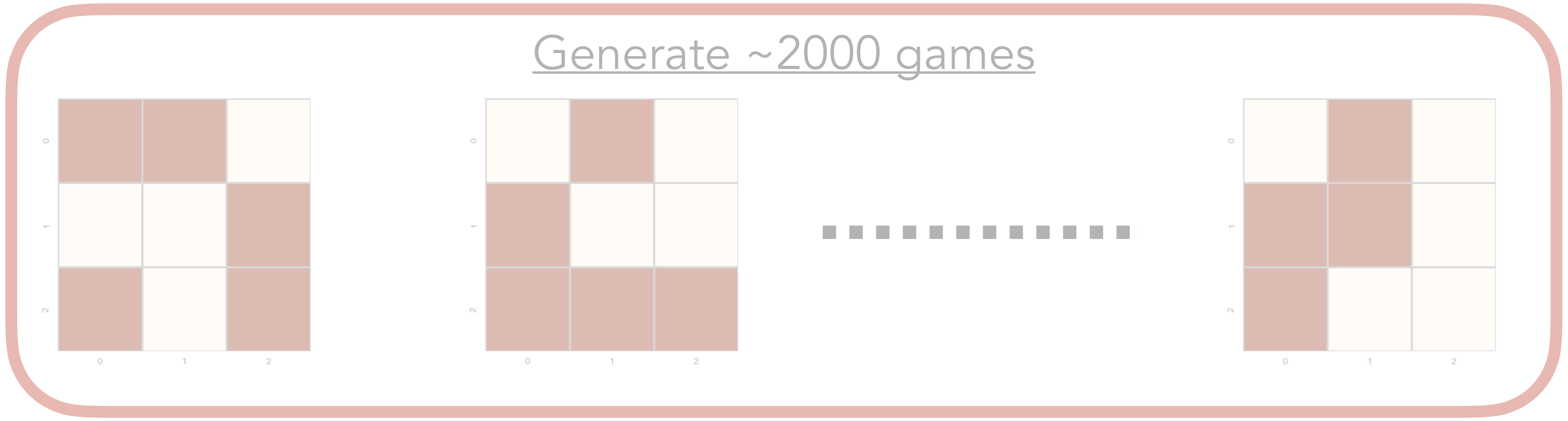
Next Steps

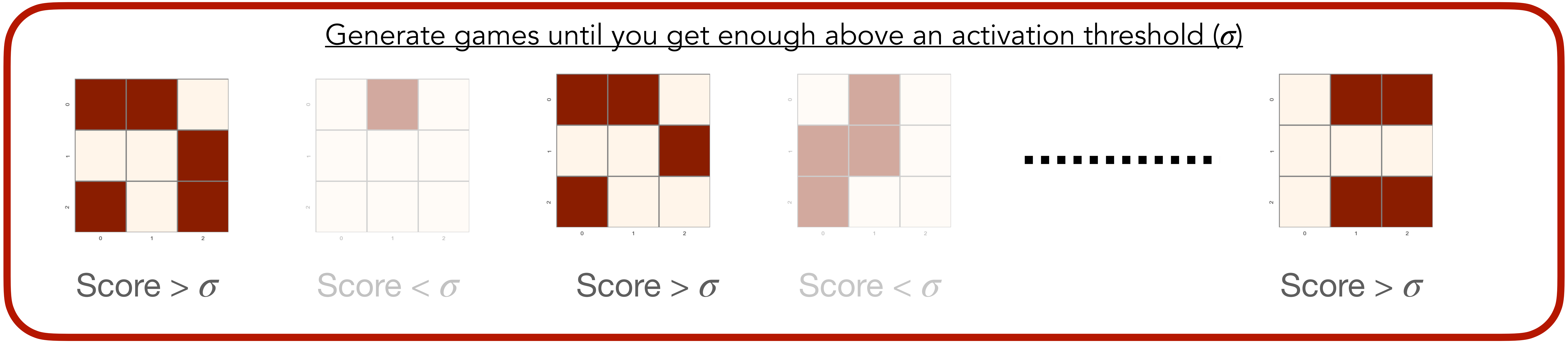
1. Activation Thresholding



Next Steps

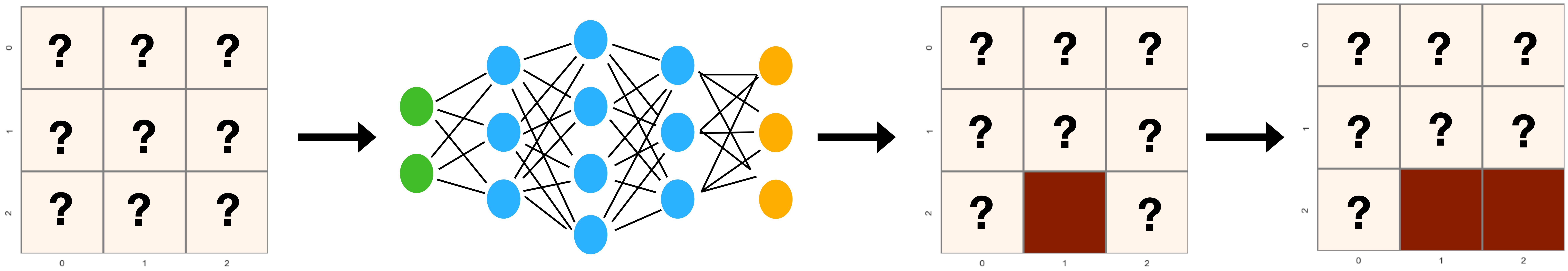
1. Activation Thresholding





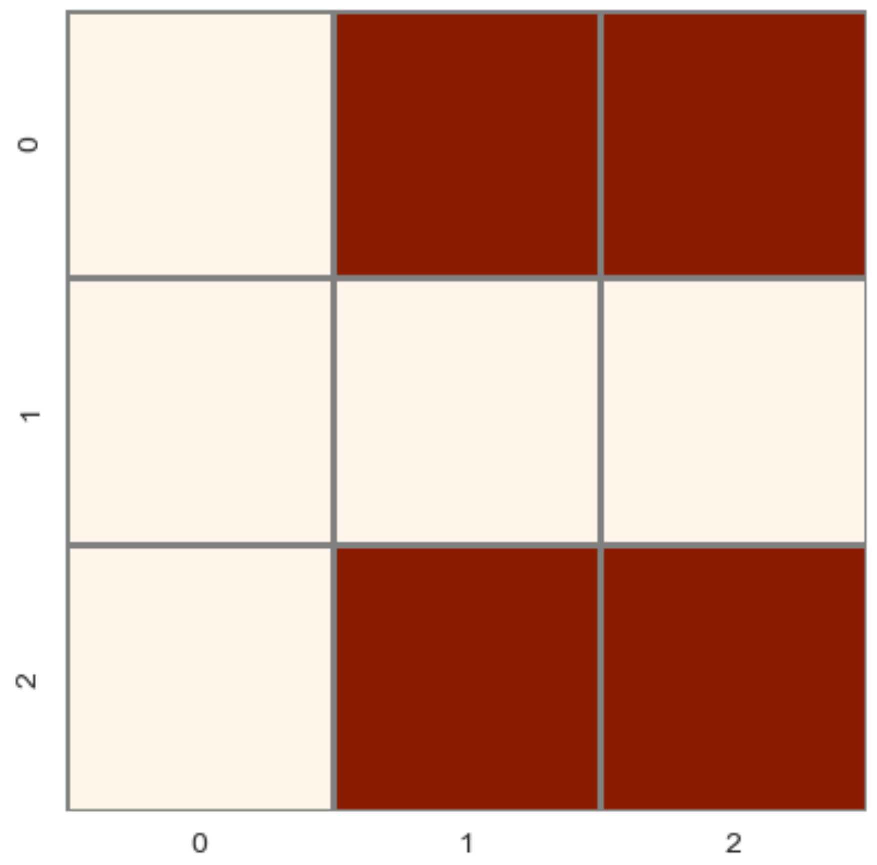
Next Steps

2. Set up the game differently:

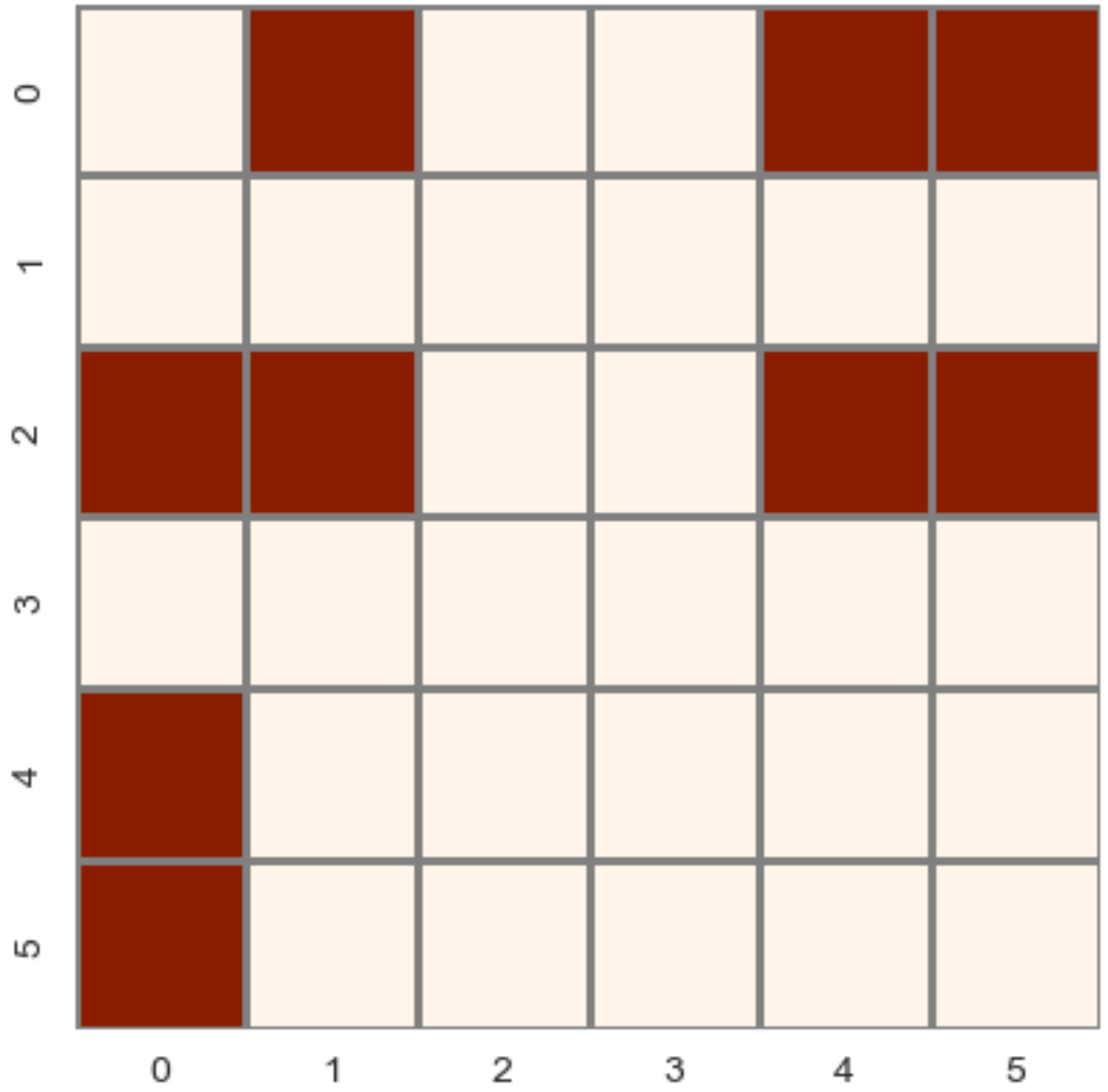


Next Steps

3. Inductive Thinking - Transfer Learning



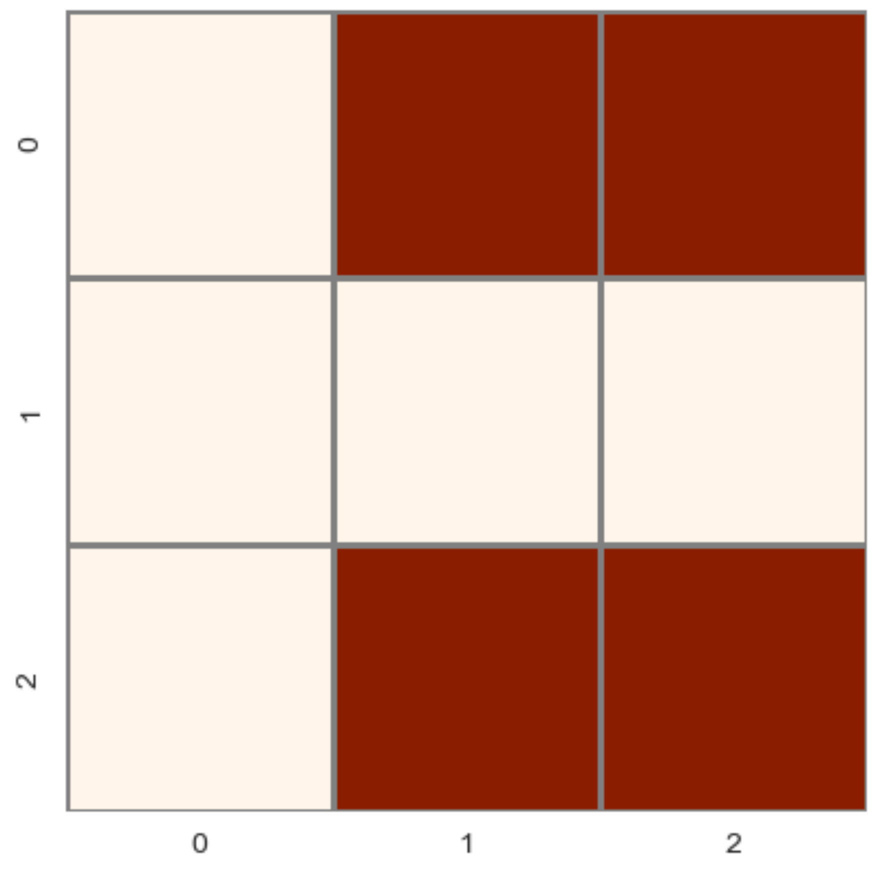
n = 3



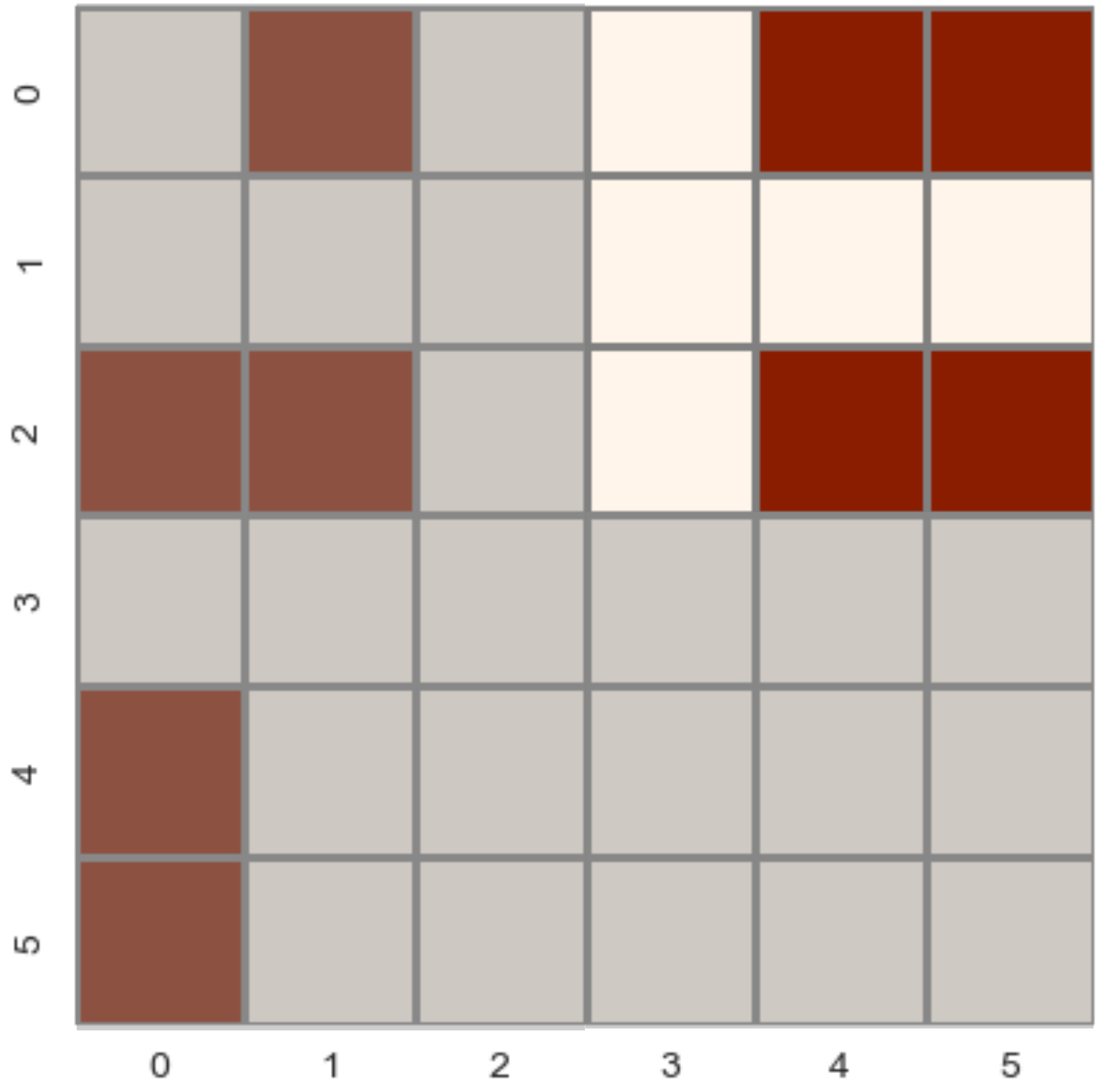
n = 6

<u>Next Steps</u>

3. Inductive Thinking - Transfer Learning



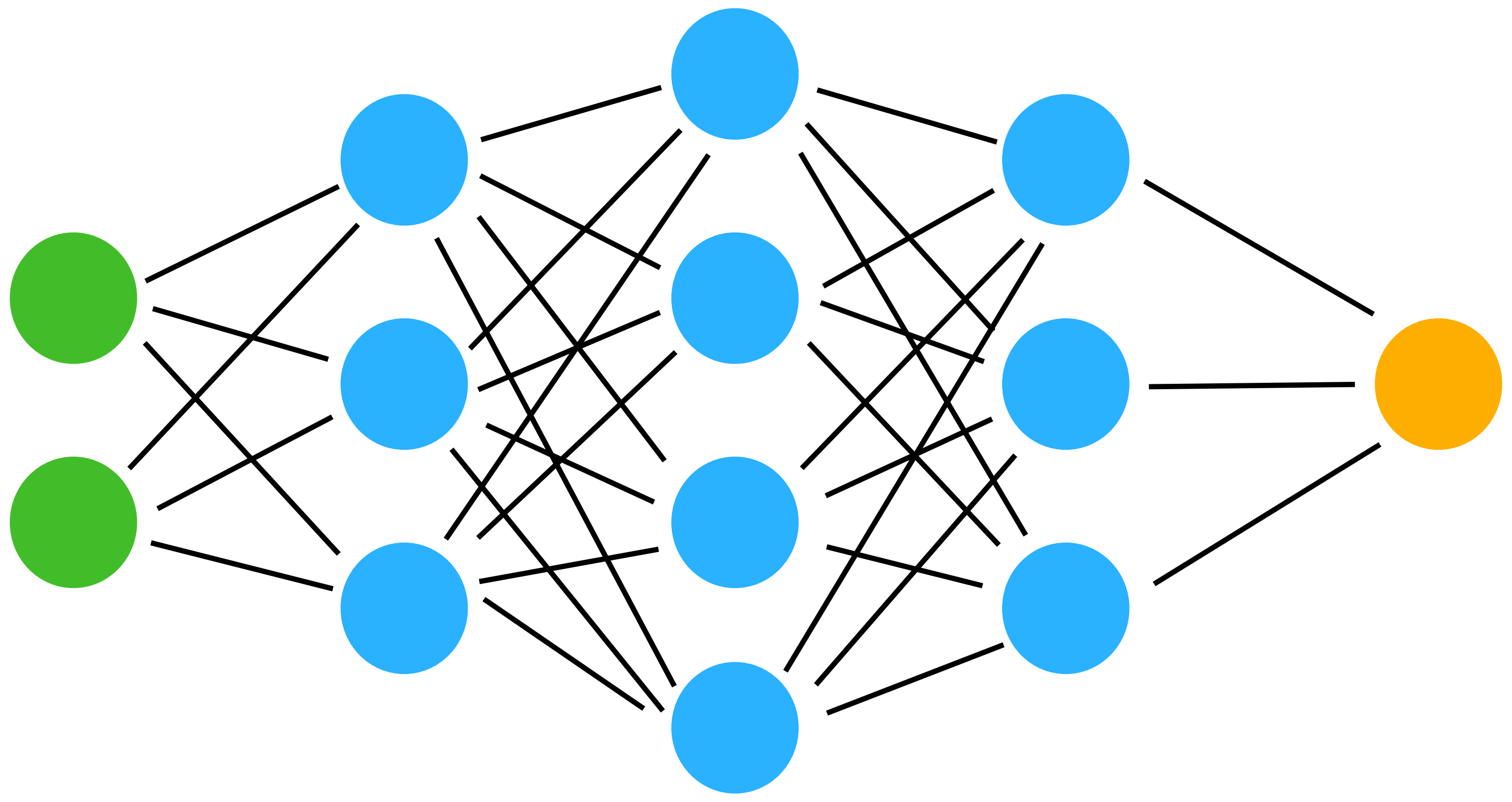
n = 3



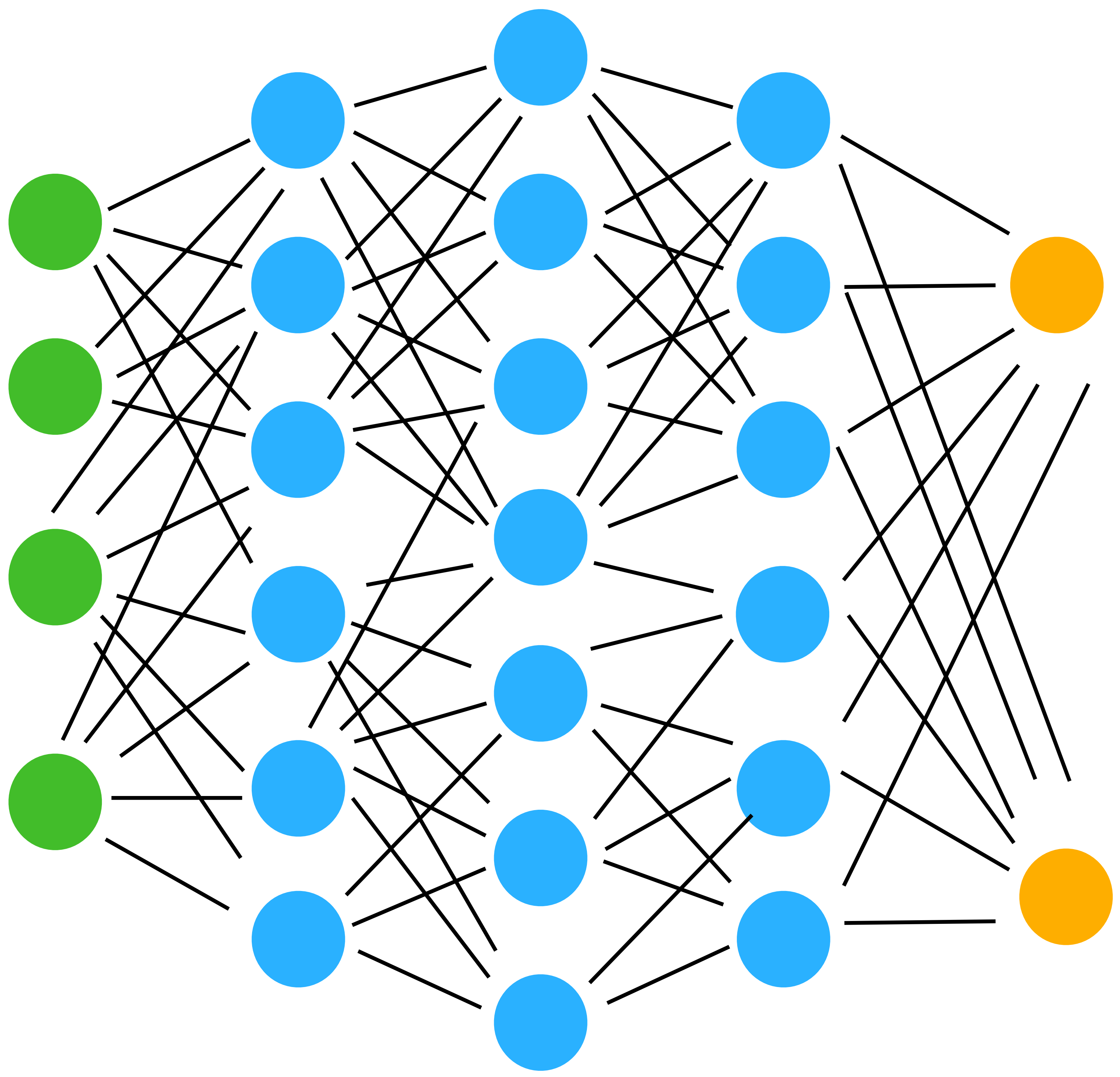
n = 6

Next Steps

3. Inductive Thinking - Transfer Learning



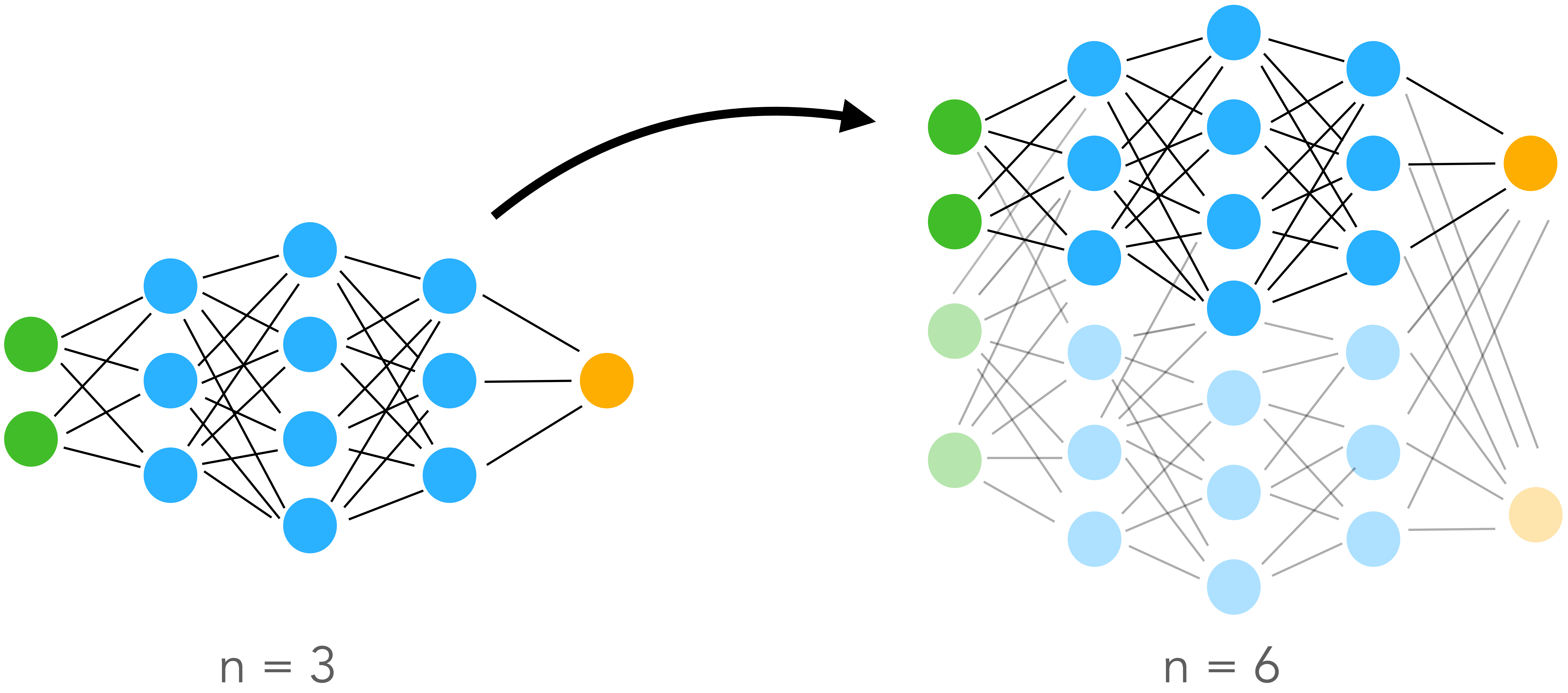
n = 3



n = 6

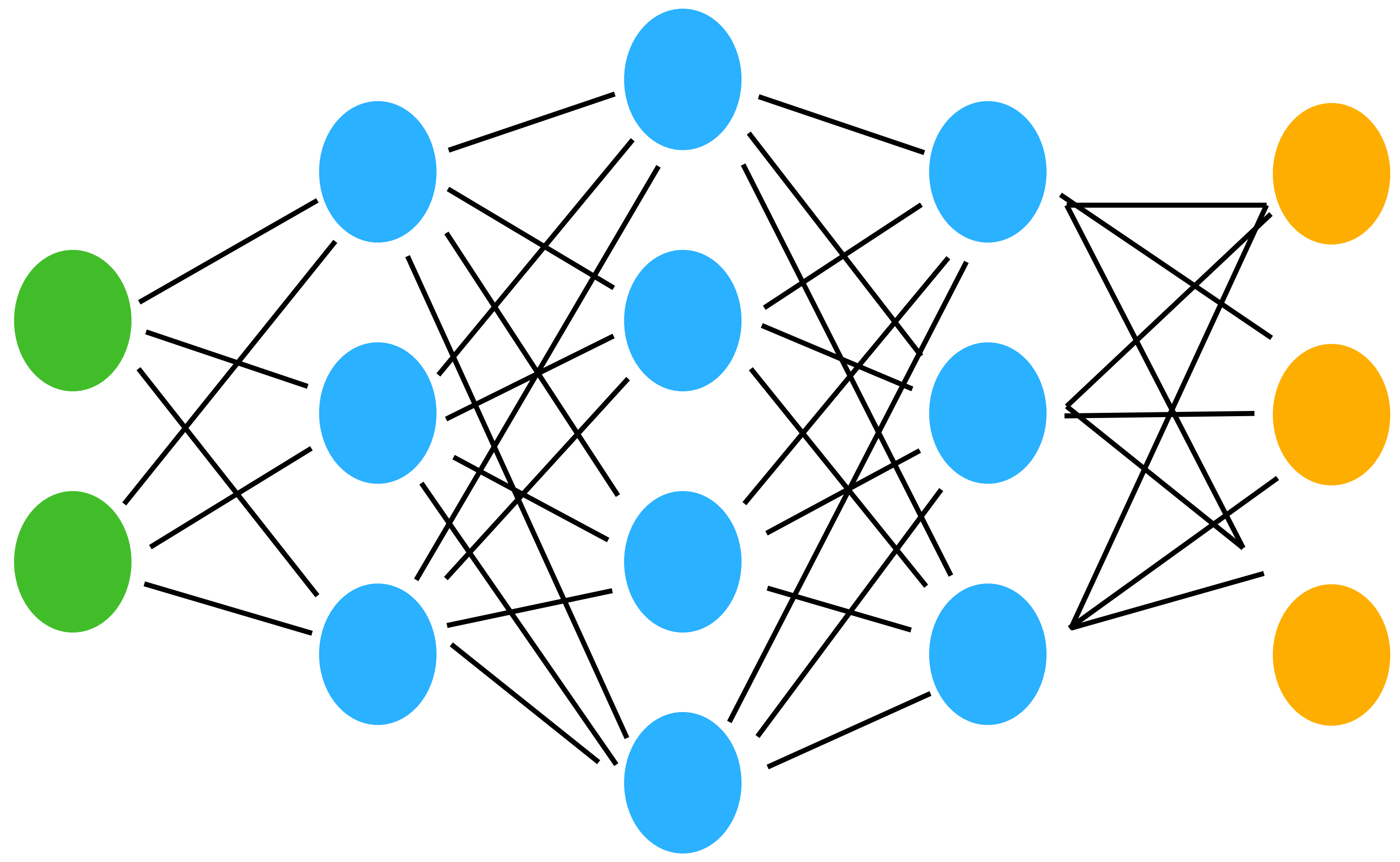
Next Steps

3. Inductive Thinking - Transfer Learning

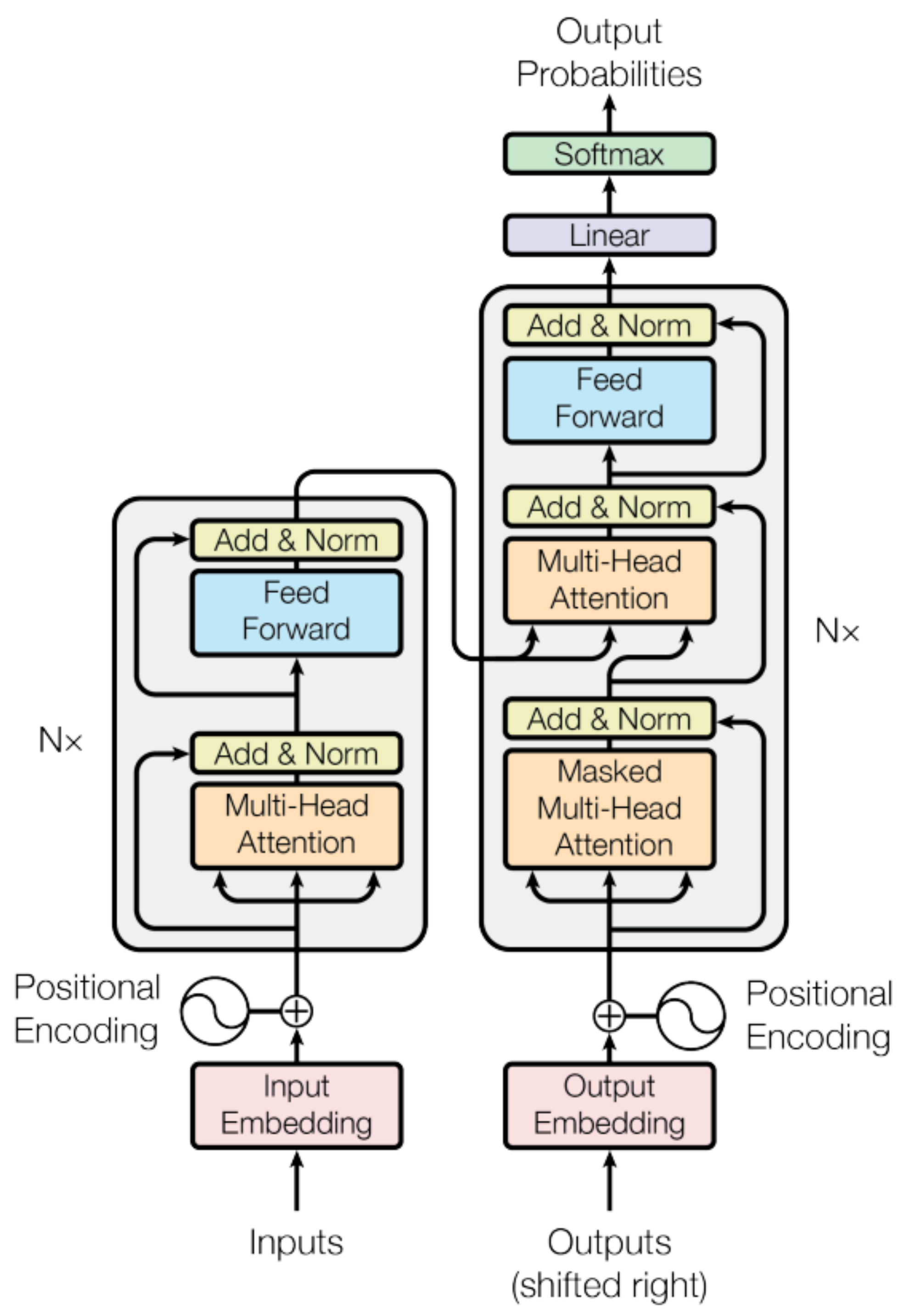


Next Steps

4. Experimenting more extensively with other architectures



Different Architectures for NNs



Transformers

Improvements

With Heuristics:

For large boards
(e.g. 64 x 64)

Found largest
known generations

64 x 64: ~~108~~ Points
110 Points

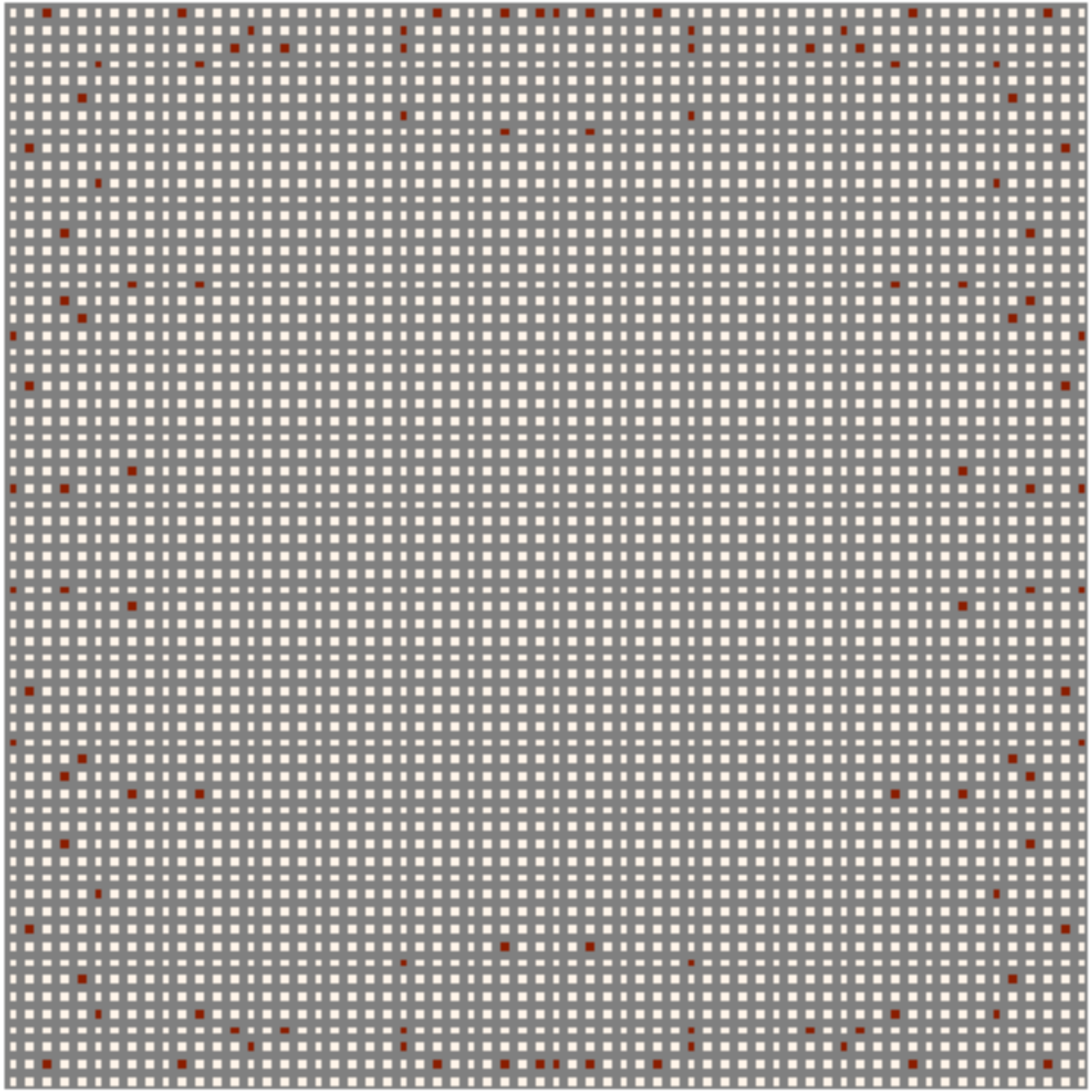
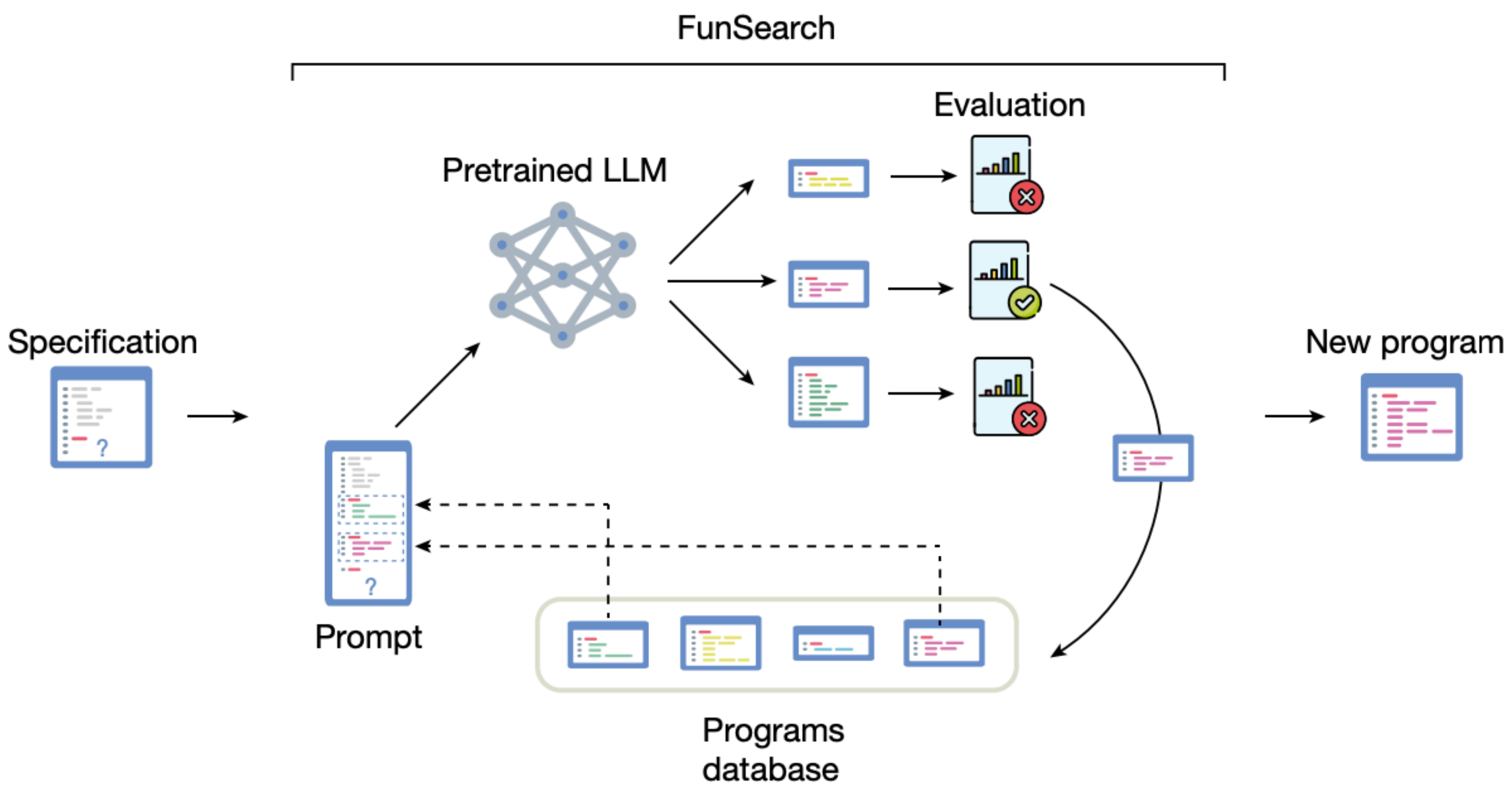


Image and Generation by Adam Z. Wagner

Next Steps

4. Experimenting more extensively with other architectures



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

Currently Ongoing Progress

1. Set Up Game Differently - Learn entire board at once
2. Activation Thresholding
3. Inductive Thinking - Transfer Learning
4. Experiment with different architectures - Other NNs or Transformers

Karan Srivastava
ksrivastava4@wisc.edu