**Spreading disease through the air.**

# presentation
# summary

**01** Data exploration

*How have we built our **network** from the data ?*

**02** Graph signal processing

*How have we used the **heat kernel** to evaluate the efficiency of propagation ?*

**03** Spectral clustering

*How have we achieved meaningful **clusters** using K-Means ?*

**04** Methods & results

*What **methods** were used and how do they compare ?*

# network creating
# steps

## from data to network

- data **collected**
- 2 **datasets** :
  *routes* & *airports* ~ edges & nodes
- **adjacency matrix**

## cleaning

- **self-loops**
- **isolated** nodes

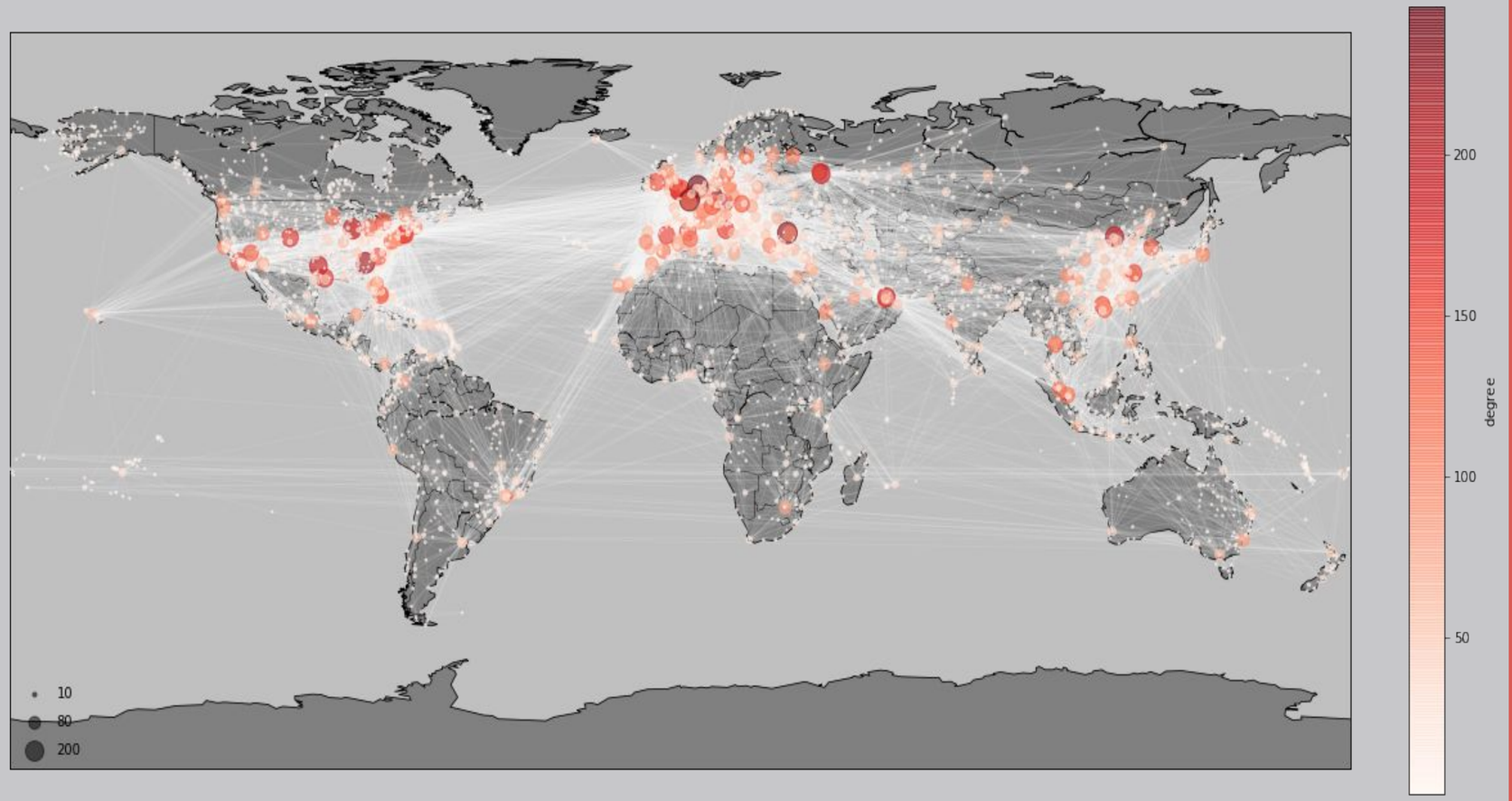
1 — 2 — 3 — 4

## enhancing

- directed > **undirected**
- adding **weights**

## giant component

- 7 **connected component**
- 1 giant component
- 6 small components




# Our network.

- 3 154 airports ~ nodes
- 18 592 edges
- 11.8 average degree
- 12 diameter
- 0.49 average clustering coefficient
- 3.96 average shortest path

# Model

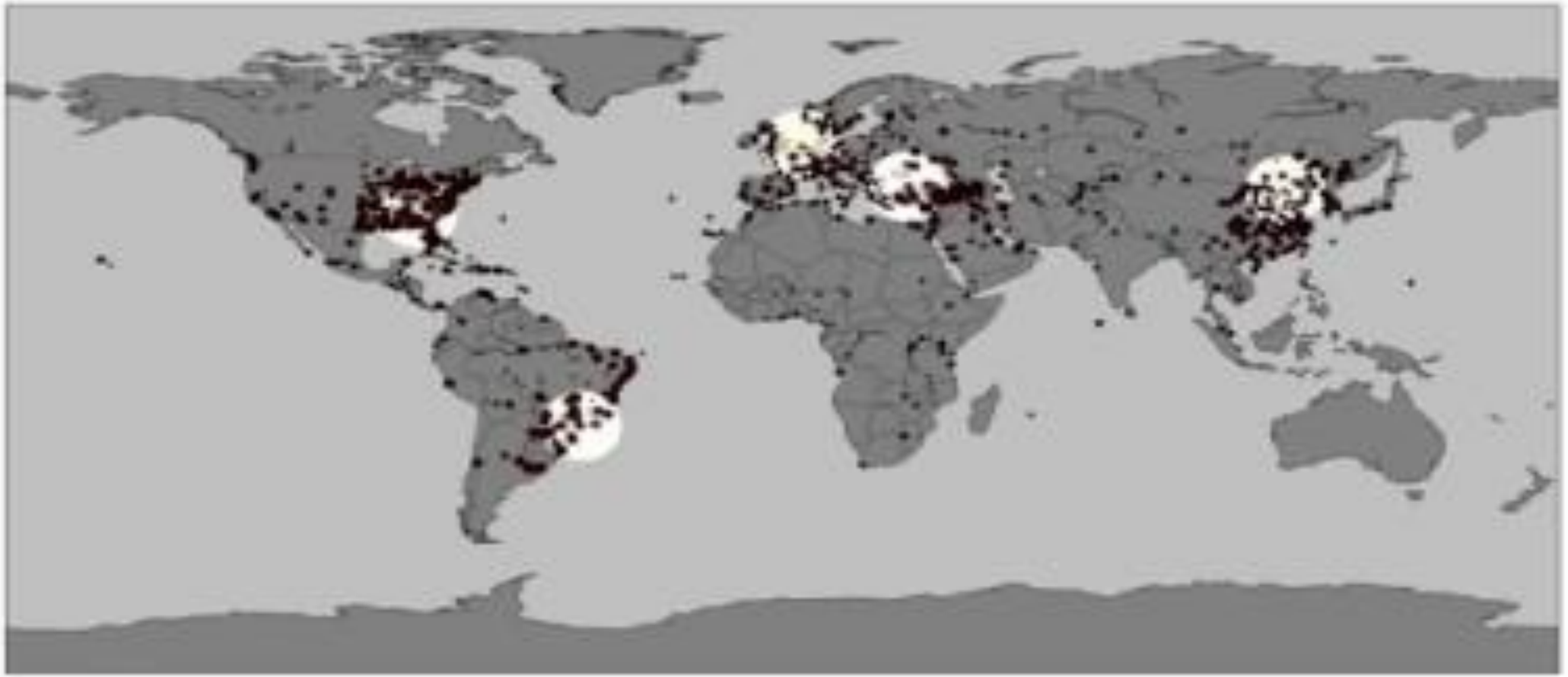
# simulation

## Modeling disease as a signal

- The **infection score** of each airport
- Initial state is a sum of 5 **Dirac** functions with magnitude ⅕

## Heat kernel describing the evolution over time

- Complexity does not depend on the number of sources
- Continuous over time



# measurement

## Signal properties

- Always sum up to **1** on the whole graph

## Our score

- Number of **infected airports**

## When is an airport infected ?

- Threshold of **1/N**

# Clustering

# motivation

## NP : cannot try all solution

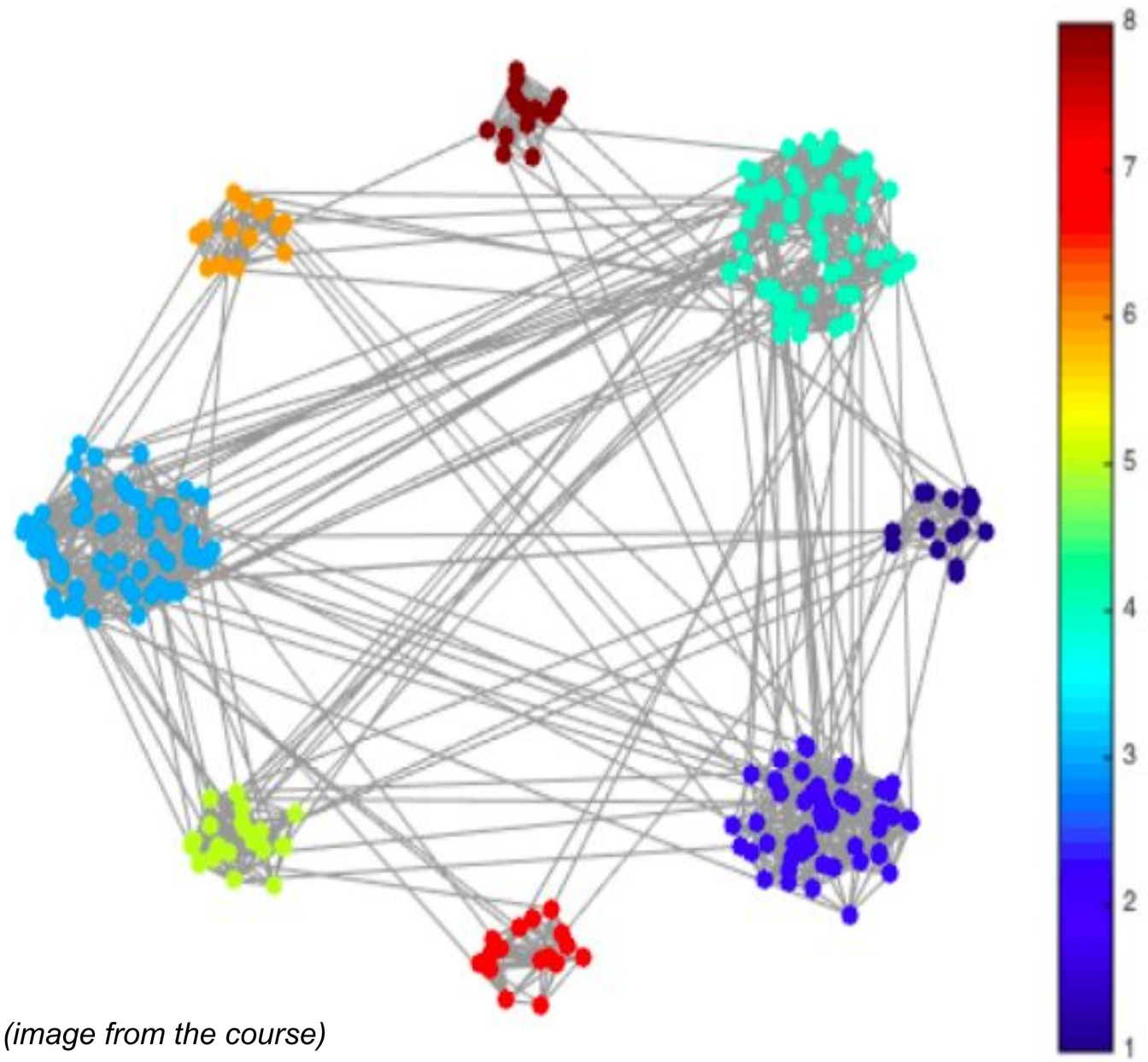
The possible *k* sources over *N* nodes is *C(N, k)*, not polynomial !

→ need to make assumptions

## Our solution : *clustering*

- Compute the effect of each source individually
- Cluster the network in k groups
- Final solution: the strongest source of each cluster

→ the clustering must be equally sized




*(image from the course)*

# Why spectral?

What we want:

- few interconnections between clusters
- many intra-connections inside each clusters
- **Balanced clusters**

=> We need to use the **Normalized** one

How:

**Algorithm:** Normalized Spectral Clustering

Compute the matrix $H$ of first $k$ eigenvectors of $\mathbf{L}_{\text{norm}}$

Apply k-means to rows of $H$ to obtain cluster assignments

# tweaking parameters

## Find the best embedding dimension d

It shouldn't be too high.

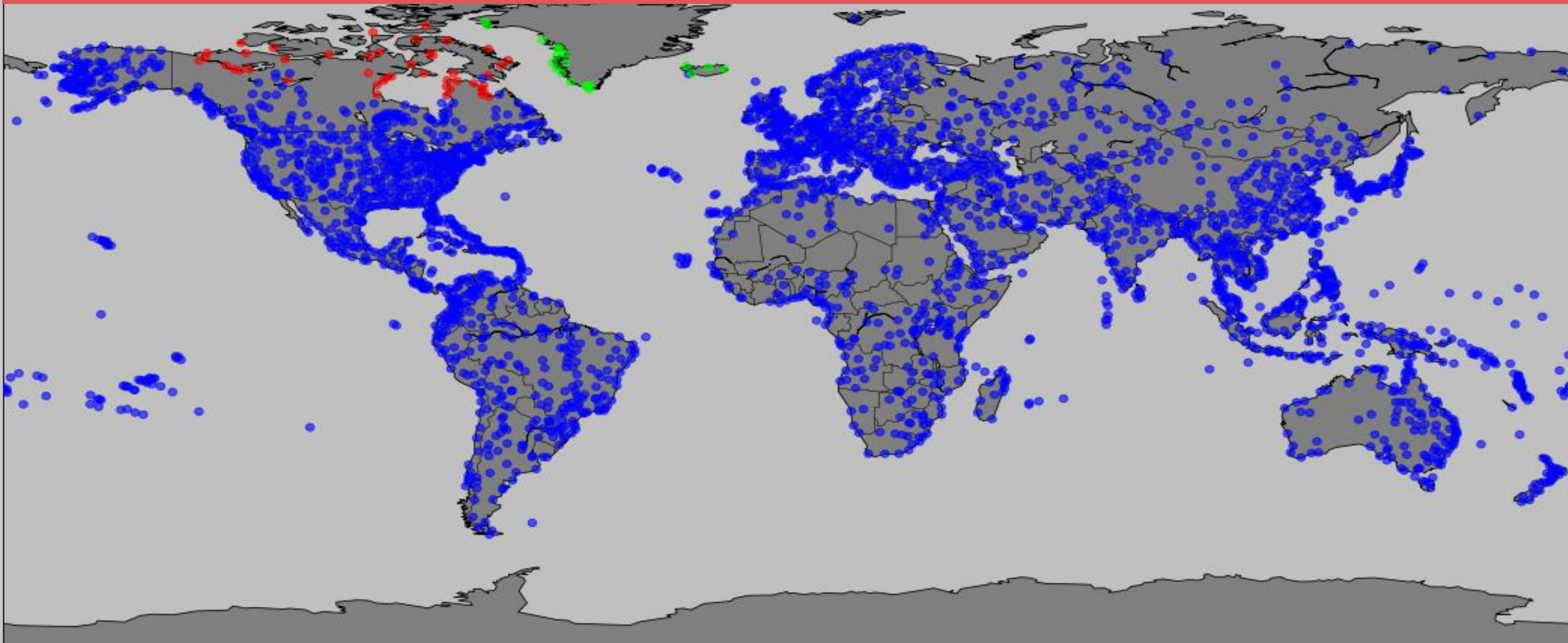
## Adapt k to the problem

For our question, just take 5.

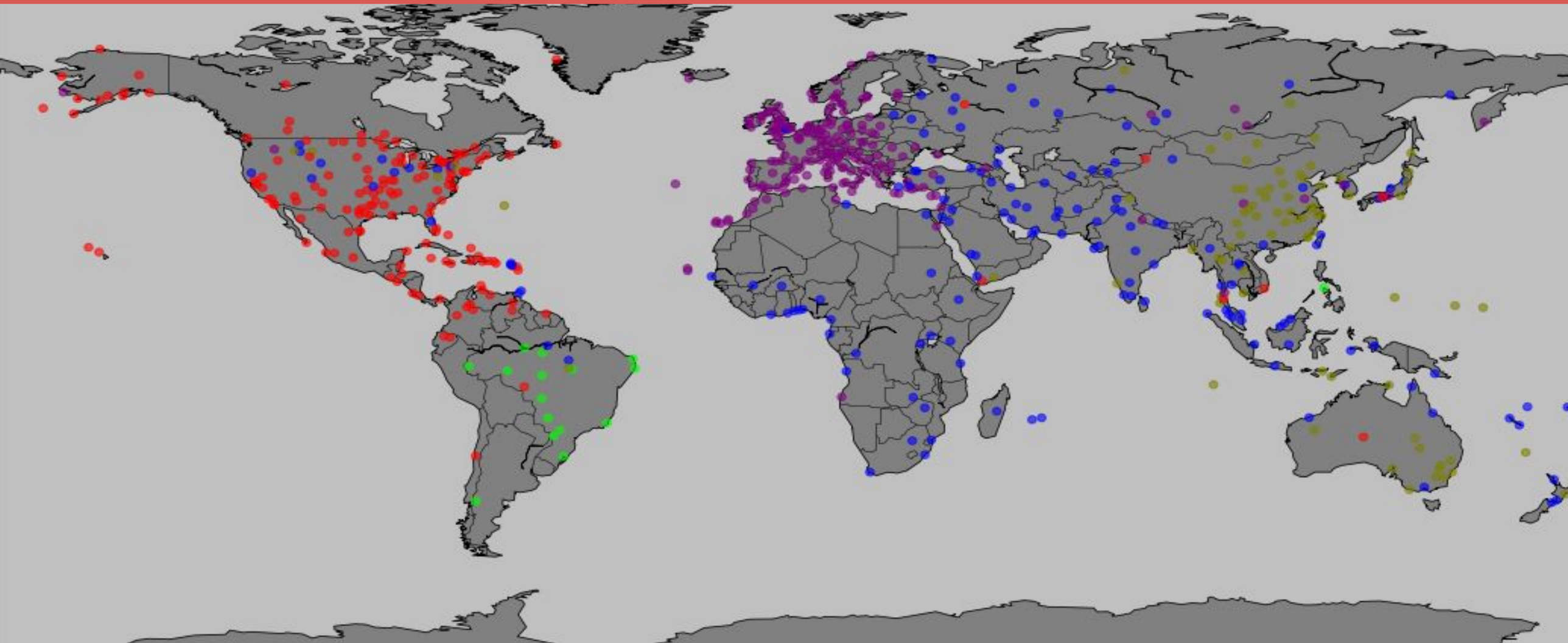
## Use Normalized Spectral Clustering

We should avoid unbalanced clusters

# Before Filtering



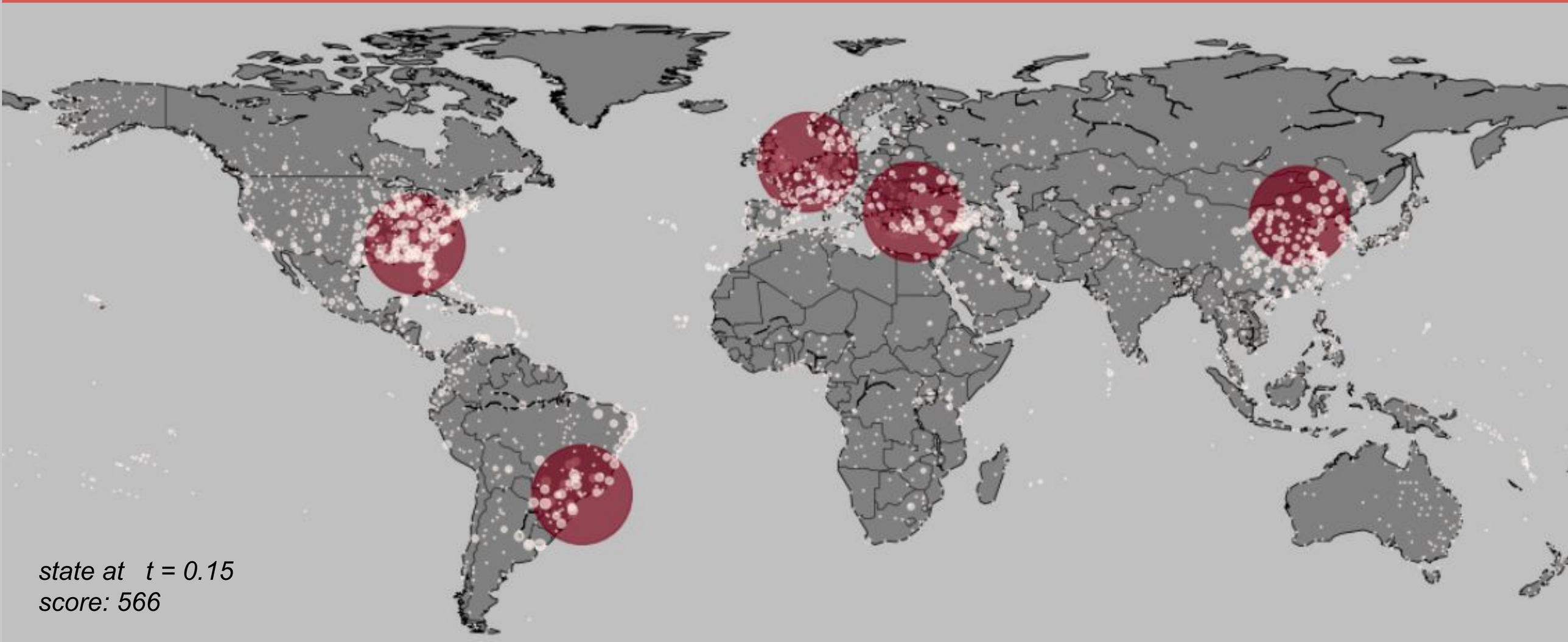
# After Filtering



# Results

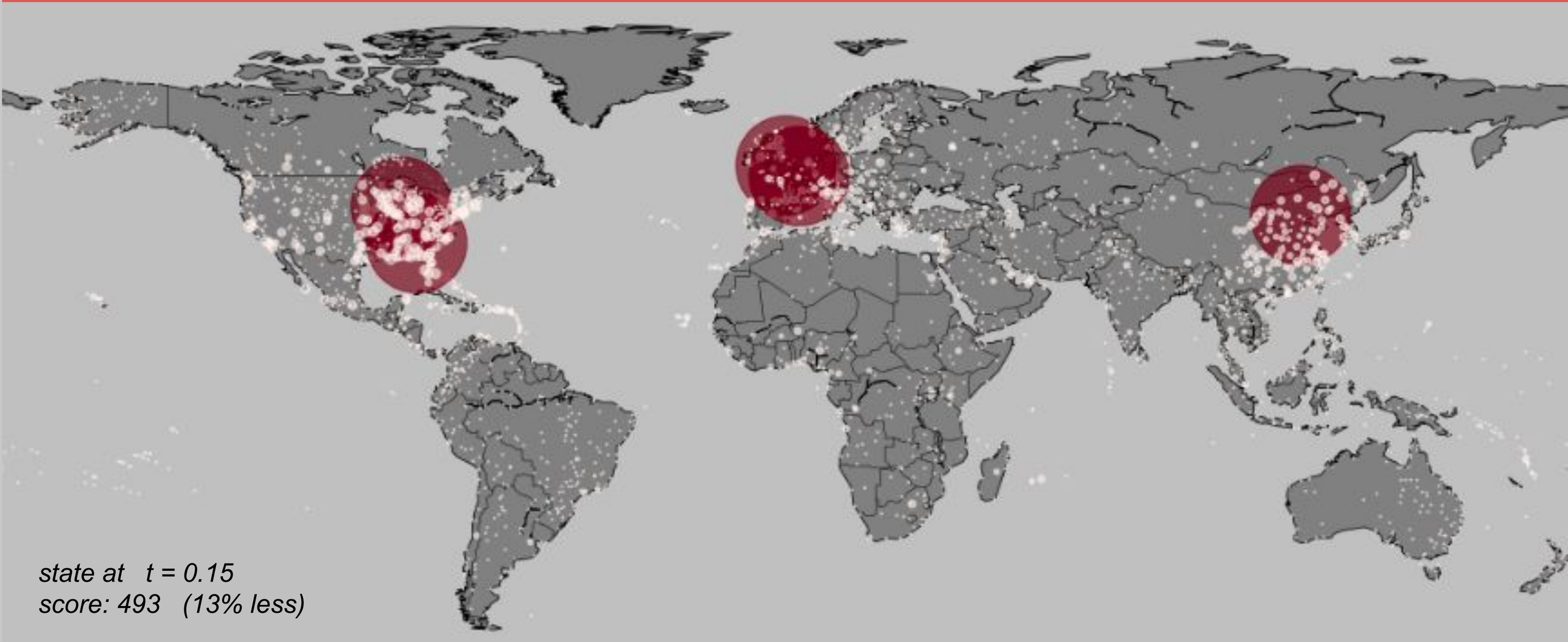
# Our solution

**Beijing, Atlanta, Sao Polo, Amsterdam, Istanbul**



# Top-5 degree

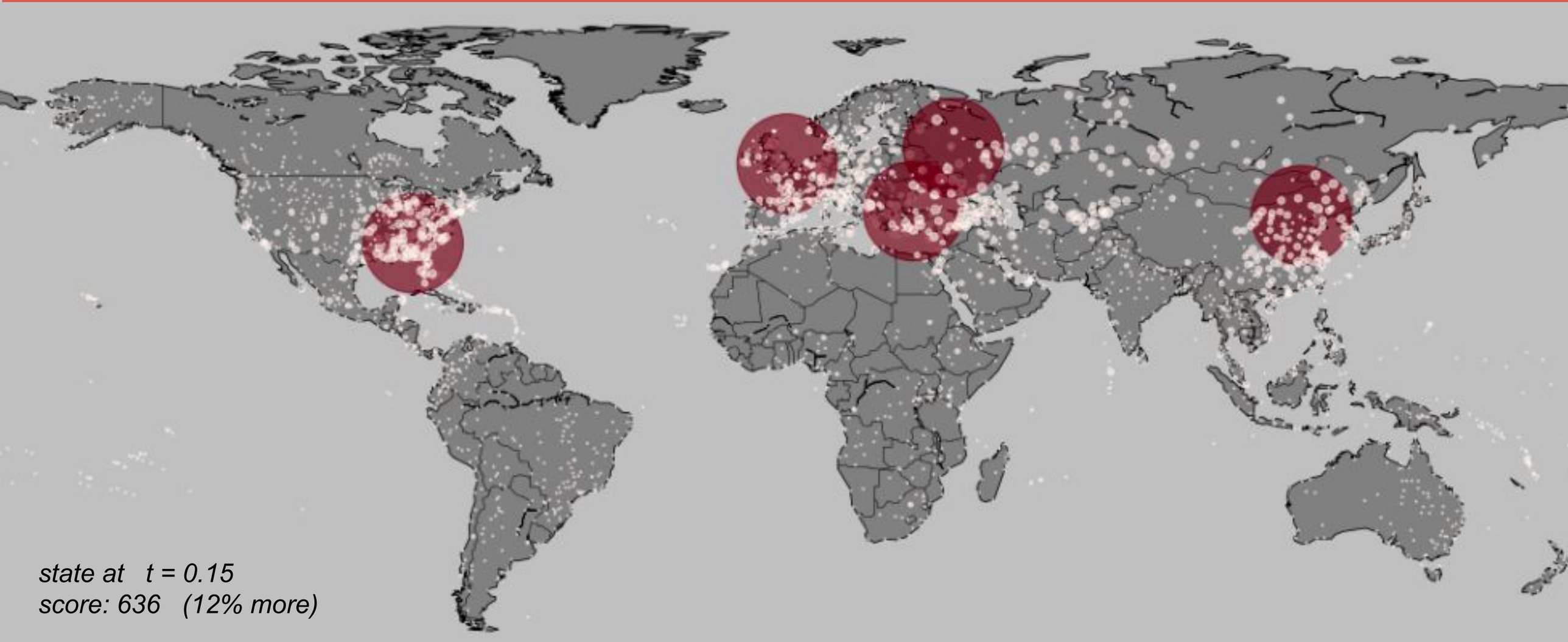
**Beijing, Atlanta, Chicago, Paris, London**



*state at t = 0.15*
*score: 493 (13% less)*

# Can we do better?

## Beijing, Atlanta, Istanbul, London, Moscow



*state at  t = 0.15*
*score: 636  (12% more)*

# conclusion?
## *“it works !”*

### We found a solution

We answered our main question

### Visually, the solution is intuitive

- The k points are all important cities
- The k points are well spread geographically

### The whole process is efficient

- Complexity does not explode even when k is large

# the end

## Questions

# thank you.

# k=20

# k=50

# k=100