Randomly Walking a fine line

Flavio Chierichetti
Sapienza University

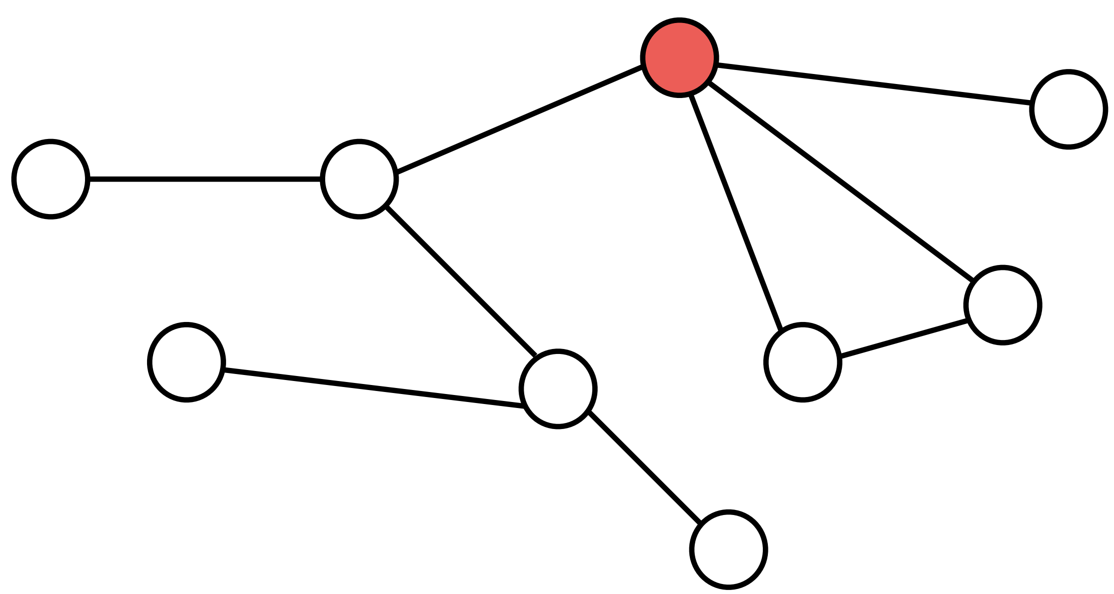
Joint with subsets of
Marco Bressan, Anirban Dasgupta, Shahrzad Haddadan, Ravi Kumar, Silvio Lattanzi, Stefano Leucci, Alessandro Panconesi, Tamás Sarlós

Random Walks

- Random Walks are a very useful tool for studying online and offline large-scale graphs

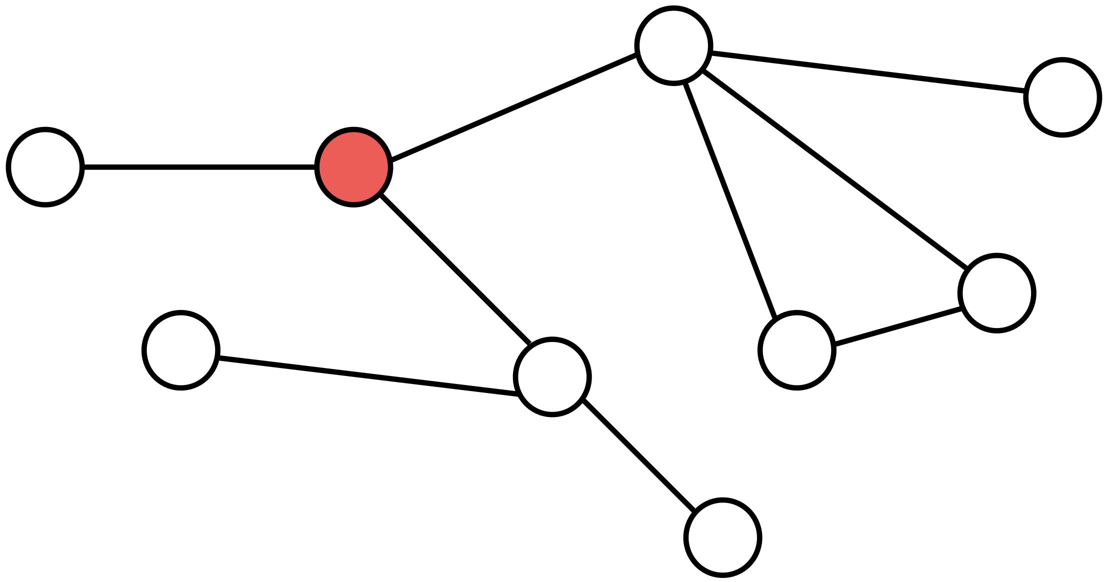
Random Walks

- Random Walks are a very useful tool for studying online and offline large-scale graphs



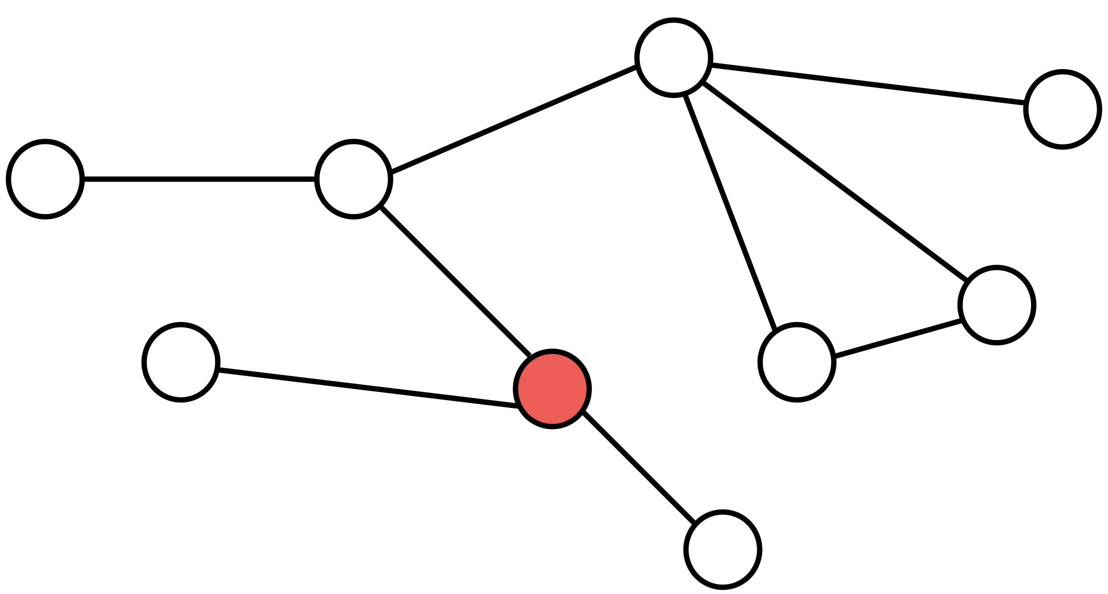
Random Walks

- Random Walks are a very useful tool for studying online and offline large-scale graphs



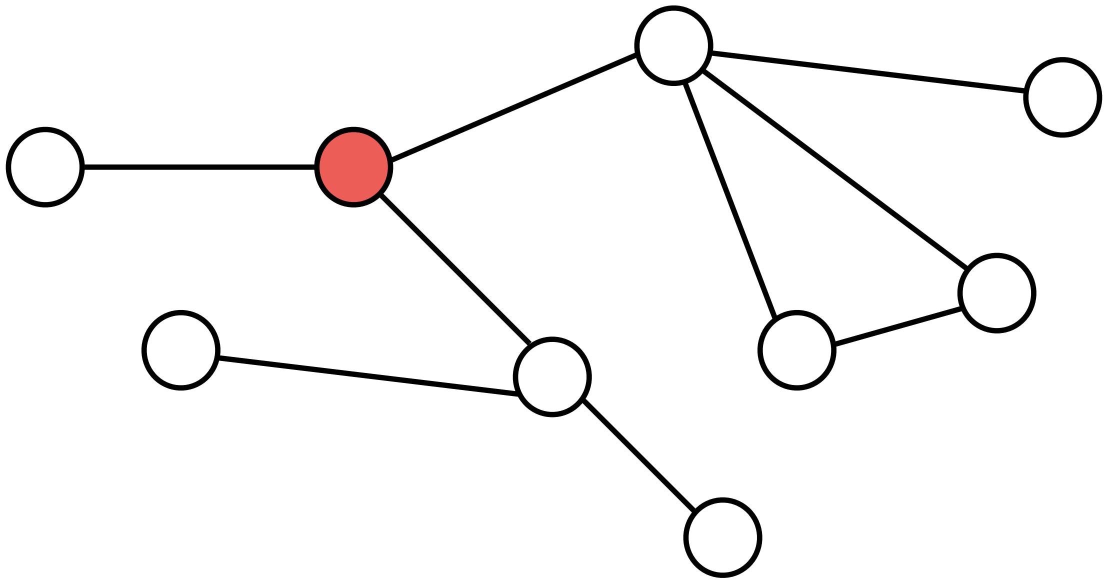
Random Walks

- Random Walks are a very useful tool for studying online and offline large-scale graphs



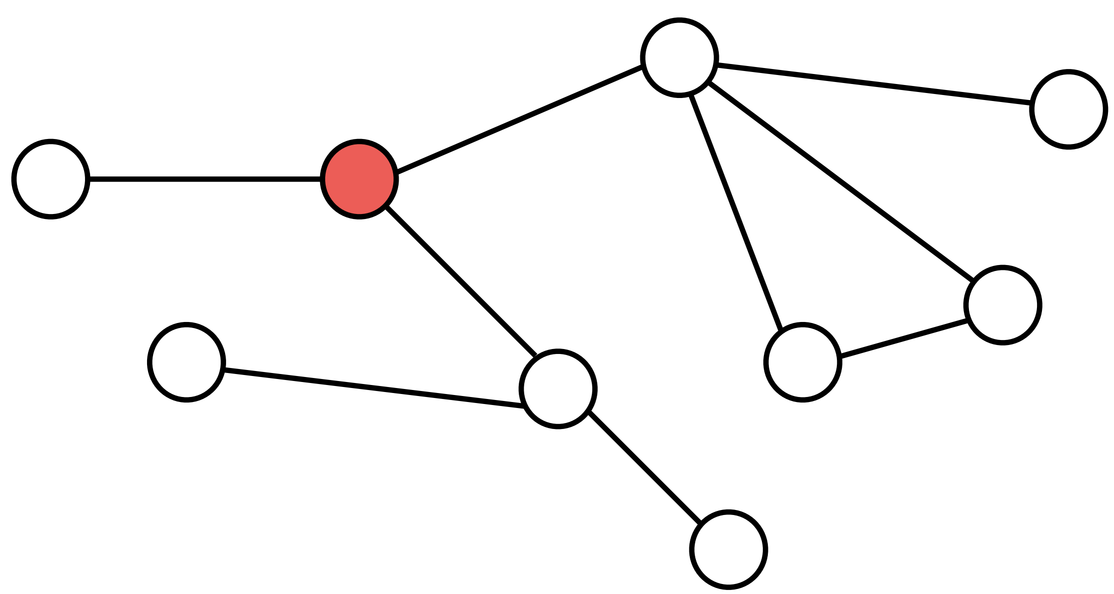
Random Walks

- Random Walks are a very useful tool for studying online and offline large-scale graphs



Random Walks

- Random Walks are a very useful tool for studying online and offline large-scale graphs:
  easy to implement,
  small memory footprint



Random Walks

- Random Walks are a very useful tool for studying online and offline large-scale graphs:
  easy to implement,
  small memory footprint

- Random Walks should be used carefully, if one aims for statistically meaningful results

Random Walks

- Random Walks are a very useful tool for studying online and offline large-scale graphs:
  easy to implement,
  small memory footprint

- Random Walks should be used carefully, if one aims for statistically meaningful results

- In this talk, we will be considering two examples…

Picking *Uniform-at-Random* users from a Social Network

Learning Average Opinions

Learning Average Opinions

Learning Average Opinions

Learning Average Opinions

What is the fraction of ?

Learning Average Opinions

What is the fraction of  ?

Asking all the users is too costly!

Learning Average Opinions

Select some people uniformly-at-random and ask them their opinion

Learning Average Opinions

Select some people uniformly-at-random and ask them their opinion

Learning Average Opinions

Select some people uniformly-at-random and ask them their opinion

Learning Average Opinions

Select some people uniformly-at-random and ask them their opinion

Learning Average Opinions

Select some people uniformly-at-random and ask them their opinion

Learning Average Opinions

Select some people uniformly-at-random and ask them their opinion

The empirical fraction of is provably close to the real fraction!

Learning Average Opinions

Learning Average Opinions

Learning Average Opinions

Learning Average Opinions

Learning Average Opinions

Select some people uniformly-at-random and ask them their opinion

Learning Average Opinions

Select some people uniformly-at-random and ask them their opinion

The empirical average is provably close to the real average

How do we select uniform-at-random profiles in a Social Network?

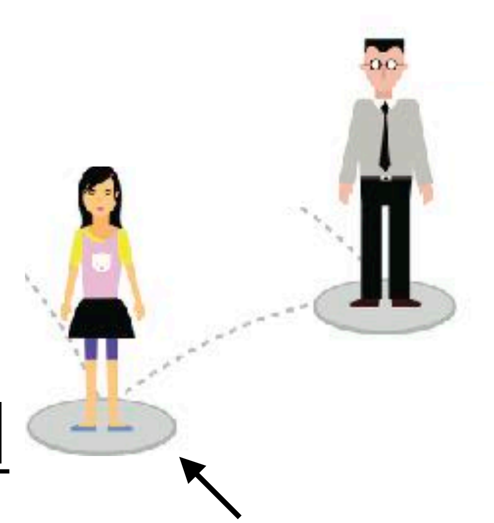
- We can access the SN through a crawling process.

http://s-n.com/001.html

How do we select uniform-at-random profiles in a Social Network?

- We can access the SN through a crawling process.

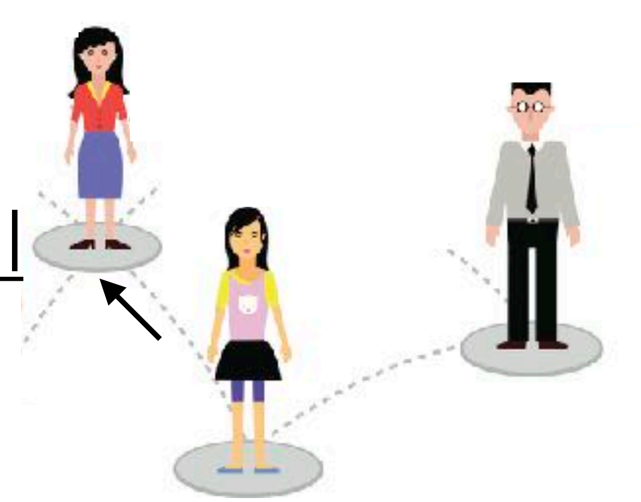
http://s-n.com/005.html



How do we select uniform-at-random profiles in a Social Network?

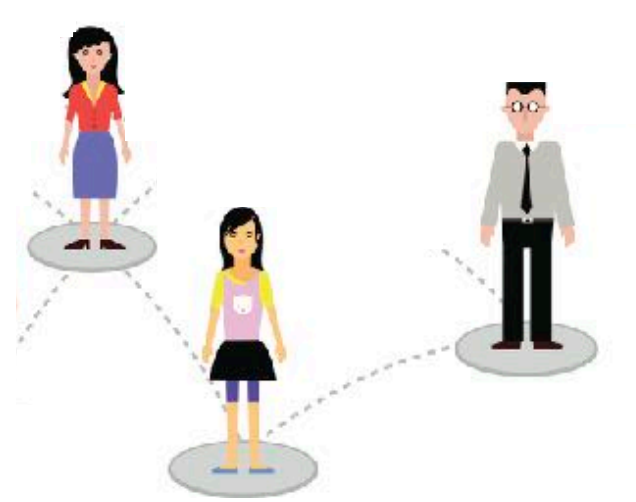
- We can access the SN through a crawling process.

http://s-n.com/011.html

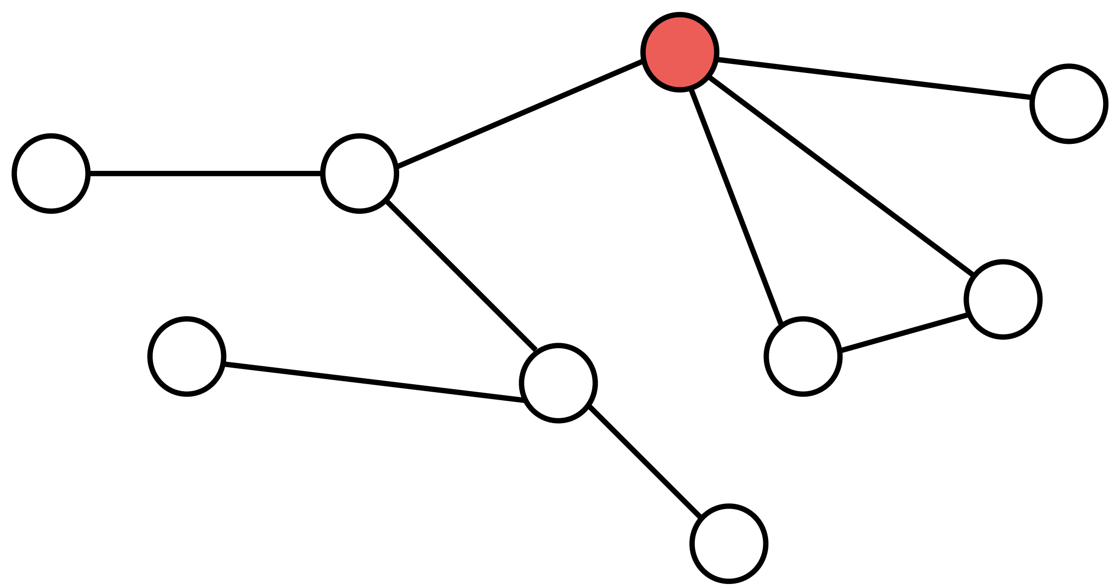


How do we select uniform-at-random profiles in a Social Network?

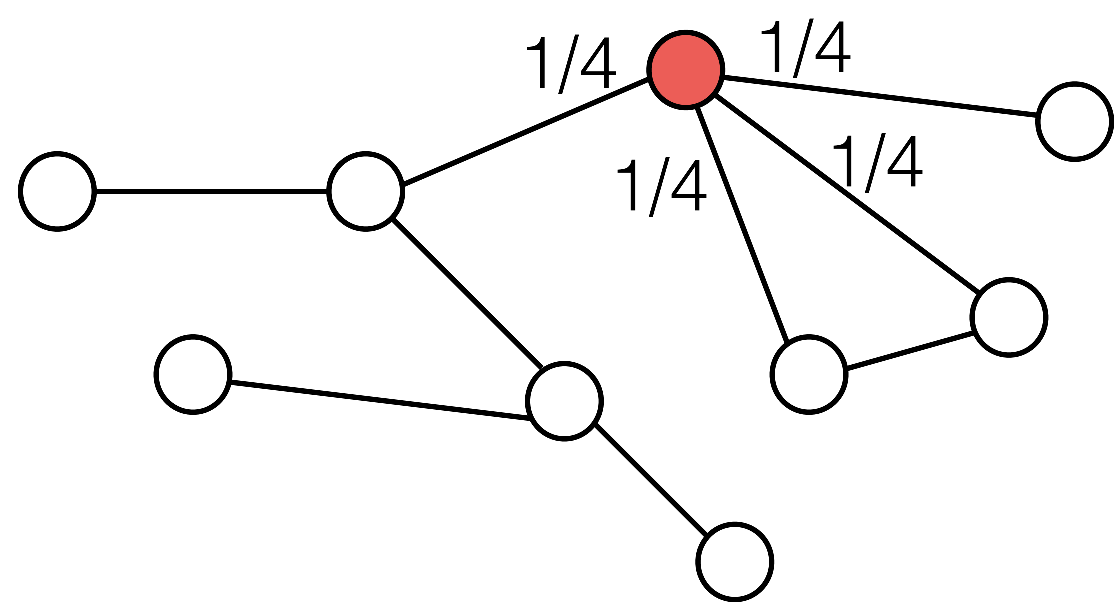
- We can access the SN through a crawling process.
- But we **cannot** crawl the whole network.
  Then, what can we do?



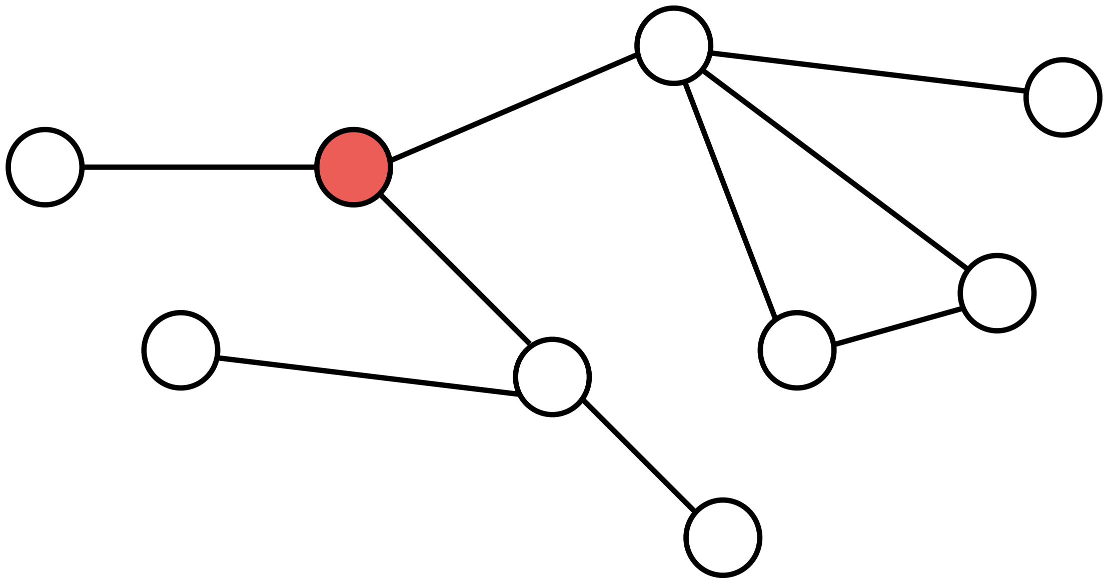
Random Walks



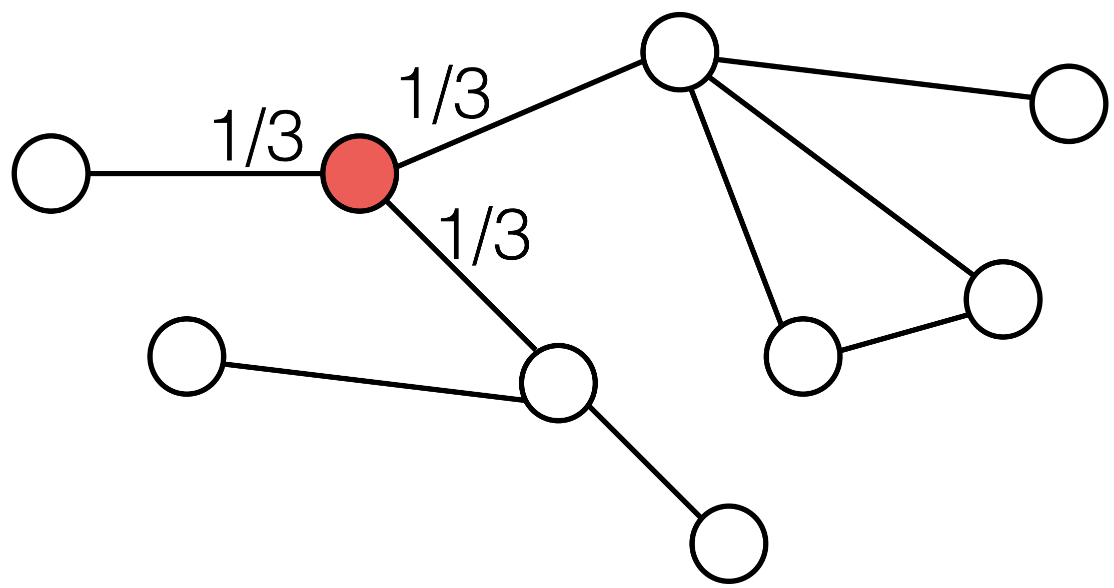
Random Walks



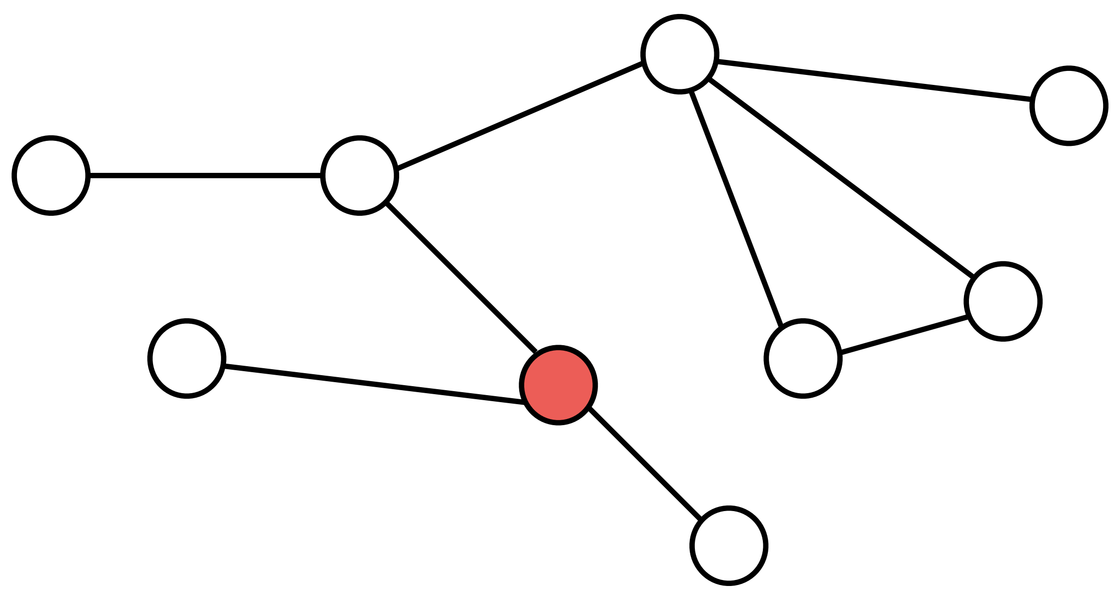
Random Walks



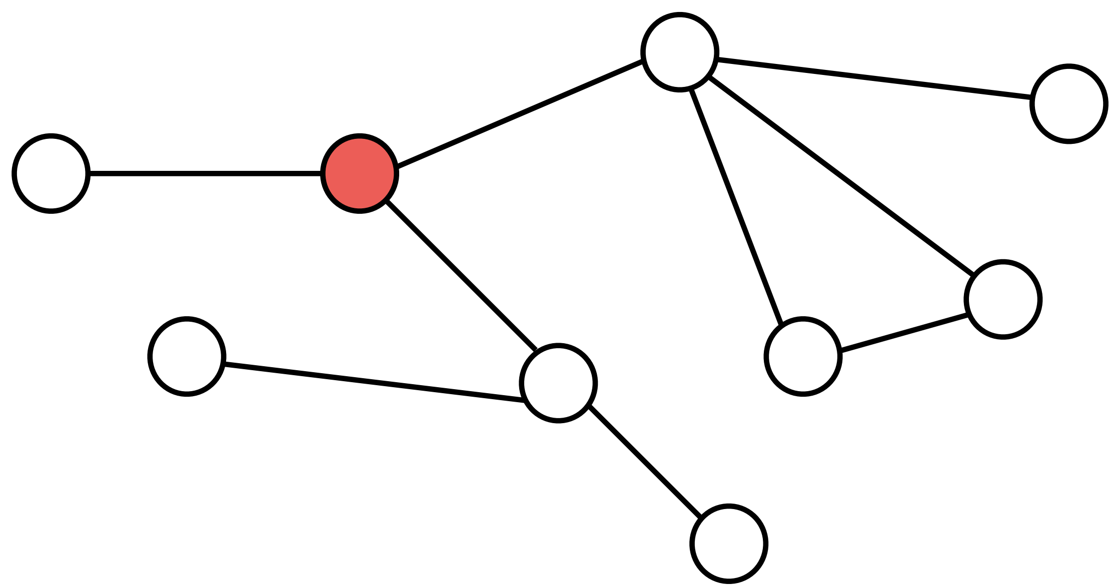
Random Walks



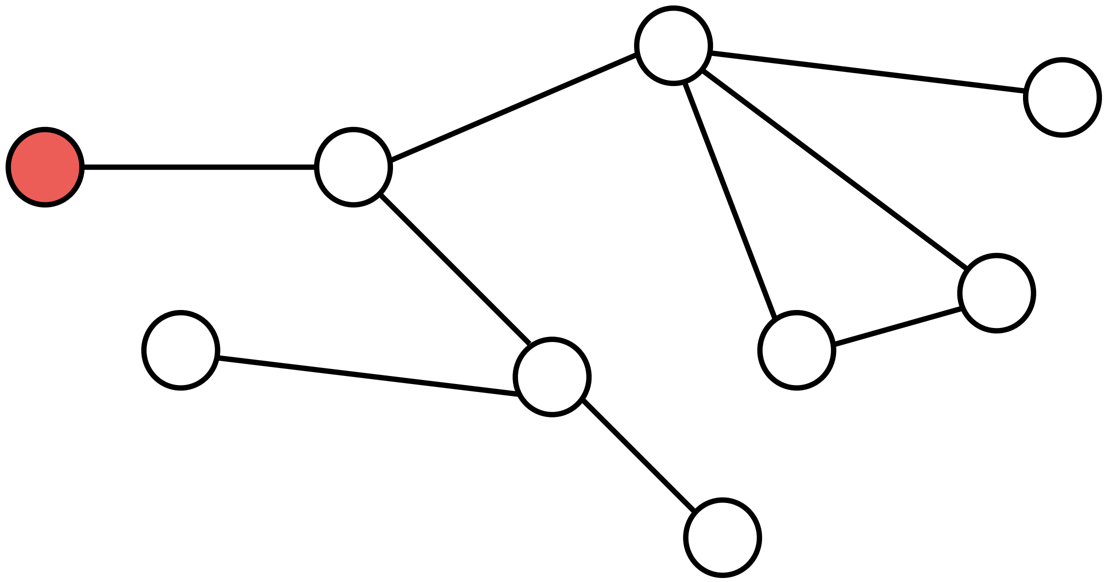
Random Walks



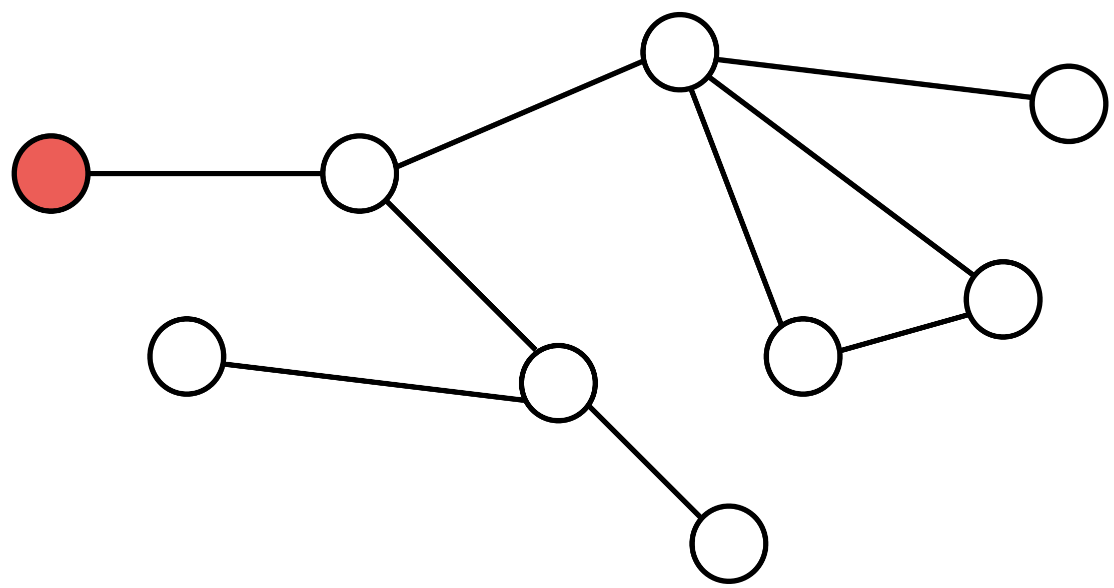
Random Walks



Random Walks

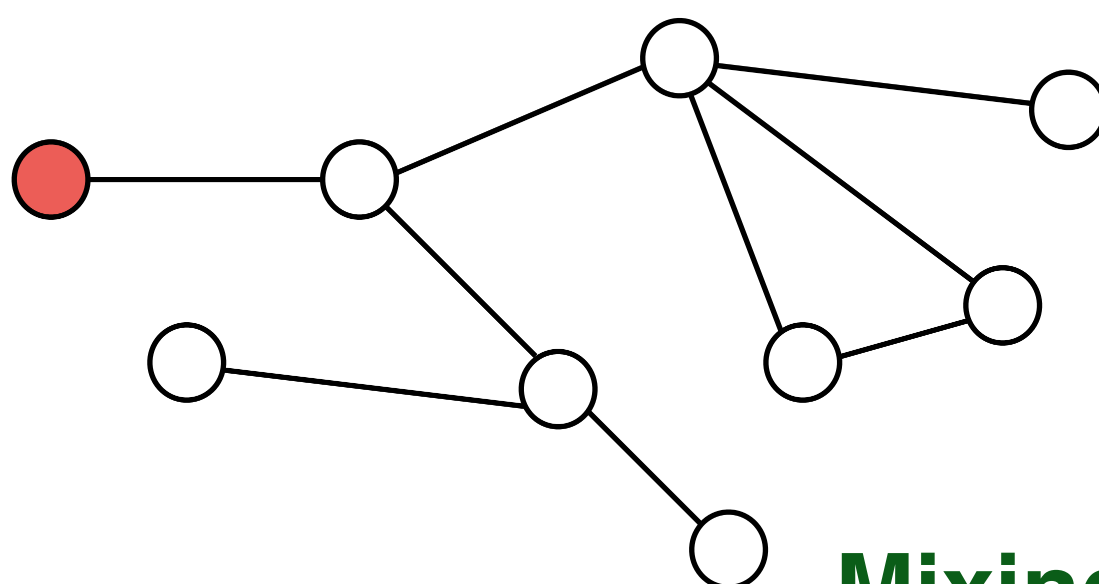


Random Walks



If the process goes on for enough many steps,
the random node it ends up on will be “random”

Random Walks

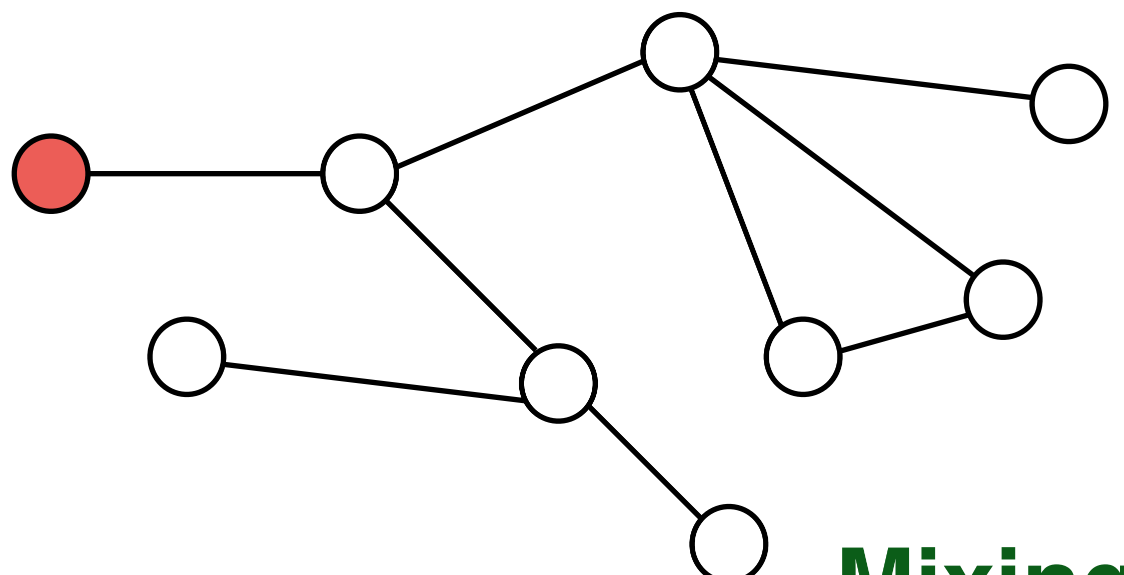


Mixing Time MT(G)

If the process goes on for enough many steps,
the random node it ends up on will be “random”

Random Walks

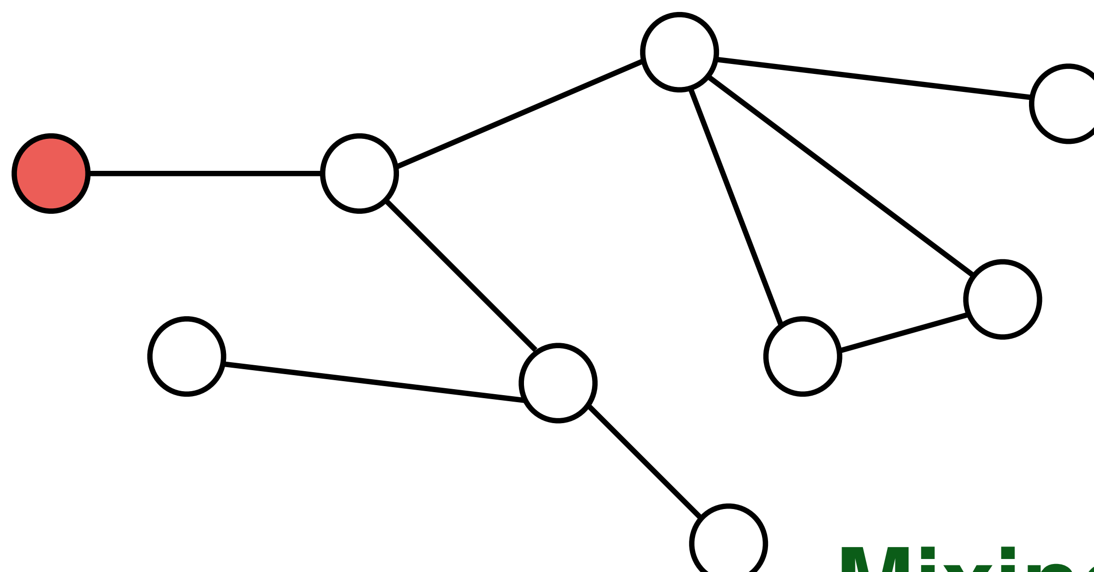
The Mixing Times of many “Social Networks” are small
[Leskovec et al, ’08]



Mixing Time MT(G)

If the process goes on for enough many steps,
the random node it ends up on will be “random”

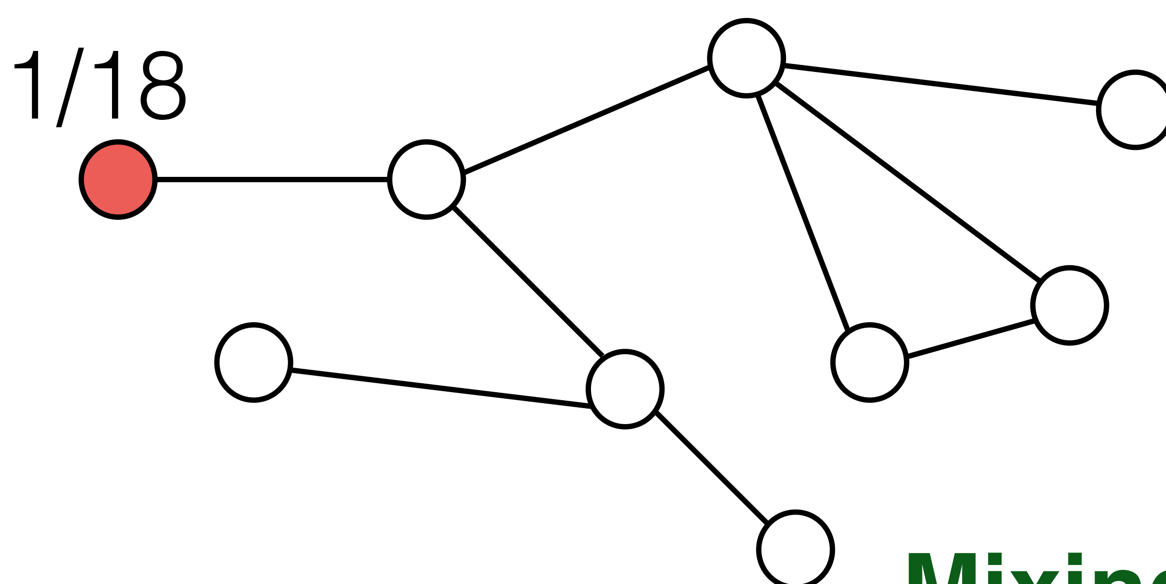
Random Walks



Mixing Time MT(G)

If the process goes on for enough many steps, the random node it ends up on will be "random", *chosen with probability proportional to its degree*

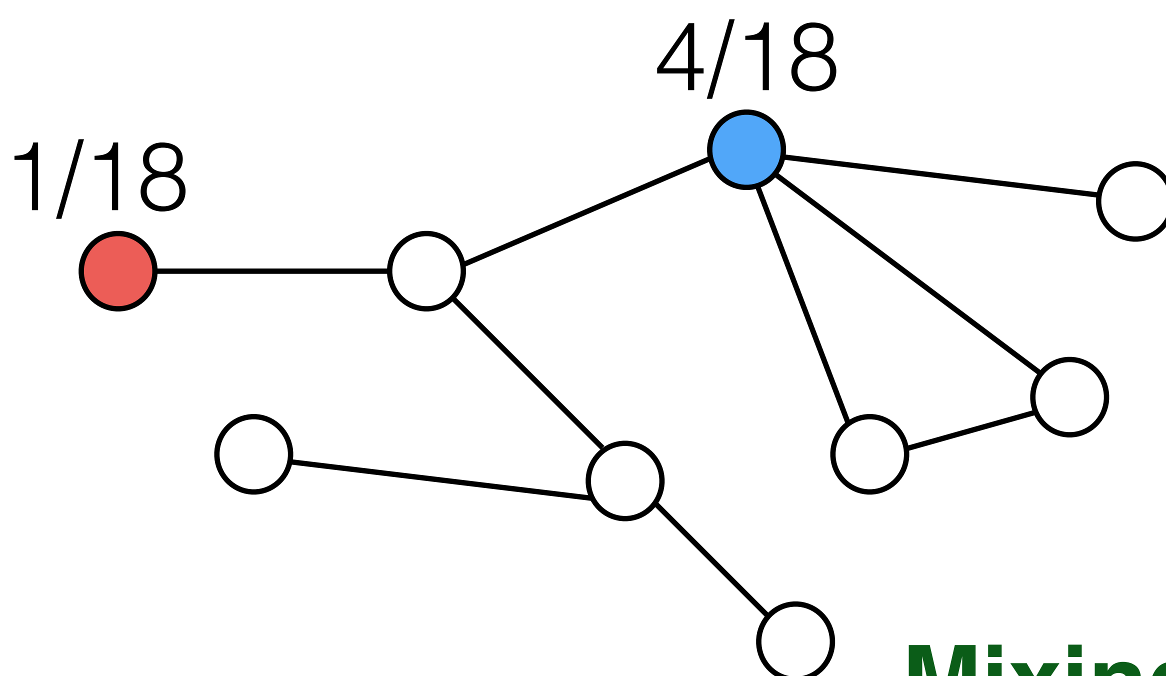
Random Walks



Mixing Time MT(G)

If the process goes on for enough many steps,
the random node it ends up on will be "random",
chosen with probability proportional to its degree

Random Walks



Mixing Time MT(G)

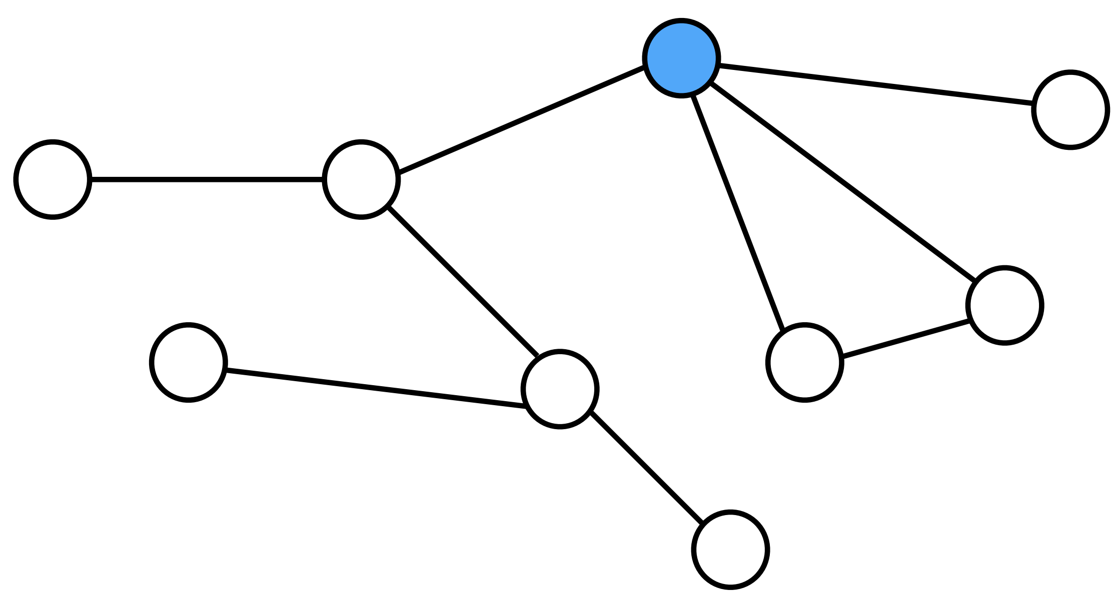
If the process goes on for enough many steps, the random node it ends up on will be "random", *chosen with probability proportional to its degree*

A Folklore Algorithm

- **While** True:
  - run the random walk for MT(G) steps;
  - suppose it ends on the node v;
  - **return** v with probability 1/deg(v).

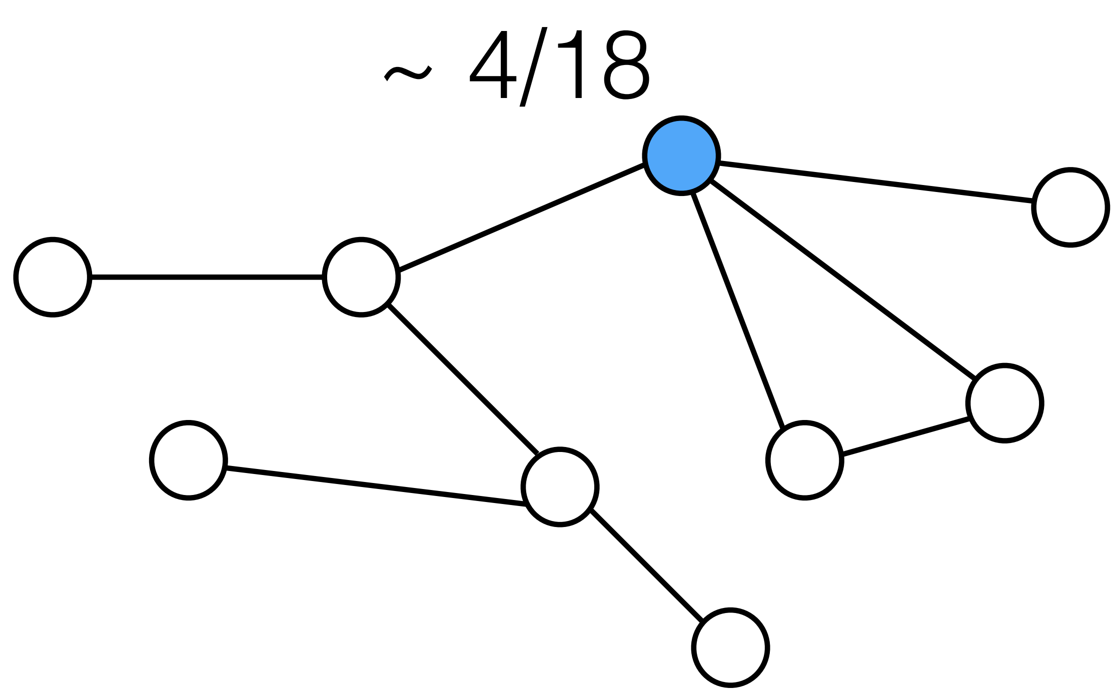
A Folklore Algorithm

- **While** True:
  - run the random walk for MT(G) steps;
  - suppose it ends on the node v;
  - **return** v with probability 1/deg(v).



A Folklore Algorithm

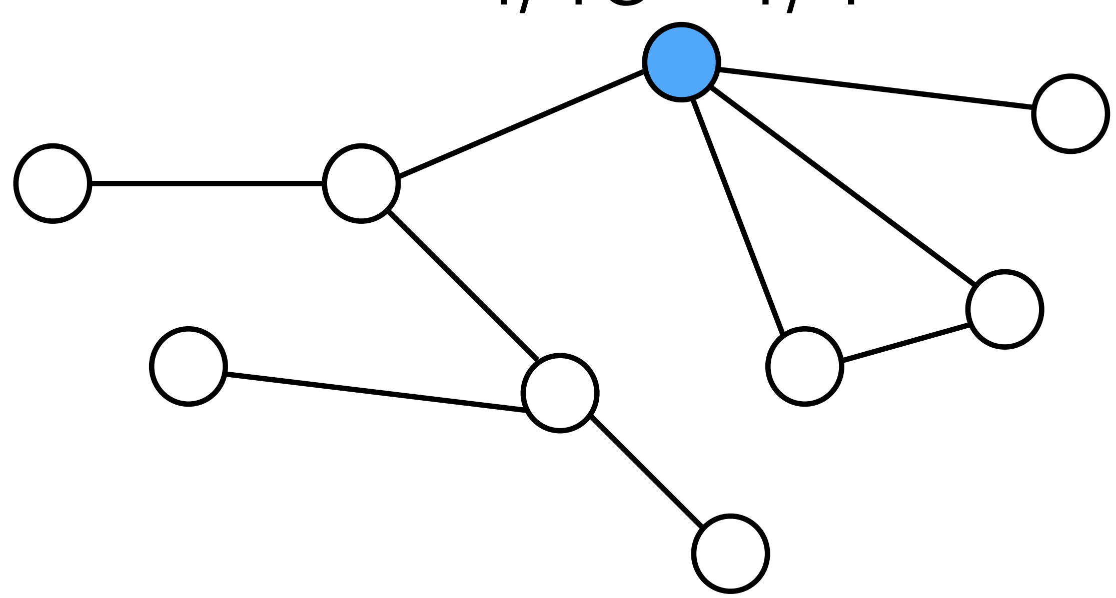
- **While** True:
  - run the random walk for MT(G) steps;
  - suppose it ends on the node v;
  - **return** v with probability 1/deg(v).



A Folklore Algorithm

- **While** True:
  - run the random walk for MT(G) steps;
  - suppose it ends on the node v;
  - **return** v with probability 1/deg(v).

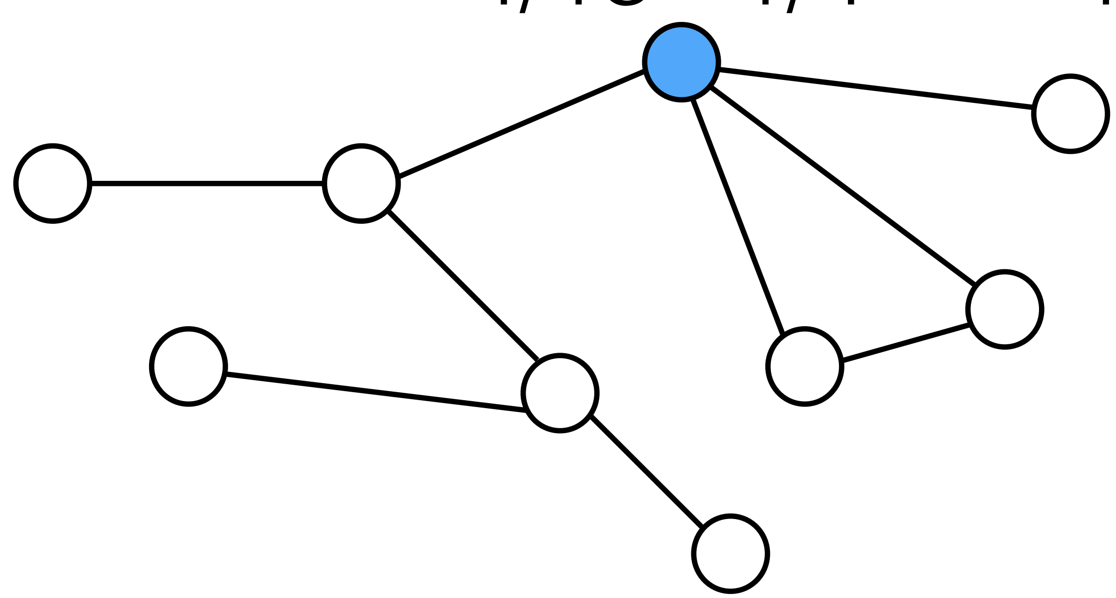
~ 4/18 · 1/4



A Folklore Algorithm

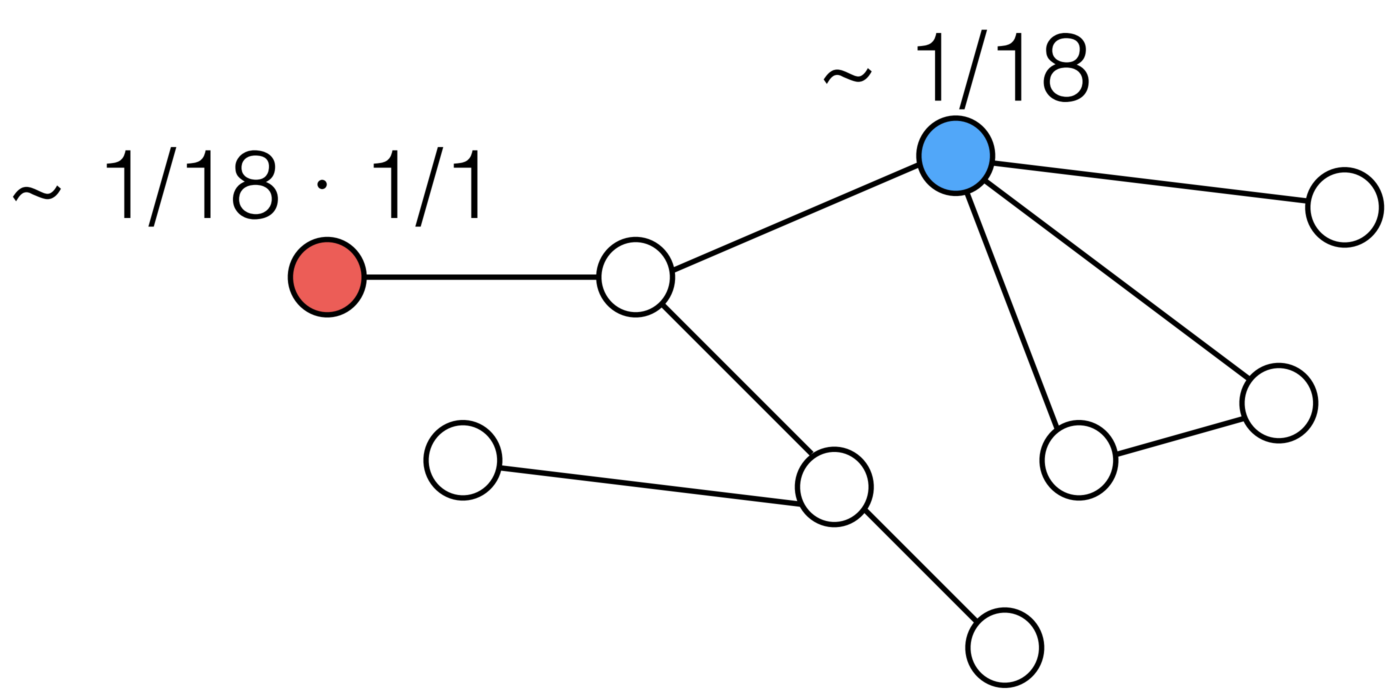
- **While** True:
  - run the random walk for MT(G) steps;
  - suppose it ends on the node v;
  - **return** v with probability 1/deg(v).

$\sim 4/18 \cdot 1/4 = \sim 1/18$



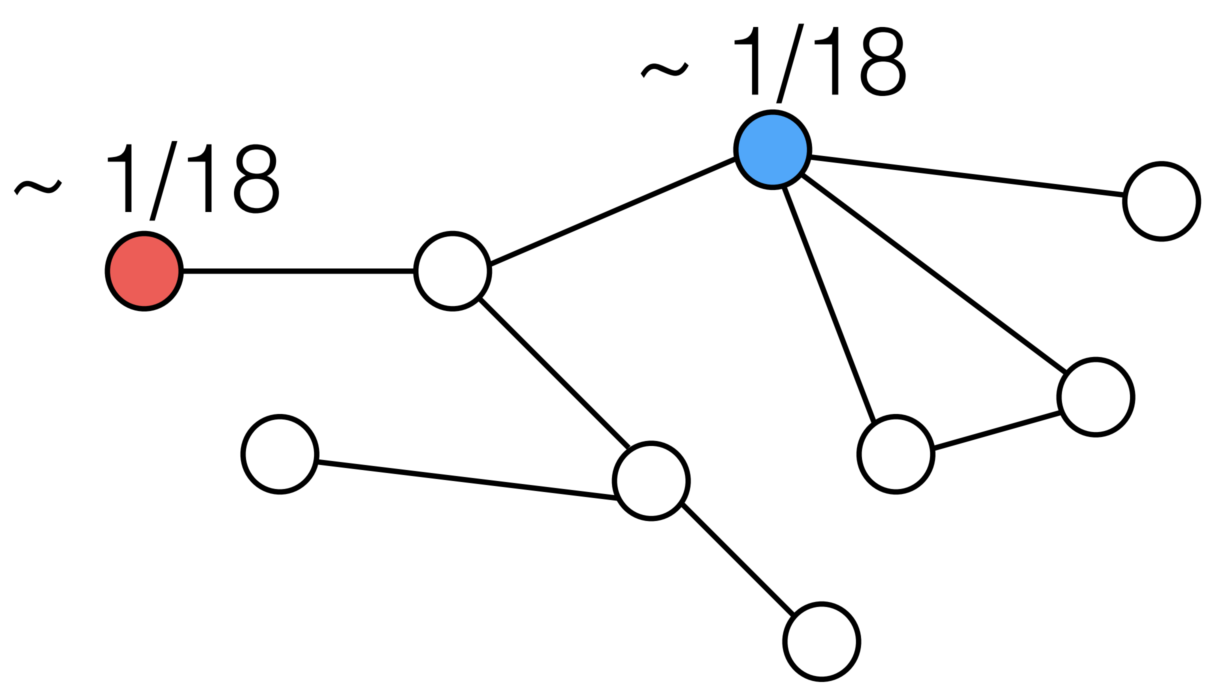
A Folklore Algorithm

- **While** True:
  - run the random walk for MT(G) steps;
  - suppose it ends on the node v;
  - **return** v with probability 1/deg(v).



A Folklore Algorithm

- **While** True:
  - run the random walk for MT(G) steps;
  - suppose it ends on the node v;
  - **return** v with probability 1/deg(v).



A Folklore Algorithm

- **While** True:
  - run the random walk for MT(G) steps;
  - suppose it ends on the node v;
  - **return** v with probability 1/deg(v).

This algorithm returns a node chosen *(arbitrarily close to)* **uniformly at random**

A Folklore Algorithm

- **While** True:
  - run the random walk for MT(G) steps;
  - suppose it ends on the node v;
  - **return** v with probability 1/deg(v).

One can easily show that this algorithm **downloads**, with high probability, at most *O(MT(G) · AvgDeg(G))* nodes from the network

Can one do better?

- In *[C., Dasgupta, Kumar, Lattanzi, Sarlós,'16]* we analyzed various algorithms for selecting a UAR node.

Can one do better?

- In *[C., Dasgupta, Kumar, Lattanzi, Sarlós,’16]* we analyzed various algorithms for selecting a UAR node.
- Some of them were on-par with the Folklore Algorithm, some of them were worse.

Can one do better?

- In *[C., Dasgupta, Kumar, Lattanzi, Sarlós,'16]* we analyzed various algorithms for selecting a UAR node.
- Some of them were on-par with the Folklore Algorithm, some of them were worse.
- In *[C., Haddadan]* we show that:
  - if an algorithm downloads < *o(MT(G) AvgDeg(G))* nodes from the network, then it cannot return anything close to a uniform-at-random node.
- ***That is, the Folklore algorithm is optimal.***

Statistical Significance

- The results show that, if one does not run the walk (or, generally, the algorithm), for enough many steps, then one cannot have any statistical significance on its result.

Statistical Significance

- The results show that, if one does not run the walk (or, generally, the algorithm), for enough many steps, then one cannot have any statistical significance on its result.

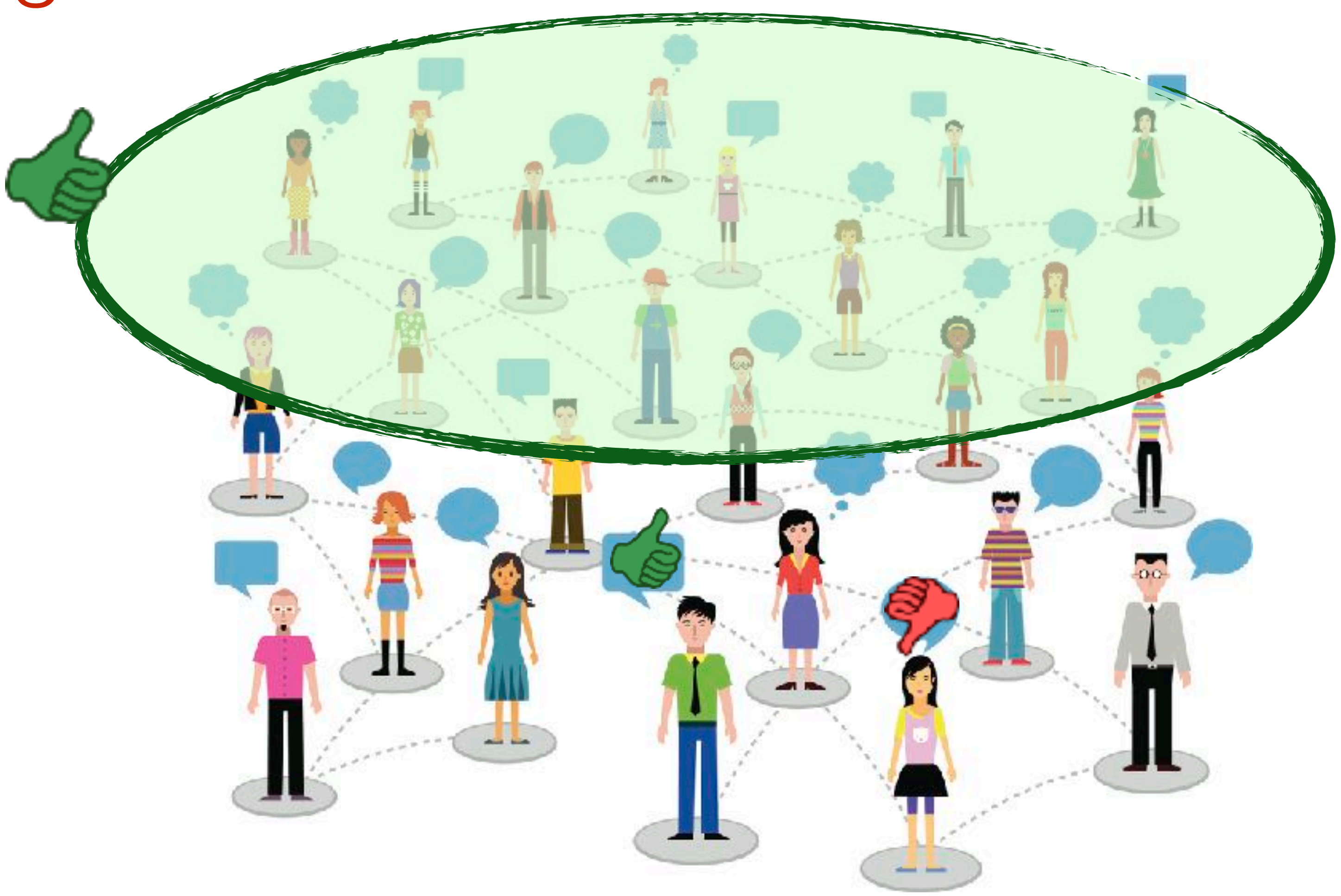
Statistical Significance

- The results show that, if one does not run the walk (or, generally, the algorithm), for enough many steps, then one cannot have any statistical significance on its result.

If the walk was not ran for long enough, claiming that “ *is the opinion of (roughly) half of the network”* might just be very wrong.

Statistical Significance

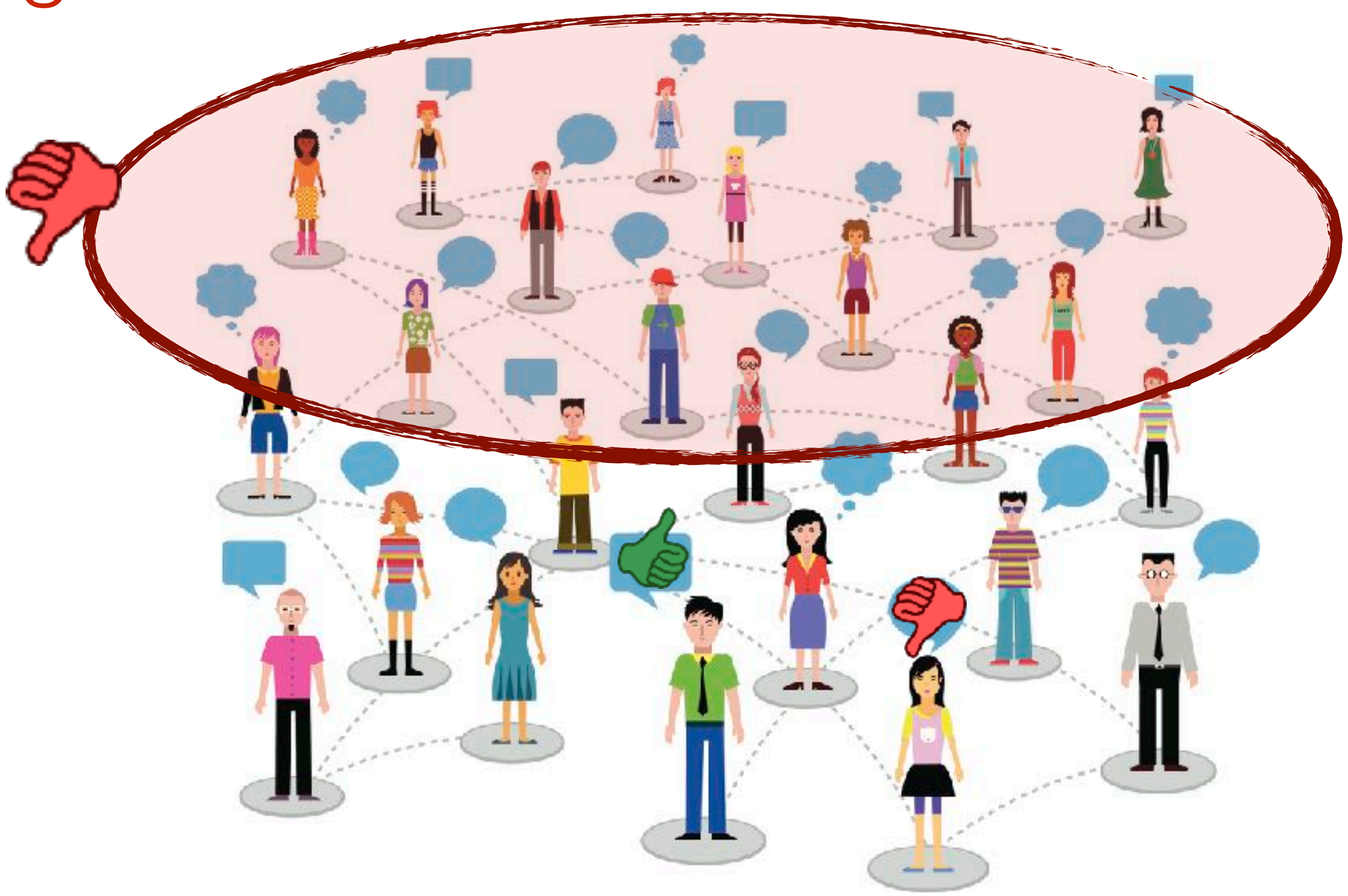
- The results show that, if one does not run the walk (or, generally, the algorithm), for enough many steps, then one cannot have any statistical significance on its result.



If the walk was not ran for long enough, claiming that " *is the opinion of (roughly) half of the network*" might just be very wrong.

Statistical Significance

- The results show that, if one does not run the walk (or, generally, the algorithm), for enough many steps, then one cannot have any statistical significance on its result.



If the walk was not ran for long enough, claiming that "👍 *is the opinion of (roughly) half of the network*" might just be very wrong.

Statistical Significance

- The results show that, if one does not run the walk (or, generally, the algorithm), for enough many steps, then one cannot have any statistical significance on its result.

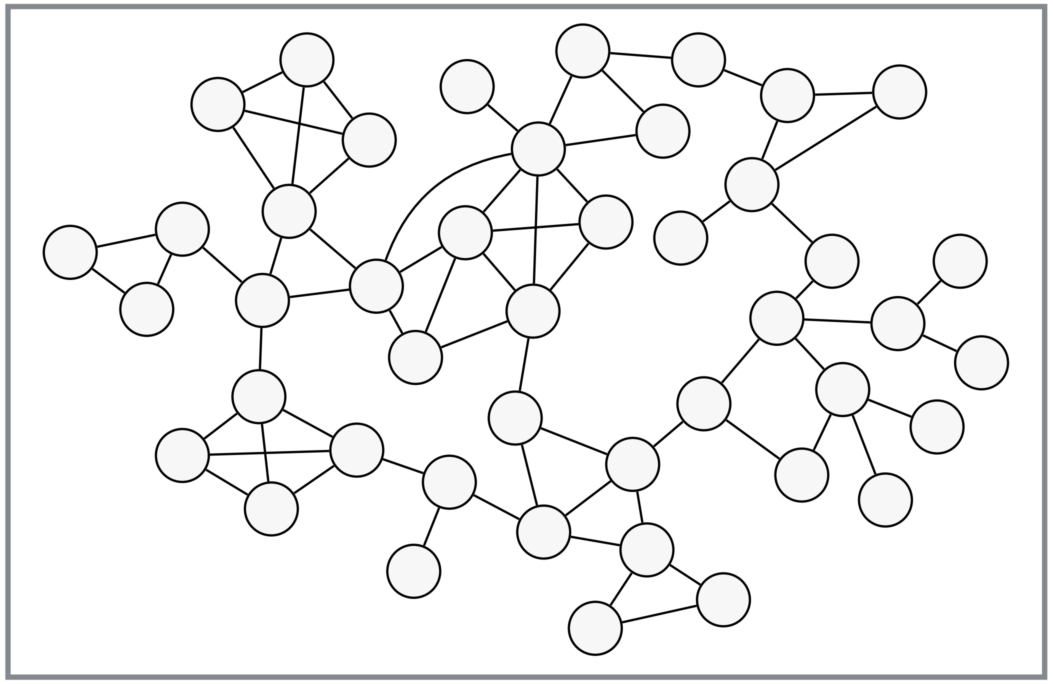
- This is something that it is important to keep in mind in this setting, and in many others as well…

Other Distributions…

- In *[C., Dasgupta, Kumar, Lattanzi, Sarlós,'16]*, we also give algorithms that select nodes randomly according to various skewed distributions *(e.g., probability proportional to some power of the degree).*

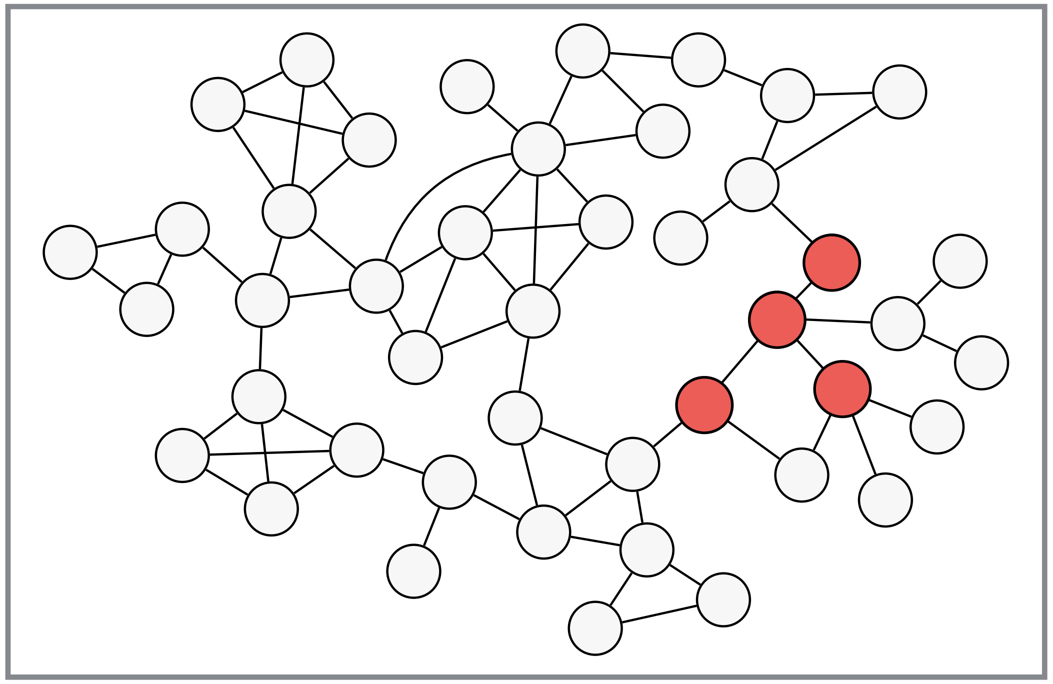
Counting Graphlets

Graphlets



Graph on n nodes

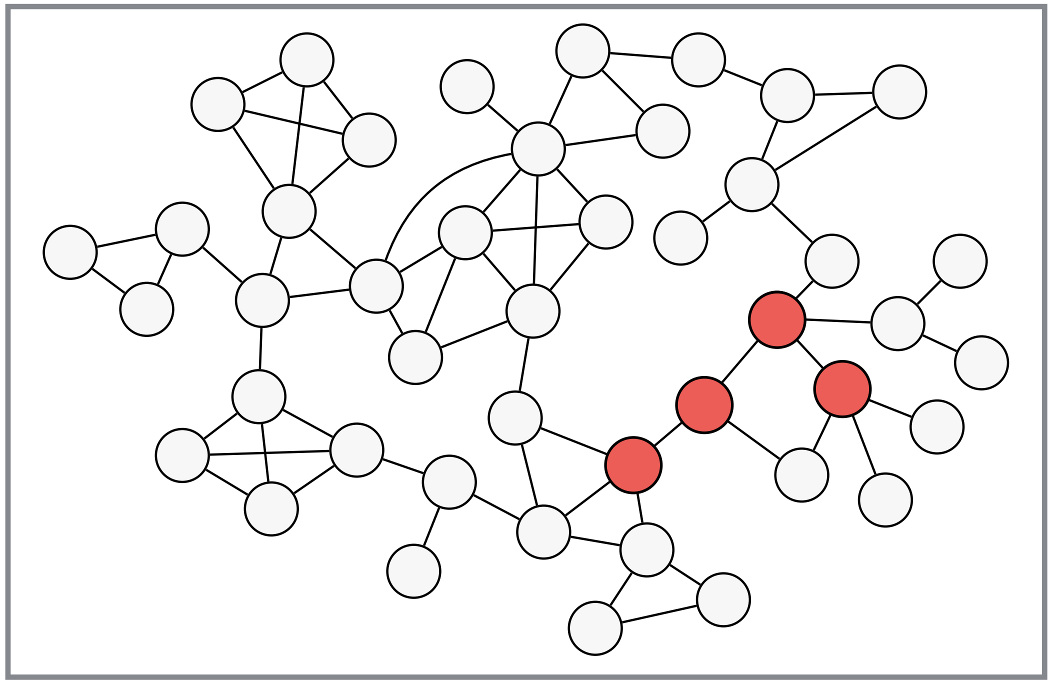
Graphlets



Graph on n nodes

A k-graphlet is a connected induced subgraph of k nodes

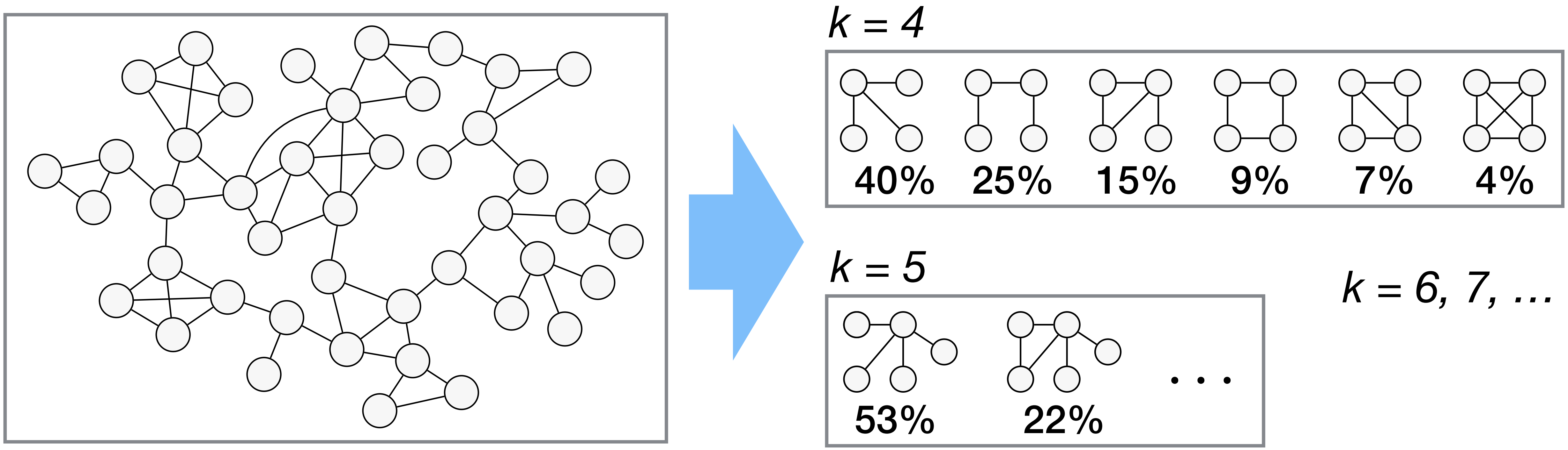
Graphlets



Graph on n nodes

A k-graphlet is a connected induced subgraph of k nodes

Graphlets



Graph on n nodes

Distribution of graphlets on k nodes

Applications:

- social network analysis
- graph mining
- computational biology

Challenges

- exact counting is infeasible ($n^{\Omega(k)}$)
- even approximations are costly
- scaling *n* and *k* is hard in practice

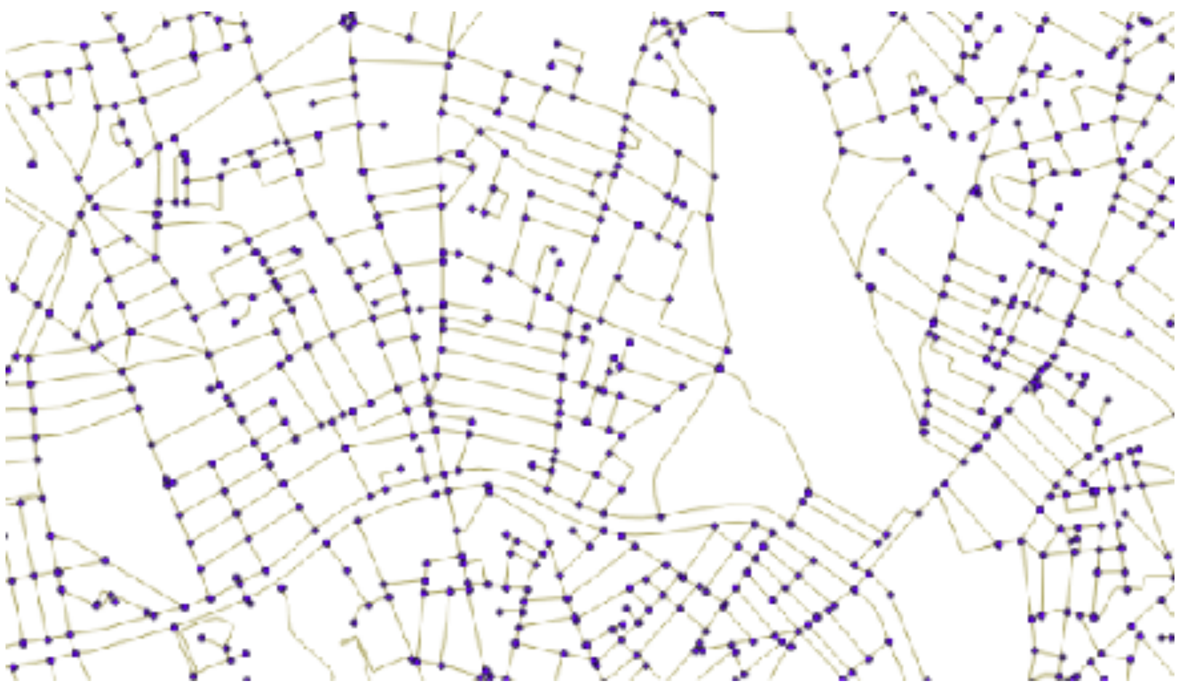
Graphlet Distribution

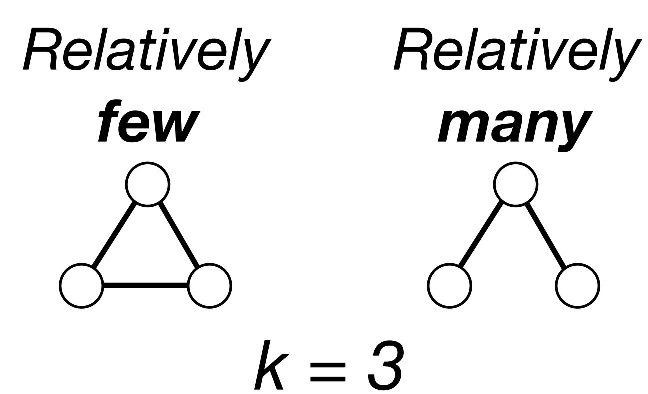
Why is it interesting?

- The Graphlet Distribution has been used to
  - *classify*, and
  - *understand*
- networks (and different parts of the same network)

Graphlet Distribution

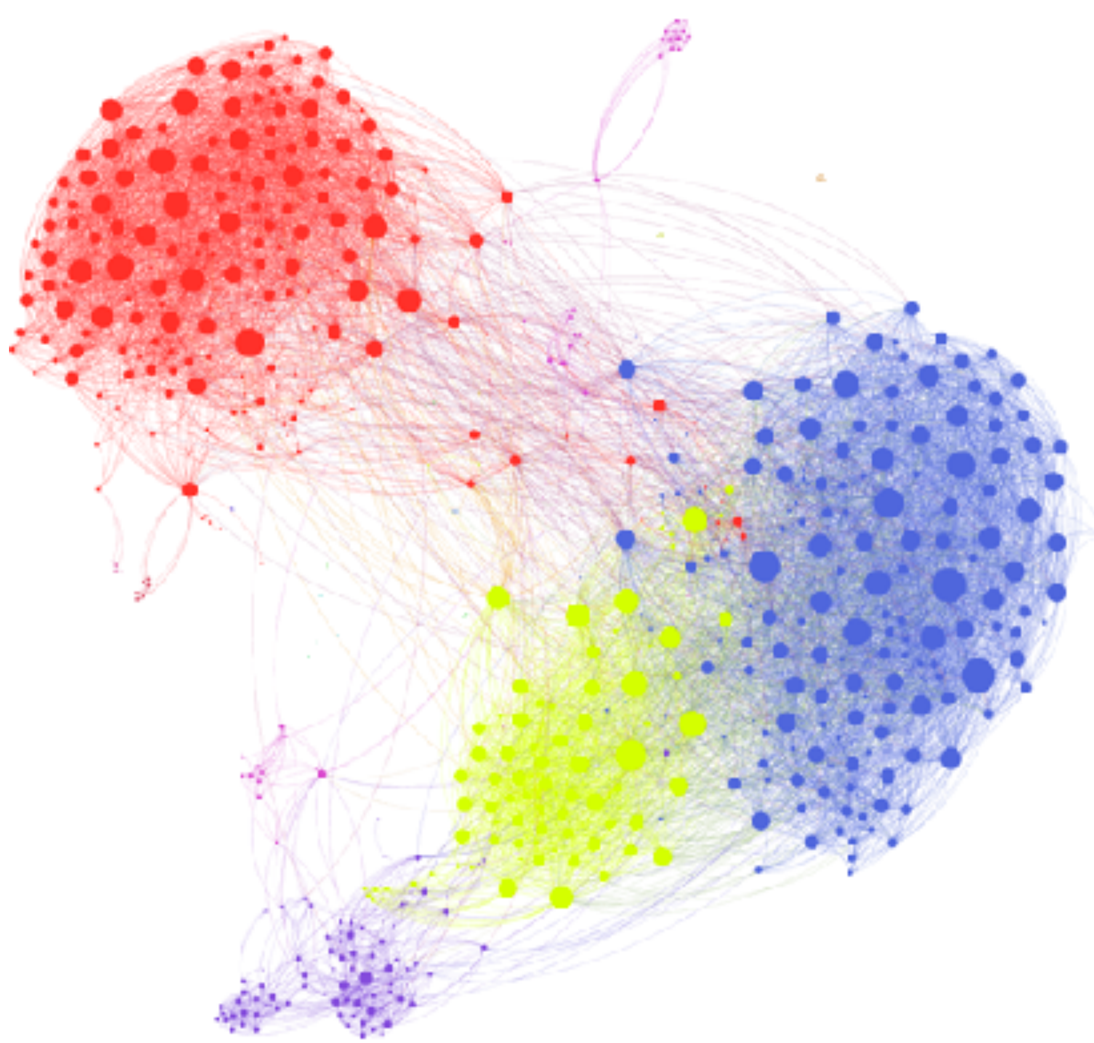
Road Networks do not contain many triangles

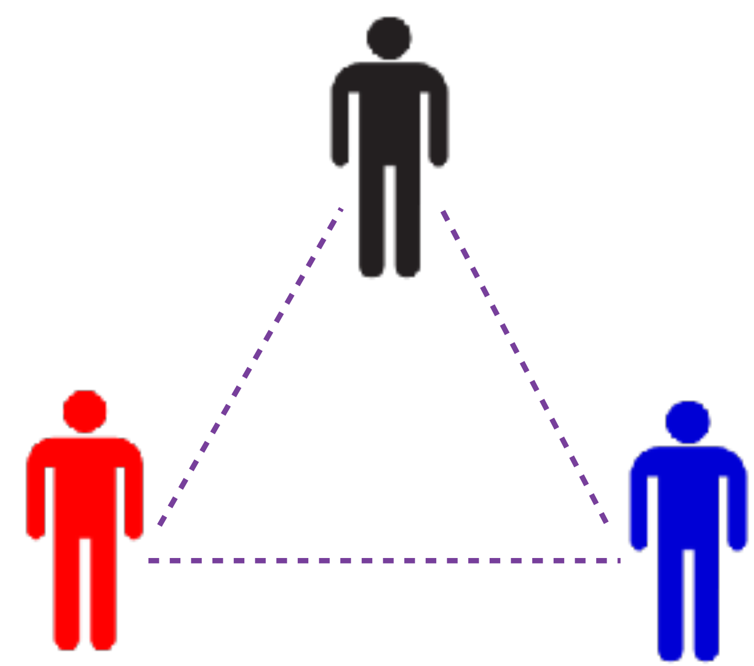


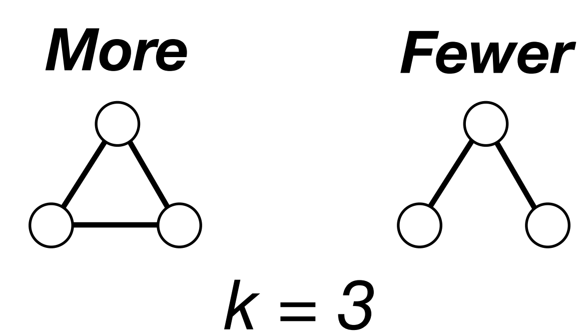


Graphlet Distribution

Many Social Networks contain Dense Communities



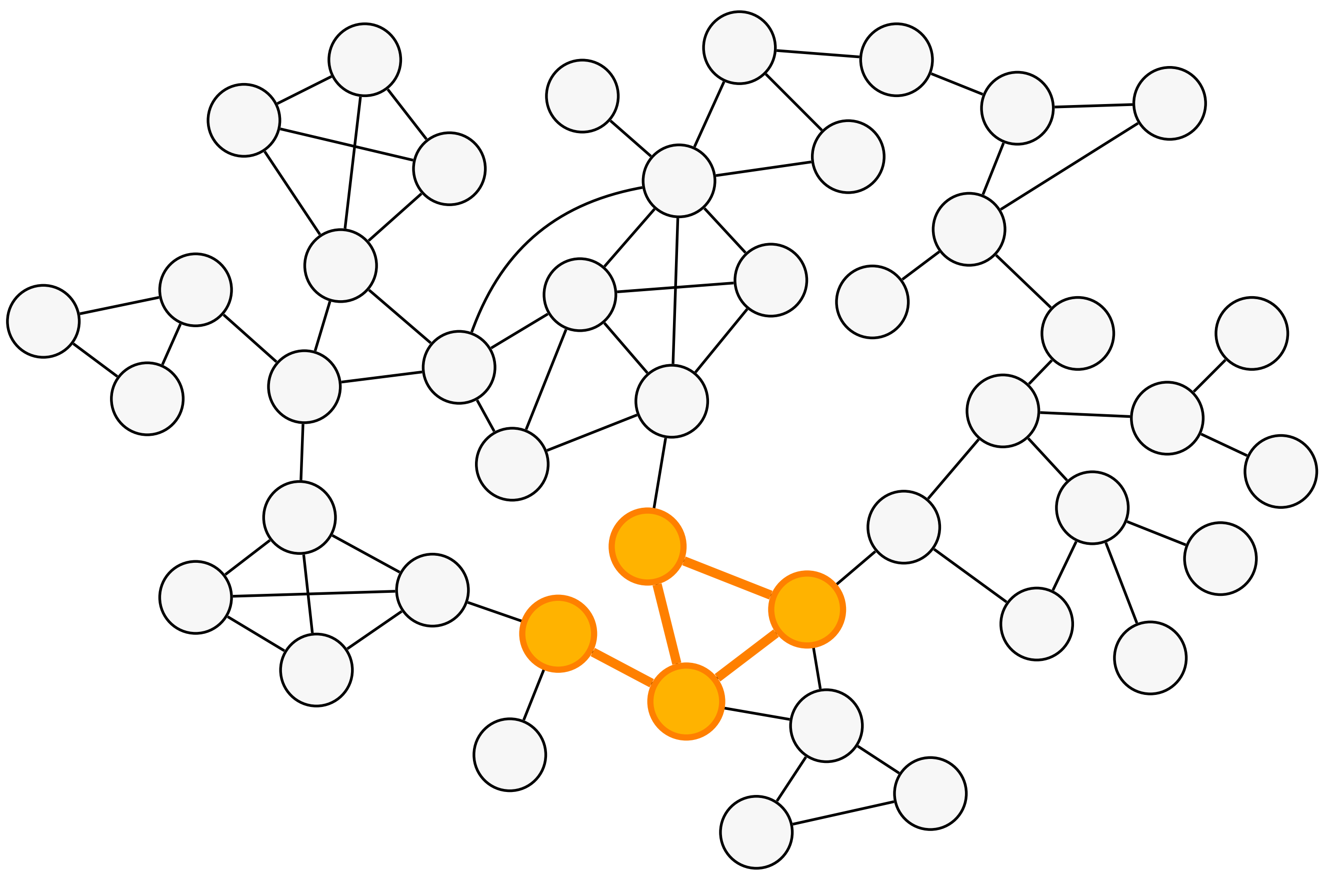




Computing the Graphlet Distribution

Random Walk

[Bhuyian et al., ICDM, 2012]



Random walk over adjacent graphlets in the graph

Two graphlets are **adjacent** if they share *k-1* nodes in the graph

If the walk is **sufficiently long**, it will end on a uniform-at-random graphlet of the graph

How long does the walk take to converge (Mixing Time)?

Issues with the Random Walk

[Bressan, C., Kumar, Leucci, Panconesi, ’17]

1. There are graphs where the mixing time of the RW is $\Omega(n^{k-1})$

(almost as bad as naive enumeration!)

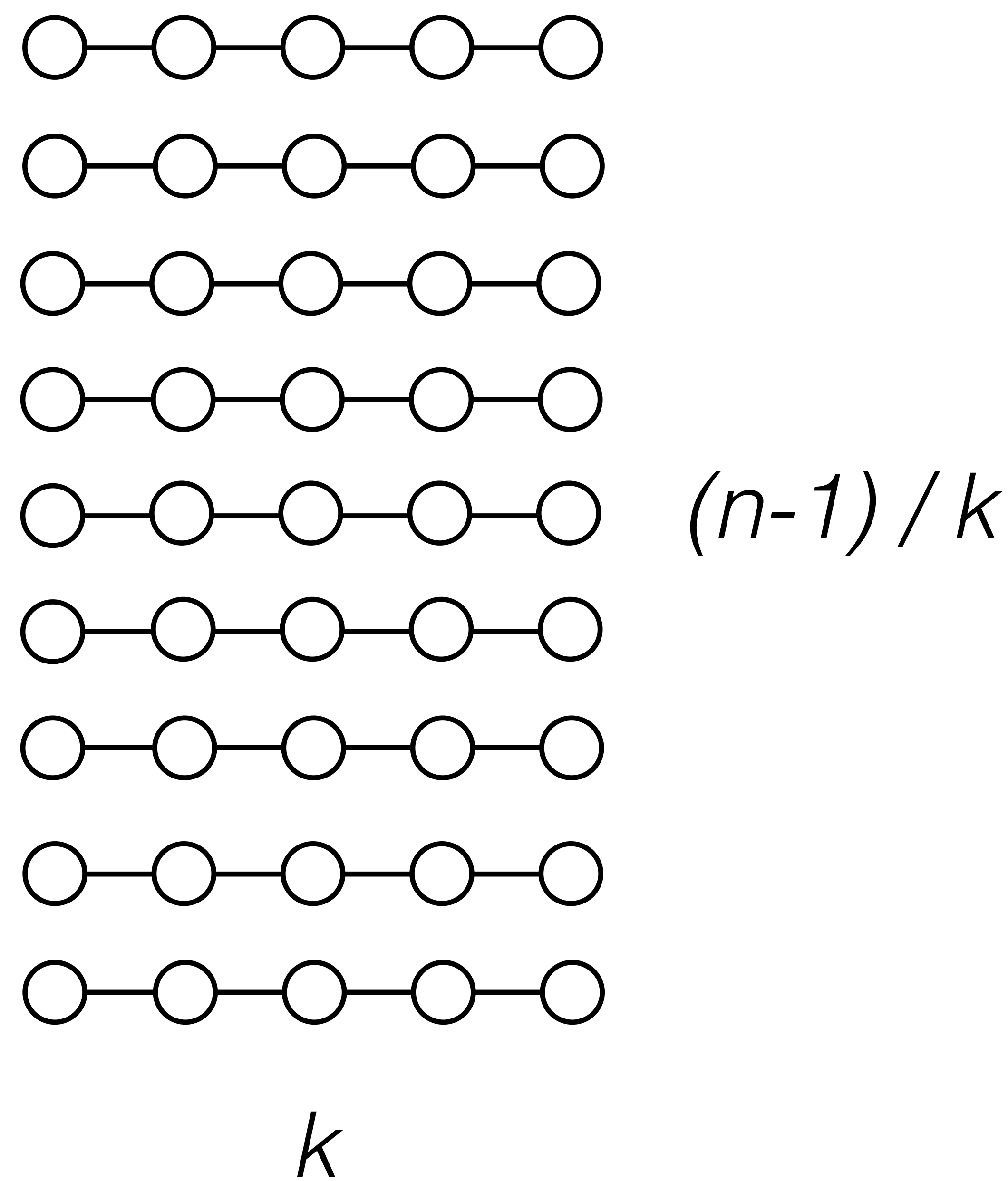
2. Happens even if one graphlet appears 99.99% of the time

3. Happens even on nice graphs, i.e., with high conductance

(a property believed to be shared by many social networks)

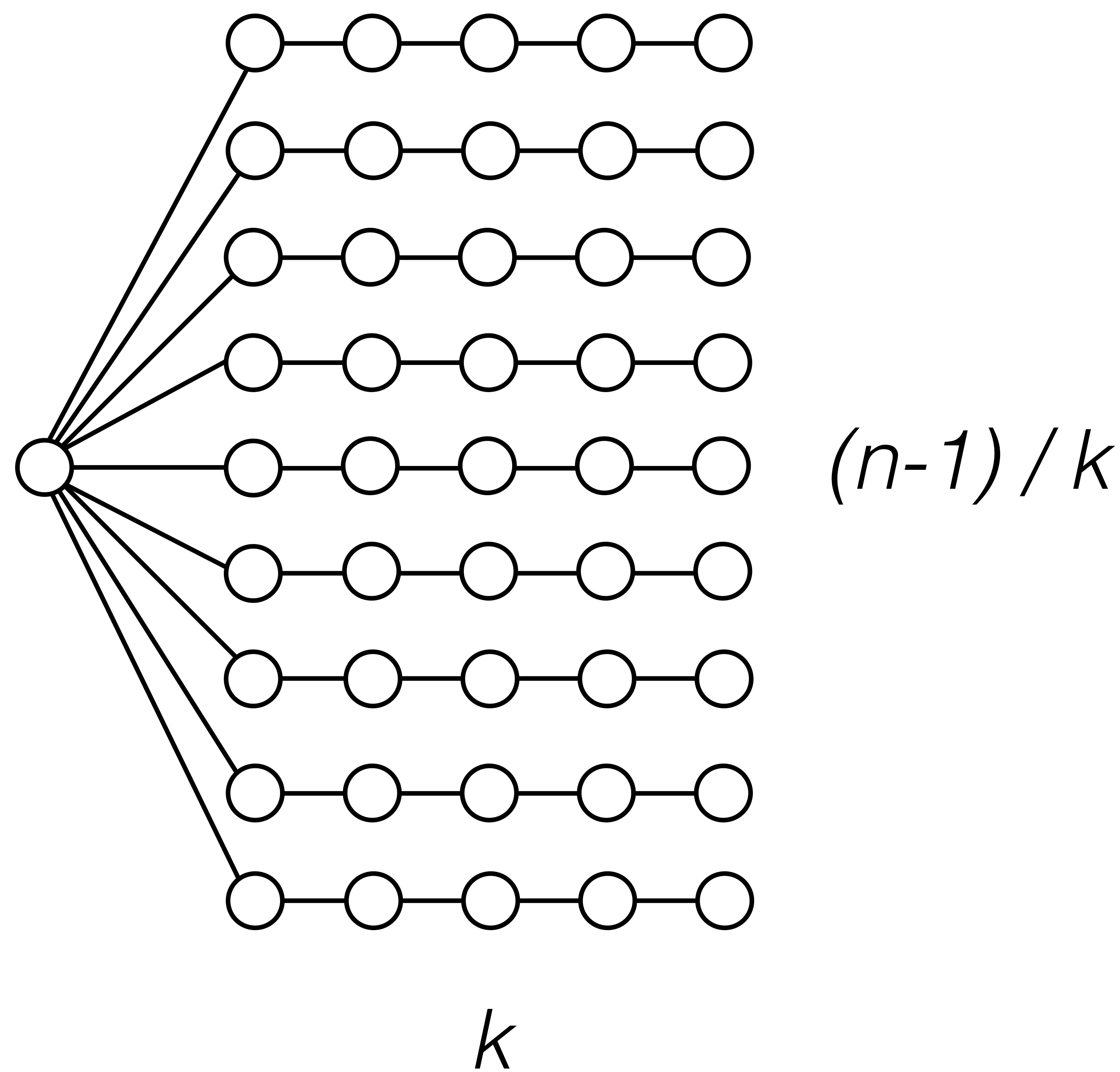
Issues with the Random Walk

[Bressan, C., Kumar, Leucci, Panconesi, '17]



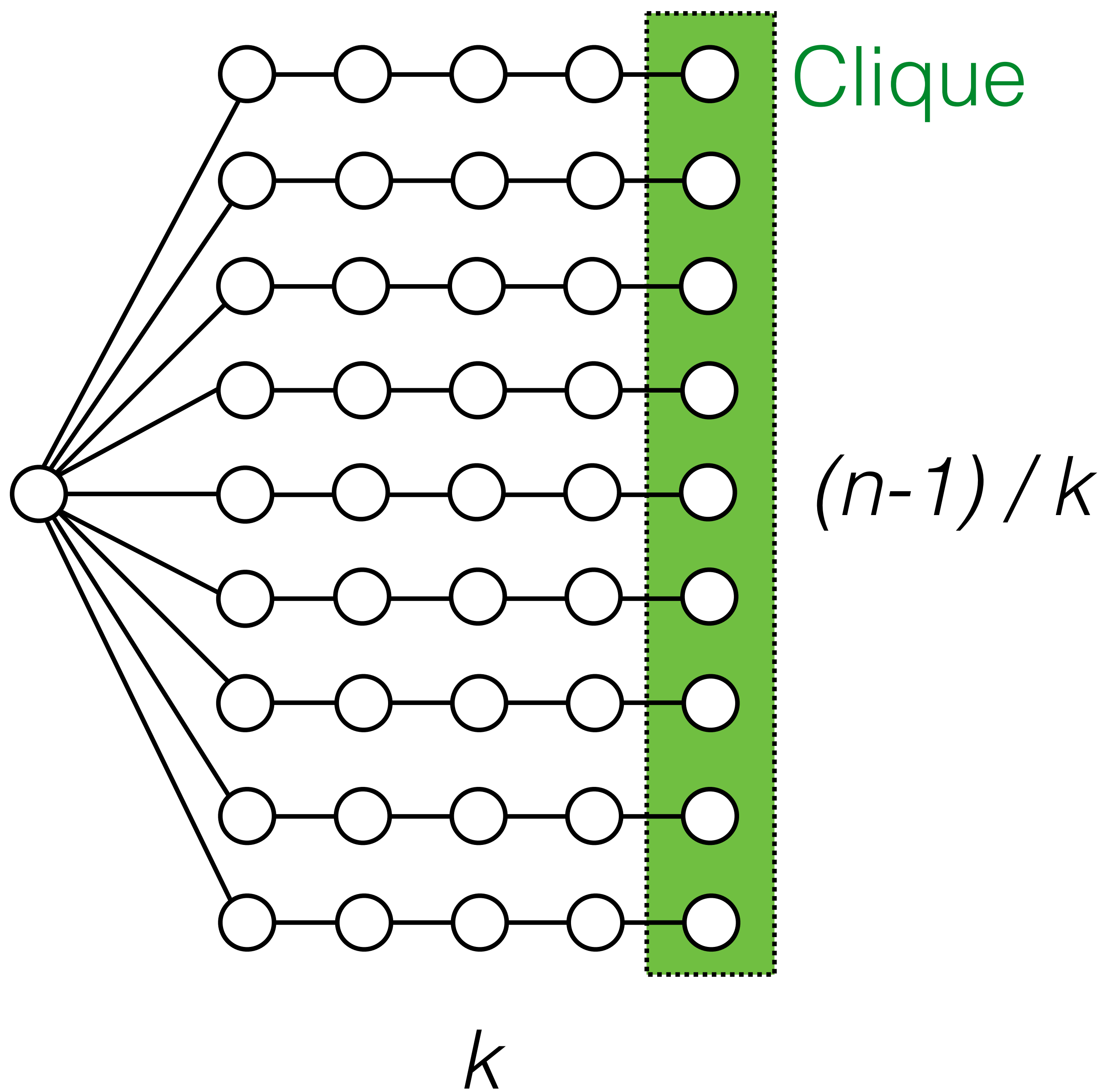
Issues with the Random Walk

[Bressan, C., Kumar, Leucci, Panconesi, '17]



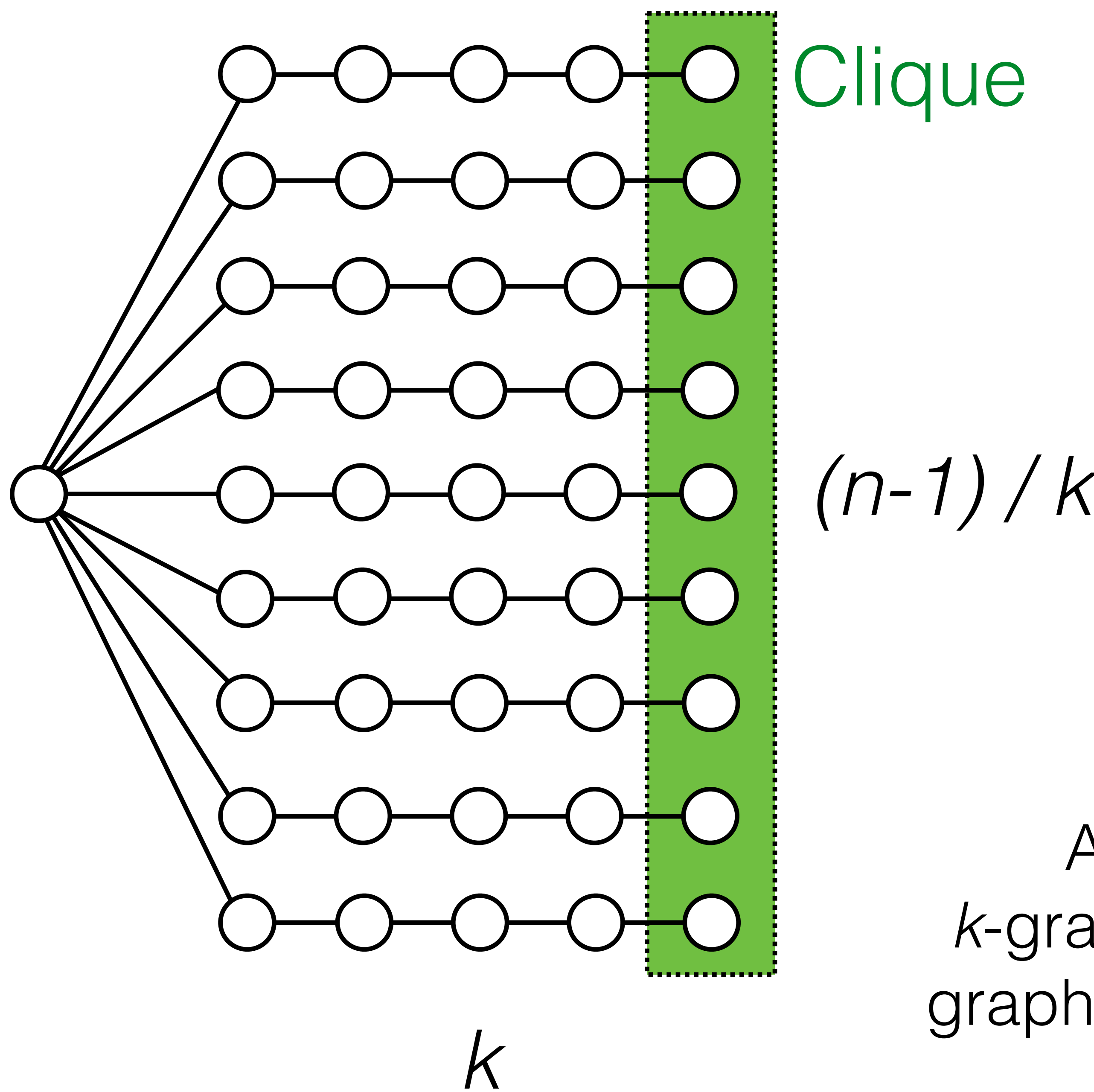
Issues with the Random Walk

[Bressan, C., Kumar, Leucci, Panconesi, '17]



Issues with the Random Walk

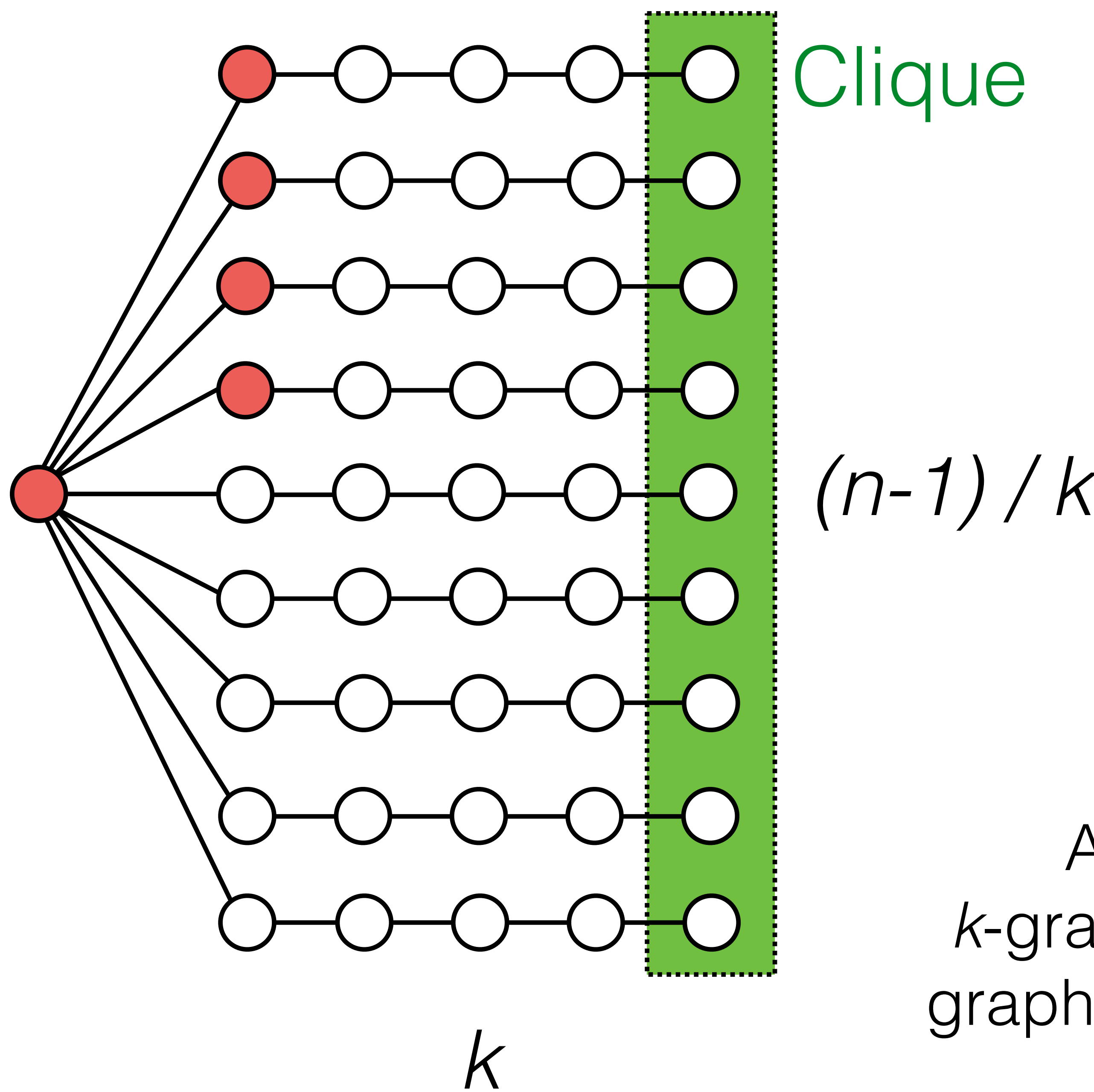
[Bressan, C., Kumar, Leucci, Panconesi, '17]



Almost all
k-graphlets in this
graph are *k*-cliques

Issues with the Random Walk

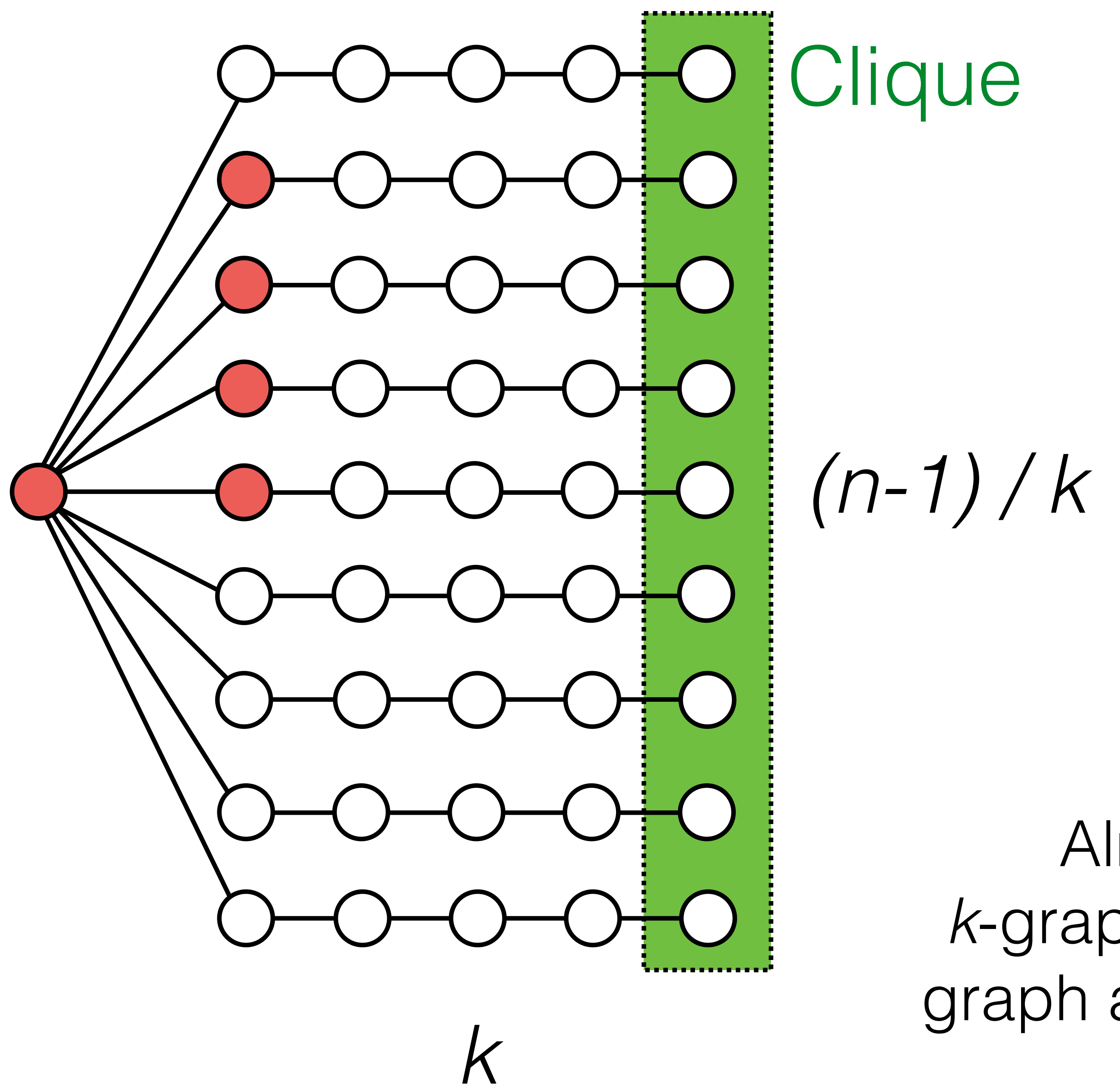
[Bressan, C., Kumar, Leucci, Panconesi, '17]



Almost all
k-graphlets in this
graph are *k*-cliques

Issues with the Random Walk

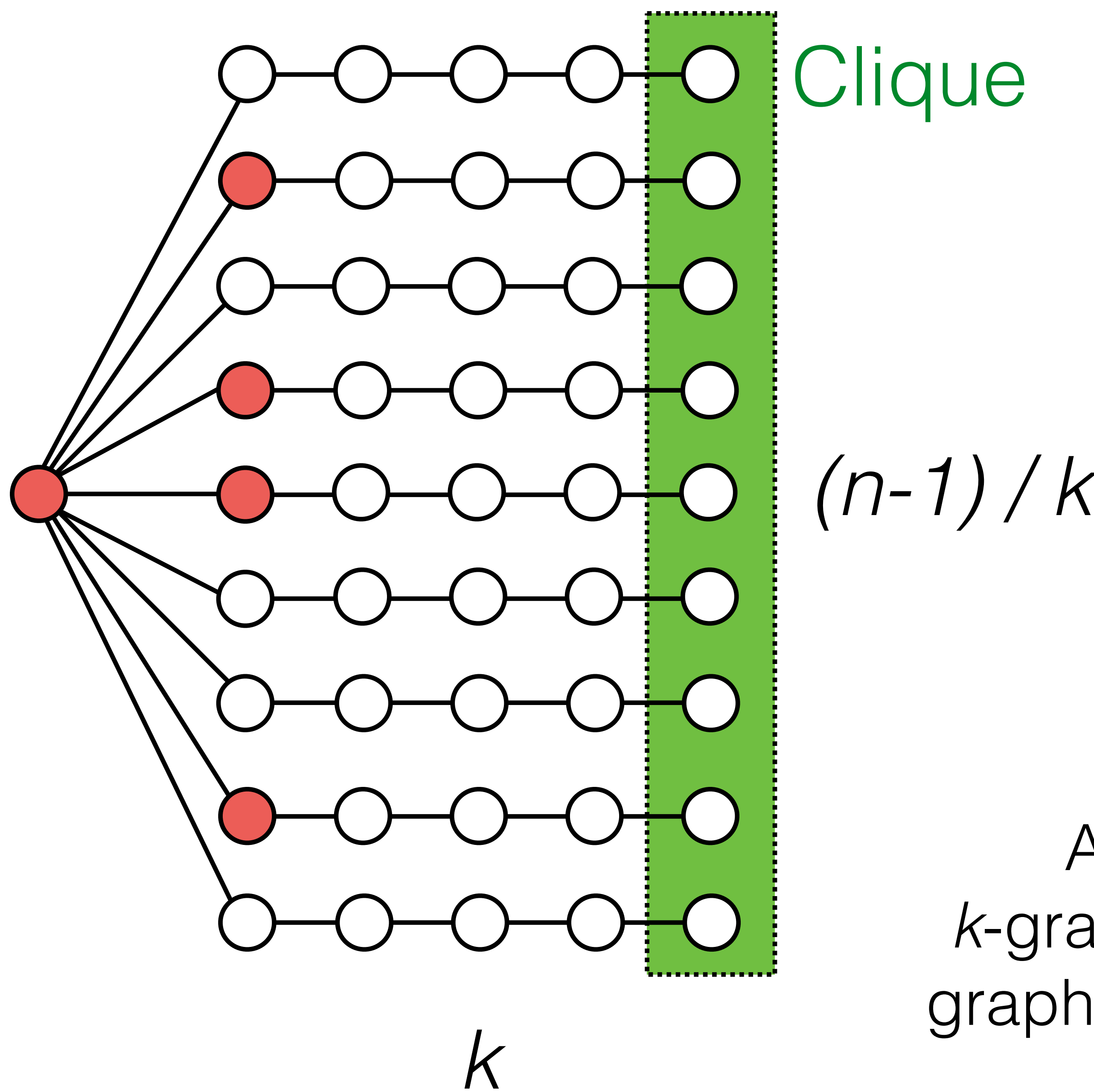
[Bressan, C., Kumar, Leucci, Panconesi, '17]



Almost all
k-graphlets in this
graph are *k*-cliques

Issues with the Random Walk

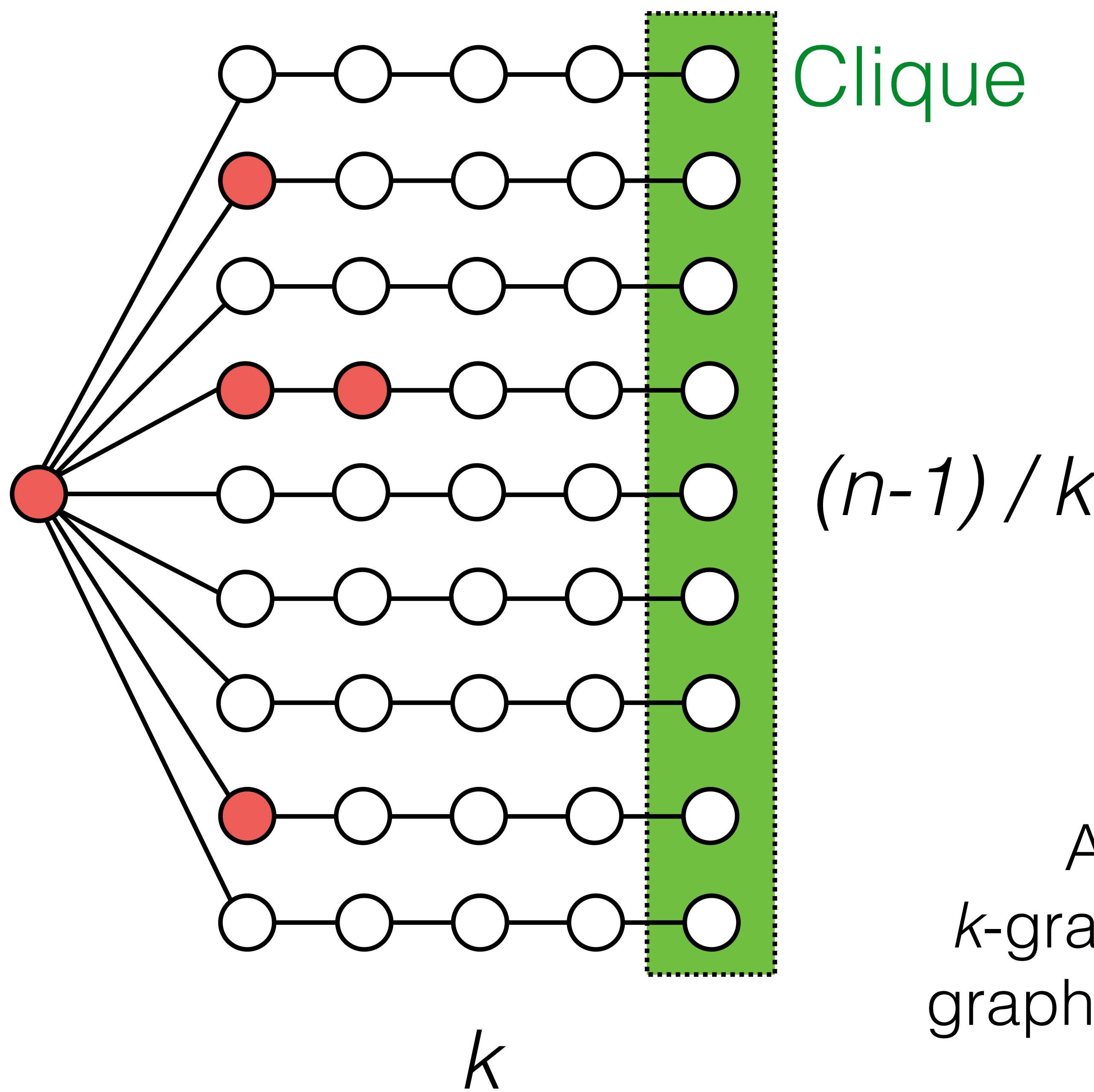
[Bressan, C., Kumar, Leucci, Panconesi, '17]



Almost all
k-graphlets in this
graph are *k*-cliques

Issues with the Random Walk

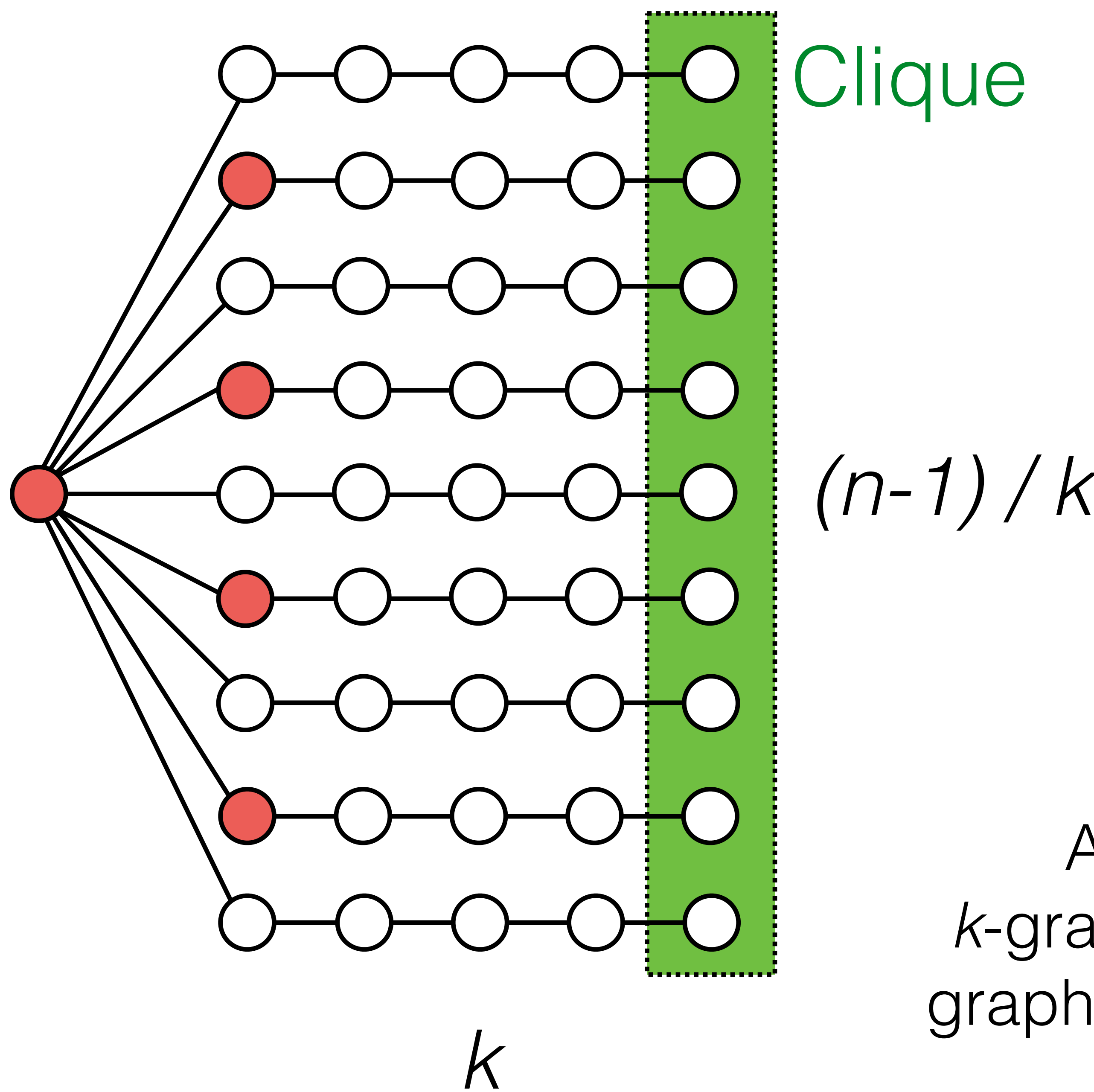
[Bressan, C., Kumar, Leucci, Panconesi, ’17]



Almost all
k-graphlets in this
graph are *k*-cliques

Issues with the Random Walk

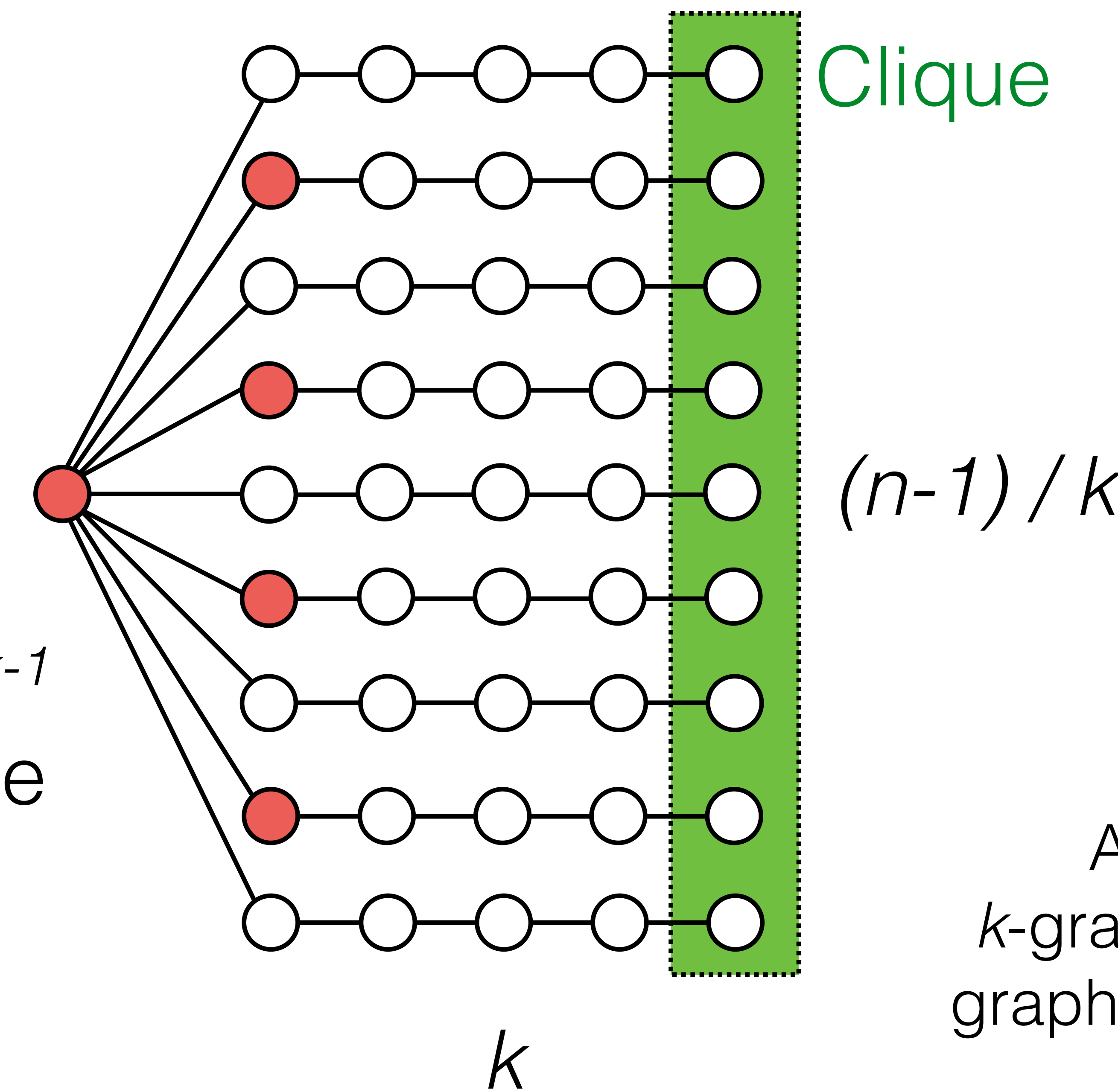
[Bressan, C., Kumar, Leucci, Panconesi, '17]



Almost all
k-graphlets in this
graph are *k*-cliques

Issues with the Random Walk

[Bressan, C., Kumar, Leucci, Panconesi, '17]

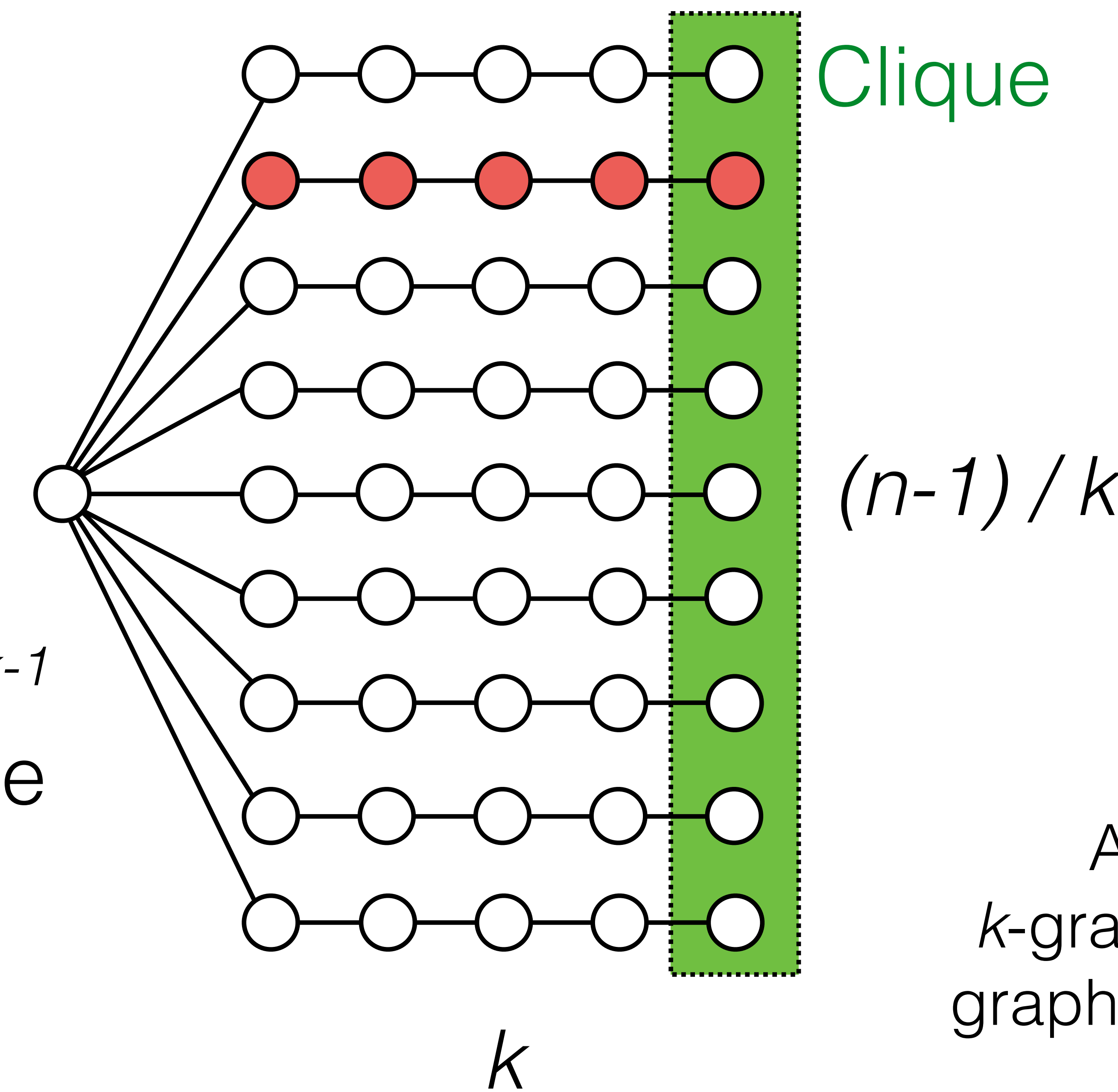


One needs ~ n^{k-1} steps to drop the star center

Almost all k-graphlets in this graph are k-cliques

Issues with the Random Walk

[Bressan, C., Kumar, Leucci, Panconesi, '17]

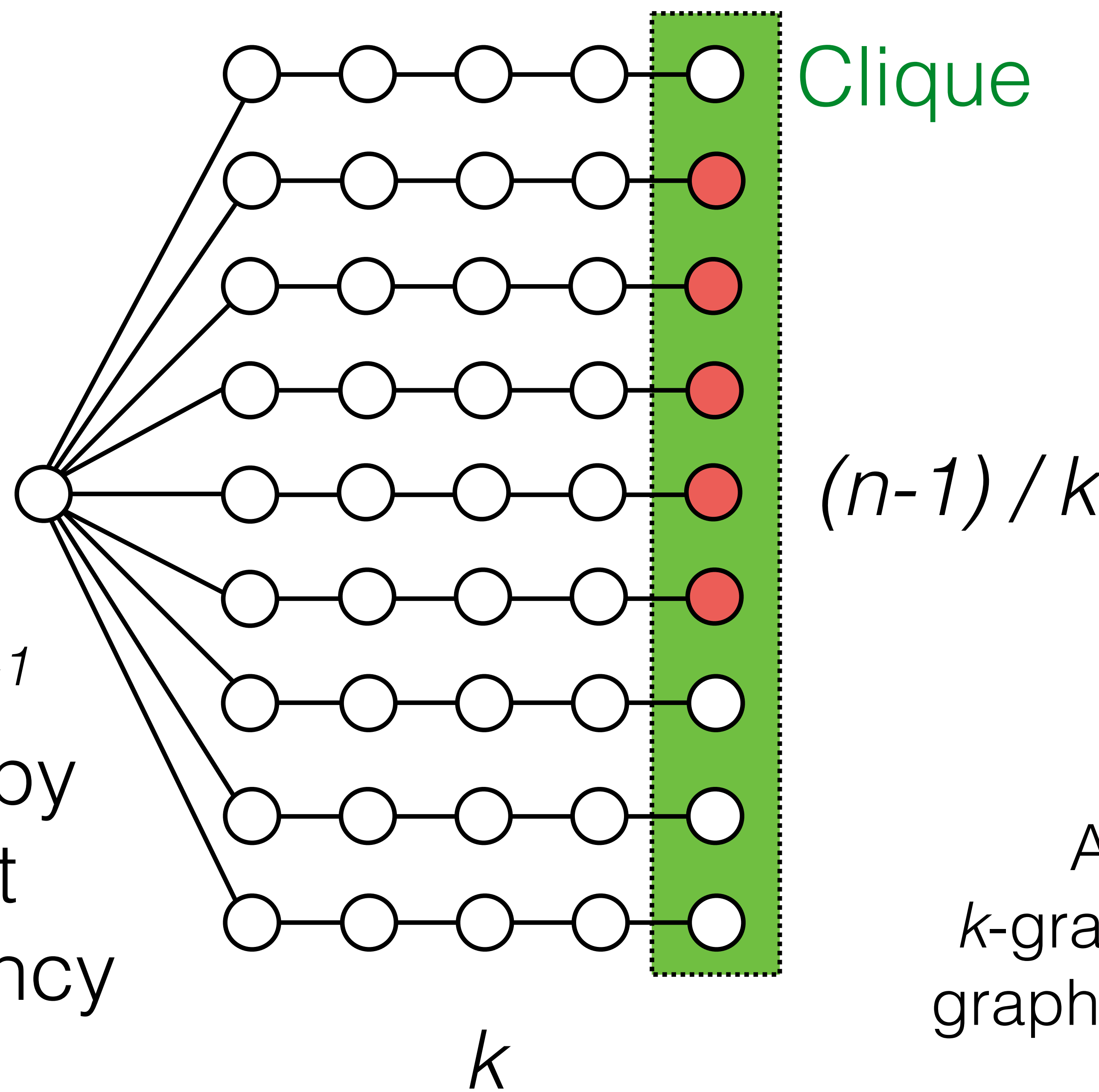


One needs ~ n^{k-1} steps to drop the star center

Almost all k-graphlets in this graph are k-cliques

Issues with the Random Walk

[Bressan, C., Kumar, Leucci, Panconesi, ’17]



One needs > n^{k-1} steps to see a copy of the k-graphlet with > 99% frequency

Almost all k-graphlets in this graph are k-cliques

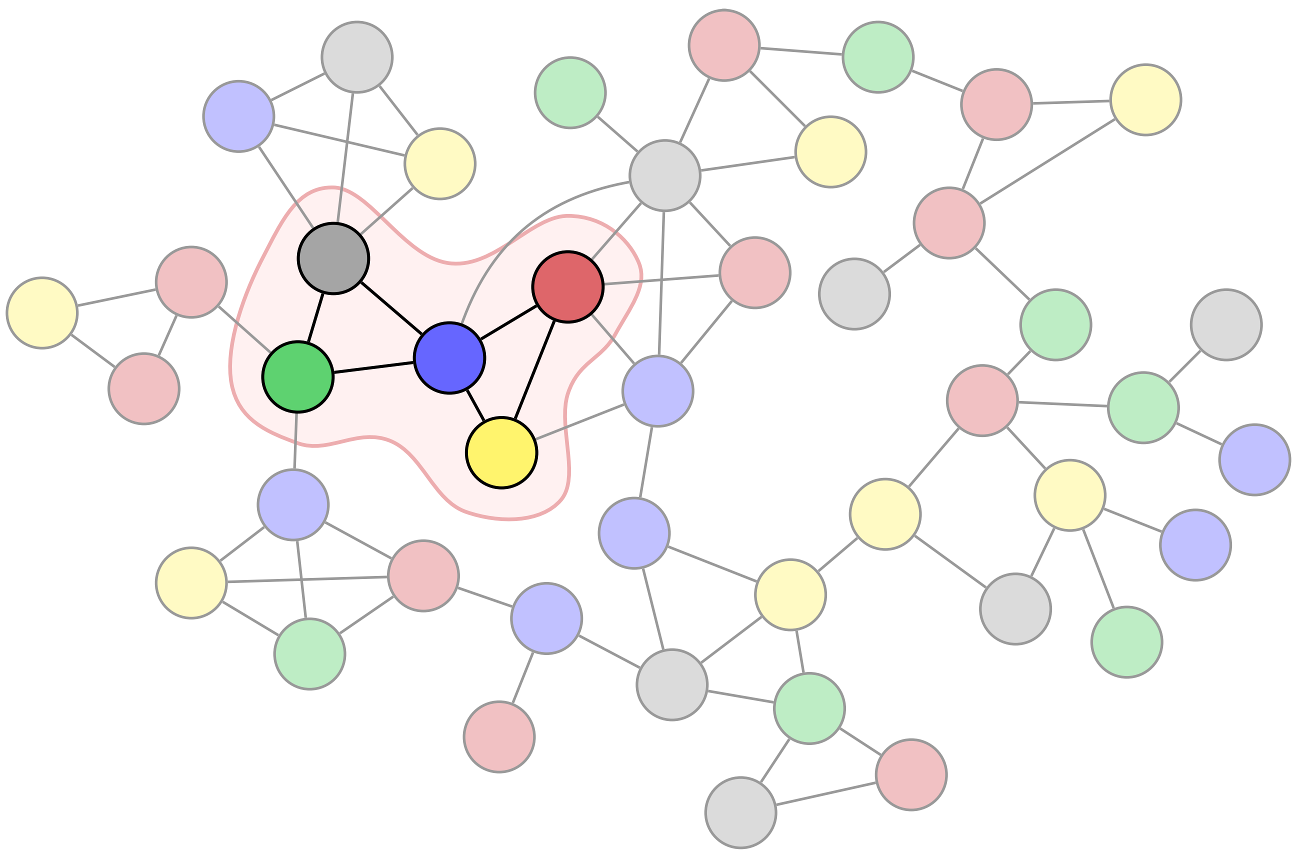
Random Walk

- If we run a *"short"* random walk repeatedly to sample UAR graphlets, and we return the empirical distribution that we obtain,

- we **cannot** be sure that the returned distribution will be (even moderately) close to the real one.

Color Coding (CC)

[Alon et al., JACM, 1995]



Randomly color the vertices of the graph with k colors.

A graphlet, with some probability, receives k distinct colors, i.e., becomes **colorful**

Can count **non-induced colorful trees** in $O(m\ c^k)$ time and $O(n\ c^k)$ space

In *[Bressan, C., Kumar, Leucci, Panconesi, '17]*, we modify CC to sample graphlets with bounded error

Experiments

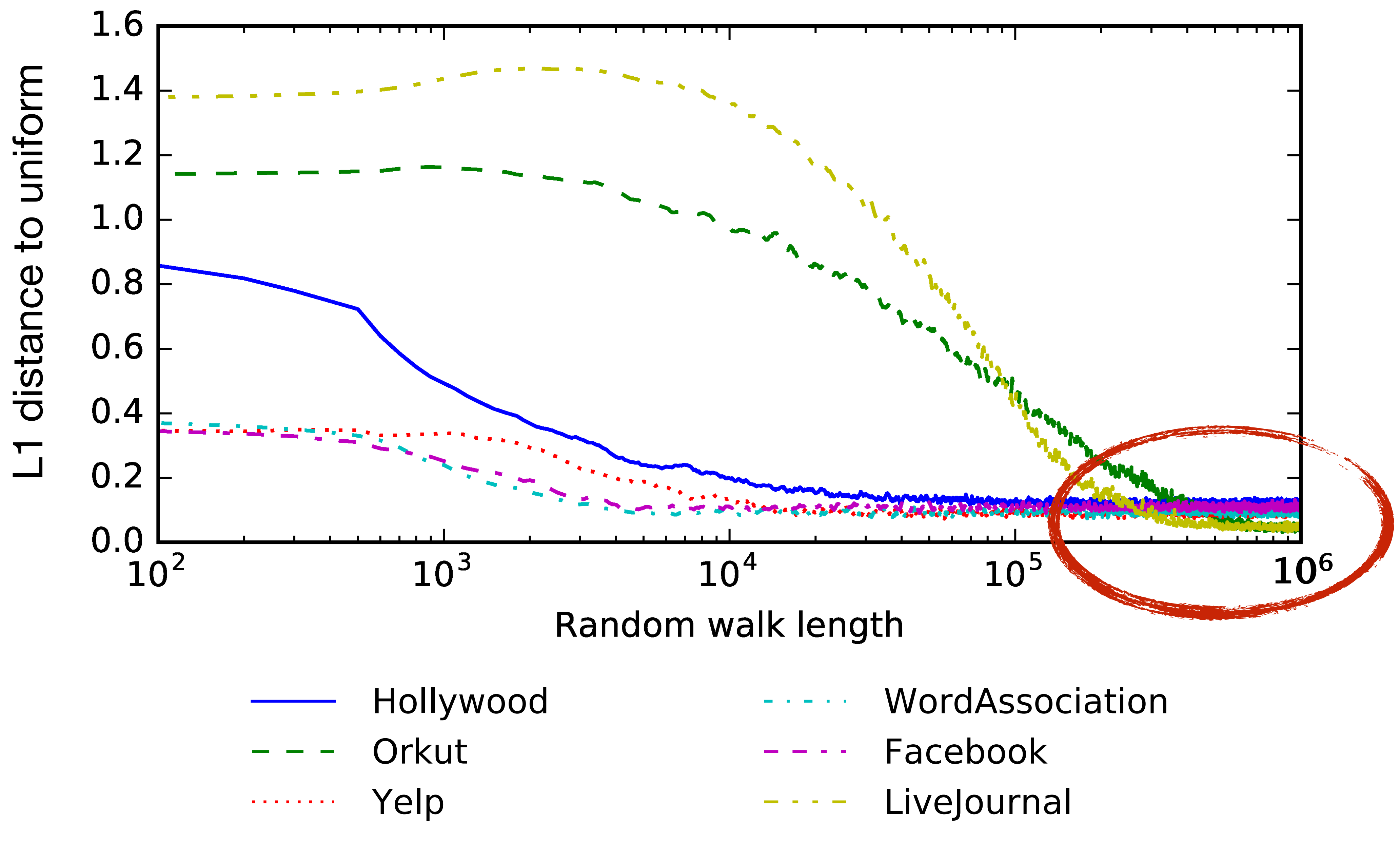
[Bressan, C., Kumar, Leucci, Panconesi, ’17]

Graph datasets

	nodes (millions)	edges (millions)
WordAssociation	0.01	0.06
Facebook	0.06	0.8
Yelp	0.2	1.3
Hollywood	2	114
Orkut	3	223
LiveJournal	5	49
Twitter	42	117

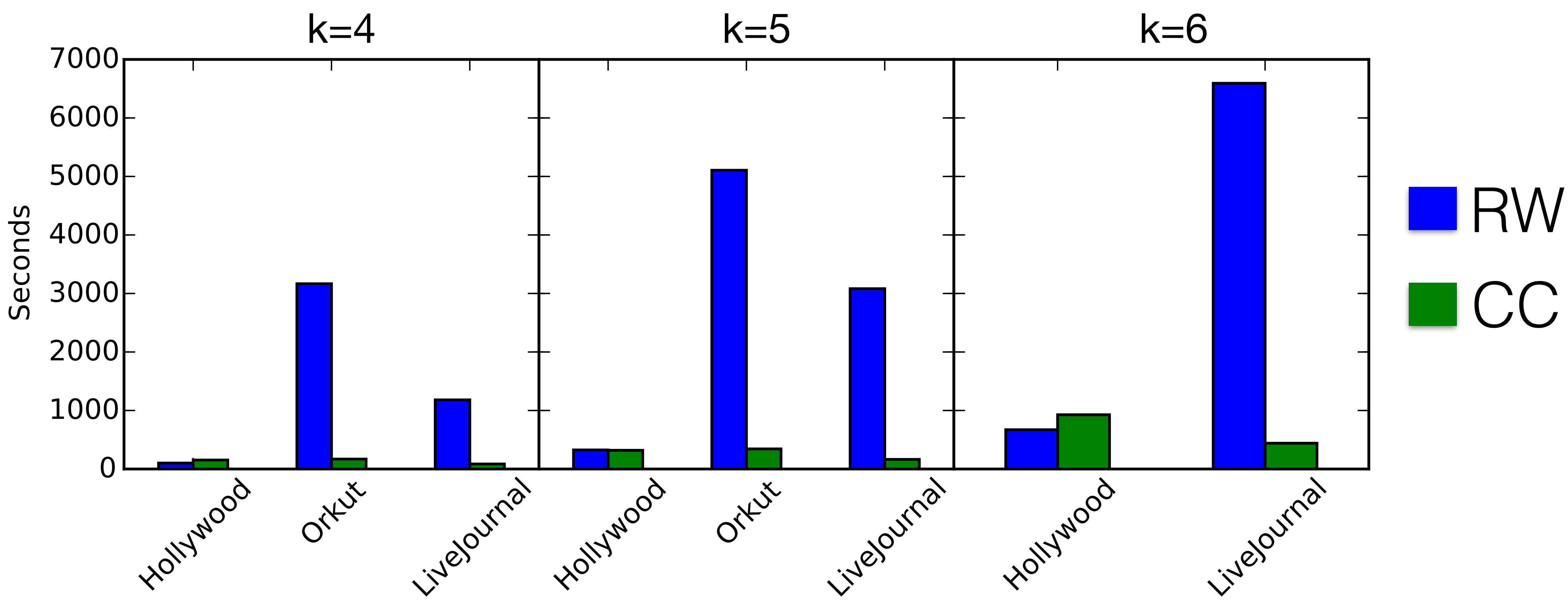
How does the RW behave in practice?

Distance of the Random Walk samples from the uniform distribution, as a function of the random walk length



Random Walk vs Color Coding

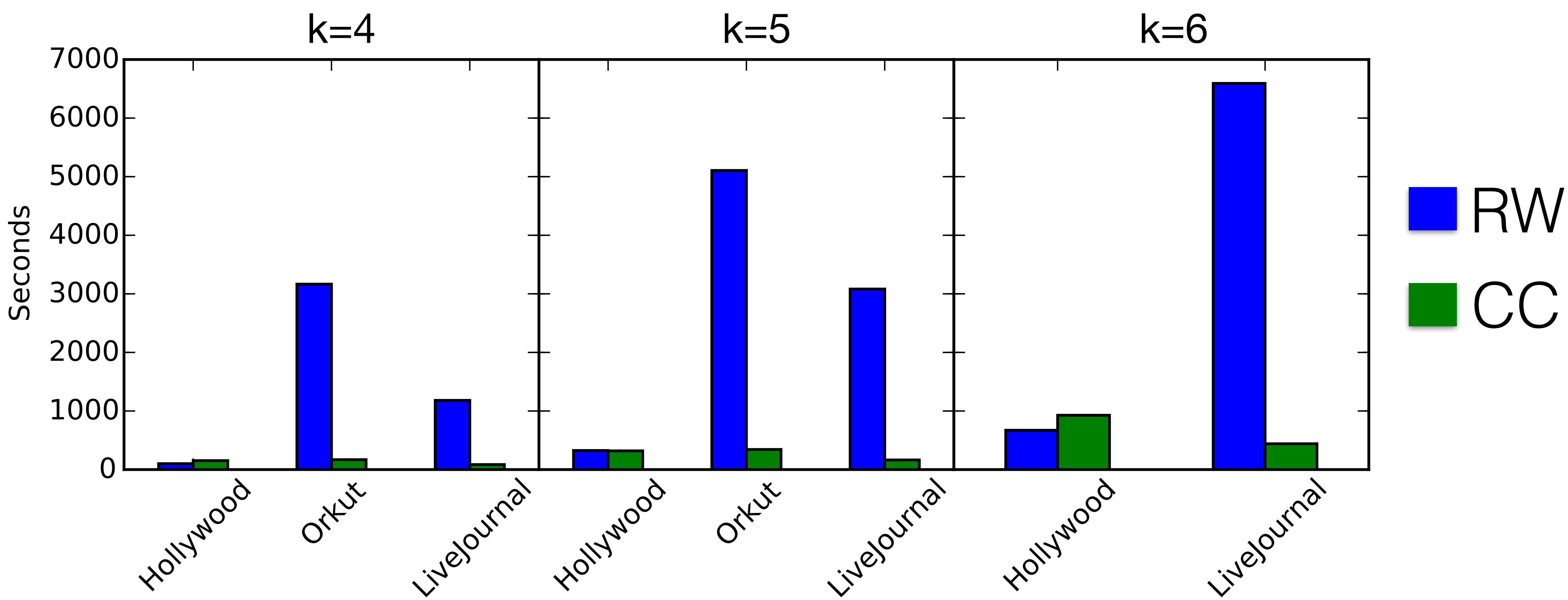
Time to get 1000 graphlet samples



Note: CC time includes preprocessing

Random Walk vs Color Coding

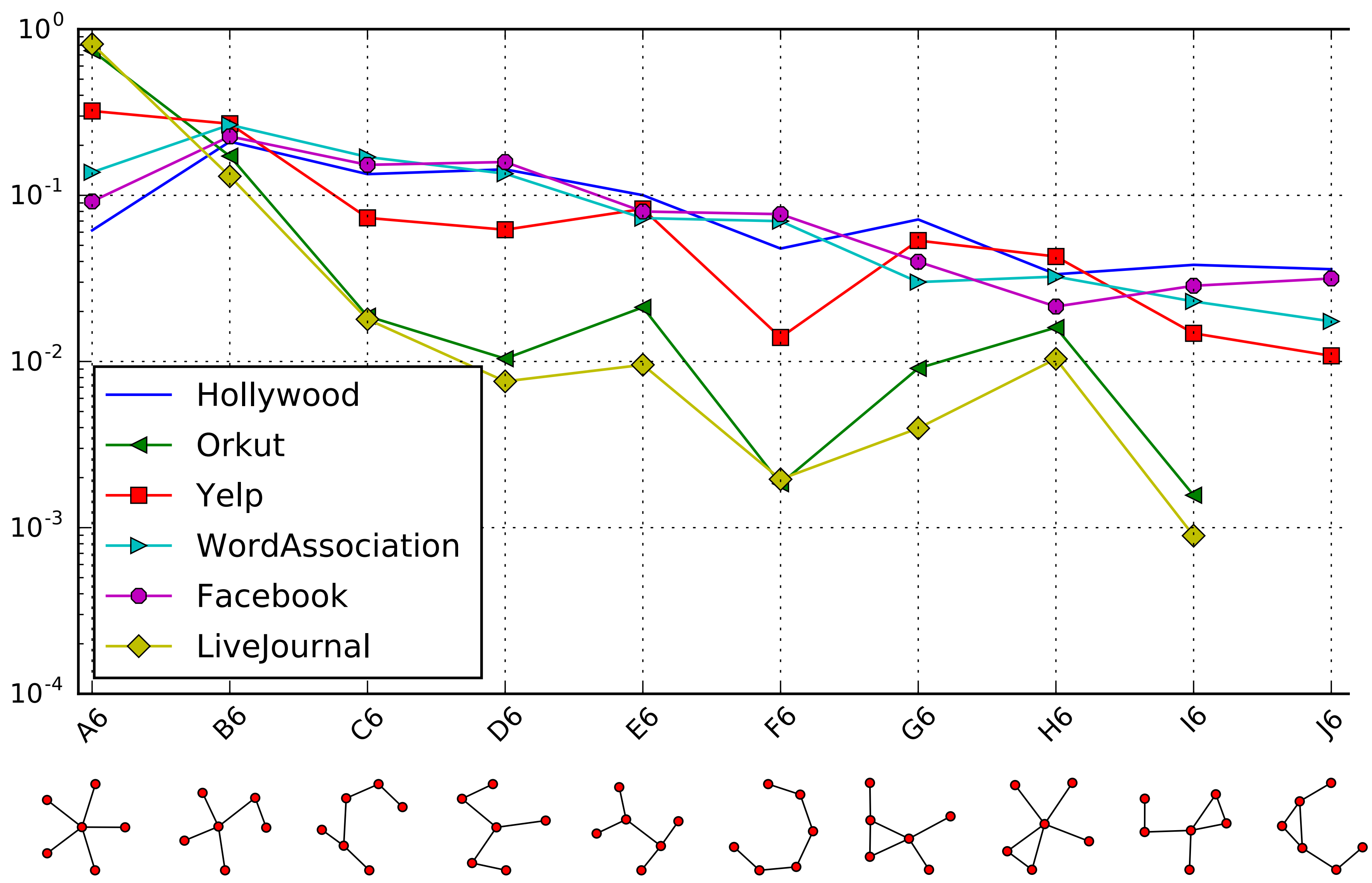
Time to get 1000 graphlet samples



Note: CC time includes preprocessing

CC required 200+GB of main memory for LJ!

The 6-graphlet distribution



Random Walks

A fine line between Efficiency and Precision

- Tiny memory footprint
- Speed and precision often in conflict

Random Walks

A fine line between Efficiency and Precision

- Tiny memory footprint
- Speed and precision often in conflict
- Understanding how much time is “enough” for a given statistical precision is often non-trivial

Random Walks

A fine line between Efficiency and Precision

Tiny memory footprint

Speed and precision often in conflict

- Understanding how much time is “enough” for a given statistical precision is often non-trivial

- *Exercise caution in using Random Walks :-)*

Thanks!