Different community finding algorithms on classic generated model and real networks

- Generate a test bed
- Evaluate different algorithms on the test bed
- Observe how the algorithms work on the test bed and on real networks

Classical generated model

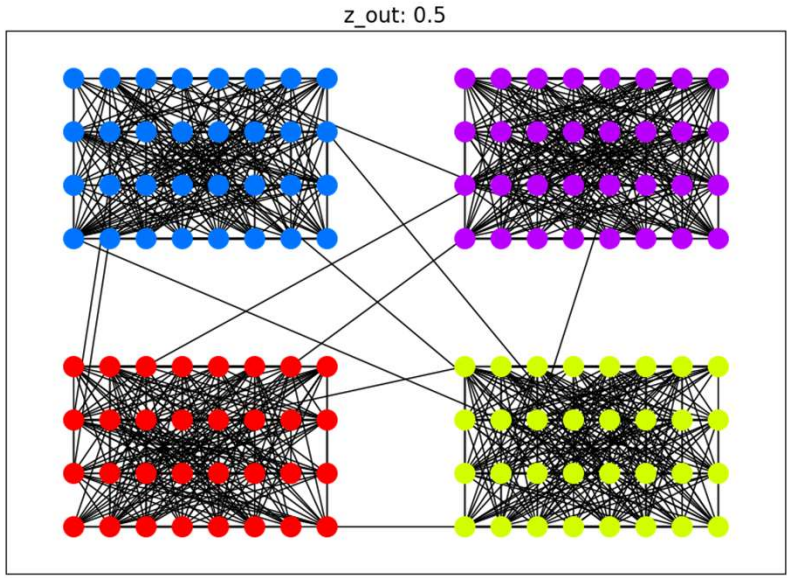
- 4 communities (32 nodes each) with random links
- Average connections within-community: z_in
- Average connections out-community: z_out
- z_in + z_out = ⟨k⟩ = 16

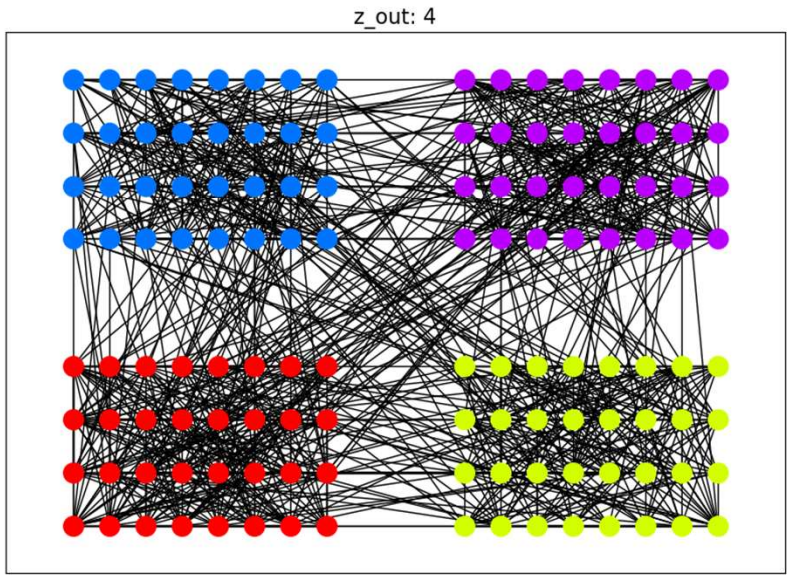
==>

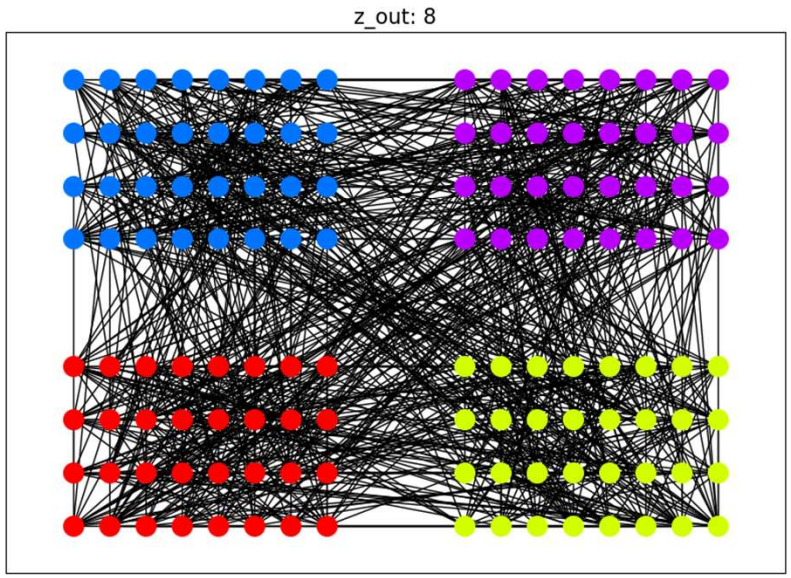
- When z_out is close to 0, it is easy to identify communities
- When z_out is close to 8, the graph becomes homogeneous
- When z_out is close to 16, nodes tend to avoid their own community

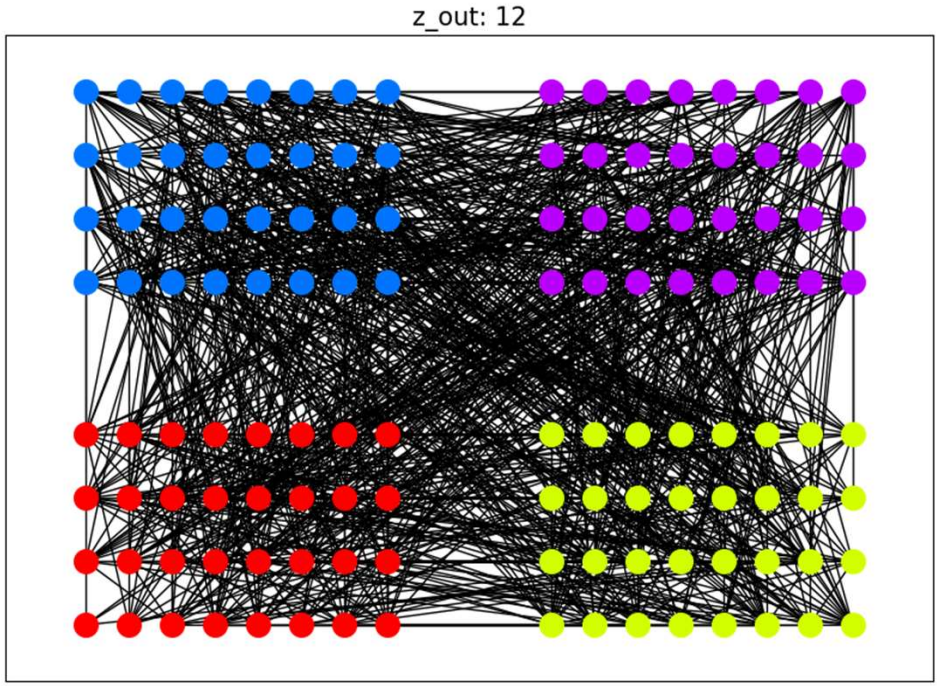
Example of generated model with different z_out

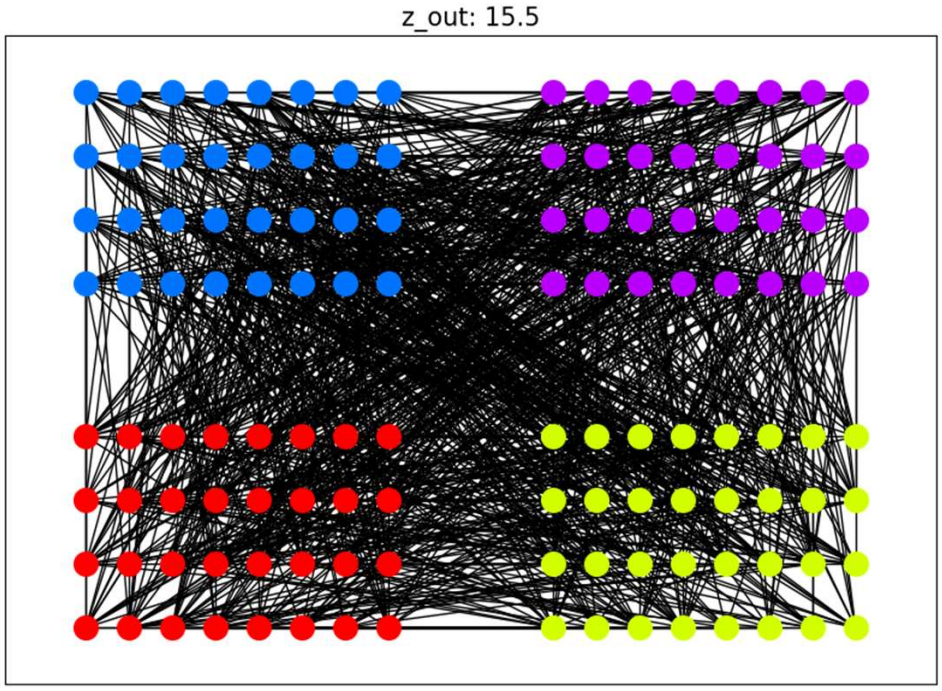
We will generate graph with z_out at:
0.5, 4, 8, 12, 15.5





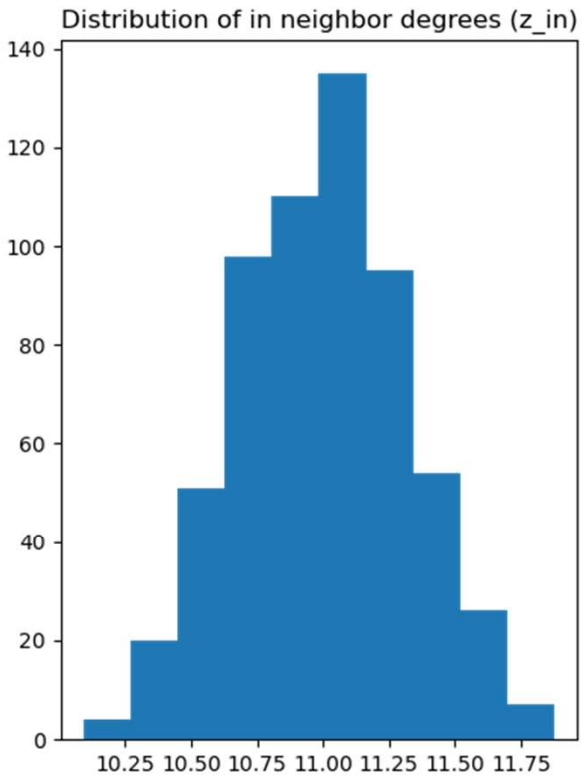


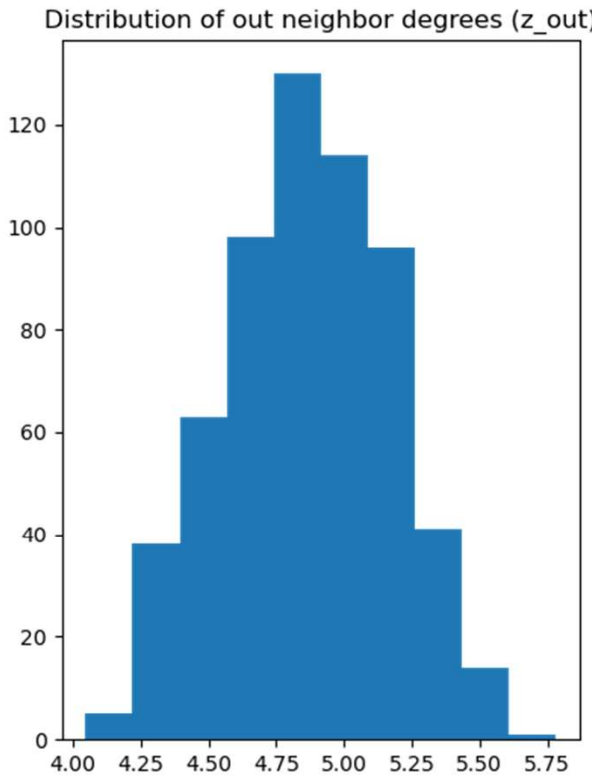

z_out: 12

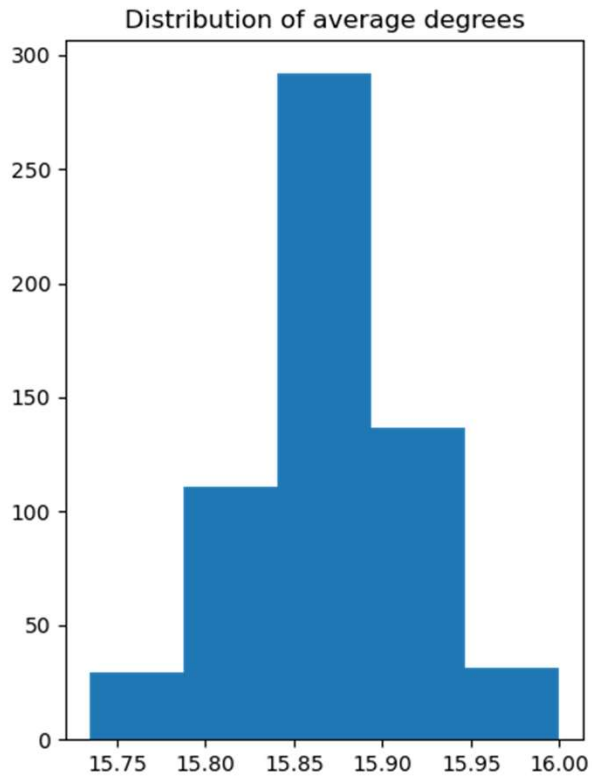

z_out: 15.5

Distribution of degrees

Calculated based on 500 graphs generated at z_out = 5







Different community finding algorithms

- Algorithms:
  - Louvain
  - Surprises community
  - Leiden
  - Walktrap

To evaluate the algorithms, we check the similarity between four ground truth groups and the found groups.

We will observe for different z_out values:

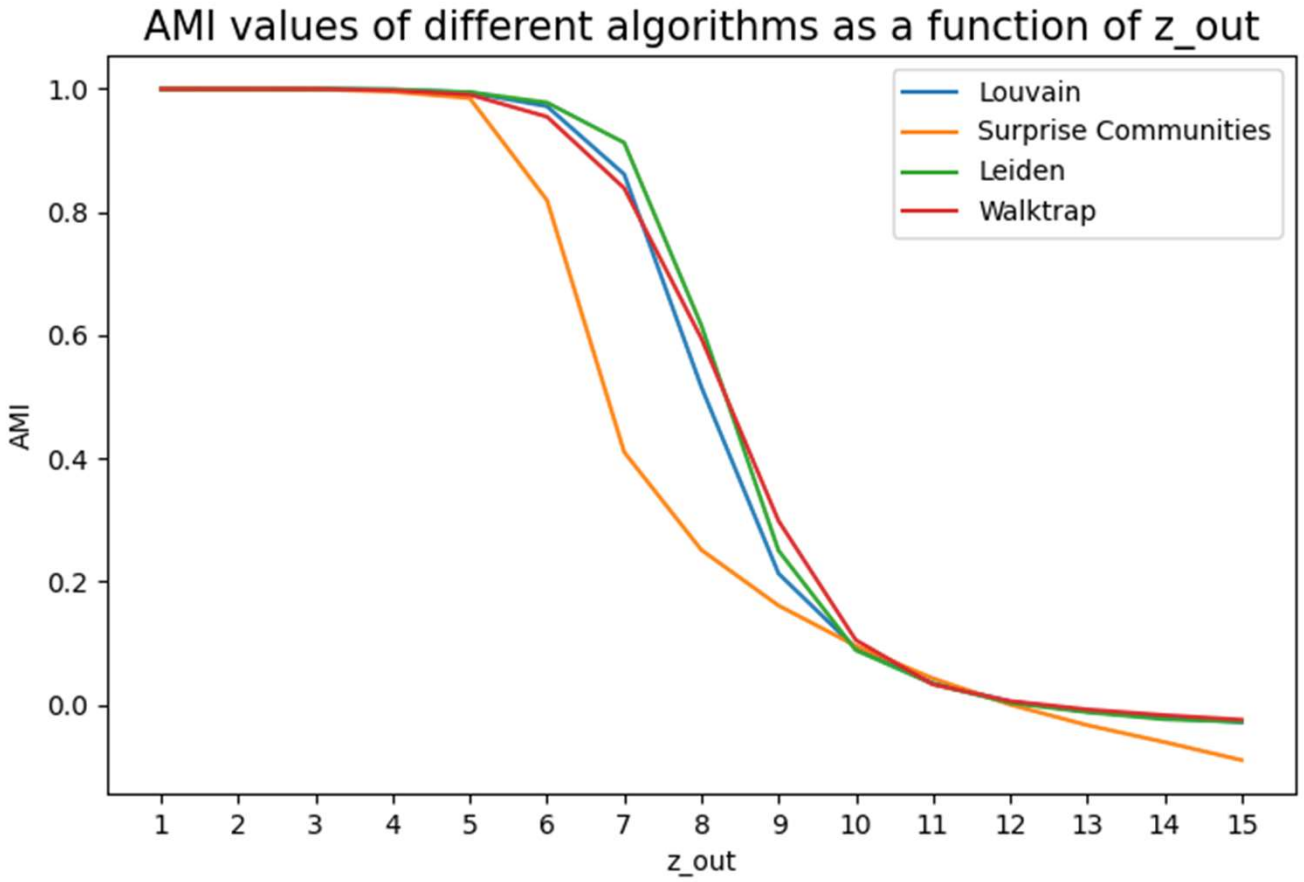
- AMI (Adjusted Mutual Information): a metric to compare similarity of two clusterings for a dataset. The higher AMI, the better accuracy algorithm has.
- Number of communities found by algorithms

The data will be simulated 100 times to calculated for randomness.

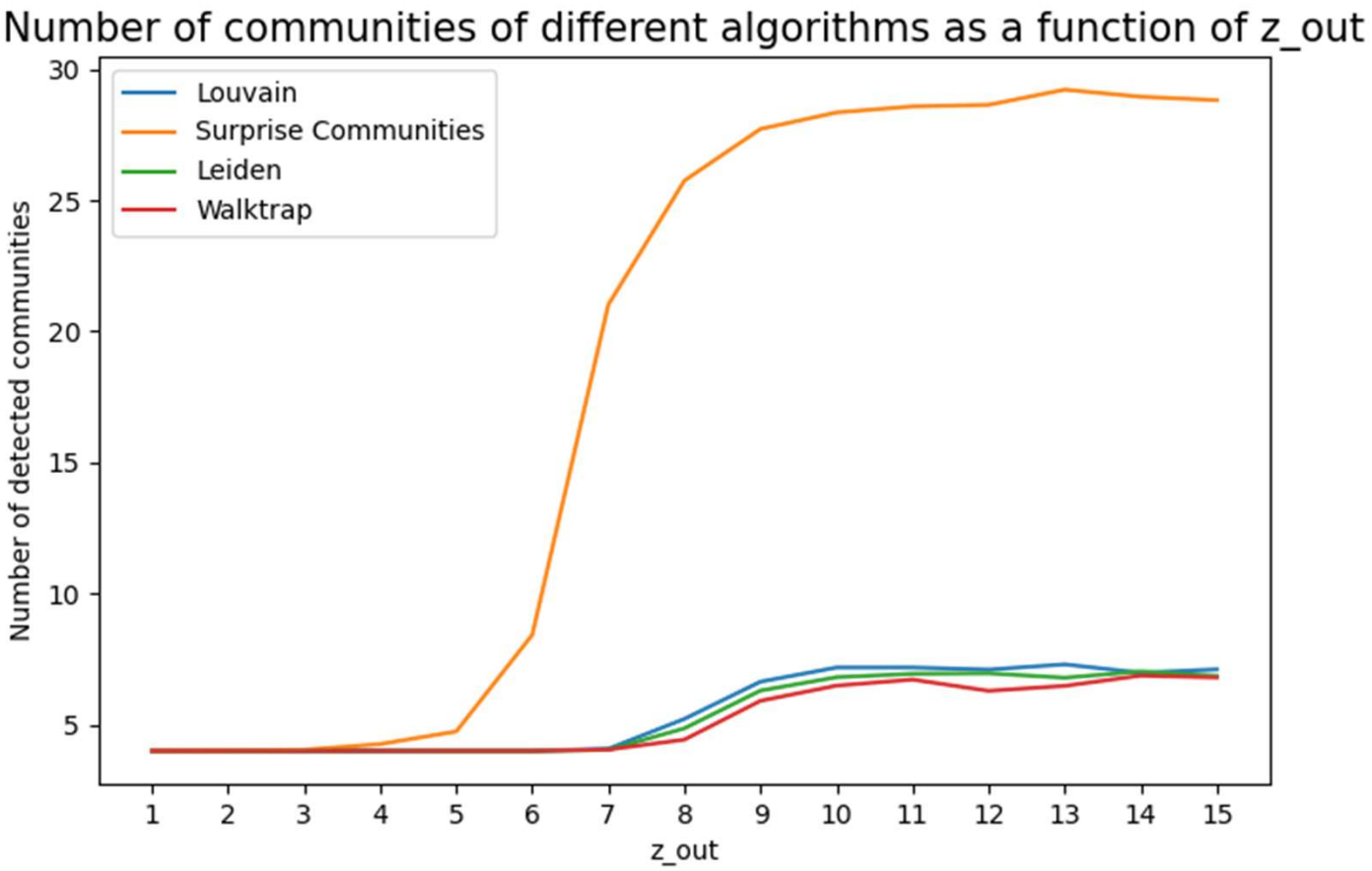
References:

- https://en.wikipedia.org/wiki/Louvain_method
- https://cdlib.readthedocs.io/en/latest/reference/cd_algorithms/algs/cdlib.algorithms.surprise_communities.html
- https://cdlib.readthedocs.io/en/latest/reference/cd_algorithms/algs/cdlib.algorithms.leiden.html
- https://cdlib.readthedocs.io/en/latest/reference/cd_algorithms/algs/cdlib.algorithms.walktrap.html
- https://en.wikipedia.org/wiki/Adjusted_mutual_information

Different community finding algorithms



Different community finding algorithms



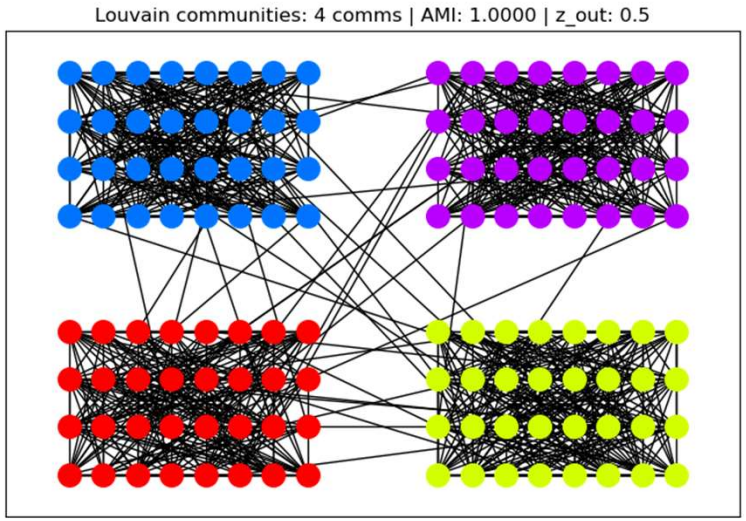
Observe algorithms on generated networks

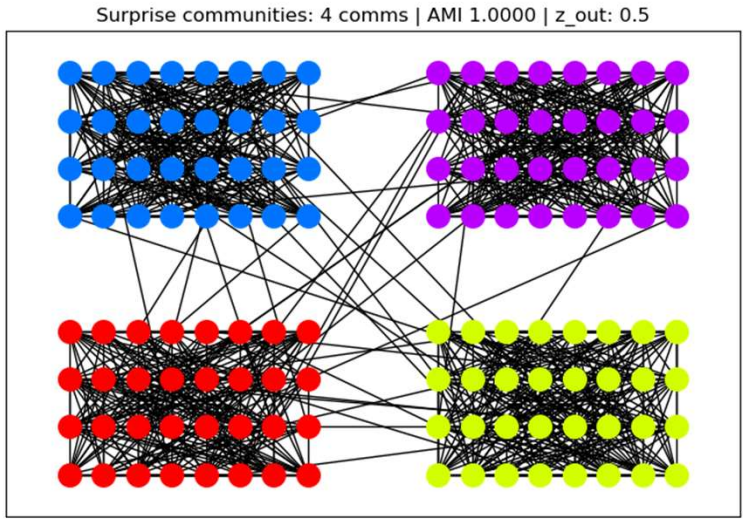
The algorithms performances start to degenerate at z_out = 7, thus we will generate graph with z_out at:

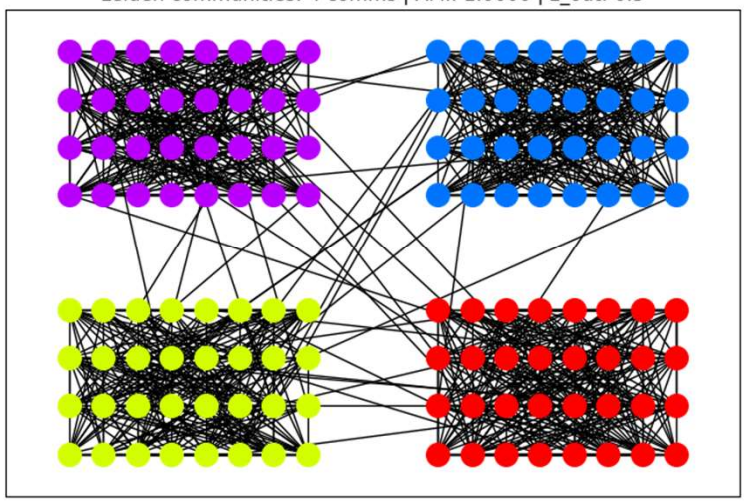
0.5, 4, 7, 8, 12, 15.5

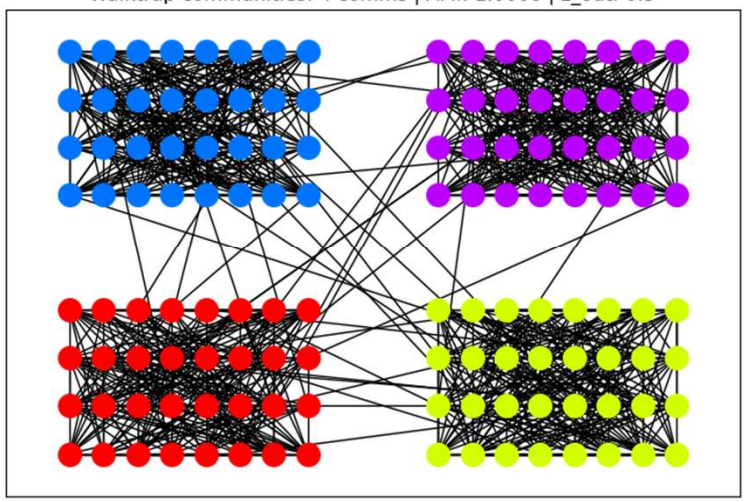
Then, we will observe:

- The number of communities found
- AMI value
- The graph of found communities

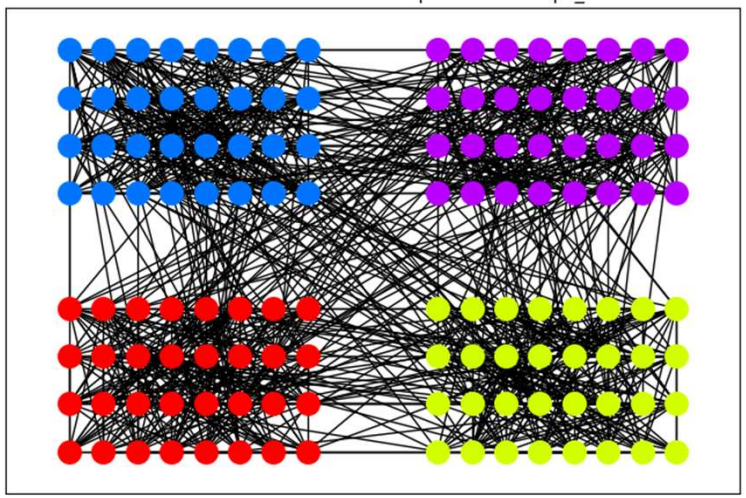

Louvain communities: 4 comms | AMI: 1.0000 | z_out: 0.5


Surprise communities: 4 comms | AMI 1.0000 | z_out: 0.5

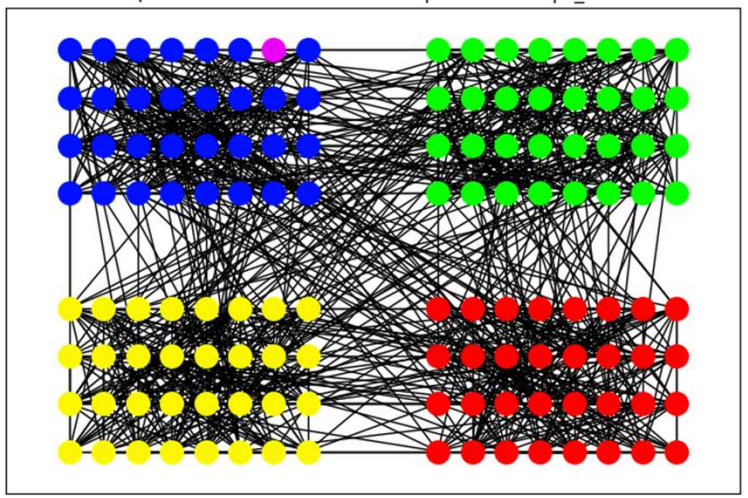

Leiden communities: 4 comms | AMI: 1.0000 | z_out: 0.5


Walktrap communities: 4 comms | AMI: 1.0000 | z_out: 0.5

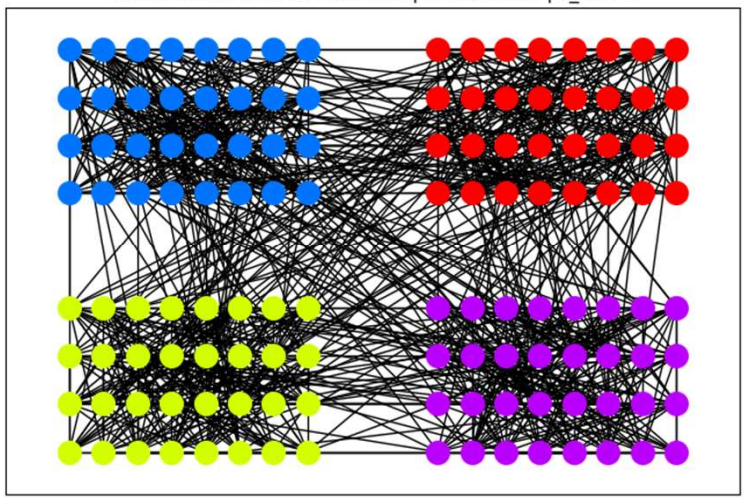
Louvain communities: 4 comms | AMI: 1.0000 | z_out: 4



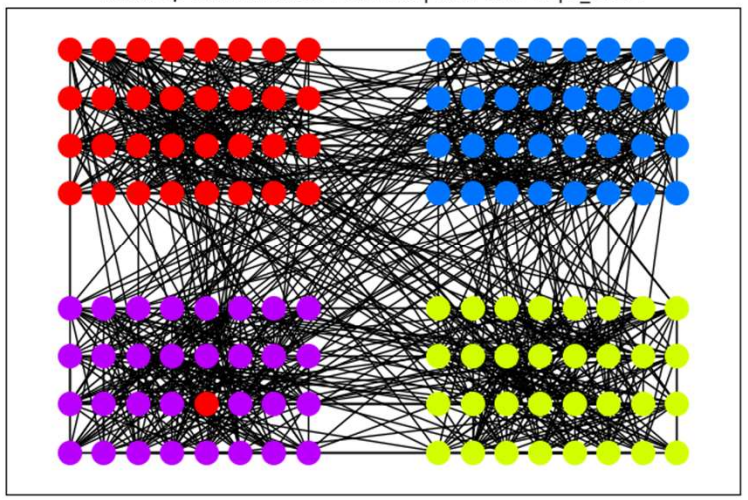
Surprise communities: 5 comms | AMI 0.9872 | z_out: 4



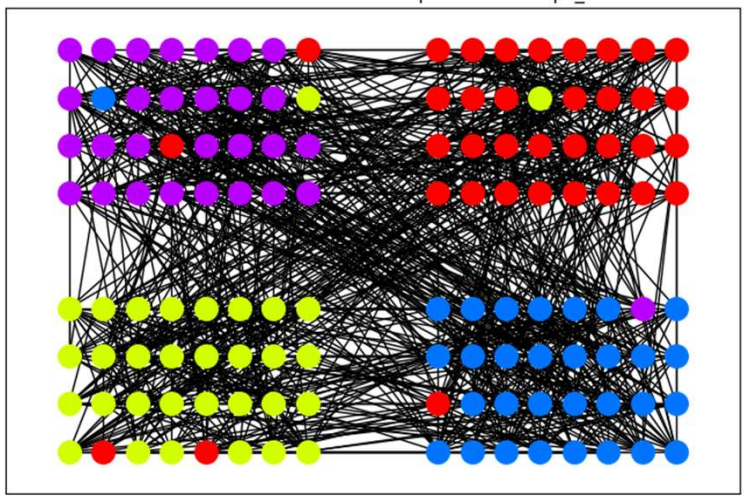
Leiden communities: 4 comms | AMI: 1.0000 | z_out: 4



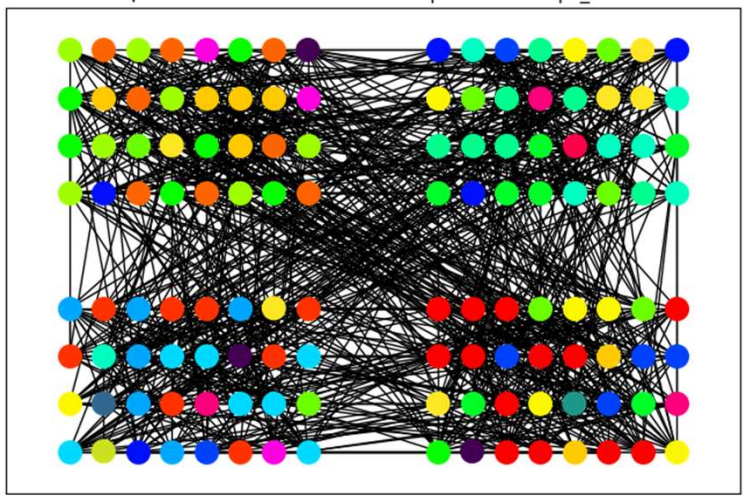
Walktrap communities: 4 comms | AMI: 0.9742 | z_out: 4



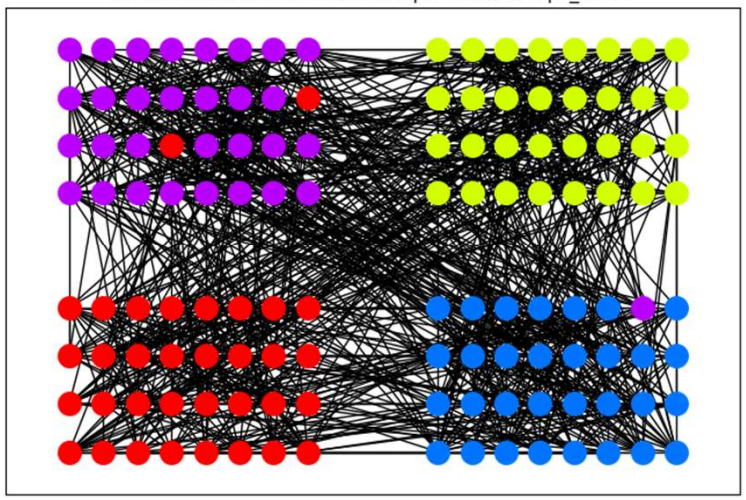
Louvain communities: 4 comms | AMI: 0.7844 | z_out: 7



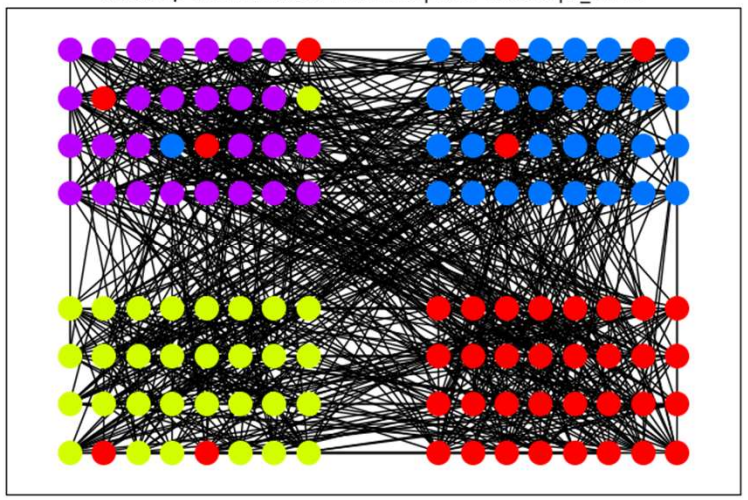
Surprise communities: 21 comms | AMI 0.3543 | z_out: 7



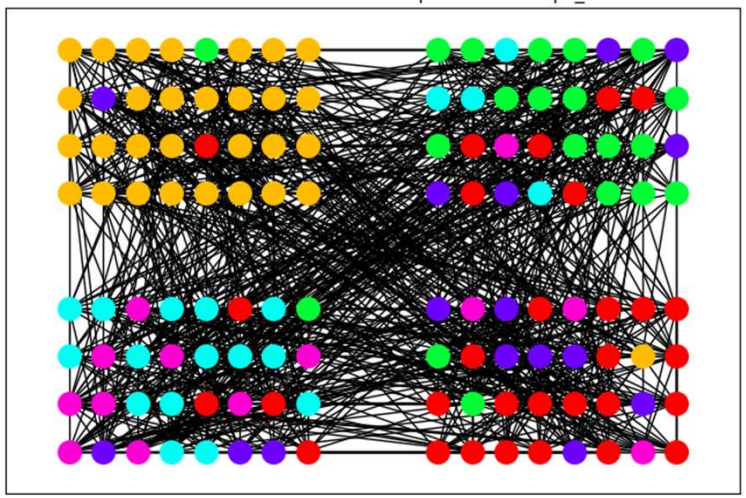
Leiden communities: 4 comms | AMI: 0.9307 | z_out: 7



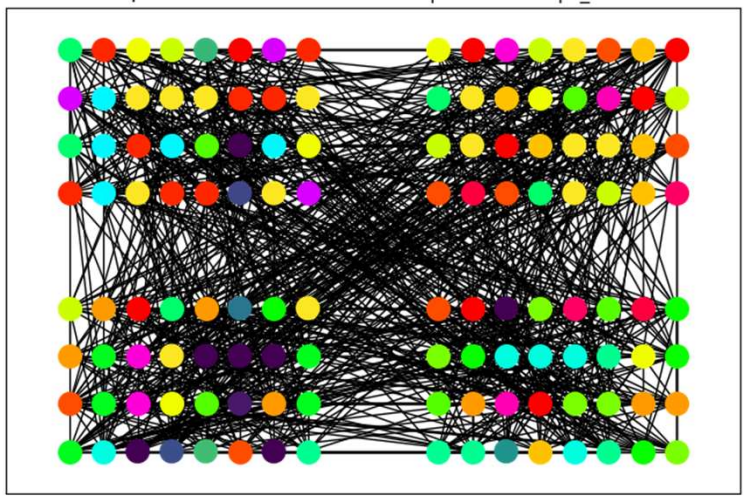
Walktrap communities: 4 comms | AMI: 0.7863 | z_out: 7



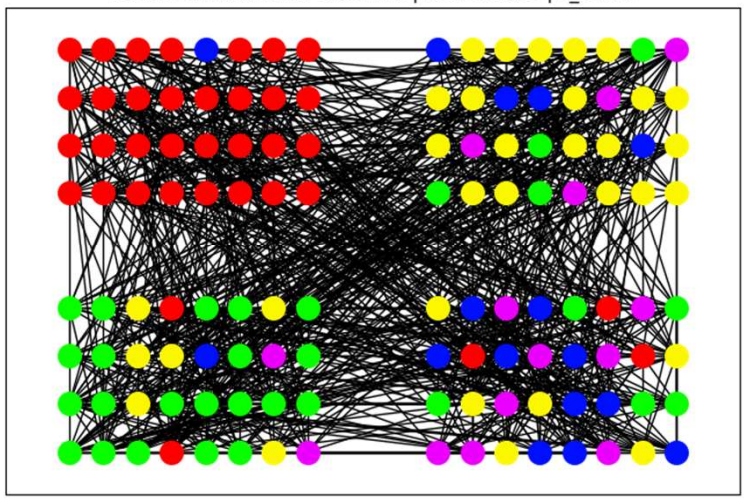
Louvain communities: 6 comms | AMI: 0.4243 | z_out: 8



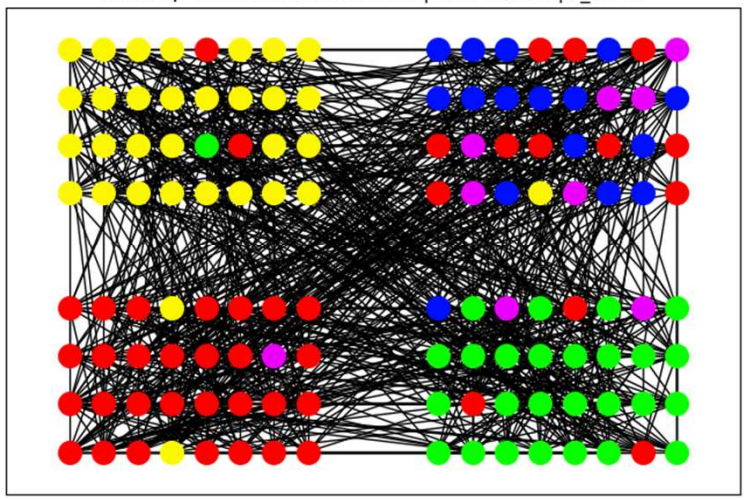
Surprise communities: 27 comms | AMI 0.1668 | z_out: 8



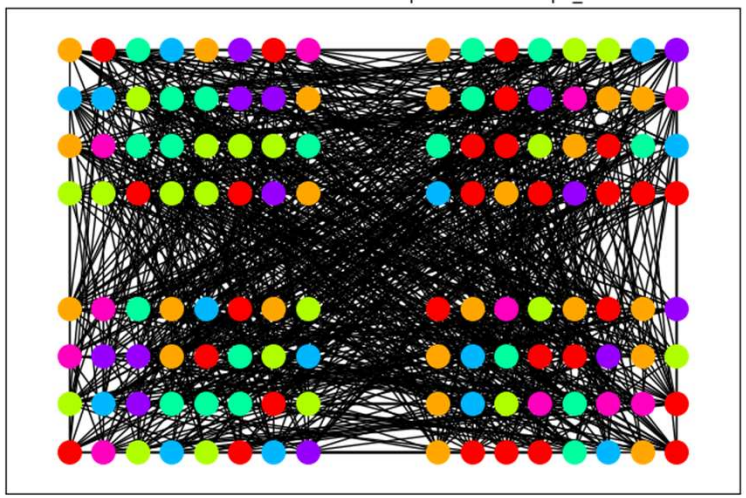
Leiden communities: 5 comms | AMI: 0.3865 | z_out: 8



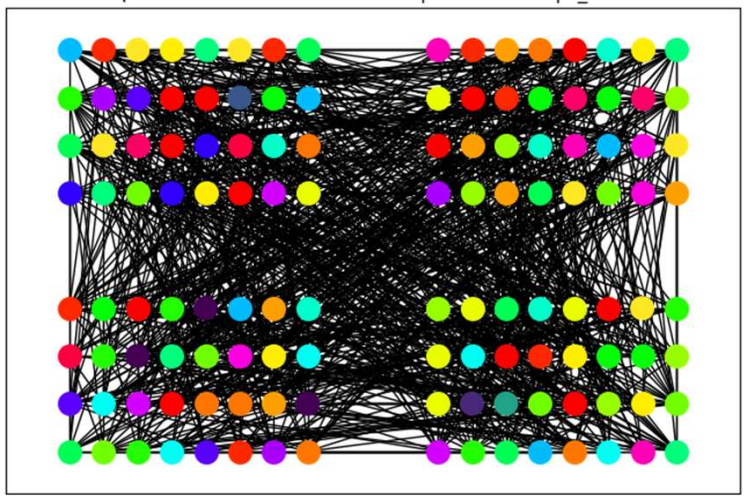
Walktrap communities: 5 comms | AMI: 0.5764 | z_out: 8



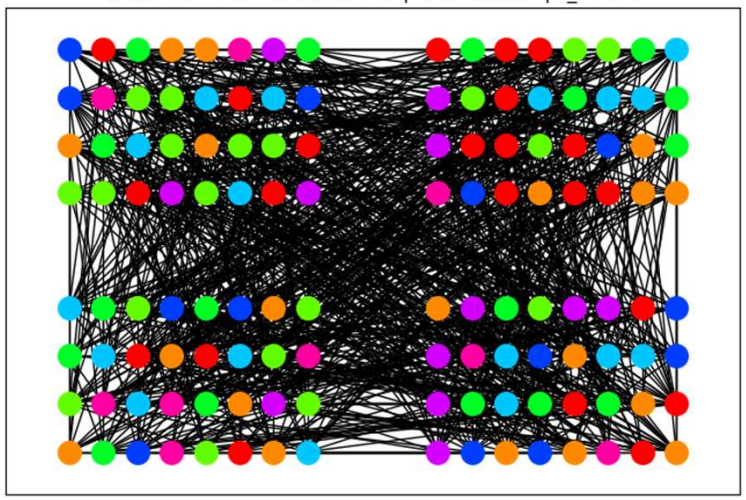
Louvain communities: 7 comms | AMI: -0.0170 | z_out: 12



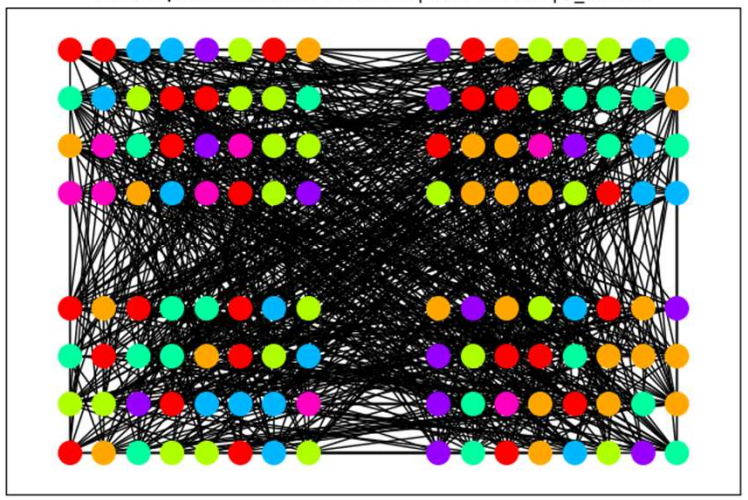
Surprise communities: 26 comms | AMI -0.0195 | z_out: 12



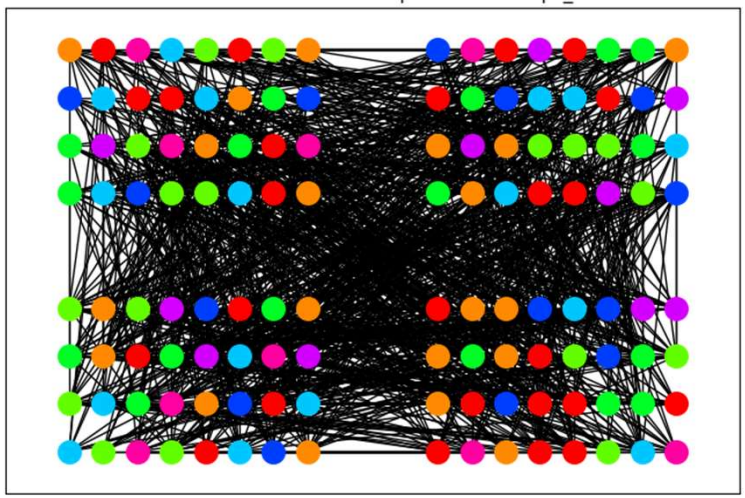
Leiden communities: 8 comms | AMI: -0.0050 | z_out: 12



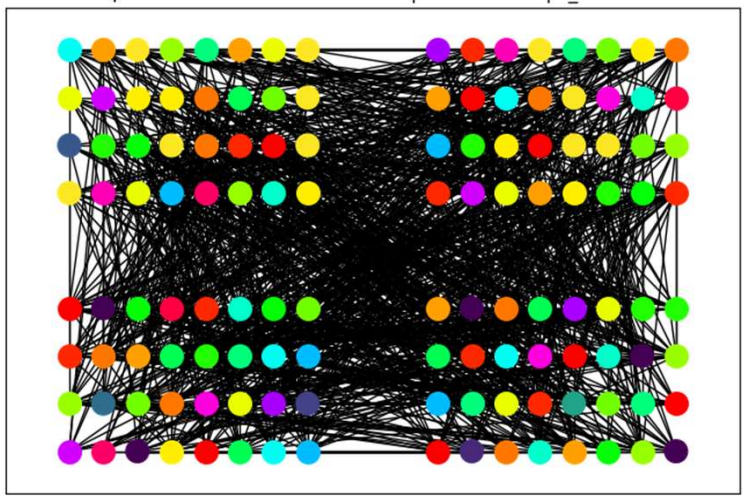
Walktrap communities: 7 comms | AMI: 0.0053 | z_out: 12



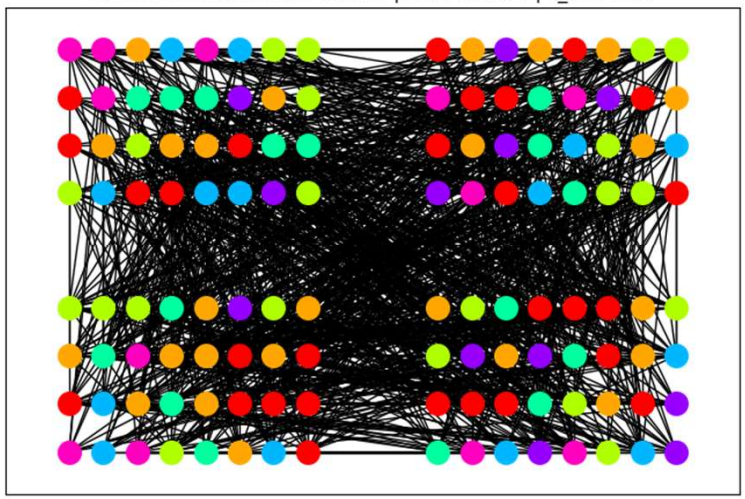
Louvain communities: 8 comms | AMI: -0.0344 | z_out: 15.5



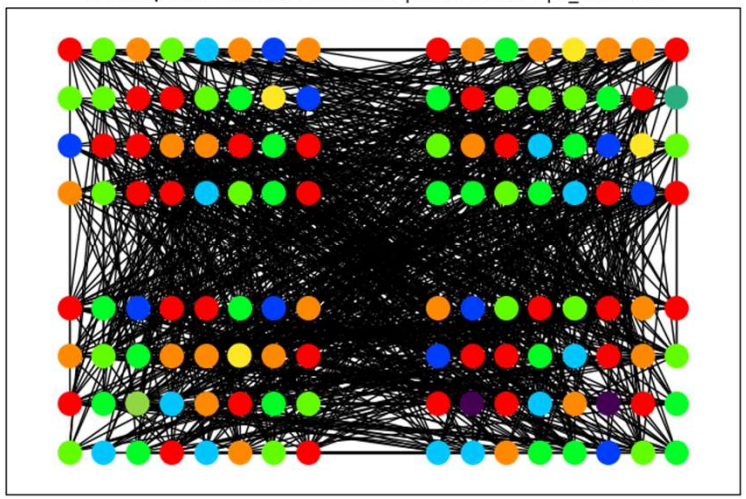
Surprise communities: 26 comms | AMI -0.1132 | z_out: 15.5



Leiden communities: 7 comms | AMI: -0.0280 | z_out: 15.5



Walktrap communities: 8 comms | AMI: -0.0394 | z_out: 15.5

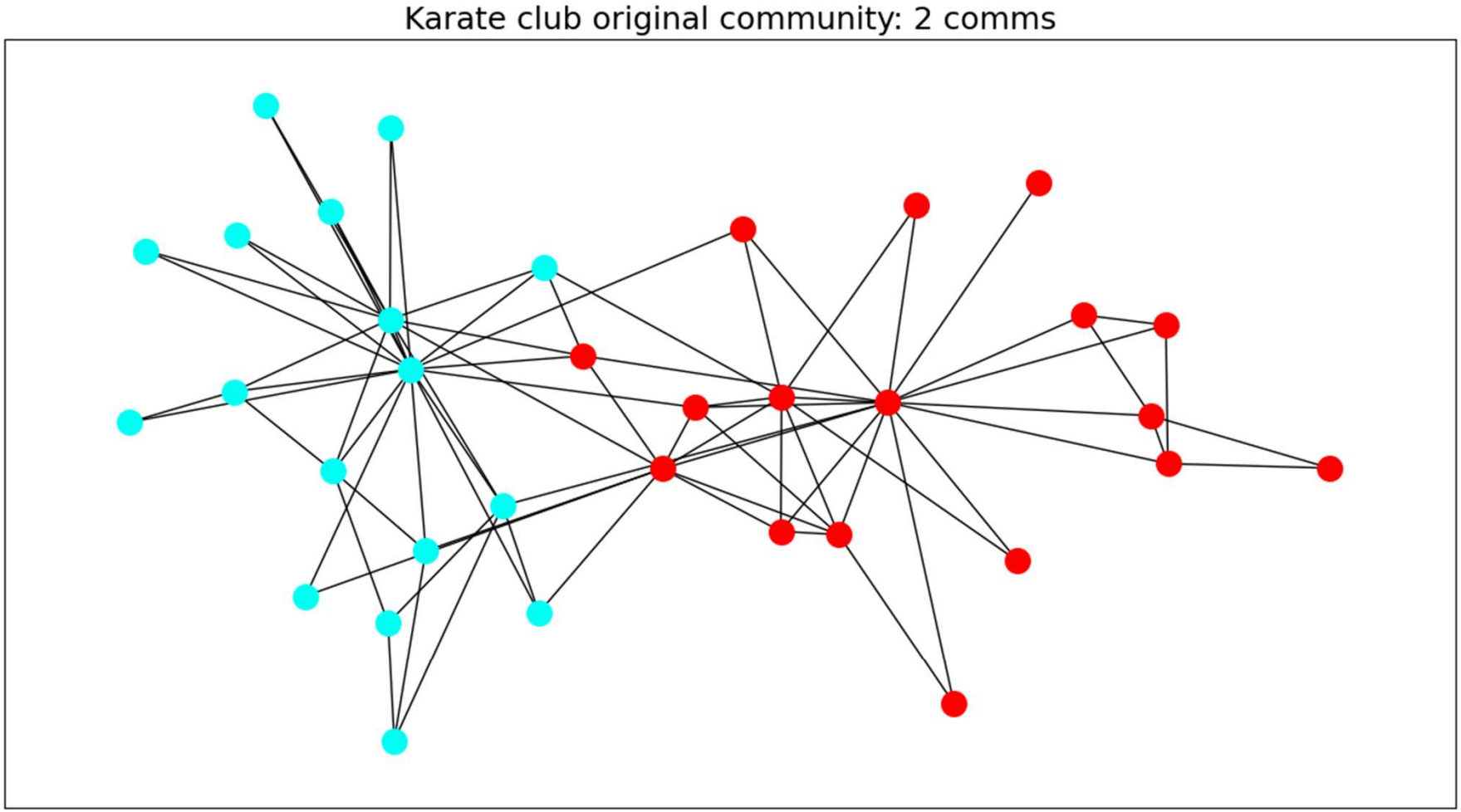


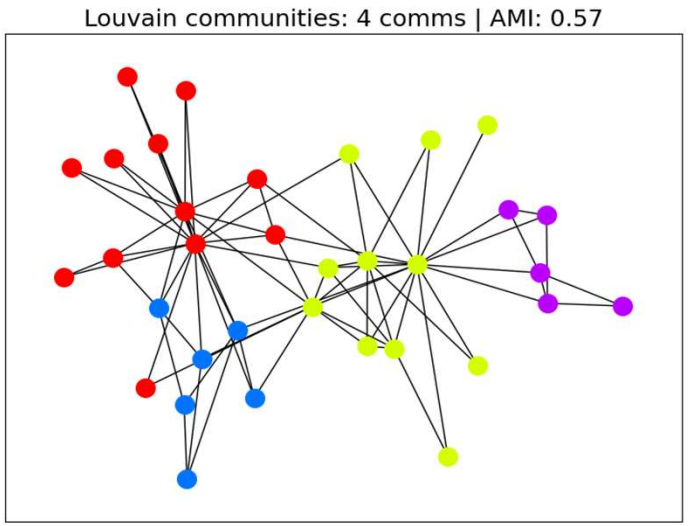
Observe algorithms on real networks

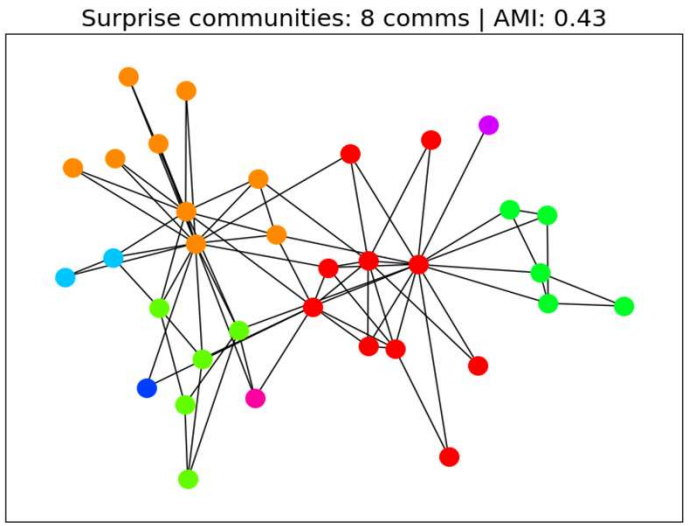
- The Zachary's Karate Club network:
  - 34 nodes
  - 78 edges
  - 2 ground truth communities
- The email-Eu-core network:
  - 1005 nodes
  - 25571 edges
  - 42 ground truth communities

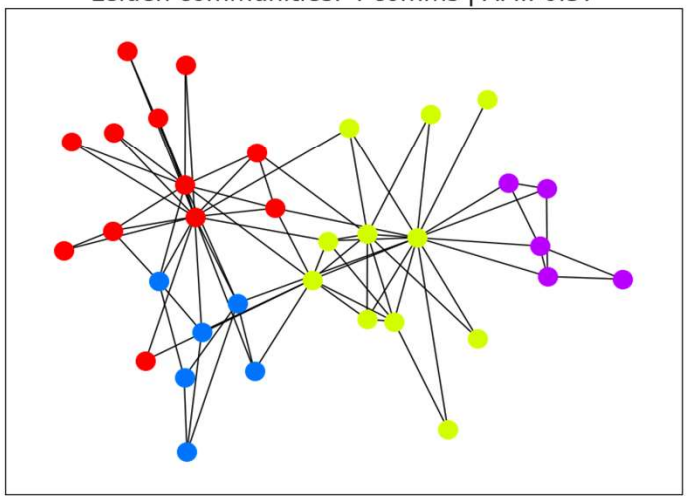
References:

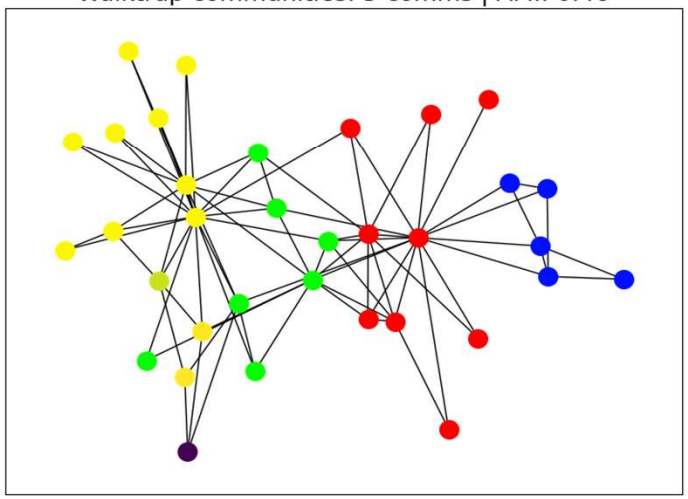
- https://networkx.org/documentation/stable/reference/generated/networkx.generators.social.karate_club_graph.html#networkx.generators.social.karate_club_graph
- https://snap.stanford.edu/data/email-Eu-core.html

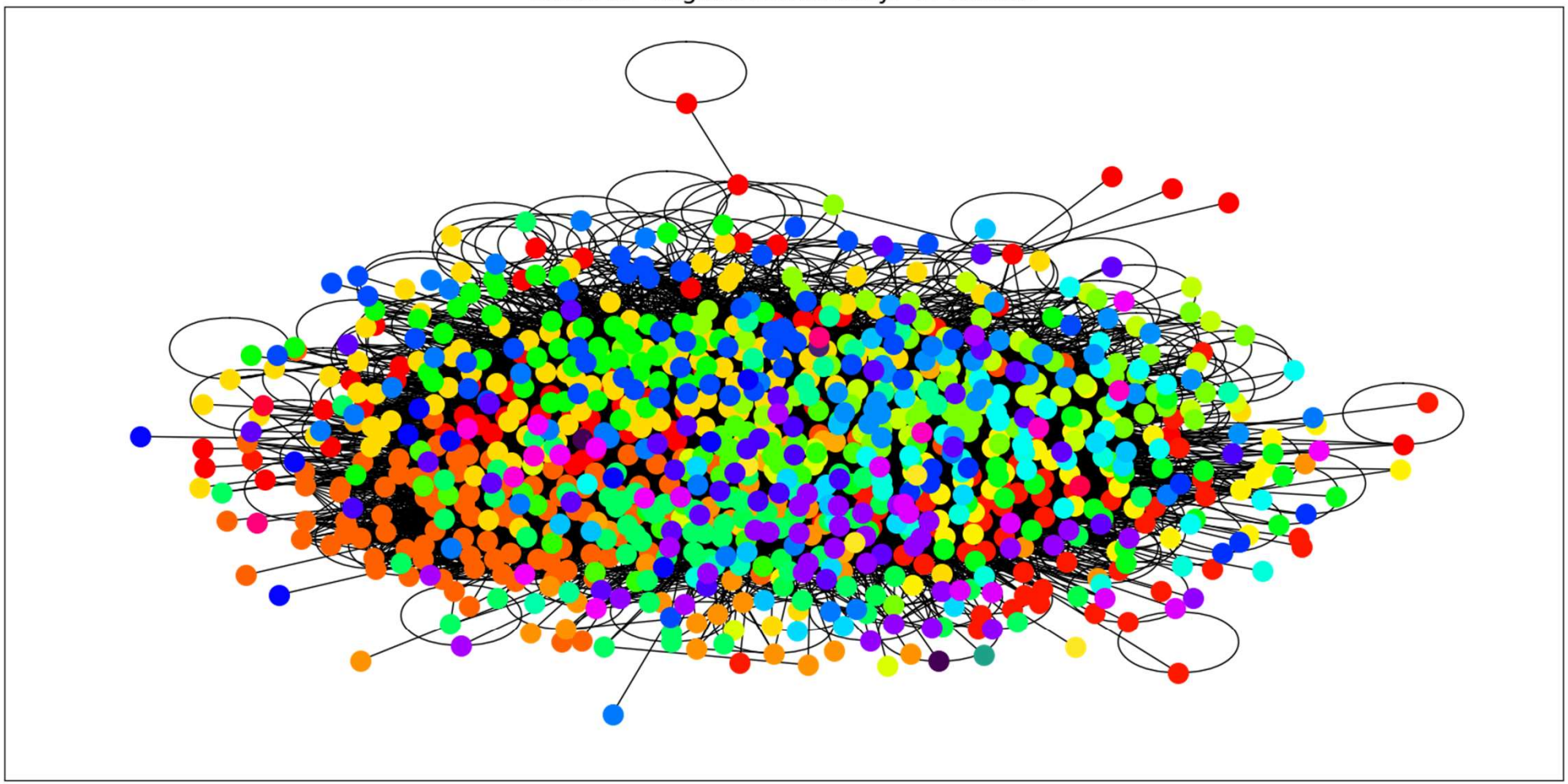

Karate club original community: 2 comms


Louvain communities: 4 comms | AMI: 0.57

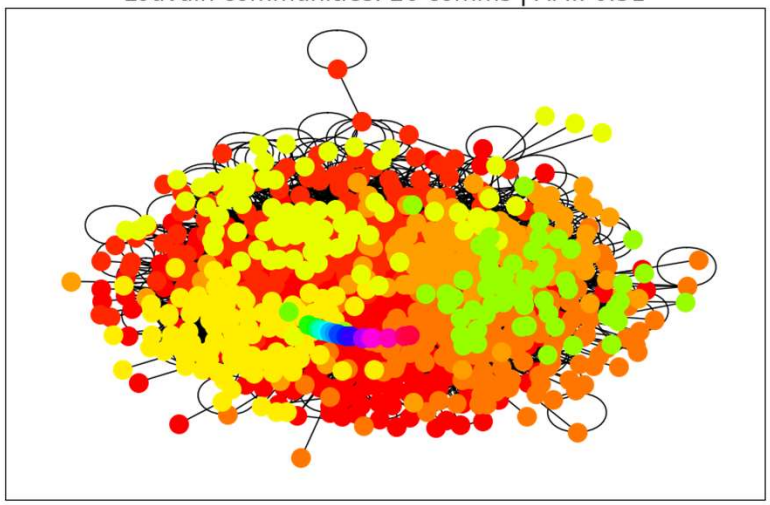

Surprise communities: 8 comms | AMI: 0.43


Leiden communities: 4 comms | AMI: 0.57

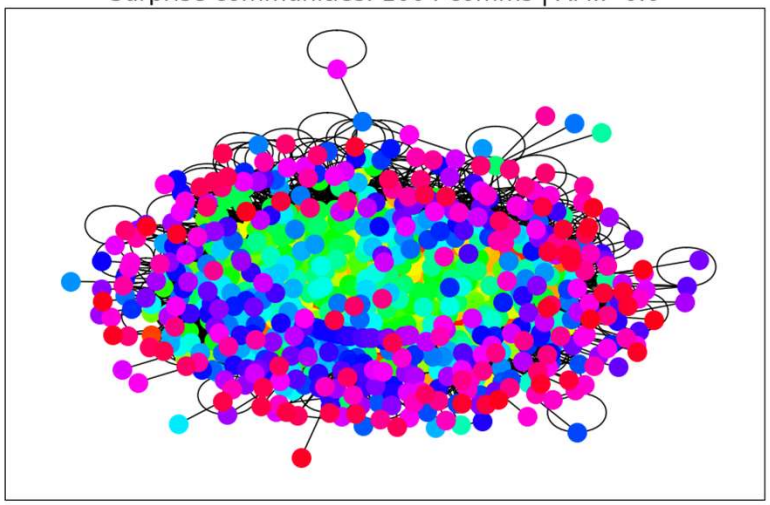

Walktrap communities: 5 comms | AMI: 0.46


Email EU original community: 42 comms

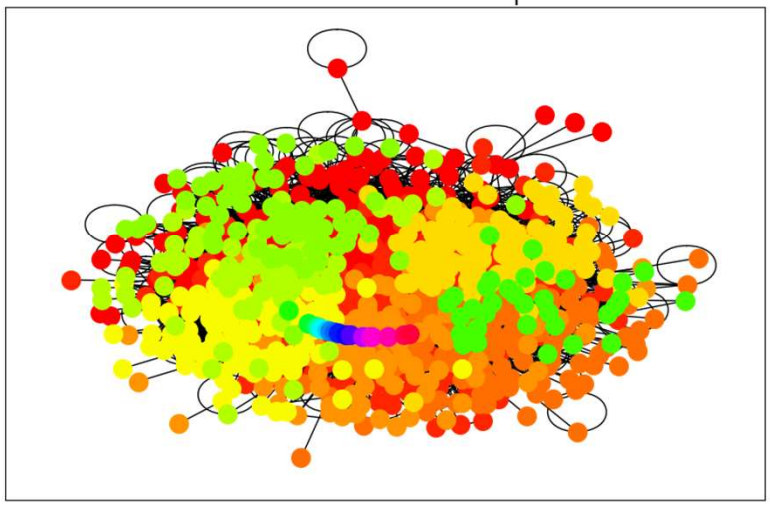
Louvain communities: 26 comms | AMI: 0.51



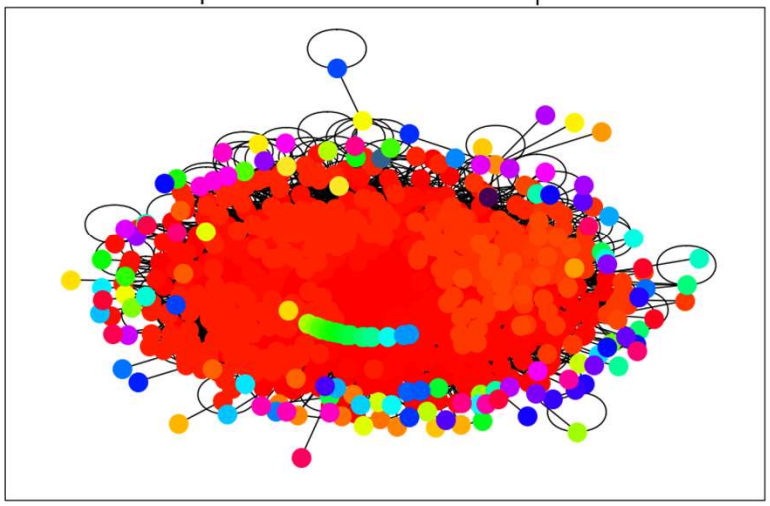
Surprise communities: 1004 comms | AMI: -0.0



Leiden communities: 28 comms | AMI: 0.56



Walktrap communities: 145 comms | AMI: 0.47



Conclusion

- Surprise Community performs worse compared to other algorithms.
- It has an earlier drop off point compared to others.
- It also tends to explode with the number of algorithms found.

- Louvain, Leiden, and Walktrap perform similar on test beds.
- However, Walktrap tends to perform slightly worse on real networks.

Thank you for your attention

Q&A