DYNAMIC NETWORKS

DYNAMIC NETWORKS

- Most real networks are dynamic
  - Facebook friendship
    - People joining/leaving
    - Friend/Unfriend
  - Twitter mention network
    - Each mention has a timestamp
    - If aggregated: every day/month/year, #messages change
  - Soccer pass network (during match, between match)
  - Internet infrastructure
  - …

DYNAMIC NETWORKS

- Most real world networks evolve
  - Nodes can appear/disappear
  - Edges can appear/disappear
  - Nature of relations can change
- How to represent those changes?
- How to manipulate dynamic networks?

DYNAMIC NETWORKS

Semantic level

Relations	Interactions

Long term

-Friend
-Colleague
-Family relation
-…

Short term ?

-Collaborators in the same project
-Same team in a game
-Attendees of the same meeting
-…

Instantaneous

-e-mail
-Text message
-Co-authoring
…

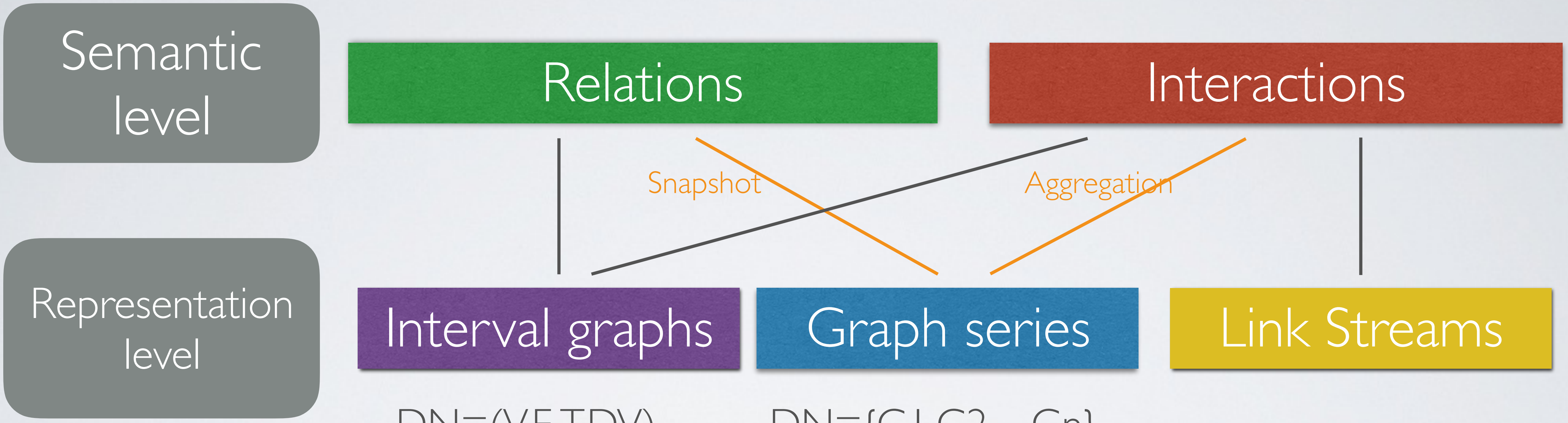
With duration

-Phone call
-Discussion in real life
-Participate in a same meeting

DYNAMIC NETWORKS

Semantic level	Relations		Interactions
Representation level	Interval graphs	Graph series	Link Streams
	DN=(V,E,T,DV) DV:VxTxT E:VxVxTxT	DN={G1,G2…Gn} Gi=(V,E) E :VxV	DN=(V,E,T) E:VxVxT

DYNAMIC NETWORKS

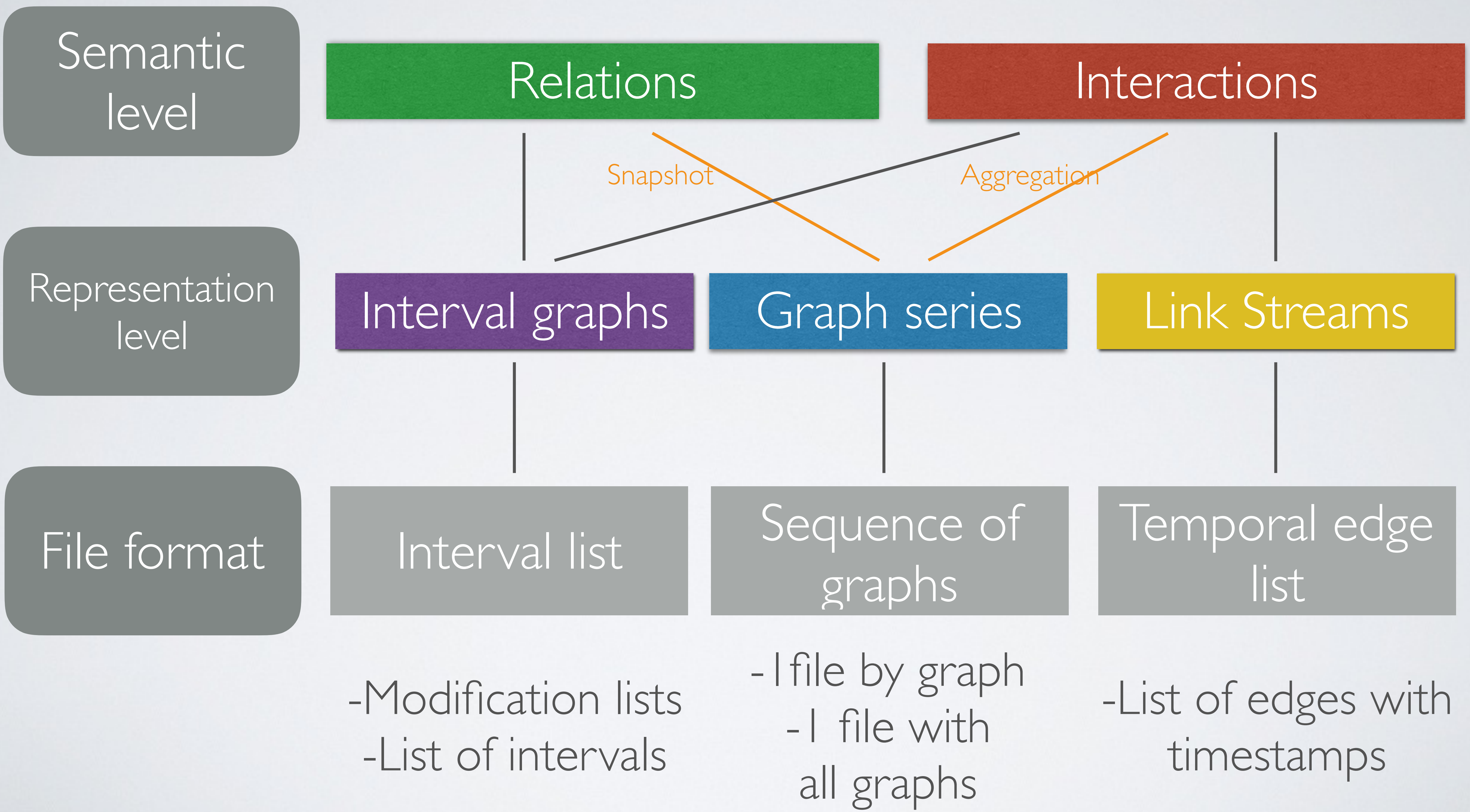


$DN=(V,E,T,DV)$
$DV: V \times T \times T$
$E: V \times V \times T \times T$

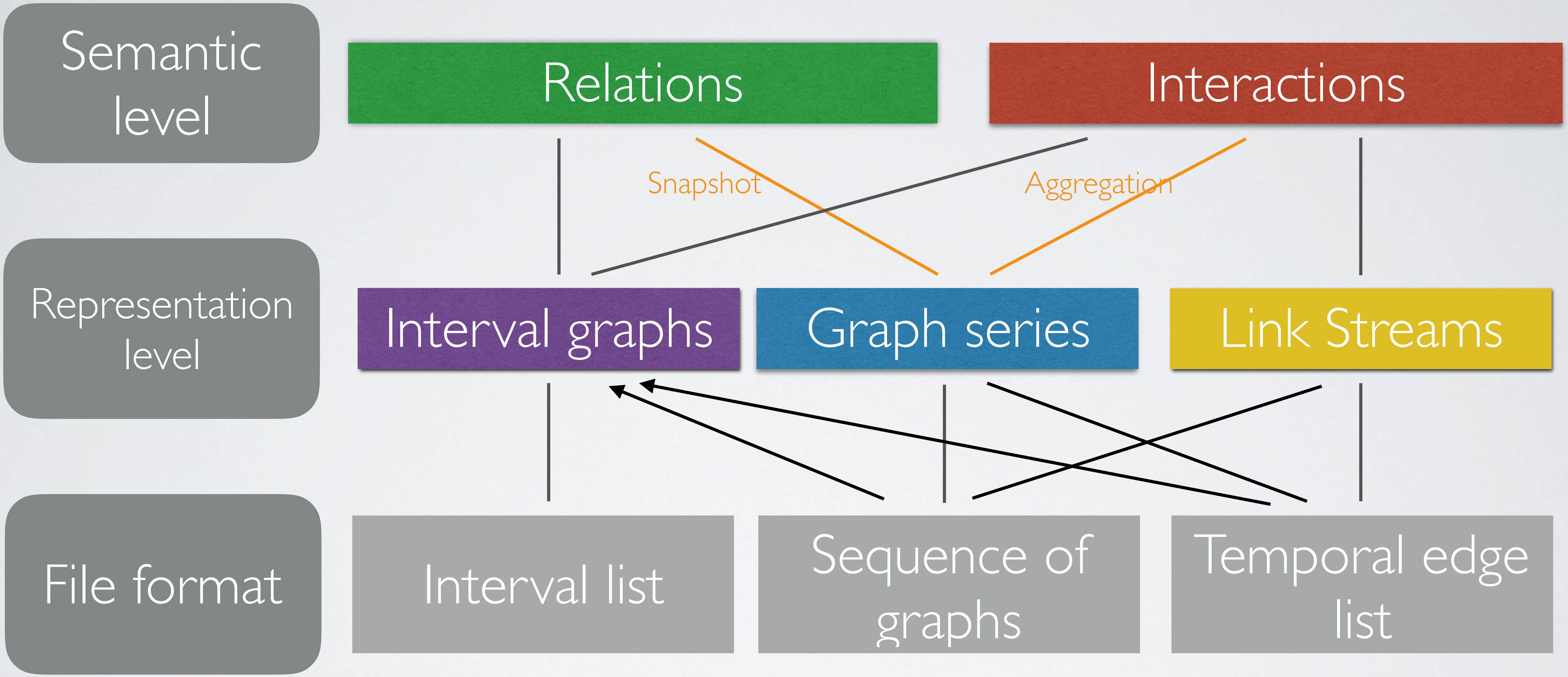
$DN=\{G1,G2\ldots Gn\}$
$Gi=(V,E)$
$E: V \times V$

$DN=(V,E,T)$
$E: V \times V \times T$

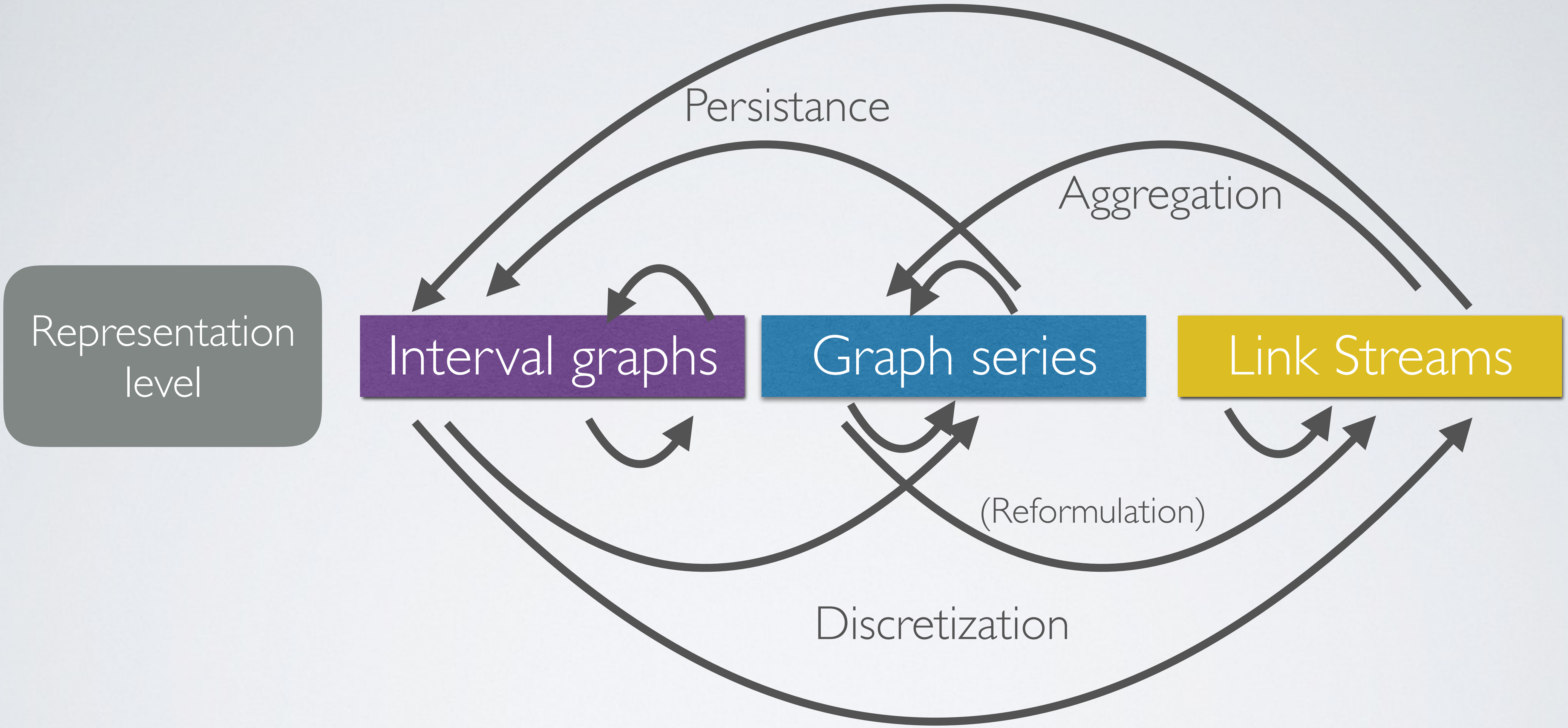
DYNAMIC NETWORKS



DYNAMIC NETWORKS



DYNAMIC NETWORKS



DYNAMIC NETWORKS

- Exemple in practice: Sociopattern dataset
  - Every 20s, list of individuals at distance ≈ 1,5m
  - Dataset : **sequence of graphs** or **temporal edge list**

1353304100	1148	1644
1353304100	1613	1672
1353304100	656	682
1353304100	1632	1671
1353304120	1492	1613
1353304120	656	682
1353304120	1632	1671
1353304140	1148	1644
1353304160	656	682
1353304160	1108	1601
1353304160	1632	1671
1353304160	626	698

DYNAMIC NETWORKS

Interval Graph

Graph series

Link stream

start=end

1353304100	1148	1644
1353304100	1613	1672
1353304100	656	682
1353304100	1632	1671
1353304120	1492	1613
1353304120	656	682
1353304120	1632	1671
1353304140	1148	1644

DYNAMIC NETWORKS

Interval Graph

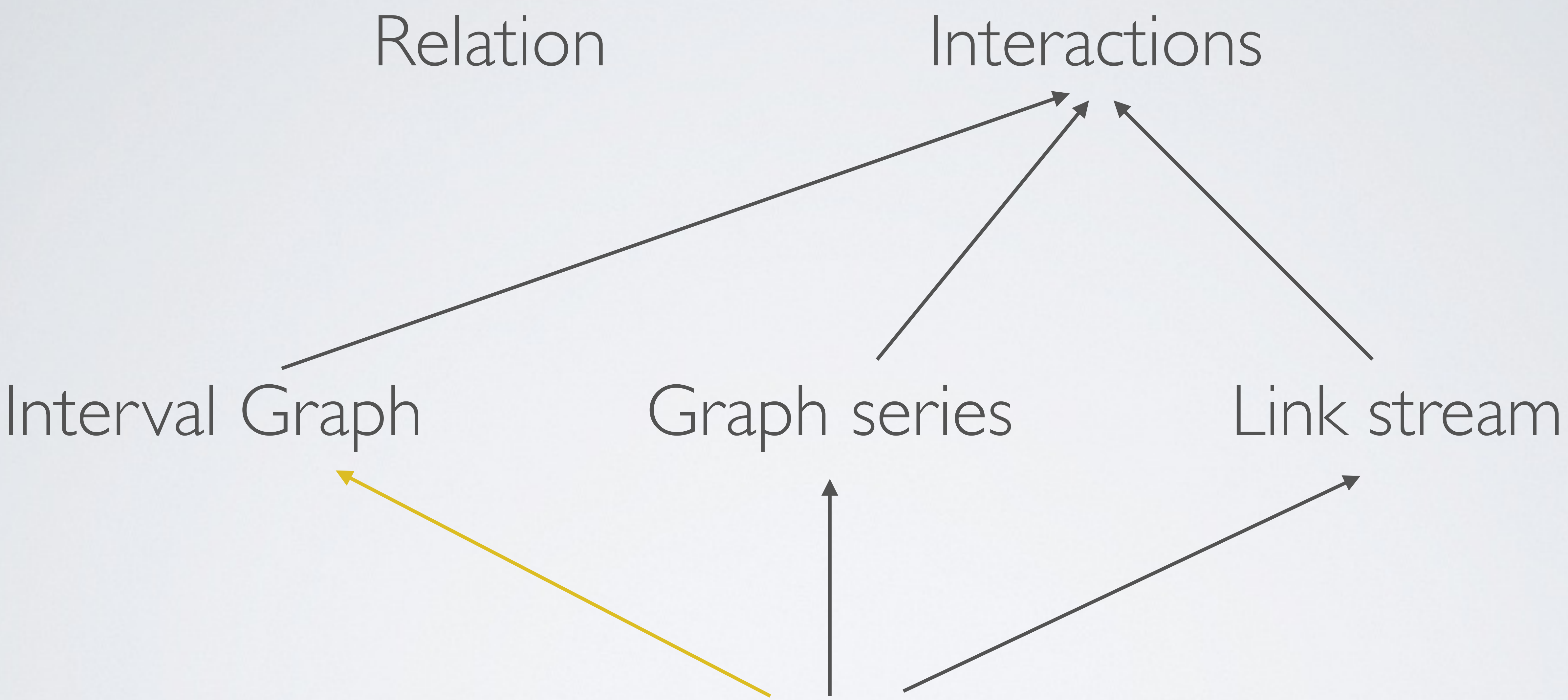
Graph series

Link stream

Persistant link
=interval

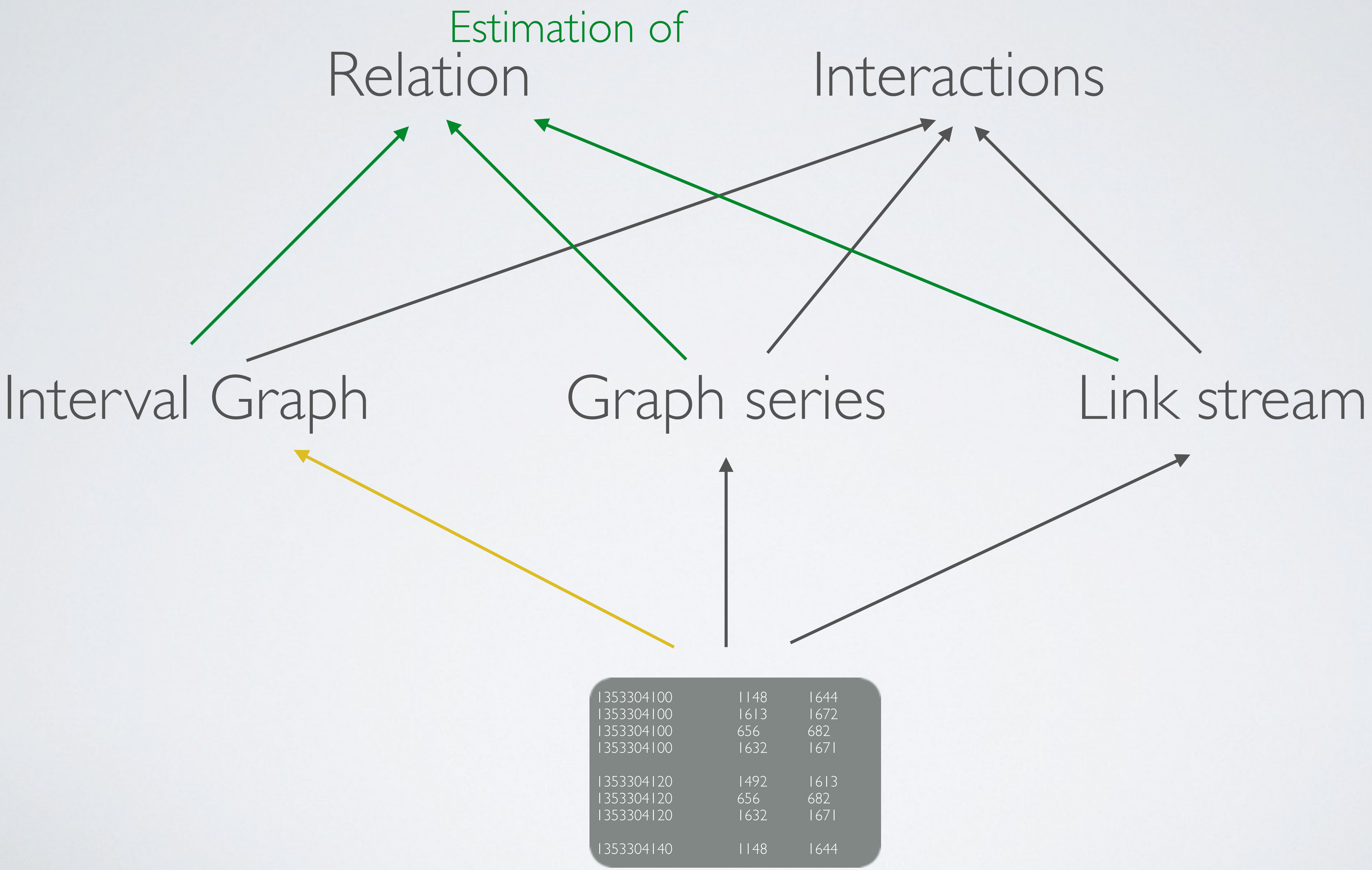
1353304100	1148	1644
1353304100	1613	1672
1353304100	656	682
1353304100	1632	1671
1353304120	1492	1613
1353304120	656	682
1353304120	1632	1671
1353304140	1148	1644

DYNAMIC NETWORKS

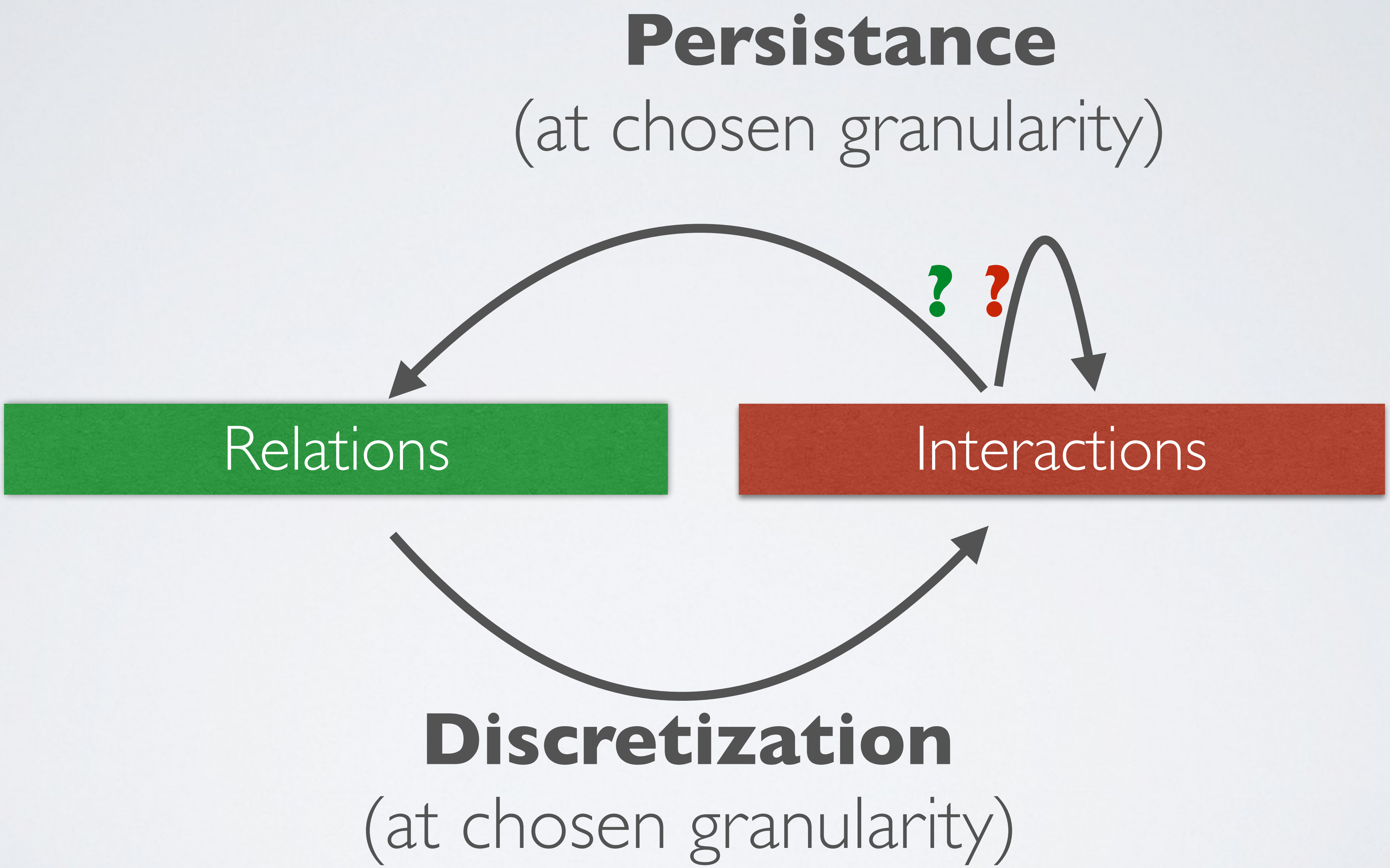


1353304100	1148	1644
1353304100	1613	1672
1353304100	656	682
1353304100	1632	1671
1353304120	1492	1613
1353304120	656	682
1353304120	1632	1671
1353304140	1148	1644

DYNAMIC NETWORKS



DYNAMIC NETWORKS



DYNAMIC NETWORKS

To sum up

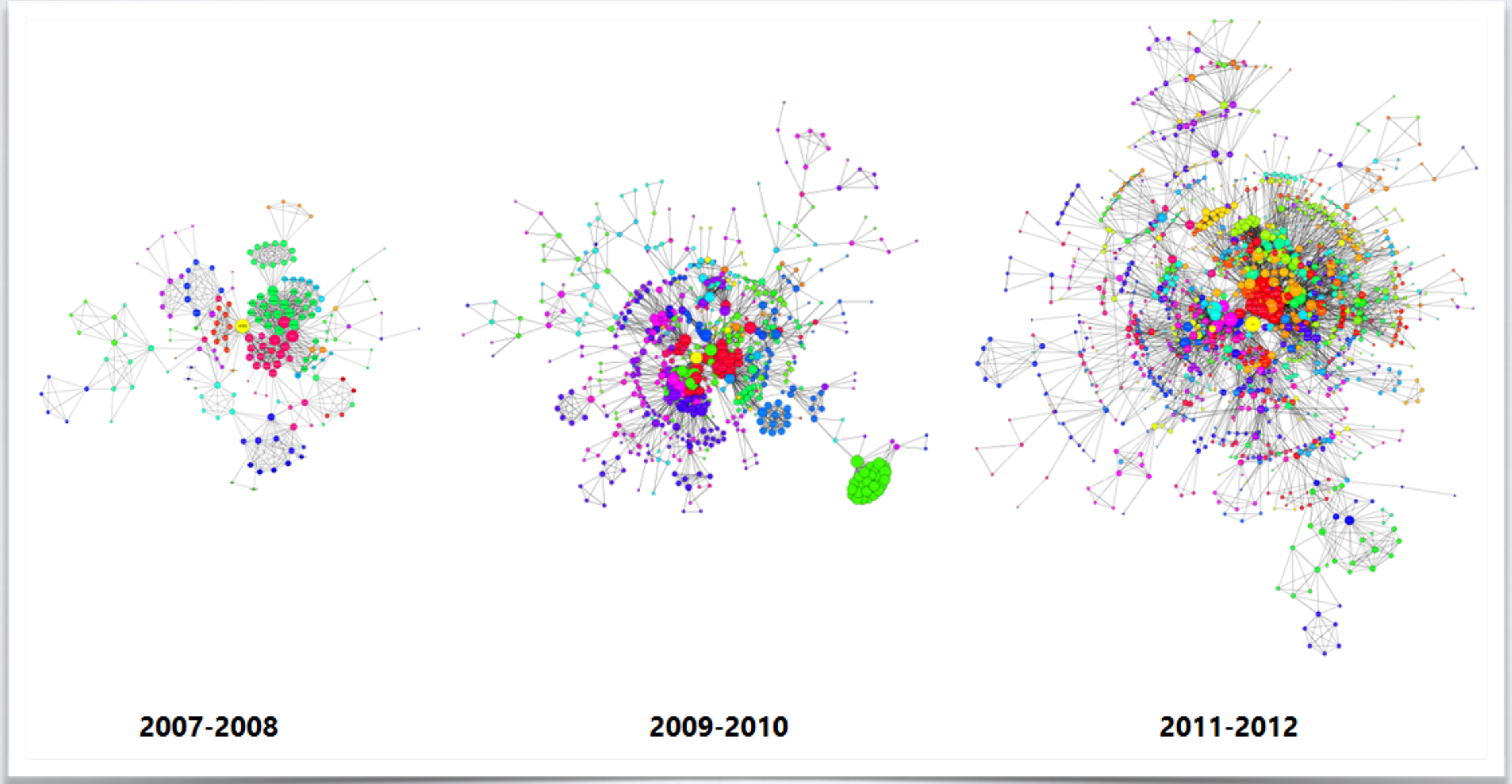
- Collected dataset of relations (Facebook “friends”, affiliations, etc.)
  - Can be discretized for analysis if needed
- Collected dataset of interactions (e-mail, face to face, etc)
  - Can be aggregated in series of graphs
  - Can be transformed in intervals (relations or lasting interactions)
  - Can be studied as it is

ANALYZING DYNAMIC NETWORKS

UNSTABLE SNAPSHOTS

UNSTABLE SNAPSHOTS

- The evolution is represented as a series of *a few* snapshots.
- Many changes between snapshots
  - Cannot be plotted as a "movie"



UNSTABLE SNAPSHOTS

- Each snapshot can be studied as a static graph
- The evolution of the properties can be studied "manually"
- "Node X had low centrality in snapshot t and high centrality in snapshot t+n"

STABLE NETWORKS

STABLE NETWORK

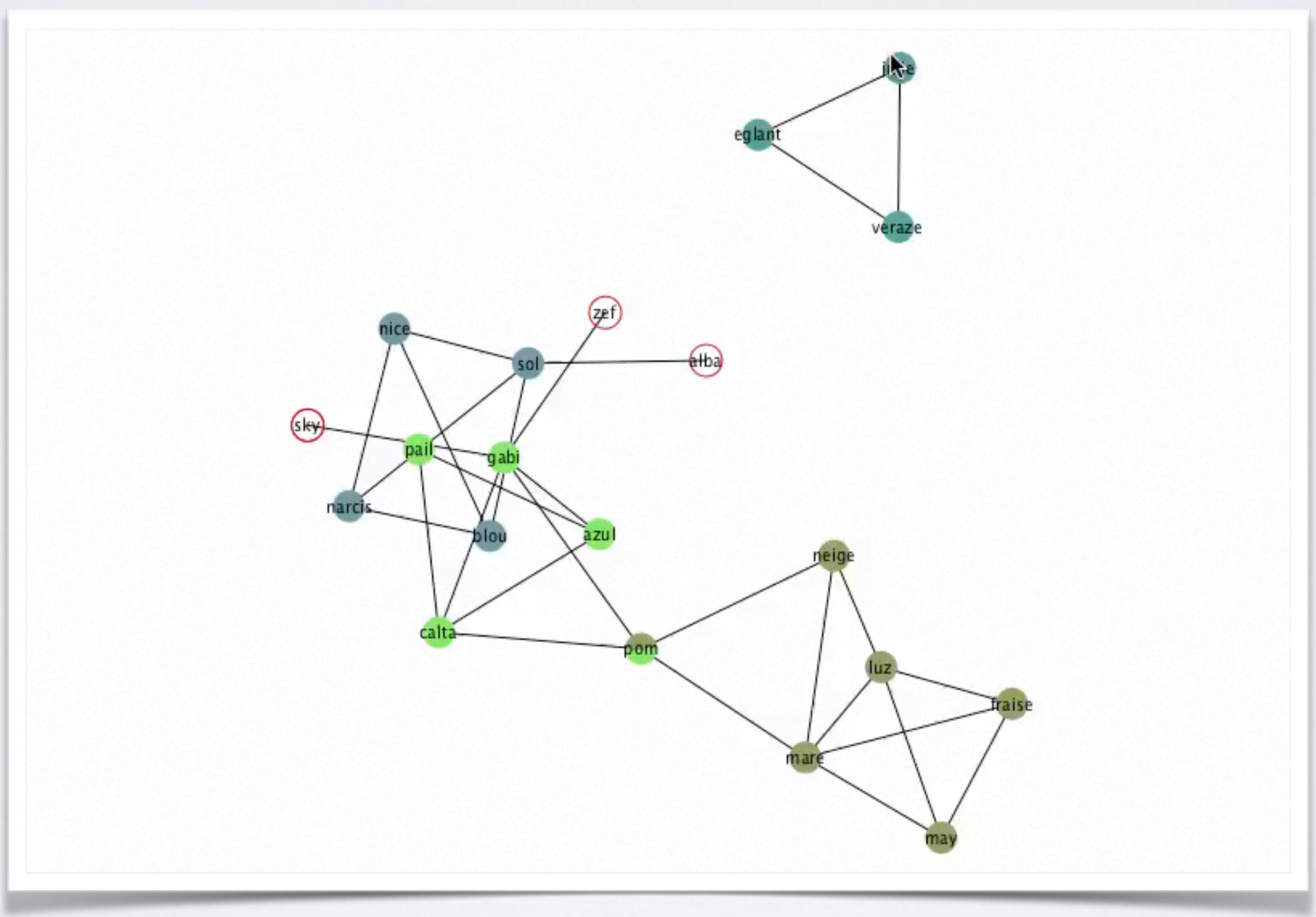
- Edges correspond to relations
  - They change slowly
  - The network is well defined at any t
    - Temporal network: nodes/edges described by (long lasting) intervals
    - Enough snapshots to track nodes

- A static analysis at every (relevant) t gives a dynamic vision

- No formal distinction with previous case

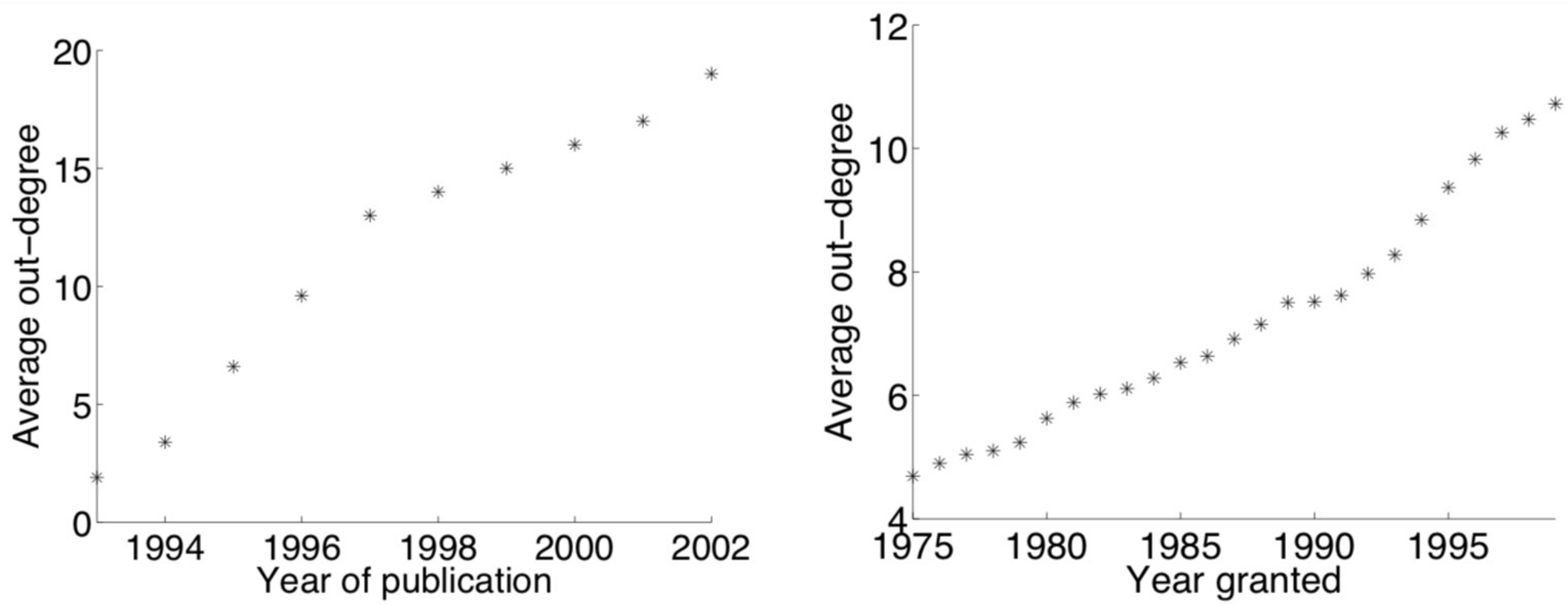
STABLE NETWORK

- Visualization
  - Problem of stability of node positions

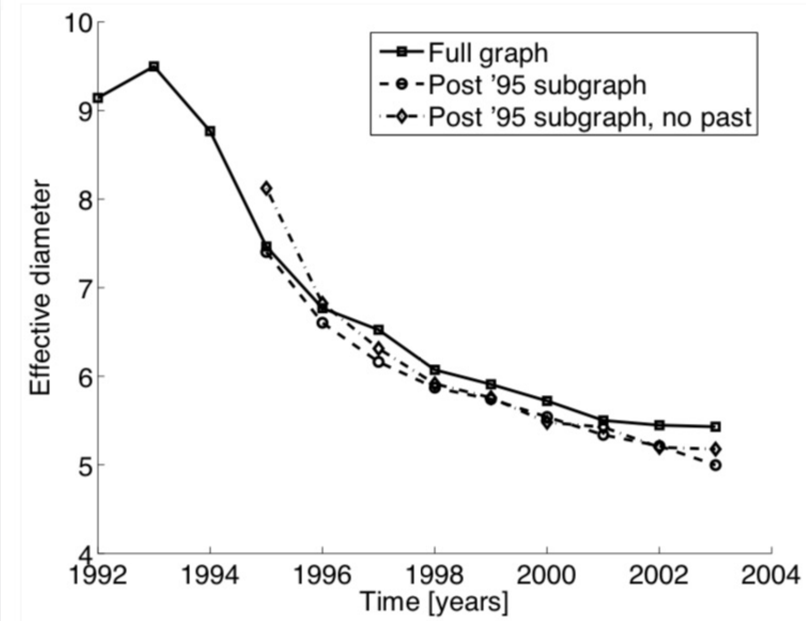


STABLE NETWORK

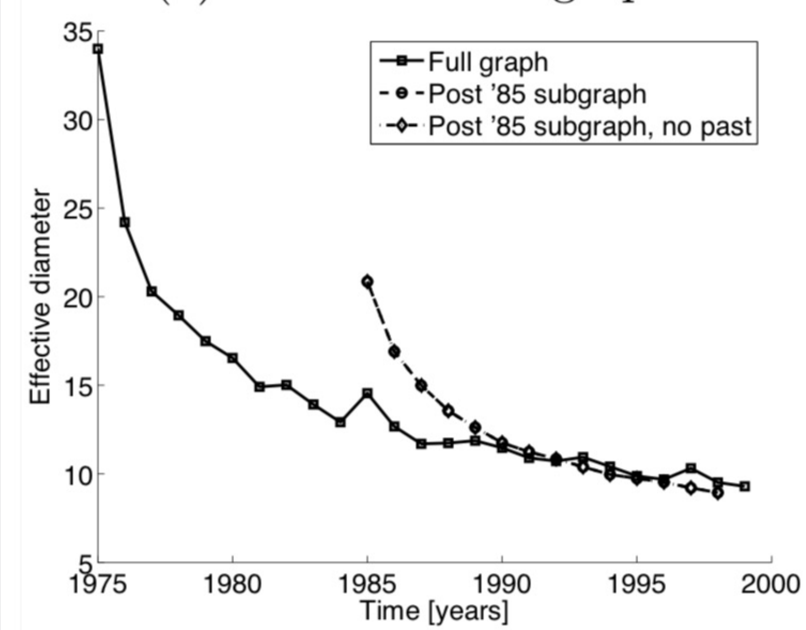
- Global graph properties



(a) arXiv (b) Patents



(a) arXiv citation graph

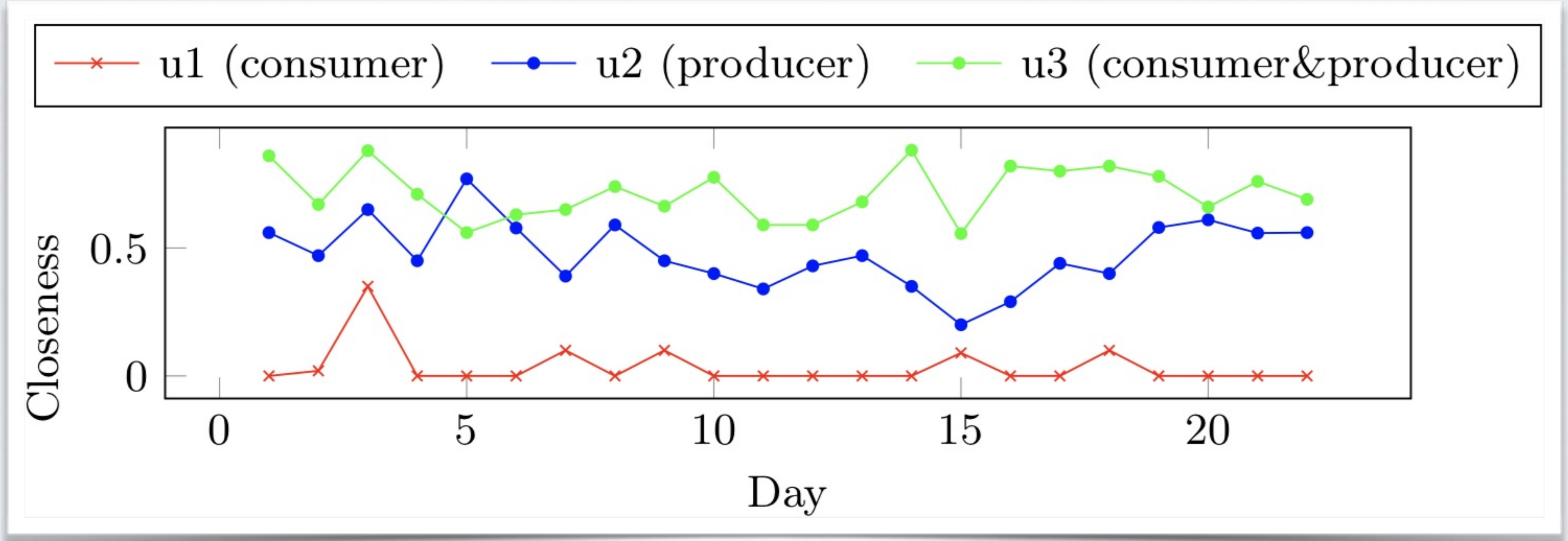


(c) Patents citation graph

[Leskovec: Graph Evolution: Densification and Shrinking Diameters]

STABLE NETWORK

- Centralities



TIME SERIES ANALYSIS

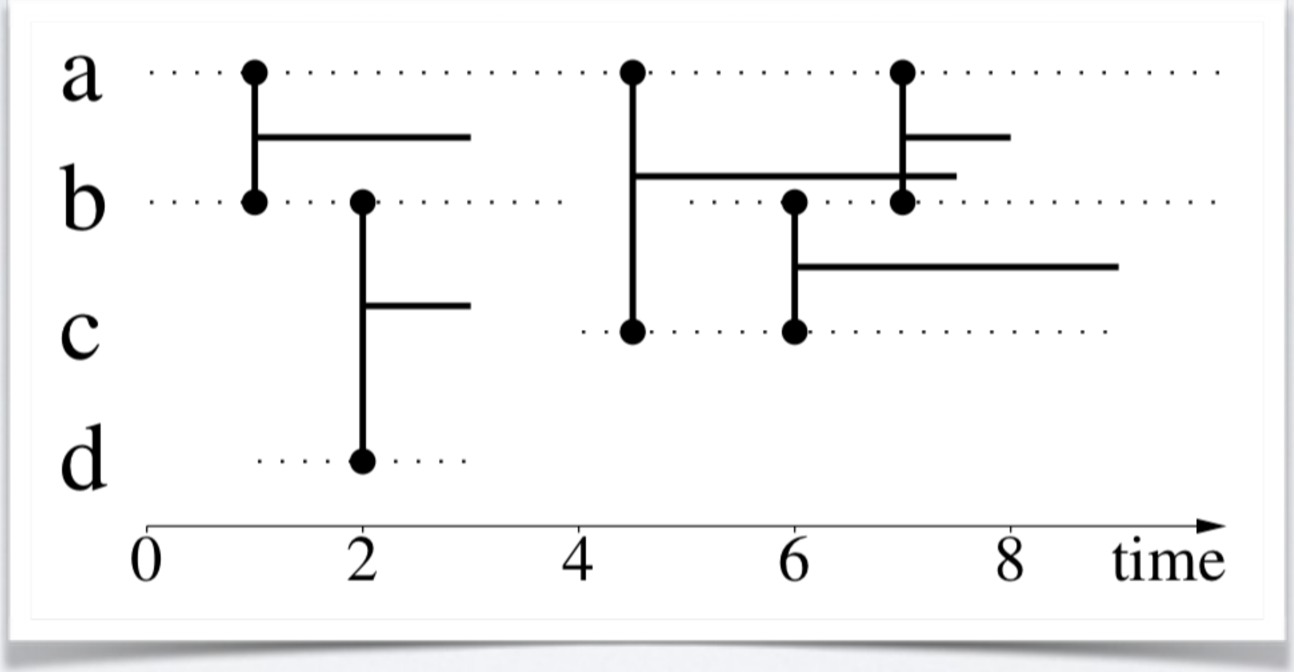
- TS analysis is a large field of research
- Time series: evolution of a value over time
  - Very popular for stock market analysis…
- Killer applications:
  - Detection of periodic patterns
  - Detection of anomalies
  - Identification of global trends
  - Evaluation of auto-correlation
  - Prediction of future values
- e.g. ARIMA (Autoregressive integrated moving average)
  - *From statsmodels.tsa.arima_model import ARIMA*

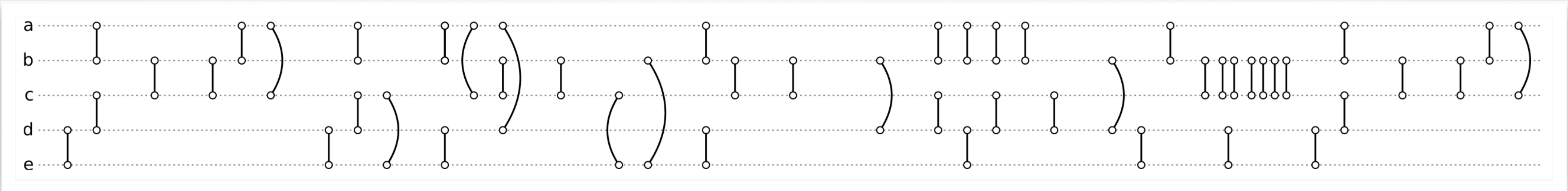
UNSTABLE TEMPORAL NETWORK

UNSTABLE TEMPORAL NETWORK

- Temporal Network of interactions
  - Link stream (twitter messages,…)
  - Interval graph (Physical interactions, …)
- The network at a given *t* is not meaningful
- How to analyze such a network?

UNSTABLE TEMPORAL NETWORK





UNSTABLE TEMPORAL NETWORK

- Until recently, network was transformed using snapshot/sliding windows
- Tools developed to deal with such networks

UNSTABLE TEMPORAL NETWORK

- [Holme 2012]: mostly about paths, walks, distances…
- [Latapy 2018]: Other things (centralities, …)
- Idea: Generalize all graphs definitions to temporal networks
- => If all nodes and all edges always present, same values as for a static graph

CENTRALITIES
&
NETWORK PROPERTIES
IN STREAM GRAPHS

INDICES IN STREAM GRAPHS

$$\textbf{stream graph } S = (T, V, W, E)$$

T: Possible Time

V: vertices

W: Vertices presence time

E: Edges presence time

INDICES IN STREAM GRAPHS

Number of nodes:

$$\frac{\text{Total presence of nodes}}{\text{Total dataset duration}}$$

$$n = \sum_{v \in V} n_v = \frac{|W|}{|T|}$$

(not an integer value…)

e.g.: 2 if 4 nodes
half the time

INDICES IN STREAM GRAPHS

Number of edges:

$$\frac{\text{Total presence of edges}}{\text{Total dataset duration}}$$

$$m = \sum_{uv \in V \otimes V} m_{uv} = \frac{|E|}{|T|}$$

(not an integer value…)

e.g.: 1 if 1 edge all the time

INDICES IN STREAM GRAPHS

Neighborhood of a node

$$N(v) = \{(t, u), (t, uv) \in E\}$$

Degree of a node

$$d(v) = \frac{|N(v)|}{|T|} = \sum_{u \in V} \frac{|T_{uv}|}{|T|}$$

INDICES IN STREAM GRAPHS

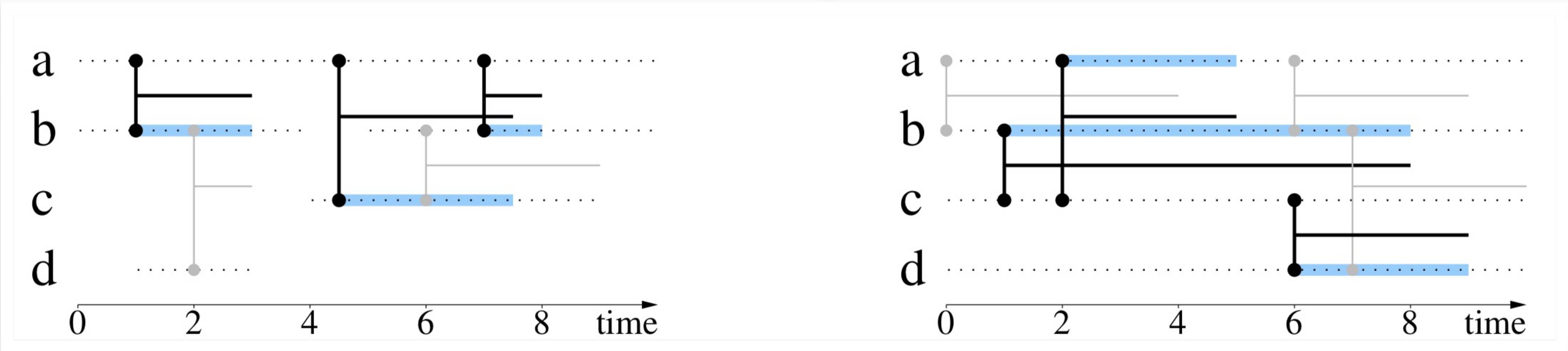


Figure 5: **Two examples of neighborhoods and degrees of nodes.** We display in black the links involving the node under concern, and in grey the other links. Left: $N(a) = ([1,3] \cup [7,8]) \times \{b\} \cup [4.5, 7.5] \times \{c\}$ is in blue, leading to $d(a) = \frac{3}{10} + \frac{3}{10} = 0.6$. Right: $N(c) = [2,5] \times \{a\} \cup [1,8] \times \{b\} \cup [6,9] \times \{d\}$ is in blue, leading to $d(c) = \frac{13}{10} = 1.3$.

INDICES IN STREAM GRAPHS

Average node degree

$$d(V) = \frac{1}{n} \cdot \sum_{v \in V} n_v \cdot d(v) = \sum_{v \in V} \frac{|T_v|}{|W|} \cdot d(v)$$

INDICES IN STREAM GRAPHS

Clustering coefficient of a node

$$cc(v) = \delta(N(v)) = \frac{\sum_{uw \in V \otimes V} |T_{vu} \cap T_{vw} \cap T_{uw}|}{\sum_{uw \in V \otimes V} |T_{vu} \cap T_{vw}|}$$

Probability that if we take 2 random neighbors at random time, they are linked

INDICES IN STREAM GRAPHS

$$\delta(S) = \frac{\sum\limits_{uv \in V \otimes V} |T_{uv}|}{\sum\limits_{uv \in V \otimes V} |T_u \cap T_v|} = \frac{\int\limits_{t \in T} |E_t| \, \mathrm{d}t}{\int\limits_{t \in T} |V_t \otimes V_t| \, \mathrm{d}t}$$

Density: probability if we take a random pair of nodes
at random time that there is an edge

INDICES IN STREAM GRAPHS

$$\delta(S) = \frac{\sum_{uv \in V \otimes V} |T_{uv}|}{\sum_{uv \in V \otimes V} |T_u \cap T_v|} = \frac{\int_{t \in T} |E_t| \,\mathrm{d}t}{\int_{t \in T} |V_t \otimes V_t| \,\mathrm{d}t}$$

Total edge presence

e.g.: 10 if 2 edges present over 5 periods

INDICES IN STREAM GRAPHS

$$\delta(S) = \frac{\sum_{uv \in V \otimes V} |T_{uv}|}{\sum_{uv \in V \otimes V} |T_u \cap T_v|} = \frac{\int_{t \in T} |E_t| \, \mathrm{d}t}{\int_{t \in T} |V_t \otimes V_t| \, \mathrm{d}t}$$

Total **overlapping time** between each pair of nodes

=>An edge is possible

INDICES IN STREAM GRAPHS

Figure 2: **Two stream graphs with $n = 2$ nodes, $m = 1$ link, but with different densities:** Left: $\delta = 0.75$. Right: $\delta = 1$.

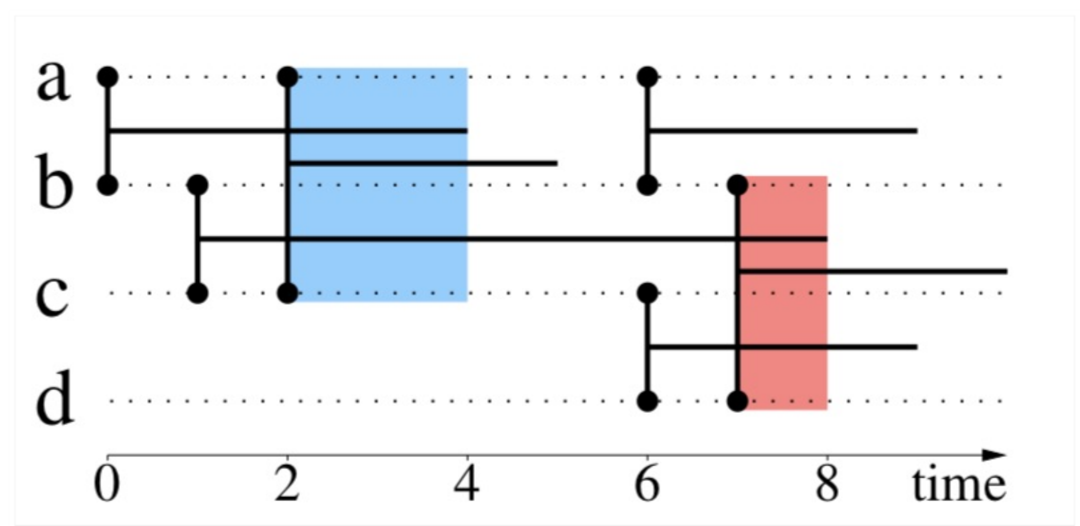
INDICES IN STREAM GRAPHS

- Note that we can define particular cases of density:
  - Density of a pair of nodes
  - Density of a node

$$\delta(uv) = \frac{|T_{uv}|}{|T_u \cap T_v|}, \quad \delta(v) = \frac{\sum_{u \in V, u \neq v} |T_{uv}|}{\sum_{u \in V, u \neq v} |T_u \cap T_v|}$$

INDICES IN STREAM GRAPHS

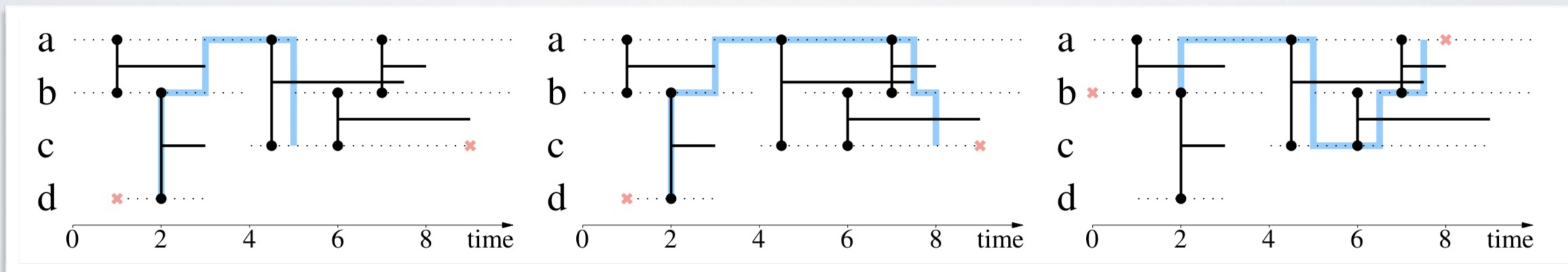
A clique of graph G is a cluster C of G of density 1. In other words, all pairs of nodes involved in C are linked together in G. A clique C is maximal if there is no other clique C' such that $C \subset C'$.



PATHS AND DISTANCES IN STREAM GRAPHS

PATHS

- A path in a stream graphs
  - Starts at a node and a date
  - Ends at a node and a date
  - Has a length (number of hops)
  - Has a duration (duration from leaving node to reaching node)



Path(d,1)(c9)
Length:3
Duration: 3

?

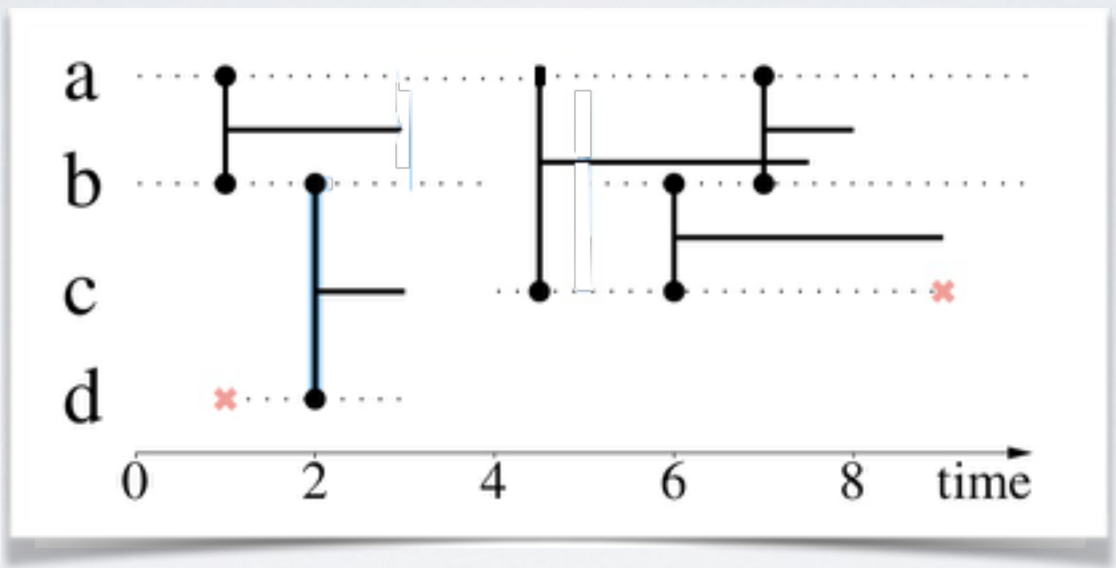
?

SHORTEST PATHS

- Several types of shortest paths in Stream graphs:
  - Shortest path: minimal length
  - Fastest path: minimal duration
  - Foremost path: first to reach

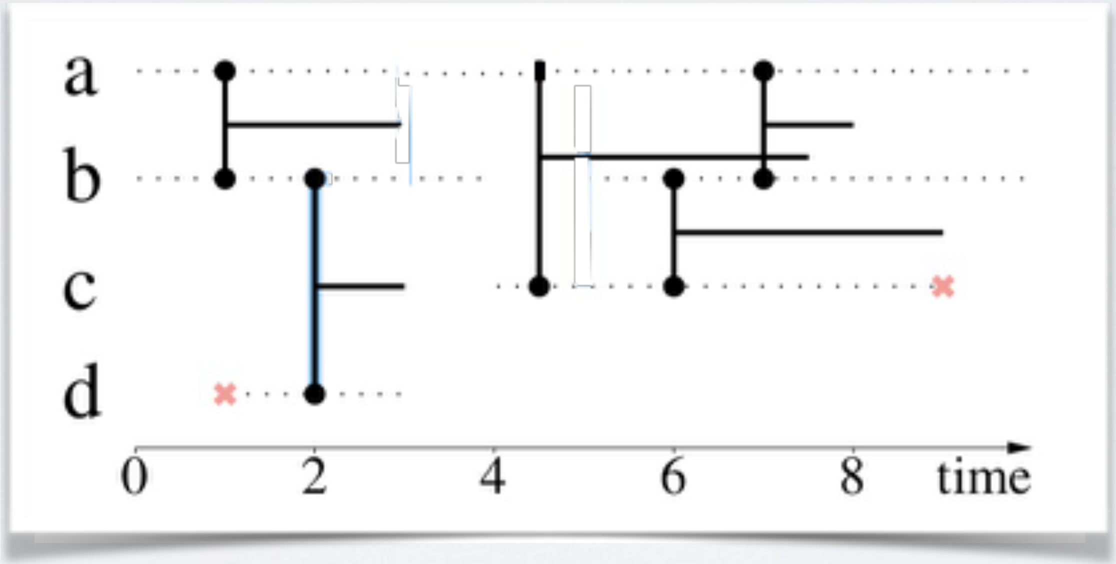
  - Fastest shortest paths
    - Minimum duration among minimal length
  - Shortest fastest paths
    - Minimal length among minimal duration

SHORTEST PATHS



Shortest paths from (1, d) to (9, c) ?

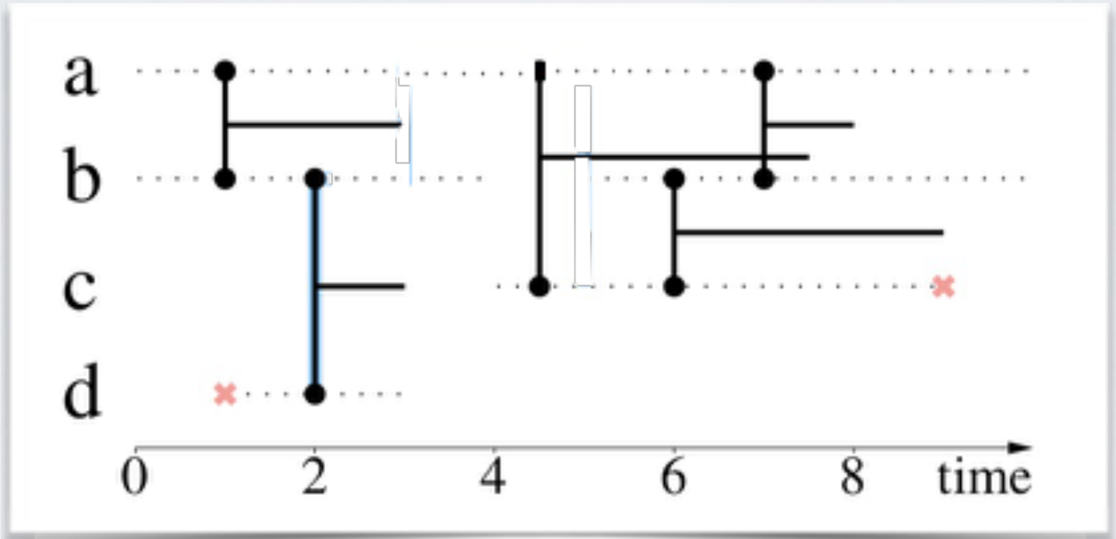
SHORTEST PATHS



Shortest paths from (1, d) to (9, c) ?

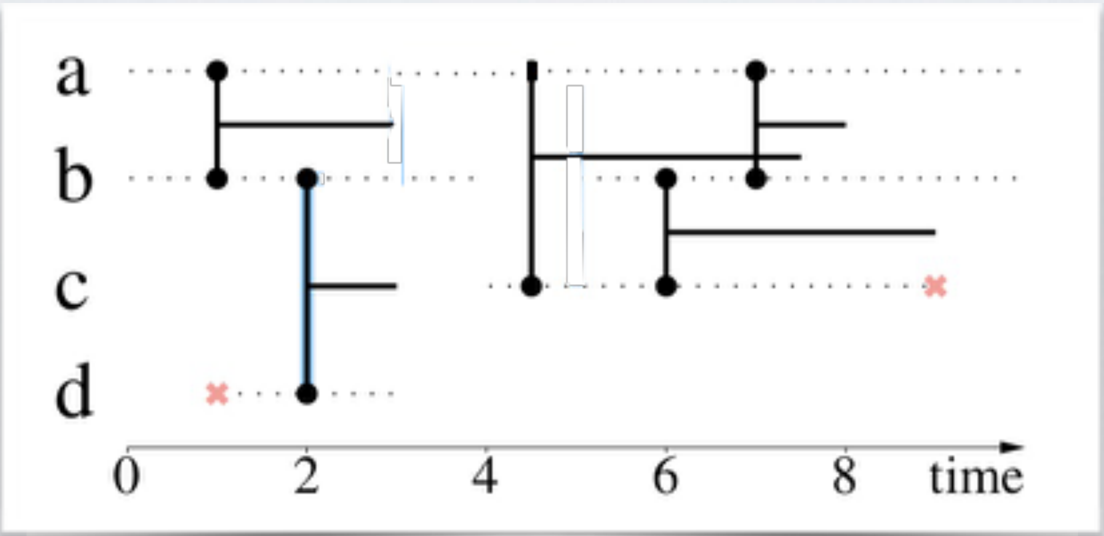
=>e.g. (2.5,d,b)(3,b,a)(7,a,c)

SHORTEST PATHS



Fastest paths from (1, d) to (9, c) ?

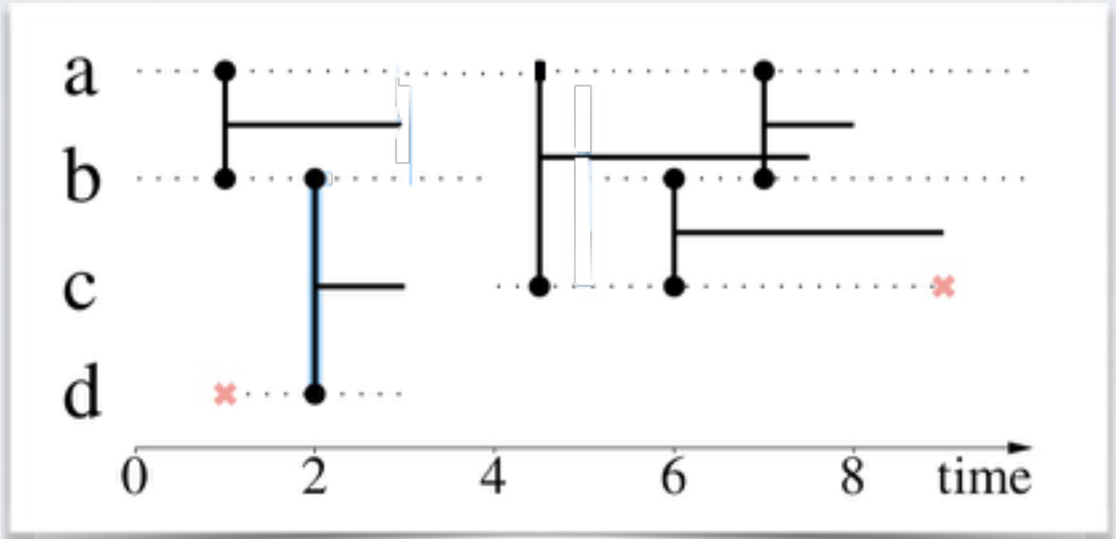
SHORTEST PATHS



Fastest paths from (1, d) to (9, c) ?

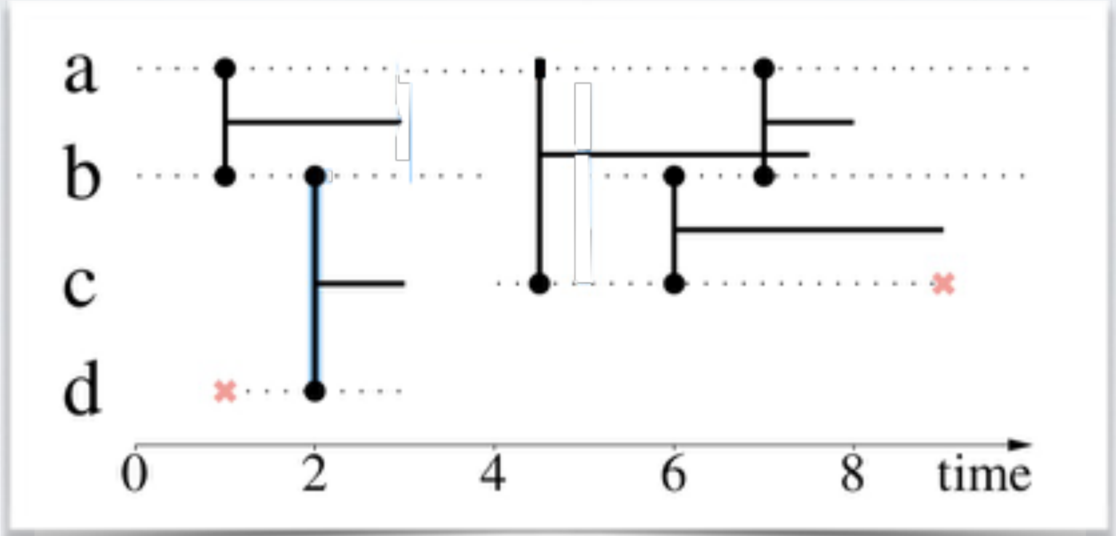
(3,d,b),(3,b,a),(4.5,a,c)

SHORTEST PATHS



Foremost paths from (0, a) to (9, c) ?

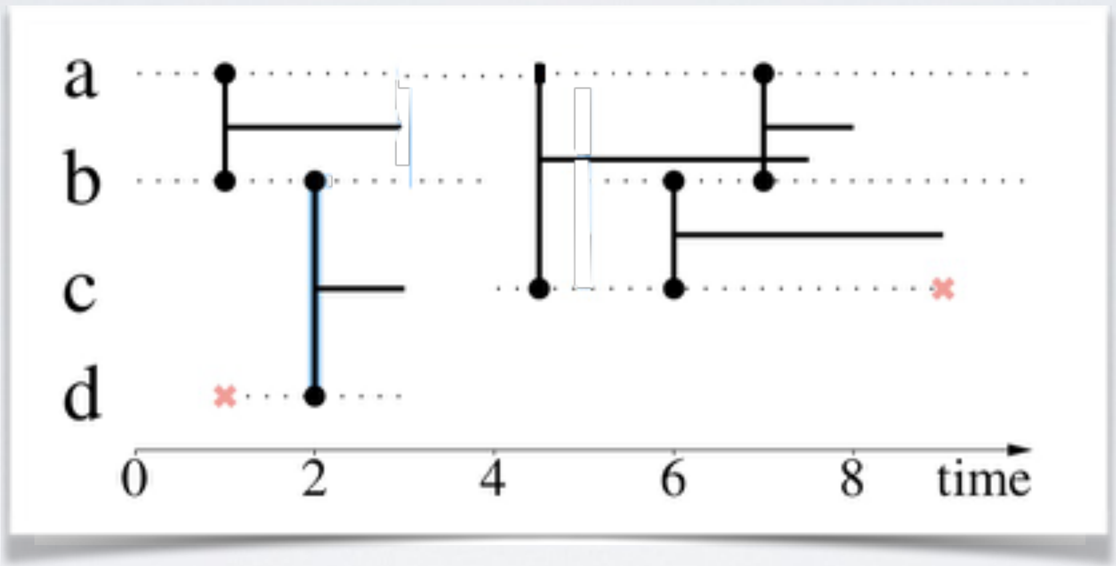
SHORTEST PATHS



Foremost paths from (0, a) to (9, c) ?

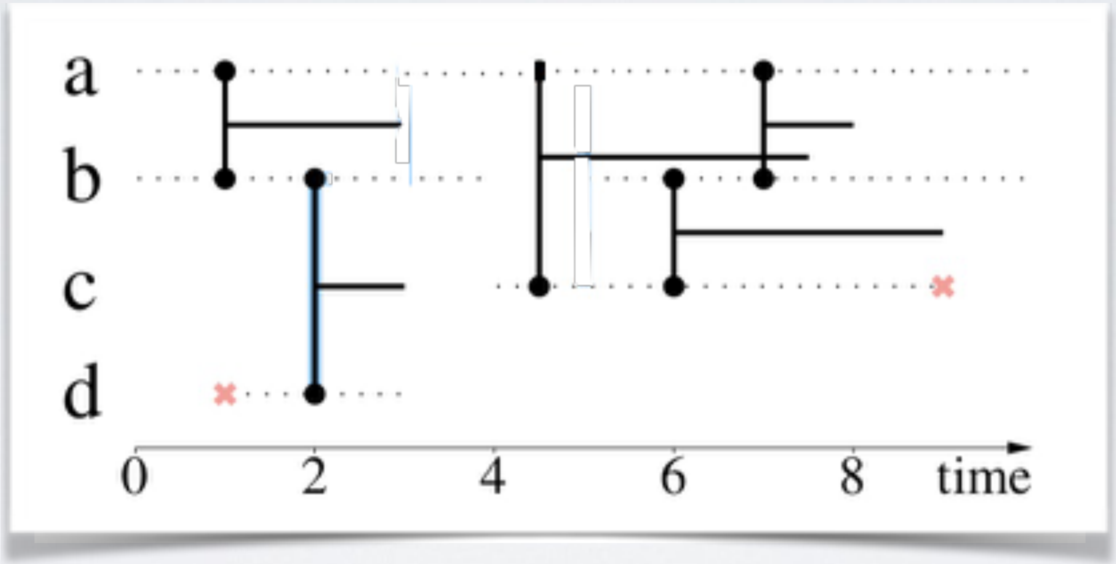
…(4.5,a,c)

SHORTEST PATHS



Fastest shortest path from (1, d) to (9, c) ?

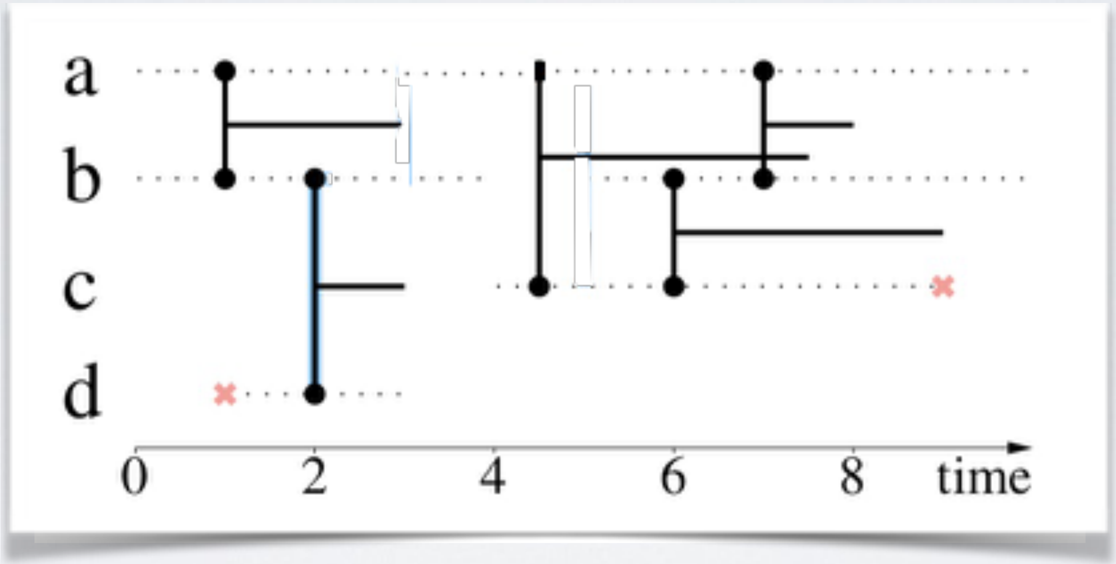
SHORTEST PATHS



Fastest shortest path from (1, d) to (9, c) ?

(3, d, b), (3, b, a), (4.5, a, c)

SHORTEST PATHS

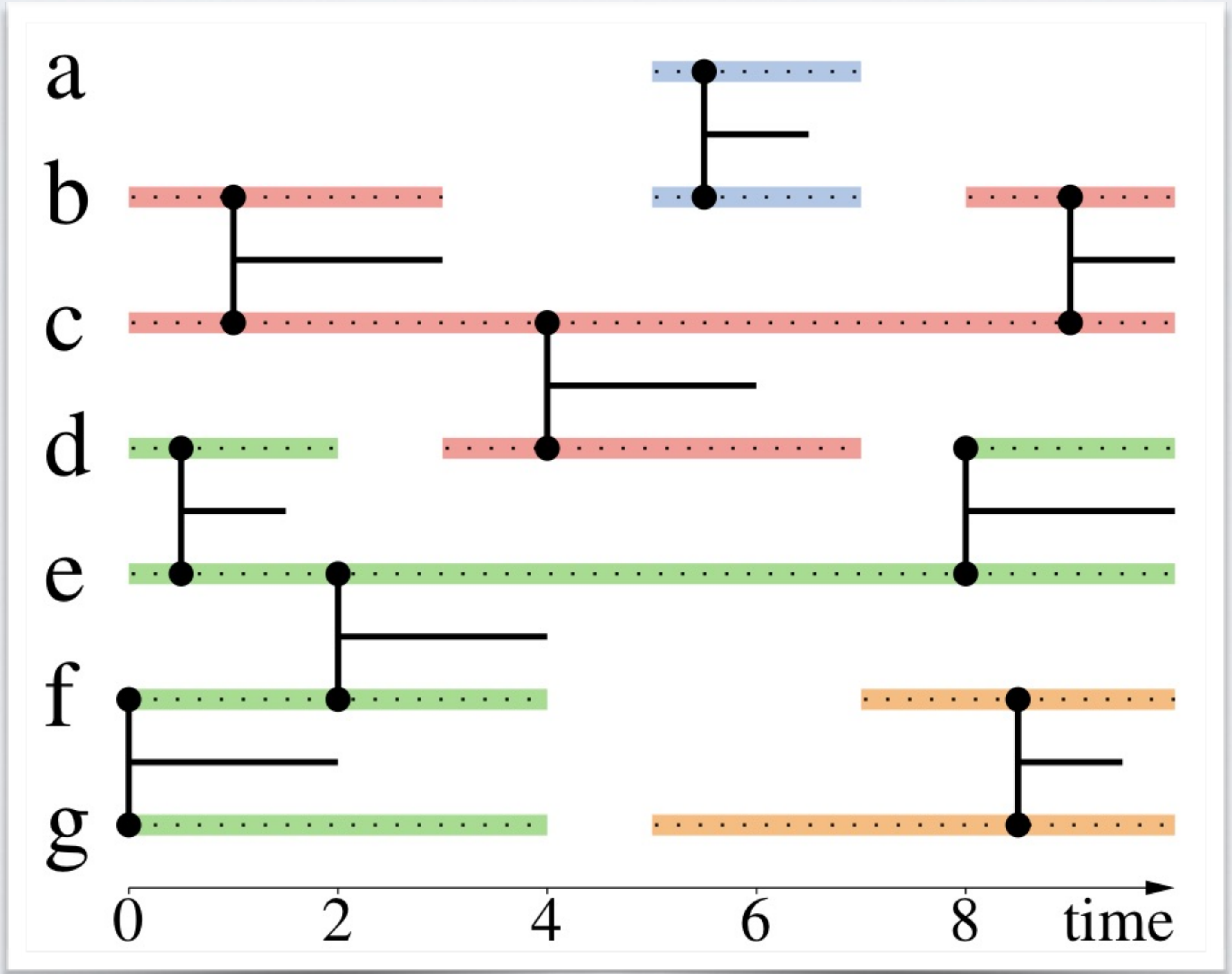


Shortest Fastest path from (1, d) to (9, c) ?

OTHER DEFINITIONS ON STREAM GRAPHS

CONNECTED COMPONENTS

- Weakly connected component:
  - There is at least a non-temporally respecting path



CLOSENESS - BETWEENESS

$$\mathcal{C}_t(v) = \sum_{u \in V} \int_{\substack{s \in T \\ (s,u) \neq (t,v)}} \frac{1}{c_t(v, (s, u))} \, \mathrm{d}s$$

$$\mathcal{B}(t, v) = \sum_{u \in V, w \in V} \int_{i \in T_u, j \in T_w} \frac{\sigma((i, u), (j, w), (t, v))}{\sigma((i, u), (j, w))} \, \mathrm{d}i \, \mathrm{d}j$$

RANDOM MODELS FOR DYNAMIC NETWORKS

RANDOM MODELS

- In many cases, in network analysis, useful to compare a network to a randomized version of it
  - Clustering coefficient, assortativity, modularity, …
- In a static graph, 2 main choices:
  - Keep only the number of edges (ER model)
  - Keep the number of edges and the degree of nodes (Configuration model)
- In dynamic networks, it is more complex…

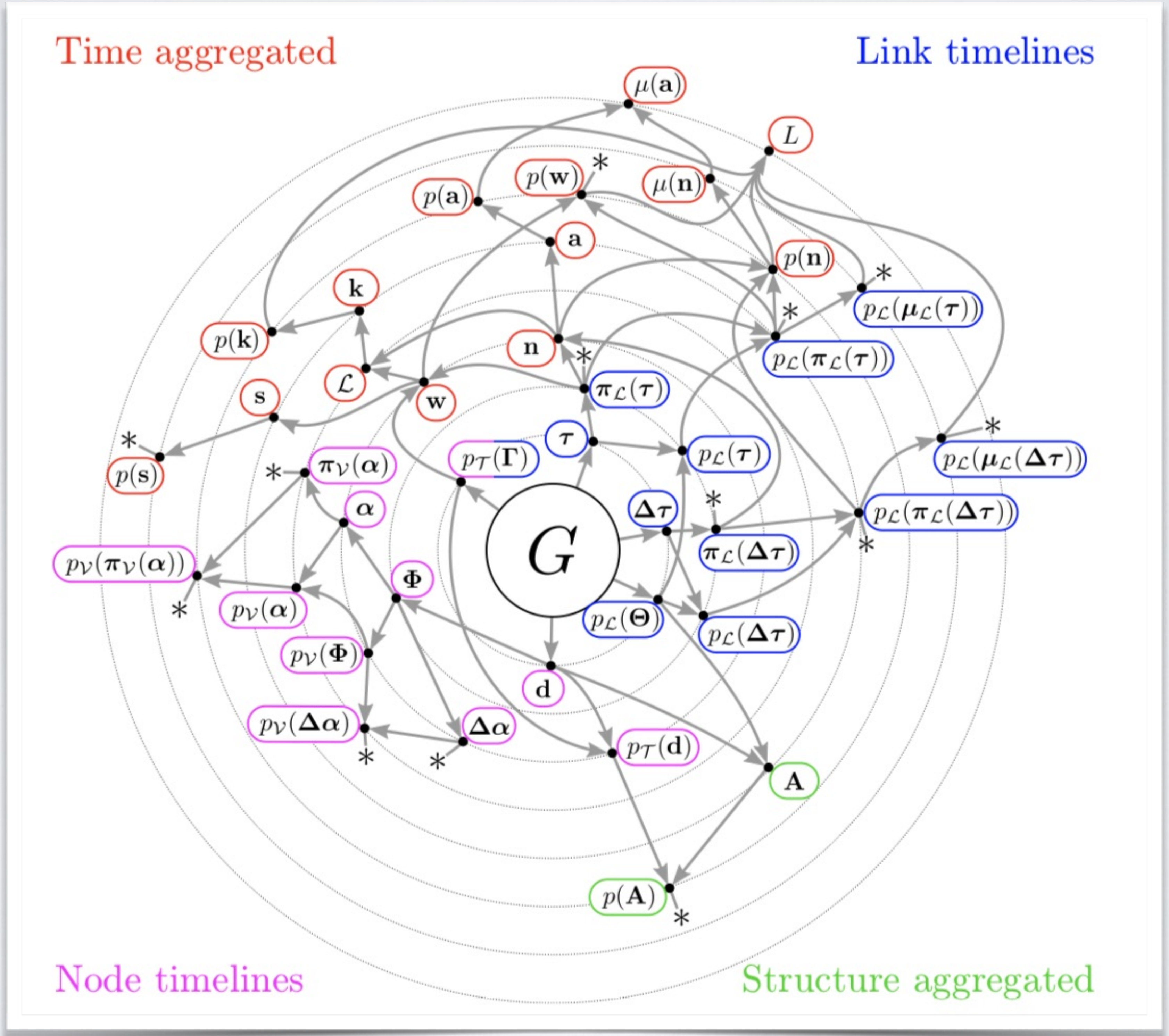
RANDOM MODELS

- [Gauvin 2018]

- Four families of shuffling:
  - Link Shuffling
    - => Conserve time of events by nodes and edges, switch end points
  - Timeline shuffling
    - => Switch time of events between elements (e.g. activations of nodes a and b)
  - Snapshot shuffling
    - =>Keep the order of snapshots, randomize network inside snapshot
  - Sequence Shuffling
    - =>Keep each snapshot identical, but switch randomly their order

- Shufflings can be combined…

RANDOM MODELS



TO SUM UP

CONCLUSION

- Currently, most people still use the snapshot approaches
- Most researchers agree that it has many drawbacks
- But currently:
  - No single framework
  - No library
  - Dataset not has ubiquitous as static graphs
    - (often privates…)

PRACTICALS

PRACTICALS

- 1)Select a temporal networks composed of snapshots
  - 2)Plot the evolution of centralities
- 3)Try to smooth out your network by merging snapshots
  - 4)Recompute evolution centralities
- 5)Compare this evolution with a snapshot randomized network (Randomize networks, keep snapshot orders)
  - 6)Is the tendency correlated with tendencies in the random network?
- 7)Compute the edge density of pairs of nodes, for nodes that are not present all the time. What do you observe? Does it makes sense? In which case is it important?

PRACTICALS

- Datasets:
  - On my webpage, Game of thrones, 2 possibilities:
    - Books: One network by book. Stable. Can study differences
    - Series: One SN every 10 scenes. Unstable. More interesting (in particular for time series analysis), more difficult
  - There Series:
  - https://figshare.com/articles/TV_Series_Networks_of_characters/2199646/11