# Community detection on geometric graphs

A local-to-global perspective

**B. R. Vinay Kumar**

Indian Institute of Technology-Bombay
August 19, 2024
Mumbai, India

# Background

# Background

**Capture real-world phenomena using ideas from random graphs and network science.**

# Background

**Capture real-world phenomena using ideas from random graphs and network science.**

**Wireless networks**

# Background

**Capture real-world phenomena using ideas from random graphs and network science.**

Wireless networks

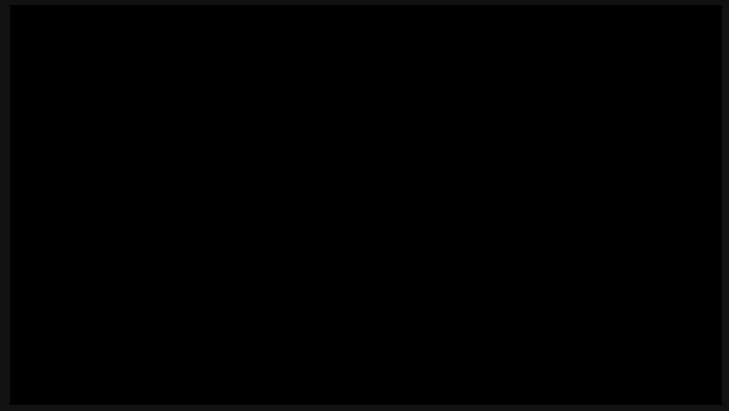

Social networks

Transportation networks

Data science

# Background

**Capture real-world phenomena using ideas from random graphs and network science.**



**Wireless networks**


**Social networks**

**Data science**

- Geometric graphs? Vertices are embedded in space and edges depend on the distance between nodes.
- Presence of short edges and abundance of triangles.

**Questions?**

1. How does the network structure affect processes or information on the network?
2. Can local algorithms help to solve a global problem?
3. Does geometry help in solving a global problem efficiently?

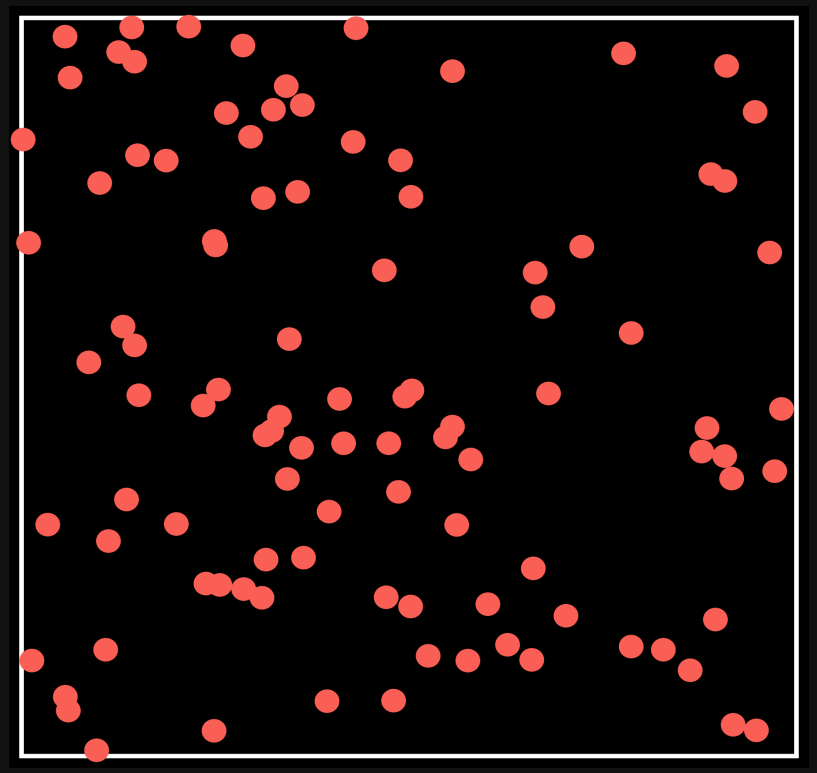
# Model for geometric graphs

**Poisson point process on** $S = \left[\frac{-1}{2}, \frac{1}{2}\right]^d$

Given $n \geq 1$ and $\lambda > 0$,

- Sample $N \sim \mathrm{Poi}(\lambda n)$.
- Choose $X_1, X_2, \cdots, X_N$ uniformly from $S$.
- The collection $\mathbb{X} = \{X_u\}_{u=1}^{N}$ forms PPP$(\lambda n)$.

# Model for geometric graphs

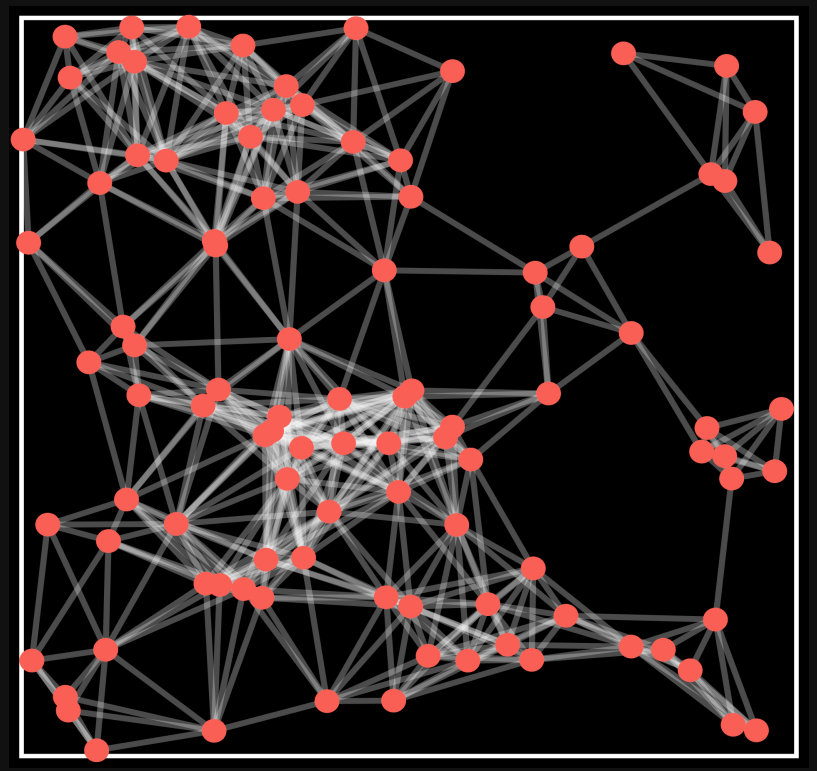


**Poisson point process on** $S = \left[\frac{-1}{2}, \frac{1}{2}\right]^d$

Given $n \geq 1$ and $\lambda > 0$,

- Sample $N \sim \text{Poi}(\lambda n)$.
- Choose $X_1, X_2, \cdots, X_N$ uniformly from $S$.
- The collection $\mathbf{X} = \{X_u\}_{u=1}^{N}$ forms PPP($\lambda n$).

# Model for geometric graphs



**Poisson point process on** $S = \left[\frac{-1}{2}, \frac{1}{2}\right]^d$

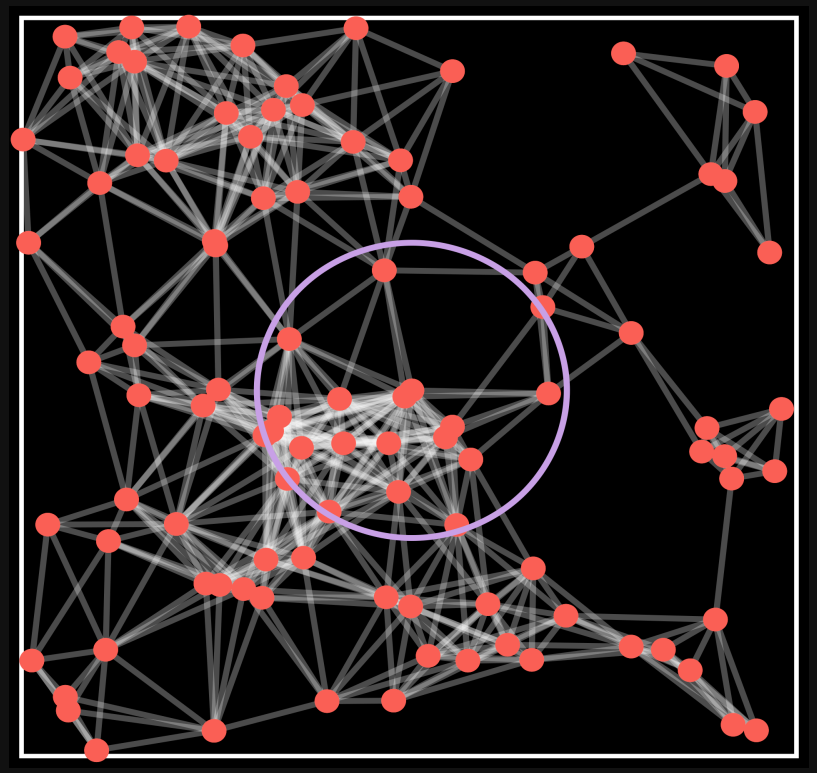
Given $n \geq 1$ and $\lambda > 0$,

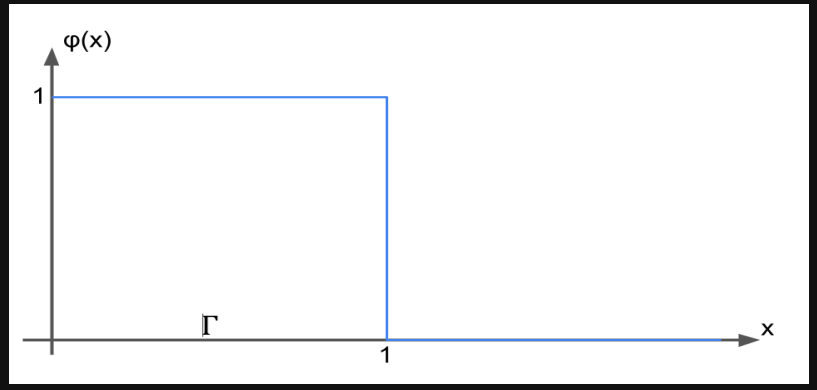
- Sample $N \sim \mathrm{Poi}(\lambda n)$.
- Choose $X_1, X_2, \cdots, X_N$ uniformly from $S$.
- The collection $\mathbf{X} = \{X_u\}_{u=1}^{N}$ forms PPP($\lambda n$).

**Random connection model**

- Geometric kernel: $\varphi : \mathrm{R}_+ \to [0, 1]$.
- Connect two nodes at $X_u$ and $X_v$ with probability $\varphi\left(\frac{\|X_u - X_v\|}{\rho_n^{1/d}}\right)$, where $\rho_n = \frac{\log n}{n}$

# Model for geometric graphs






**Poisson point process on** $S = \left[\frac{-1}{2}, \frac{1}{2}\right]^d$

Given $n \geq 1$ and $\lambda > 0$,

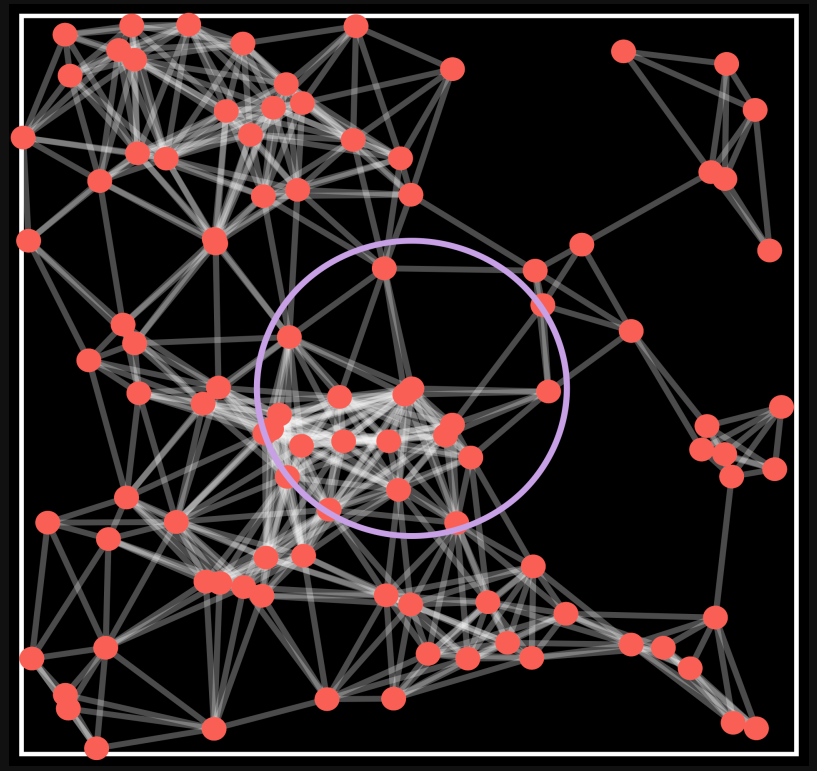
- Sample $N \sim \mathrm{Poi}(\lambda n)$.
- Choose $X_1, X_2, \cdots, X_N$ uniformly from $S$.
- The collection $\mathbf{X} = \{X_u\}_{u=1}^{N}$ forms PPP($\lambda n$).

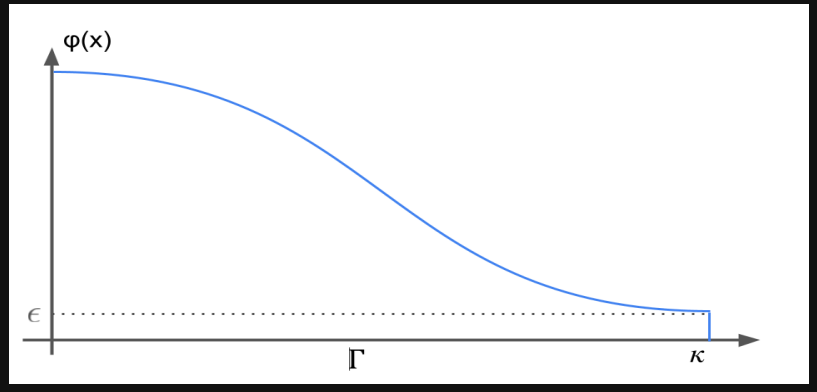
**Random connection model**

- Geometric kernel: $\varphi : \mathrm{R}_+ \to [0, 1]$.
- Connect two nodes at $X_u$ and $X_v$ with probability $\varphi\left(\frac{\|X_u - X_v\|}{\rho_n^{1/d}}\right)$, where $\rho_n = \frac{\log n}{n}$

- Examples:
  1. $\varphi(r) = \mathbf{1}\{r \leq 1\}$: random geometric graph
  2. General kernels

# Model for geometric graphs






**Poisson point process on** $S = \left[\frac{-1}{2}, \frac{1}{2}\right]^d$

Given $n \geq 1$ and $\lambda > 0$,

- Sample $N \sim \mathrm{Poi}(\lambda n)$.
- Choose $X_1, X_2, \cdots, X_N$ uniformly from $S$.
- The collection $\mathbf{X} = \{X_u\}_{u=1}^{N}$ forms PPP($\lambda n$).

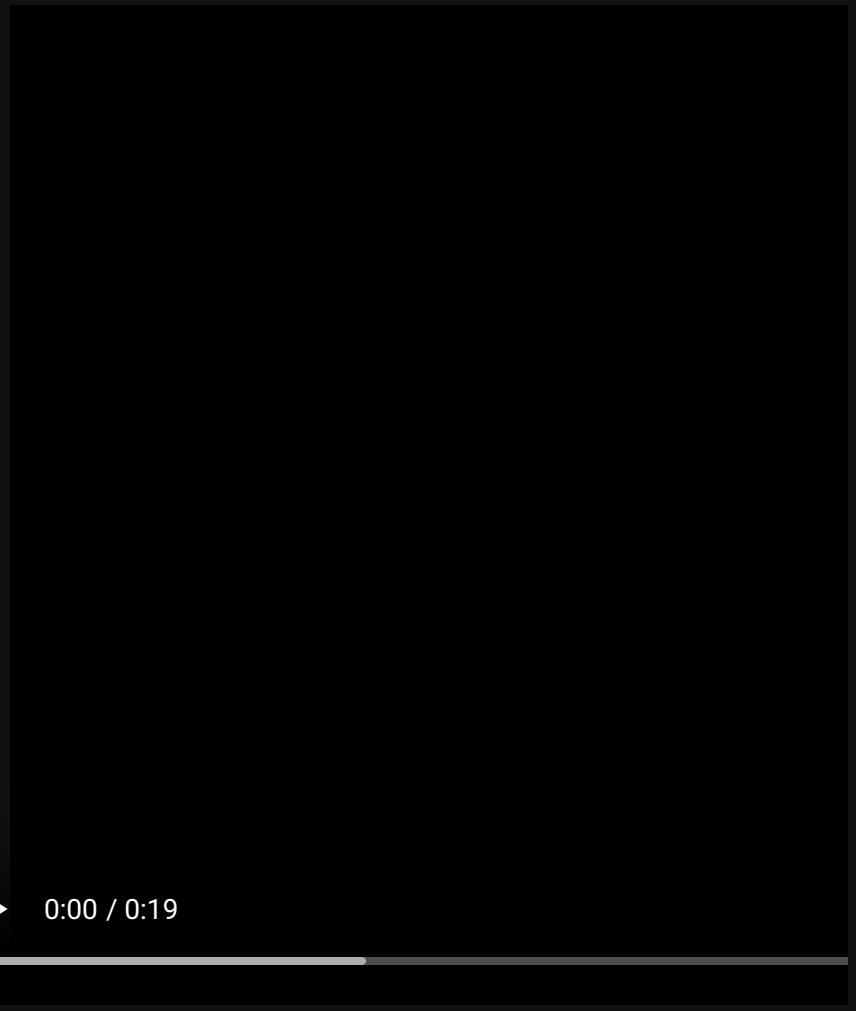
**Random connection model**

- Geometric kernel: $\varphi : \mathrm{R}_+ \to [0, 1]$.
- Connect two nodes at $X_u$ and $X_v$ with probability $\varphi\left(\frac{\|X_u - X_v\|}{\rho_n^{1/d}}\right)$, where $\rho_n = \frac{\log n}{n}$

- Examples:
  1. $\varphi(r) = \mathbf{1}\{r \leq 1\}$: random geometric graph
  2. General kernels

# Probabilistic broadcast with coded packets

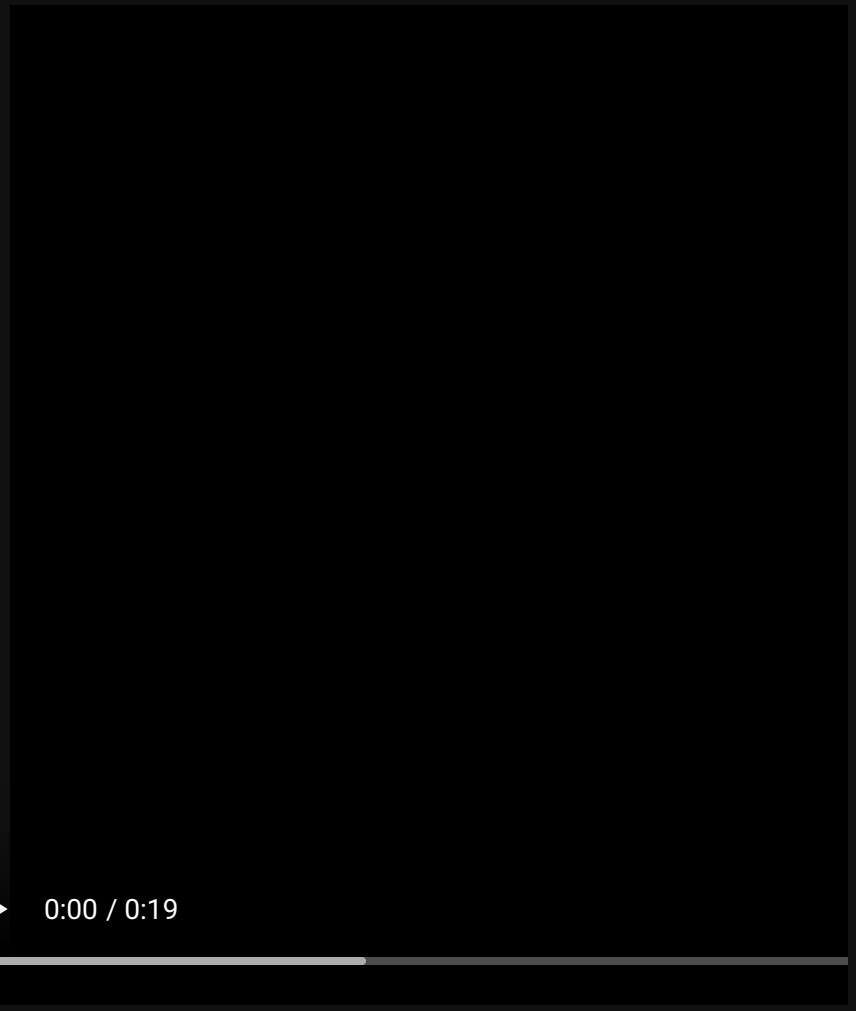
# Probabilistic broadcast with coded packets



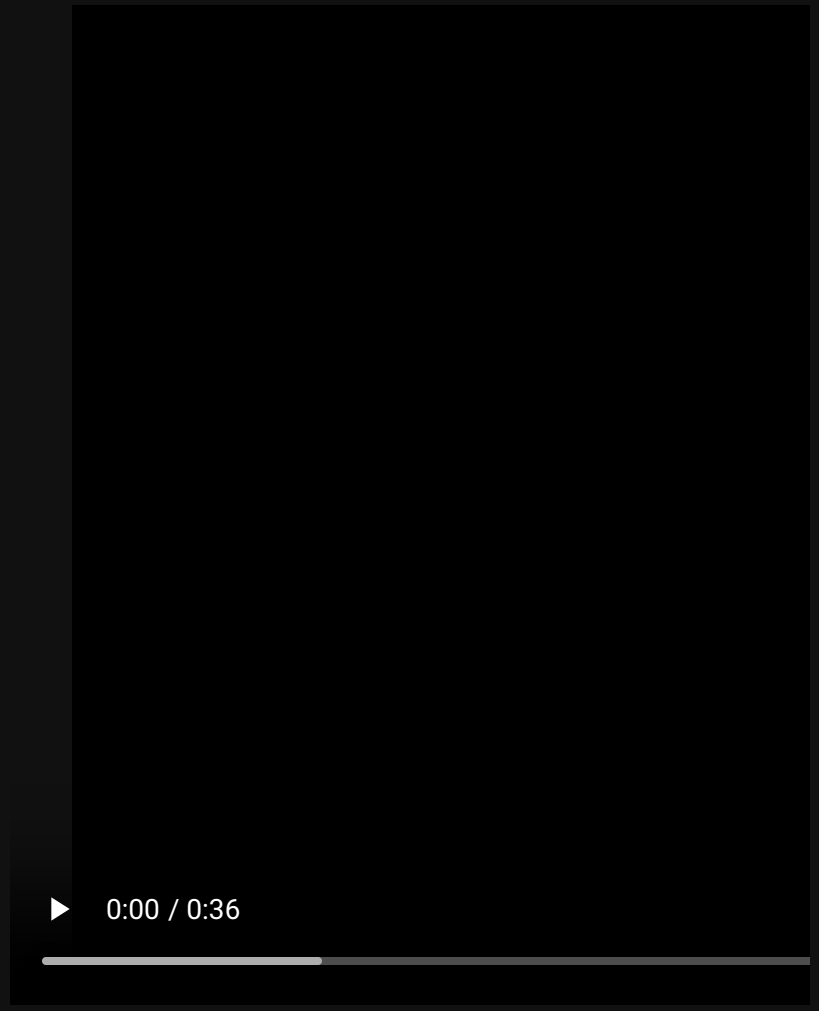

$k$ data

- Goal Broadcast the $k$ data packets from the source with minimum transmissions.

# Probabilistic broadcast with coded packets




$k$ data

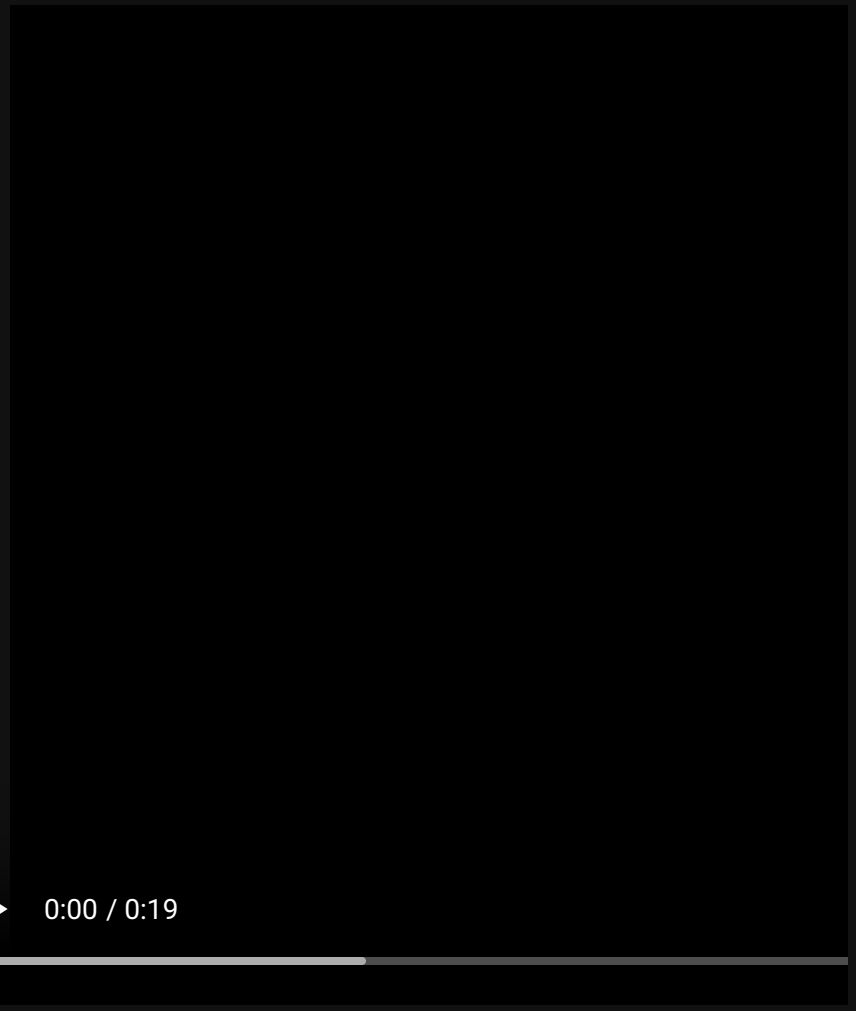



$n$ coded

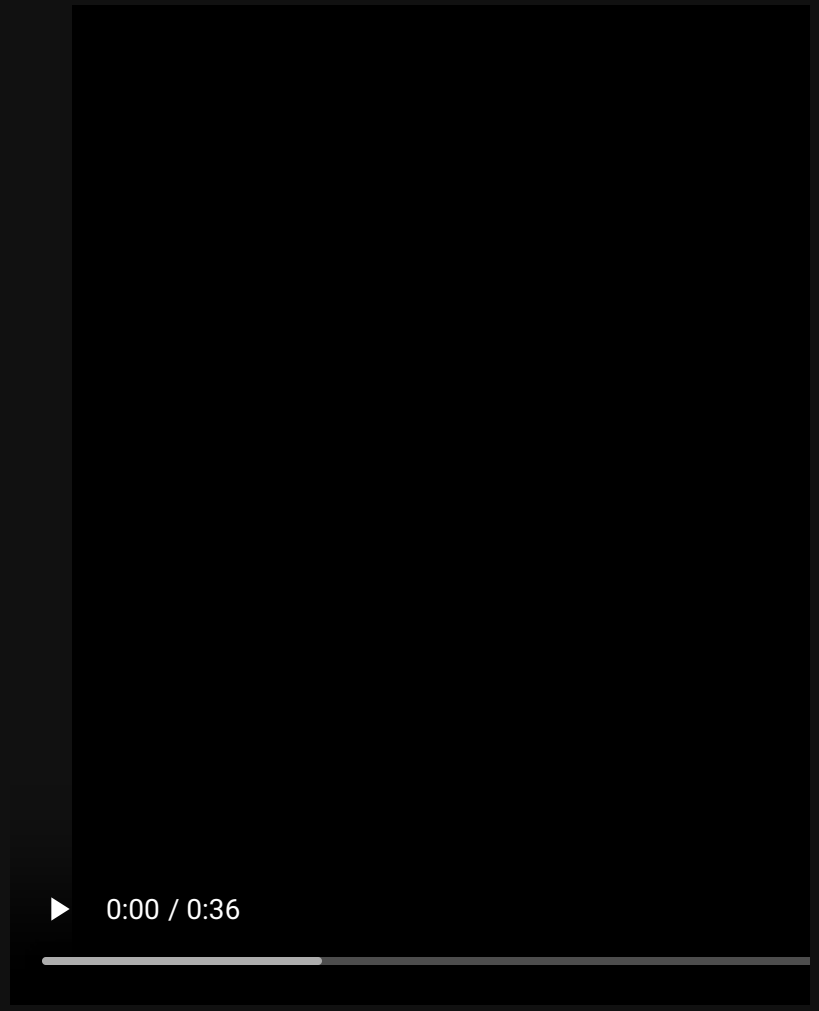
- Goal Broadcast the $k$ data packets from the source with minimum transmissions.
- Coding scheme: Any node receiving at least $k$ out of the $n$ coded packets is able to recover the $k$ data packets from the source.

packets

# Probabilistic broadcast with coded packets

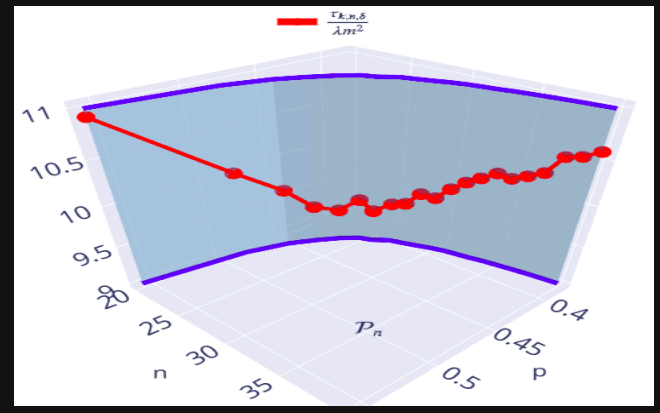



$k$ data




$n$ coded

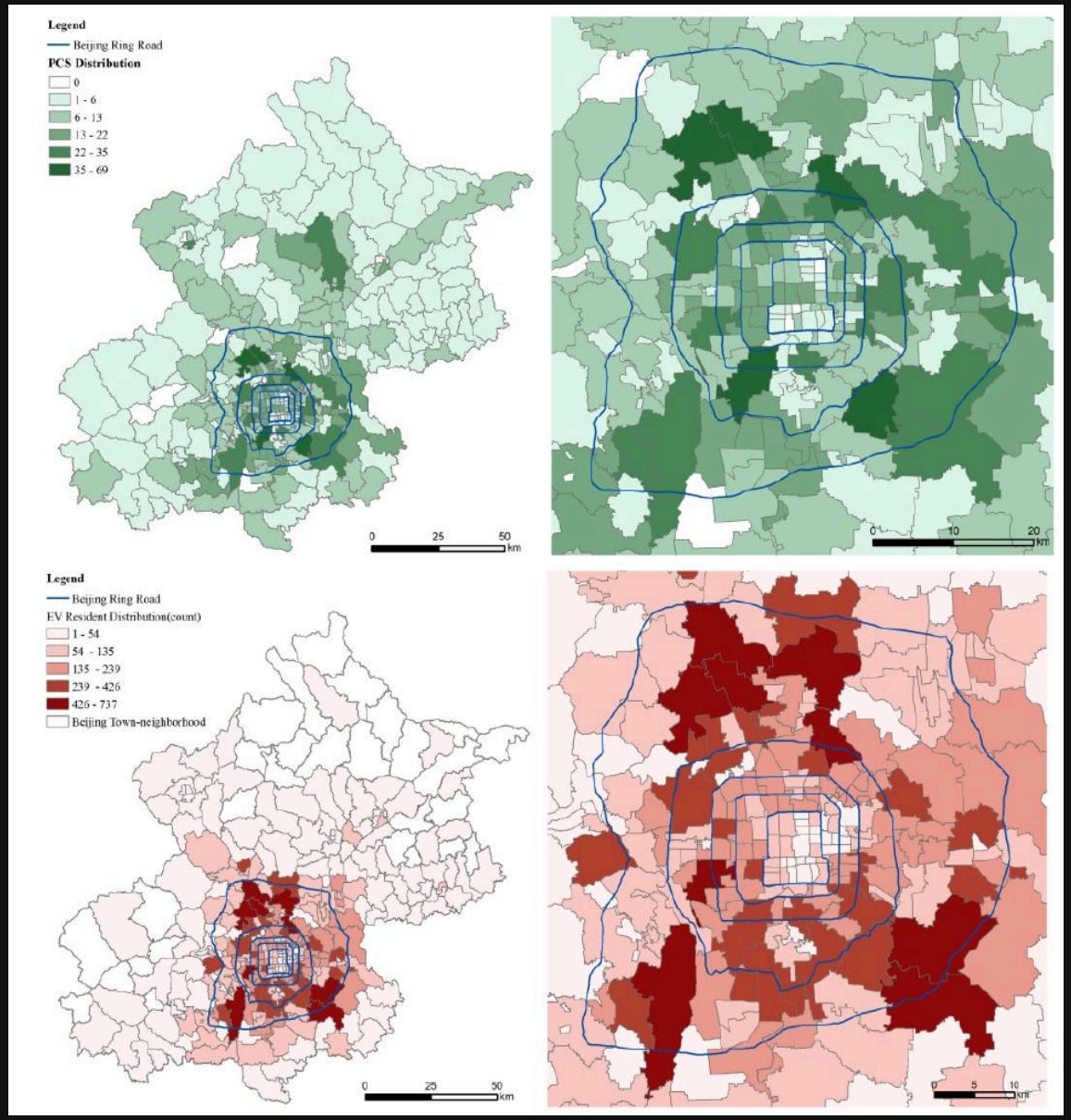
- Goal Broadcast the $k$ data packets from the source with minimum transmissions.
- Coding scheme: Any node receiving at least $k$ out of the $n$ coded packets is able to recover the $k$ data packets from the source.



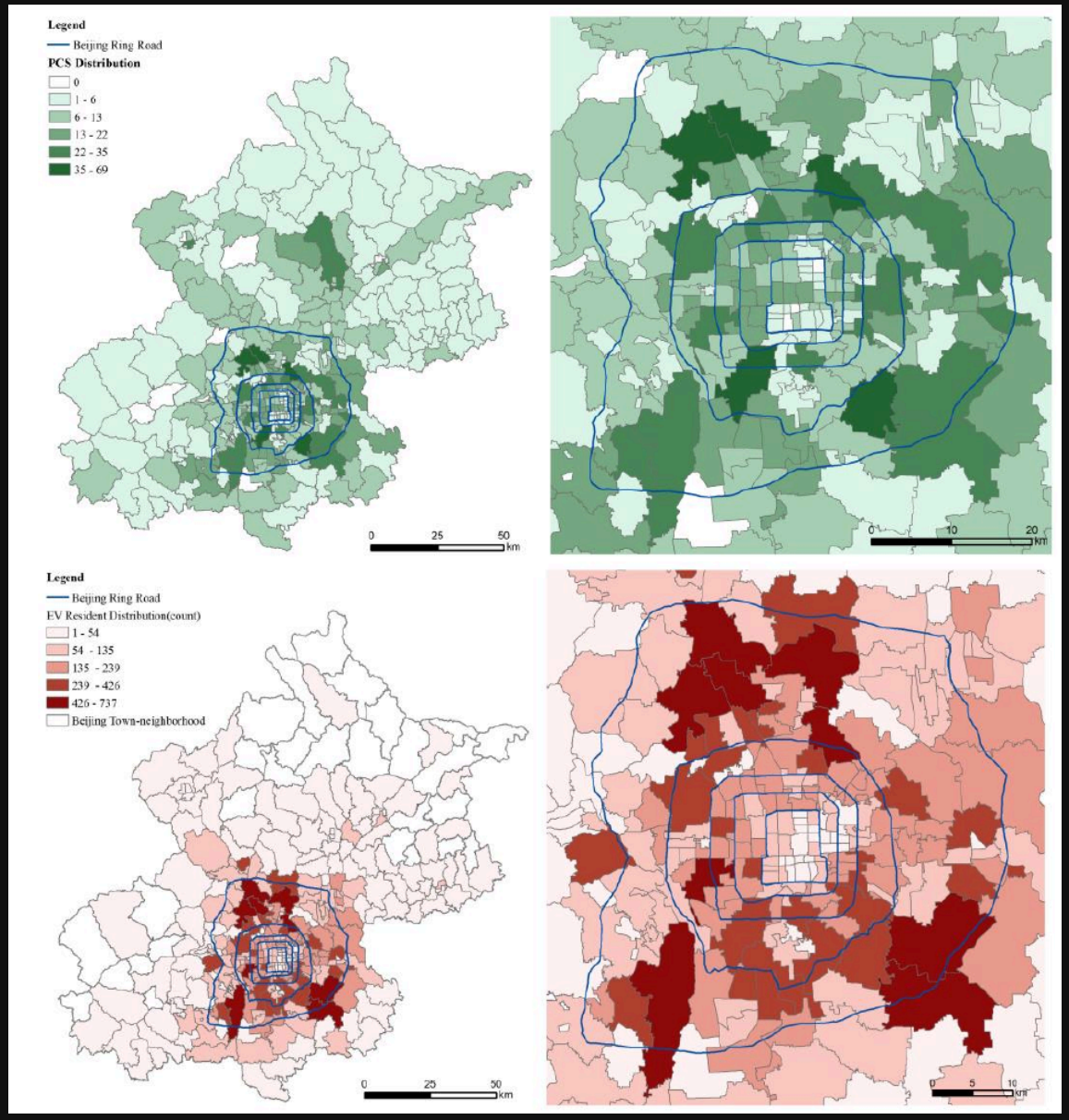
packets

# Spatial Queues

# Spatial Queues

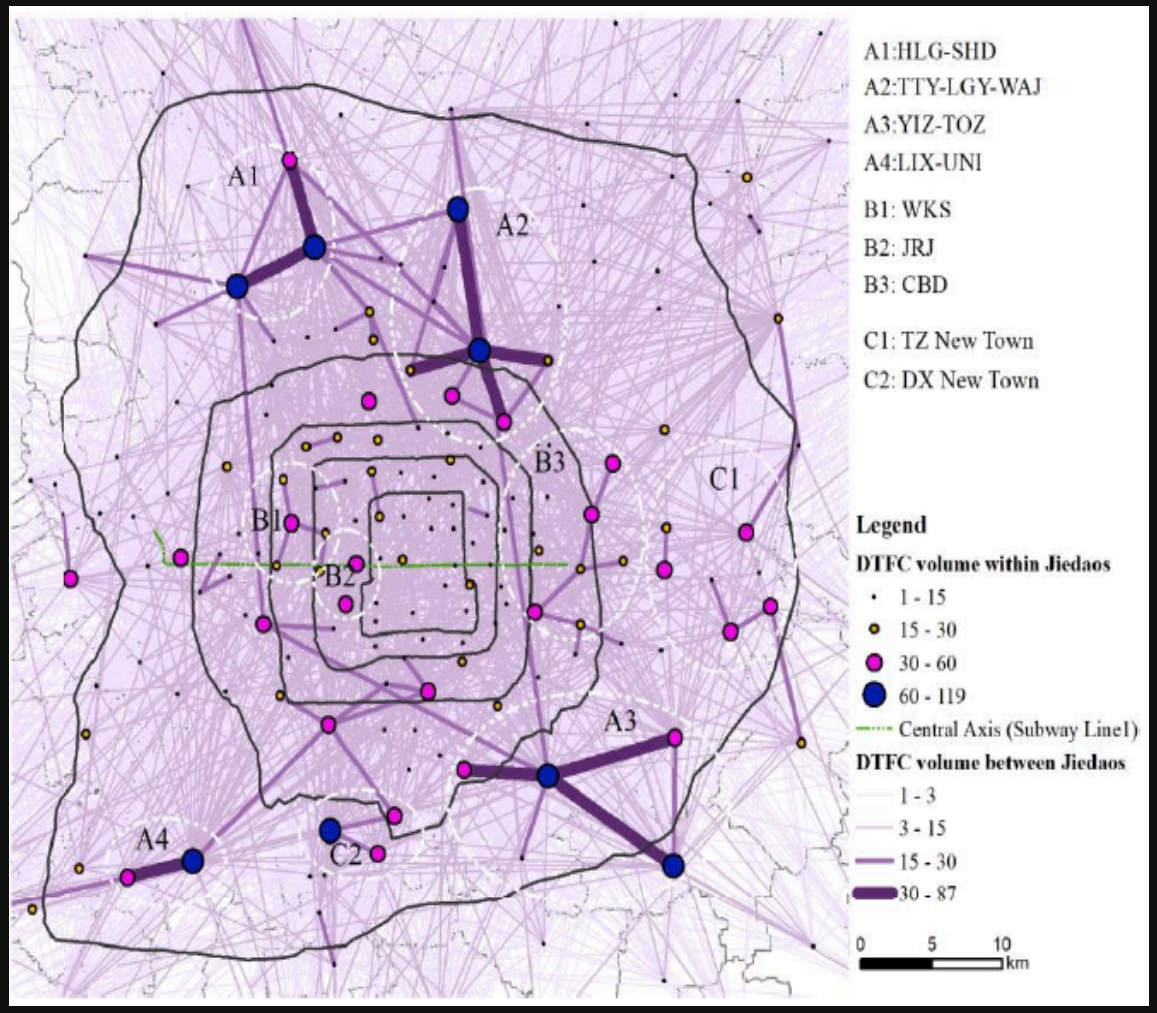


# Spatial Queues



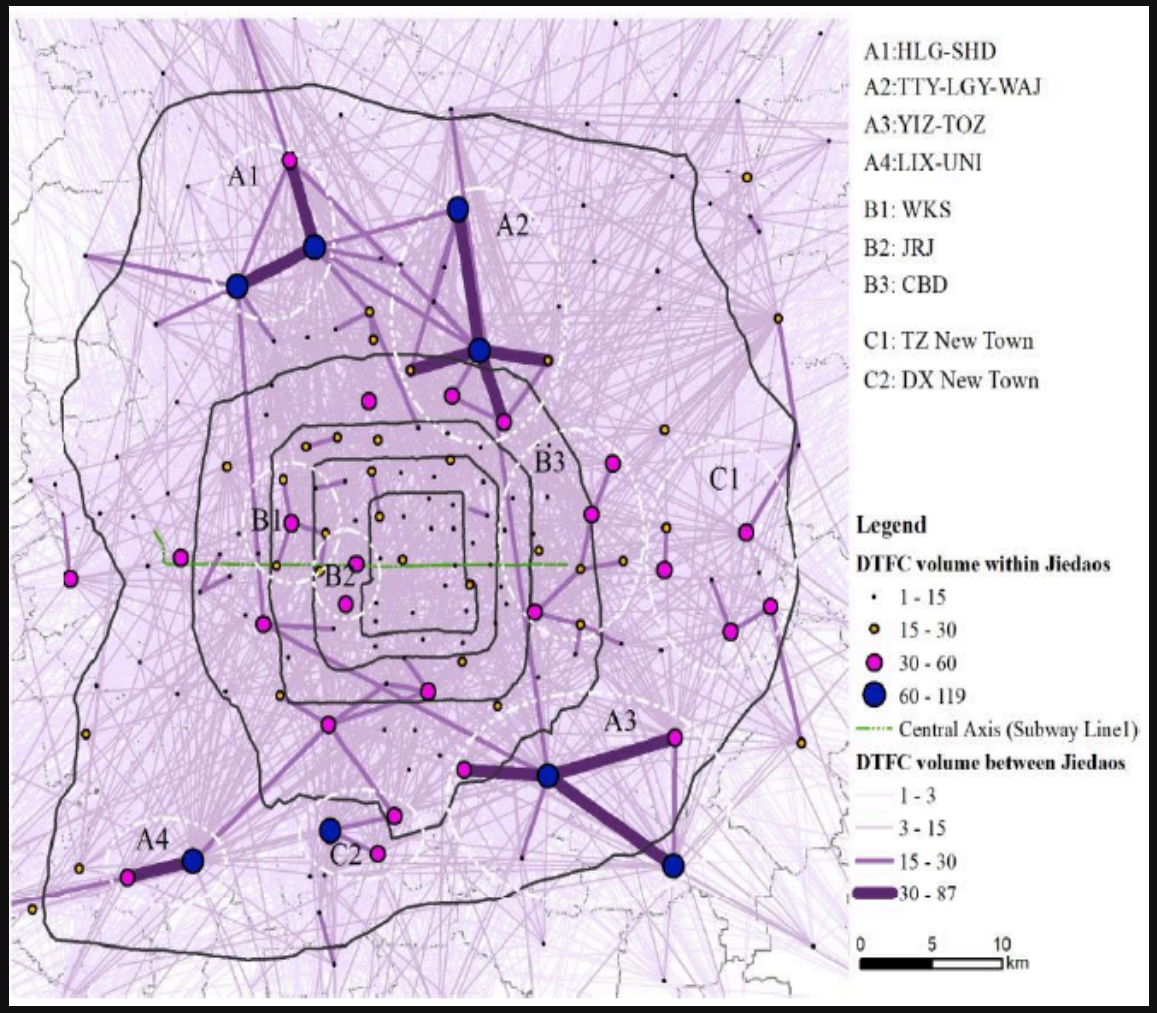

J. Kang, C. Kan, and Z. Lin, “Are Electric Vehicles Reshaping the City? An Investigation of the Clustering of Electric Vehicle Owners’ Dwellings and Their Interaction with Urban Spaces,” ISPRS International Journal of Geo-Information, vol. 10, no. 5, May 2021.

# Spatial Queues




J. Kang, C. Kan, and Z. Lin, "Are Electric Vehicles Reshaping the City? An Investigation of the Clustering of Electric Vehicle Owners' Dwellings and Their Interaction with Urban Spaces," ISPRS International Journal of Geo-Information, vol. 10, no. 5, May 2021.

# Spatial Queues

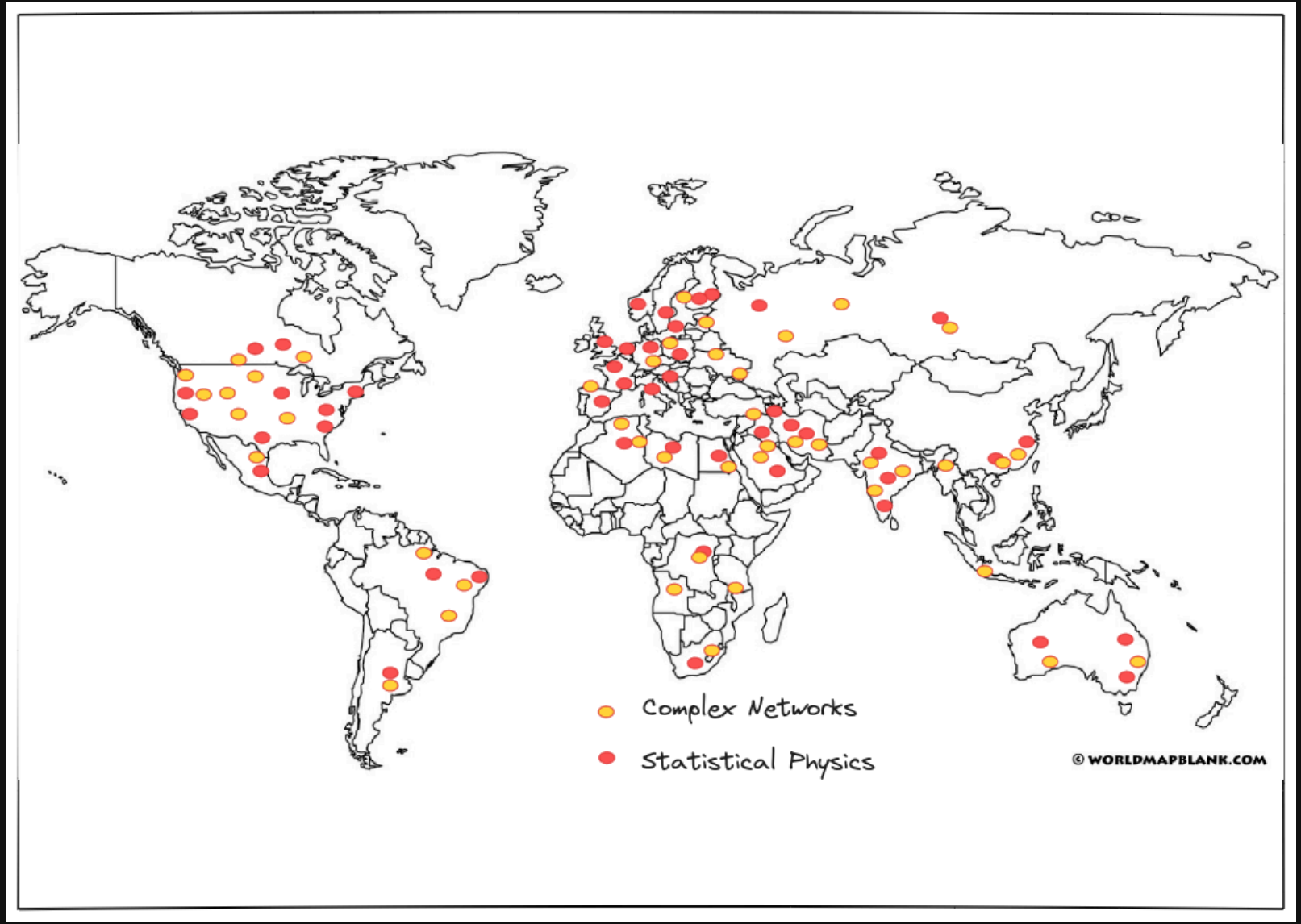



J. Kang, C. Kan, and Z. Lin, "Are Electric Vehicles Reshaping the City? An Investigation of the Clustering of Electric Vehicle Owners' Dwellings and Their Interaction with Urban Spaces," ISPRS International Journal of Geo-Information, vol. 10, no. 5, May 2021.

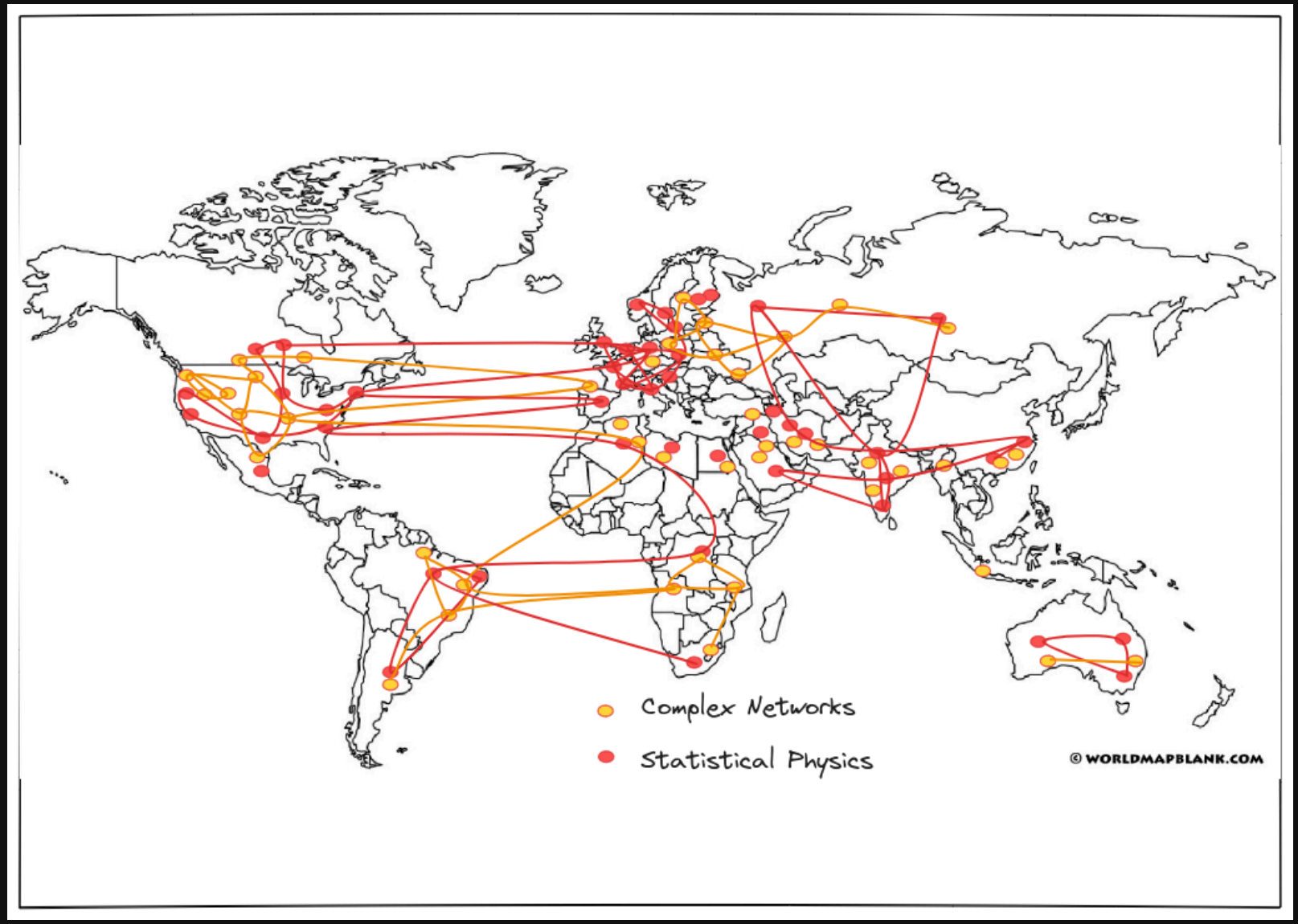
## Model

- Charging stations distributed as a PPP
- Customers follow nearest neighbour mobility strategies
- Characterize overloaded servers in the system
- Spatial distribution of overloaded servers?

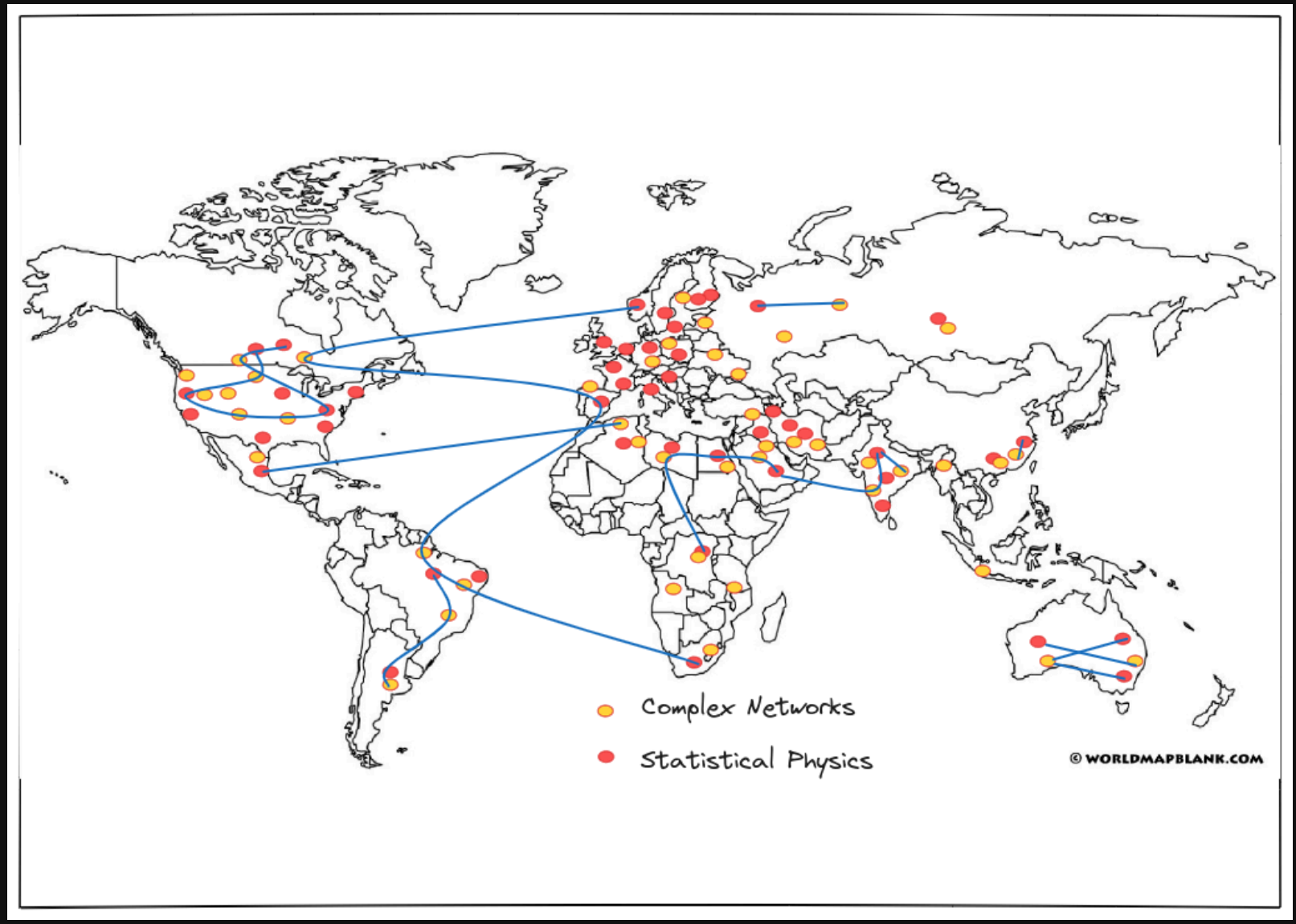
# Community detection



# Community detection



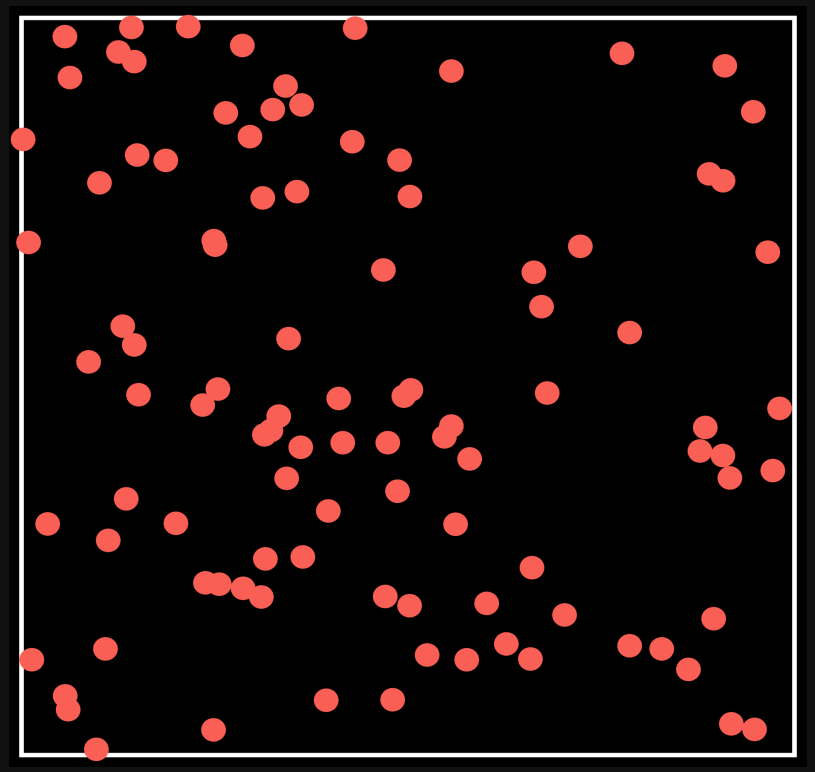
# Community detection



# Model

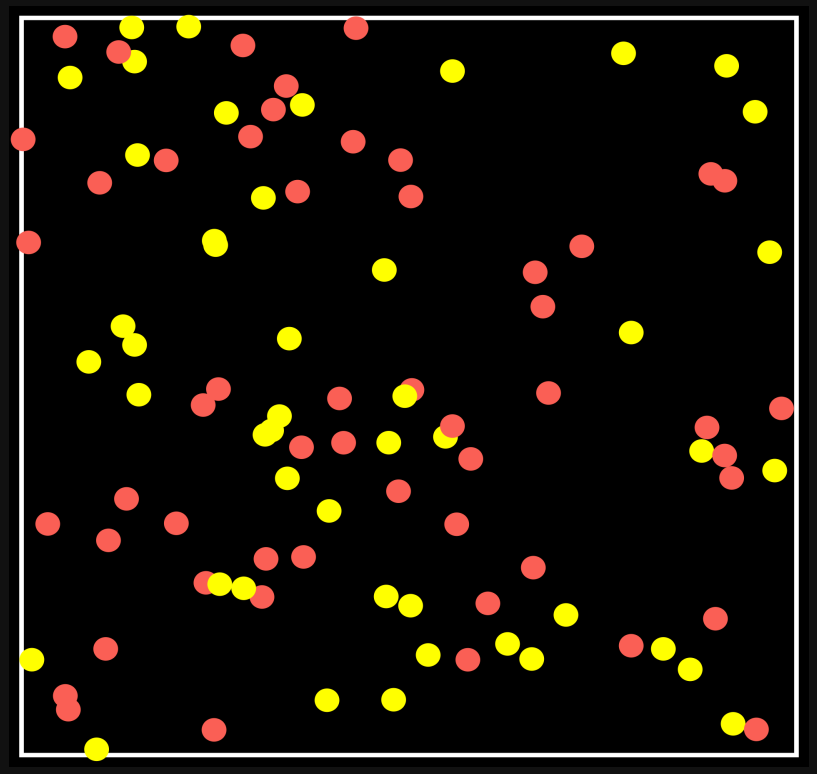
- Poisson point process $\mathbb{X} = \{X_u\}_{u=1}^{N}$ of intensity $\lambda n$

Torus $S = \left[\frac{-1}{2}, \frac{1}{2}\right]^d$



# Model

Torus $S = \left[\frac{-1}{2}, \frac{1}{2}\right]^d$

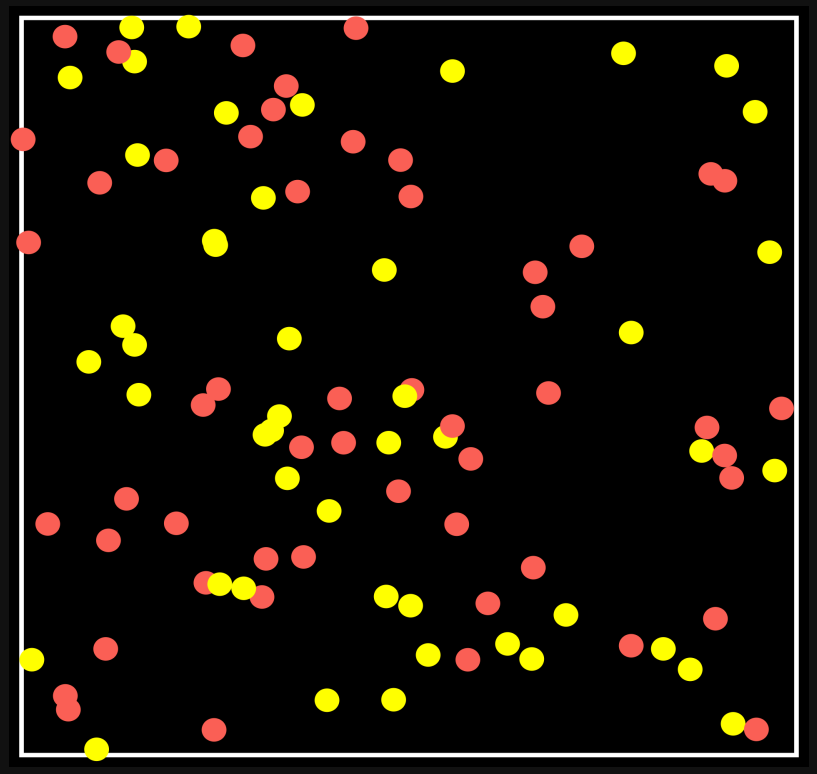


- Poisson point process $\mathrm{X} = \{X_u\}_{u=1}^{N}$ of intensity $\lambda n$
- Two communities: $\sigma = (\sigma(1), \cdots, \sigma(N))$

$$\mathbf{P}(\sigma(u) = +1) = \mathbf{P}(\sigma(u) = -1) = \frac{1}{2}$$

# Model

Torus $S = \left[\frac{-1}{2}, \frac{1}{2}\right]^d$



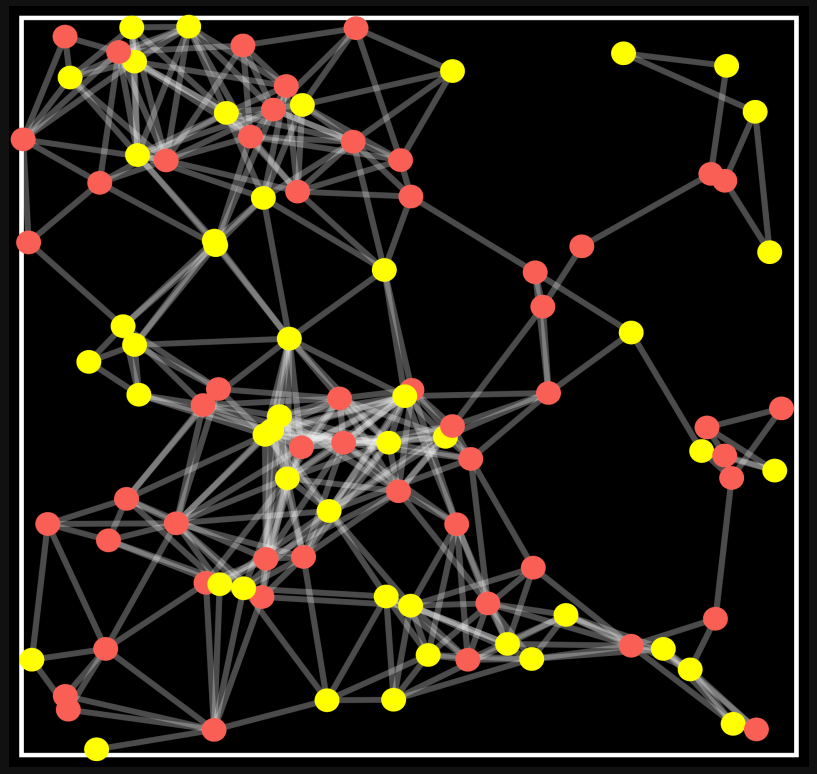
- Poisson point process $\mathrm{X} = \{X_u\}_{u=1}^{N}$ of intensity $\lambda n$
- Two communities: $\sigma = (\sigma(1), \cdots, \sigma(N))$

$$\mathbf{P}(\sigma(u) = +1) = \mathbf{P}(\sigma(u) = -1) = \frac{1}{2}$$

- Geometric kernel: $\varphi : \mathrm{R}_+ \to [0, 1]$
- Connection probabilities:
    1. Within community: $p$
    2. Between community: $q$

# Model

Torus $S = \left[\frac{-1}{2}, \frac{1}{2}\right]^d$



- Poisson point process $\mathrm{X} = \{X_u\}_{u=1}^{N}$ of intensity $\lambda n$
- Two communities: $\sigma = (\sigma(1), \cdots, \sigma(N))$

$$\mathbf{P}(\sigma(u) = +1) = \mathbf{P}(\sigma(u) = -1) = \frac{1}{2}$$

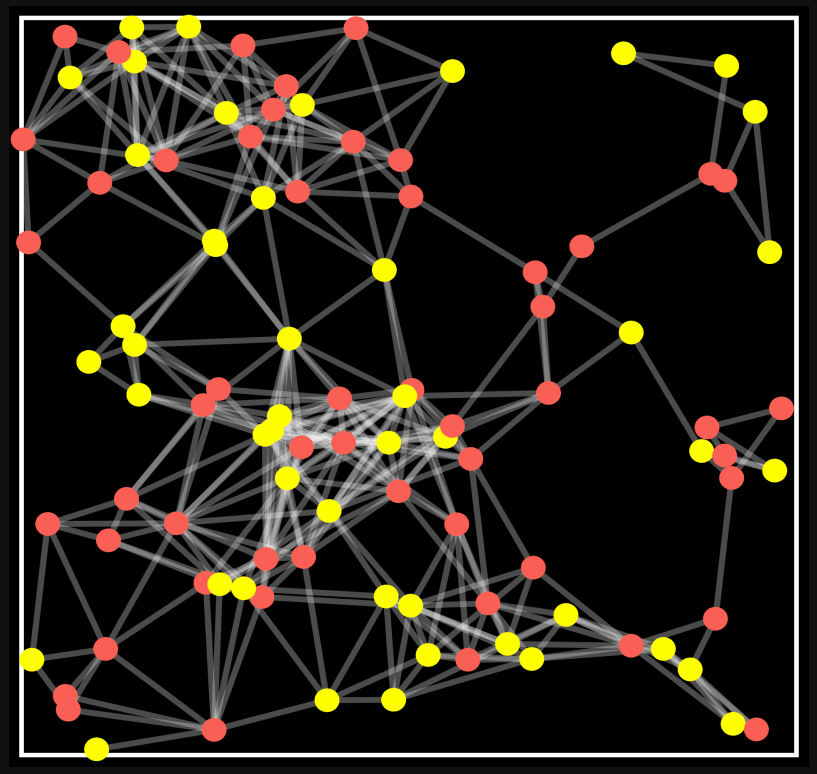
- Geometric kernel: $\varphi : \mathrm{R}_+ \rightarrow [0, 1]$
- Connection probabilities:
    1. Within community: $p$
    2. Between community: $q$

Given locations $\mathrm{X}$ and communities $\sigma$

$$A_{uv} = 1 \text{ w.p. } \begin{cases} p\varphi\left(\left(\frac{n}{\log n}\right)^{1/d}\|X_u - X_v\|\right) & \text{if } \sigma(u) = \sigma(v) \\ q\varphi\left(\left(\frac{n}{\log n}\right)^{1/d}\|X_u - X_v\|\right) & \text{if } \sigma(u) \neq \sigma(v) \end{cases}$$

# Model

Torus $S = \left[\frac{-1}{2}, \frac{1}{2}\right]^d$



$\mathbf{A} \sim GKBM(\lambda, n, d, p, q, \varphi)$

- Poisson point process $\mathrm{X} = \{X_u\}_{u=1}^{N}$ of intensity $\lambda n$
- Two communities: $\sigma = (\sigma(1), \cdots, \sigma(N))$

$$\mathbf{P}(\sigma(u) = +1) = \mathbf{P}(\sigma(u) = -1) = \frac{1}{2}$$

- Geometric kernel: $\varphi : \mathrm{R}_+ \rightarrow [0, 1]$
- Connection probabilities:
  1. Within community: $p$
  2. Between community: $q$

Given locations $\mathrm{X}$ and communities $\sigma$

$$A_{uv} = 1 \text{ w.p. } \begin{cases} p\varphi\left(\left(\frac{n}{\log n}\right)^{1/d}\|X_u - X_v\|\right) & \text{if } \sigma(u) = \sigma(v) \\ q\varphi\left(\left(\frac{n}{\log n}\right)^{1/d}\|X_u - X_v\|\right) & \text{if } \sigma(u) \neq \sigma(v) \end{cases}$$

# Problem Formulation

# Problem Formulation

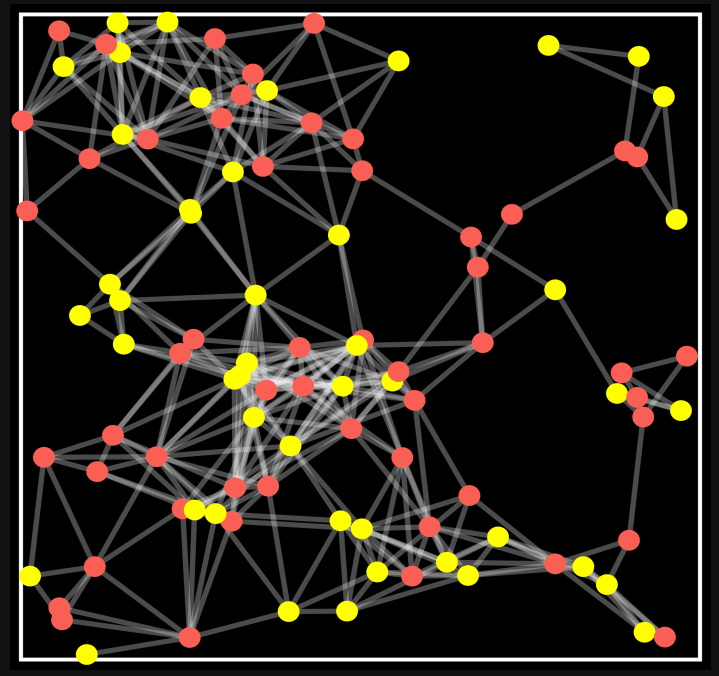
$$\mathbf{A} \sim GKBM(\lambda, n, d, p, q, \varphi)$$

Problem: Given the locations $\mathbf{X}$ and the graph $\mathbf{A}$, recover $\sigma_n$ exactly.

# Problem Formulation

$$\mathbf{A} \sim GKBM(\lambda, n, d, p, q, \varphi)$$

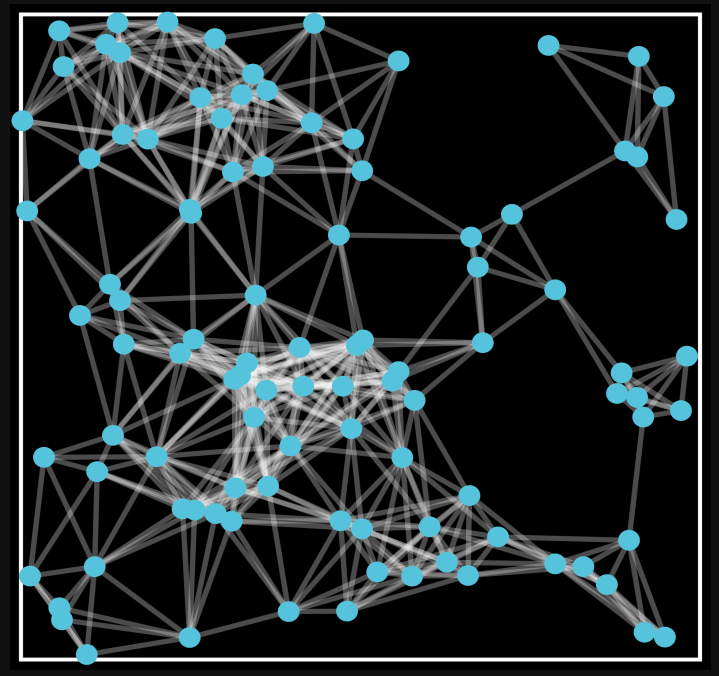
Problem: Given the locations $\mathbf{X}$ and the graph $\mathbf{A}$, recover $\sigma_n$ exactly.



# Problem Formulation

$$\mathbf{A} \sim GKBM(\lambda, n, d, p, q, \varphi)$$

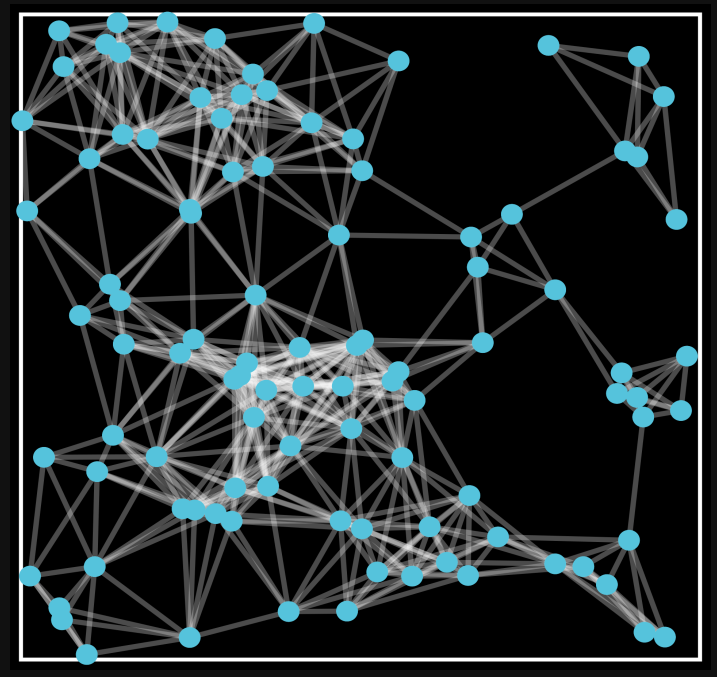
Problem: Given the locations $\mathbf{X}$ and the graph $\mathbf{A}$, recover $\sigma_n$ exactly.



# Problem Formulation

$$\mathbf{A} \sim GKBM(\lambda, n, d, p, q, \varphi)$$

Problem: Given the locations $\mathbf{X}$ and the graph $\mathbf{A}$, recover $\sigma_n$ exactly.



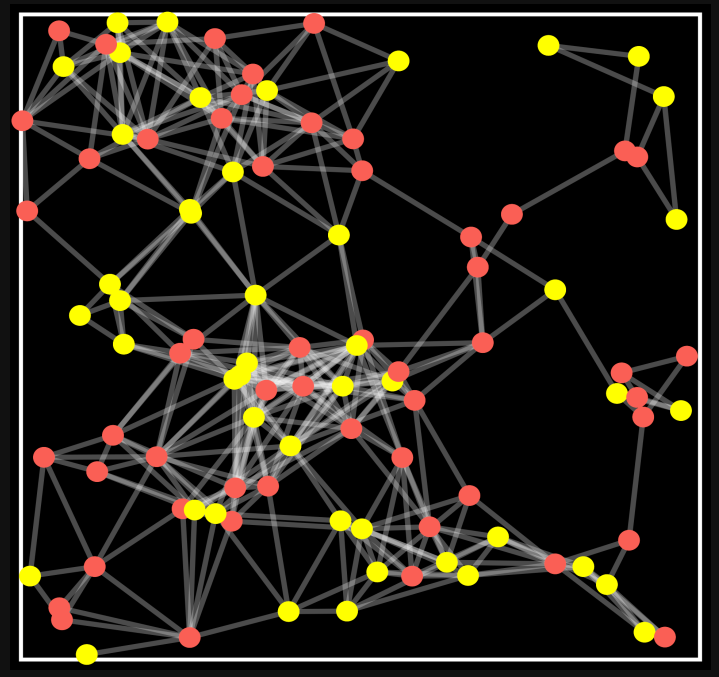
An estimate $\hat{\sigma}_n$ of $\sigma_n$ is said to recover the communities exactly if

$$\lim_{n \to \infty} \mathrm{P}\left(\hat{\sigma}_n \in \{\pm \sigma_n\}\right) = 1$$

# Problem Formulation

$$\mathbf{A} \sim GKBM(\lambda, n, d, p, q, \varphi)$$

Problem: Given the locations $\mathbf{X}$ and the graph $\mathbf{A}$, recover $\sigma_n$ exactly.



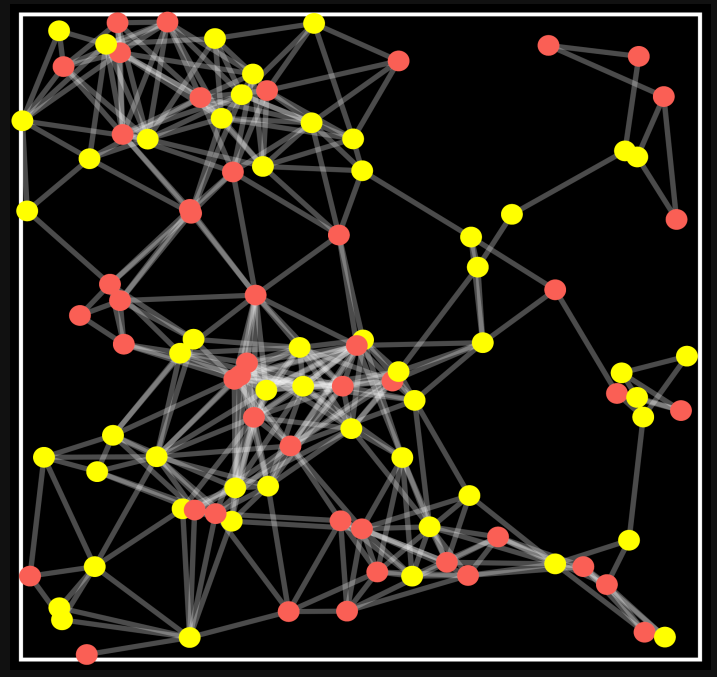
An estimate $\hat{\sigma}_n$ of $\sigma_n$ is said to recover the communities exactly if

$$\lim_{n \to \infty} \mathrm{P}\left(\hat{\sigma}_n \in \{\pm \sigma_n\}\right) = 1$$

# Problem Formulation

$$\mathbf{A} \sim GKBM(\lambda, n, d, p, q, \varphi)$$

Problem: Given the locations $\mathbf{X}$ and the graph $\mathbf{A}$, recover $\sigma_n$ exactly.



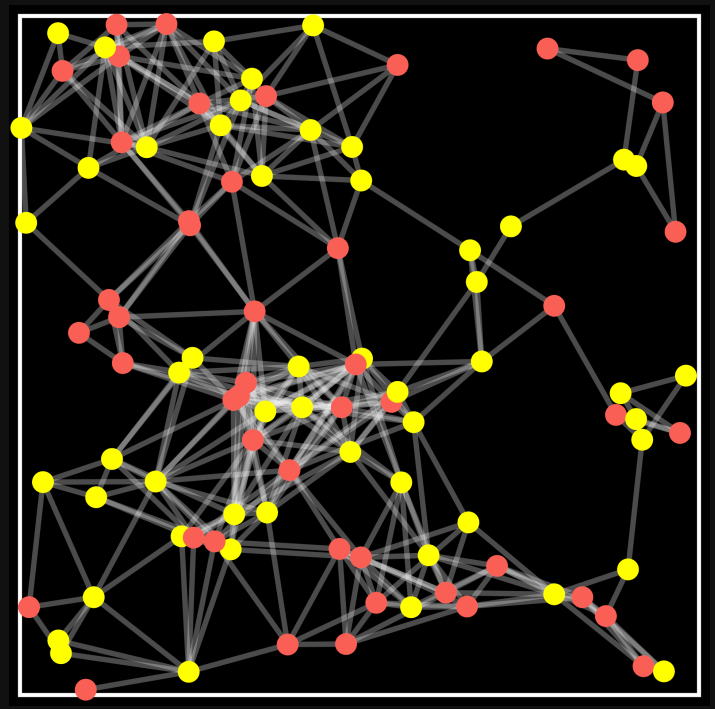
An estimate $\hat{\sigma}_n$ of $\sigma_n$ is said to recover the communities exactly if

$$\lim_{n \to \infty} \mathrm{P}\left(\hat{\sigma}_n \in \{\pm \sigma_n\}\right) = 1$$

# Problem Formulation

$$\mathbf{A} \sim GKBM(\lambda, n, d, p, q, \varphi)$$

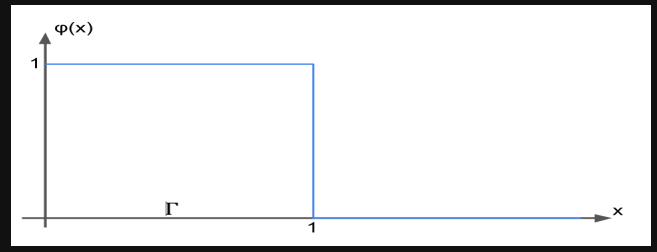
Problem: Given the locations $\mathbb{X}$ and the graph $\mathbf{A}$, recover $\sigma_n$ exactly.



An estimate $\hat{\sigma}_n$ of $\sigma_n$ is said to recover the communities exactly if

$$\lim_{n\to\infty} \mathbb{P}\big(\hat{\sigma}_n \in \{\pm\sigma_n\}\big) = 1$$

**Prior work**





1. Abbe, E., Baccelli, F., and Sankararaman, A. (2021). Community detection on Euclidean random graphs. Information and Inference: A Journal of the IMA, 10(1), 109-160.
2. Gaudio, J., Niu, X. and Wei, E., 2024. Exact community recovery in the geometric SBM. In Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2158-2184.
3. Gaudio, J., Guan, C., Niu, X. and Wei, E., 2024. Exact Label Recovery in Euclidean Random Graphs. arXiv preprint arXiv:2407.11163.

# Model-1d

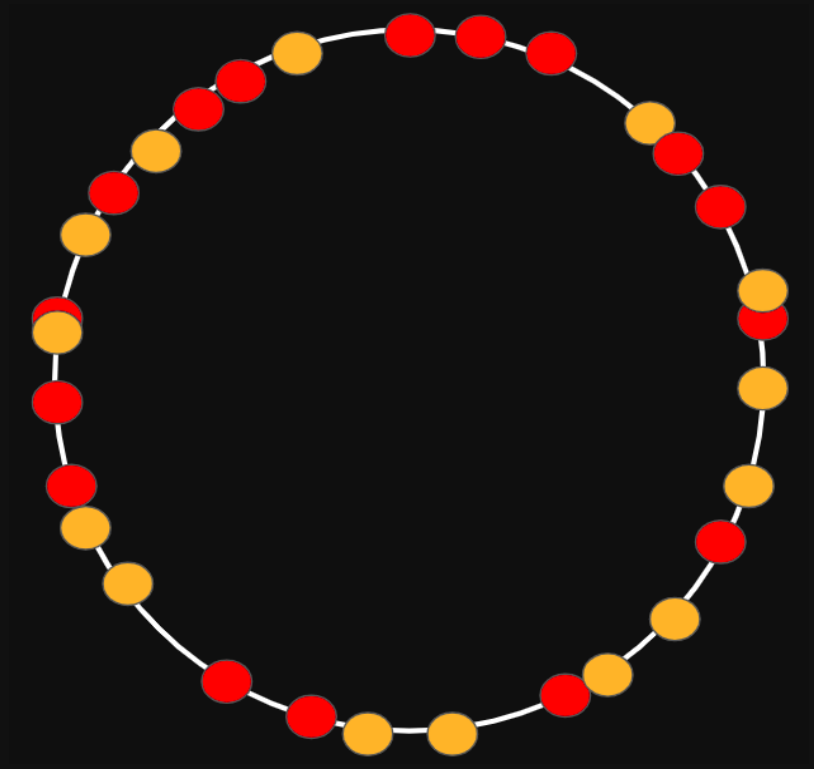
Torus $S = \left[\frac{-1}{2}, \frac{1}{2}\right]$


- Poisson point process $\mathbb{X} = \{X_u\}_{u=1}^{N}$ of intensity $\lambda n$
- Two communities: $\sigma = \big(\sigma(1), \cdots, \sigma(N)\big)$

$$\mathbf{P}(\sigma(u) = +1) = \mathbf{P}(\sigma(u) = -1) = \frac{1}{2}$$

# Model-1d

Torus $S = \left[\frac{-1}{2}, \frac{1}{2}\right]$

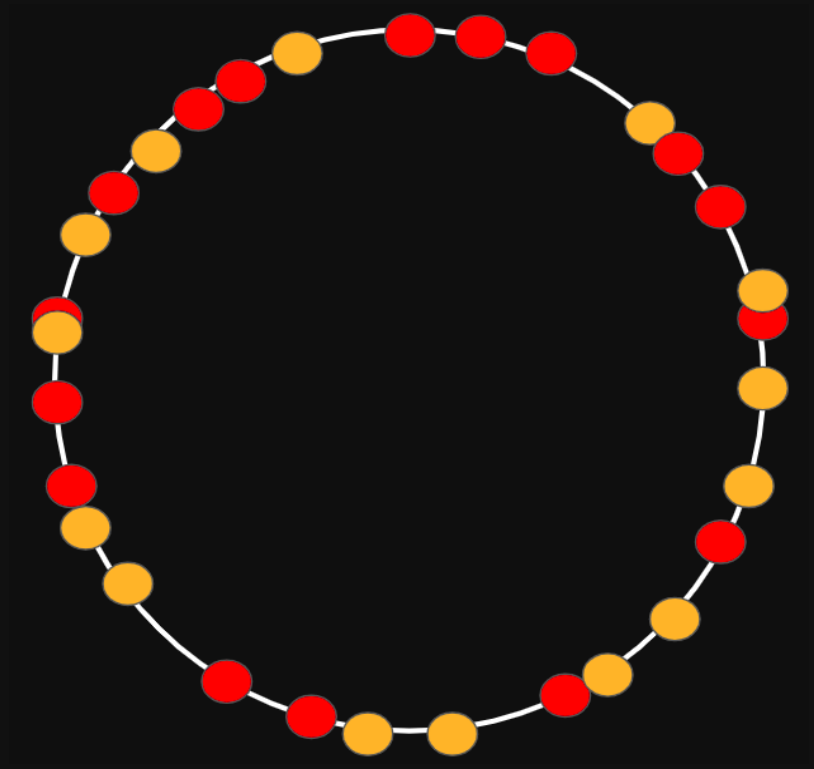


- Poisson point process $\mathrm{X} = \{X_u\}_{u=1}^{N}$ of intensity $\lambda n$
- Two communities: $\sigma = (\sigma(1), \cdots, \sigma(N))$

$$\mathbf{P}(\sigma(u) = +1) = \mathbf{P}(\sigma(u) = -1) = \frac{1}{2}$$

# Model-1d

Torus $S = \left[\frac{-1}{2}, \frac{1}{2}\right]$



$\mathbf{A} \sim GKBM(\lambda, n, p, q, \varphi)$

- Poisson point process $\mathrm{X} = \{X_u\}_{u=1}^{N}$ of intensity $\lambda n$
- Two communities: $\sigma = (\sigma(1), \cdots, \sigma(N))$

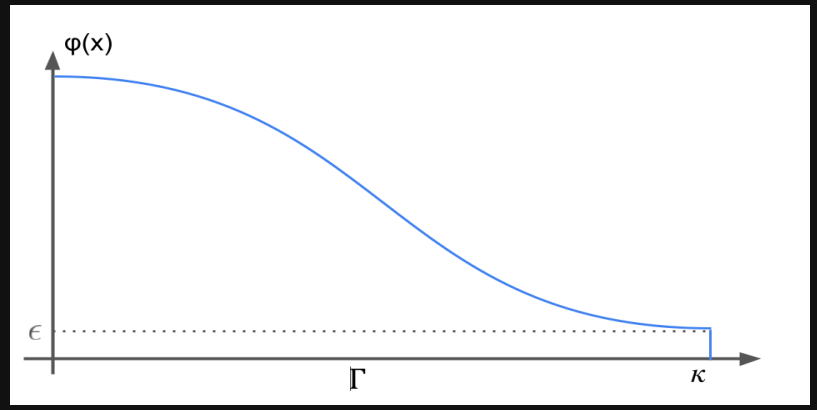
$$\mathbf{P}(\sigma(u) = +1) = \mathbf{P}(\sigma(u) = -1) = \frac{1}{2}$$

- Geometric kernel: $\varphi : \mathrm{R}_+ \rightarrow [0, 1]$
- Connection probabilities:
  1. Within community: $p$
  2. Between community: $q$

Given locations $\mathrm{X}$ and communities $\sigma$

$$A_{uv} = 1 \text{ w.p. } \begin{cases} p\varphi\left(\frac{n}{\log n}\|X_u - X_v\|\right) & \text{if } \sigma(u) = \sigma(v) \\ q\varphi\left(\frac{n}{\log n}\|X_u - X_v\|\right) & \text{if } \sigma(u) \neq \sigma(v) \end{cases}$$

# Main results




Define $\kappa = \sup_{x \in \Gamma} x, \;\; 0 < \kappa < \infty$ and

$$I_\varphi(p,q) := 2 \int_{\mathbb{R}_+} \left[1 - \sqrt{pq}\varphi(x) - \sqrt{(1-p\varphi(x))(1-q\varphi(x))}\right] dx$$

Converse: If $\lambda\kappa < 1$ or $\lambda I_\varphi(p,q) < 1$, then exact recovery is not possible using any algorithm.

Achievability: If $\lambda\kappa > 1$ and $\lambda I_\varphi(p,q) > 1$, then there exists a linear time algorithm (in the number of edges) achieving exact recovery.

# Impossibility: Idea

If $\lambda\kappa < 1$ or $\lambda I_\varphi(p,q) < 1$, then exact recovery is not possible.

$$I_\varphi(p,q) := 2\int_{\mathbb{R}_+} \left[1 - \sqrt{pq}\varphi(x) - \sqrt{(1-p\varphi(x))(1-q\varphi(x))}\right] dx$$

# Impossibility: Idea

If $\lambda\kappa < 1$ or $\lambda I_\varphi(p, q) < 1$, then exact recovery is not possible.

$$I_\varphi(p, q) := 2\int_{\mathrm{R}_+} \left[1 - \sqrt{pq}\varphi(x) - \sqrt{(1 - p\varphi(x))(1 - q\varphi(x))}\right] dx$$

- Genie-based estimator: Log-likelihood

# Impossibility: Idea

If $\lambda\kappa < 1$ or $\lambda I_\varphi(p, q) < 1$, then exact recovery is not possible.

$$I_\varphi(p, q) := 2\int_{\mathrm{R}_+} \left[1 - \sqrt{pq}\varphi(x) - \sqrt{(1 - p\varphi(x))(1 - q\varphi(x))}\right] dx$$

- Genie-based estimator: Log-likelihood

$$\sum_{\substack{A_{0\nu}=1 \\ \sigma_\nu = \sigma_0}} \log(p\varphi_{\nu 0}) + \sum_{\substack{A_{0\nu}=1 \\ \sigma_\nu \neq \sigma_0}} \log(q\varphi_{\nu 0}) + \sum_{\substack{A_{0\nu}=0 \\ \sigma_\nu = \sigma_0}} \log(1 - p\varphi_{\nu 0}) + \sum_{\substack{A_{0\nu}=0 \\ \sigma_\nu \neq \sigma_0}} \log(1 - q\varphi_{\nu 0})$$

# Impossibility: Idea

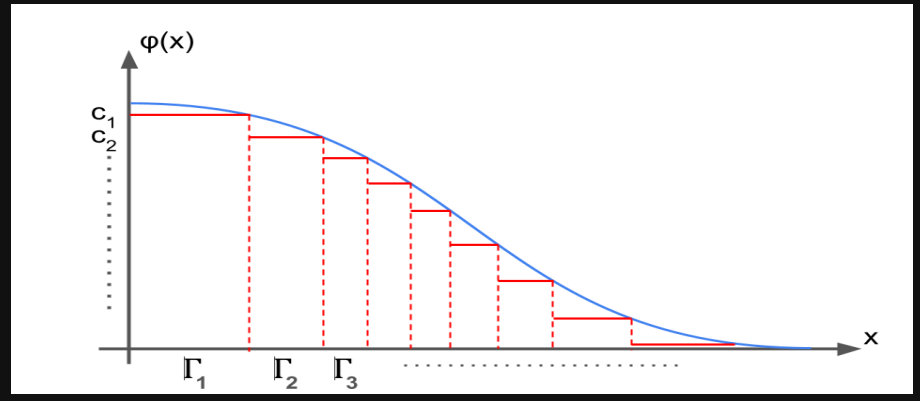
If $\lambda\kappa < 1$ or $\lambda I_\varphi(p, q) < 1$, then exact recovery is not possible.

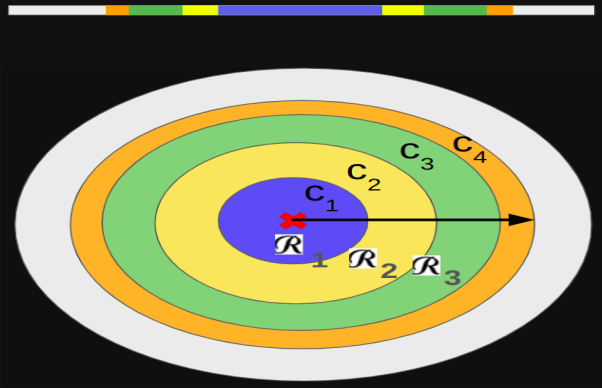
$$I_\varphi(p, q) := 2\int_{\mathrm{R}_+} \left[1 - \sqrt{pq}\varphi(x) - \sqrt{(1 - p\varphi(x))(1 - q\varphi(x))}\right]dx$$

- Genie-based estimator: Log-likelihood

$$\sum_{\substack{A_{0v}=1 \\ \sigma_v=\sigma_0}} \log(p\varphi_{v0}) + \sum_{\substack{A_{0v}=1 \\ \sigma_v\neq\sigma_0}} \log(q\varphi_{v0}) + \sum_{\substack{A_{0v}=0 \\ \sigma_v=\sigma_0}} \log(1 - p\varphi_{v0}) + \sum_{\substack{A_{0v}=0 \\ \sigma_v\neq\sigma_0}} \log(1 - q\varphi_{v0})$$

Approximate by simple functions



# Impossibility: Idea

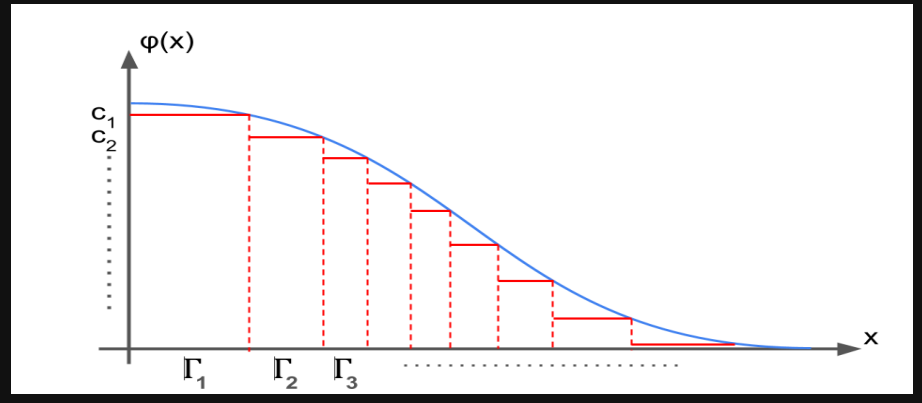
If $\lambda\kappa < 1$ or $\lambda I_\varphi(p, q) < 1$, then exact recovery is not possible.

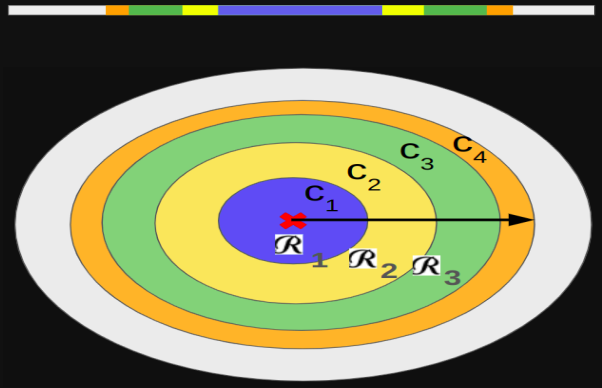
$$I_\varphi(p, q) := 2\int_{\mathrm{R}_+} \left[1 - \sqrt{pq}\varphi(x) - \sqrt{(1 - p\varphi(x))(1 - q\varphi(x))}\right] dx$$

- Genie-based estimator: Log-likelihood

$$\sum_{s=1}^{\ell} \sum_{v \in \mathcal{R}_s} \sum_{\substack{A_{0v}=1 \\ \sigma_v = \sigma_0}} \log(pc_s) + \sum_{\substack{A_{0v}=1 \\ \sigma_v \neq \sigma_0}} \log(qc_s) + \sum_{\substack{A_{0v}=0 \\ \sigma_v = \sigma_0}} \log(1 - pc_s) + \sum_{\substack{A_{0v}=0 \\ \sigma_v \neq \sigma_0}} \log(1 - qc_s)$$

Approximate by simple functions



# Impossibility: Idea

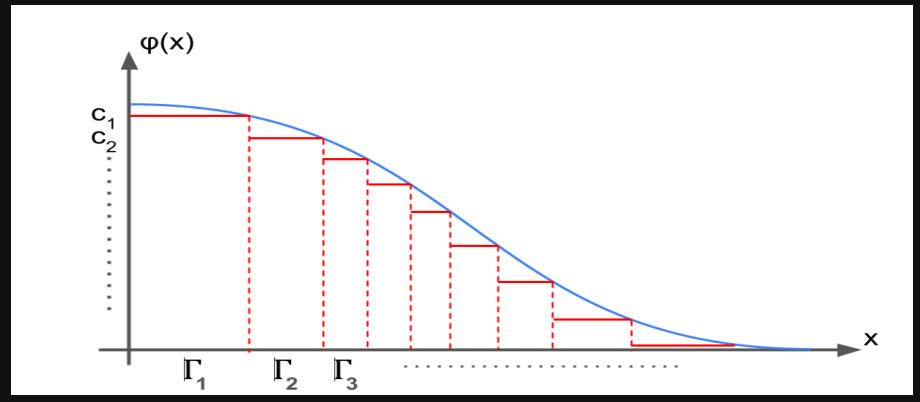
If $\lambda\kappa < 1$ or $\lambda I_{\varphi}(p, q) < 1$, then exact recovery is not possible.

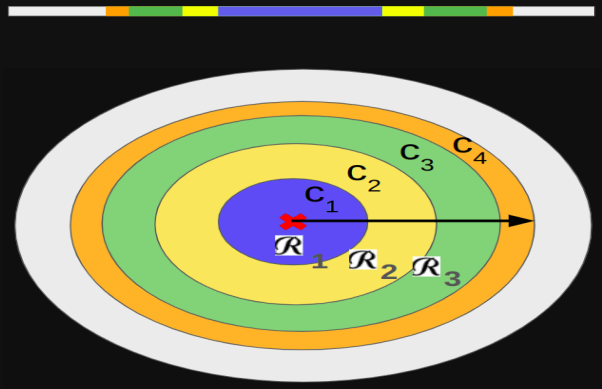
$$I_{\varphi}(p, q) := 2\int_{\mathrm{R}_+} \left[1 - \sqrt{pq}\varphi(x) - \sqrt{(1 - p\varphi(x))(1 - q\varphi(x))}\right] dx$$

- Genie-based estimator: Log-likelihood

$$\sum_{s=1}^{\ell} P_s^{+} \log(pc_s) + Q_s^{+} \log(qc_s) + P_s^{-} \log(1 - pc_s) + Q_s^{-} \log(1 - qc_s)$$

Approximate by simple functions



# Impossibility: Idea

If $\lambda\kappa < 1$ or $\lambda I_\varphi(p, q) < 1$, then exact recovery is not possible.

$$I_\varphi(p, q) := 2\int_{\mathrm{R}_+} \left[1 - \sqrt{pq}\varphi(x) - \sqrt{(1 - p\varphi(x))(1 - q\varphi(x))}\right] dx$$

- Genie-based estimator: Log-likelihood

$$\sum_{s=1}^{\ell} P_s^+ \log(pc_s) + Q_s^+ \log(qc_s) + P_s^- \log(1 - pc_s) + Q_s^- \log(1 - qc_s)$$

| In $\mathcal{R}_s$ | Neighbours | Non-neighbours |
|---|---|---|
| Same | $P_s^+ \sim \mathrm{Poi}\left(\lambda \mathrm{vol}(\Gamma_s)\log(n)pc_s\right)$ | $P_s^- \sim \mathrm{Poi}\left(\lambda \mathrm{vol}(\Gamma_s)\log(n)(1 - pc_s)\right)$ |
| Different | $Q_s^+ \sim \mathrm{Poi}\left(\lambda \mathrm{vol}(\Gamma_s)\log(n)qc_s\right)$ | $Q_s^- \sim \mathrm{Poi}\left(\lambda \mathrm{vol}(\Gamma_s)\log(n)(1 - qc_s)\right)$ |

# Impossibility: Idea

If $\lambda\kappa < 1$ or $\lambda I_\varphi(p,q) < 1$ , then exact recovery is not possible.

$$I_\varphi(p,q) := 2\int_{\mathbb{R}_+} \left[1 - \sqrt{pq}\varphi(x) - \sqrt{(1-p\varphi(x))(1-q\varphi(x))}\right] dx$$

| In $\mathcal{R}_s$ | **Neighbours** | **Non-neighbours** |
|---|---|---|
| Same | $P_s^+ \sim \mathrm{Poi}\big(\lambda \mathrm{vol}(\Gamma_s)\log(n) p c_s\big)$ | $P_s^- \sim \mathrm{Poi}\big(\lambda \mathrm{vol}(\Gamma_s)\log(n)(1-p c_s)\big)$ |
| Different | $Q_s^+ \sim \mathrm{Poi}\big(\lambda \mathrm{vol}(\Gamma_s)\log(n) q c_s\big)$ | $Q_s^- \sim \mathrm{Poi}\big(\lambda \mathrm{vol}(\Gamma_s)\log(n)(1-q c_s)\big)$ |

- Testing Poisson vectors: Prob. of error $= n^{-\lambda \mathrm{vol}(\Gamma_s) D_+(\mathbf{p},\mathbf{q})}$
- Chernoff-Hellinger distance:

$$D_+(\mathbf{p},\mathbf{q}) := 2\sum_{s=1}^{\ell}\left[1 - \sqrt{pq}c_s - \sqrt{(1-pc_s)(1-qc_s)}\right]$$

- Error probability $\to n^{-\lambda I_\varphi(p,q)}$
- Total number of errors $\approx \lambda n^{1-\lambda I_\varphi(p,q)} \to \infty$ when $\lambda I_\varphi(p,q) < 1$

# Achievability

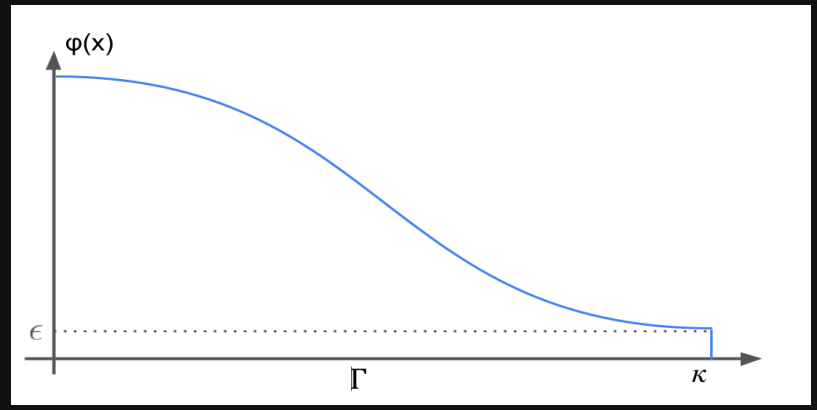
Q. How to recover the communities exactly when $\lambda I_\varphi(p, q) < 1$?

# Achievability

Q. How to recover the communities exactly when $\lambda I_{\varphi}(p, q) < 1$?

**Two phase algorithm:**

Recall $\kappa$: maximum interaction distance



# Achievability

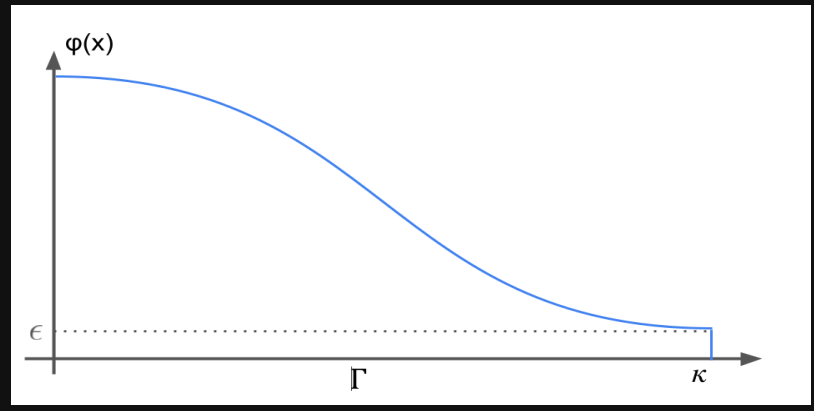
Q. How to recover the communities exactly when $\lambda I_{\varphi}(p, q) < 1$?

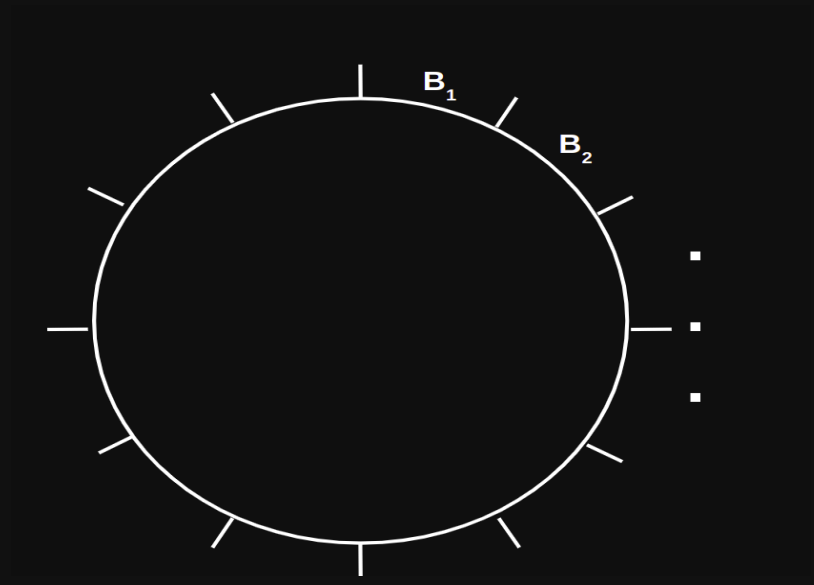
**Two phase algorithm:**

Recall $\kappa$: maximum interaction distance

**Phase-I: Almost-exact recovery**

- Divide into blocks of size $\frac{\kappa}{2}\frac{\log n}{n}$



# Achievability

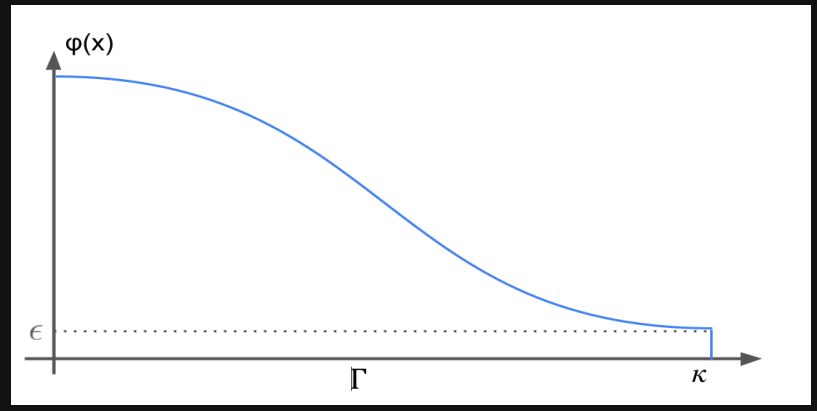
Q. How to recover the communities exactly when $\lambda I_{\varphi}(p, q) < 1$?

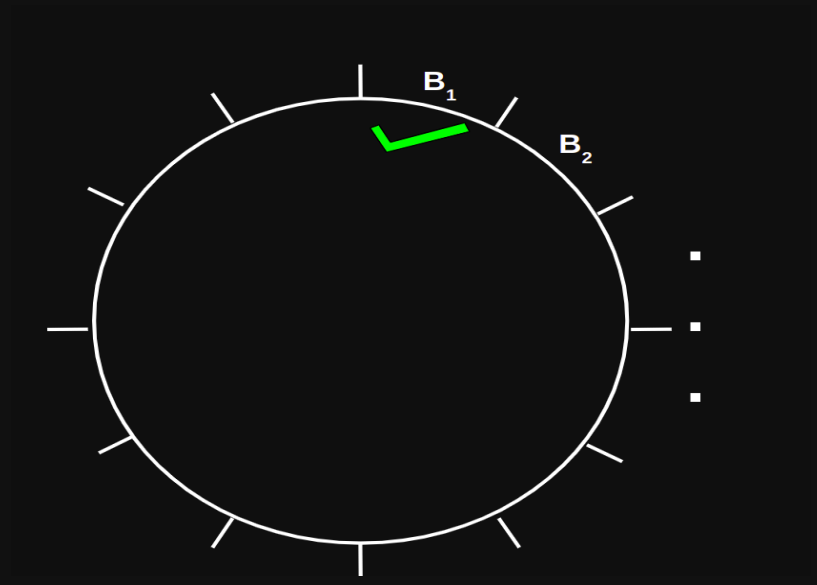
**Two phase algorithm:**

Recall $\kappa$: maximum interaction distance

**Phase-I: Almost-exact recovery**

- Divide into blocks of size $\frac{\kappa}{2}\frac{\log n}{n}$
- Recover exactly in an initial block



# Achievability

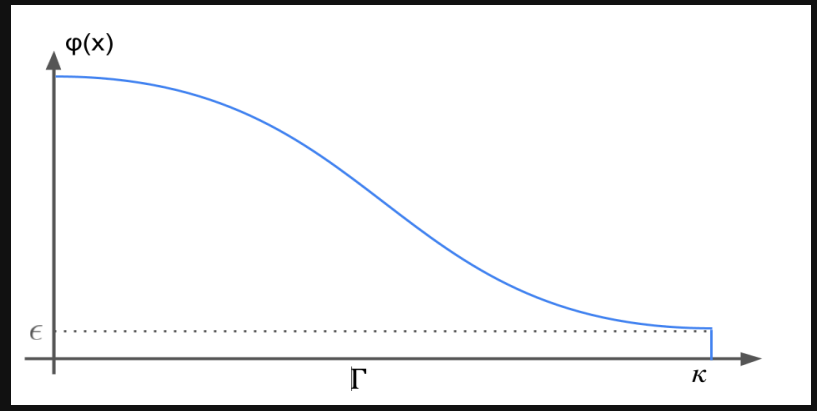
Q. How to recover the communities exactly when $\lambda I_{\varphi}(p, q) < 1$?

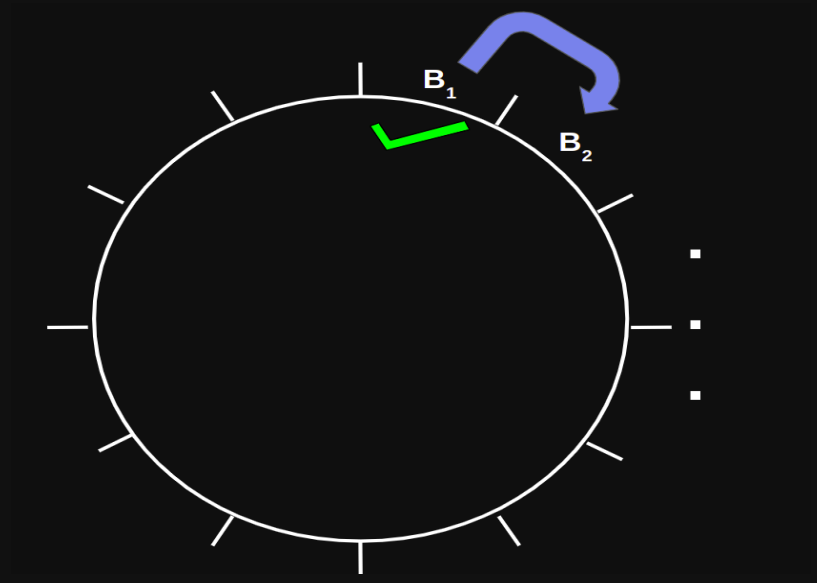
**Two phase algorithm:**

Recall $\kappa$: maximum interaction distance

**Phase-I: Almost-exact recovery**

- Divide into blocks of size $\frac{\kappa}{2}\frac{\log n}{n}$
- Recover exactly in an initial block
- Propagate from a recovered block to adjacent block and so on



# Achievability

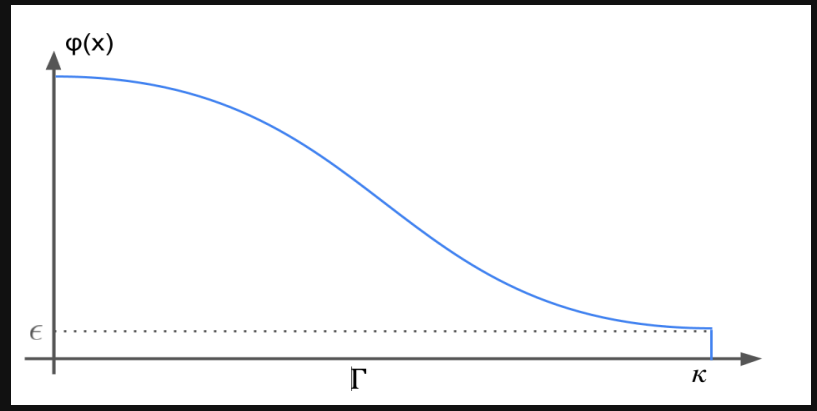
Q. How to recover the communities exactly when $\lambda I_{\varphi}(p, q) < 1$?

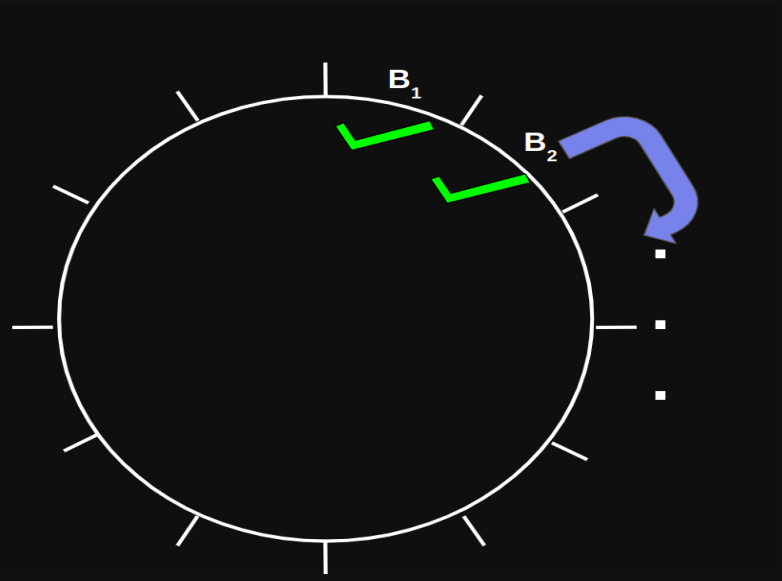
**Two phase algorithm:**

Recall $\kappa$: maximum interaction distance

**Phase-I: Almost-exact recovery**

- Divide into blocks of size $\frac{\kappa}{2}\frac{\log n}{n}$
- Recover exactly in an initial block
- Propagate from a recovered block to adjacent block and so on



# Achievability

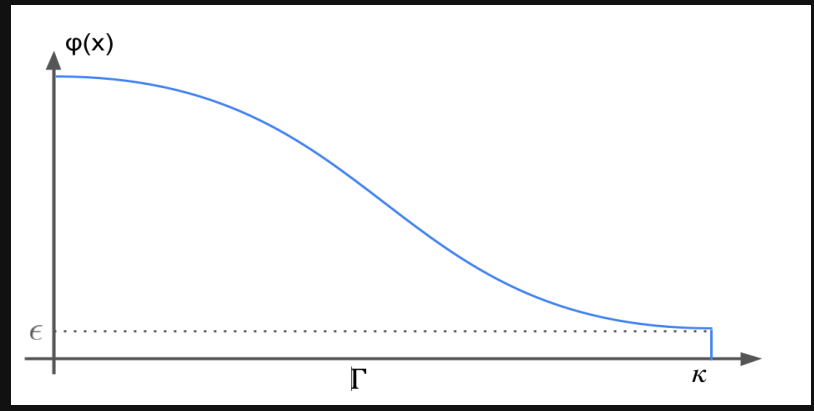
Q. How to recover the communities exactly when $\lambda I_{\varphi}(p, q) < 1$?

**Two phase algorithm:**

Recall $\kappa$: maximum interaction distance

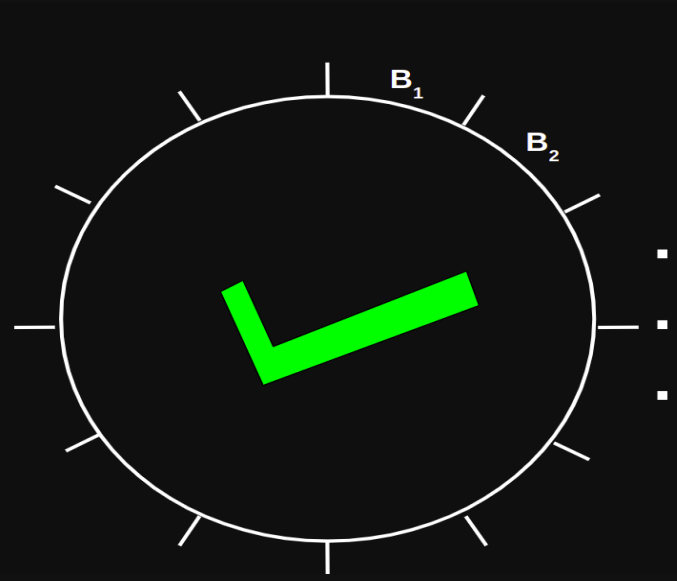
**Phase-I: Almost-exact recovery**

- Divide into blocks of size $\frac{\kappa}{2}\frac{\log n}{n}$
- Recover exactly in an initial block
- Propagate from a recovered block to adjacent block and so on



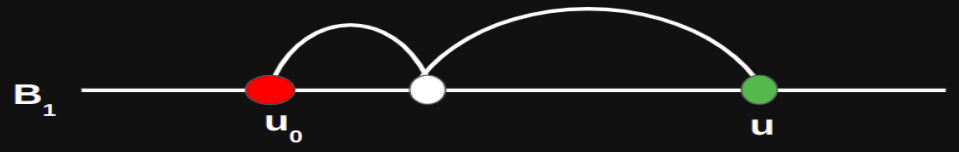

**Phase-II: Refinement step**

- Genie-based correction step



# Recovering the initial block

# Recovering the initial block




- Dense graph within the block.
- Off-the-shelf algorithms for e.g., spectral.
- Choose $u_0 \in V_1$ and set $\hat{\sigma}(u_0) =  + 1$.
- Cluster using number of common neighbours of $u$ and $u_0$

# Recovering the initial block

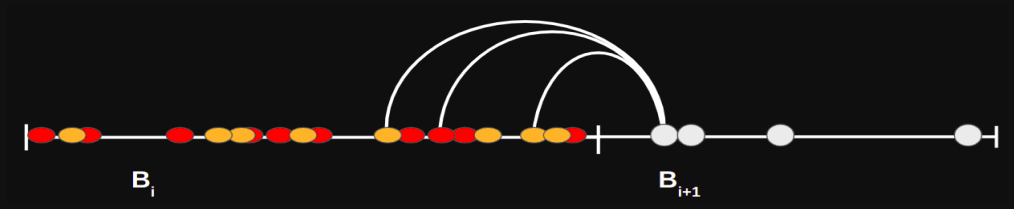

- Dense graph within the block.
- Off-the-shelf algorithms for e.g., spectral.
- Choose $u_0 \in V_1$ and set $\hat{\sigma}(u_0) =  + 1$.
- Cluster using number of common neighbours of $u$ and $u_0$

**Lemma**: For any $p > q$ and $\Delta > 0$, communities of nodes in the initial block $B_1$ are recovered w.h.p., i.e.,

$$\mathrm{P}\Big(\bigcap_{u \in V_1} \{\hat{\sigma}(u) = \sigma(u)\}\Big) \geq 1 - \Delta n^{-c_1 \log n}$$

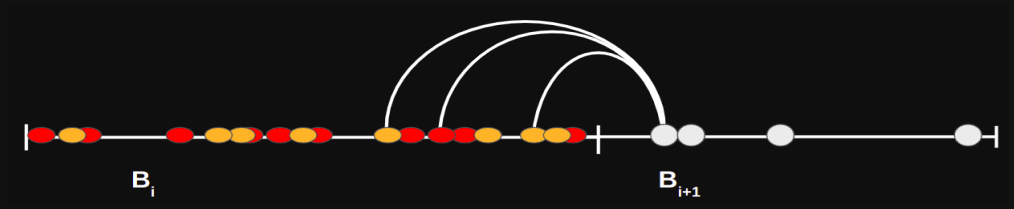
# Label propagation

# Label propagation




- Assume that the estimated communities in block $B_i$ are the true communities.
- Evaluate the log-likelihood of edges to $u \in B_{i+1}$ from vertices in $B_i$
- Set $\hat{\sigma}(u) = \operatorname{sign}\left(\sum_{v \in B_i} \hat{\sigma}(v)\, \mathrm{LLR}(A_{uv})\right)$

# Label propagation




- Assume that the estimated communities in block $B_i$ are the true communities.
- Evaluate the log-likelihood of edges to $u \in B_{i+1}$ from vertices in $B_i$
- Set $\hat{\sigma}(u) = \operatorname{sign}\left(\sum_{v \in B_i} \hat{\sigma}(v)\, \mathrm{LLR}(A_{uv})\right)$

**Lemma**: There exists an $M \equiv M(p, q, \varphi) > 0$ such that the event

$$\mathcal{A}_i := \{\text{at most } M \text{ mistakes within block } B_i\}$$

satisfies $\mathrm{P}\left(\mathcal{A}_{i+1} \,\middle|\, \mathcal{A}_i\right) \geq 1 - c_2 n^{-9/8}$.

# Label propagation: Main idea

Let $\mathcal{A}_i = \{\text{at most } M \text{ mistakes within block } B_i\}$ and $b = \#\text{ of blocks} = \frac{2n}{\kappa \log n}$.

$$\mathbb{P}\Big(\bigcap_{i=1}^{b} \mathcal{A}_i\Big) = \mathbb{P}(\mathcal{A}_1) \prod_{i=2}^{b} \mathbb{P}\Big(\mathcal{A}_i \Big| \bigcap_{j<i} \mathcal{A}_j\Big)$$

# Label propagation: Main idea

Let $\mathcal{A}_i = \{\text{at most } M \text{ mistakes within block } B_i\}$ and $b = \#\text{ of blocks} = \frac{2n}{\kappa \log n}$.

$$\mathrm{P}\left(\bigcap_{i=1}^{b} \mathcal{A}_i\right) = \mathrm{P}(\mathcal{A}_1) \cdot \prod_{i=2}^{b} \mathrm{P}\left(\mathcal{A}_i \,\middle|\, \mathcal{A}_{i-1}\right)$$

# Label propagation: Main idea

Let $\mathcal{A}_i = \{\text{at most } M \text{ mistakes within block } B_i\}$ and $b = \#\text{ of blocks} = \frac{2n}{\kappa \log n}$.

$$\mathrm{P}\left(\bigcap_{i=1}^{b} \mathcal{A}_i\right) = \mathrm{P}(\mathcal{A}_1) \cdot \prod_{i=2}^{b} \mathrm{P}\left(\mathcal{A}_i \,\middle|\, \mathcal{A}_{i-1}\right)$$

$$\geq (1 - \Delta n^{-c_1} \log n) \cdot (1 - c_2 n^{-9/8})^{\frac{2n}{\kappa \log n}}$$

# Label propagation: Main idea

Let $\mathcal{A}_i = \{\text{at most } M \text{ mistakes within block } B_i\}$ and $b = \#\text{ of blocks} = \frac{2n}{\kappa \log n}$.

$$\mathrm{P}\left(\bigcap_{i=1}^{b} \mathcal{A}_i\right) = \mathrm{P}(\mathcal{A}_1) \cdot \prod_{i=2}^{b} \mathrm{P}\left(\mathcal{A}_i \,\middle|\, \mathcal{A}_{i-1}\right)$$

$$\geq (1 - \Delta n^{-c_1} \log n) \cdot (1 - c_2 n^{-9/8})^{\frac{2n}{\kappa \log n}}$$

$$\geq (1 - \Delta n^{-c_1} \log n) \cdot \left(1 - \frac{2c_2}{n^{1/8} \kappa \log n}\right)$$

Sacrifice on probability but ensure at most $M$ mistakes within each block.

# Label propagation: Main idea

Let $\mathcal{A}_i = \{\text{at most } M \text{ mistakes within block } B_i\}$ and $b = \#\text{ of blocks} = \frac{2n}{\kappa \log n}$.

$$\mathrm{P}\left(\bigcap_{i=1}^{b} \mathcal{A}_i\right) = \mathrm{P}(\mathcal{A}_1) \cdot \prod_{i=2}^{b} \mathrm{P}\left(\mathcal{A}_i \,\middle|\, \mathcal{A}_{i-1}\right)$$

$$\geq (1 - \Delta n^{-c_1 \log n}) \cdot (1 - c_2 n^{-9/8})^{\frac{2n}{\kappa \log n}}$$

$$\geq (1 - \Delta n^{-c_1 \log n}) \cdot \left(1 - \frac{2c_2}{n^{1/8} \kappa \log n}\right)$$

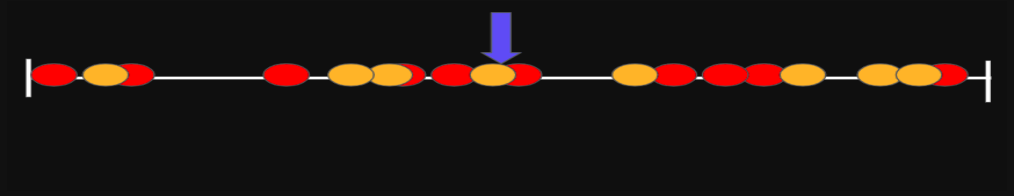
Sacrifice on probability but ensure at most $M$ mistakes within each block.

**Lemma**: Fix any $\eta > 0$. For $A \sim \mathrm{GKBM}(\lambda, n, p, q, \varphi)$, we have that satisfies

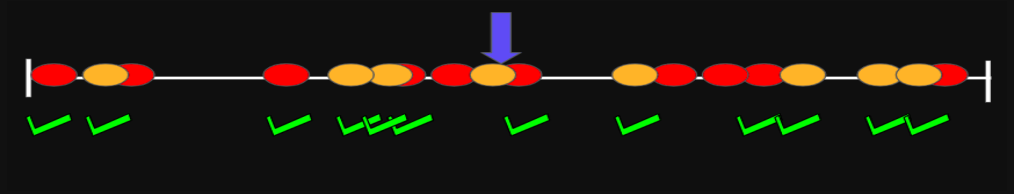
$$\mathrm{P}\left(\text{Total \# of mistakes} \leq \frac{\eta n}{3\kappa}\right) = 1 - o(1).$$

# Refinement step: Phase II

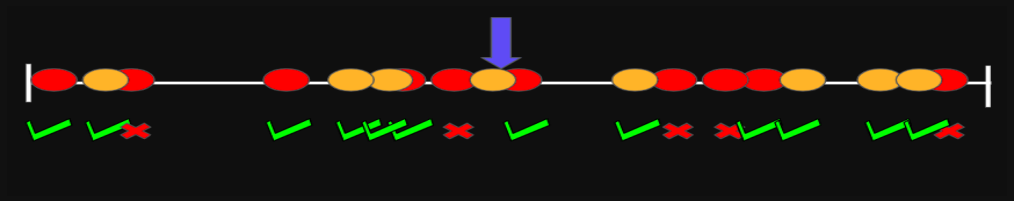
# Refinement step: Phase II



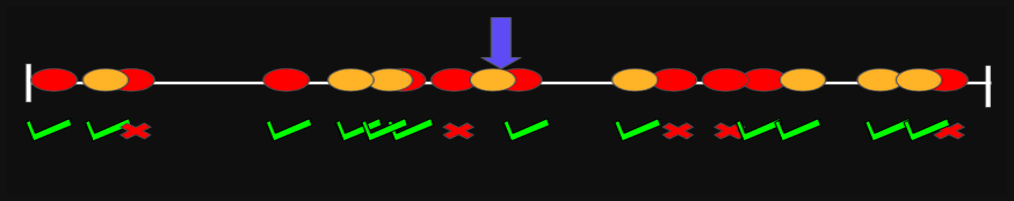
# Refinement step: Phase II



# Refinement step: Phase II

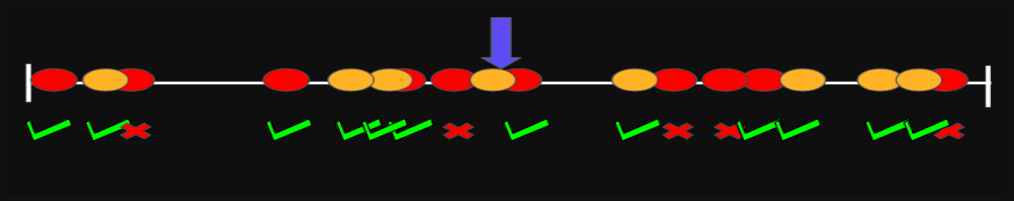


# Refinement step: Phase II



- Genie-based estimate: Assign $g(u, \hat{\sigma}) = \text{sign}\left( \sum_{v \in \mathcal{V}_i} \hat{\sigma}(v) \, \text{LLR}(A_{uv}) \right)$

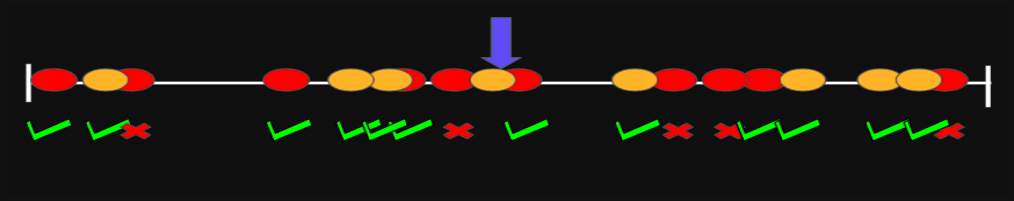
# Refinement step: Phase II



- Genie-based estimate: Assign $g(u, \hat{\sigma}) = \text{sign}\left( \sum_{v \in \mathcal{V}_i} \hat{\sigma}(v) \, \text{LLR}(A_{uv}) \right)$
- Bound the worst-case error:

$$|g(u, \hat{\sigma}) - g(u, \sigma)| \leq \beta\eta \log n.$$

# Refinement step: Phase II



- Genie-based estimate: Assign $g(u, \hat{\sigma}) = \operatorname{sign}\left(\sum_{v \in \mathcal{V}_i} \hat{\sigma}(v)\, \mathrm{LLR}(A_{uv})\right)$
- Bound the worst-case error:

$$|g(u, \hat{\sigma}) - g(u, \sigma)| \leq \beta\eta \log n.$$

- Use simple function approximation

$$\mathrm{P}(g(u, \hat{\sigma}) > 0 \mid \sigma(u) = -1) \leq n^{[\beta\eta - \lambda I_\varphi(p,q)]}.$$

- Take $\eta = \dfrac{\lambda I_\varphi(p,q) - 1}{2\beta} > 0$ and using the union bound

$$\mathrm{P}(\exists u : g(u, \hat{\sigma}) \neq \sigma(u)) = o(1).$$

# Conclusions

- $\lambda\kappa < 1$ or $\lambda I_\varphi(p, q) < 1$, then communities cannot be recovered.
- $\lambda\kappa > 1$ and $\lambda I_\varphi(p, q) > 1$, then exact community recovery possible using a linear time algorithm.
- Multiple communities and higher dimensions.
- Main takeaway: Geometry helps in global inference tasks.

# Conclusions

- $\lambda\kappa < 1$ or $\lambda I_\varphi(p, q) < 1$, then communities cannot be recovered.
- $\lambda\kappa > 1$ and $\lambda I_\varphi(p, q) > 1$, then exact community recovery possible using a linear time algorithm.
- Multiple communities and higher dimensions.
- Main takeaway: Geometry helps in global inference tasks.

# Current and future work

- Graphs with node inhomogenities and long-distance edges.

- Joint kernel estimation and community detection.
- Detecting communities with no information of location or in the semi-supervised regime.
- Spectral and SDP algorithms.[Abbe, Fan, Wang, Zhong (2020)]

- Inference problems on dynamic graphs
- Community detection and percolation.
- Percolation games and games on geometric graphs.

# Teaching profile

- Teaching aids: online assignments, explanatory videos, in-class interactive quizzes

# Teaching profile

- Teaching aids: online assignments, explanatory videos, in-class interactive quizzes
- Courses @ IIT-B

| Dept. | Existing courses | New courses |
|---|---|---|
| IEOR | IE 611 - Introduction to Stochastic Models | **Network Science**: small-world networks, hubs, directed and weighted networks, clustering, multilayer networks, hypergraph models, SIR models, mean-field approaches |
| | IE 615 - Data Analytics in Operations Research | |
| | IE 506 - Machine Learning: Principles and Techniques | |
| Electrical Engineering | EE 621 - Markov Chains and Queuing Systems | **Network Statistics**: random graph models, network motifs, community detection, link prediction, contagion processes on networks, spectral graph theory |
| | EE 229 - Signal Processing I | |
| | EE 706 - Communication Networks | |
| C-MInDS | DS 303 - Introduction to Machine Learning | |
| Mathematics | SI 539 - Random Graphs | **Percolation**: bond percolation on $\mathbb{Z}^2$, continuum percolation |
| | SI 424 - Statistical Inference I | |

# Teaching profile

- Teaching aids: online assignments, explanatory videos, in-class interactive quizzes
- Courses @ IIT-B

| Dept. | Existing courses | New courses |
|---|---|---|
| IEOR | IE 611 - Introduction to Stochastic Models | **Network Science**: small-world networks, hubs, directed and weighted networks, clustering, multilayer networks, hypergraph models, SIR models, mean-field approaches |
| | IE 615 - Data Analytics in Operations Research | |
| | IE 506 - Machine Learning: Principles and Techniques | |
| Electrical Engineering | EE 621 - Markov Chains and Queuing Systems | **Network Statistics**: random graph models, network motifs, community detection, link prediction, contagion processes on networks, spectral graph theory |
| | EE 229 - Signal Processing I | |
| | EE 706 - Communication Networks | |
| C-MInDS | DS 303 - Introduction to Machine Learning | |
| Mathematics | SI 539 - Random Graphs | **Percolation**: bond percolation on $\mathbb{Z}^2$, continuum percolation |
| | SI 424 - Statistical Inference I | |

- Inter-disciplinary field: adaptable to different levels of mathematical difficulty