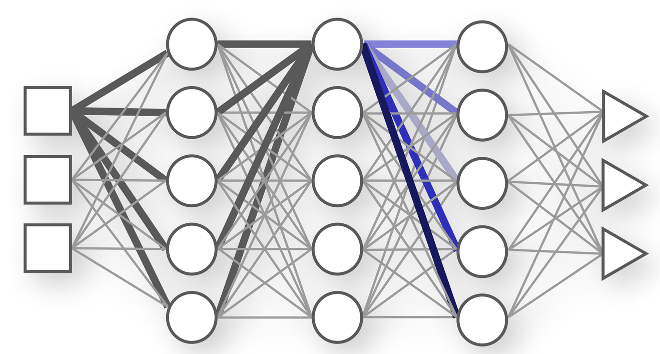
Network Defense Against Adversarial, Deep Learning Equipped Agents

Jordan Lanctôt[1], Sean P. Cornelius[1,2,†]

[1] Department of Physics, Ryerson University, Toronto, ON M5B 2K3, Canada
[2] Center for Complex Network Research, Northeastern University, Boston, MA 02115, USA
[†] To whom correspondence should be addressed

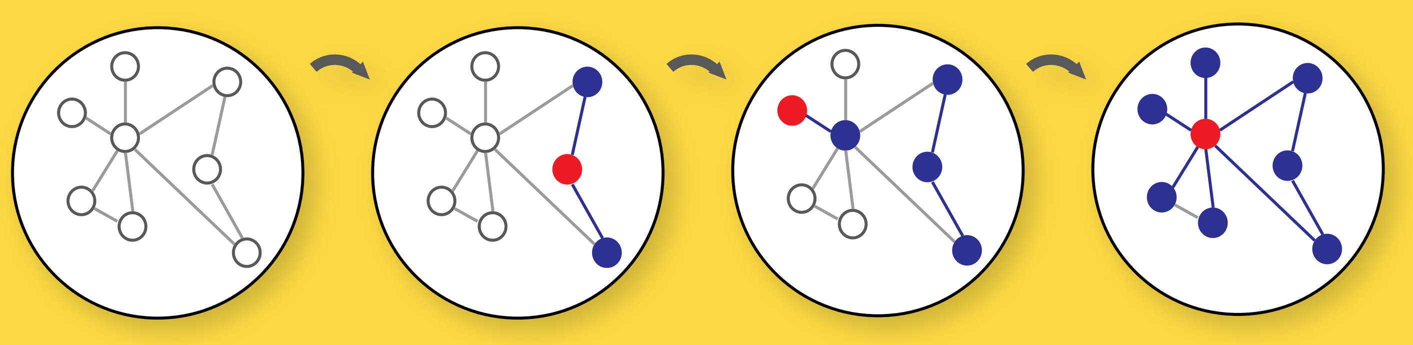
1 Rationale

NEURAL NETWORKS



- Neural networks are a parallel series of linear combinations, neurons, applied in a series of layers.
- Each linear combination has a non-linear function (ReLU) applied to it before propagating the resulting value. [1][4]
- Through updating the weights of the linear combinations, the neural net can universaly approximate functions, transforming an input vector into an output vector of desired dimensions.

GRAPH PERCOLATION



- Graph percolation is the process where selecting a node adds it and its neighbour's to a subset.
- Red nodes indicates nodes selected, blue nodes indicate the percolated nodes.
- Nodes that have been chosen (red) cannot be chosen in subsequent choices.
- Percolation ends when the subset nodes make up a particular fraction of the total graph.

BELLMAN EQUATION

$$Q(s,a) = r(s,a) + \max_{a' \in \Gamma(x)} \{\beta Q(s',a')\}$$

- Q is the expected return of rewards until the terminal state
- r(s,a) is the reward for taking action, a, for the state, s.
- s', a' are subsequent state-action pairs.
- β modulates priority on early vs late rewards.

STRUCTURE2VECTOR (S2V) [1]

$$\mu_v^{(t+1)} \leftarrow ReLU(\theta_1 x_v + \theta_2 \sum_{u \in N(v)} \mu_v^{(t)} + \theta_3 \sum_{u \in N(v)} ReLU(\theta_4 w(u,v)))$$

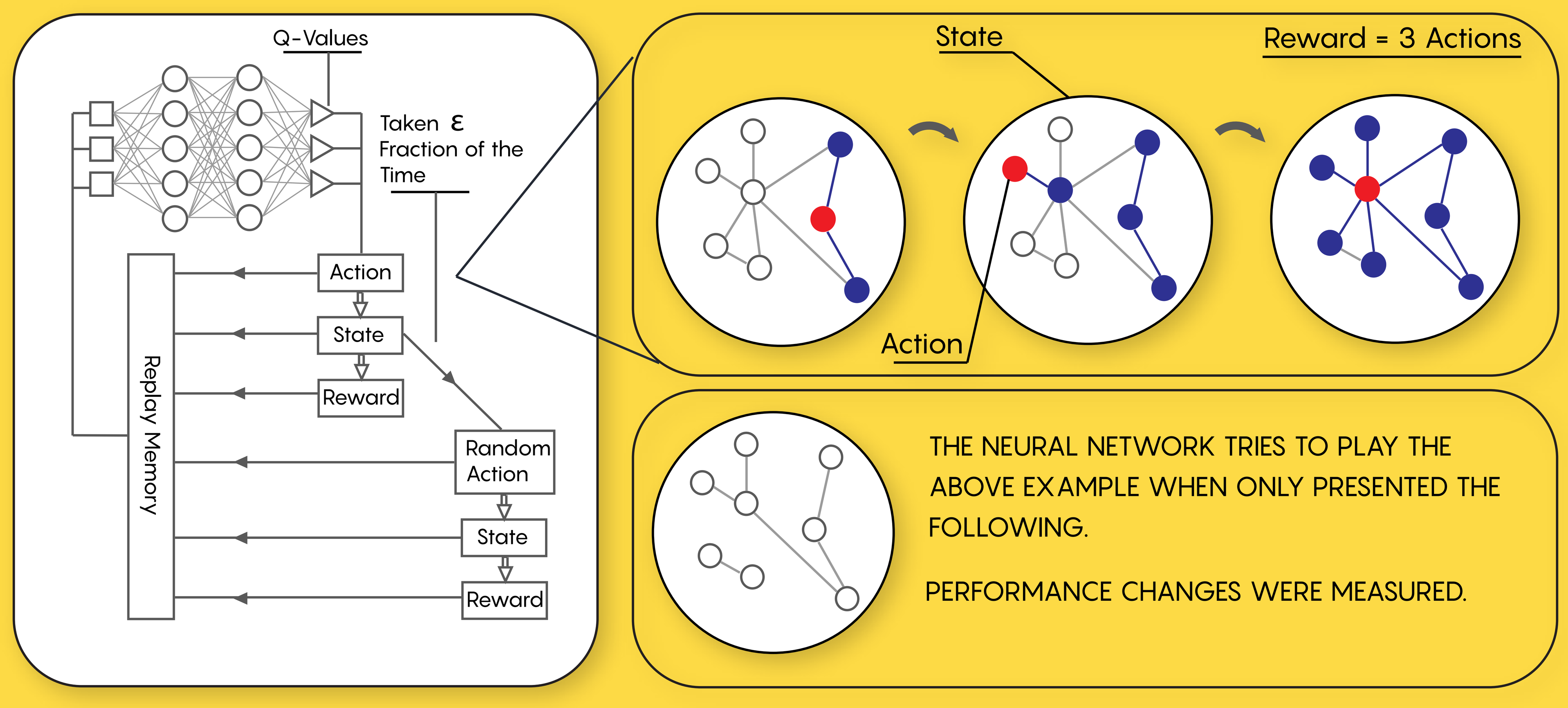
- S2V allows for recurssive calls to a neural network to learn to embed a graph in a N-dimensional space.
- The θs represent hidden layers.
- We account for factors based on node parameters (x), embedding of neighbours, and link parameters (w).

2 Objective & Hypothesis

Objective: To determine the capacity of neural nets to learn key features of networks, and how this capacity to learn changes as the percentage of network features concealed from the neural net is increased.

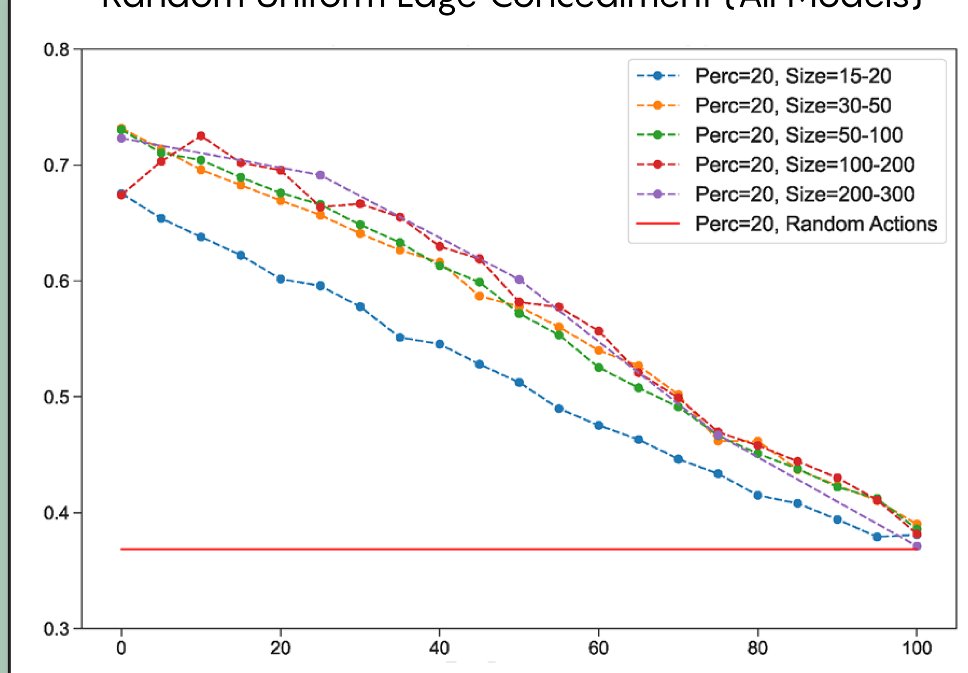
Hypothesis: Machine learning will have the capacity to discover key network features at decreasing rates as more of network information is hidden from the neural network. [2] [3],

3 Methods



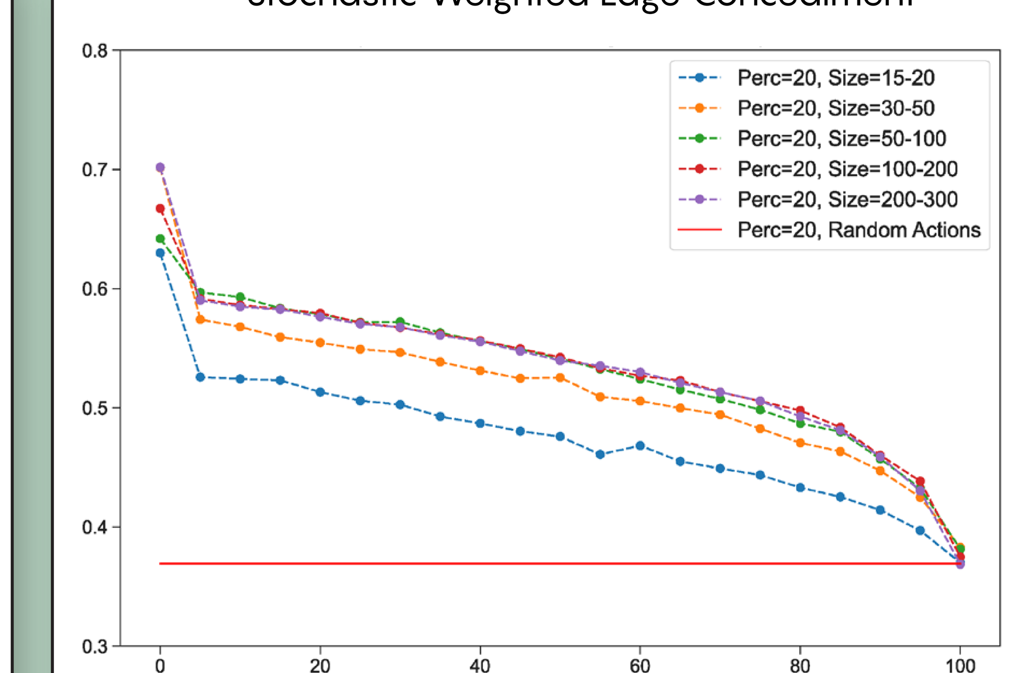
4 Results

Random Uniform Edge Concealment (All Models)



- Any link has equal probability of being concealed.
- Edges are concealed until a required fraction of the links are concealed.

Stochastic Weighted Edge Concealment



- Links are ordered by the product of how many incident links the nodes they join have.
- Edges are concealed with a probability of their link weight divided by the total link weight of the entire graph.

Training Requirements For Random Uniform Edge Concealment

Network Size	Area (All Models)	Area (0.0 Model)	Area (0.5 Model)	Area (0-0.95 Model)
15-20	0.13275	0.12930	0.14490	0.14508
30-50	0.18738	0.17184	0.17193	0.18667
50-100	0.18663	0.20129	0.19666	0.19311
100-200	0.20822	0.12452	0.19526	0.12469
200-300	0.20805 *	0.18448	0.20636	0.09451

Contextualizing Weighted Edge Concealment

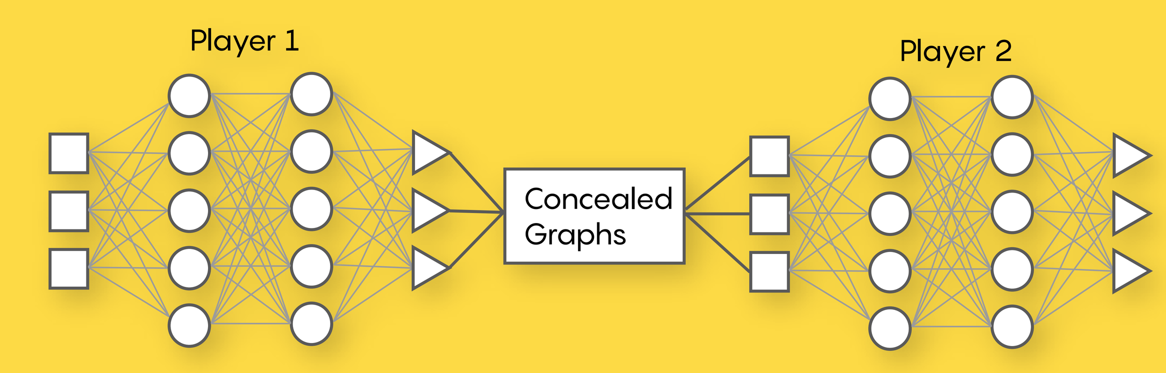
Network Size	Area (0-0.95 Model)	Area (Deterministic)	Area (Stochastic)
15-20	0.14508	0.17626	0.10443
30-50	0.18667	0.20822	0.14615
50-100	0.19311	0.18237	0.16311
100-200	0.12469	0.12756	0.16515
200-300	0.09451	0.21049	0.16452

5 Conclusions

- Network Concealment approaches yield trends in the relationship between concealment percentage and percolation performance.
- Percolation performance only ever approached the performance of random actions when concealment percentages approached 100%.
- Network Concealment Heuristics were largely similar in performance relative to the random action baseline.
- Increasing the stochastic measures taken during concealment during training reduced neural network performance including:
  - Providing uniformly random concealment percentage during training instead of constant concealment percentages.
  - Concealing links with a probability proportional to the product of the degrees that they connect.
- Implications are that graphs cannot be defended from deep learning equipped agents with rigid heuristics.

6 Future Directions

- Conclusions motivate the concept that increased variance within the supplied experiences in the training data might decrease performance of deep learning equipped agents.



- Replace concealment heuristics with a Deep-Learning Agent
- Replace Q-Value Outputs with π-Value to allow for interpretting the output as a probability of taking a given action instead of determining the best action.

7 References

[1] Dai, H., Khalil, E. B., Zhang, Y., Dilkina, B. & Song, L. Learning combinatorial opti- mization algorithms over graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, 6351—6361 (Curran Associates Inc.).

[2] Bennett, H., Reichman, D. & Shinkar, I. On percolation and NP-hardness 54, 228—257. URL https://onlinelibrary.wiley.com/doi/10.1002/rsa.20772.

[3] Morone, F. & Makse, H. A. Influence maximization in complex networks through optimal percolation 524, 65—68. URL http://arxiv.org/abs/1506.08326. 1506. 08326.

[4] Agarap, A. F. Deep learning using rectified linear units (ReLU) URL https://arxiv.org/abs/1803.08375v2.

8 Acknowledgements

I would like to thank all of the support of Dr. Sean P. Cornelius and researchers within his lab for supporting the development of this research.