Benchmarking Graph Rewiring Approaches for Graph Attention Networks

Julia Balla

January 2023

Introduction

Graph neural networks are often prone to over-squashing in tasks that require propagating information over long distances. However, the prevalence of over-squashing for a given task has been shown to vary across different GNN architectures [1]. In the case of Graph Attention Networks, each node is able to use attention-based weighting to filter out incoming messages containing irrelevant information, which helps mitigate information bottlenecks. However, the attention mechanism does not entirely eliminate the problem of over-squashing in deep models. Most solutions to over-squashing rely on rewiring the input graph to avoid information bottlenecks. In this study, we investigate the performance of GATs with respect to various rewiring techniques. Specifically, we aim to compare the effectiveness of several popular rewiring-based solutions to over-squashing and identify those that best complement the attention mechanism. The code is available at https://github.com/julballa/gat_rewiring.

Methodology

We compare the performance of the base GAT model against the following graph rewiring methods:

- **Fully Adjacent layer (+FA)**: In the original paper which identified over-squashing as a standalone phenomenon, Alon and Yahav proposed to replace the last layer in an already extensively tuned model with a fully connected layer [1]. Critically, this method is also used to detect the presence of over-squashing; an increase in model performance upon connecting all of the nodes in the last layer indicates that the original model struggles to propagate information between distance nodes.

- **Virtual Node**: Rather than connecting all nodes to each other, Gilmer et al. introduced a "virtual node" which all other nodes could read from and write to [2]. The virtual node has its own weights and update function through which it aggregates global information. We follow the implementation in the Open Graph Benchmark graph-level GCN example, where the virtual node's update function is given by a 2-layer MLP [3].

- **Expander Graph Propagation (EGP)**: A recent paper by Deac et al. suggests a new rewiring approach based on propagating messages over expander graphs [4]. These graphs have a low diameter and a large Cheeger constant (any 2 subsets of vertices have many links between them, thus avoiding bottlenecks), making them a good candidate for message propagation. The expander graph is constructed as $G^{\mathrm{Cay}(n)} \in \mathrm{Cay}(\mathrm{SL}\,(2, \mathbb{Z}_n)\,; S_n)$, where the special linear group $\mathrm{SL}\,(2, \mathbb{Z}_n)$ denotes the group of 2 x 2 matrices with determinant 1 and entries that are integers modulo n which is generated by the symmetric group S_n. For a given input graph with $|V|$ vertices, the corresponding Cayley graph size is calculated as

$$n = \mathrm{argmin}_{m \in \mathbb{N}}\, n^3 \prod_{\text{prime } p \mid n} \left(1 - \frac{1}{p^2}\right) \geq |V|.$$

  We pre-generate the required Cayley graphs for our experiments using the SageMathCell web interface [5]. Since n may not always align with $|V|$, we sample the first $|V|$ nodes in the Cayley graph to obtain the new adjacency matrix. The procedure for expander graph propagation is then given by alternating between GAT convolution on the input graph for odd layers and on the expander graph for even layers.

- **All FA**: To provide more insight into a learning task's reliance on graph topology, we evaluate the performance of a GAT model in which each layer is fully adjacent. In tasks where the graph's structure is highly relevant to the prediction, we expect to see much poorer performance from the AllFA model since it has no access to any topological information.

Each method is evaluated on two graph-level classification datasets. We report the training accuracy to detect over-squashing by measuring the model's ability to fit the input data, as well as the test accuracy to measure the model's ability to generalize.

Datasets

The above approaches are evaluated on the MUTAG and ENZYMES datasets from the TUDataset [6]. These datasets have been shown to require long-range interactions between nodes and have been used as benchmarks for studying oversquashing [7]. MUTAG yields a binary classifcation task for predicting the mutagencity of molecular compounds. The task in ENZYMES is to classify protein structures across 6 different enzyme classes.

Experiments

Our baseline GAT model consists of single-head attention with ReLU activations. The node representations from the convolutional layers are aggregated via a global mean pooling operation before being fed into a final linear layer. For the virtual node method, we use a 2-layer MLP for the virtual node's update function after pooling the other node representations via global mean pooling. To speed up the training process, we employ graph mini-batching and early stopping such that the training terminates once the training accuracy decreases for a set number of epochs. For each dataset, we use the same hyperparameters (shown in Table 1) across all models to evaluate the effects of the rewiring methods irrespective of parameter tuning. Hyperparameters are chosen based on the best-tuned GNN models for similar experiments in literature [7]. We perform 10-fold cross-validation by randomly splitting the data into 10 disjoint training and test sets. The mean and standard error for the classification accuracy is recorded in Table 2.

Table 1: Shared hyperparameters for each dataset.

Hyperparameter	**MUTAG**	**ENZYMES**
`num_layers`	4	4
`hidden_dimension`	64	64
`dropout`	0.3	0.3
`batch_size`	32	32
`epochs`	500	500
`patience`	5	5
`learning_rate`	0.001	0.001

Table 2: Classification accuracy with standard error (%).

Dataset	**GAT**	**+FA**	**Virtual Node**	**EGP**	**AllFA**
MUTAG (Train)	87.29 ± 0.10	89.84 ± 0.06	90.84 ± 0.12	88.06 ± 0.08	89.72 ± 0.01
MUTAG (Test)	76.78 ± 2.21	79.47 ± 1.98	78.42 ± 2.12	79.94 ± 2.05	79.41 ± 1.96
ENZYMES (Train)	43.07 ± 0.0	42.14 ± 0.01	70.63 ± 0.41	47.84 ± 0.01	37.30 ± 0.04
ENZYMEs (Test)	32.74 ± 0.09	31.53 ± 0.11	41.59 ± 0.70	35.72 ± 0.12	28.71 ± 0.98

Results

The improvement in training accuracy with the use of the FA layer in MUTAG indicates the presence of over-squashing. However, the AllFA approach also leads to a similar improvement, which suggests that graph topology is not very important in this benchmark. This is also likely why both +FA and AllFA outperform expander graph propagation, where the rewiring is much sparser. In both datasets, the virtual node rewiring was most effective in improving the training accuracy, and hence in mitigating over-squashing. Since MUTAG doesn't heavily rely on structural information, this may be a result of the additional expressivity offered by the MLP that computes the virtual node embedding. In the ENZYMES task, the training accuracy decreases with the introduction of a FA layer, which may suggest that the model is not in fact suffering from over-squashing. Given that the training accuracy for the ordinary GAT is quite low, it is possible that the base model is not maximally tuned and a more extensive hyperparameter search may be necessary. The decrease in training accuracy for AllFA also indicates that graph topology is important to the classification. The sparsity of the propagation graphs in EGP may therefore be the basis for the accuracy increase over the base GAT, +FA, and AllFA models. Furthermore, the virtual node approach sees a drastic improvement in performance which likely results from each layer both preserving structural information and utilizing global information.

Discussion

The superior performance of the virtual node approach indicates that for tasks that rely on the graph structure, GATs benefit most from rewirings that preserve topological information as much as possible. However, this claim can only be validated through a much more rigorous analysis on a wider pool of datasets. One major limitation of this study is the lack of a graph classification dataset that both suffers from over-squashing and relies heavily on the relational structure; ENZYMES appears to be missing the former, and MUTAG is missing the latter. The study would be better supplemented by a synthetic dataset where we could directly control both factors. One such dataset is the TreeNeighborsMatch benchmark designed by Alon and Yahav [1]. A disadvantage of using this benchmark is the size of the dataset required to observe the phenomenon of over-squashing: the training accuracy of GATs only starts decreasing at a tree depth of 5 (which calls for a model with 6 layers). The dataset samples 32,000 trees at this depth for training, leading to very long training times and high computational costs. The observation that it takes more layers for GAT performance to suffer from over-squashing as compared to other GNN architectures [1] is indicative of a more fundamental issue with the motivation behind this study. Since GATs are inherently more robust to information bottlenecks, a benchmarking of over-squashing solutions is better motivated for architectures like GCN or GIN.

Future Work

Given the positive results of rewiring methods that preserve structural information, such as the virtual node approach, it would be interesting to test whether similar results can be seen from over-squashing solutions that preserve topological statistics despite completely changing the prorogation graph. Candidate methods include Stochastic Discrete Ricci Flow [8] or DropEdge [9]. Furthermore, a future study could try implementing attention-based aggregation of node representations at the virtual node, which may lead an improvement in model performance over mean pooling where all nodes have the same weight. Finally, it would be interesting to compare GATs with rewired shortcuts to Graph Transformers [10], which are assumed to be fundamentally more powerful since they are not limited by local information.

References

[1] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," in *International Conference on Learning Representations*, 2021.

[2] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," *CoRR*, 2017.

[3] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *arXiv preprint arXiv:2005.00687*, 2020.

[4] A. Deac, M. Lackenby, and P. Veličković, "Expander graph propagation," in *The First Learning on Graphs Conference*, 2022.

[5] J. Grout, I. H. Hanson, S. Johnson, A. Kramer, A. Novoseltsev, and W. Stein, "Sage math cell."

[6] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "Tudataset: A collection of benchmark datasets for learning with graphs," in *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020.

[7] K. Karhadkar, P. K. Banerjee, and G. Montúfar, "Fosr: First-order spectral rewiring for addressing oversquashing in gnns," 2022.

[8] J. Topping, F. D. Giovanni, B. P. Chamberlain, X. Dong, and M. M. Bronstein, "Understanding over-squashing and bottlenecks on graphs via curvature," in *International Conference on Learning Representations*, 2022.

[9] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *International Conference on Learning Representations*, 2020.

[10] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.