Verification-Aided Deep Ensemble Selection

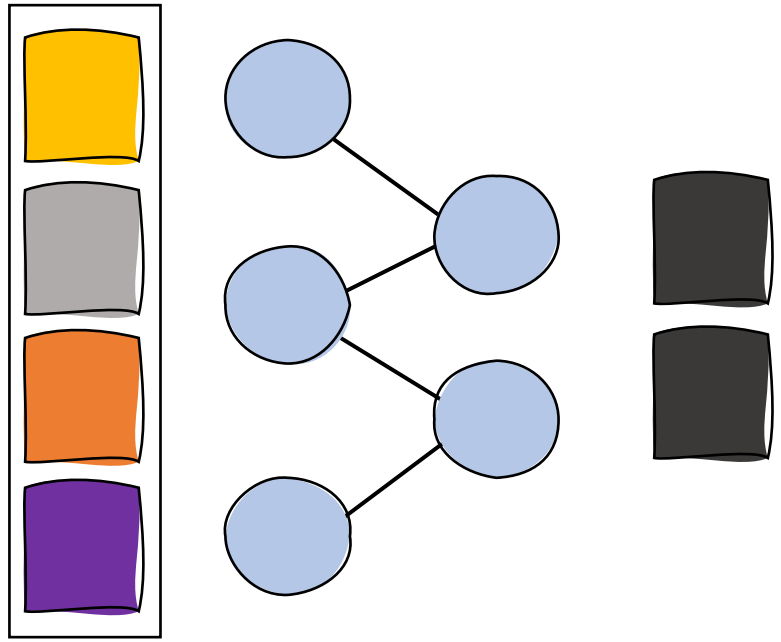
October 2022

Guy Amir Tom Zelazny
Guy Katz Michael Schapira

fmcad

Motivation

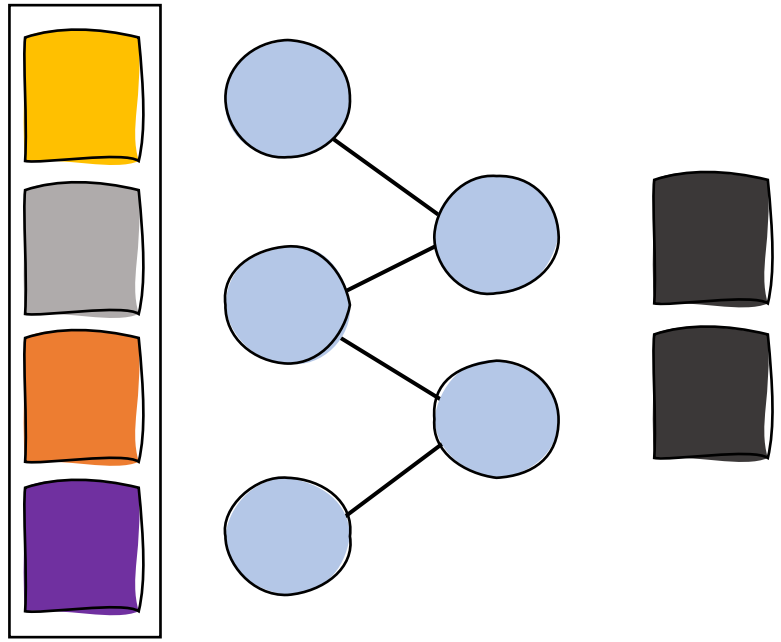
Deep Neural Networks (DNNs) achieve state-of-the-art results, but:



Motivation

Deep Neural Networks (DNNs) achieve state-of-the-art results, but:

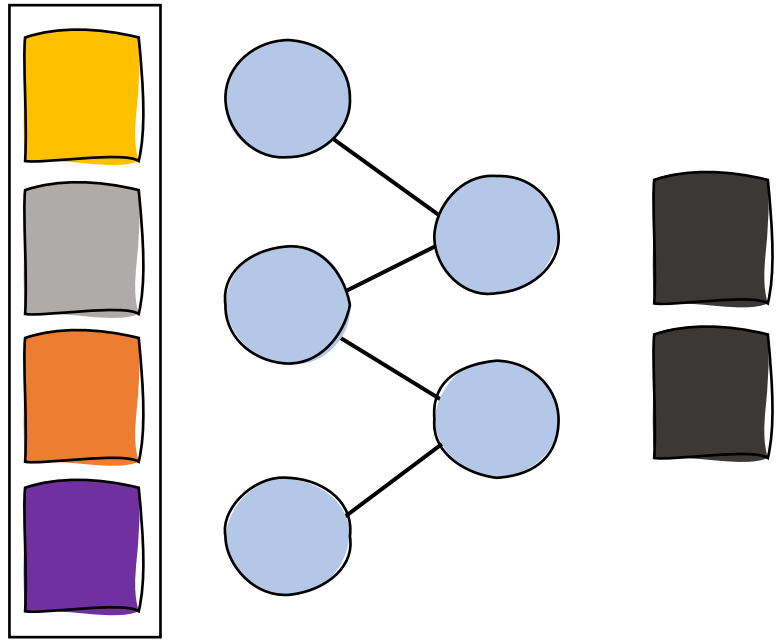
✗ Training stochasticity affects **accuracy** and **robustness**



Motivation

Deep Neural Networks (DNNs) achieve state-of-the-art results, but:

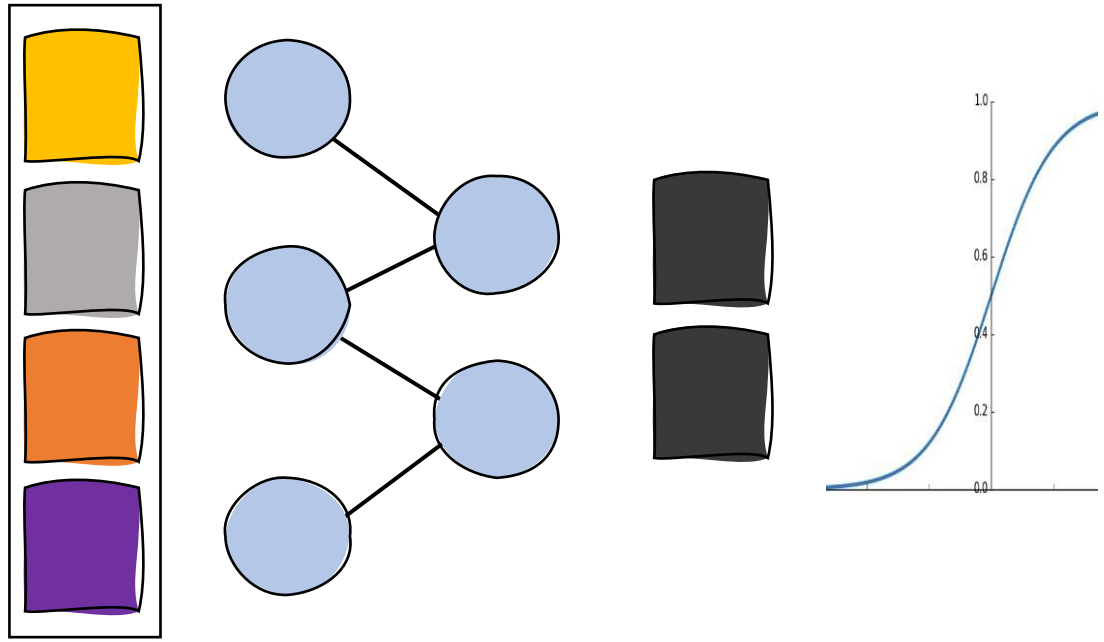
- ✗ Training stochasticity affects **accuracy** and **robustness**
- ✗ Do not express **uncertainty**



Motivation

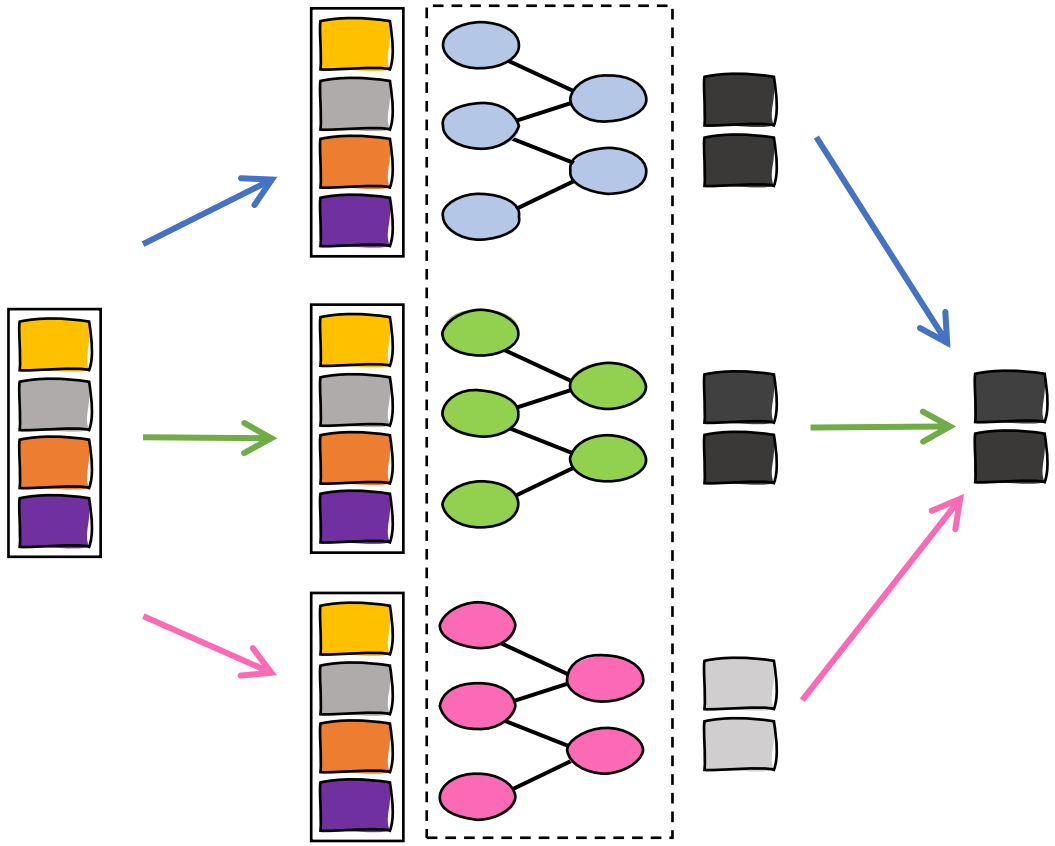
Deep Neural Networks (DNNs) achieve state-of-the-art results, but:

- ✗ Training stochasticity affects **accuracy** and **robustness**
- ✗ Do not express **uncertainty**



Solution?

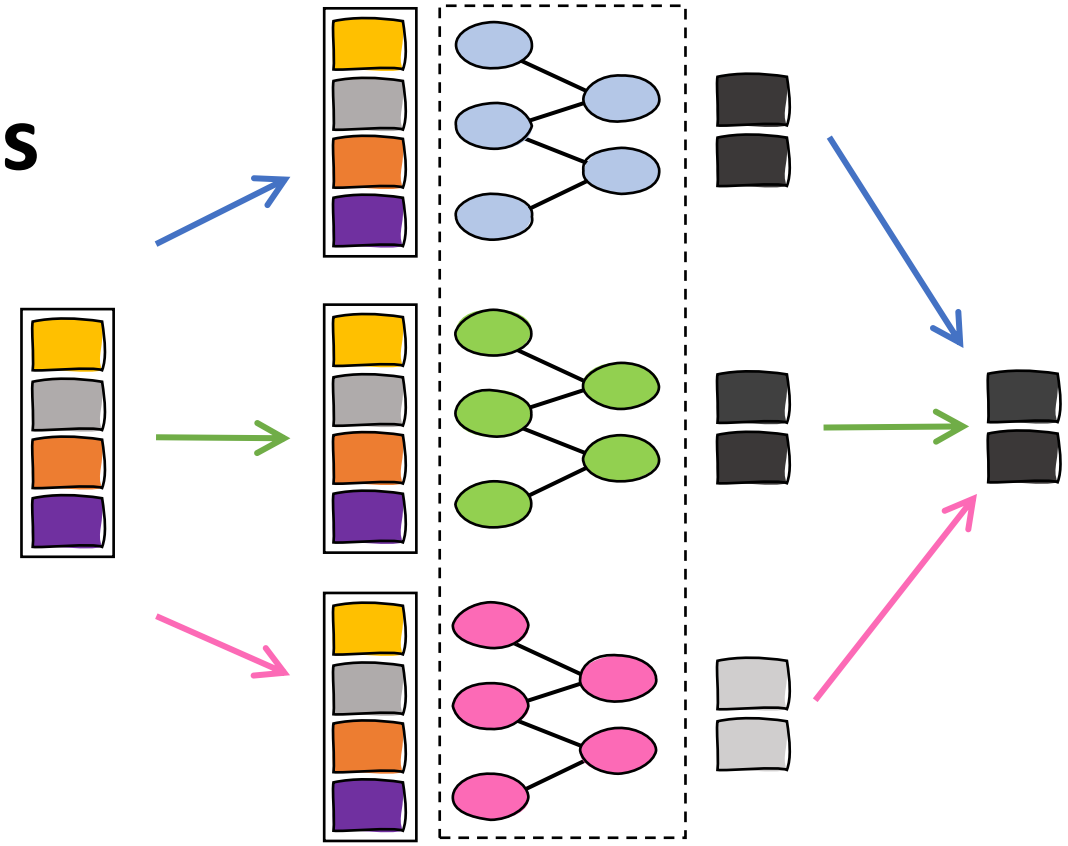
Train **ensembles** of DNN models



Solution?

Train **ensembles** of DNN models

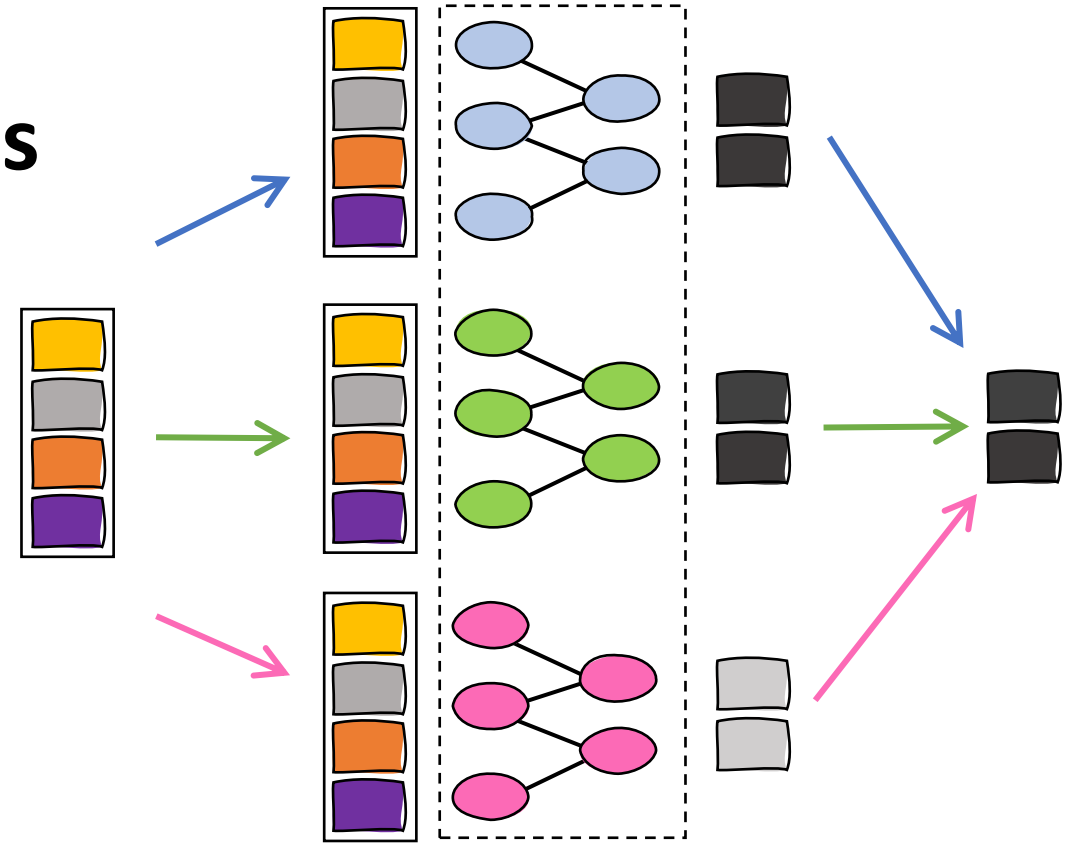
- ☑ Higher **accuracy** and **robustness**



Solution?

Train **ensembles** of DNN models

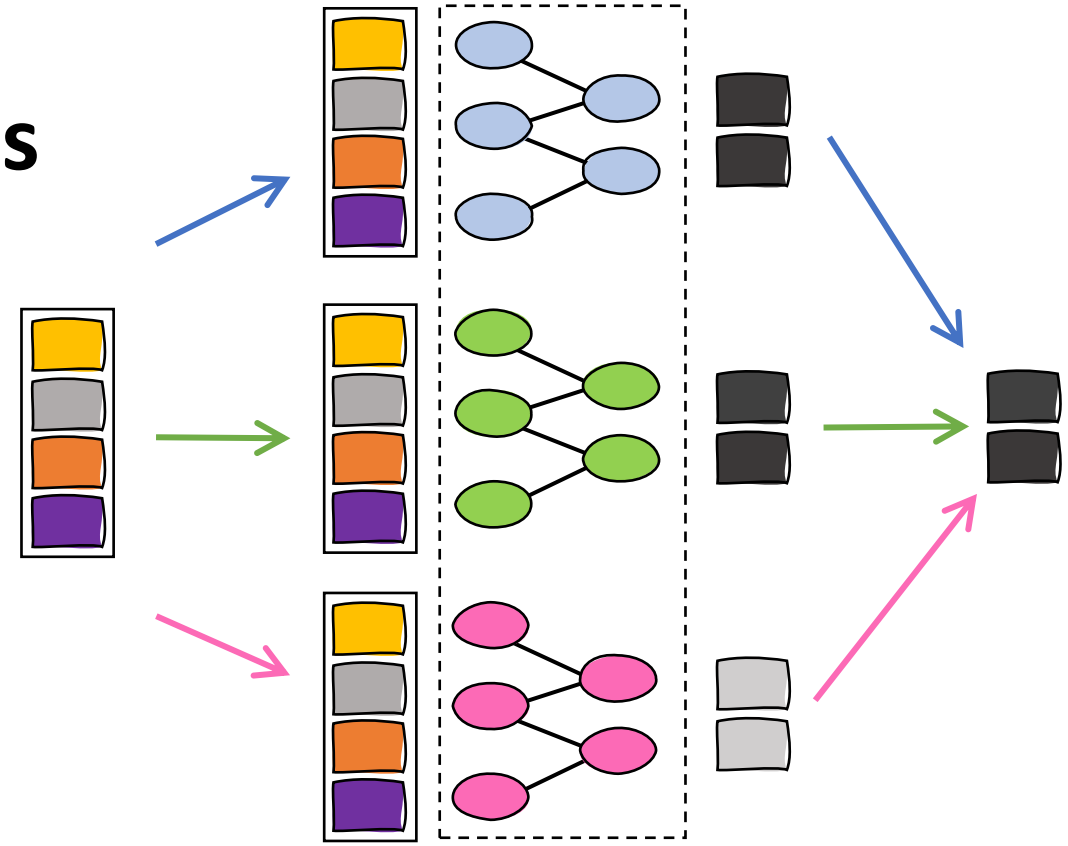
- ☑ Higher **accuracy** and **robustness**
- ☑ **Uncertainty** estimation



Solution?

Train **ensembles** of DNN models

- Higher **accuracy** and **robustness**
- **Uncertainty** estimation
- **Parallelize** training



Ensemble Selection

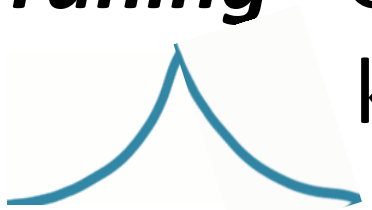
Ensemble Pruning - Given $n > k$ trained DNNs, how can we choose the **``best''** k-sized ensemble ?

Ensemble Selection

Ensemble Pruning - Given $n > k$ trained DNNs, how can we choose the  k-sized ensemble ?

most diverse

Ensemble Selection

Ensemble Pruning - Given $n > k$ trained DNNs, how can we choose the  k-sized ensemble ?

most diverse

Option 1 - choose all $k = n$ trained DNNs

Ensemble Selection

Ensemble Pruning - Given $n > k$ trained DNNs, how can we choose the  k-sized ensemble ?

most diverse

Option 1 – choose all $k = n$ trained DNNs

✗ **"Ensembling Neural Networks: Many Could be Better Than All"**
Zhou et. al. (2010)

Ensemble Selection

Ensemble Pruning - Given $n > k$ trained DNNs, how can we choose the  k-sized ensemble ?

most diverse

Option 2 –

use **formal verification** to select the best subset k

Ensemble Selection

Option 2 –

use **formal verification** to select the best subset k

How?

Verify all possible ensembles and choose the best one

Ensemble Selection

Option 2 –

use **formal verification** to select the best subset k

How?

Verify all possible ensembles and choose the best one

Setbacks

Ensemble Selection

Option 2 –

use **formal verification** to select the best subset k

How?

Verify all possible ensembles and choose the best one

Setbacks

✗ There are $\binom{n}{k}$ **k-sized ensemble combinations**

Ensemble Selection

Option 2 –

use **formal verification** to select the best subset k

How?

Verify all possible ensembles and choose the best one

Setbacks

✗ There are $\binom{n}{k}$ **k-sized ensemble combinations**

✗ Each query is **NP-hard**

Ensemble Selection

Option 2 –

use **formal verification** to select the best subset k

How?

~~***Verify all possible ensembles and choose the best one***~~

Ensemble Selection

Option 2 –

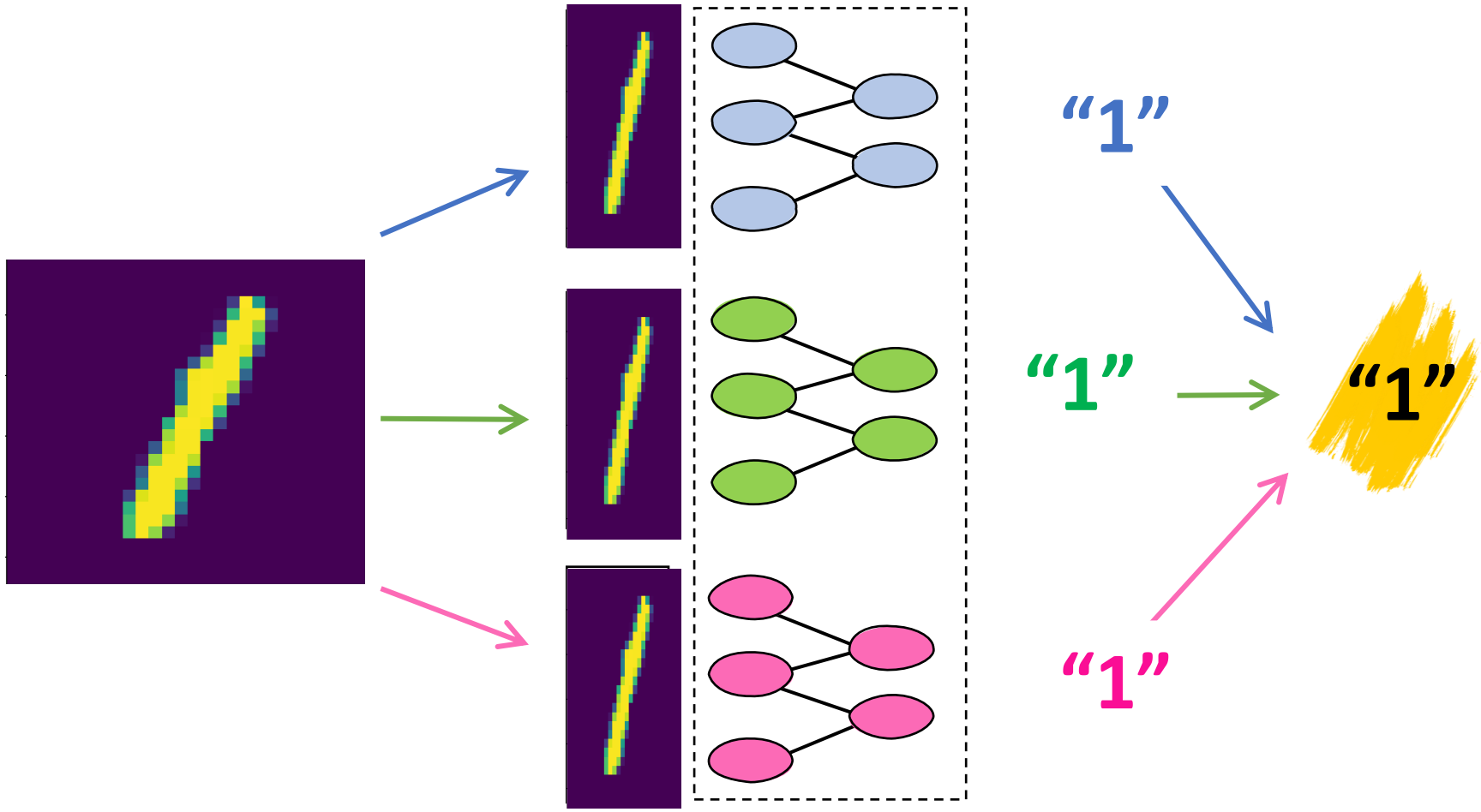
use **formal verification** to select the best subset k

How?

~~***Verify all possible ensembles and choose the best one***~~

Check if DNNs tend to err simultaneously

Agreement Points



Agreement Points

Data points from the test set which are ***classified the same*** by all DNN members

Agreement Points

Data points from the test set which are ***classified the same*** by all DNN members

Intuitively – areas with a ***high consensus*** among the DNNs

Agreement Points

Data points from the test set which are ***classified the same*** by all DNN members

Intuitively – areas with a ***high consensus*** among the DNNs

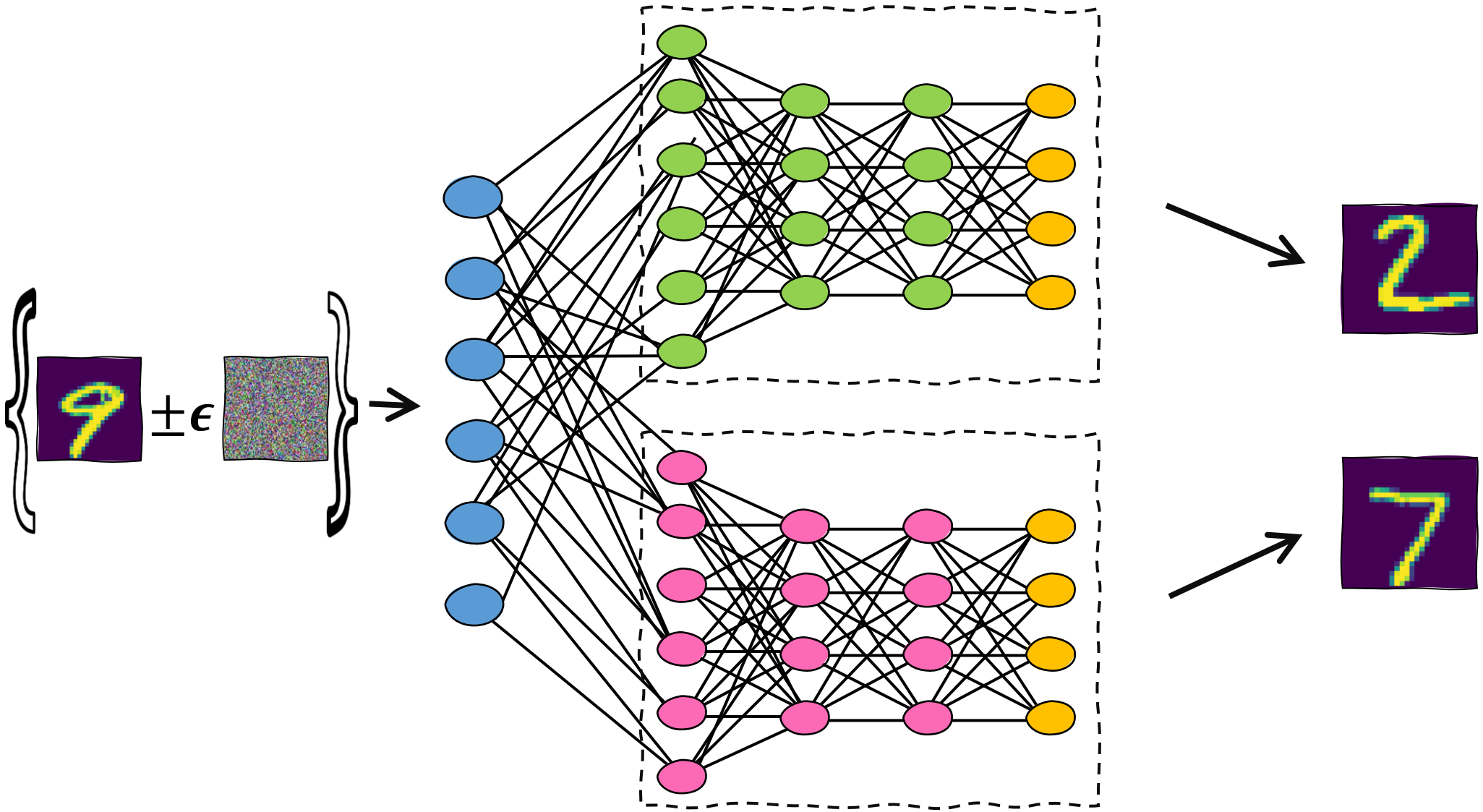
Allows a ***fair comparison*** between DNN pairs

Mutual Error

Given a fixed ***agreement point*** and a ***pair*** of DNNs

A ***mutual error*** is a perturbation that ***simultaneously*** causes a pair of DNNs to ***misclassify*** the point

Mutual Error (with verifier)



Uniqueness Score

The **mutual error score** is the average ***mutual errors*** a pair of DNNs have on a set of agreement points

Uniqueness Score

The **mutual error score** is the average ***mutual errors*** a pair of DNNs have on a set of agreement points

The ***uniqueness score*** is a score indicating how (un)likely it is for a single DNN to **err** with the remaining ensemble DNNs

Uniqueness Score

The **mutual error score** is the average ***mutual errors*** a pair of DNNs have on a set of agreement points

The ***uniqueness score*** is a score indicating how (un)likely it is for a single DNN to **err** with the remaining ensemble DNNs

The **higher** the ***uniqueness score*** - the **better** the DNN

Greedy-Based Strategy

Initially:

1 Independently ***train n DNNs*** and fix a set of ***agreement points***

Greedy-Based Strategy

Initially:

1. Independently ***train n DNNs*** and fix a set of ***agreement points***
2. Create an **arbitrary ensemble E**

Greedy-Based Strategy

Initially:

1 Independently ***train n DNNs*** and fix a set of ***agreement points***

2 Create an **arbitrary ensemble E**

While not TIMEOUT:

3 Compute the ***uniqueness score*** for each member

Greedy-Based Strategy

Initially:

1 Independently ***train n DNNs*** and fix a set of ***agreement points***

2 Create an **arbitrary ensemble E**

While not TIMEOUT:

3 Compute the ***uniqueness score*** for each member

4 Choose ***member with lowest score***

Greedy-Based Strategy

Initially:

1 Independently ***train n DNNs*** and fix a set of ***agreement points***

2 Create an **arbitrary ensemble E**

While not TIMEOUT:

3 Compute the ***uniqueness score*** for each member

4 Choose ***member with lowest score***

5 If there is a member with a ***higher*** score -> ***swap***

Greedy-Based Strategy

Initially:

1 Independently ***train n DNNs*** and fix a set of ***agreement points***

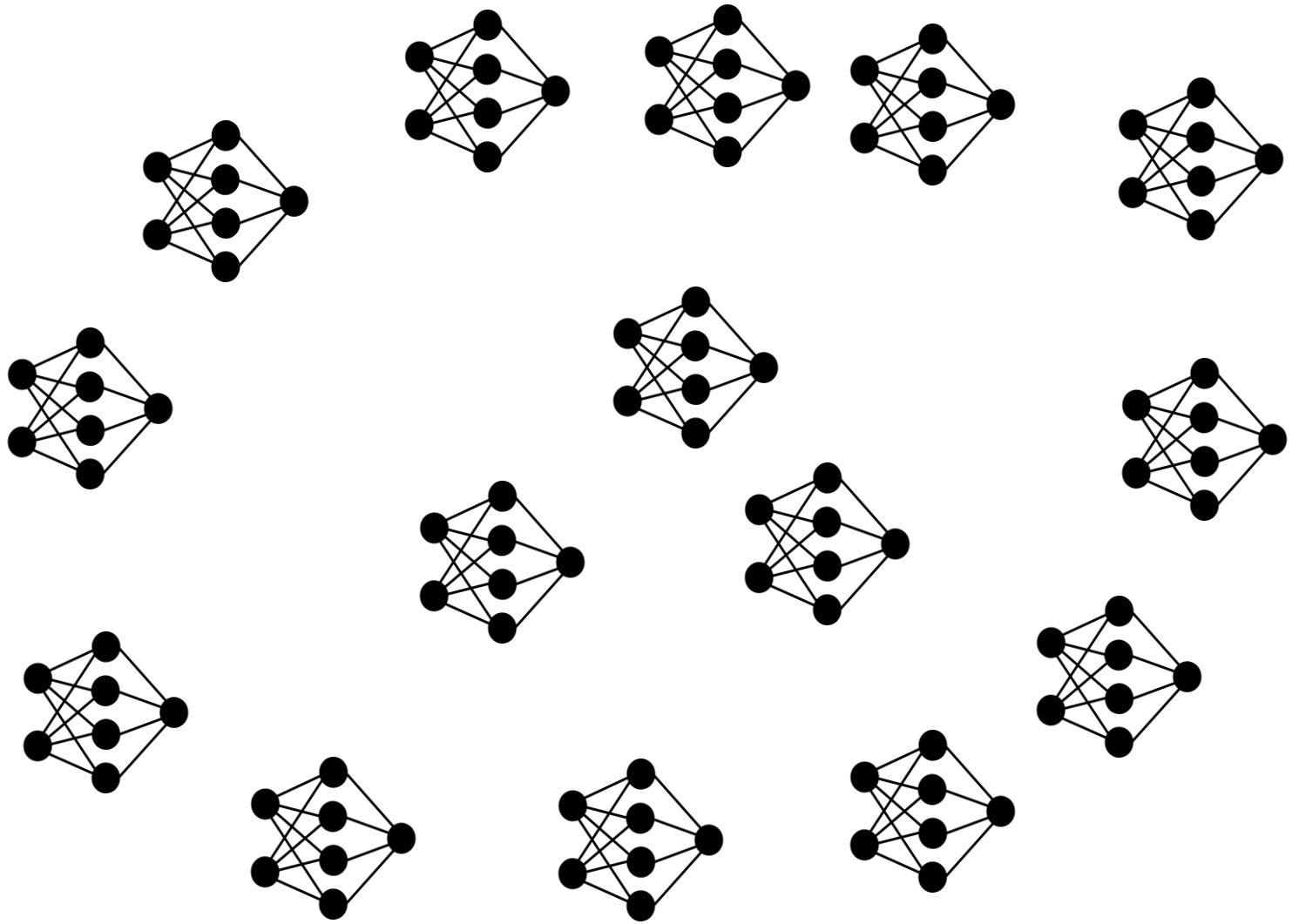
2 Create an **arbitrary ensemble E**

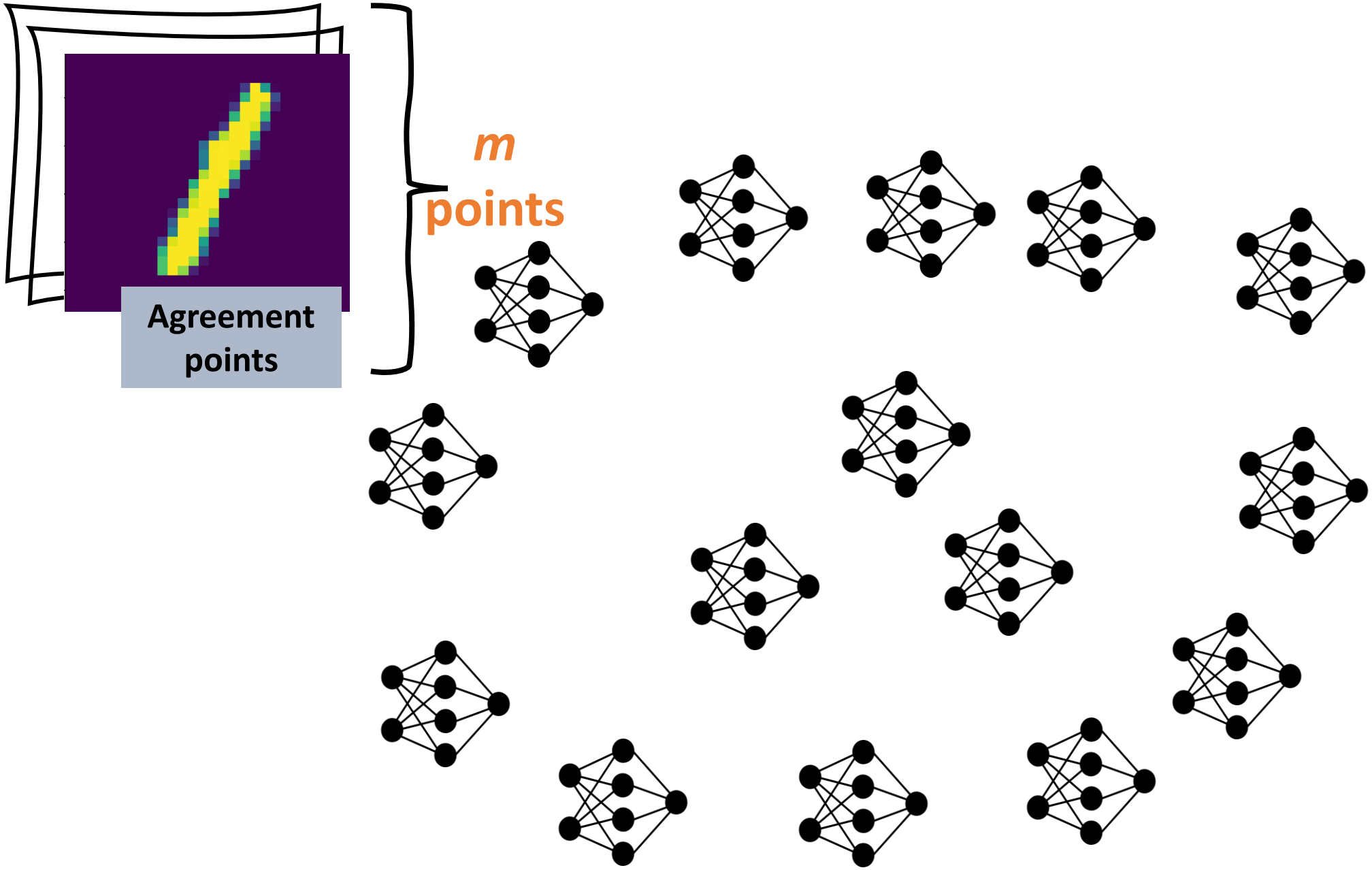
While not TIMEOUT:

3 Compute the ***uniqueness score*** for each member

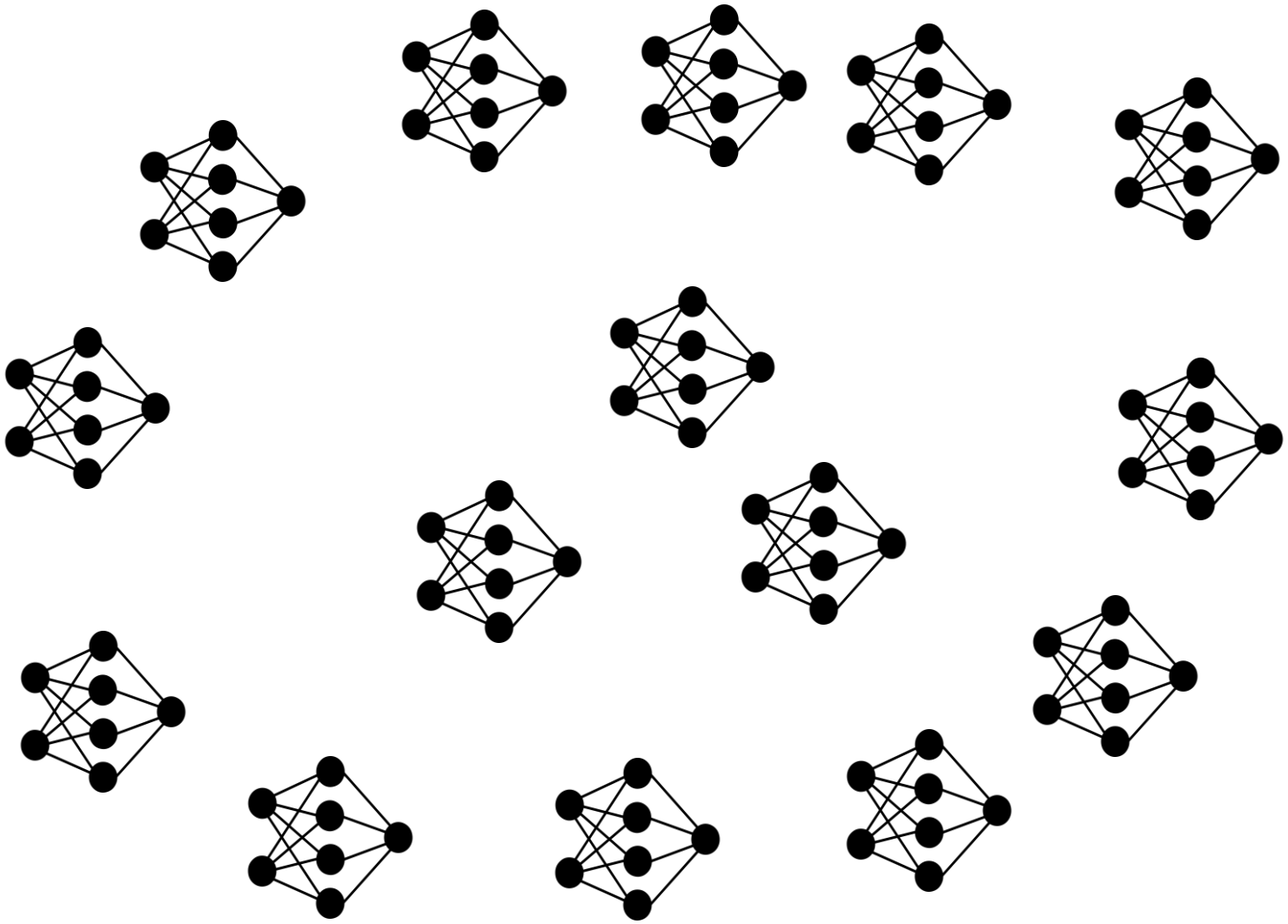
4 Choose ***member with lowest score***

5 If there is a member with a ***higher*** score -> ***swap***

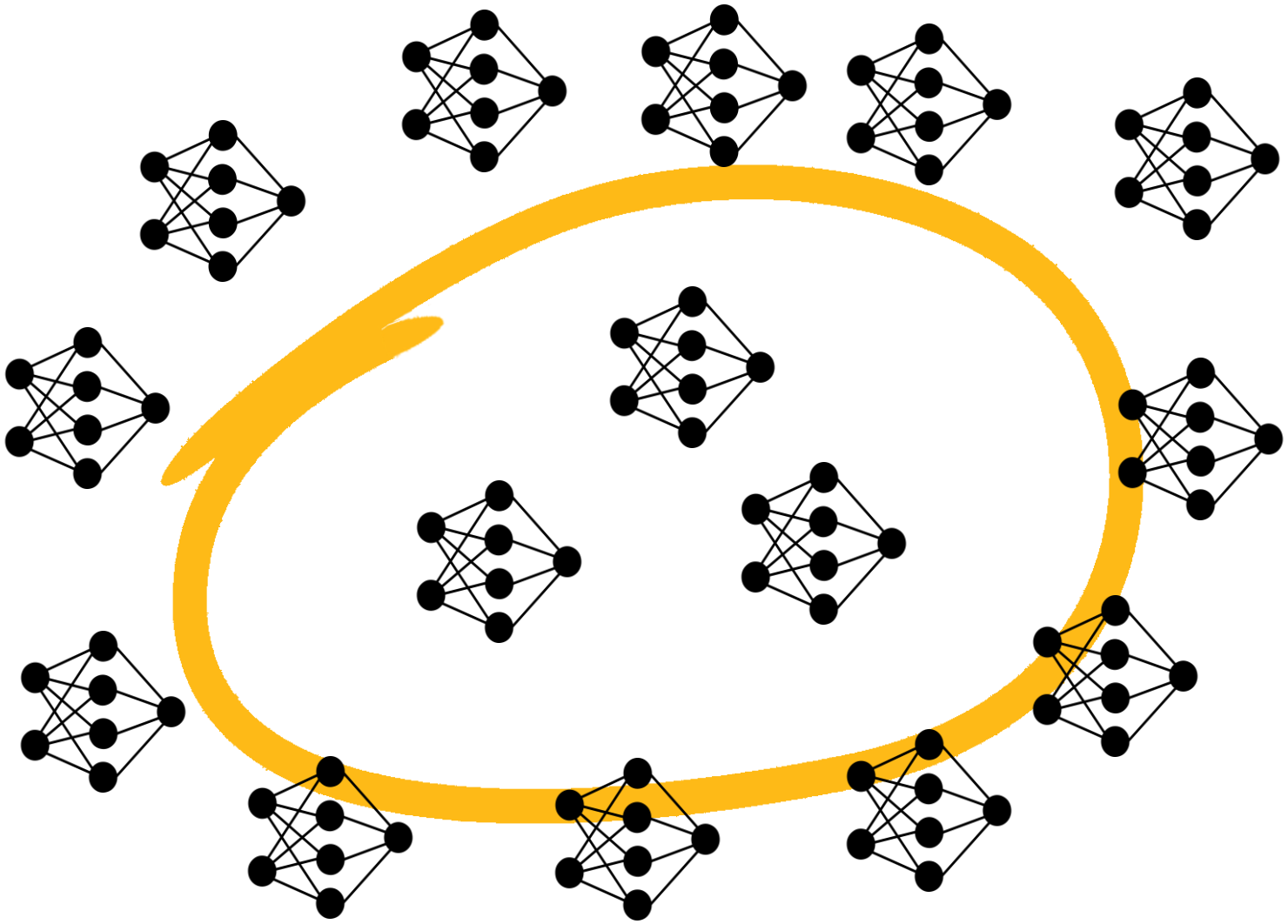



m
points
Agreement
points

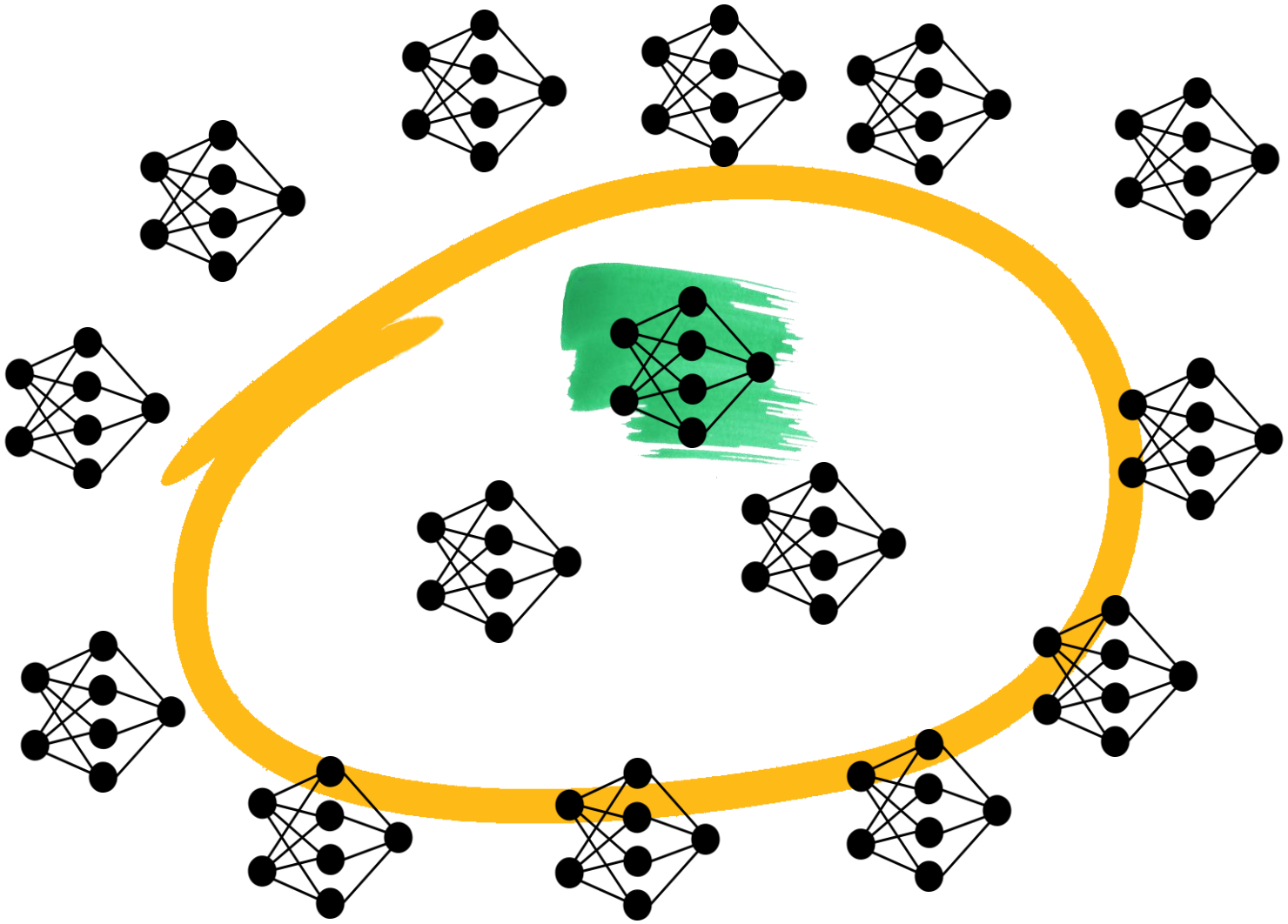
$$k = 3 < 15 = n$$



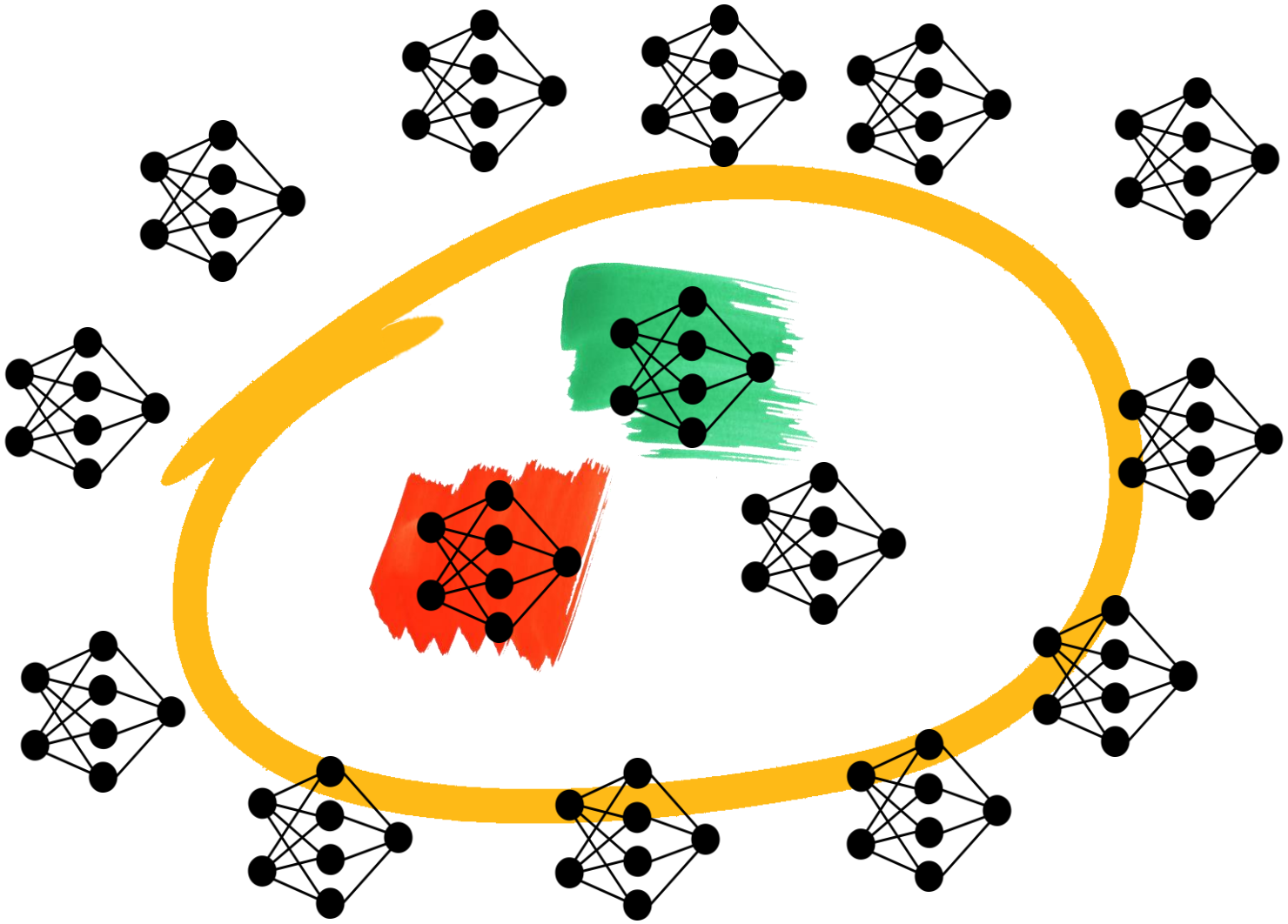
$$k = 3 < 15 = n$$



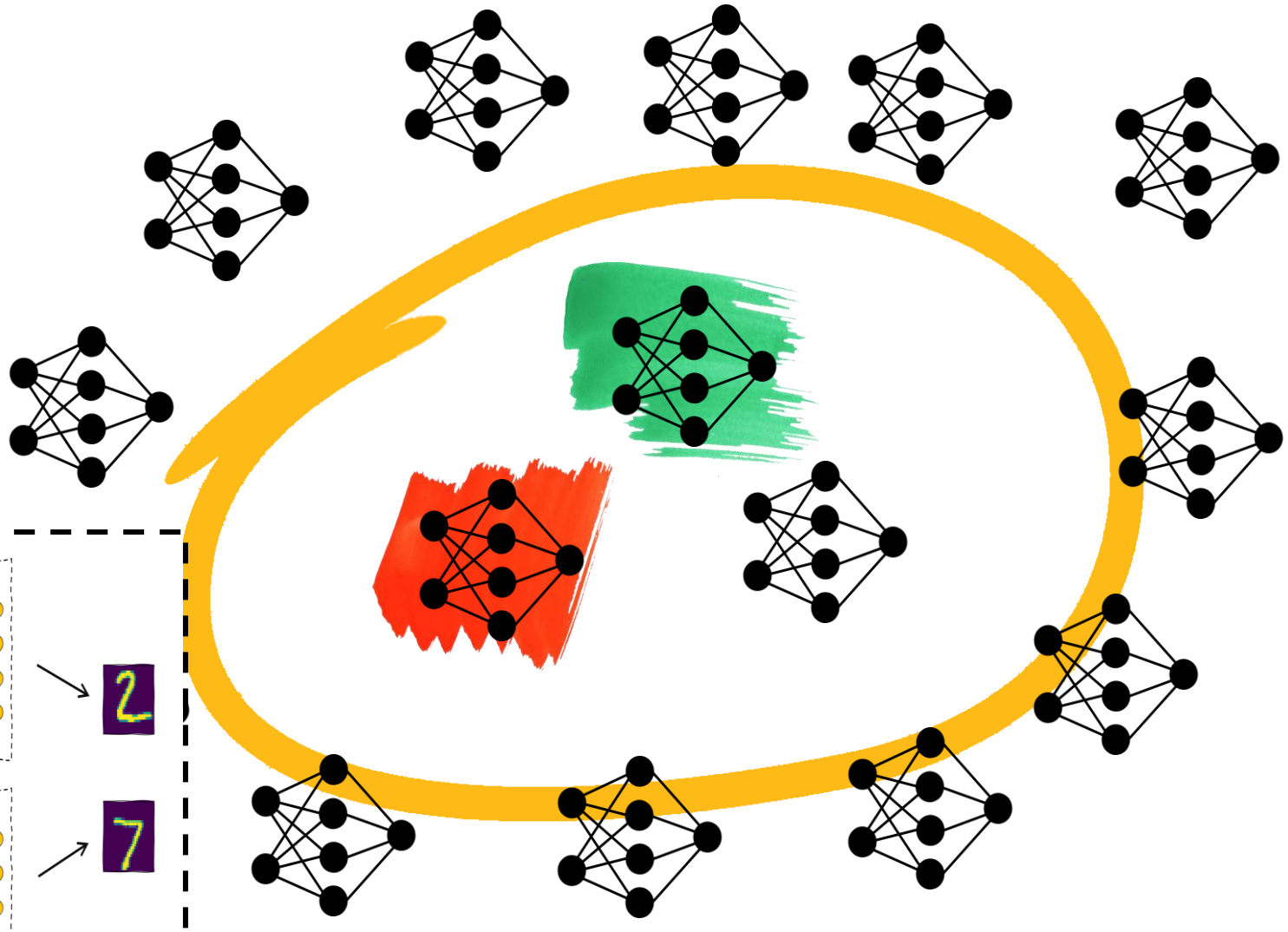
$$k = 3 < 15 = n$$



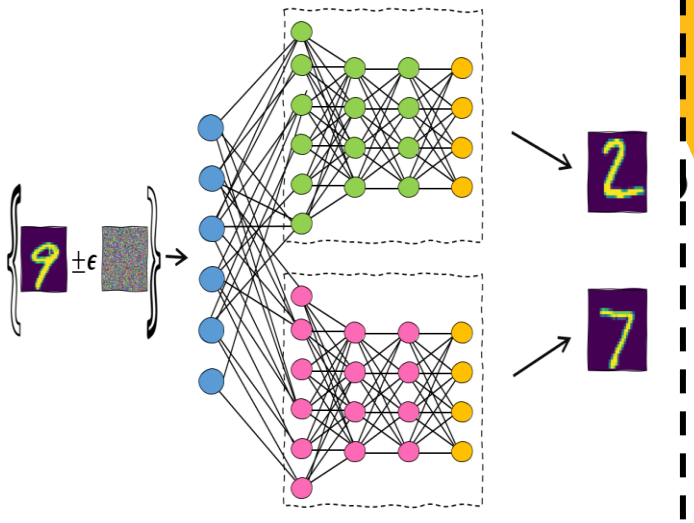
$$k = 3 < 15 = n$$



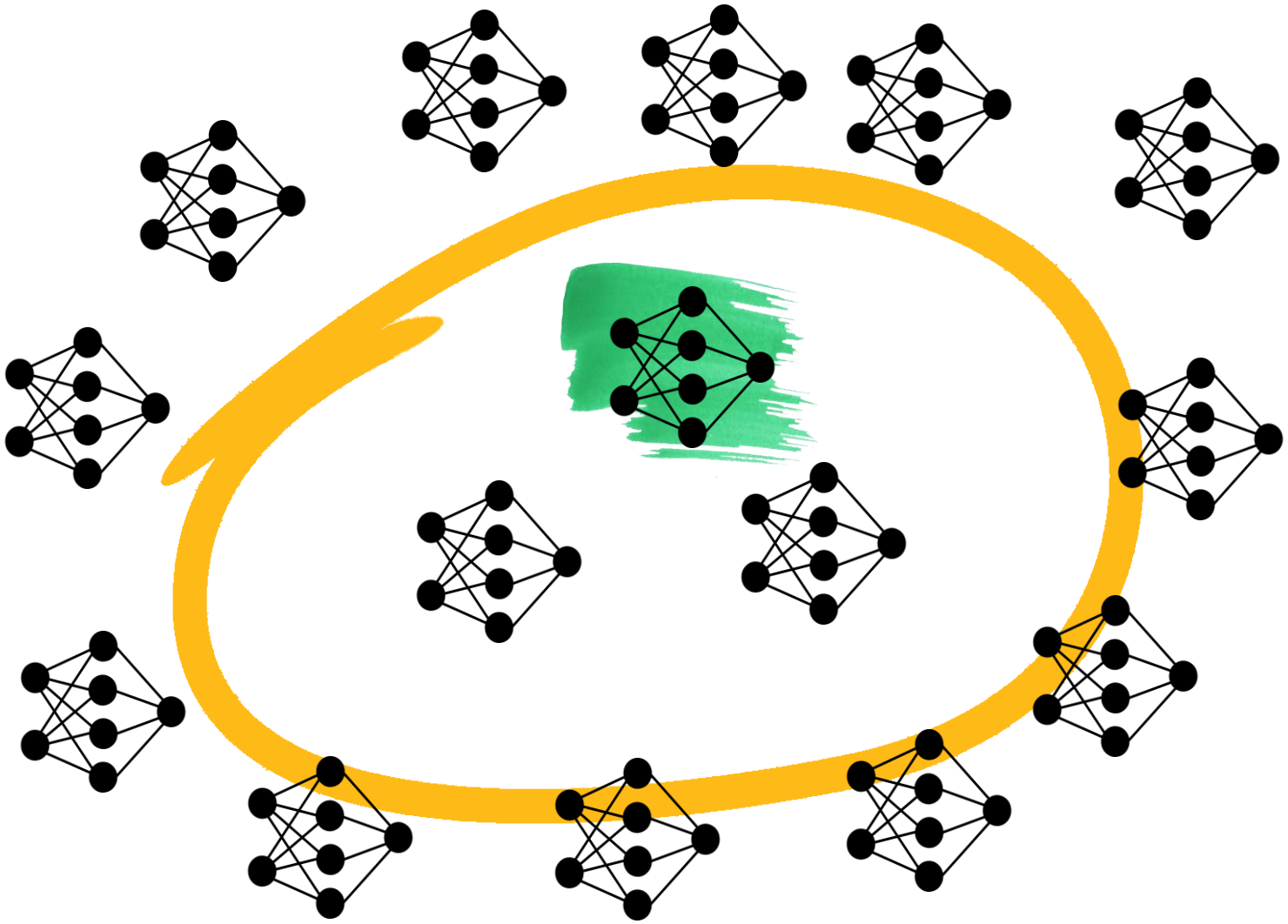
$$k = 3 < 15 = n$$



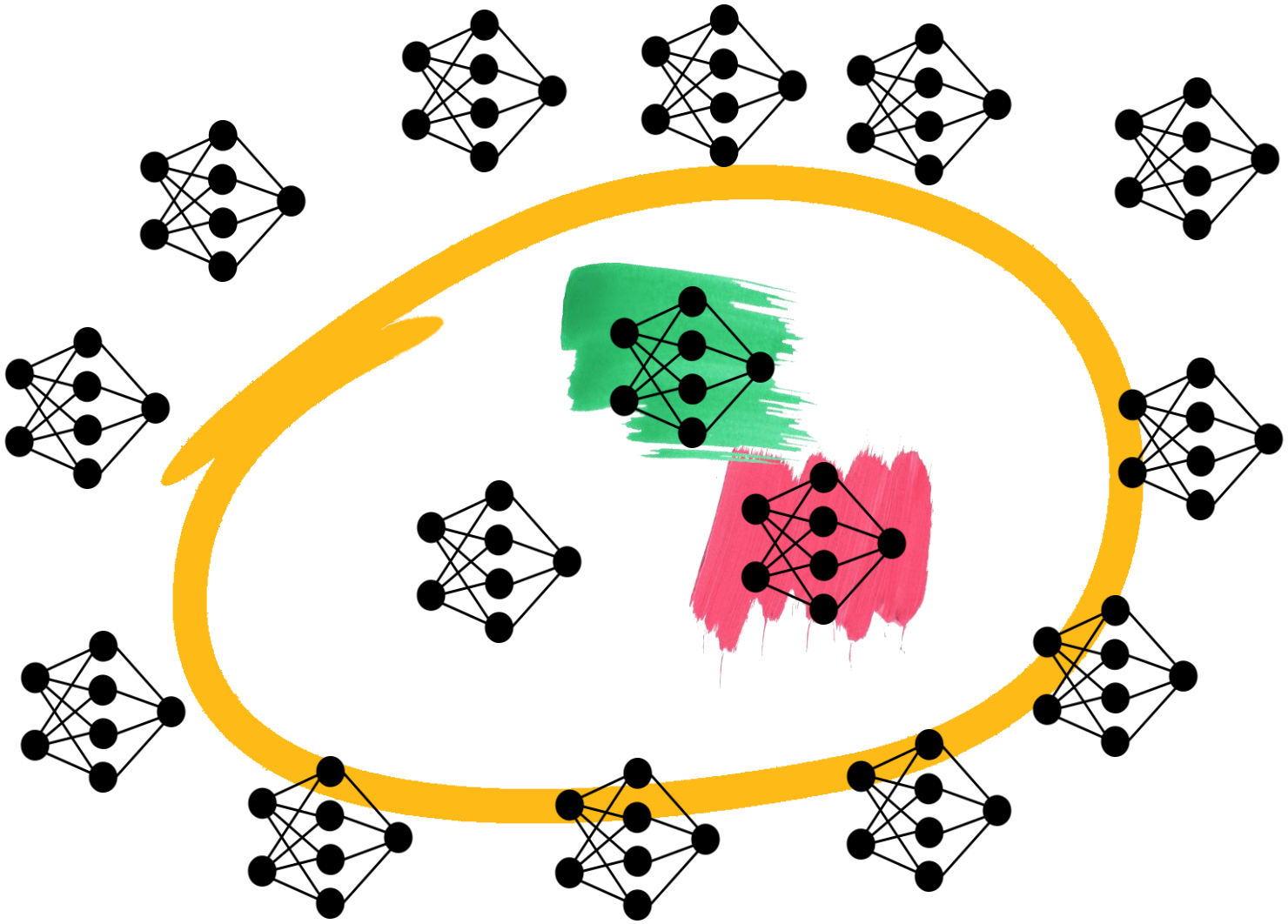
m times



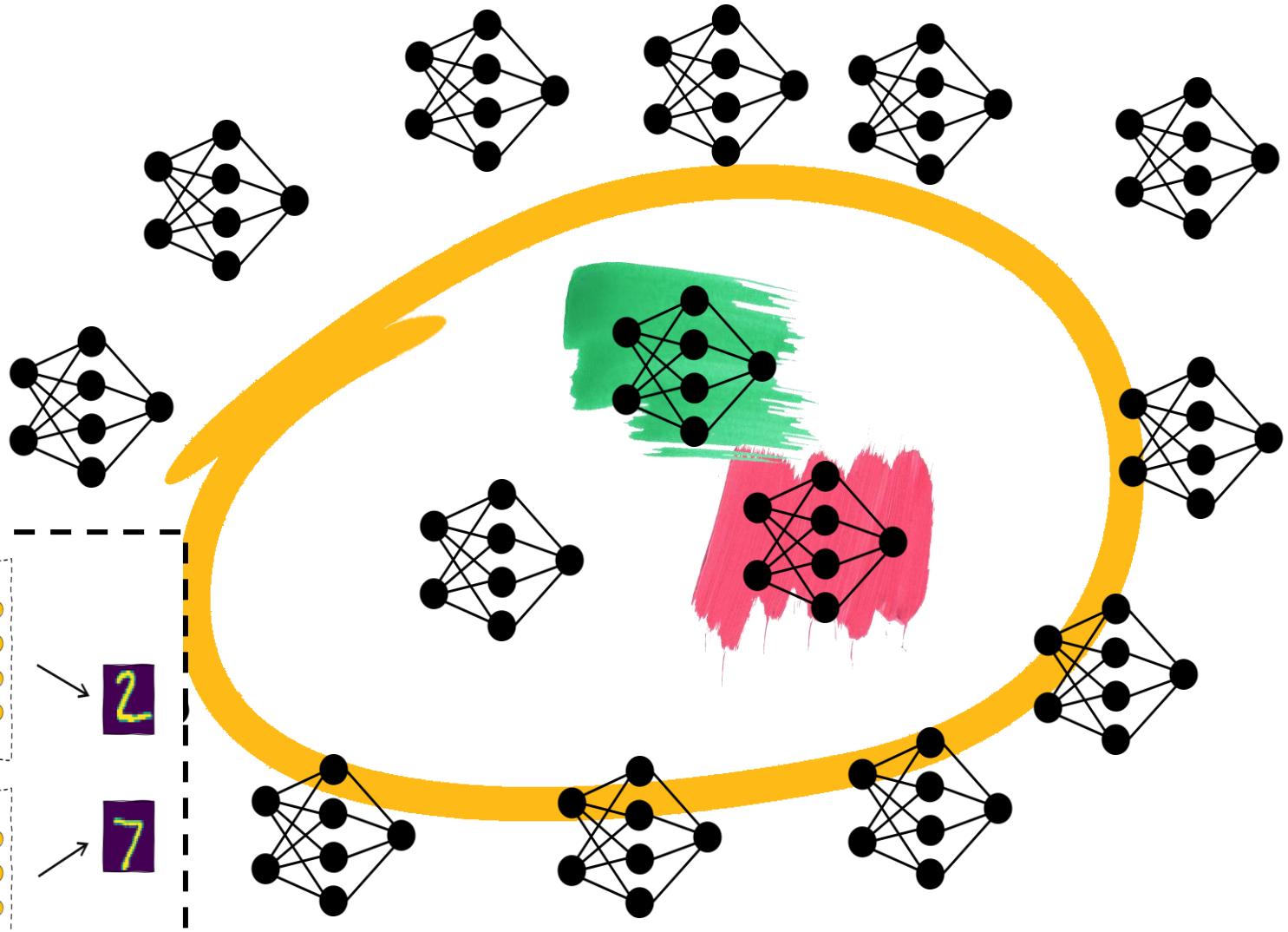
$$k = 3 < 15 = n$$



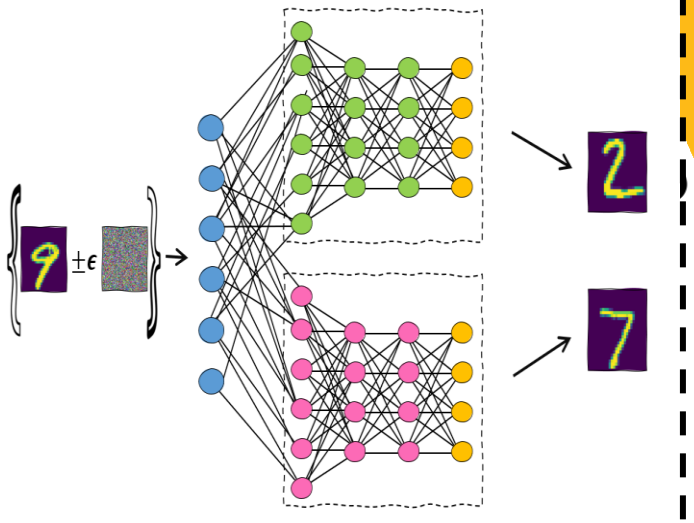
$$k = 3 < 15 = n$$

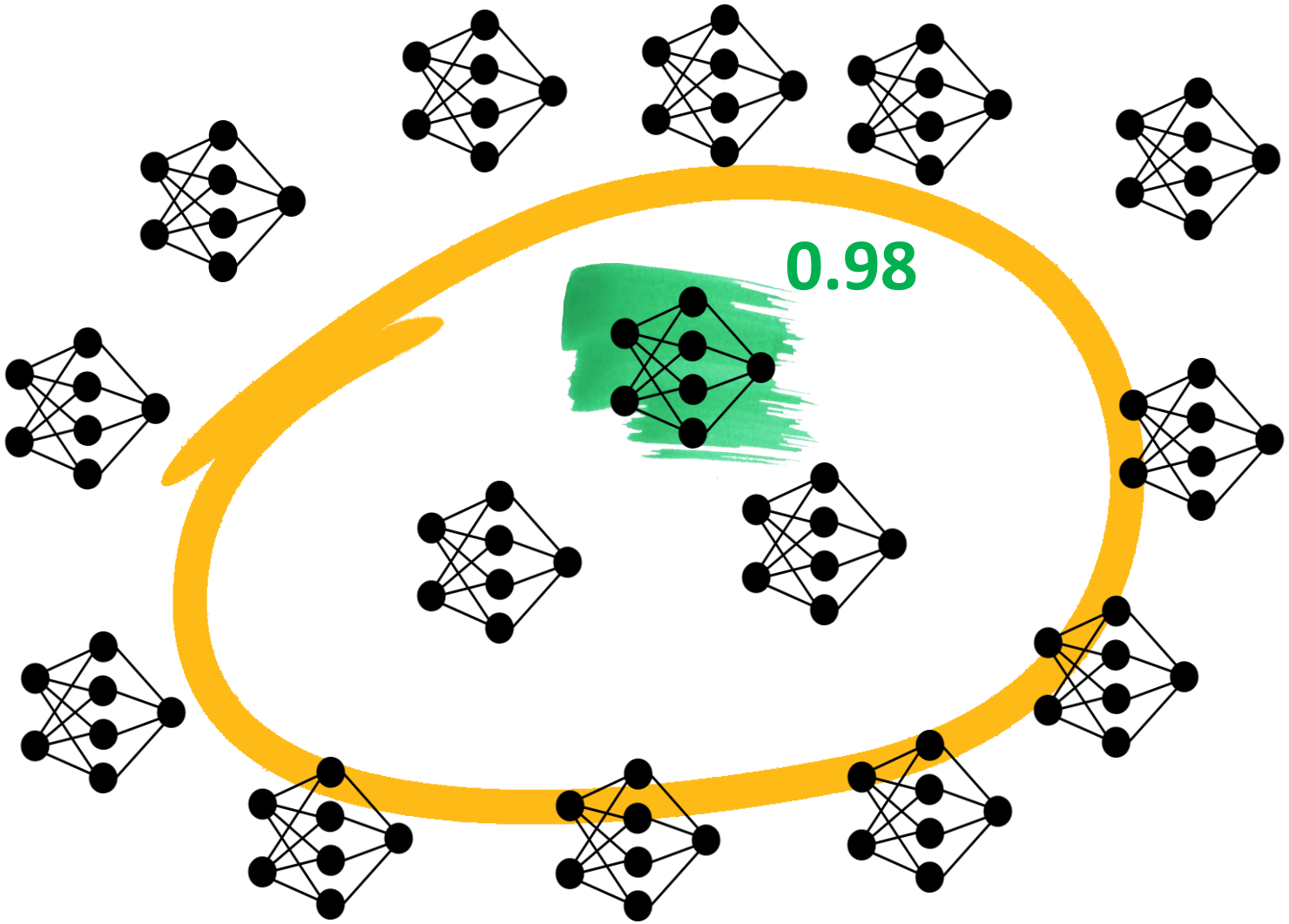


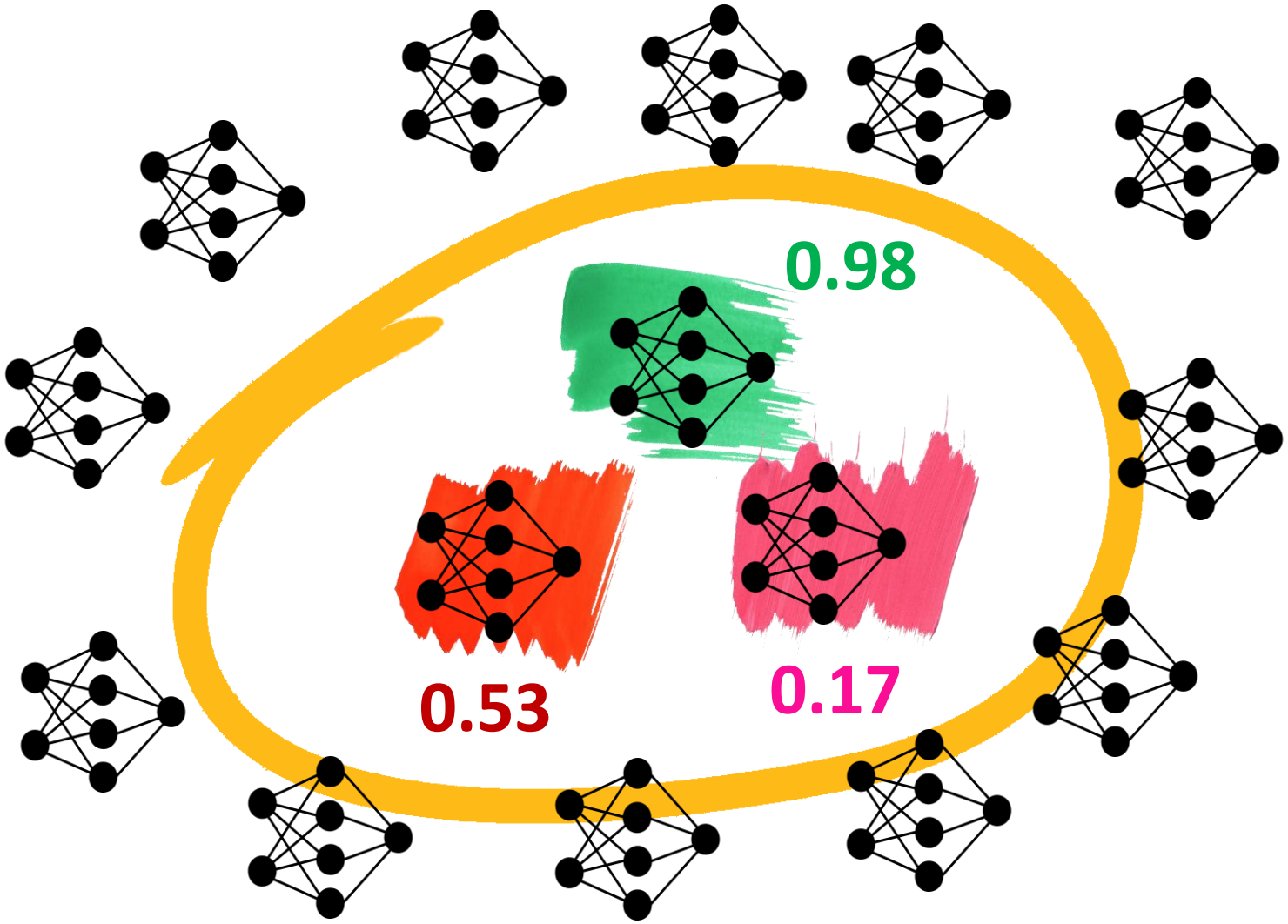
$$k = 3 < 15 = n$$

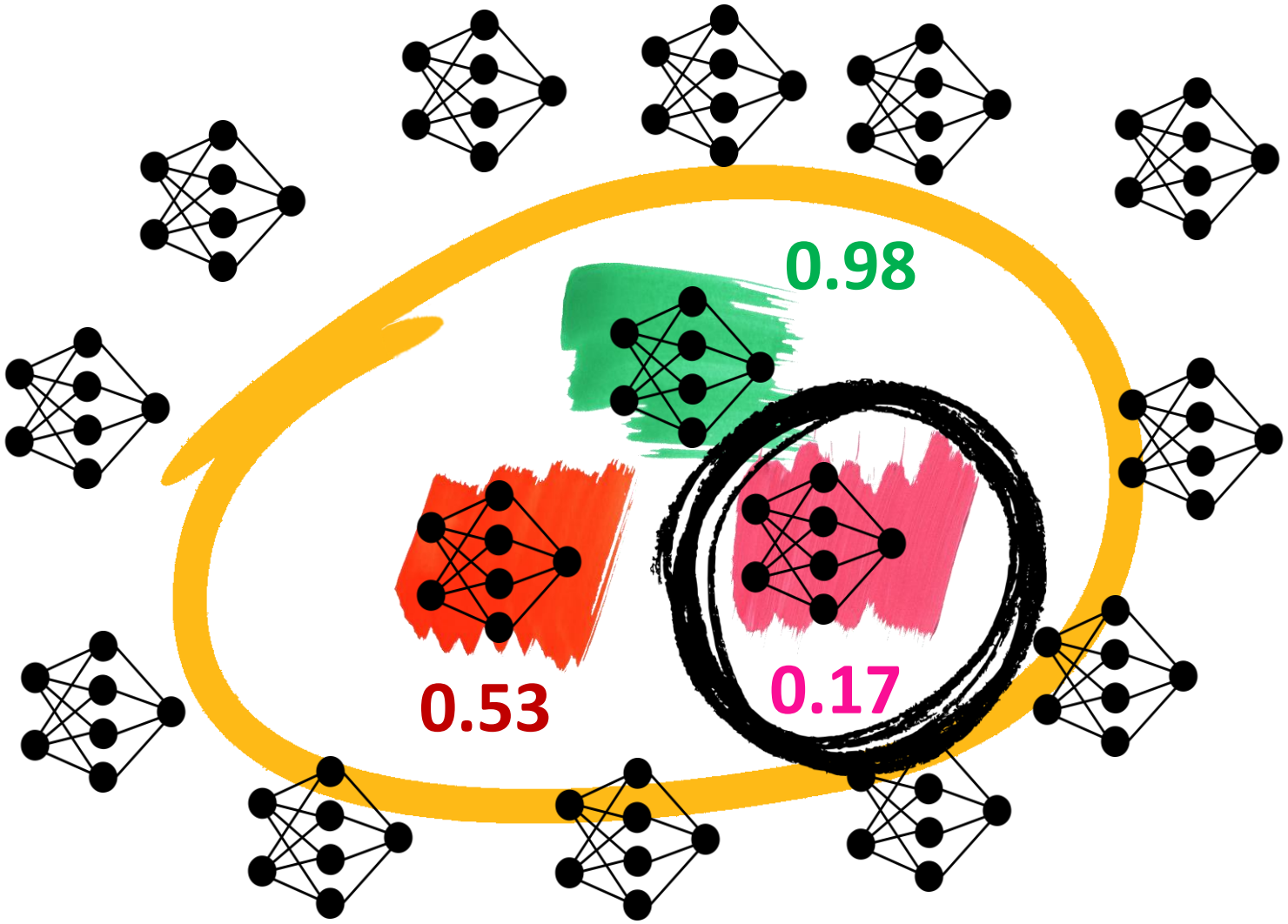


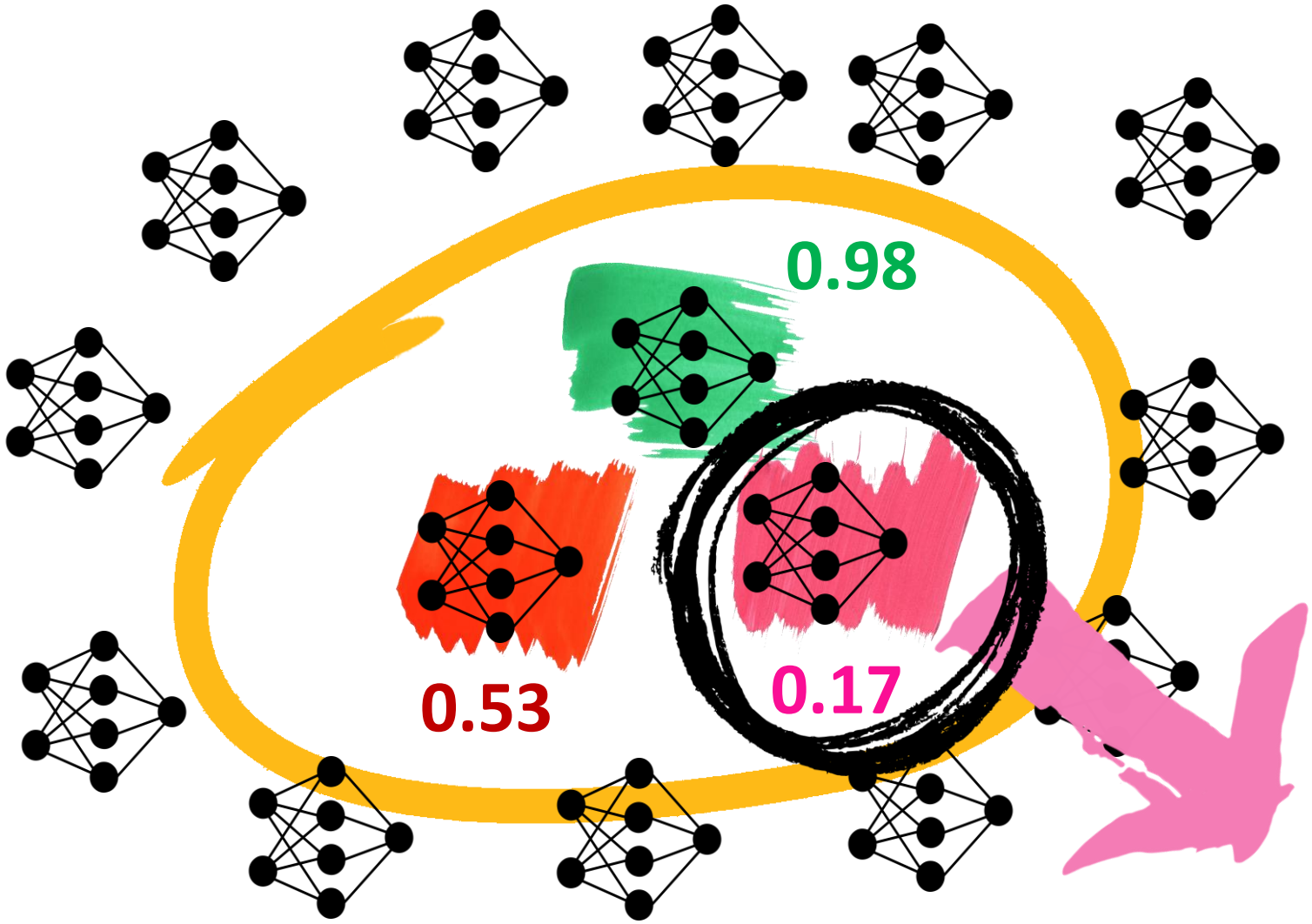
m times

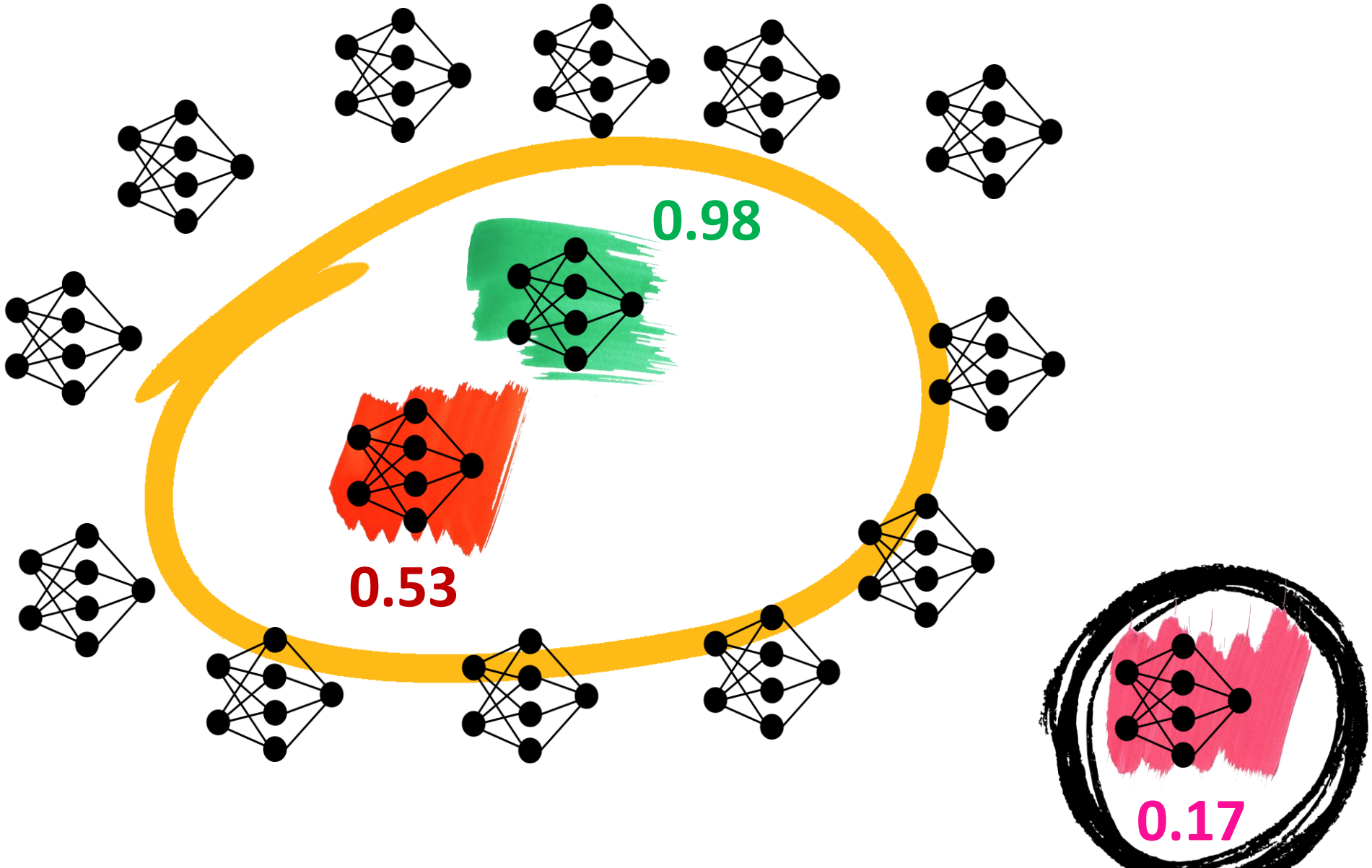


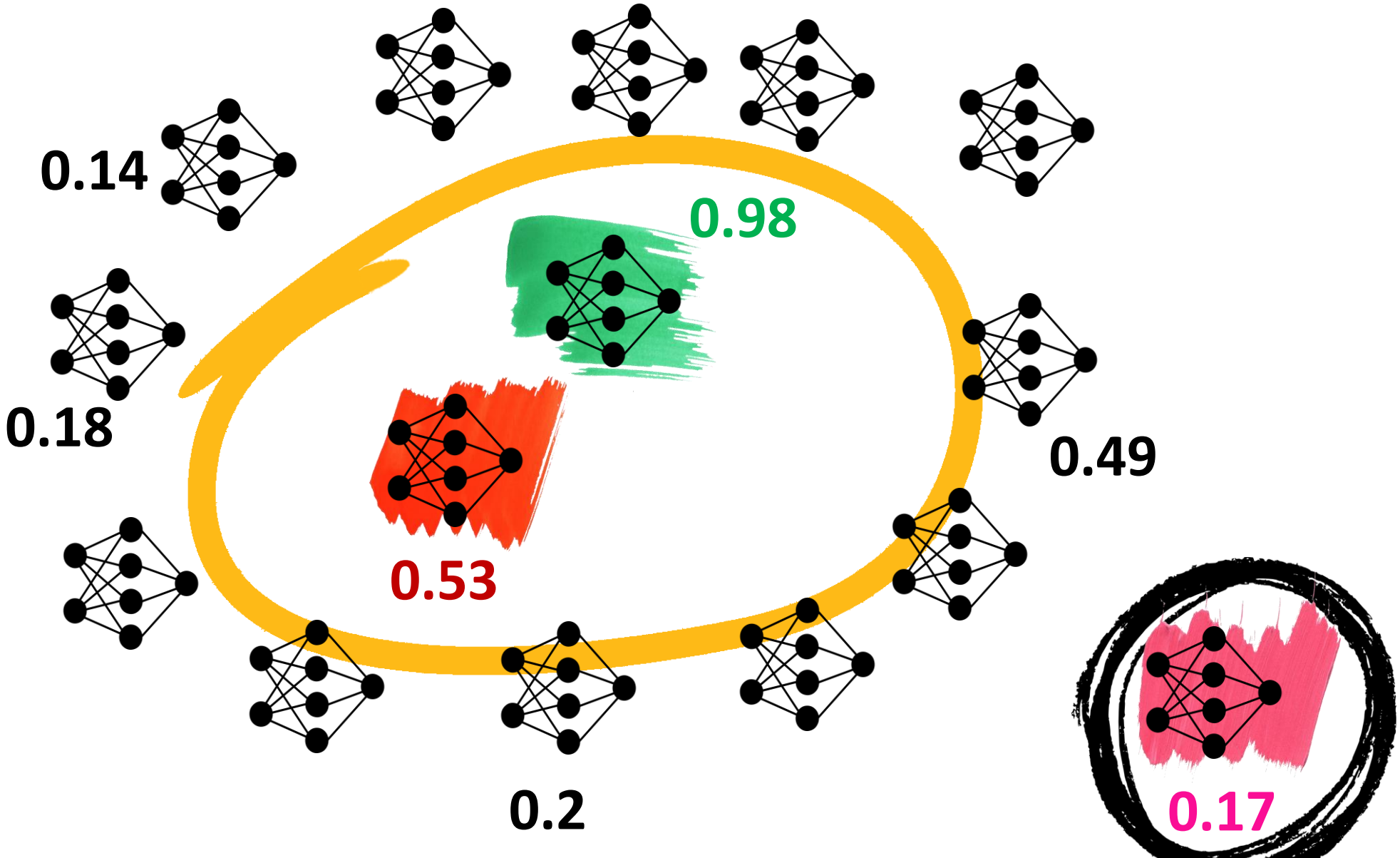

0.98

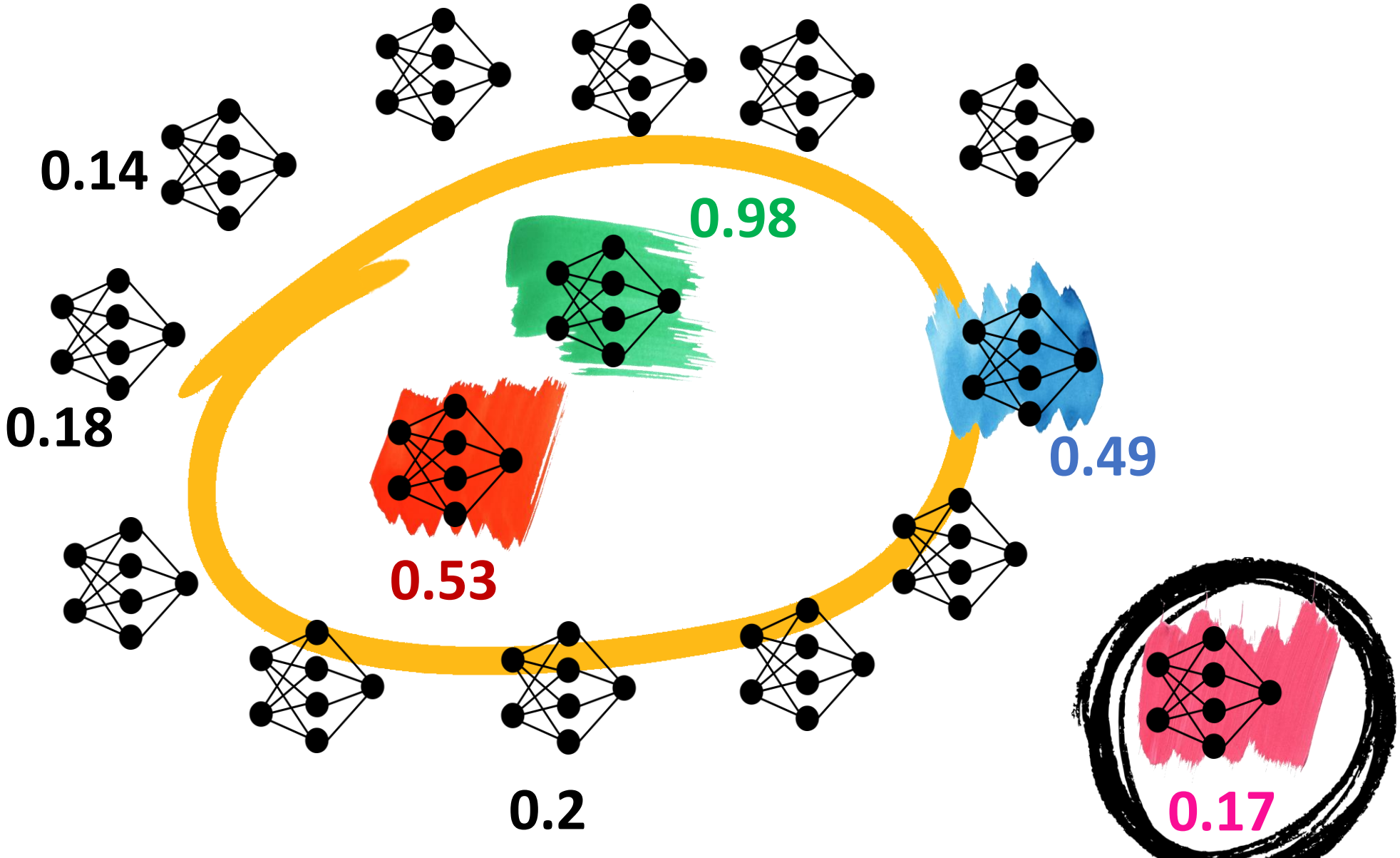

0.98
0.53
0.17

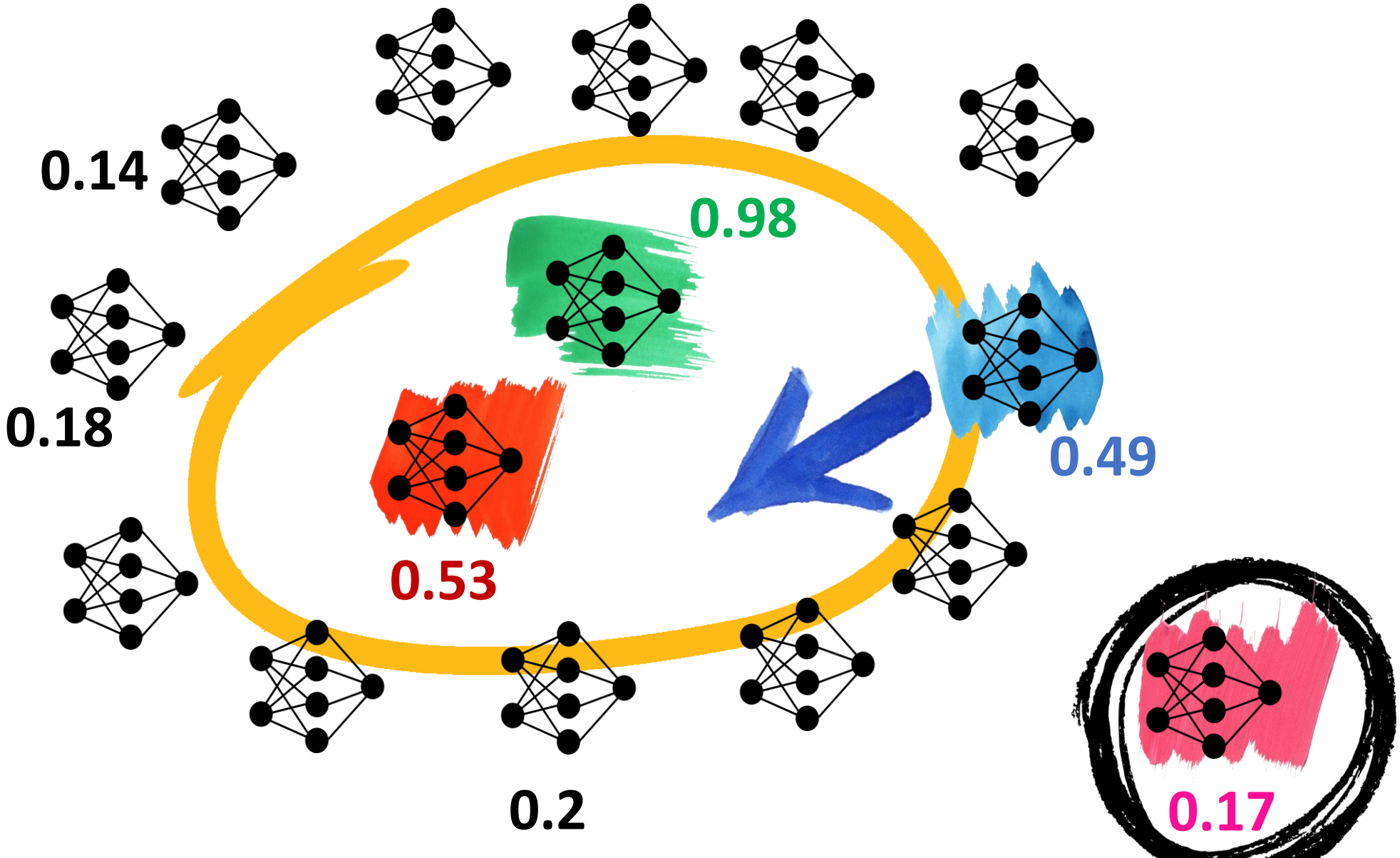

0.98
0.53
0.17

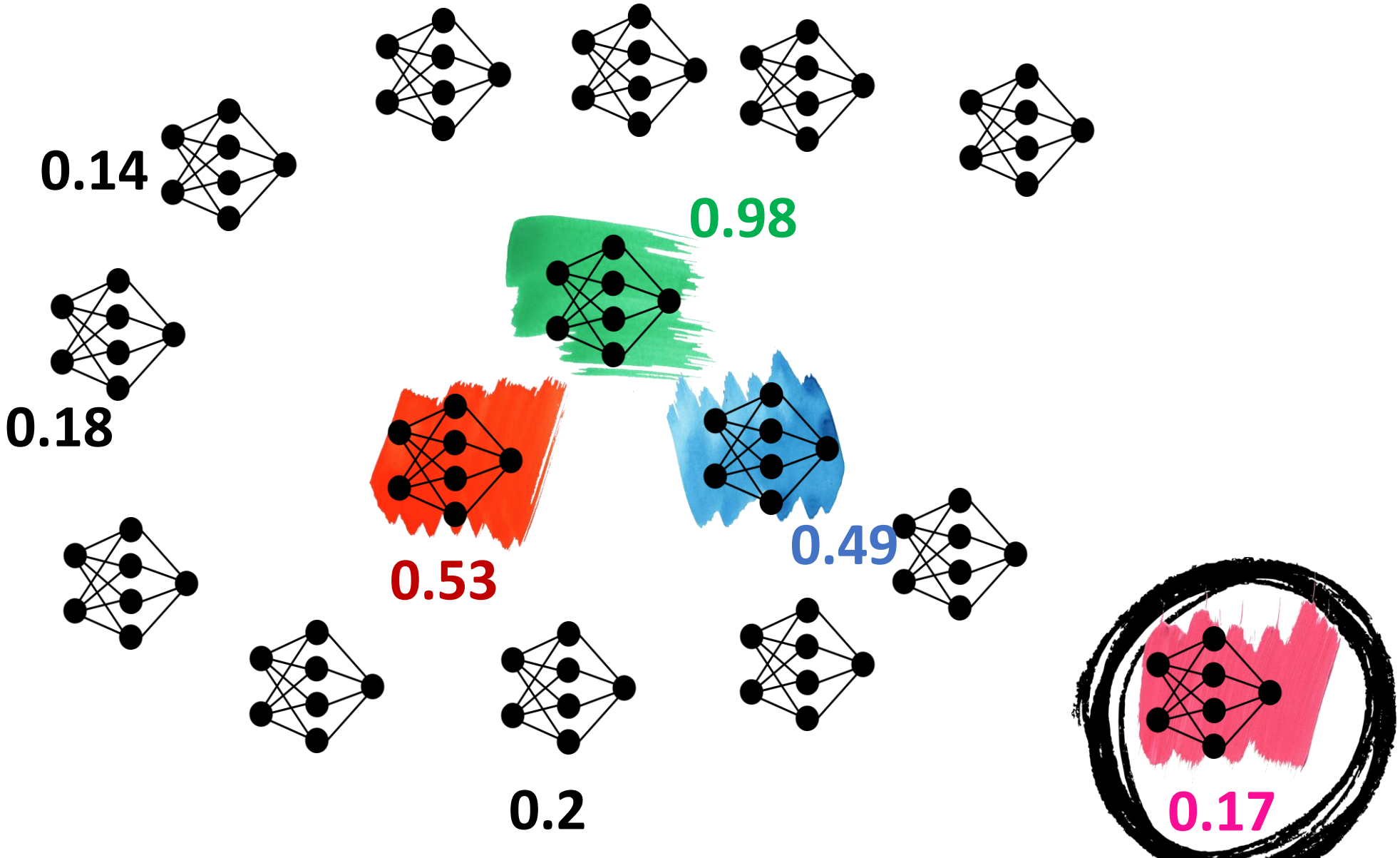

0.98
0.53
0.17

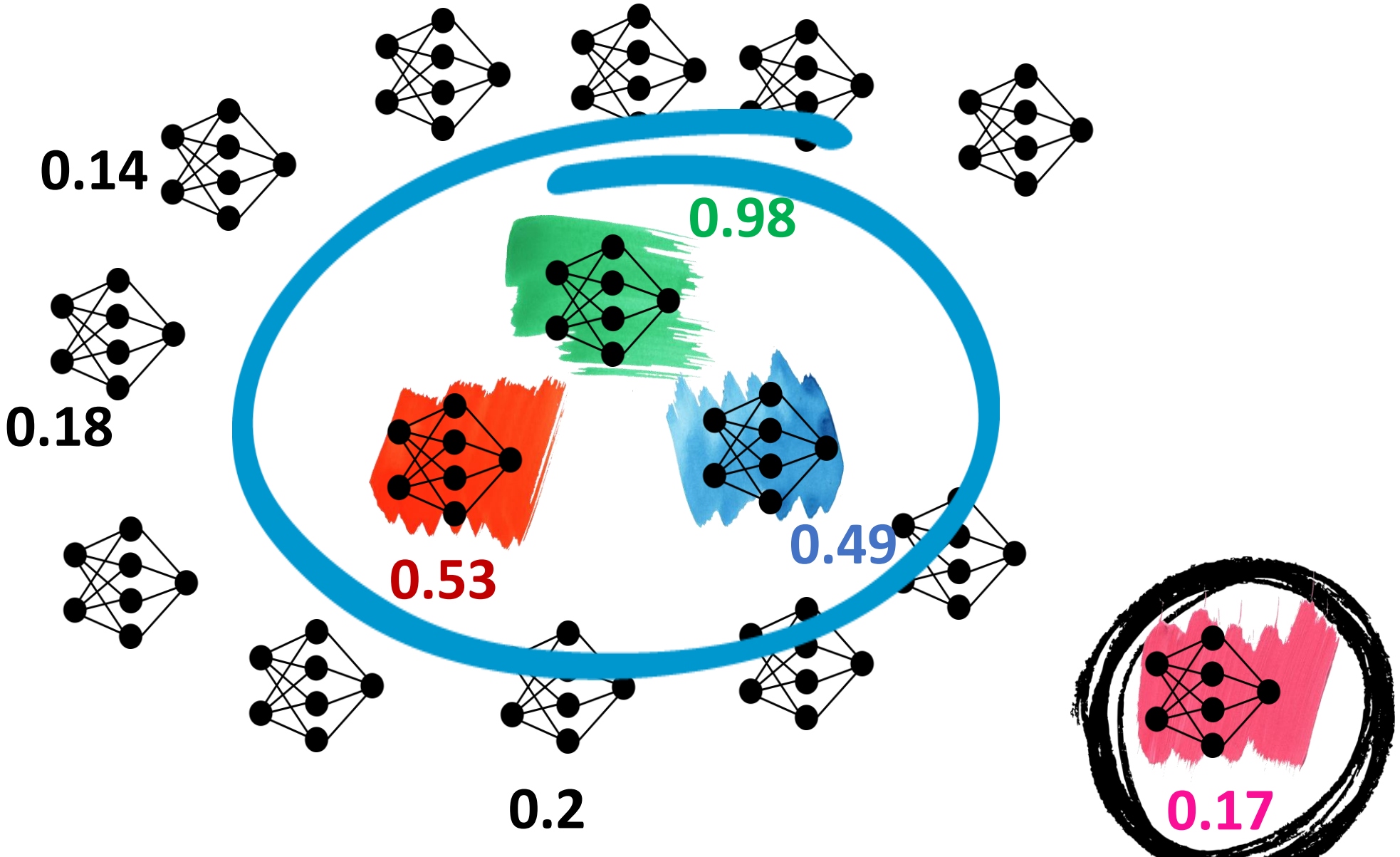

0.98
0.53
0.17


0.14
0.98
0.18
0.49
0.53
0.2
0.17


0.14
0.98
0.18
0.49
0.53
0.2
0.17


0.14
0.98
0.18
0.49
0.53
0.2
0.17


0.14
0.98
0.18
0.53
0.49
0.2
0.17


0.14
0.98
0.18
0.53
0.49
0.2
0.17

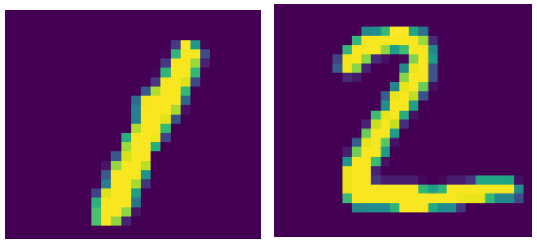
But..

But..

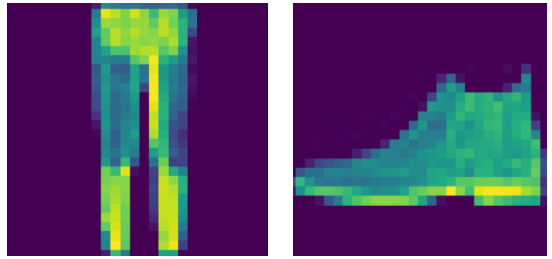
Does it improve *robustness* on non-agreement points?

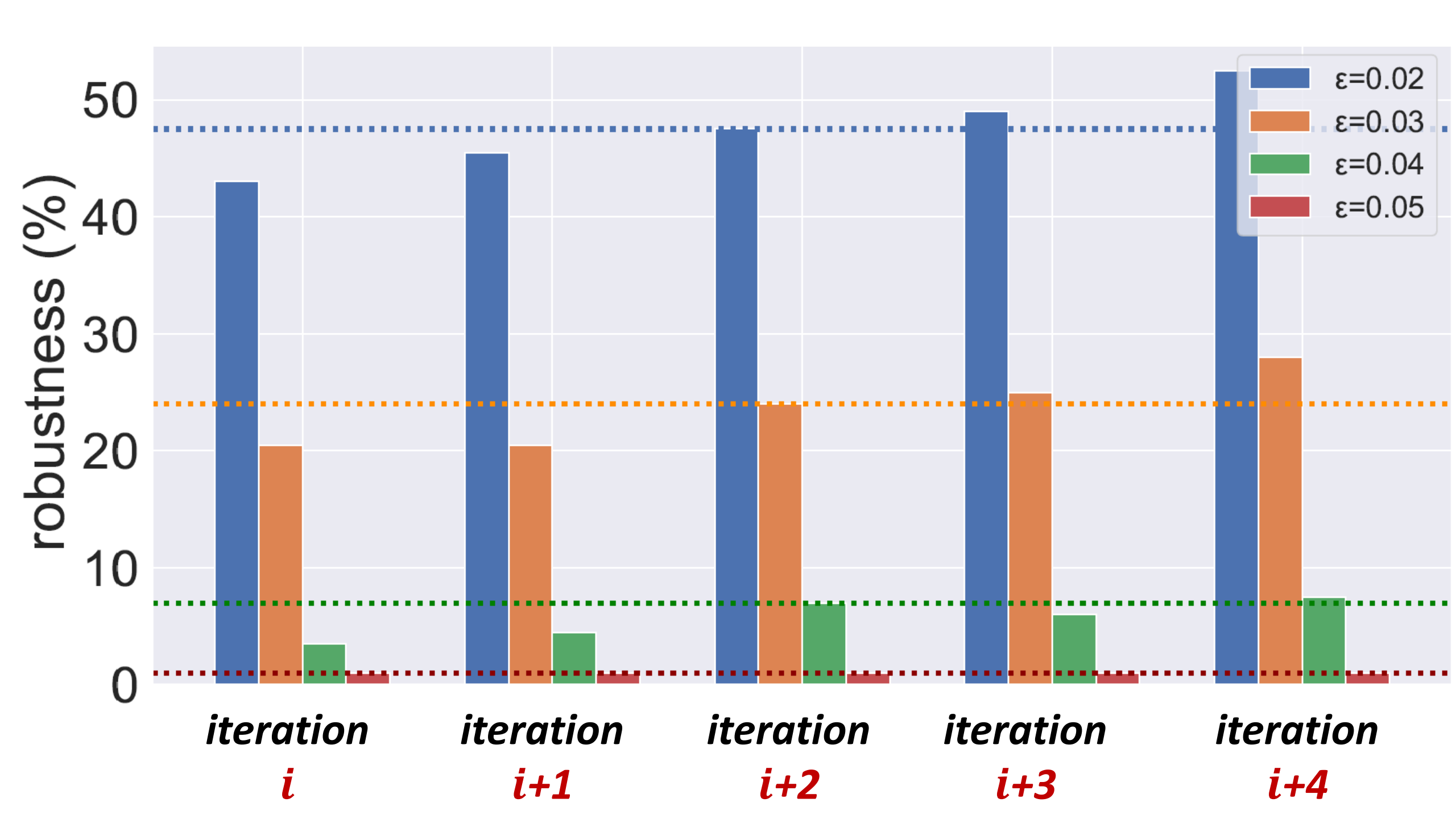
Case Studies

- [x] Train $n = 10$ DNNs $(N_1, \ldots, N_{10})$ on the **MNIST** dataset



- [x] Train $n = 10$ DNNs $(N_{11}, \ldots, N_{20})$ on the **Fashion-MNIST** dataset




robustness (%)
50
40
30
20
10
0
ε=0.02
ε=0.03
ε=0.04
ε=0.05
iteration
i
iteration
i+1
iteration
i+2
iteration
i+3
iteration
i+4

Conclusions

A **greedy-search heuristic** for verification-aided ensemble selection

Using a **polynomial** number of queries

Focusing on **agreement points** (usually) improves robustness on **additional points**

Questions