

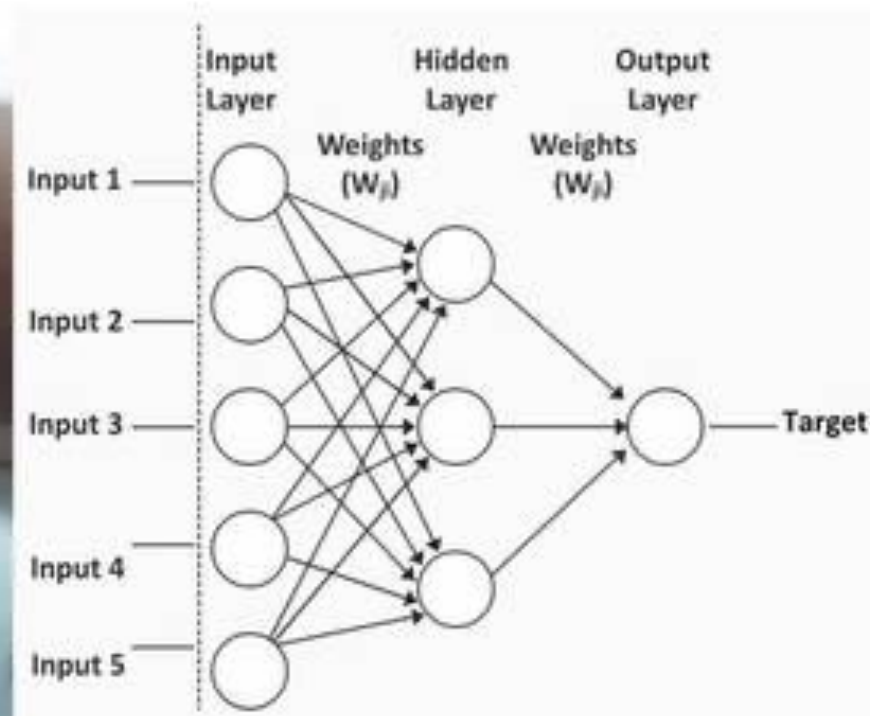
Network models - I

"I work with **models**"

Others:



Me:

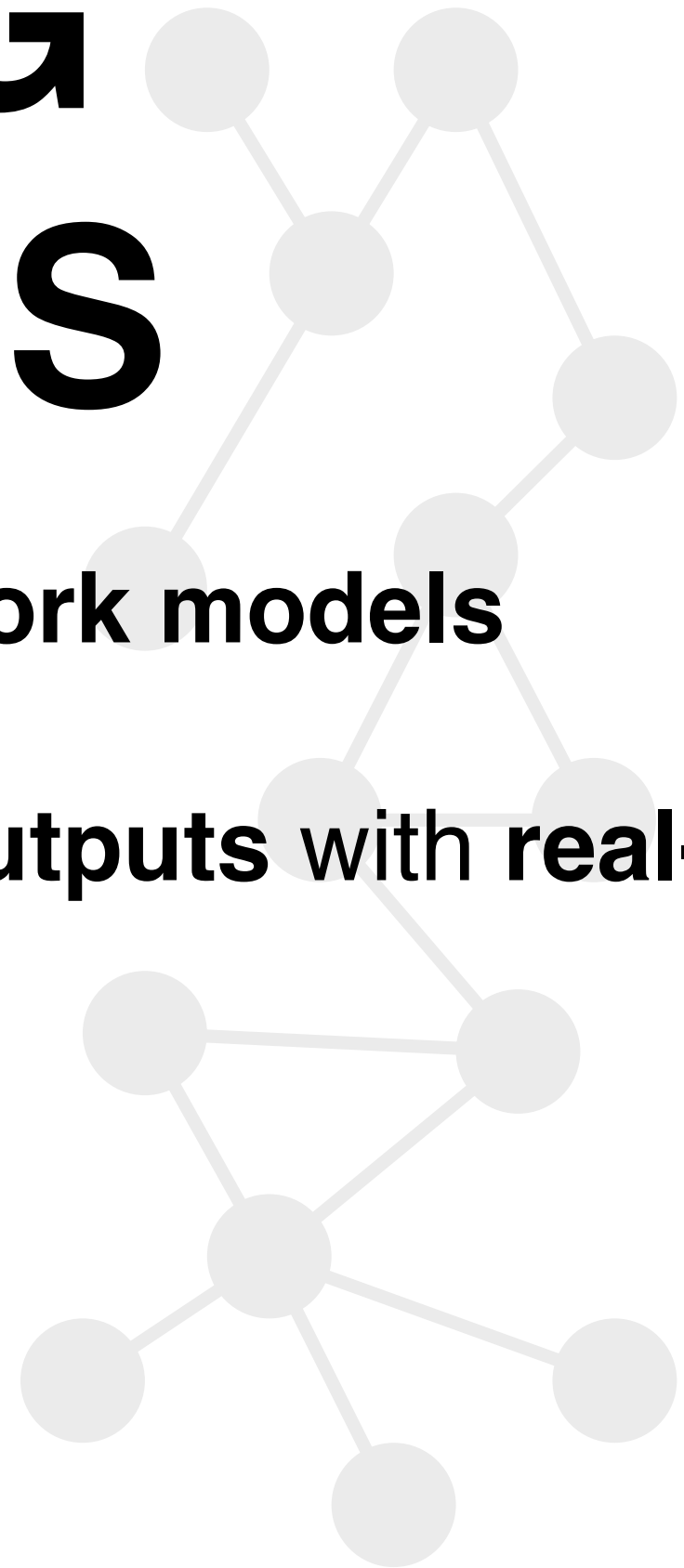


LEARNING OUTCOMES

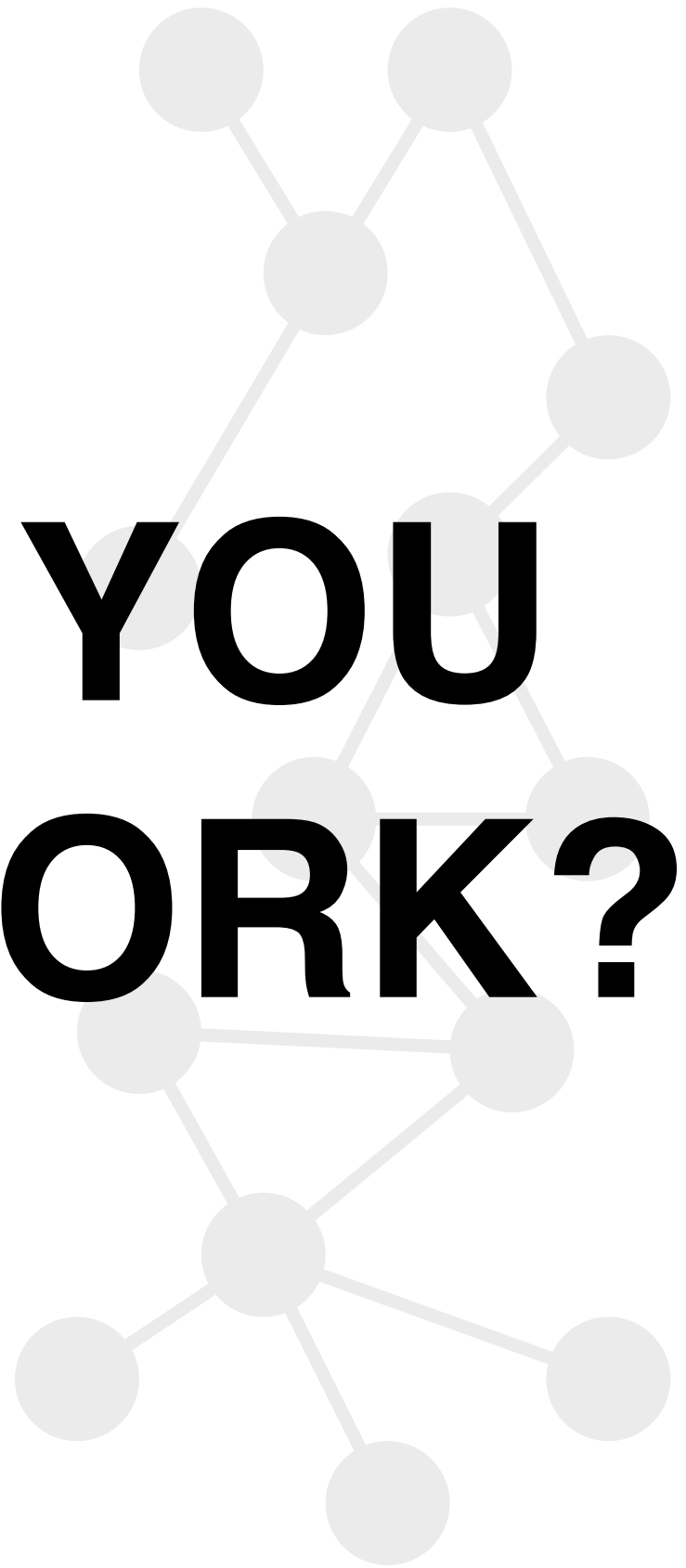
Learn about different **generative network models**

Compare these models' **theoretical outputs** with **real-world networks**

Step-up your network analysis game



**HOW WOULD YOU
BUILD A NETWORK?**

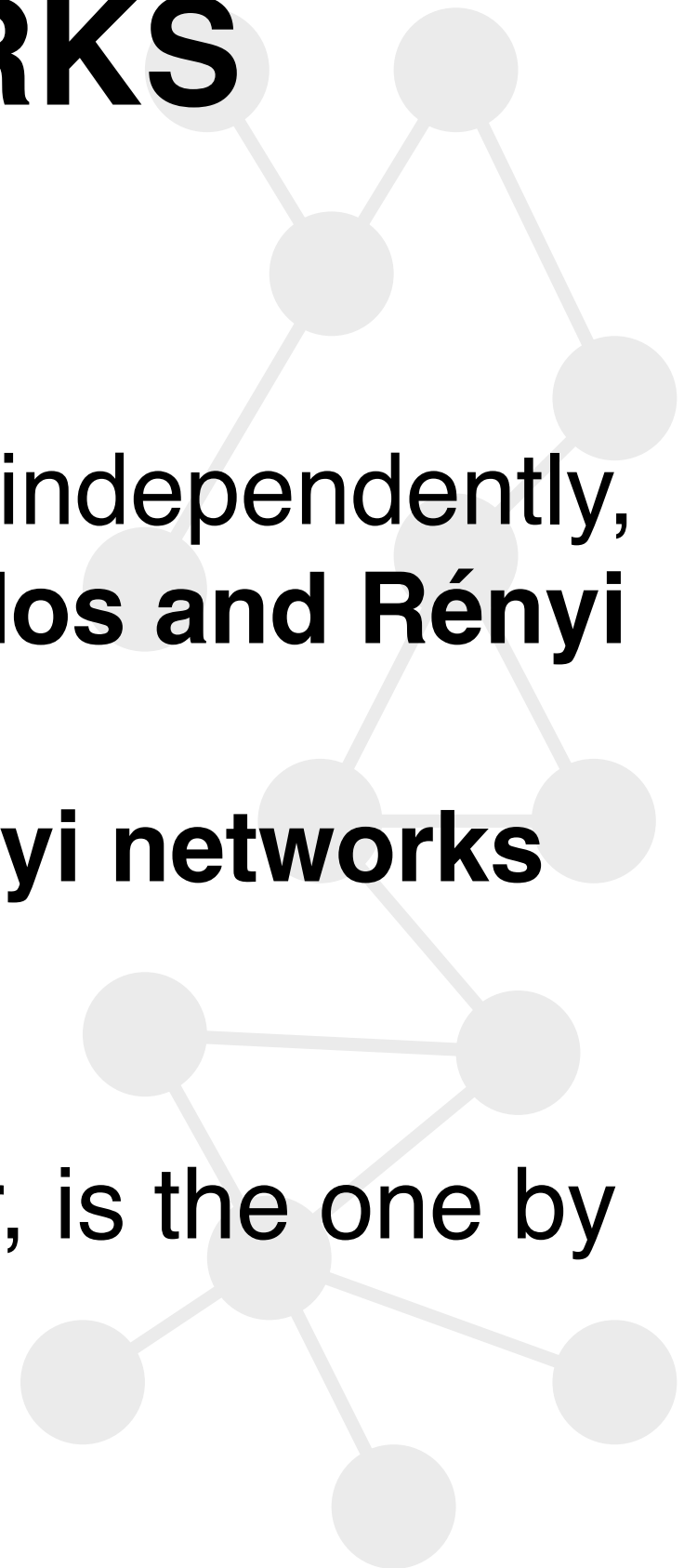


RANDOM NETWORKS

Two slightly different versions proposed independently, one by **Gilbert** and the other one by **Erdos and Rényi**

For reasons, both are called **Erdos-Rényi networks** (**ER networks**)

The most common formulation, however, is the one by **Gilbert**



RANDOM NETWORKS

Gilbert model

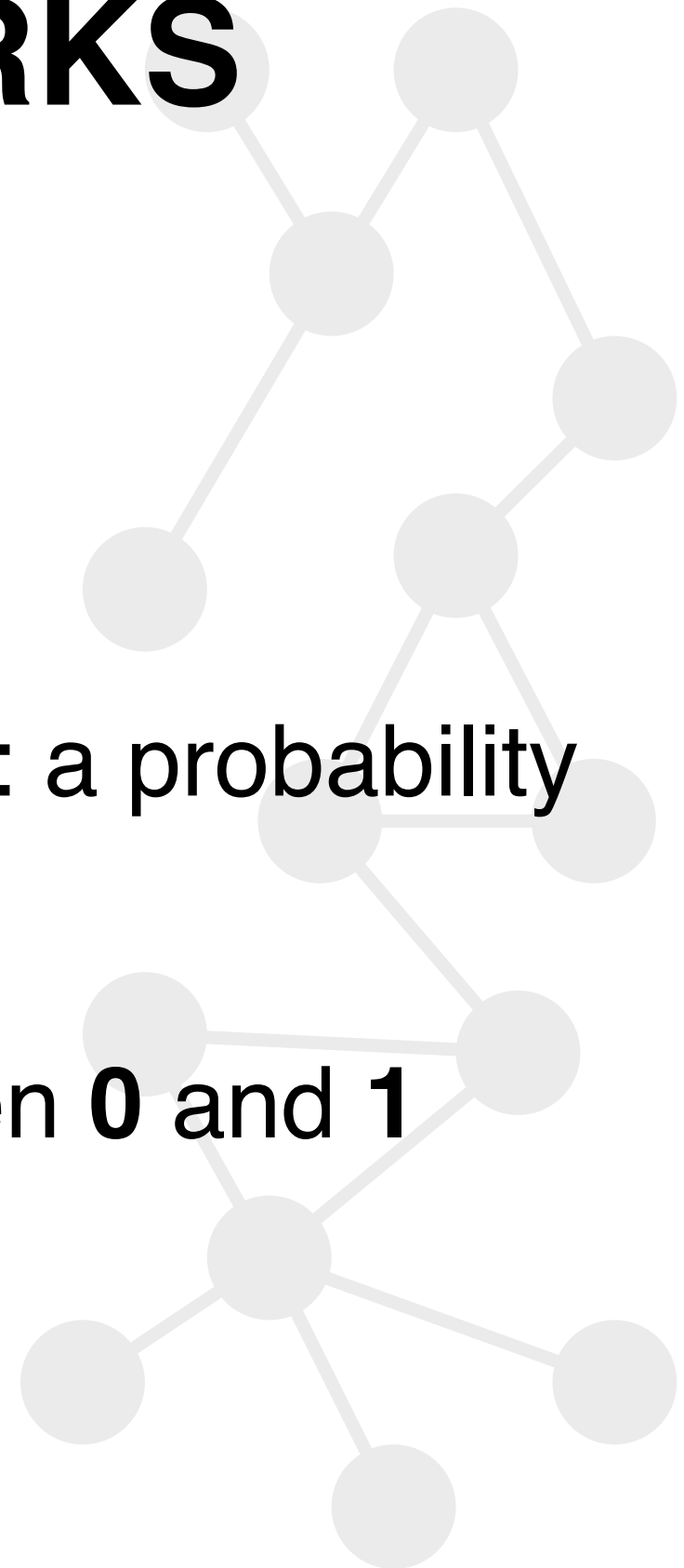
Parameters: **N**: number of nodes and **p**: a probability

Algorithm:

For each pair of nodes i, j :

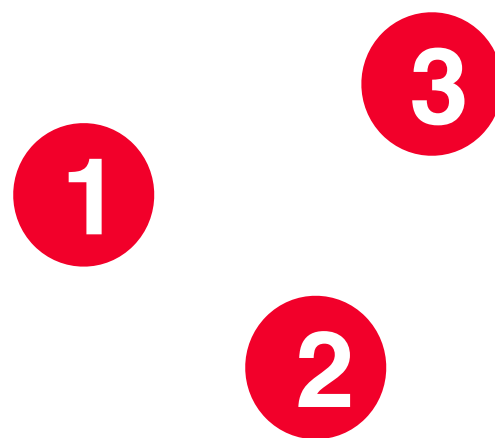
Generate a random number r between **0** and **1**
(**uniformly distributed**)

If $r < p$ make a link between i, j



RANDOM NETWORKS

Gilbert model



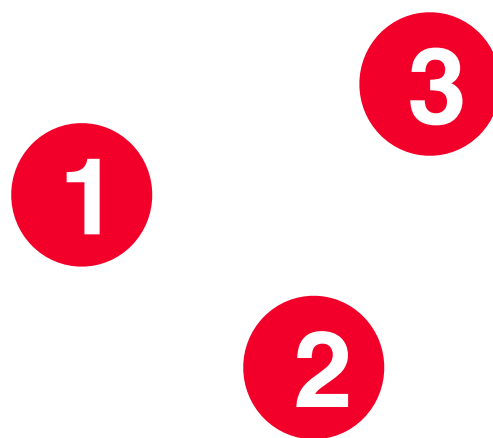
$p=0.5$



RANDOM NETWORKS

Gilbert model

i,j	r
1,2	0.12
1,3	
2,3	



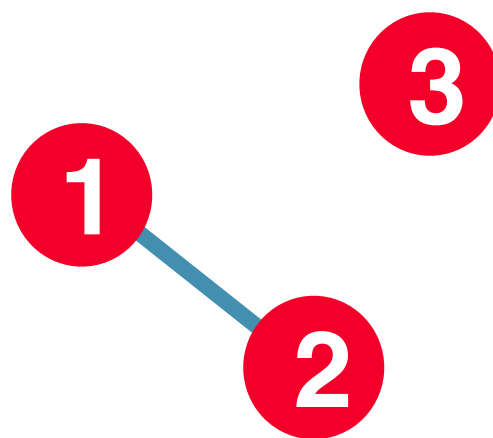
p=0.5



RANDOM NETWORKS

Gilbert model

i,j	r
1,2	0.12
1,3	
2,3	



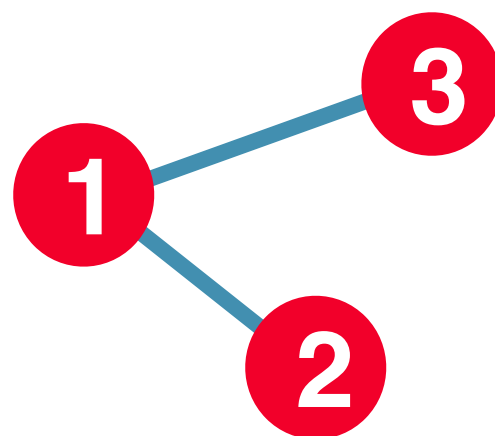
$p=0.5$



RANDOM NETWORKS

Gilbert model

i,j	r
1,2	0.12
1,3	0.29
2,3	



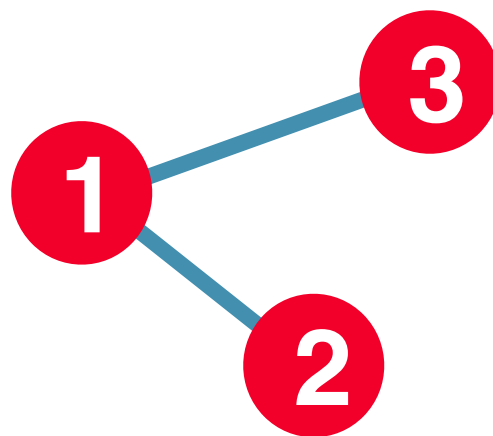
p=0.5



RANDOM NETWORKS

Gilbert model

i,j	r
1,2	0.12
1,3	0.29
2,3	0.72



p=0.5



RANDOM NETWORKS

Erdos-Renyi model

Parameters: **N**: number of nodes and **L**: number of links

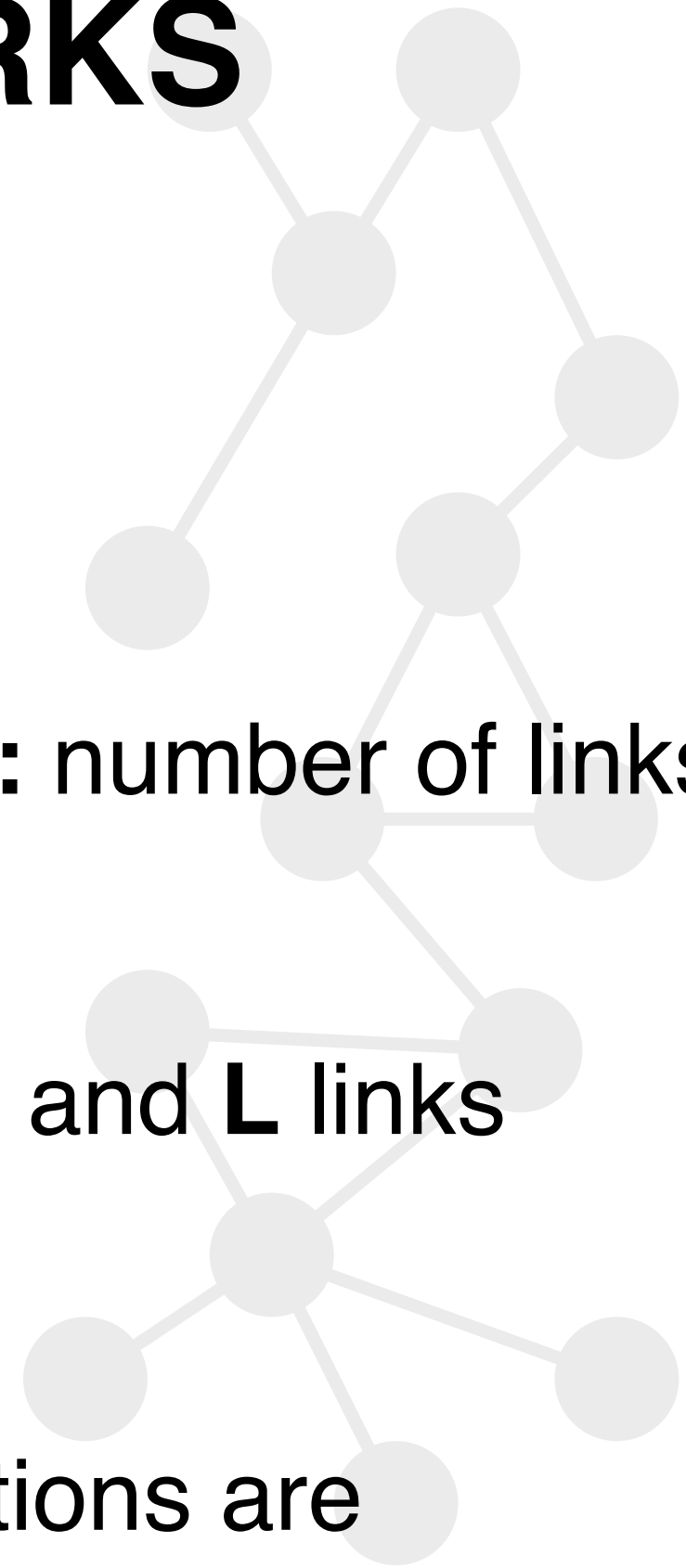
Algorithm:

Build all possible networks with **N** nodes and **L** links

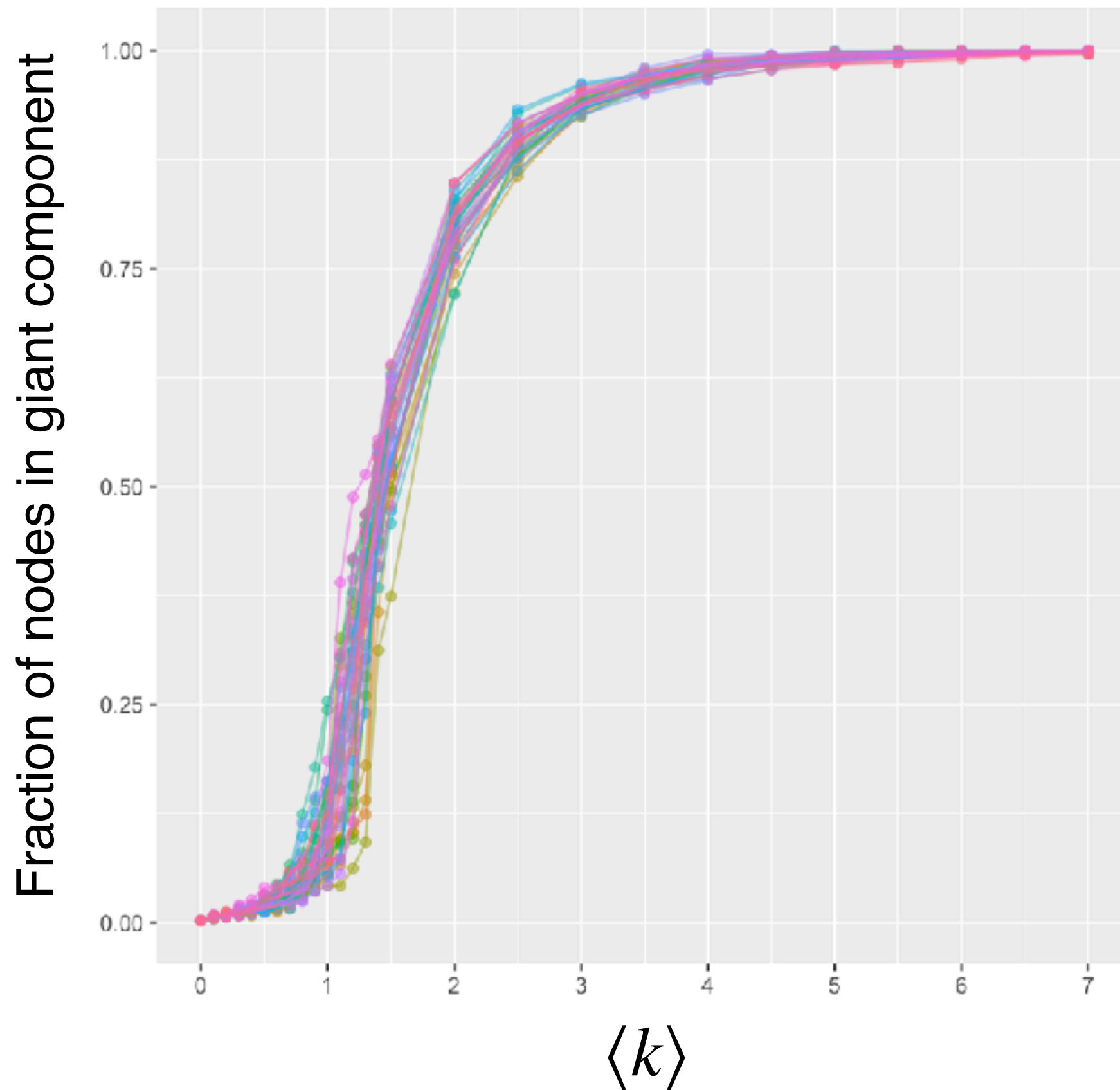
Pick one randomly

Remarks:

For large values of **L**, these two formulations are (almost) equivalent

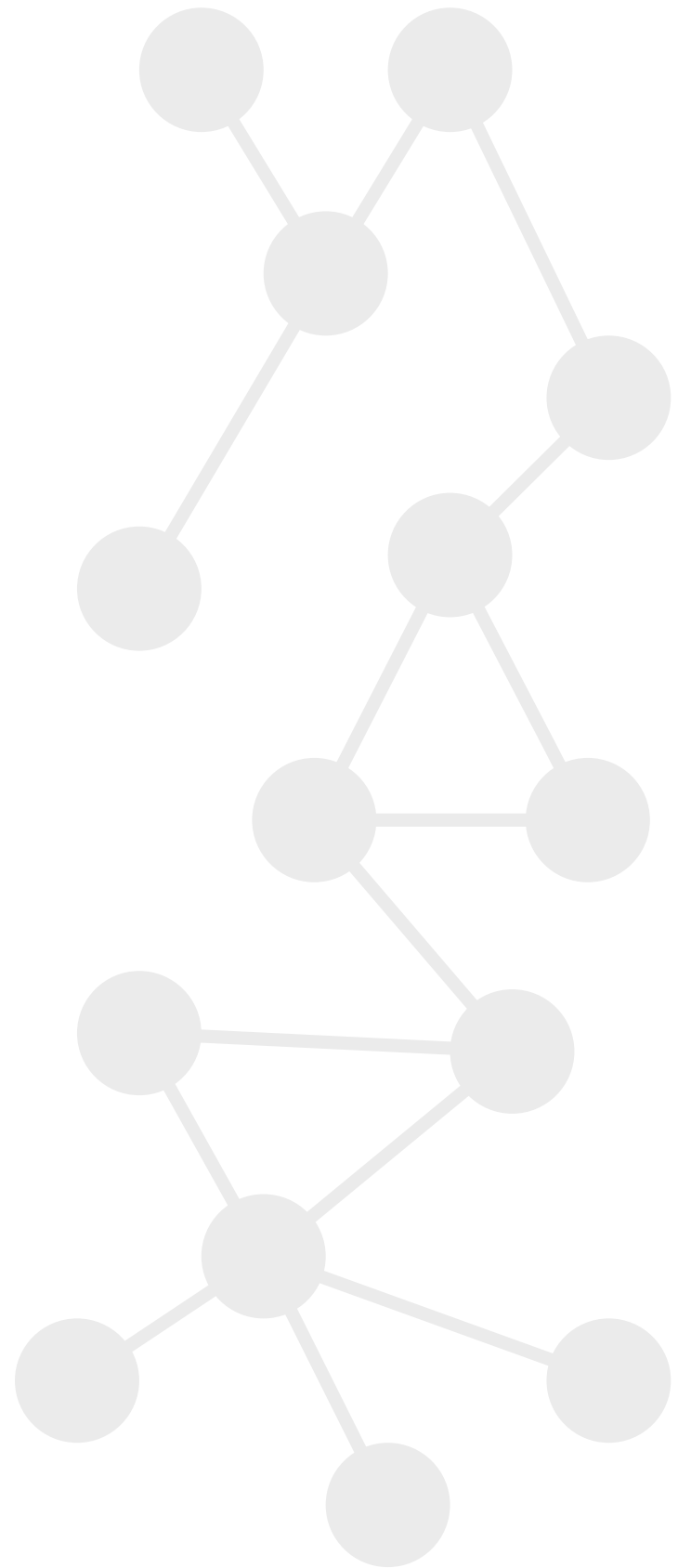


PHASE TRANSITION



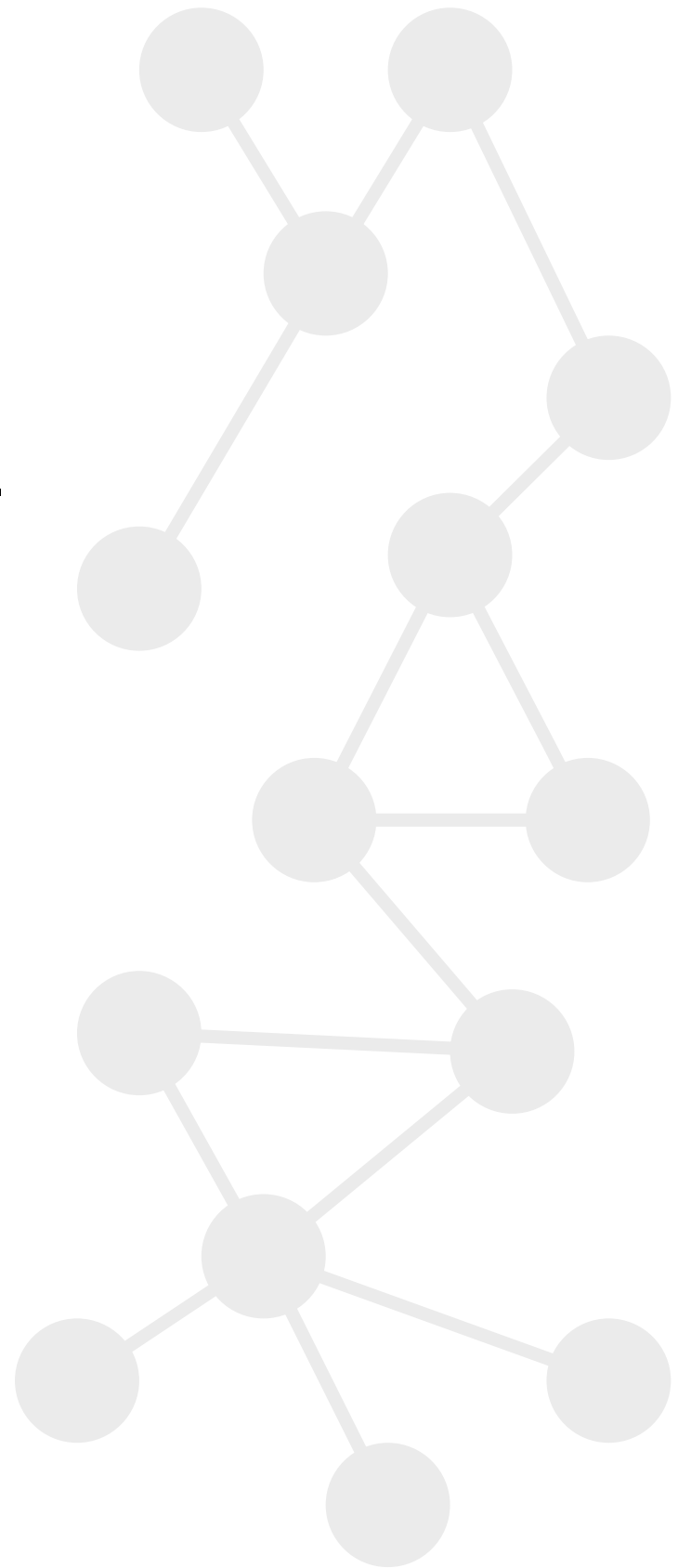
DENSITY

$$\langle L \rangle = ?$$



DENSITY

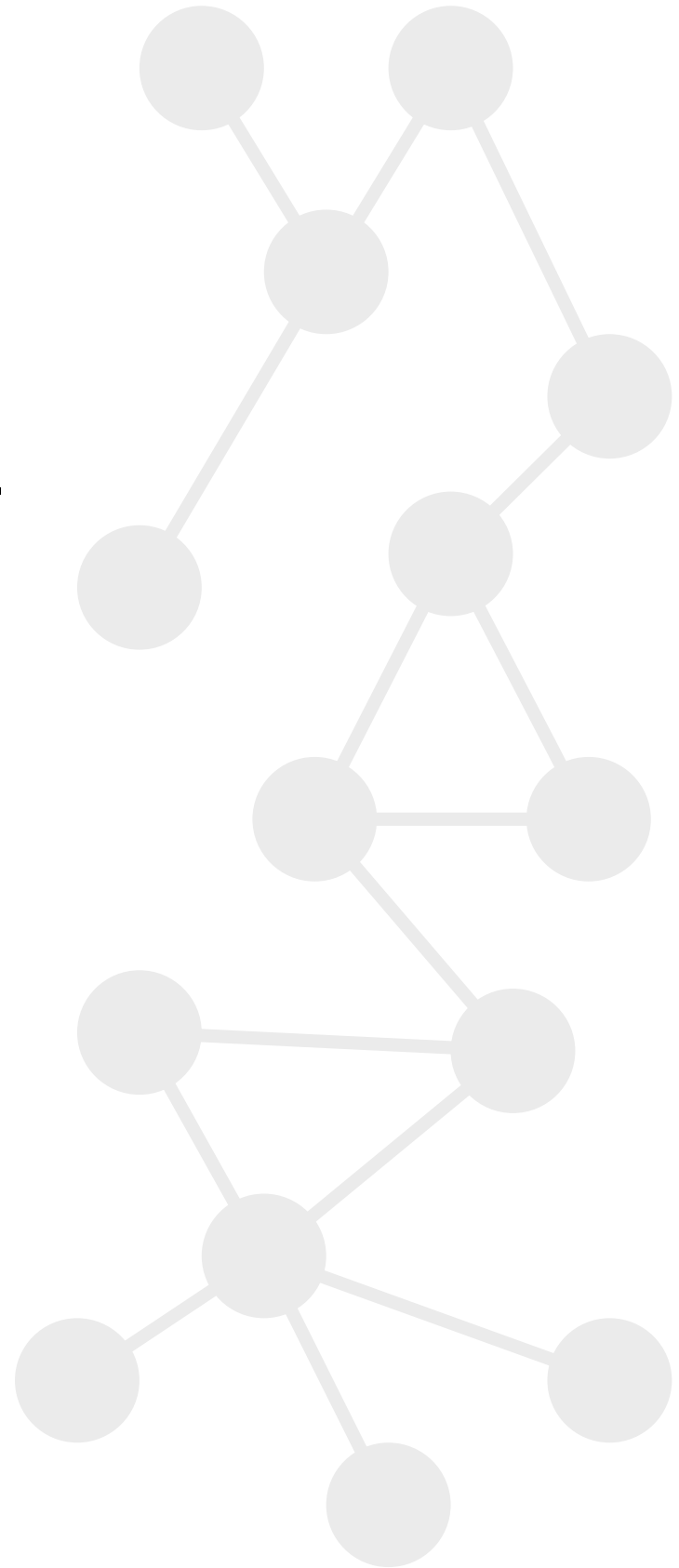
$$\langle L \rangle = p \binom{N}{2} = p \frac{N(N-1)}{2}$$



DENSITY

$$\langle L \rangle = p \binom{N}{2} = p \frac{N(N-1)}{2}$$

$$\langle k \rangle = ?$$



DENSITY

$$\langle L \rangle = p \binom{N}{2} = p \frac{N(N-1)}{2}$$

$$\langle k \rangle = \frac{2\langle L \rangle}{N} = p(N-1)$$



DENSITY

$$\langle L \rangle = p \binom{N}{2} = p \frac{N(N-1)}{2}$$

$$\langle k \rangle = \frac{2\langle L \rangle}{N} = p(N-1)$$

$$\langle d \rangle = ?$$



DENSITY

$$\langle L \rangle = p \binom{N}{2} = p \frac{N(N-1)}{2}$$

$$\langle k \rangle = \frac{2\langle L \rangle}{N} = p(N-1)$$

$$\langle d \rangle = \frac{L}{L_{max}}$$



DENSITY

$$\langle L \rangle = p \binom{N}{2} = p \frac{N(N-1)}{2}$$

$$\langle k \rangle = \frac{2\langle L \rangle}{N} = p(N-1)$$

$$\langle d \rangle = \frac{\langle L \rangle}{L_{max}} = \frac{p \frac{N(N-1)}{2}}{L_{max}}$$

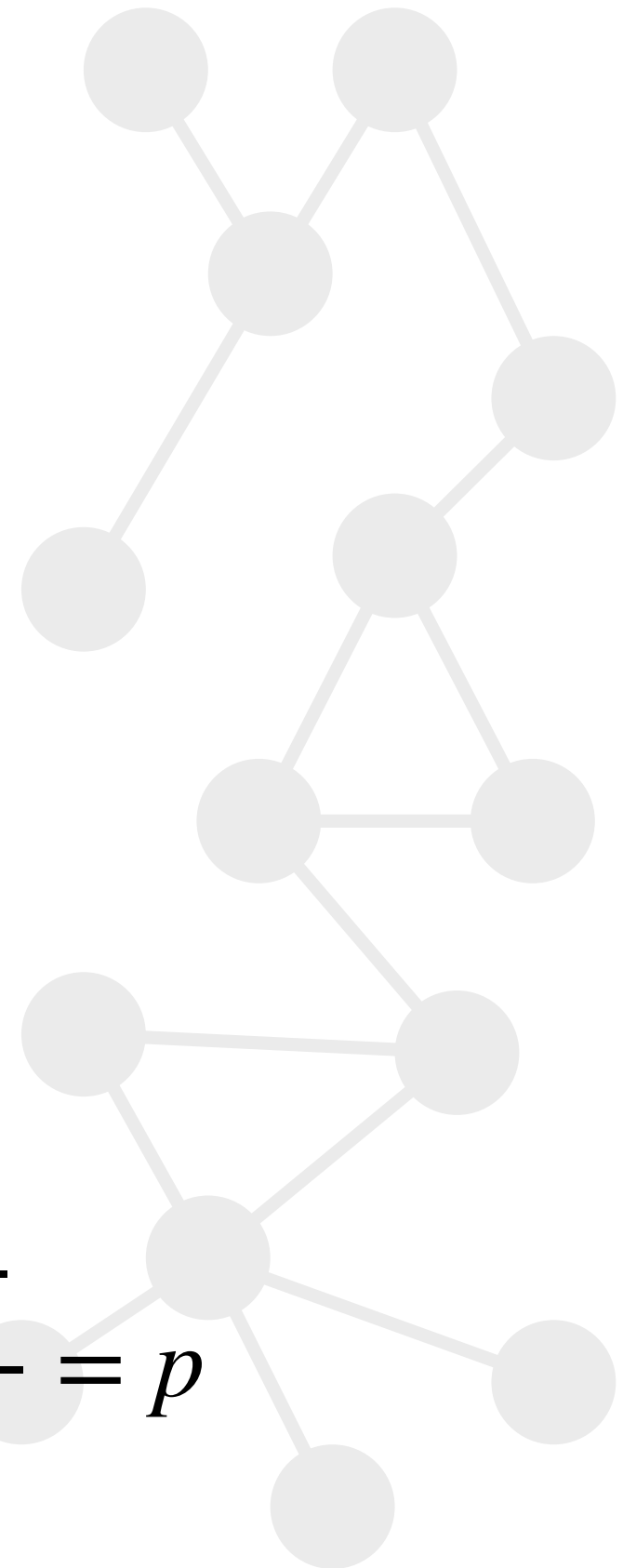


DENSITY

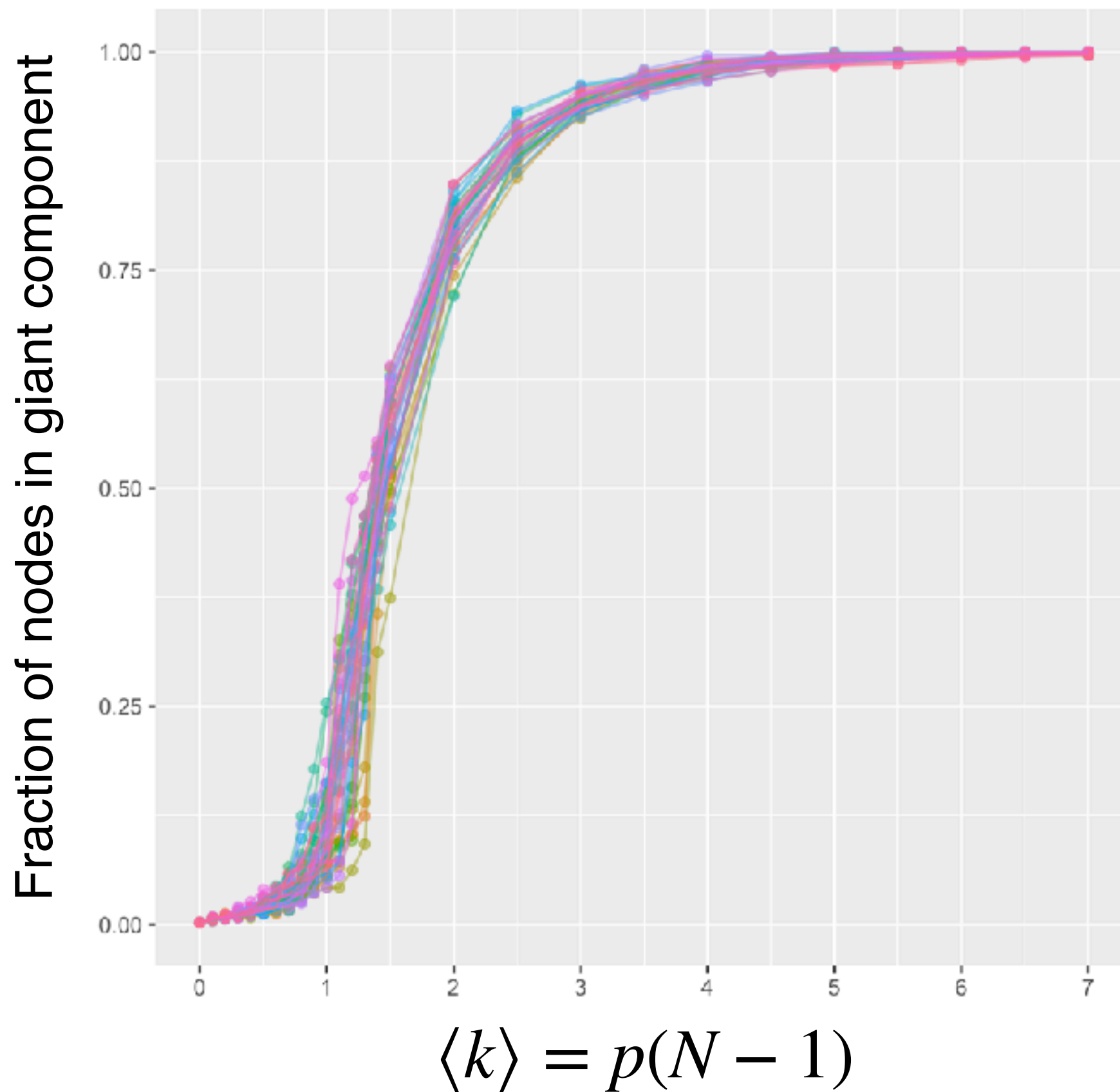
$$\langle L \rangle = p \binom{N}{2} = p \frac{N(N-1)}{2}$$

$$\langle k \rangle = \frac{2\langle L \rangle}{N} = p(N-1)$$

$$\langle d \rangle = \frac{\langle L \rangle}{L_{max}} = \frac{p \frac{N(N-1)}{2}}{\frac{N(N-1)}{2}} = p$$



PHASE TRANSITION



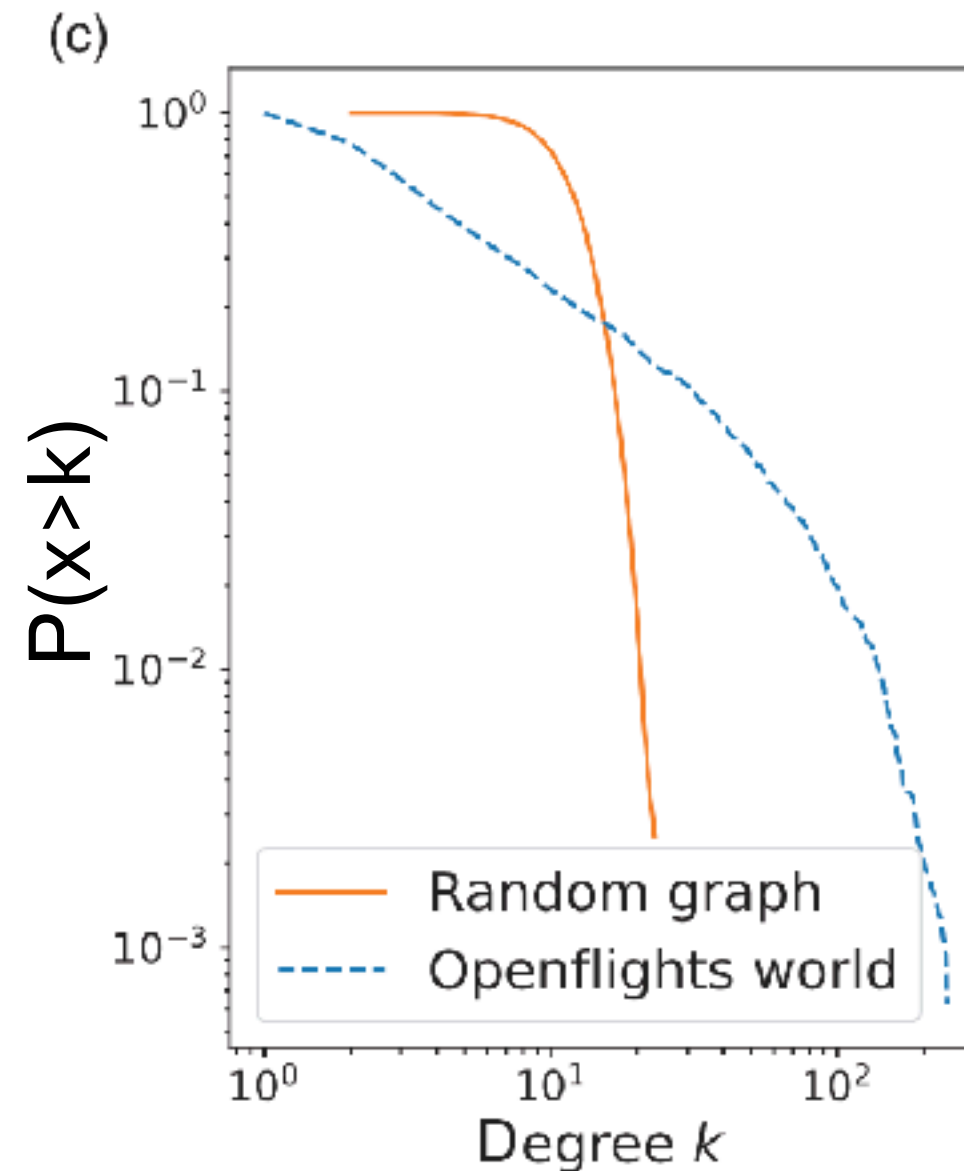
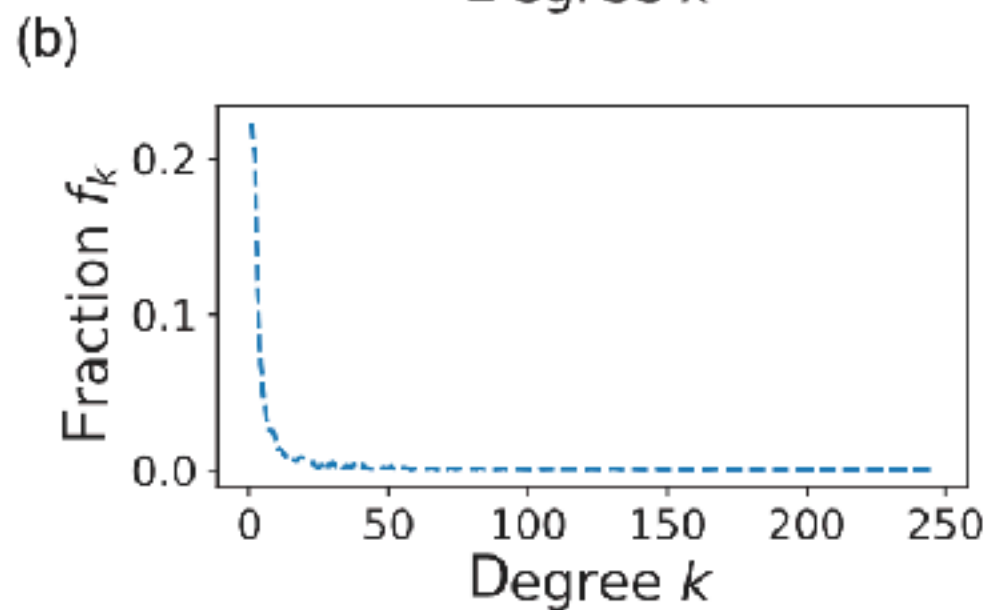
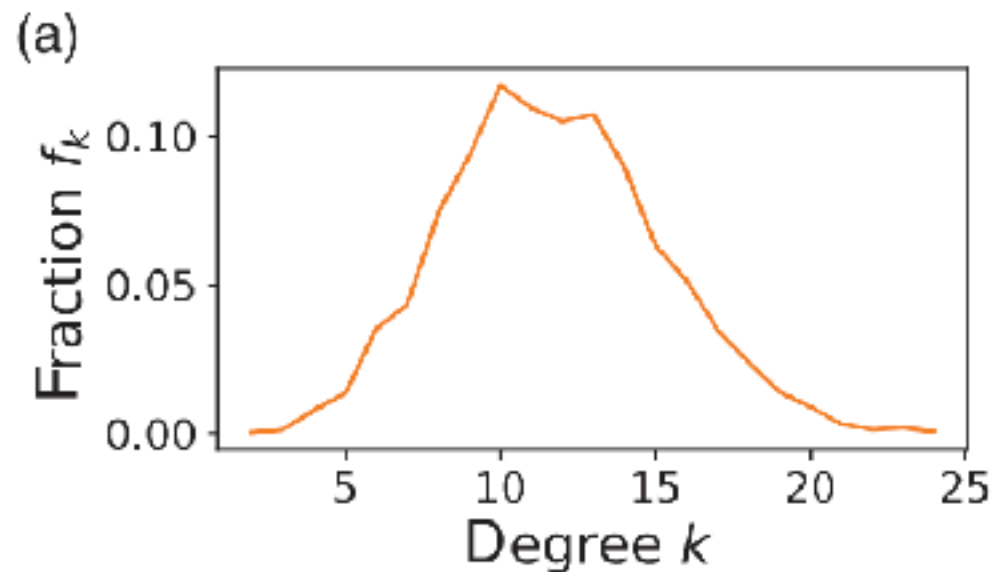
DEGREE DISTRIBUTION

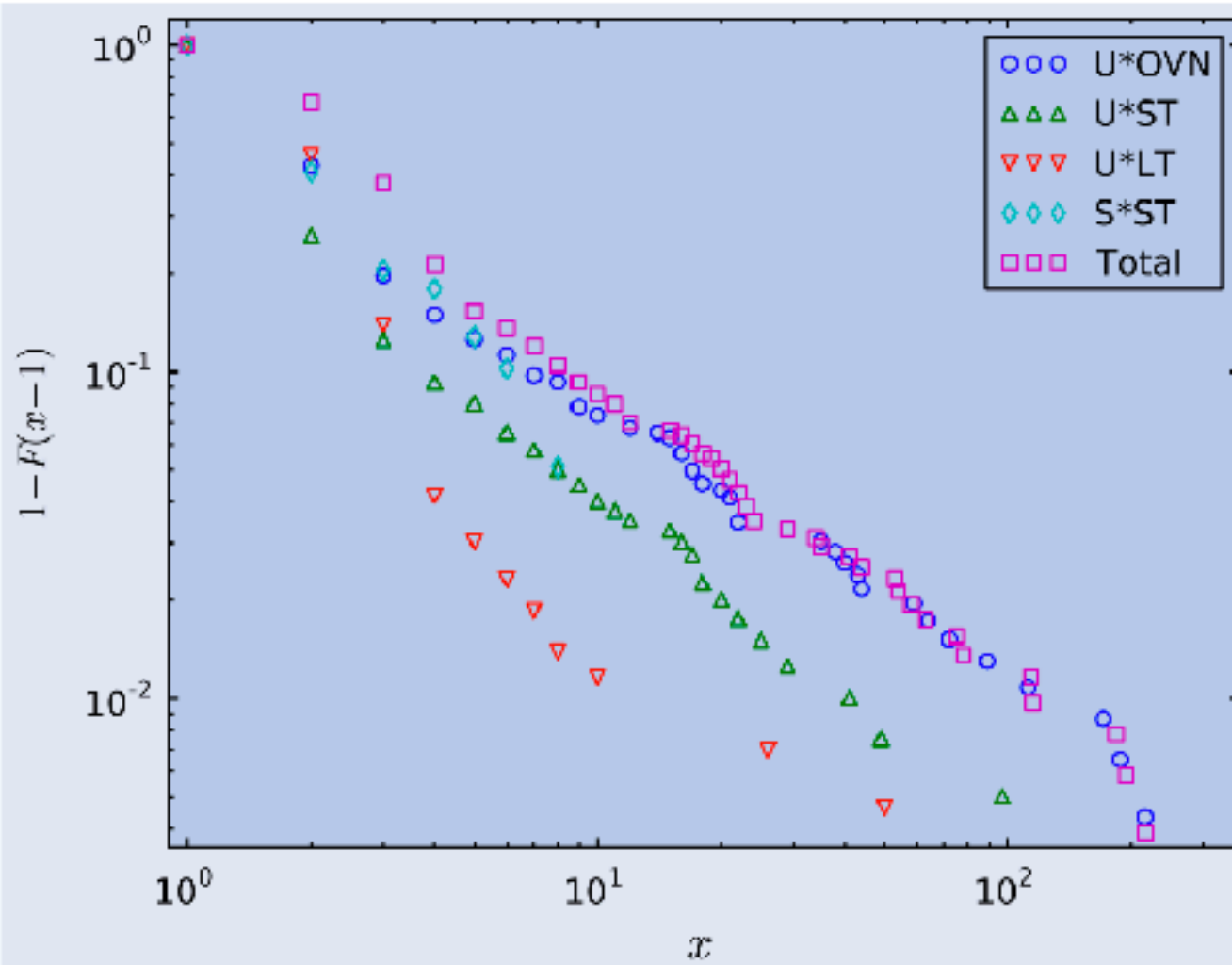
Binomial distribution

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

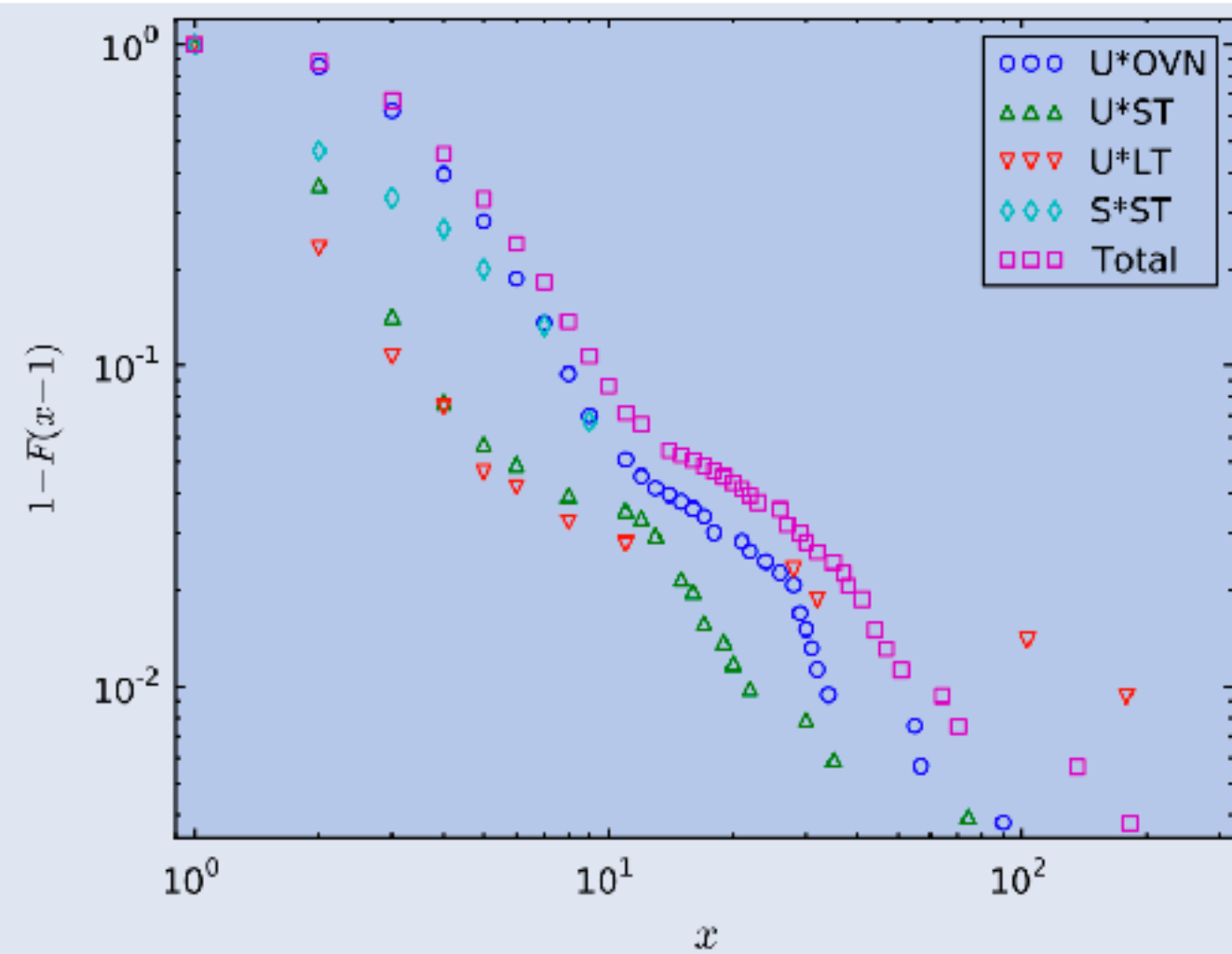


DEGREE DISTRIBUTION





(a) In-degree, 2012



(b) Out-degree, 2012

SHORT PATHS

assumption:

all nodes have degree k

Path length Nodes reached

$$\ell = 1 \quad k$$



SHORT PATHS

assumption:

all nodes have degree k

Path length	Nodes reached
-------------	---------------

$\ell = 1$	k
------------	-----

$\ell = 2$	$k(k - 1)$
------------	------------



SHORT PATHS

assumption:

all nodes have degree k

Path length	Nodes reached
-------------	---------------

$\ell = 1$	k
------------	-----

$\ell = 2$	$k(k - 1)$
------------	------------

$\ell = 3$	$k(k - 1)^2$
------------	--------------



SHORT PATHS

assumption:

all nodes have degree k

Path length	Nodes reached
-------------	---------------

$\ell = 1$	k
------------	-----

$\ell = 2$	$k(k - 1)$
------------	------------

$\ell = 3$	$k(k - 1)^2$
------------	--------------

$$k(k - 1)^{\ell - 1} \approx k^{\ell}$$

If k large

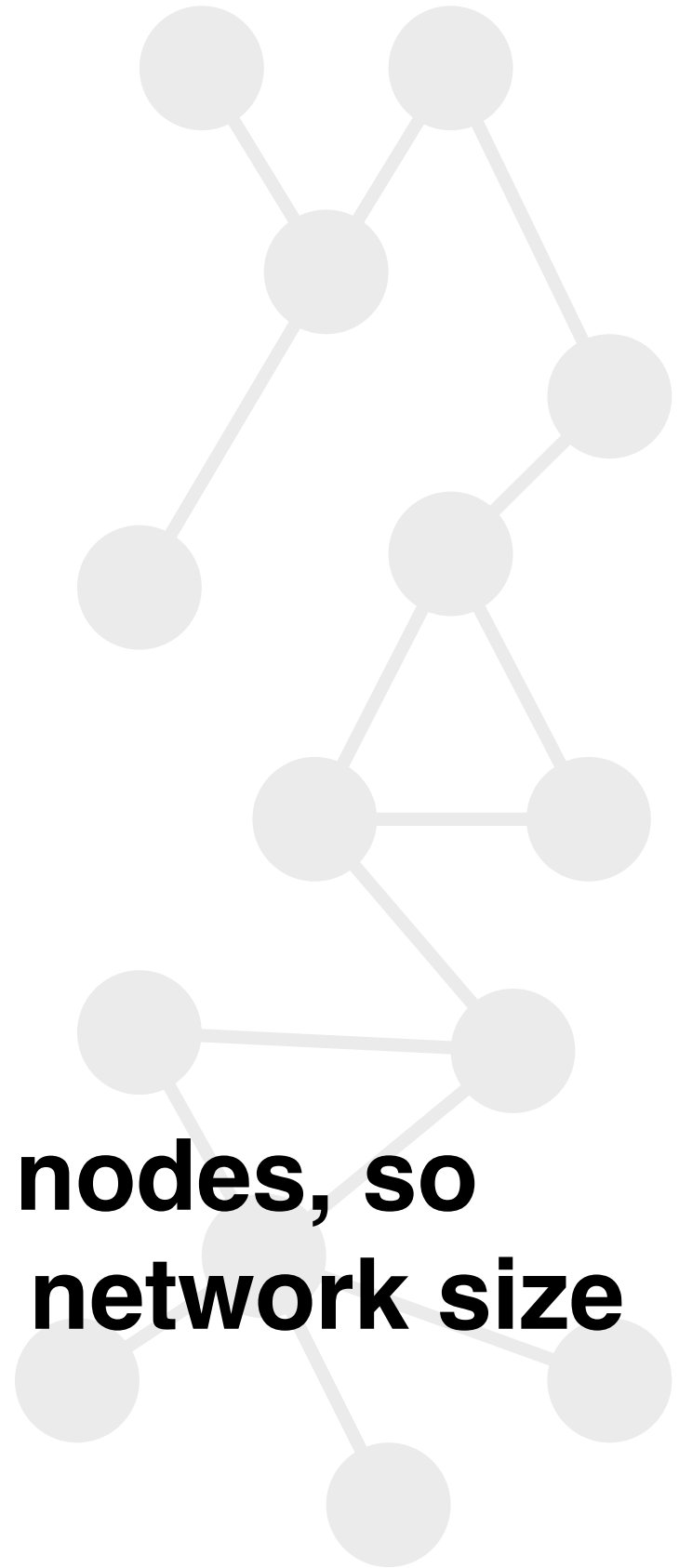


SHORT PATHS

$$k^{\ell_{max}} = N$$

$$\ell_{max} = \log_k N = \frac{\log N}{\log k}$$

Diameter grows with logarithm of nodes, so distances are fairly small despite the network size

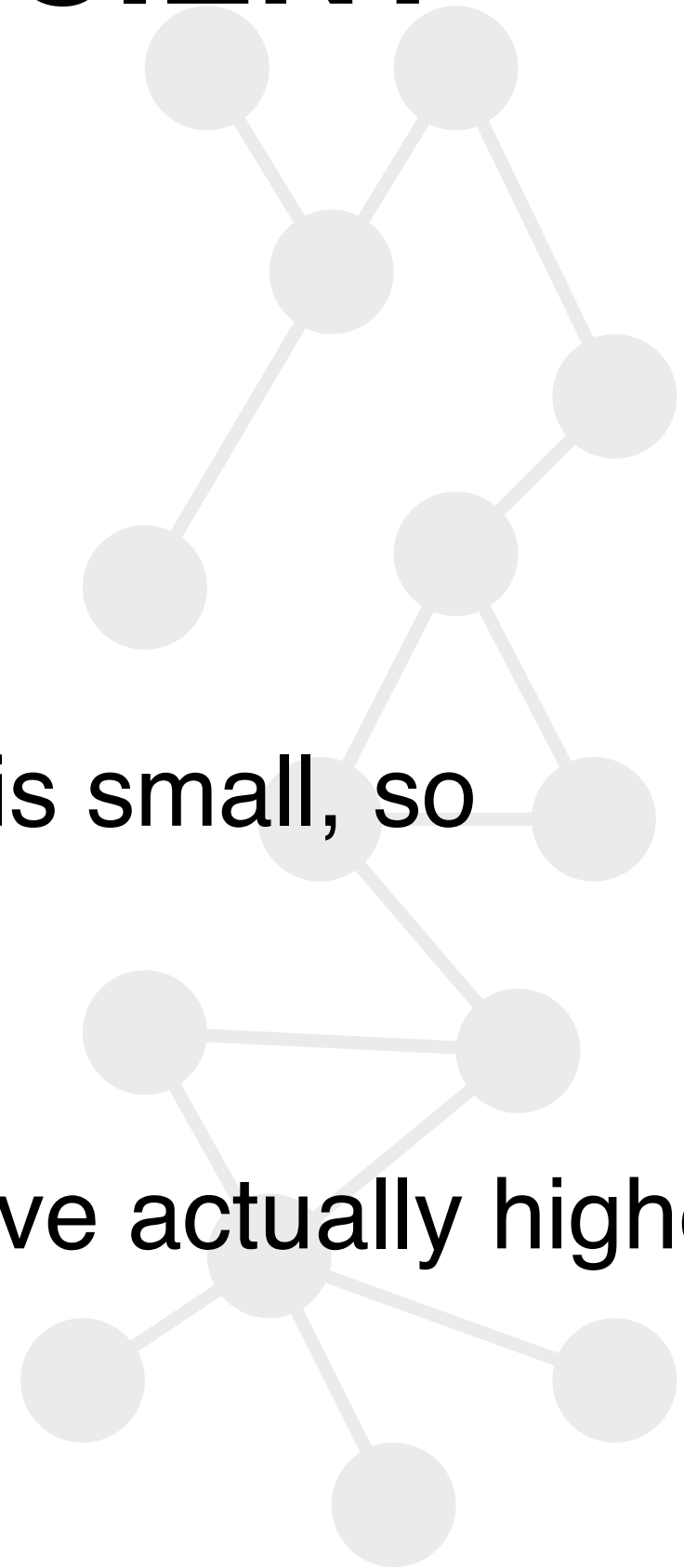


CLUSTERING COEFFICIENT

$$C(i) = p$$

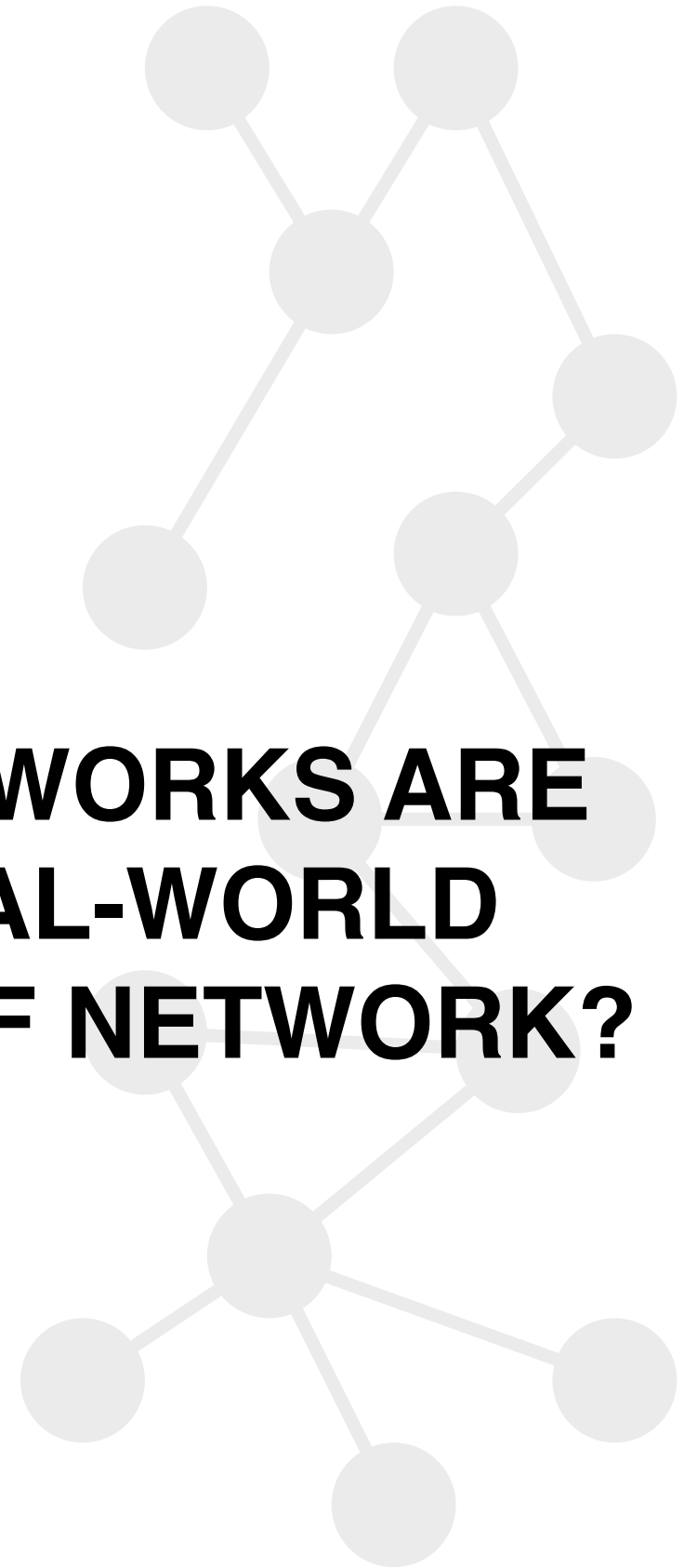
Real-world networks are **sparse**, so p is small, so **clustering is small**

However, many real-world networks have actually higher clustering coefficients



CONCLUSIONS

**DO YOU THINK ERDOS-RENYI NETWORKS ARE
GOOD APPROXIMATIONS OF REAL-WORLD
NETWORKS? IF SO, WHICH TYPE OF NETWORK?**



SMALL WORLDS



Proposed by **WATTS AND STROGATZ**
(sometimes called WS networks)

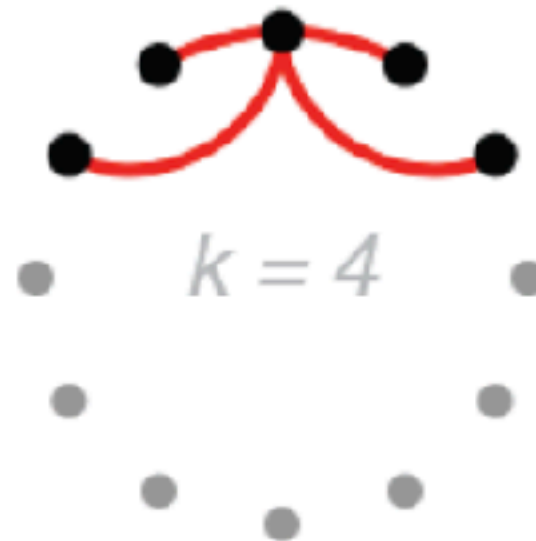
Solves the problem of low **clustering coefficients**

SMALL WORLDS

Start with n nodes



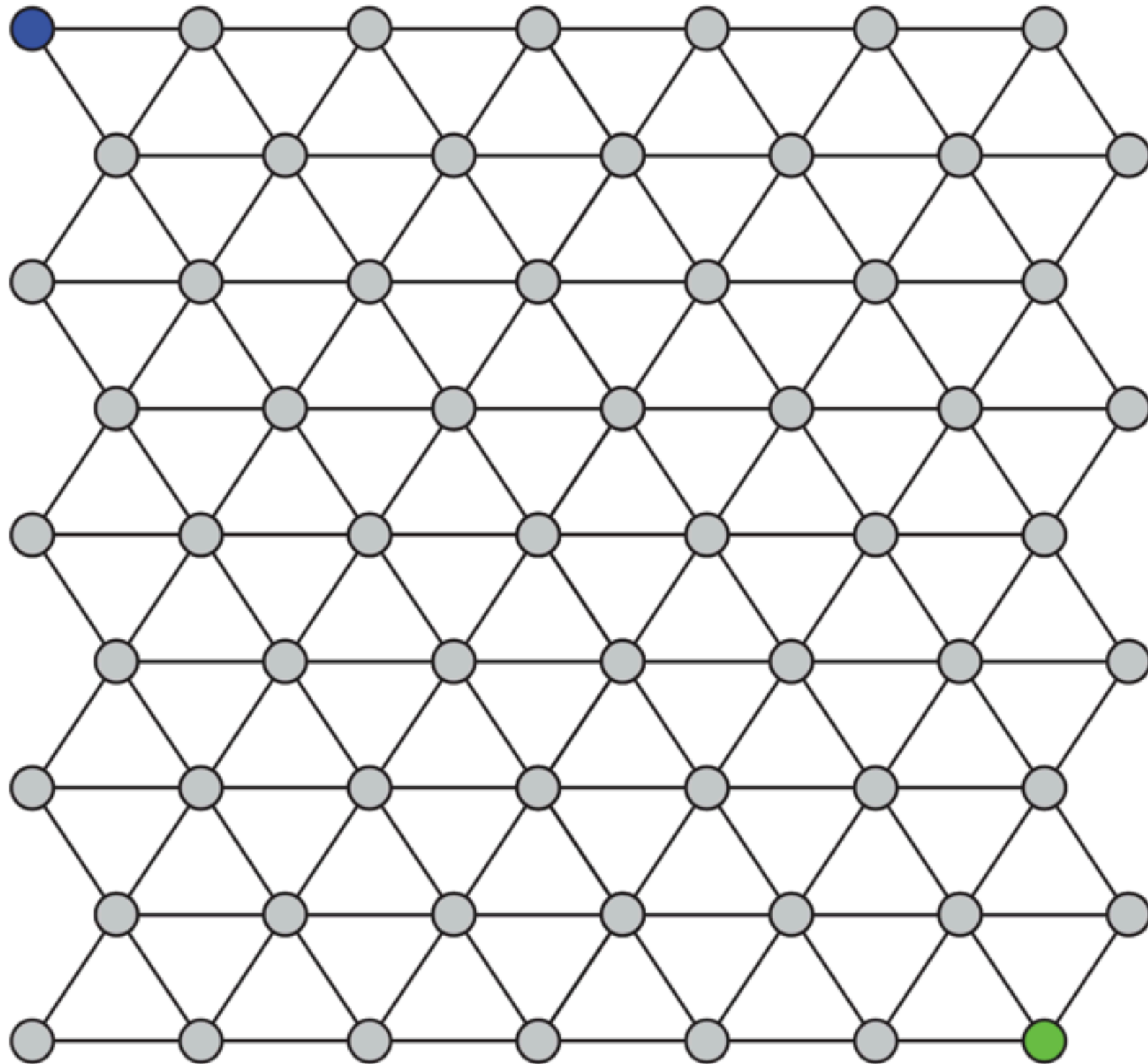
Connect a node to its
 k nearest neighbours



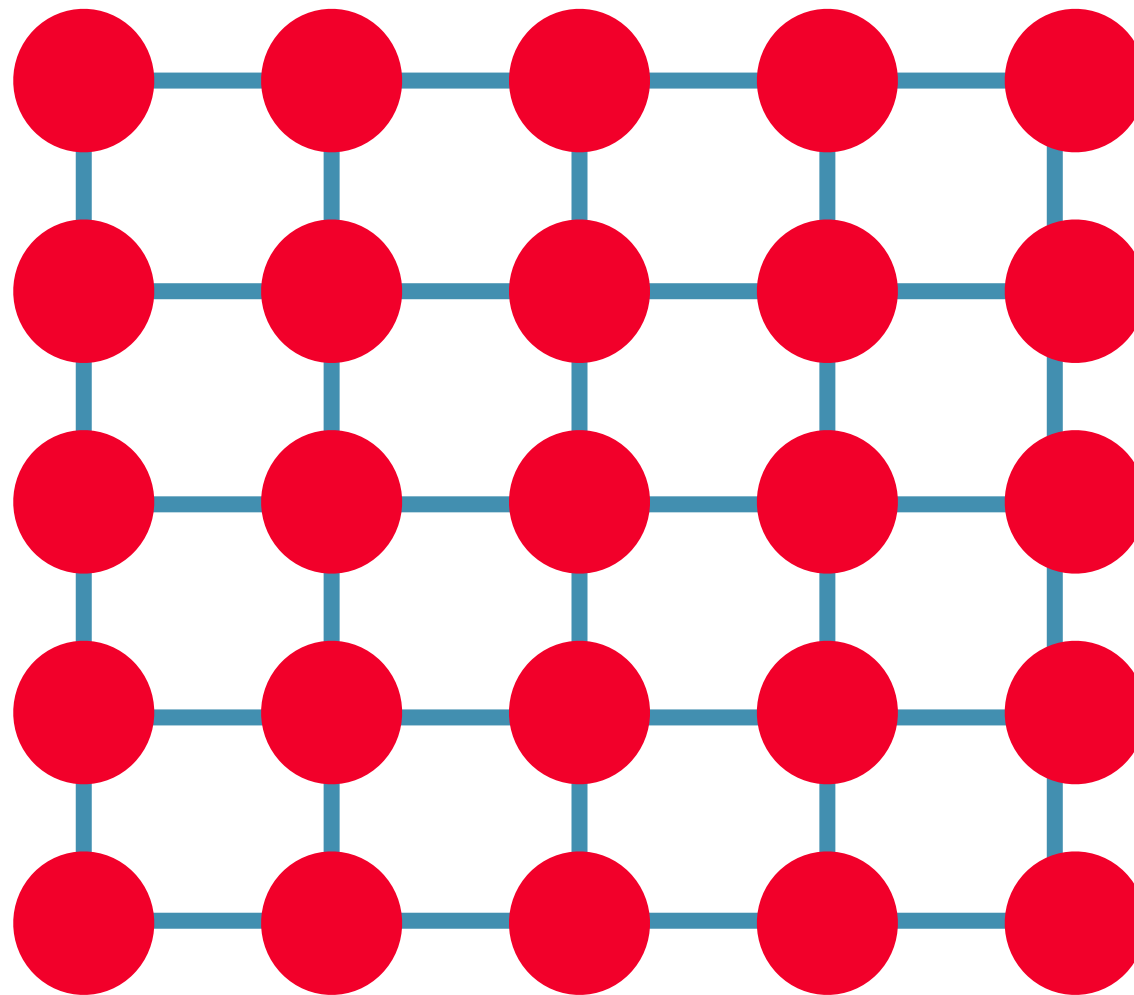
Do the same for all
the nodes in the network



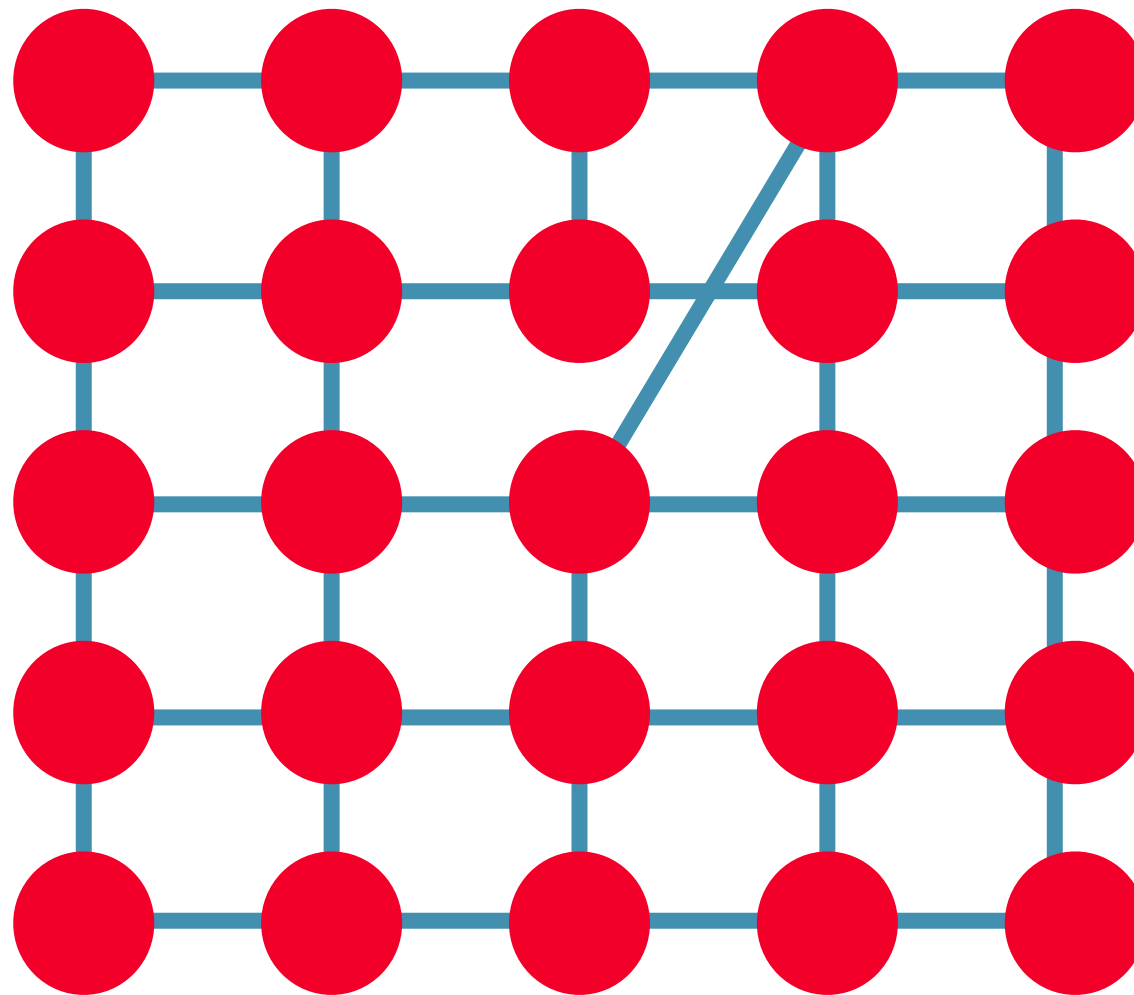
SMALL WORLDS



SMALL WORLDS

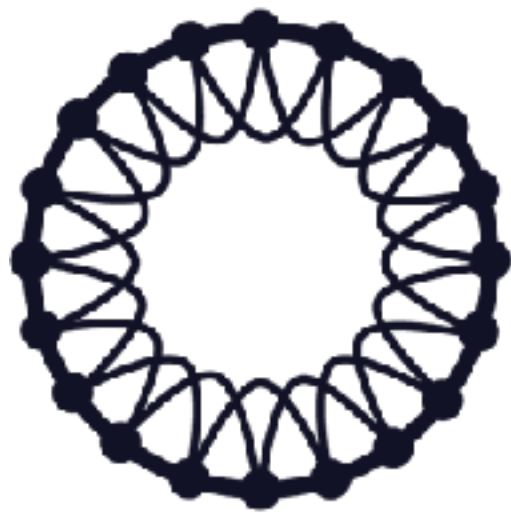


SMALL WORLDS



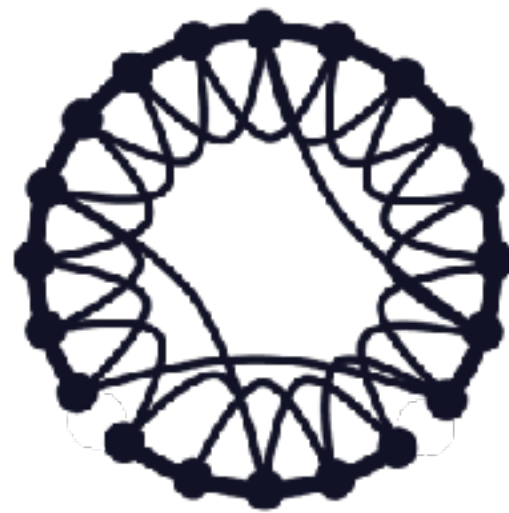
we rewire every link in the network with probability p

SMALL WORLDS



Regular

$p=0$



Small-world

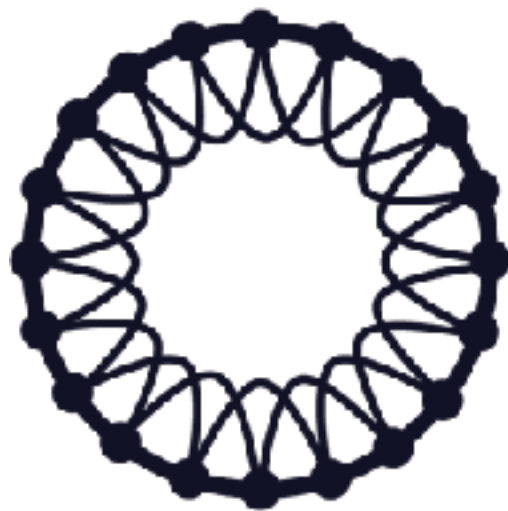


Random

$p=1$

Increasing Randomness

SMALL WORLDS



Regular

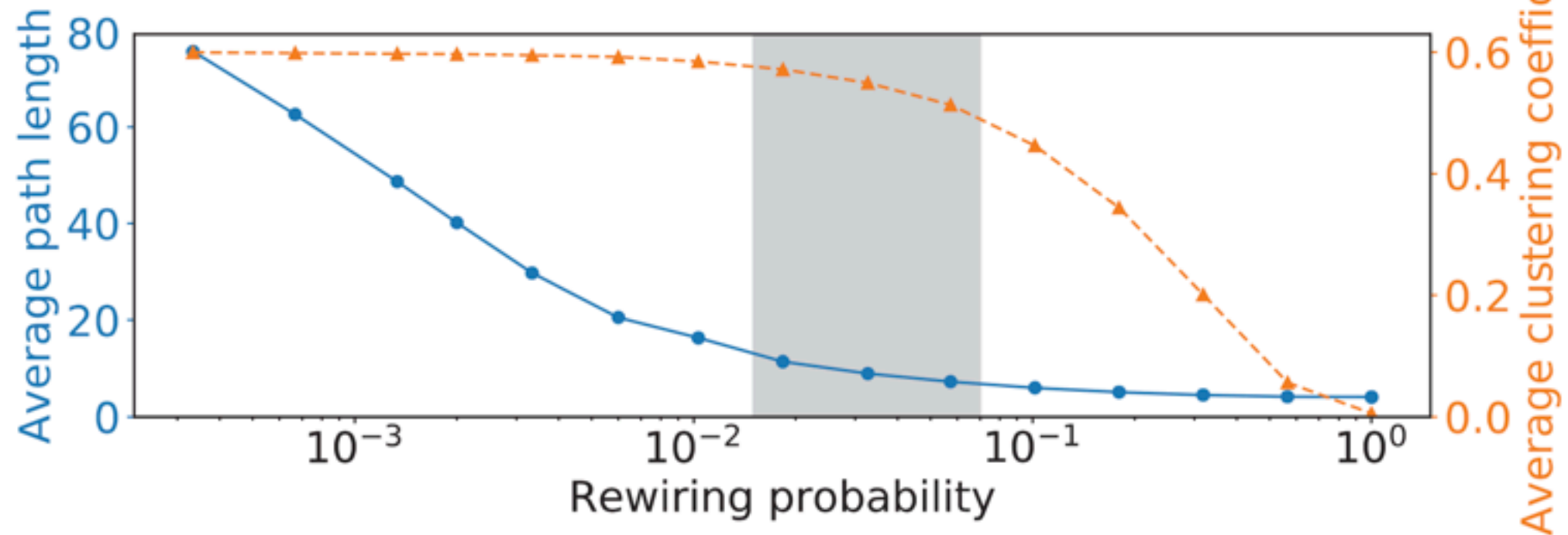


Small-world



Random

$p=0$  $p=1$
Increasing Randomness



SMALL WORLDS



Desirable properties such as clustering and short paths

What is missing?

SCALE-FREE NETWORKS

PROPOSED BY REKA **ALBERT** AND LASZLO **BARABASI** (ALSO KNOWN AS BA NETWORK)

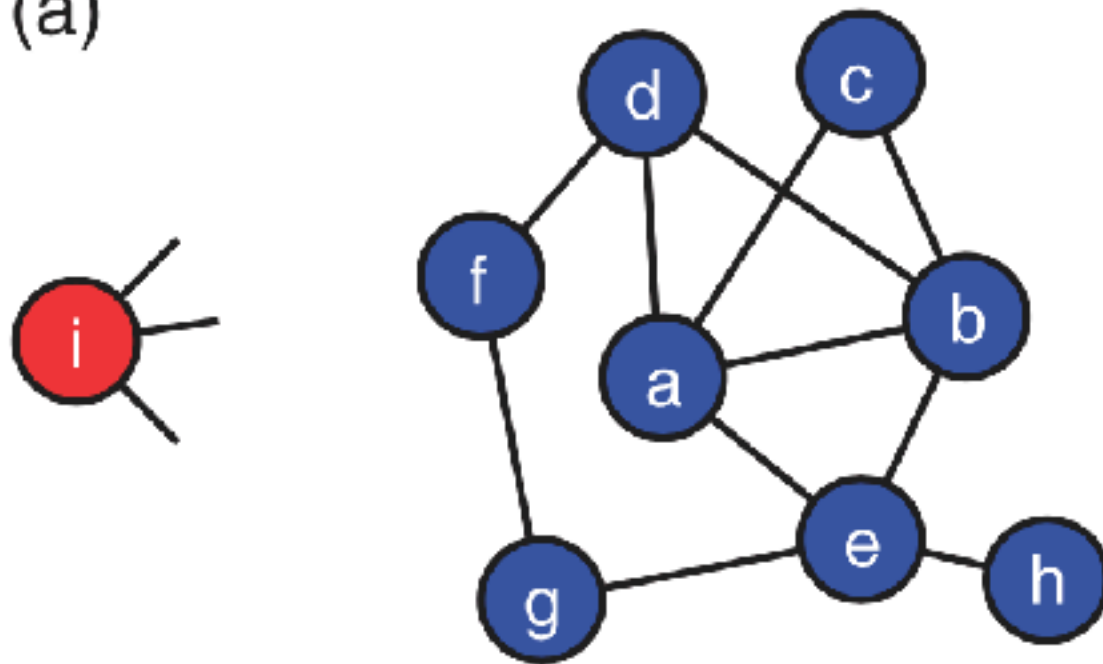
BASED ON THE IDEA OF **PREFERENTIAL ATTACHMENT**

INTRODUCES **DYNAMIC** GENERATION OF NETWORK, AND **EXPLAINS HUBS**

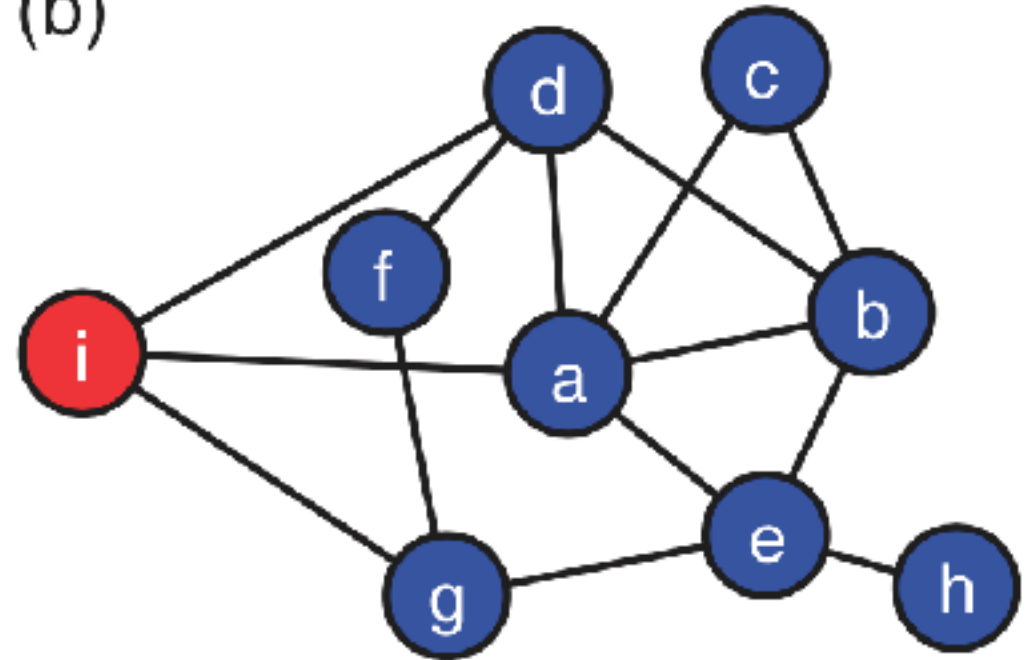


PREFERENTIAL ATTACHMENT

(a)



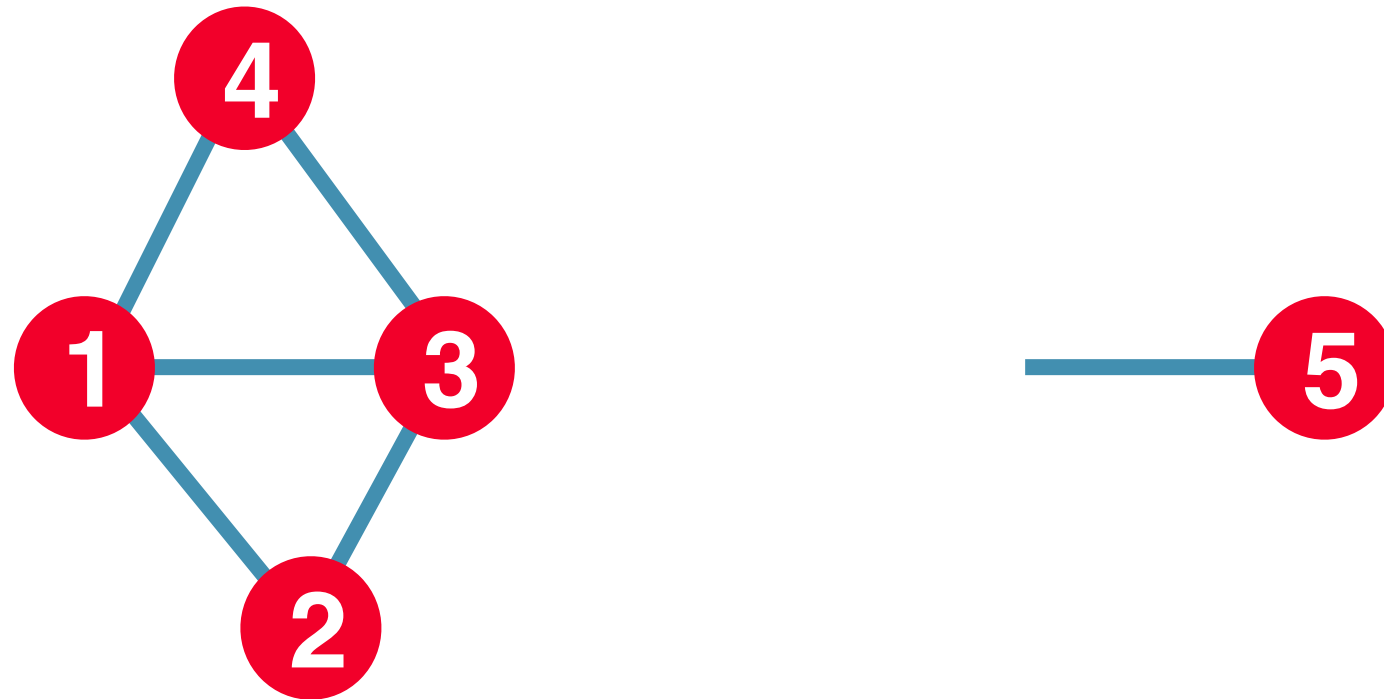
(b)



NEW NODES MAKES SOME CONNECTIONS

CONNECTIONS ARE MADE BASED
ON A GIVEN RULE (OR SET OF RULES)

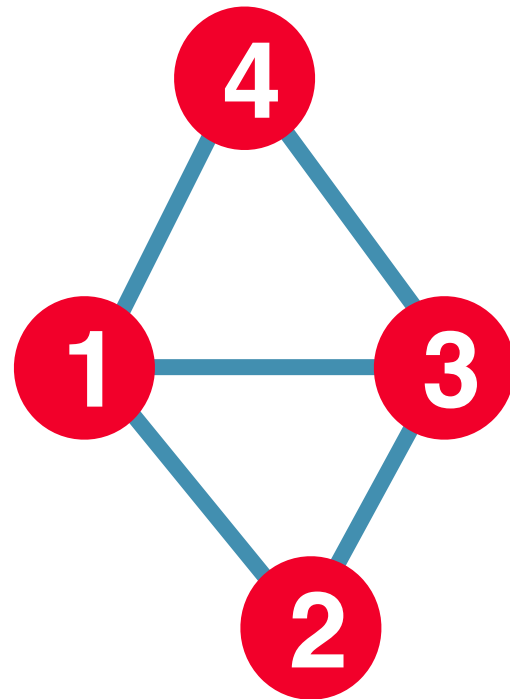
BARABASI-ALBERT MODEL



$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}$$

Probability a new node makes a connection to node i

BARABASI-ALBERT MODEL



$$\Pi(2) = \frac{2}{2 + 2 + 3 + 3} = \frac{1}{5}$$

For nodes 2 and 4

$$\Pi(3) = \frac{3}{2 + 2 + 3 + 3} = \frac{3}{10}$$

For nodes 1 and 3

BARABASI-ALBERT MODEL

Parameters:

number of initial nodes

$$m_0$$

Number of links of new nodes

$$m \leq m_0$$

Number of total nodes at the end of the process N

Algorithm:

For each new node i , with i from m_0 to: N

 Compute the probability of every node to get attached to i based on degree

 Choose m nodes randomly based on this probability

 form link between the new node and the chosen nodes

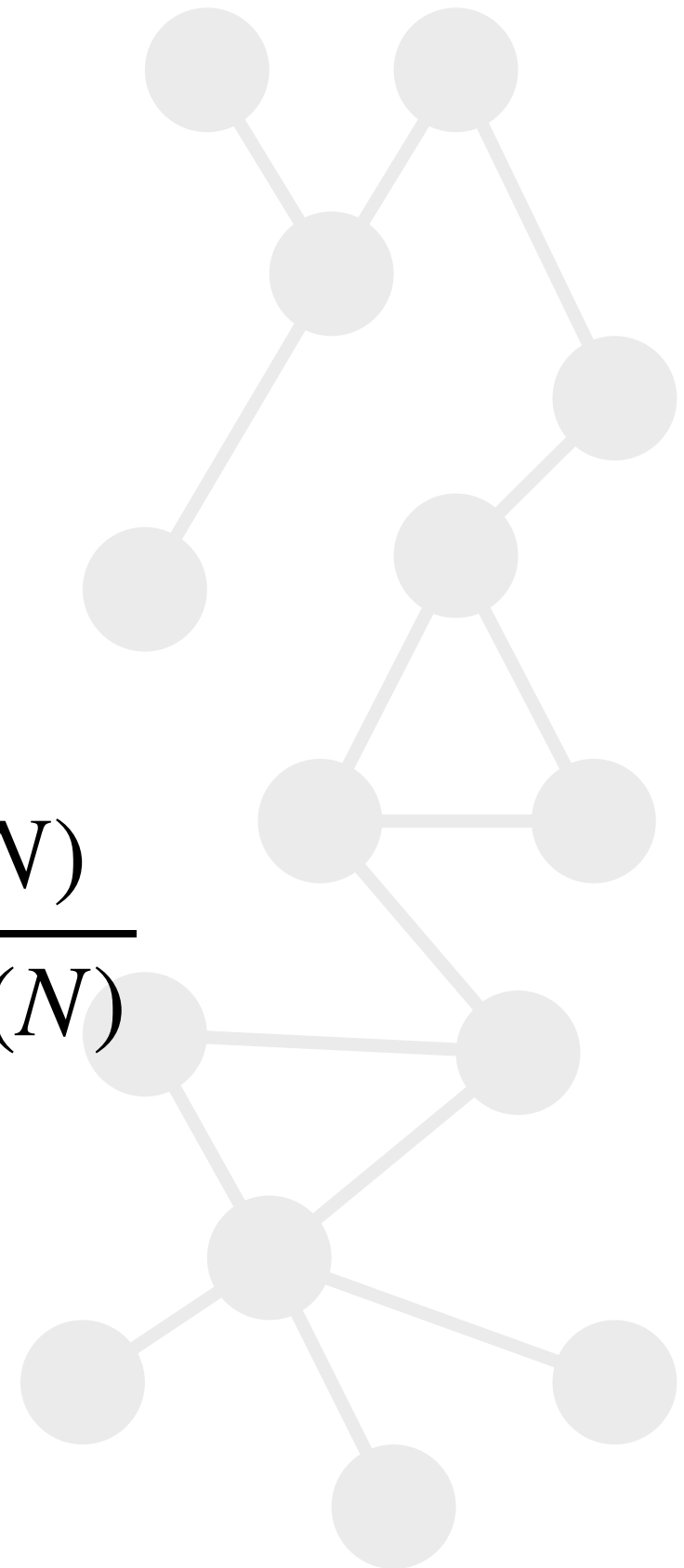
METRICS

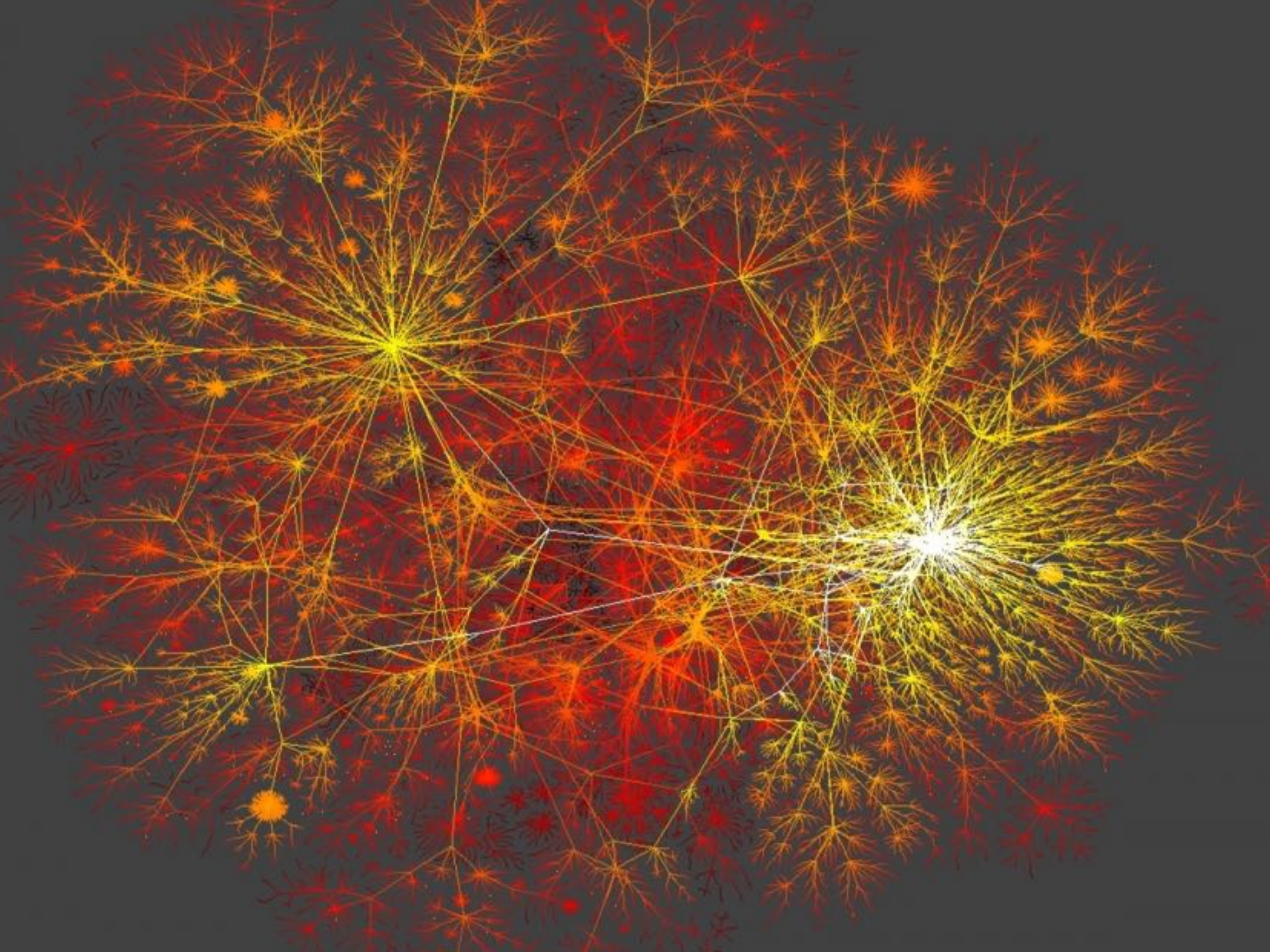
Parameters:

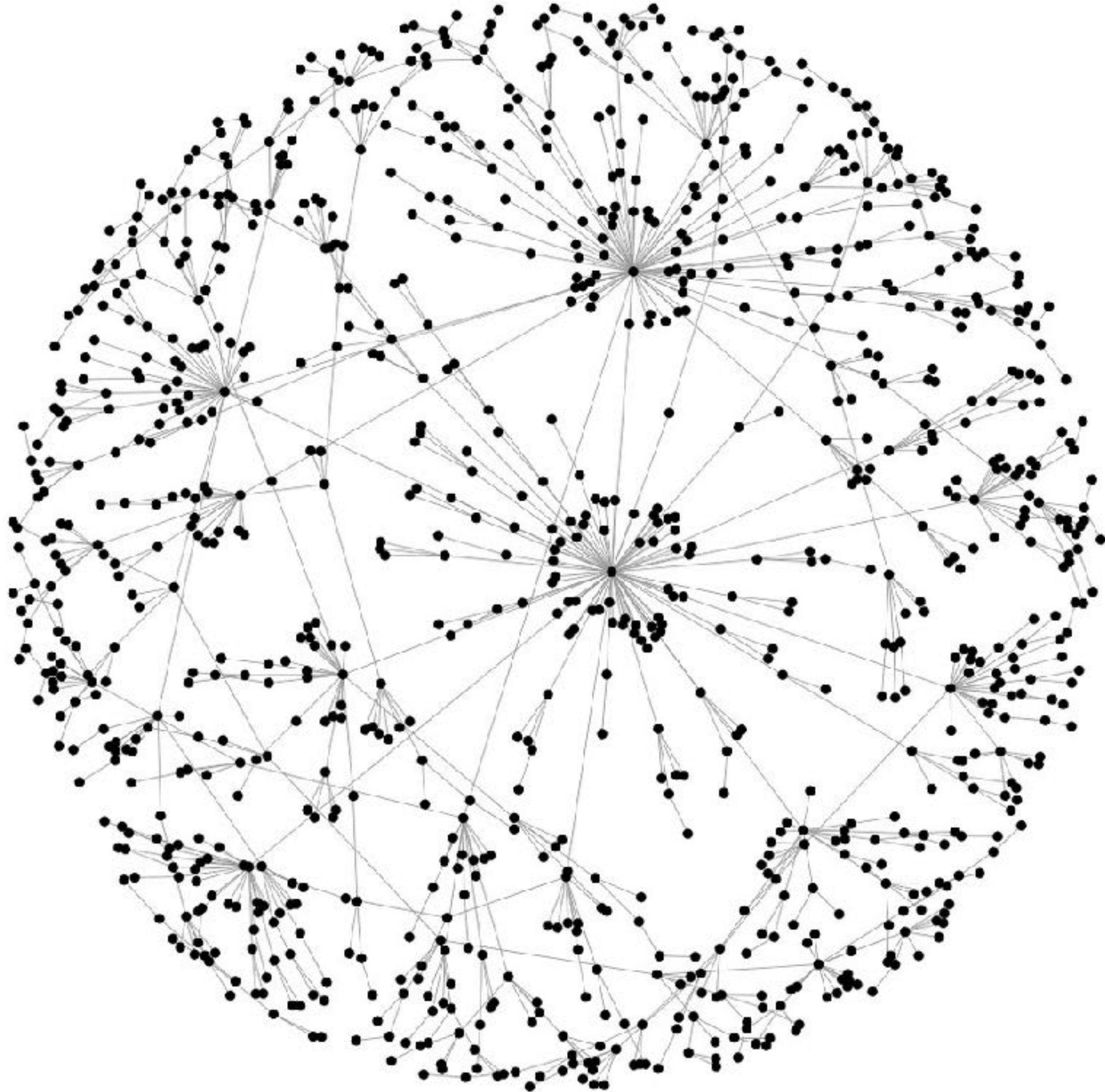
Degree distribution $P(k) \approx k^{-3}$

Clustering coefficient $C(k) \approx k^{-1}$

Diameter $\ell(k) \approx \frac{\ln(N)}{\ln \ln(N)}$







SUMMARY

We can **generate networks** with algorithms

Most **real-world networks** are compatible with one of these models

We can **compare the properties** of a real-world network with that of a model to understand how they are formed

