



# PROPAGATION DYNAMICS IN SOCIAL NETWORKS THROUGH RULE-BASED MODELING

Jason Vallet, Bruno Pinaud, Guy Melançon

Univ. Bordeaux, LaBRI, UMR 5800, Talence

# FOREWORD



We are facing the omnipresence of social networks

- ⦿ Massive amounts of data complexify the analysis of relations among the users (preferences, interests, hobbies...) but allow to search for correlations
- ⦿ Observation of general tendencies similar to a flock of birds or a swarm of insects
- ⦿ Attempts to understand the global and specific social network users behaviours (either as a part of a group or as individuals)

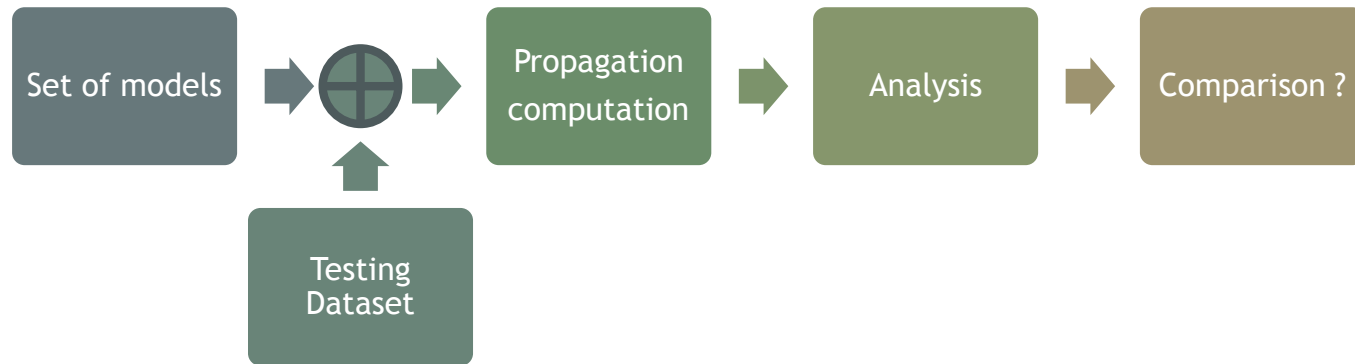
# FOREWORD

The users behaviours have raised the idea of propagation:

- ⊙ Adoption of an action previously performed by one of your « friend »
- ⊙ Notions of influence/motivation: common interests or tastes, social cohesion/pressure within a group
- ⊙ Modelisation and verification: attempts to recreate such behaviour to understand it and anticipate the propagation phenomenon

# FOREWORD

## Normal propagation workflow:



- ⊙ Direct model and propagation application
- ⊙ Limited analysis and comparisons

# PROPAGATION IN LITERATURE

- ◎ Starting with the first mention of influence from one customer to other ones  
Domingos and Richardson (2001)
- ◎ Toward marketing orientation with influence maximization in its simplest form  
Kempe *et al.* (2003)
- ◎ Or pluridisciplinary solutions considering social and biological contagion  
Dodds and Watts (2005)

# PROBLEMS AND OPPORTUNITIES

- ◎ Model comparison:  
necessity to compare the many existing models with one another
- ◎ Model generalization/unification:  
improvements to the existing models by detecting the flaws, different options give the possibility to merge different solutions
- ◎ Influence maximization/minimization:  
more than useful for viral marketing and information spreading analysis

# OUR SOLUTION

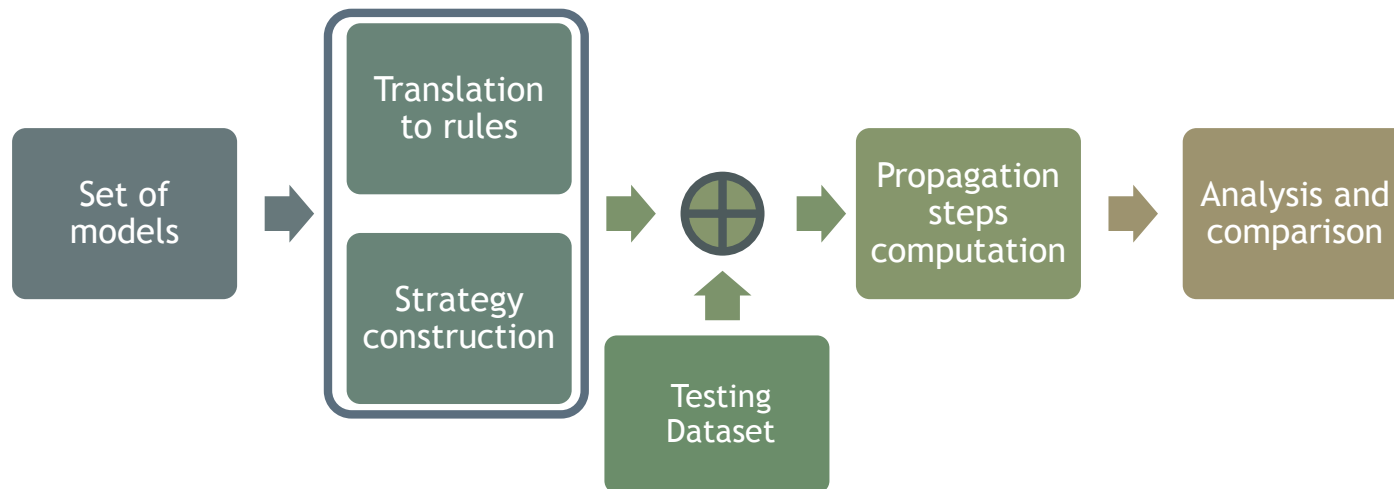


We propose to use graph rewriting techniques to:

- ⦿ Transform existing models into a set of rules: the actual transformation depends of the model complexity, the created rules are non trivial
- ⦿ Take advantage of the strategy language to manage the rules application to respect the initial model description
- ⦿ Expend the existing possibilities in model analysis and comparison

# OUR SOLUTION

## Proposed propagation workflow:

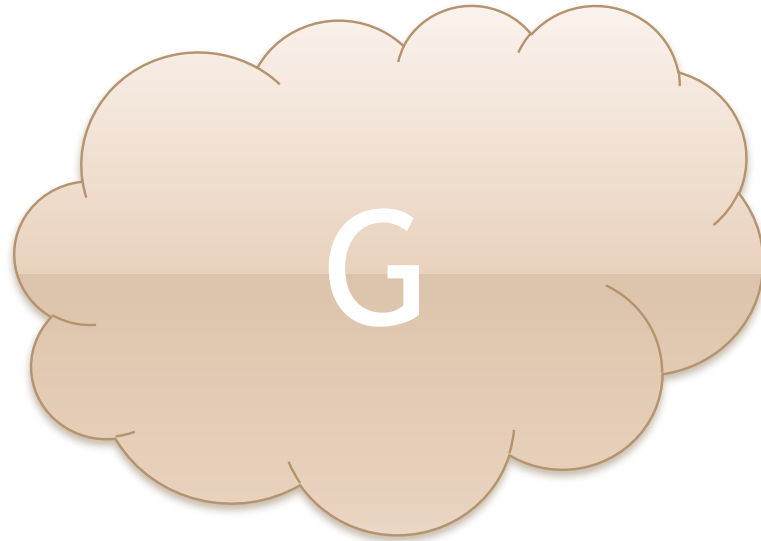


- ⦿ Models transformation
- ⦿ Computation by steps to ease analysis and comparison



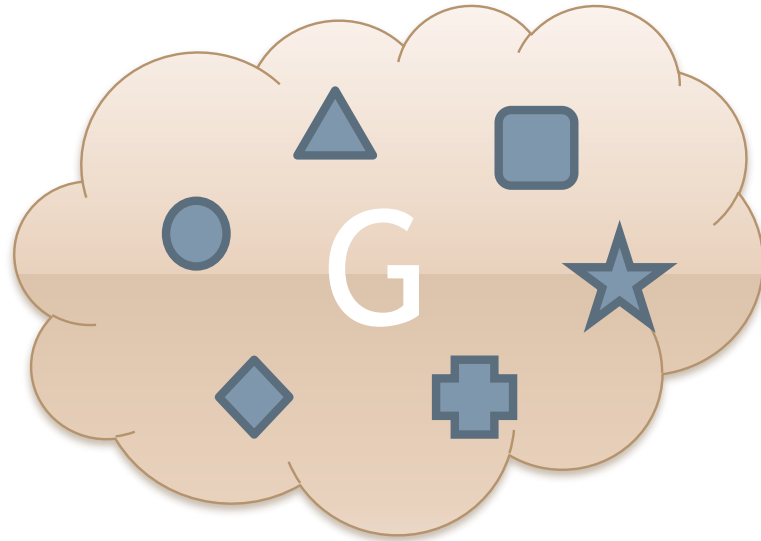
# GRAPH REWRITING

$$G = (V, E)$$



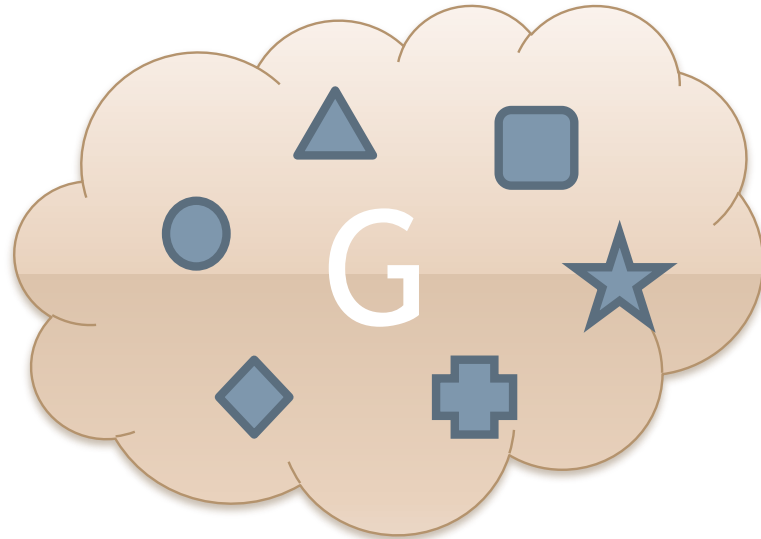
# GRAPH REWRITING

$$G = (V, E)$$

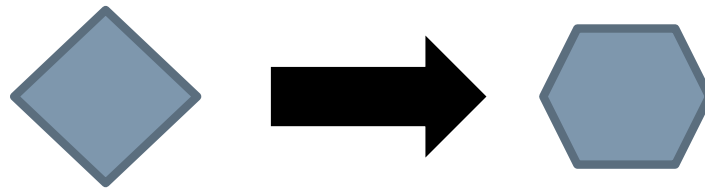


# GRAPH REWRITING

$$G = (V, E)$$

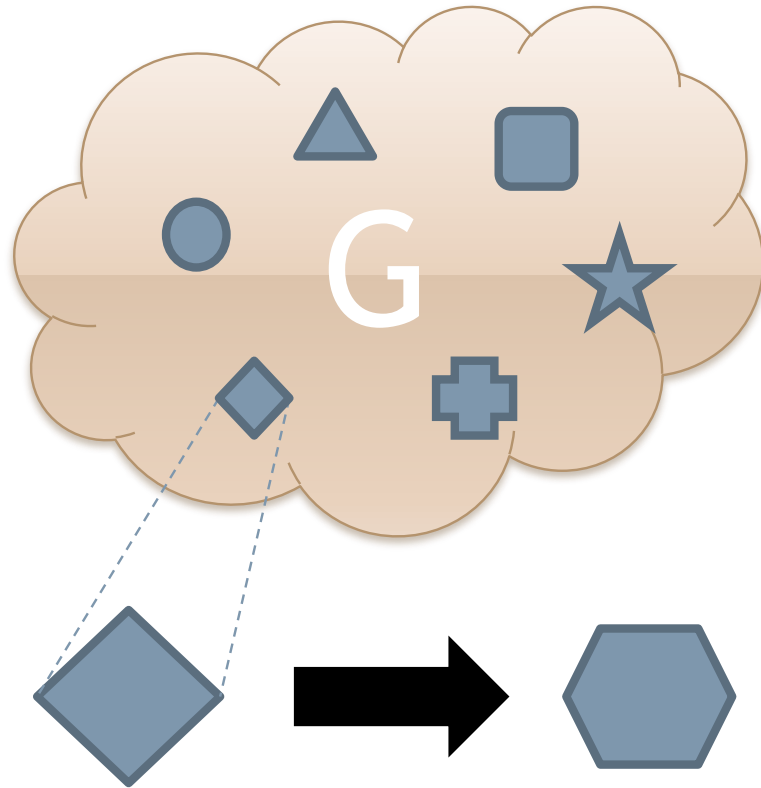


$$r: L \rightarrow R$$



# GRAPH REWRITING

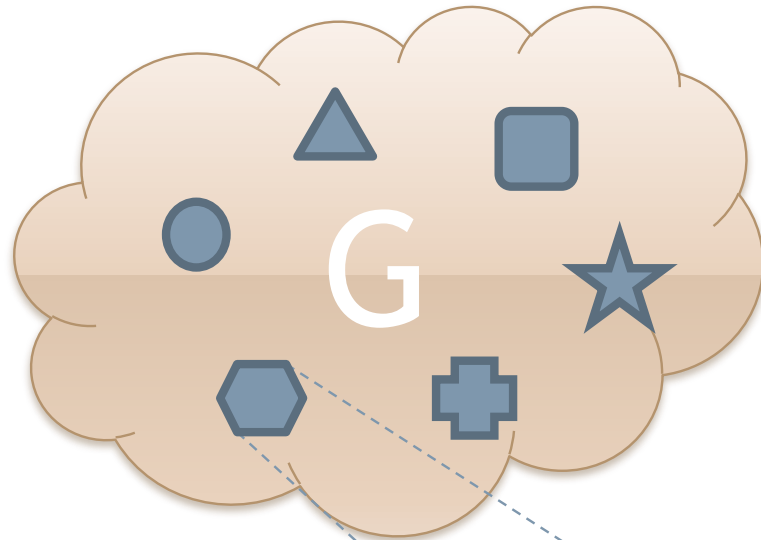
$$G = (V, E)$$



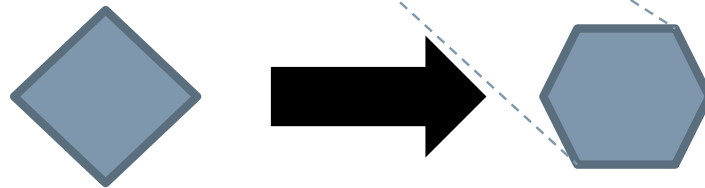
$$r: L \rightarrow R$$

# GRAPH REWRITING

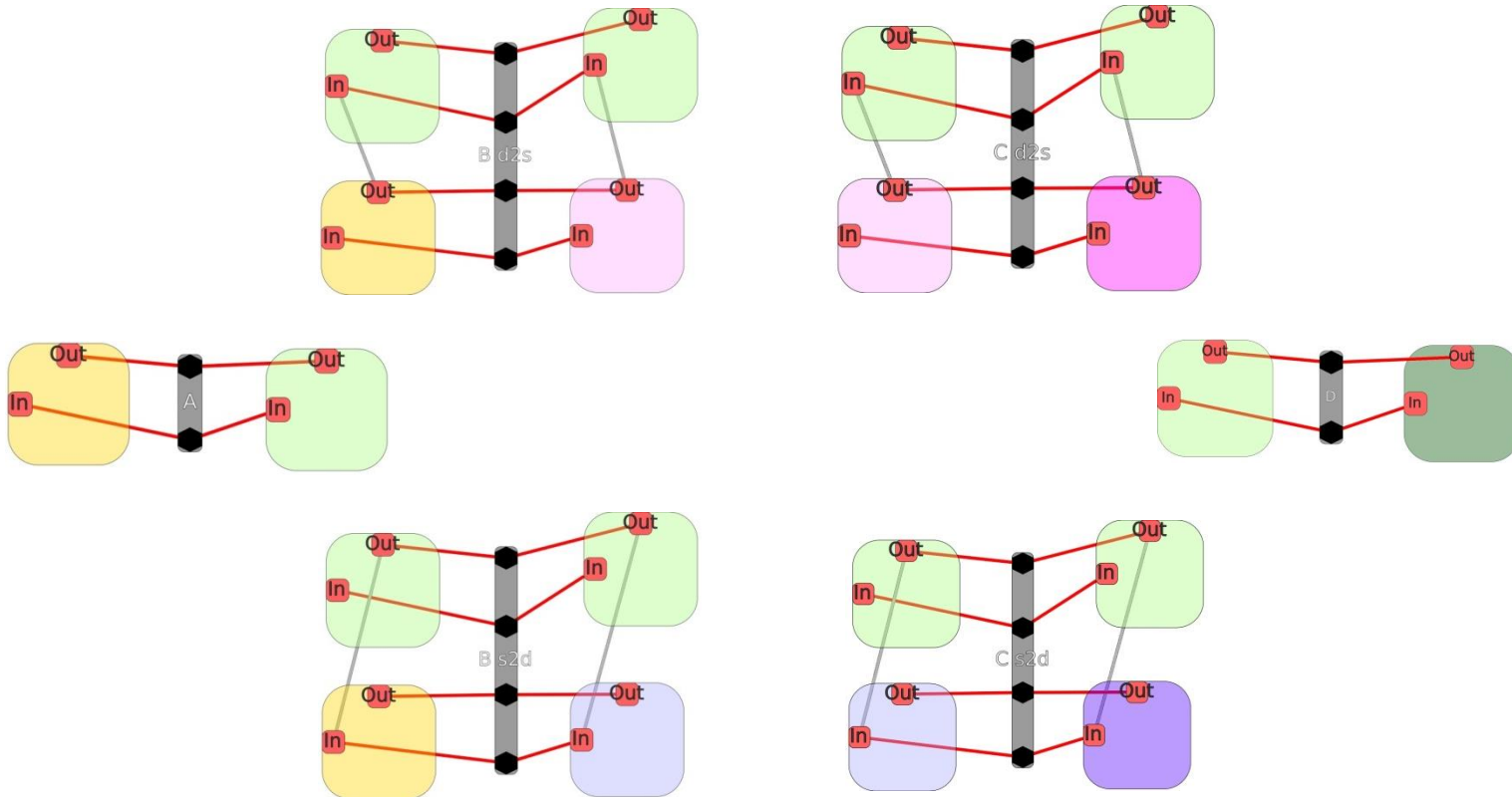
$$G = (V, E)$$



$$r: L \rightarrow R$$



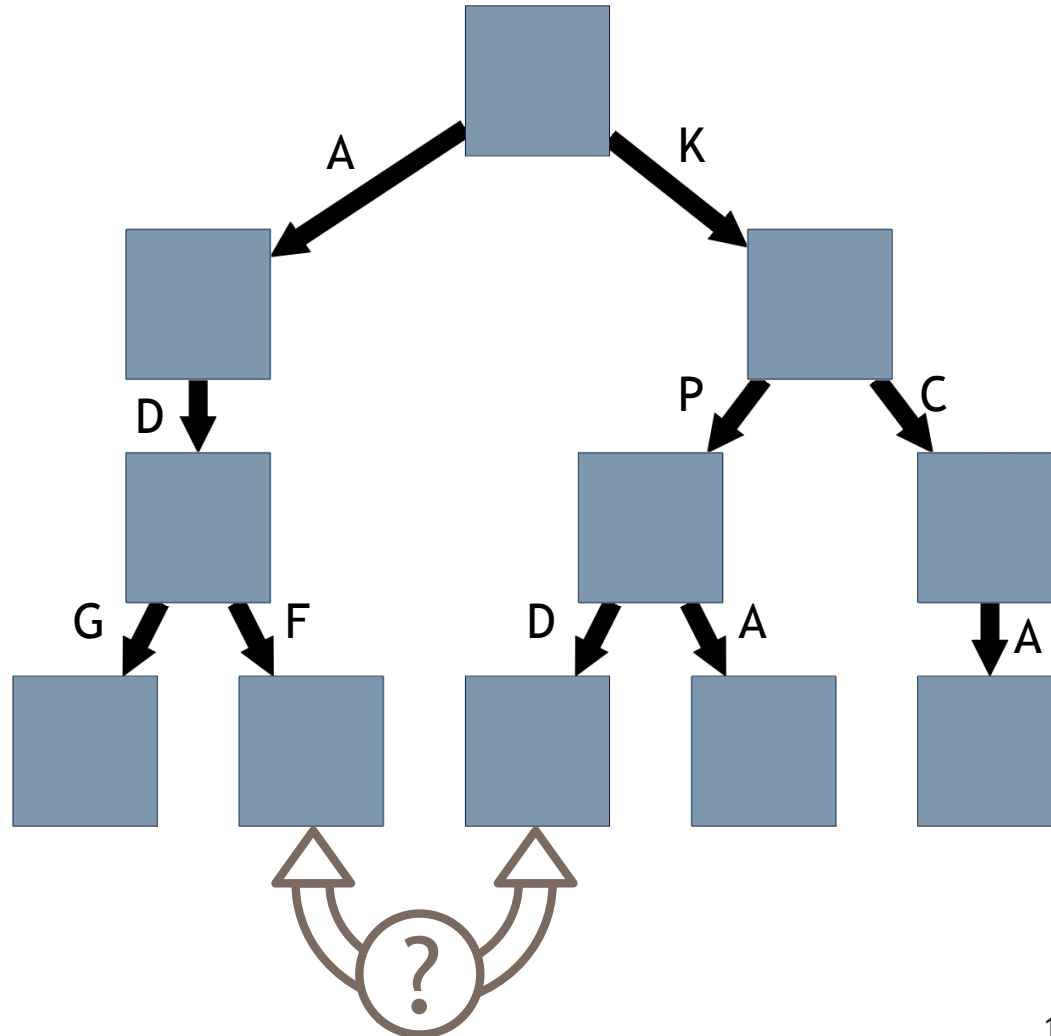
# RULE EXAMPLES



Model from Goyal *et al.* (2010)

Visualization samples from PORGY, Pinaud *et al.* (2012)

# DERIVATION TREE



# BENEFITS

The solution gives some advantages:

- ⦿ Dividing models in atomic bricks allows easier comparison
- ⦿ Exchangeable bricks through models to experiment with variable behaviour gives users the opportunity to refine the models
- ⦿ Keeping track of each atomic operation is essential to understand the missteps met



# IMPAIRS

- ◎ Translation of the initial model:  
considering the many different models, chances are, specificities will be encountered
- ◎ Computation power:  
historic conservation, rule application or influence maximization are expensive to perform
- ◎ Absence of existing comparison methodology:  
from the best of our knowledge, a tool designed to compare propagation models has never been created before

# FUTURE WORK

- ◎ Formal methodology for effective model translation and comparison: generalization for the widely used model families (LT and IC)
- ◎ Model verification: Similar to software model checking, may be used to ensure the results at each step
- ◎ Maximization/minimization discovery: automatic comparison responding to specific given criteria

# REFERENCES

- ⊙ Dodds, P. et D. Watts (2005). A generalized model of social and biological contagion. *Journal of Theoretical Biology* 232(4), 587 - 604.
- ⊙ Domingos, P. et M. Richardson (2001). Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01*, New York, NY, USA, pp. 57-66. ACM.
- ⊙ Goyal, A., F. Bonchi, et L. V. Lakshmanan (2010). Learning influence probabilities in social networks. In *Web Search and Data Mining, Proceedings of the Third ACM International Conference on, WSDM '10*, New York, NY, USA, pp. 241-250. ACM.
- ⊙ Kempe, D., J. Kleinberg, et E. Tardos (2003). Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03*, New York, NY, USA, pp. 137-146. ACM.
- ⊙ Pinaud, B., G. Melançon, et J. Dubois (2012). Porgy : A visual graph rewriting environment for complex systems. *Computer Graphics Forum* 31(3), 1265-1274

THANK YOU FOR YOUR  
ATTENTION.

Contact:

[jason.vallet@labri.fr](mailto:jason.vallet@labri.fr)

[bruno.pinaud@labri.fr](mailto:bruno.pinaud@labri.fr)

[guy.melancon@labri.fr](mailto:guy.melancon@labri.fr)

# APPENDICES

# LINEAR THRESHOLD MODEL

- ⊙ A node  $v$  has a set of neighbours  $S$
- ⊙ A neighbour  $w$  can influence  $v$  according to a weight  $p_{w,v}$  such as

$$\sum_{w \in S} p_{w,v} \leq 1$$

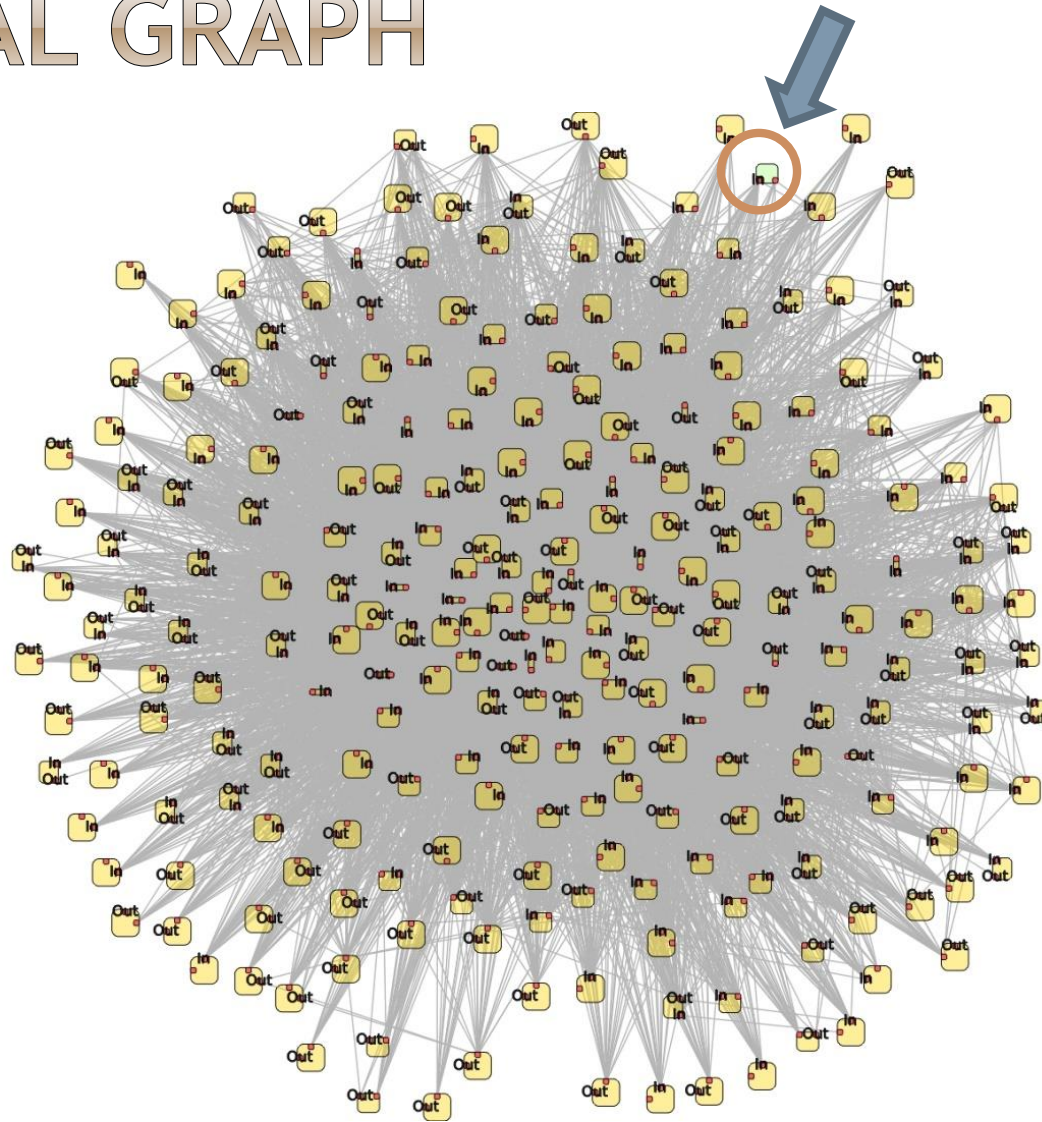
- ⊙ Each  $v$  possesses a *threshold* value  $\theta_v$  usually picked randomly in  $[0, 1]$
- ⊙  $v$  activates whenever the combined weight of its neighbours exceeds the threshold value

$$\sum_{\text{active } w \in S} p_{w,v} \geq \theta_v$$

# INDEPENDANT CASCADE MODEL

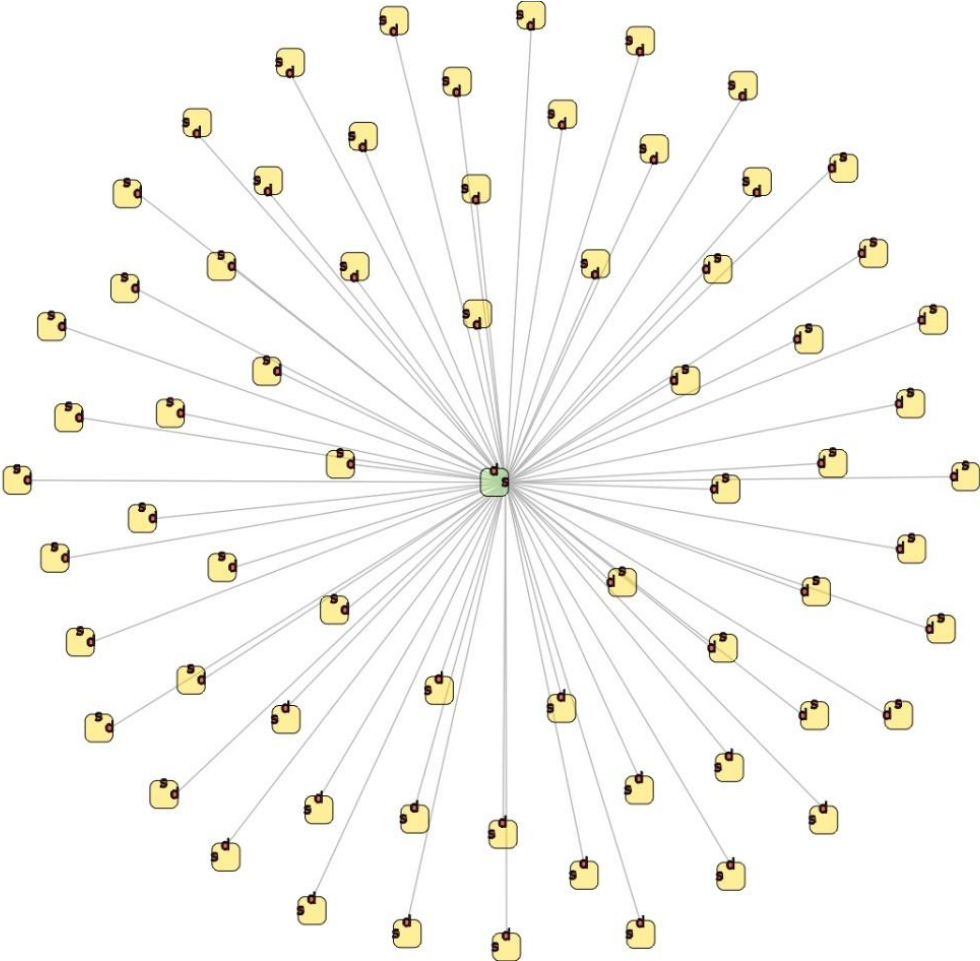
- ⊙ A node  $v$  has a set of neighbours  $S$
- ⊙ A neighbour  $w$  gets a single chance of activation following the *probability*  $p_{v,w}$
- ⊙ When  $v$  activates, each neighbour is tested:
  - If  $w$  succeeds, it becomes active and the cascade will proceed;
  - Otherwise,  $w$  stays inactive and no further attempts of activation wil be performed.

# INITIAL GRAPH

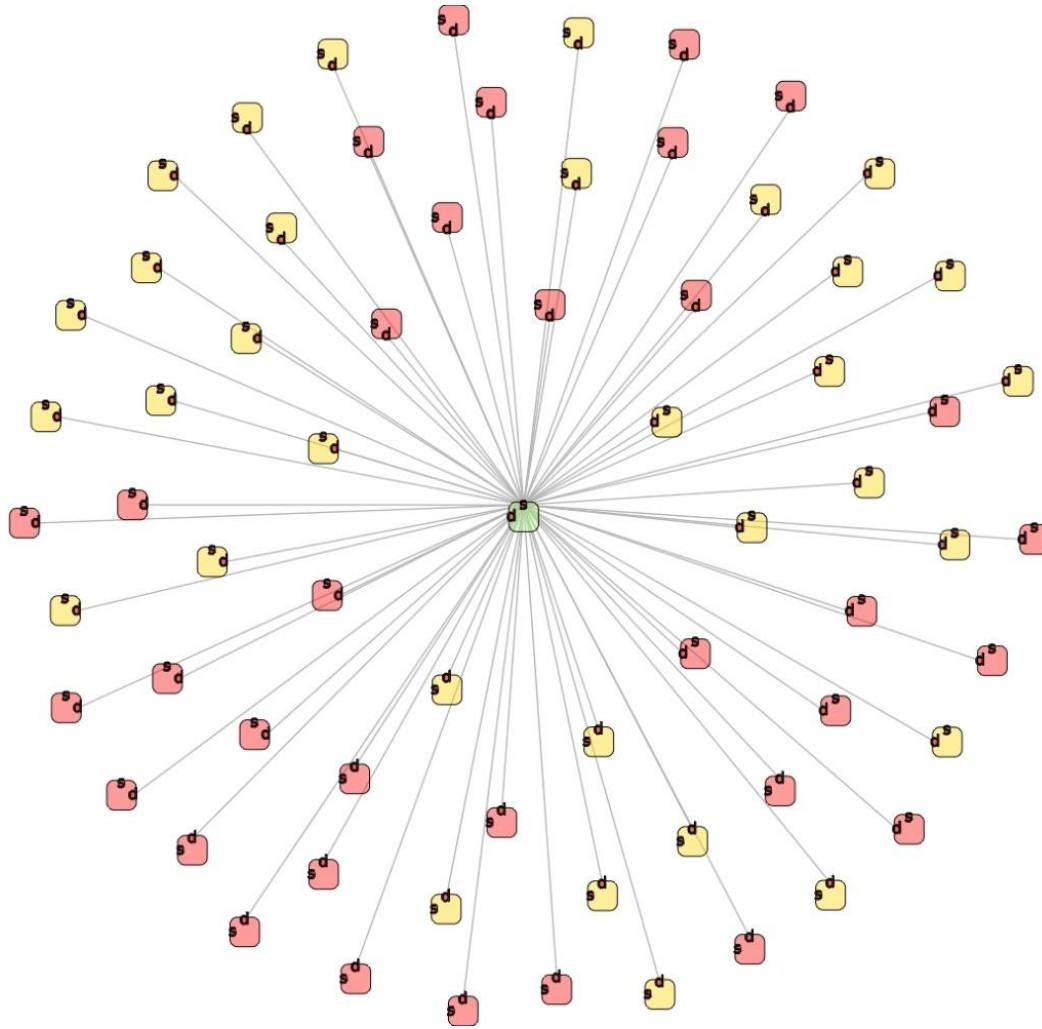




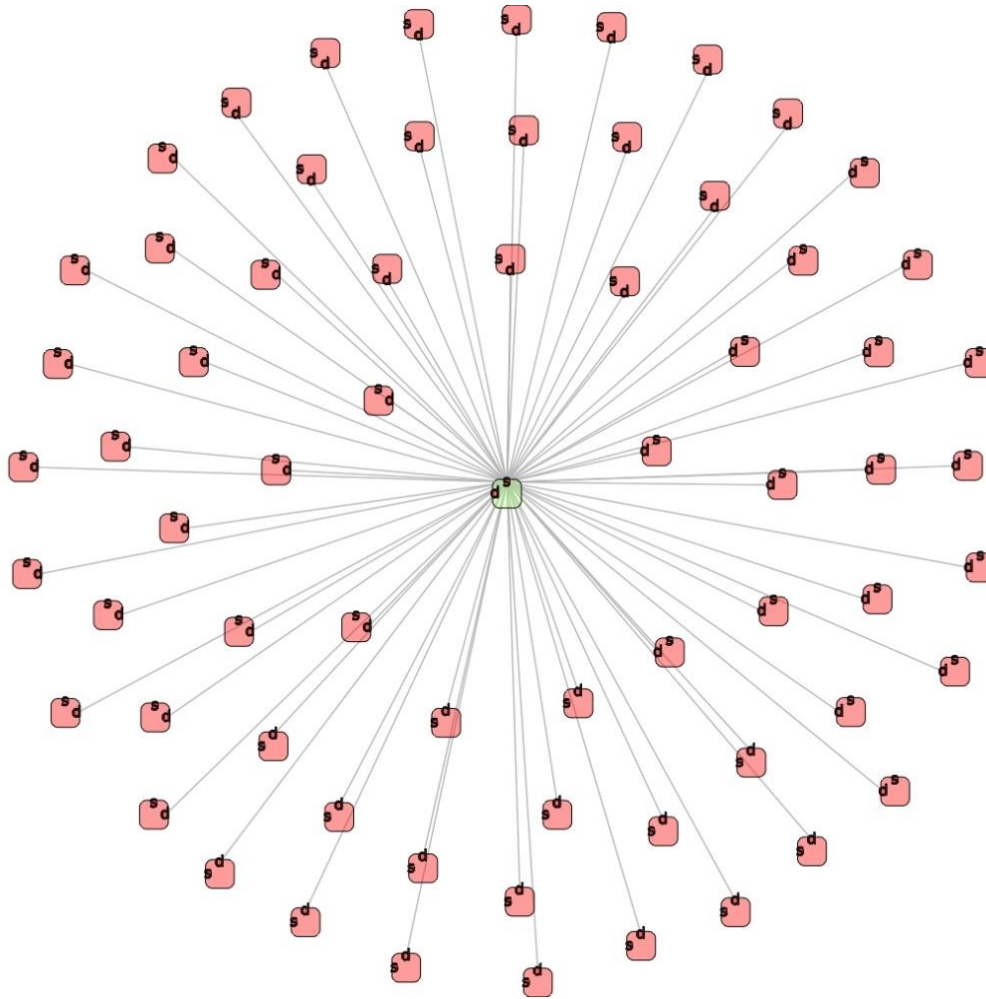
# SUBGRAPH



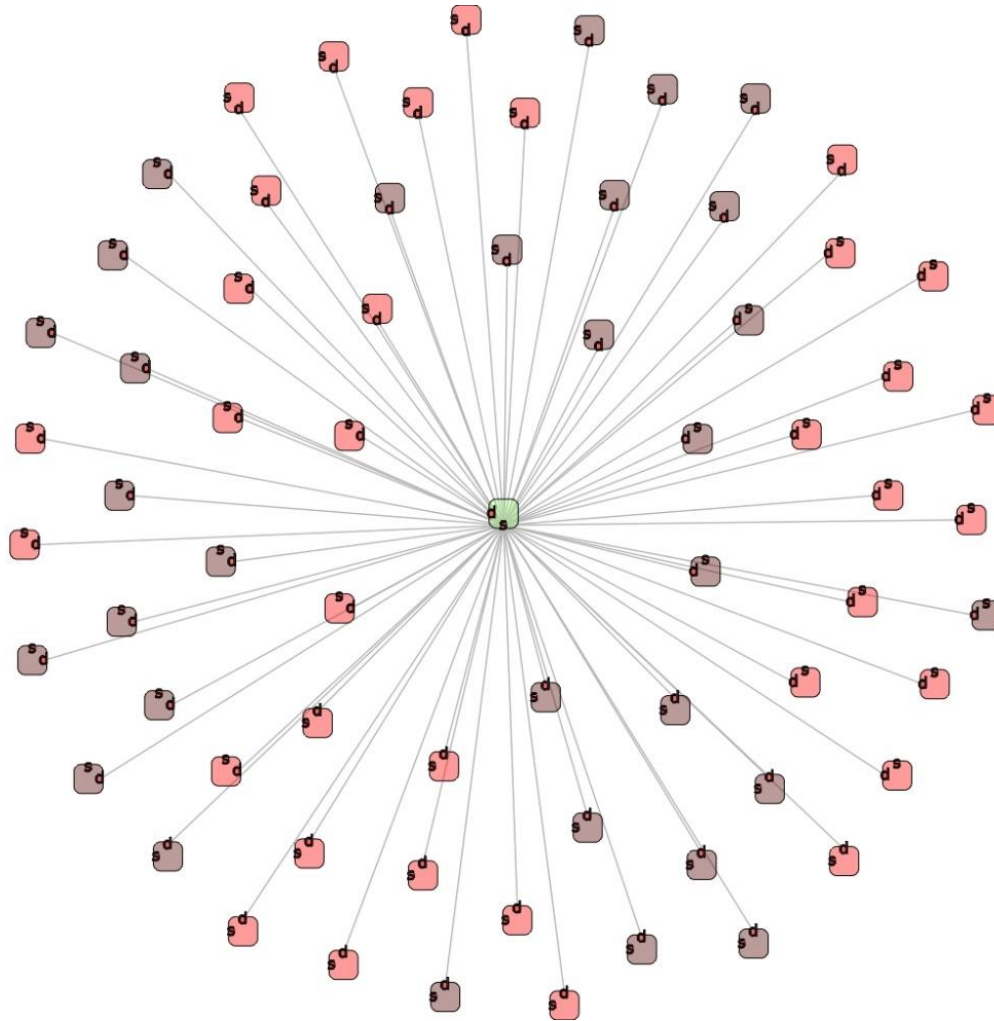
# PROPAGATION - STEP 1



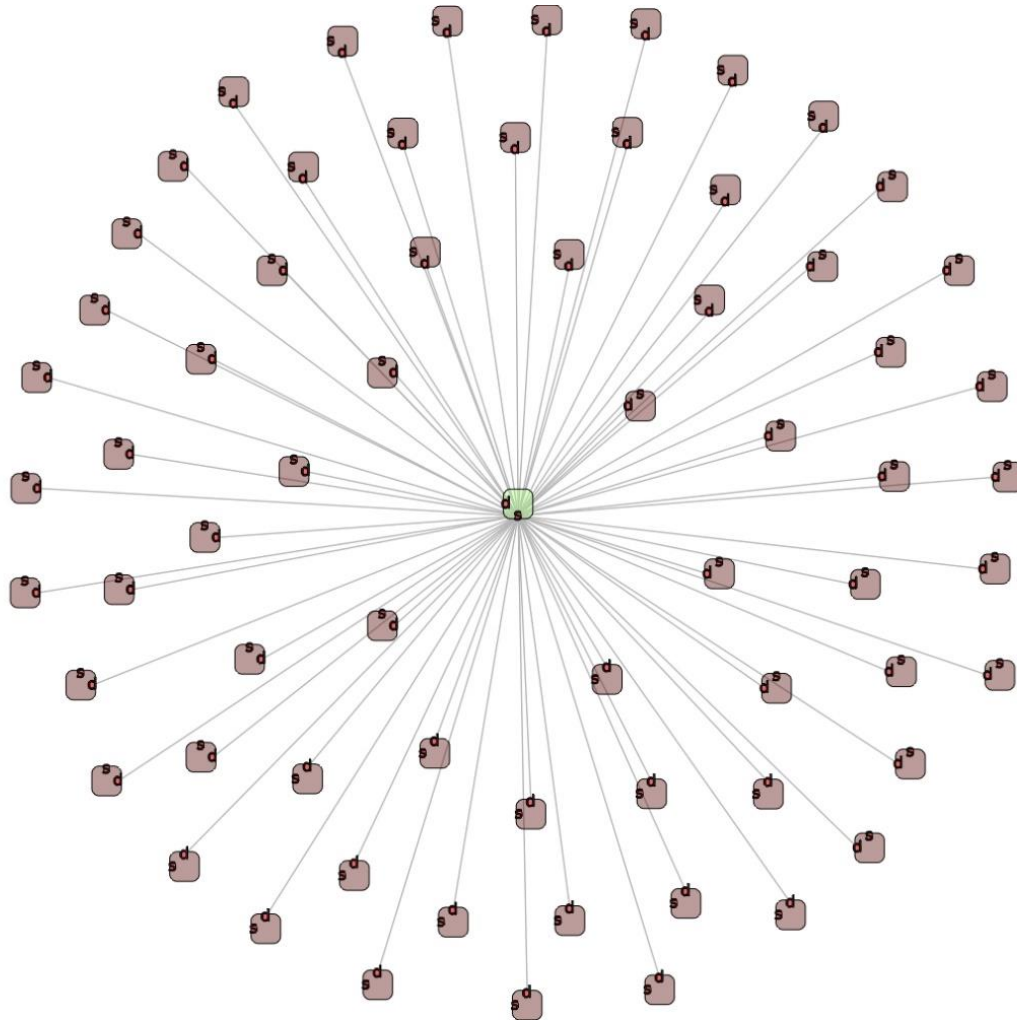
# PROPAGATION - STEP 1



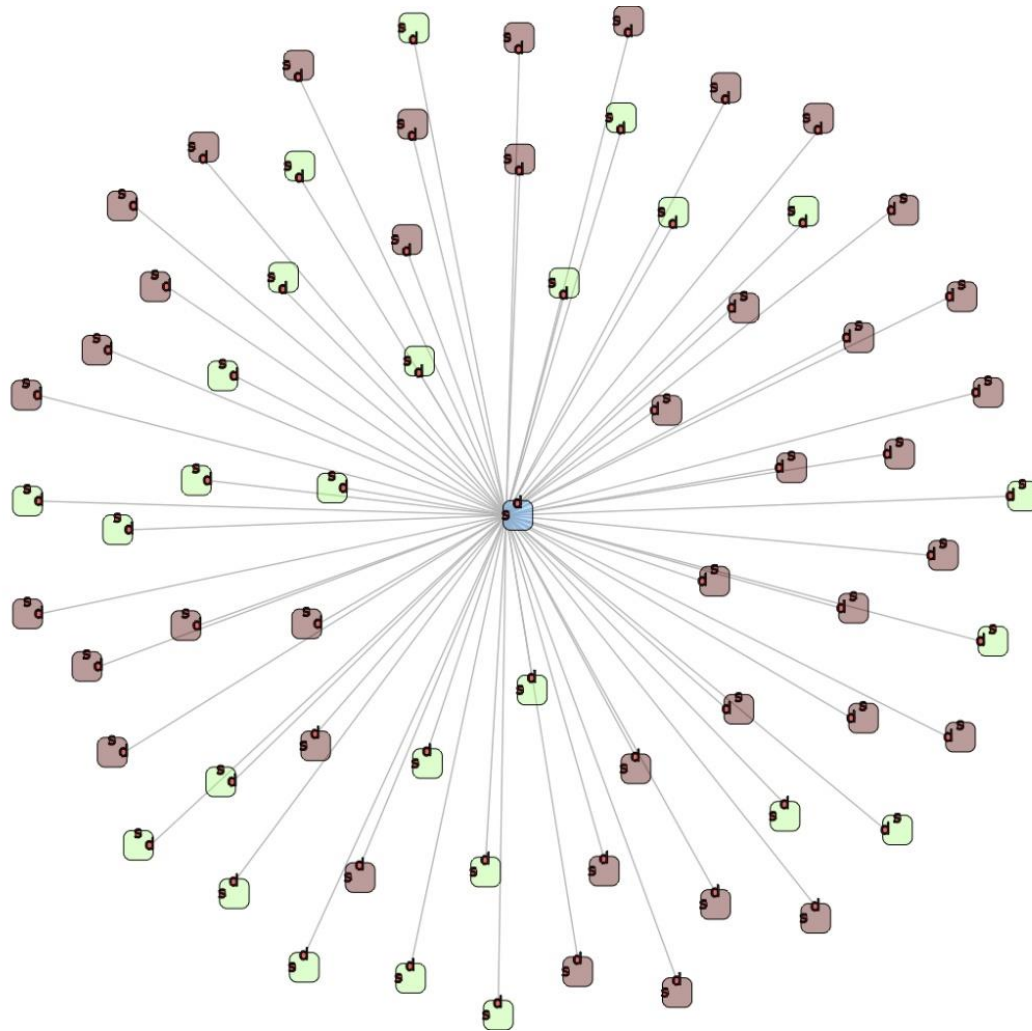
# PROPAGATION - STEP 2



# PROPAGATION - STEP 2

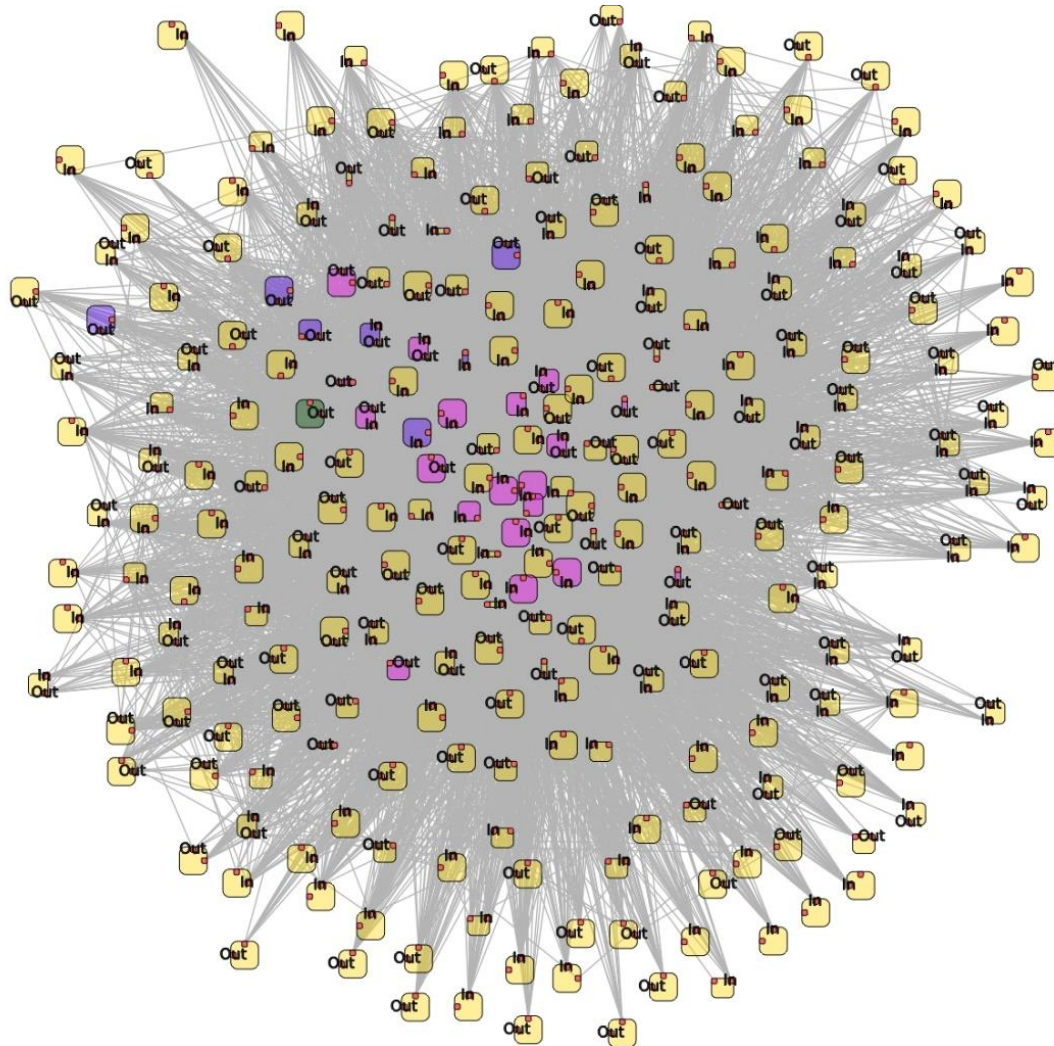


# NEXT CANDIDATES

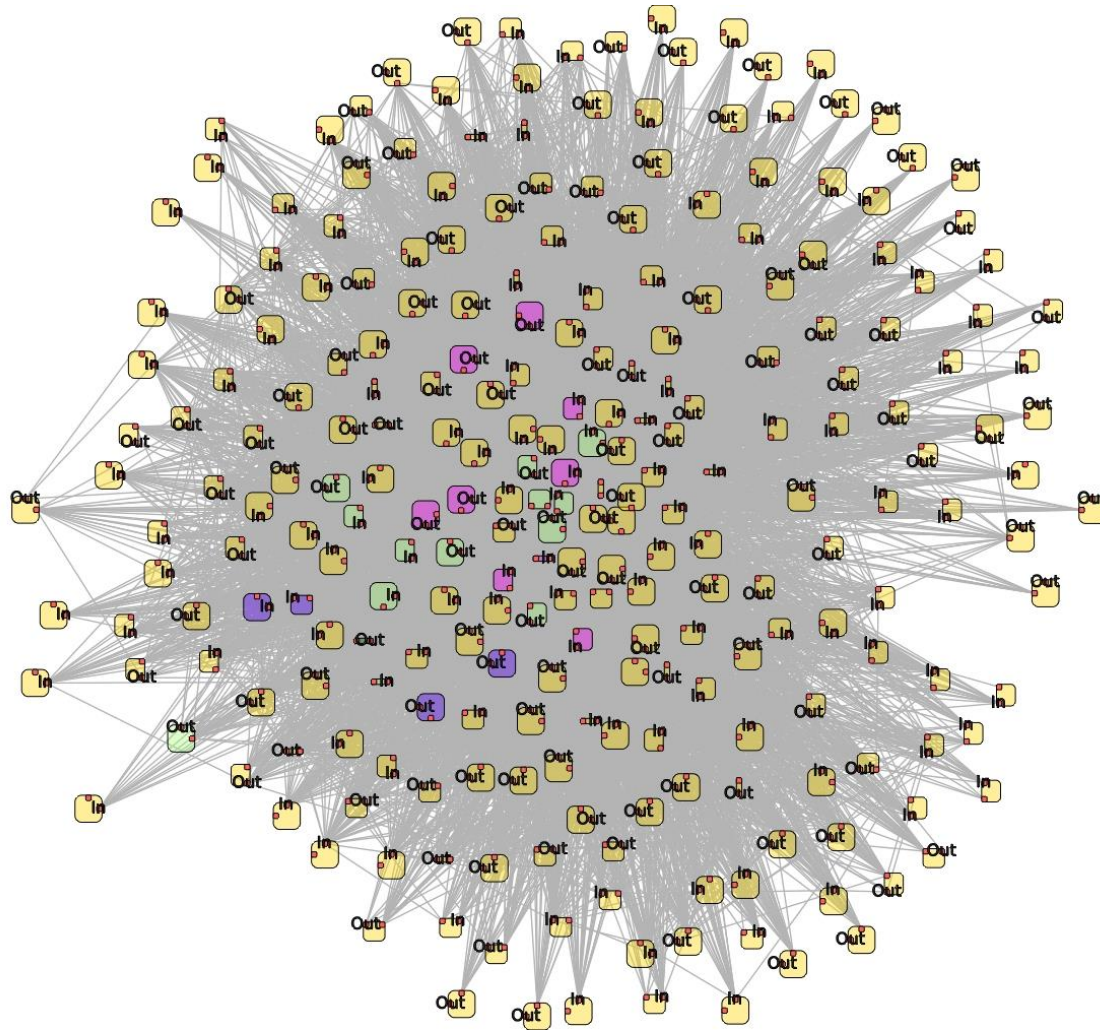




# PROPAGATION

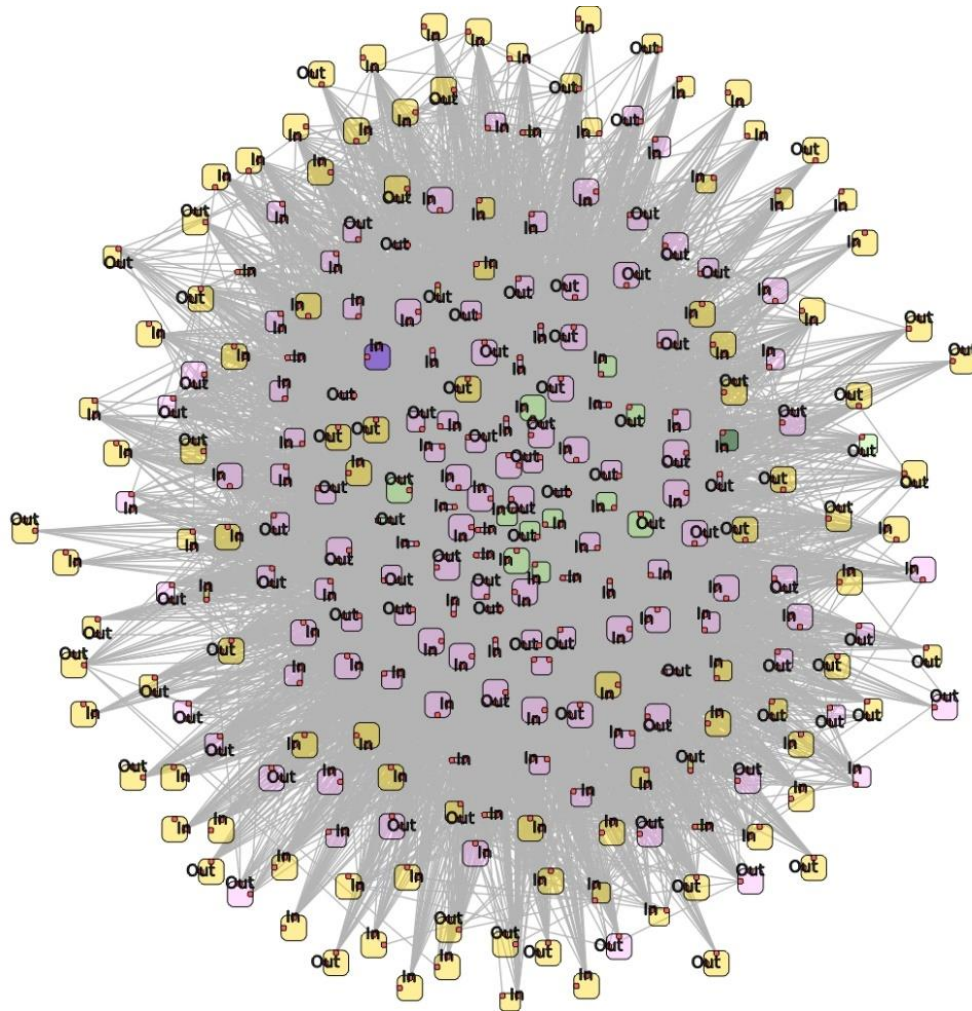


# NEXT CANDIDATES





# PERPETUATION



# PERPETUATION

