Network discovery using Reinforcement Learning

Harshavardhan Kamarthi[1], Priyesh Vijayan[3], Bryan Wilder[4], Balaraman Ravindran[1,2], and Milind Tambe[4]

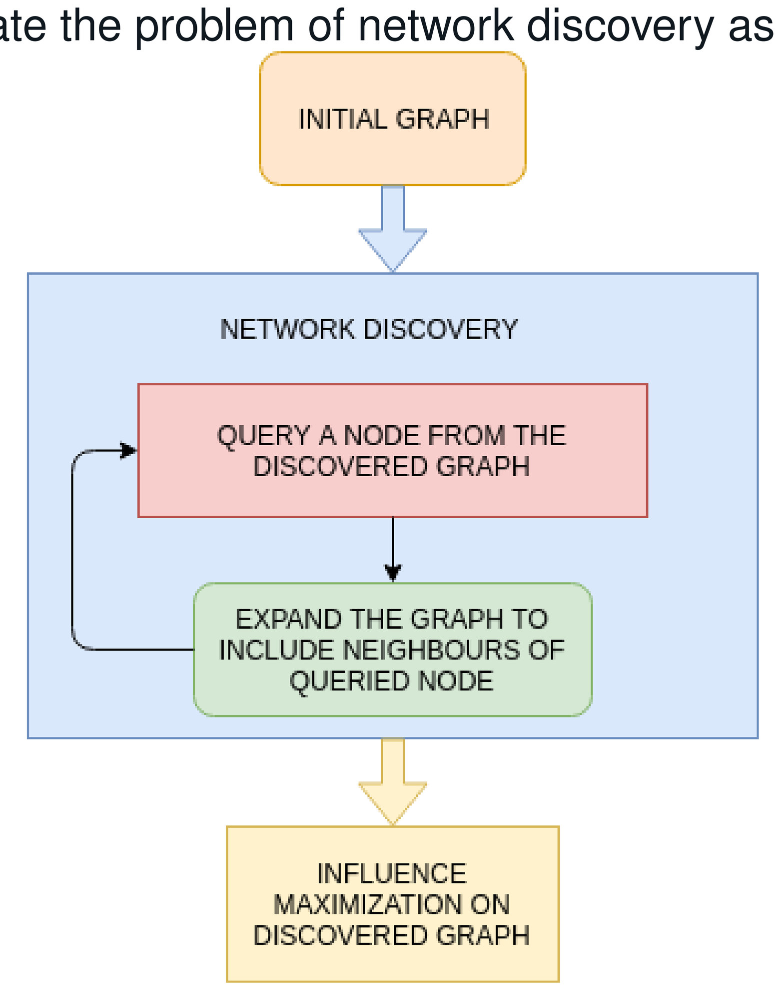
[1]Dept. of Computer Science and Engineering, Indian Institute of Technology Madras
[2]Robert Bosch Centre for Data Science and AI, Indian Institute of Technology Madras
[3]School of Computer Science, McGill University and Mila
[4]Center for Research in Computation and Society, Harvard University

Background

The goal of influence maximization is to pick influential nodes from a social network as peer leader to help disseminate information to maximum number of nodes in the network. However, real-world applications of influence maximization are often limited by the **high cost of collecting network data**.
Current state-of-the-art methods [4, 2] rely on hand-crafted sampling algorithms to sample nodes in a carefully constructed order and choose opinion leaders from this discovered network to maximize influence spread in the complete network.

Network Discovery Problem

We formulate the problem of network discovery as below.

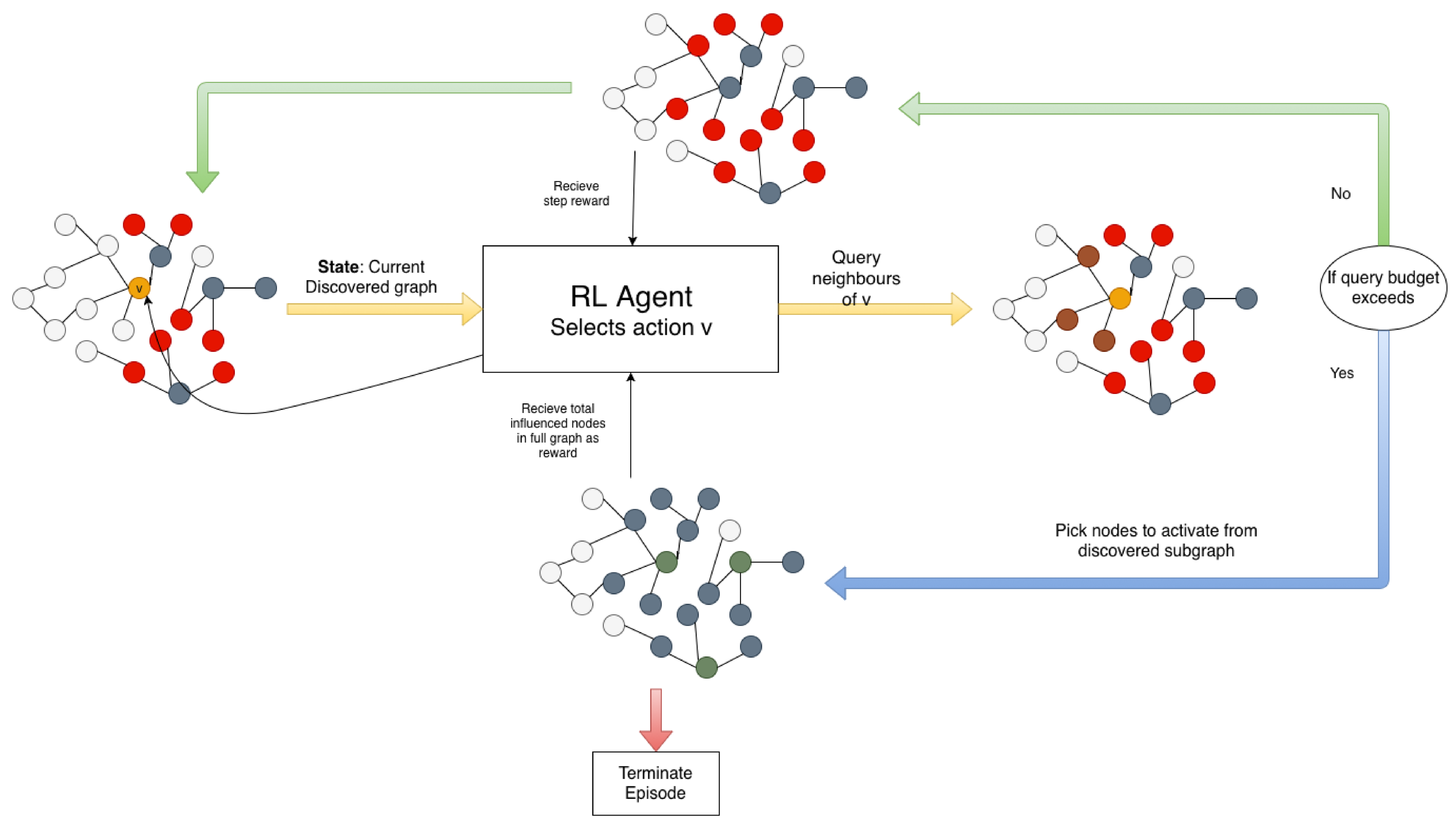


Each node that is surveyed reveals its neighbours, and the goal is to carefully select the nodes to be surveyed to choose an influential set of seed nodes.

Previous Methods

1. **RECOMMEND**: We query a node at random and its neighbors and then activate the neighbor with the maximum degree.
2. **SNOWBALL**: We start by querying a node at random and its neighbors and then activating the neighbor with the maximum degree. Then we again query neighbours of the node newly activated and repeat.
3. **ARISEN** [3]: Detect communities via random walks and

Markov Decision Process Formulation



Step Reward

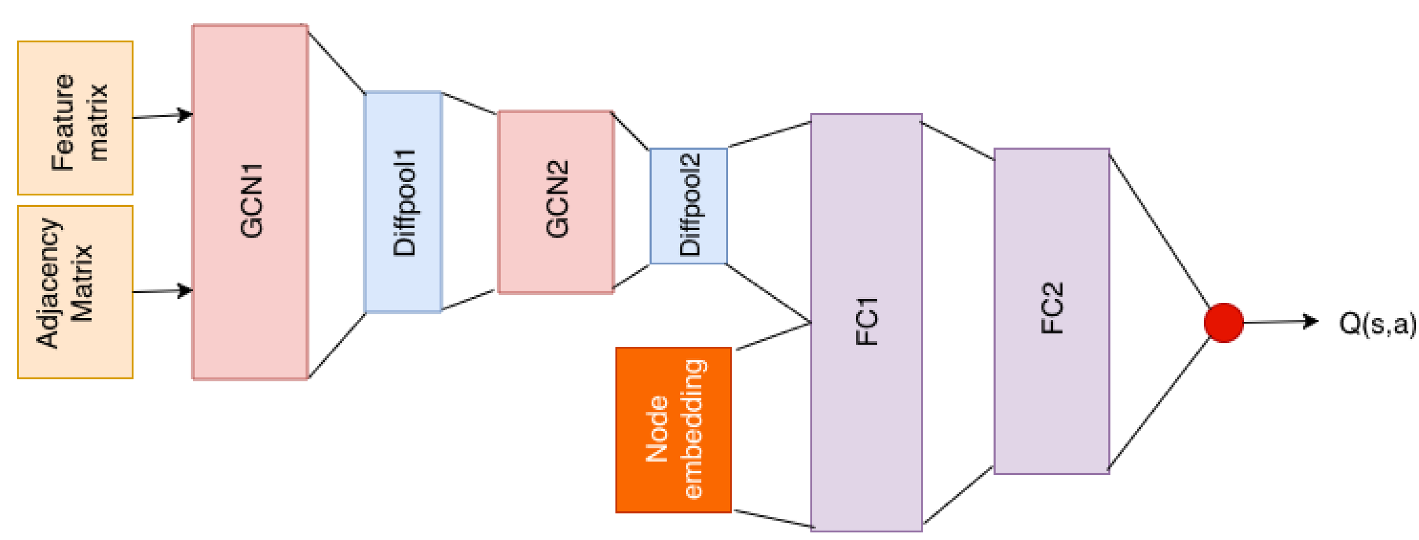
$$R_{p,t} = \frac{|V(G_t)| - |V(G_{t-1})|}{|V(G^*)|} \quad (1)$$

Influence reward (last step)

$$R_s = \frac{I_{G^*}(G_T) - CHANGE(G^*)}{OPT(G^*) - CHANGE(G^*)} \quad (2)$$

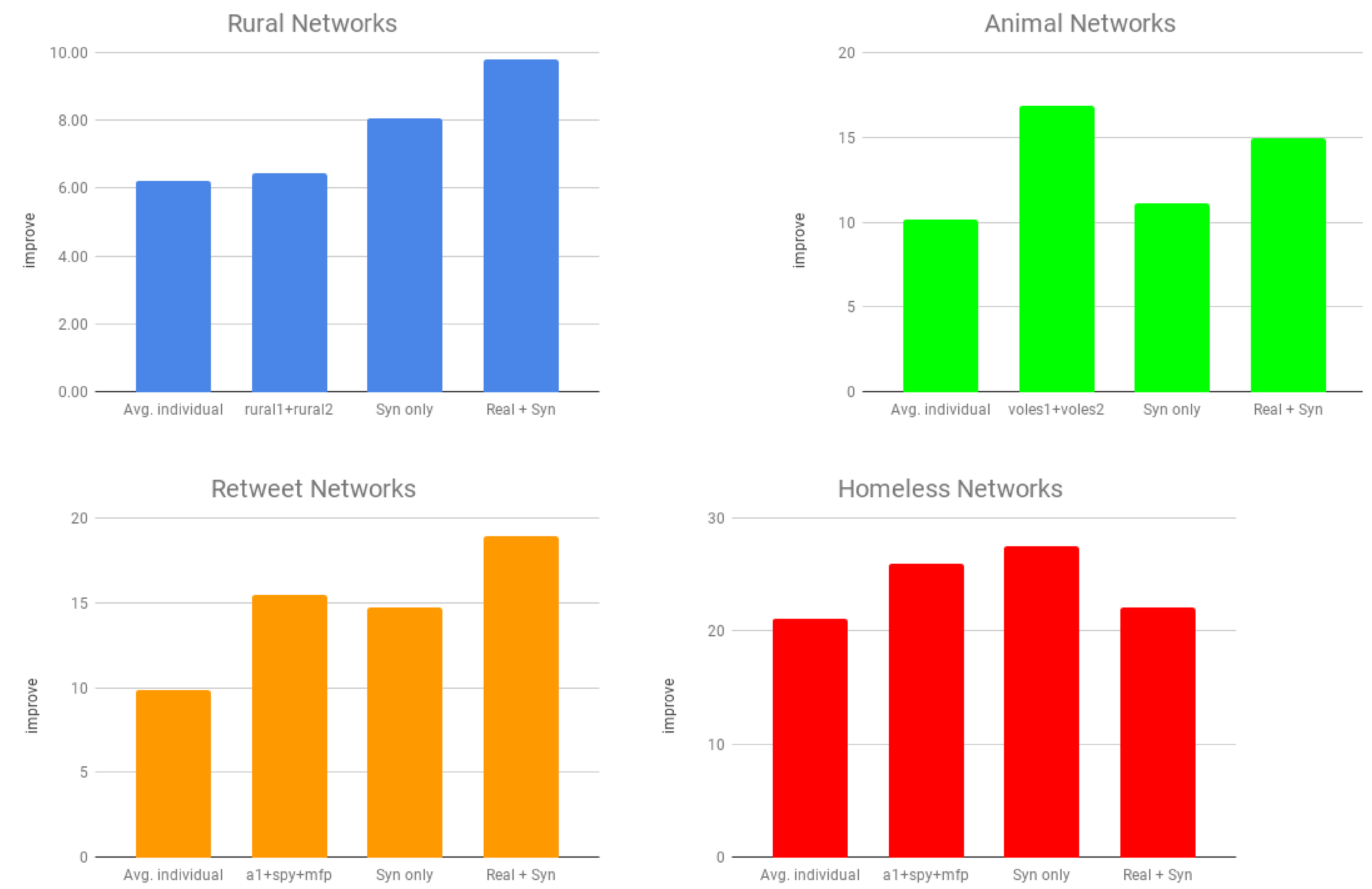
Network Architecture

We introduce a modified version of Deep Q-learning framework to account for learning graph and node representations.



State Representation
We use a neural network with permutation invariant graph convolutional layers and differentiable pooling layers [5] to obtain graph representations. While GCNs refine node features by message passing and aggregating, differential pooling seeks to learn a global representation of the graph by aggregating node features in a hierarchical manner.
Node representation

Results



- Training with multiple networks gives better performance gains compared to average performance gains received by training on single networks.
- Using only synthetic graphs are better than the average of the scores of models trained fro individual networks.

Insights from Policy learnt

- **Size of Network:** DQN policies almost always discover larger graphs tha CHANGE.
- The DQN policy tends to pick nodes of higher **degree centrality** and **be tweenness centrality** *with respect to the full graph* compared to CHANG especially during later stages of the episode (during steps 4 and 5).

References

[1] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM. 20 pp. 701–710.

[2] Bryan Wilder et al. "End-to-end influence maximization in the field". In: *Proceedings of the 17th Internatio Conference on Autonomous Agents and MultiAgent Systems.* International Foundation for Autonomous Agents a Multiagent Systems. 2018, pp. 1414–1422.

[3] Bryan Wilder et al. "Maximizing influence in an unknown social network". In: *Thirty-Second AAAI Conference Artificial Intelligence.* 2018.

[4] Amulya Yadav et al. "Bridging the Gap Between Theory and Practice in Influence Maximization: Raising Awa ness about HIV among Homeless Youth." In: *IJCAI.* 2018, pp. 5399–5403.

[5] Zhitao Ying et al. "Hierarchical graph representation learning with differentiable pooling". In: *Advances in Neu