Extended Abstract

The Self-Loop Paradox
Investigating the Impact of Self-Loops on GNNs

Moritz Lampert
moritz.lampert@uni-wuerzburg.de
Ingo Scholtes
ingo.scholtes@uni-wuerzburg.de

Abstract

Many GNNs add self-loops to a graph to include information about a node itself at each layer. However, if the GNN has multiple layers, this information can return to its origin via cycles in the graph topology. Intuition suggests that this "backflow" of information should be larger in graphs with self-loops compared to graphs without. We counter this intuition and show that this phenomenon, which we call the self-loop paradox, can depend both on the number of GNN layers k and whether k is even or odd.

Our Contributions:

- Theoretical investigation using random graph ensembles
- Empirical validation with synthetic datasets
- Further investigations on real-world graphs

Background

Random Graphs

We adopt the configuration model [1] that generates random undirected graphs $G = (\boldsymbol{V}, \boldsymbol{E})$ based on a given degree sequence $S = (d_v)_{v \in \boldsymbol{V}}$. The expected properties of these random graphs can be studied analytically:

- The expected degree:

$$\langle d \rangle = \frac{1}{|\boldsymbol{V}|} \sum_{d_v \in S} d_v$$

- The expected degree of a random neighbour of a randomly chosen node:

$$\langle d_N \rangle = \frac{\langle d^2 \rangle}{\langle d \rangle}$$

  where $\langle d^2 \rangle$ is the second raw moment. [2]

- The expected number of walks of length k to v: [2]

$$\mathbb{E}\left(\sum_{u \in \boldsymbol{V}} A_{uv}^k\right) = \langle d \rangle \langle d_N \rangle^{k-1} = \langle d \rangle \left(\frac{\langle d^2 \rangle}{\langle d \rangle}\right)^{k-1}$$

Graph Neural Networks

To measure the influence of input features on a prediction, we make use of a finding by Chen et al. [3] in the context of over-squashing for Message Passing GNNs (MPNNs) [4]:

Lemma 1: *If the MPNN passes messages along all k-length walks from u to v with equal probability, then the relative influence of input feature $\boldsymbol{h}_u^{(0)}$ on the output $\boldsymbol{h}_v^{(k)}$ is on average*

$$\mathbb{E}\left(\frac{\partial \boldsymbol{h}_v^{(k)} / \partial \boldsymbol{h}_u^{(0)}}{\sum_{\acute{u} \in \boldsymbol{V}} \partial \boldsymbol{h}_v^{(k)} / \partial \boldsymbol{h}_{\acute{u}}^{(0)}}\right) = \frac{\bar{A}_{uv}^k}{\sum_{\acute{u} \in \boldsymbol{V}} \bar{A}_{\acute{u}v}^k},$$

where $\bar{A} = A + I$ is the adjacency matrix with added self-loops.

Theoretical Analysis

Relatively speaking, more walks are expected to lead back to a node itself in graphs without self-loops compared to graphs with self-loops:

Lemma 2: *Given a graph G generated using the configuration model with adjacency matrix A without self-loops and its counterpart with self-loops $\bar{G}$ with $\bar{A} = A + I$, the proportion of cycles of length 2 from a node v to itself out of all walks of length 2 ending in v is larger in G than in $\bar{G}$:*

$$\frac{\mathbb{E}(A_{vv}^2)}{\mathbb{E}(\sum_{u \in \boldsymbol{V}} A_{uv}^2)} > \frac{\mathbb{E}(\bar{A}_{vv}^2)}{\mathbb{E}(\sum_{u \in \boldsymbol{V}} \bar{A}_{uv}^2)}$$

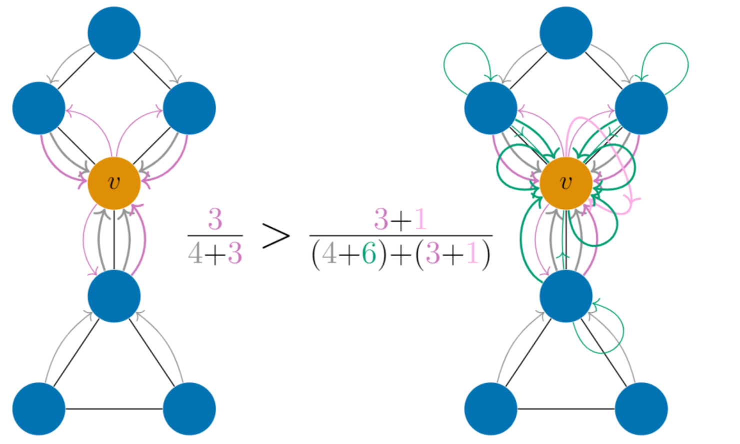

Figure 1: Example graph where the original undirected edges are coloured black. Cycles of node v with length 2 are represented in shades of purple. Walks of length 2 starting at other nodes are green if they include a self-loop and grey otherwise. The last step in each walk is highlighted.

Intuition: Although there is one more cycle when self-loops are added, there are two new walks expected from every neighbour due to their self-loops. This leads to more walks starting from other nodes so that the overall proportion of cycles out of all walks decreases. Figure 1 shows this effect on a concrete example graph.

Empirical Validation

Using the Stochastic Block Model (SBM) [5], random graphs are generated for a node classification task. We expect a dependency of the performance difference of GCNs [6] with and without self-loops on the number of layers k. In line with our theoretical findings, the results in Figure 2 show that self-loops can decrease the performance of a two-layered GCN in tasks where the own node features play an important role (bottom right).

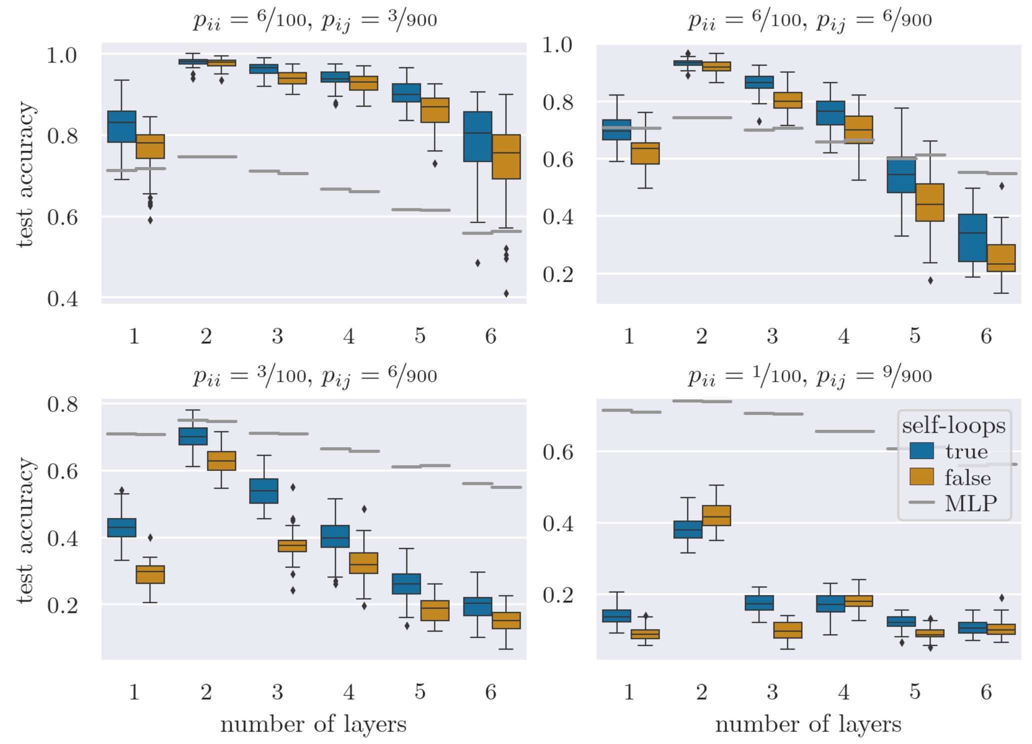

Figure 2: Test set accuracies of GCNs [6] with and without self-loops averaged over 50 runs. Random graphs are generated using SBM [5] with different cluster structures in each boxplot. The top left and the bottom right plot shows the results on graphs that exhibit the most cluster structure and no structure, respectively. The node features for each cluster are sampled from Gaussian distributions and do not change throughout the plots. The coloured boxes mark the range between the first and third quartiles. Whiskers stretch to the furthest outliers within three halves of that range while other outliers outside of the whiskers are marked with a rhombus. Grey lines show average accuracies of MLPs with the corresponding number of layers.

Real-World Examples

We repeat our evaluation from Figure 2 on 15 real-world datasets where the nodes own feature is useful for the task. The results in Figure 3 show that the parity of the number of message passing layers influences the performance of GCNs. In line with our analytical results, we find that for 11 out of 15 data sets the accuracy increase of a two-layer compared to a one-layer GCN is larger for an input graph without self-loops than a graph with self-loops. We suspect that this is due to the self-loop paradox.

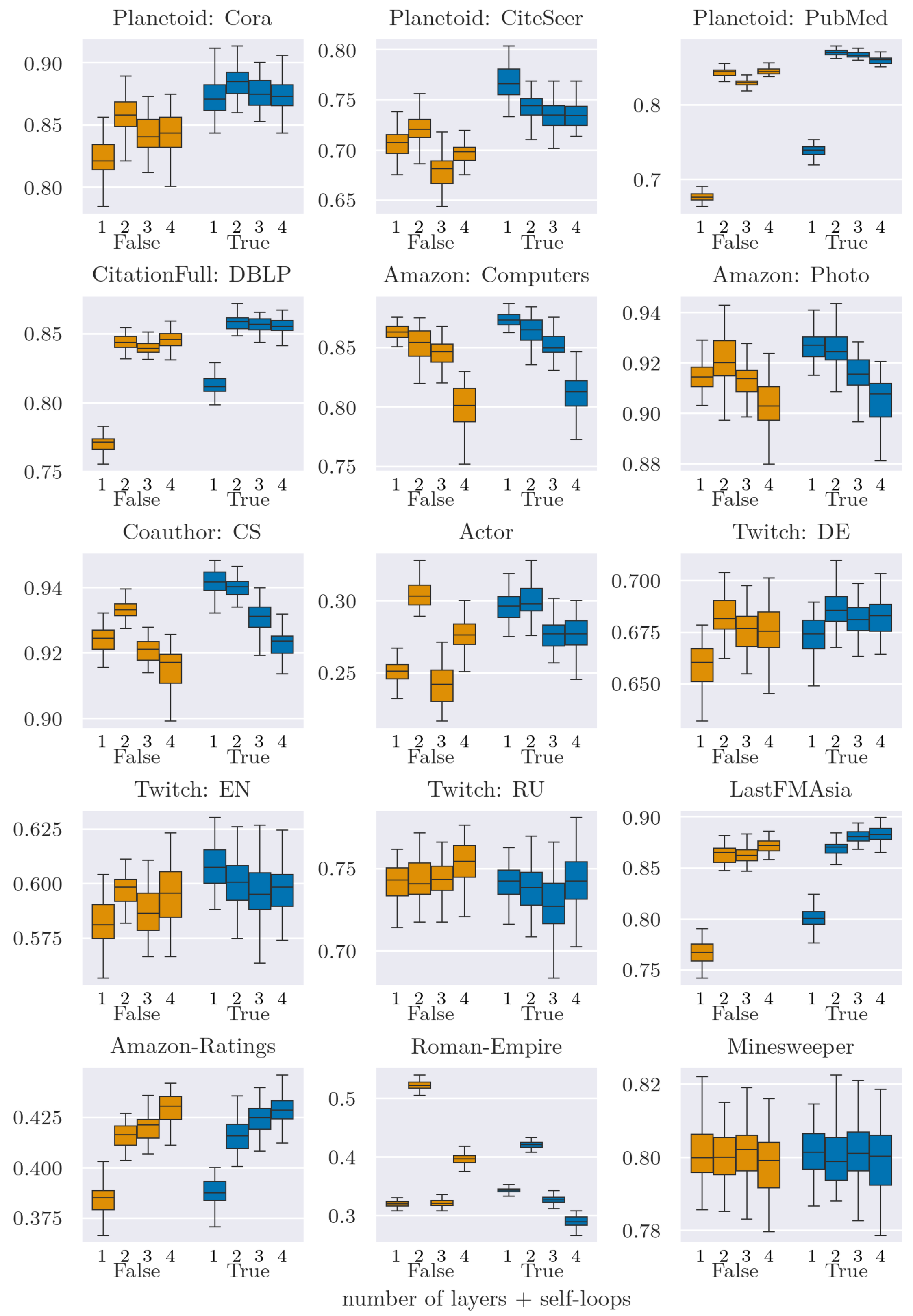

Figure 3: Test set accuracies of GCNs [6] with and without self-loops averaged over 50 runs as in Figure 2. Outliers outside of the whiskers are omitted for readability.

References

1. M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence," *Random Structures & Algorithms*, vol. 6, no. 2–3, pp. 161–180, 1995
2. M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Phys. Rev. E*, vol. 64, no. 2, p. 026118, 2001
3. J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural Message Passing for Quantum Chemistry," in *PMLR*, 2017, pp. 1263–1272.
4. R. Chen, S. Zhang, L. H. U, and Y. Li, "Redundancy-Free Message Passing for Graph Neural Networks," in *NeurIPS*, 2022.
5. P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.
6. T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *ICLR (Poster)*, OpenReview.net, 2017.

Code is available here