Adaptive Residual Connections and Graph Attention for Connectome Graphs

Benjamin Goldstein
Department of Computer Science
Yale University
New Haven, CT 06520
b.goldstein@yale.edu

Burton Lyng-Olsen
Department of Computer Science
Yale University
New Haven, CT 06520
burton.lyng-olsen@yale.edu

Abstract

Predictive machine learning models have demonstrated robustness in various fields of medical diagnostics. These models, however, have been largely focused on the image, and to a lesser degree text, modalities. Recent developments in the field of graph-based methods, which provide a more flexible computational structure than traditional image and text processing approaches, now represent the state of the art for a wide variety of machine learning tasks. In this paper, we analyze the performance of two graph neural network methods on a novel graph based fMRI dataset: NeuroGraph. The NeuroGraph dataset is the result of a processing pipeline applied to fMRI brain scans of various subjects under various conditions, and it has been shown that deep graph-based methods perform well at classification on this data. We focus on classification tasks, discerning the subject's activity at the time of the scan and gender. The two techniques examined on this task are adaptive residual connections, which are shown to be robust at classifying abnormal nodes, and the graph attention model, which has been shown to demonstrate structural awareness in its predictions.

1 Introduction

Graph neural networks (GNNs) have demonstrated strong results in a variety of domains: including bioinformatics [1], recommendation systems[2], and knowledge graph completion [3]. These tasks intuitively lend themselves in many cases to graph based methods: molecules and proteins are conveniently represented as graphs in biomedical datasets, and consumer/product and knowledge graphs are readily available as well. Given a well designed graph representation of the human brain, it seems that these graph methods would generalize well to classification tasks on that domain.

1.1 NeuroGraph background

Precise diagnostic analysis of fMRI brain scans is of huge importance to the medical community, as this could facilitate early diagnosis of dangerous pathologies and thus earlier intervention with lifesaving treatment. It has been shown that a data pipeline consisting of (1) a partitioning of the brain into regions of interest (ROIs) followed by (2) the usage of correlation matrices between such ROIs to synthesize a graph yields a graph based dataset from fMRI images that is extremely well suited to graph based deep learning. The dataset generated by this pipeline, NeuroGraph, then allowed graph based methods to outperform more traditional architectures including the multilayer perceptron, convolutional neural network, and random forest model in graph classification tasks. The architecture used in that analysis was a generic three-layer graph neural network with residual connections that was then tested using a variety of convolutional layers [4].

Preprint. Under review.

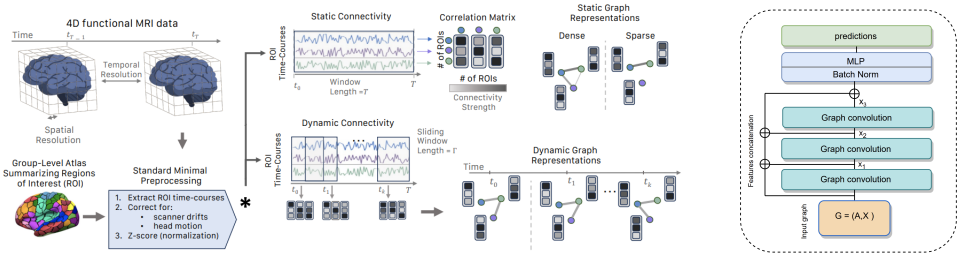

Figure 1: NeuroGraph dataset generation pipeline (left) and static graph neural model (right) [4]

The NeuroGraph analysis presented above provides a substantial proof of concept that their image to graph pipeline is well suited for graph based deep learning. However, the proposed architecture is designed to easily survey a wide set of graph convolution methods rather than pin down a performant configuration. In this paper we test a pair of methods with promise based on observations about the dataset.

1.2 Motivations

A graph neural network with adaptive residual connections was shown to demonstrate superior ability to classify nodes with abnormal properties [5]. Given the design of this dataset, it seems that a small subset of the regions of interest will be activated for a given task, while the rest are largely uninvolved and noisy. This is evident in the fact that a large majority of the nodes in the dataset are unconnected. It is therefore promising to attempt to extend the adaptive residual architecture, which appears to perform on feature aggregation for graphs with abnormal nodes, to a graph classification task. This can be done using the same post processing layers as the architecture proposed in NeuroGraph to generate a graph classification from the generated node embeddings: batch normalization followed by a concatenation based MLP.

Another methodology that has shown promise in navigating noisy datasets is the Graph Attention Model (GAM) [6]. While the entire brain network is present in a given scan, as referenced earlier, each task likely pertains primarily to a particular section of the brain and correspondingly to a subset of the graph. Thus, the GAM model is of interest as it has demonstrated to be competitive in graph identification tasks, even when limited to only sub-graphs of the entire sample. To this end, the GAM is able not only to be observant to structural properties of a graph, but selective in what sub-structures are most pertinent.

1.3 Takeaways

Overall, these architectures performed well in relation to many of the architectures posed by the dataset's creators. The adaptive residual connections outperformed all of the graph methods except for GNN*, or non-adaptive residual connections. This demonstrated the importance of some sort of residual connection structure in performing graph classification tasks on this sort of dataset. The GAM based model similarly preformed well relative to the baselines, as well as slightly exceeding the other attention based model that may be more sensitive to sparsity.

2 Related work

Much investigation has been conducted in the realm of graph representations of human biology, often called the human "connectome" [7]. One particular dataset, NeuroGraph, has shown high performance on both static and dynamic graph classification tasks when deep graph methods are used [4]. The methods posed by the NeuroGraph authors have a mixed performance record relative to non-graph methods including the MLP, which is particularly performant on these tasks. One particular architecture, the residual network, performs strongly.

A framework for adaptive message passing, alternatively interpreted as adaptive residual connections, has been posed for improved node classification of abnormal nodes [5]. This architecture builds on previous work in the realm of deepening graph neural networks, which often suffer from oversmooth-

ing [8, 9]. The adaptive residuals scheme, however, poses an explicit update formula built on top of vanilla message passing, which is not explicitly modularized for use on top of other models the way typical residuals are. The paper also focuses on node classification, while we will use adaptive residuals to generate a node embedding for downstream graph classification.

Another increasingly popular framework for graph classification tasks are attention based networks. Inspired by its success in other deep learning tasks [10], researchers have successfully applied the attention based learning model to graph problems. One such application is by having nodes attend to neighbors as well as self attend, forming more expressive understanding of graph structure without it being explicitly fed. This approach has shown state-of-the-art results in node classification tasks [11].

A model that features attention based techniques for a graph classification problem is the Graph Attention Model [6]. The GAM uses attention to build upon a LSTM model. The model features a modified version of a random walk explorer that prioritizes moving towards neighbors of higher "rank", a metric based on the node type that is dynamic as the model explores. This rank network is the analogous to attention in that it learns throughout training which types of nodes are most important to understanding the class of the graph. As exploration continues, these types of nodes will be prioritized in future traversals, leading them to dominate the state passed on through the LSTM. This model was used on a binary identification task on molecules with the nodes, individual atoms, being categorized by their element. The model showed to out preform other baselines when given partial graphs, and equal the accuracy of the baselines even when the baselines were given the full graph and the GAM was not.

3 Methods

As mentioned in the introduction, this paper is focused on applying two new methods to the task of graph classification on connectome graphs from fMRI data. The first test is an application of adaptive residual connections to the graph classification task, a domain that has not yet been examined. The second builds upon a stochastic graph attention model that was not included in the initial NeuroGraph analysis.

3.1 Adaptive residual connections for graph classification [Ben]

Given the prevalence of noise and outlier ROIs in connectome data, a graph classification approach that relies on node representation learning must be able to handle these nodes. AirGNN, the adaptive residual connection approach detailed earlier, provides an implementation of a graph neural network with learned parameters for residual connections. Because it is specific to node classification, though, it uses a sequence of layers that are not applicable to general node representation learning: the model begins with an MLP to reduce node feature dimensionality down to `num_classes` followed by adaptive message passing to generate a logit vector for each node. Since we'd like to yield a higher dimensional embedding for each node than simply the number of classes, we forego the "base model" MLP and use an alternative approach.

The NeuroGraph residual architecture as shown in figure 1 applies message passing on the raw node feature vectors before using batch normalization and a concatenation based MLP to generate the graph classification. The explicit residual architecture concatenates the embeddings for each layer of the GNN. The first layer is the raw feature vectors which represent a symmetric matrix given the fact that correlation matrices are used as node features. The upper triangular elements of this matrix (to avoid redundancy) are then flattened. The later layers of the GNN are aggregated using mean aggregation and concatenated onto the flattened upper triangular raw feature matrix. That is then passed into an MLP for final analysis.

Since the result of the adaptive architecture is effectively an aggregated series of embeddings at different layers, we can base our approach off of the NeuroGraph architecture. Mean pooling would defeat the purpose of the adaptive scheme which is designed to allow abnormal nodes to gain distinct representations. We decide to treat the output of adaptive message passing as NeuroGraph treats the input features, i.e. feed it flattened into an MLP. The overall proposed architecture using adaptive message passing (AMP) is shown below and contrasted with the AirGNN architecture:

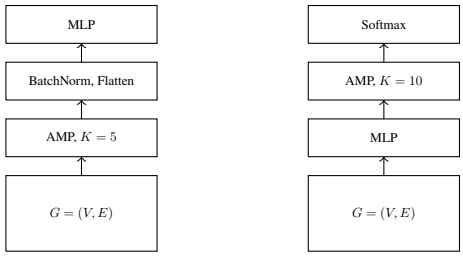

Figure 2: Proposed graph classification architecture (left), AirGNN node classification (right)

It should be noted that the proposed architecture, which we'll call AdaptIve Residual Graph Classifier (AirGC), is sensitive to node permutation. This doesn't seem to be an issue for this task, though, as this dataset is curated with a well defined node ordering based on biological priors and the NeuroGraph approach uses a permutation sensitive input to its MLP as well. The K-hop neighborhood was decreased from 10 to 5 to reflect the decreased desired receptive field of the NeuroGraph dataset: their networks only use three-layer GNNs[4]. Generally, this architecture would be applicable to graph classification tasks on datasets with non-arbitrary node orderings. Further research on permutation invariant aggregation methods on this task may prove fruitful in further generalizing the method.

3.2 GAM inspired approach to connectome classification [Burton]

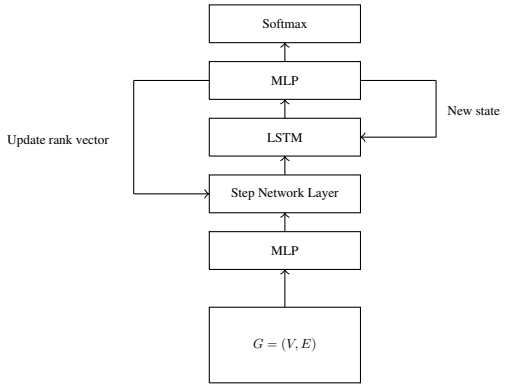

Figure 3: GAM based graph classification architecture

The GAM inspired method uses attention mechanisms built into a recurrent network to complete the identification test. As a LSTM based network, one forward pass consists of moving from one node to the next akin to random walk. This probabilistic selection occurs at the step network layer. The selection process is weighted based on the relationship between the neighboring nodes features and the rank vector. The original construction of GAM [6] is implemented over nodes with categories, with each node having a 1-hot embedding. This rank vector, which therefore ranks the "importance" of categories, is not applicable to the NeuroGraph set as each node has an exhaustive number of continuous features. Our model adds a MLP layer beneath the step network layer. Here we build on the base model GAM. as this expands the use of the rank vector system from strictly 1-hot encoded nodes to take advantage of nodes with robust features.

Once a step is determined, the step embedding as well as the history vector from the last state are used in a LSTM layer to produce a new history vector. The LSTM layer output goes through another MLP layer to produce a prediction on the graph layer as well as a new rank vector. Through training, this new rank vector gets a better sense of the important vectors to traverse in future steps. Due to the nature of the LSTM layer, this is also dynamic with respect to the exploration process.

4 Experiments

This section is dedicated to precisely explaining the experiments run to evaluate the methods proposed in section 3. These evaluation metrics largely follow the skeleton provided by NeuroGraph, however they apply the newly proposed architectures. Results are benchmarked via the results of graph and non-graph methods on the dataset detailed in the NeuroGraph paper [4].

4.1 Datasets

The datasets used in the below experiments are NeuroGraph's HCPActivity and HCPGender. These are sets of 400 node graphs, each with 400 dimensional features derived from correlation matrices during an image processing pipeline. Each graph in either dataset has a class: the activity or gender of the subject respectively. The data consists of largely unconnected nodes, though mean node degree of connected nodes is roughly 20. This confirms the priors detailed earlier: structural awareness and distinct treatment of abnormal nodes should have a positive effect on performance. Results of preliminary data analysis is shown below:

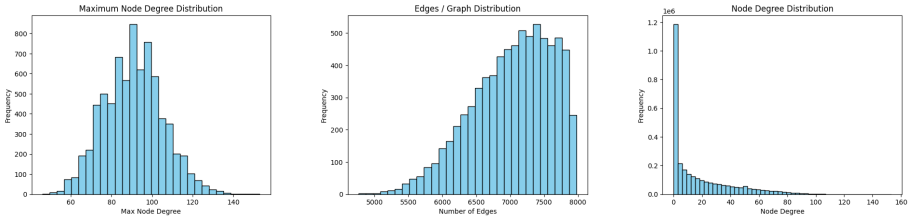

Figure 4: Preliminary data analysis on HCP datasets

4.2 Adaptive residual connection experiments [Ben]

The model was trained using the optimal hyperparameters from the respective papers: adaptive message passing paramter λ was set to 0.6 and the learning rate to 0.1 in accordance with AirGNN, while the MLP layer architecture and batch size = 16 were set in accordance with NeuroGraph. The MLP, accordingly, layer consists of four linear layers separated by ReLU activation and batch normalization. The data is preprocessed with feature normalization and loaded in as sparse tensor objects as the forward pass for AMP uses sparse tensor operations. The adjacency matrix is normalized as is standard practice using `PyG`'s `gcn_norm` [12]. Training was run for 100 epochs over five separate runs with a fixed seed per run (123 for run 1, then 124, and so on). We observed early convergence during the training process, with loss and accuracy seeming to bottom out within the first 20 epochs. The training dynamics for a run of the HCP Activity dataset are shown below:

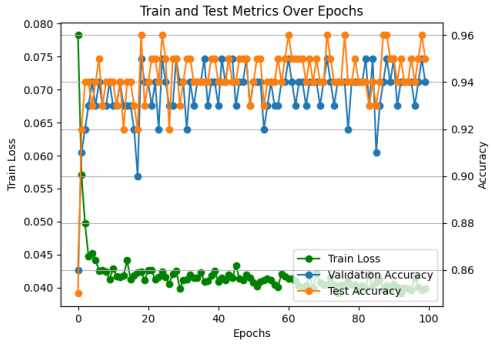

Figure 5: Train and test accuracy by epoch on HCPActivity

The results of the model runs are summarized in the table below. The table shows benchmarks as conducted by the NeuroGraph group and comparison with our proposed architecture, AirGC:

Table 1: AirGC performance results

Model	Dataset	Accuracy μ	Accuracy σ^2
AirGC	HCP-Activity	95.37	0.59
AirGC	HCP-Gender	84.54	1.08

While the benchmark paper did not report variance of accuracy, we can compare the means with the results from NeuroGraph:

Table 2: Benchmark results of various models from NeuroGraph [4]

Dataset	NN	CNN	RF	k-GNN	GCN	SAGE	UniMP	ResGCN	GIN	Cheb	GAT	SGC	General	GNN*
HCP-Activity	97.78	95.88	88.98	93.23	94.21	94.78	94.72	94.61	89.79	94.45	95.2	94.17	93.62	98.20
HCP-Gender	86.67	76.39	69.9	82.13	75.46	77.69	76.67	78.33	75.56	59.07	76.20	76.48	78.89	89.07

We can see that the AirGC model outperforms many of the graph and non-graph methods included in the benchmarking in terms of mean acccuracy. We can see that on both datasets the AirGC architecture outperforms all graph methods except the GNN*. It also outperforms the CNN on HCPGender, though the vanilla NN is extremely performant on both tasks. Since the NN is extremely performant and the GNN* uses a concatenated feature vector as the input to the vast majority of its MLP's parameters, it seems that directly learning on the correlations used as node features performs well on the task. When message passing is used, information is lost, but AirGC does extremely well in preserving that information. The performance within graph methods could be due to smoothing effects: the smoothest representations come from models without any sort of residuals, followed by adaptive, then finally followed by the full concatenation based residual connection scheme. The result of this analysis perhaps suggests that for a concatenation based classifier on top of a node representation, smoother embeddings lose expressive power needed for accurate classification.

4.3 Graph attention model experiments [Burton]

The GAM inspired model was constructed with the default parameters as suggested by NeuroGraph, with a batch size of 16. For each dataset, we completed 5 individual runs over 100 epochs. The loss and accuracy statistics across training are show in figures 6 and 7 for the activity and gender task respectively. A larger number of epochs was considered, but as observed in the figures, accuracy converged before 100 epochs, so the training length was deemed sufficient.

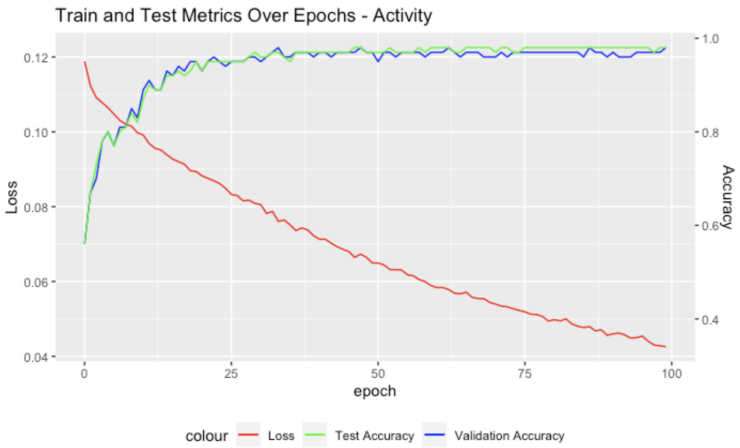

Figure 6: Train and test accuracy by epoch on HCPActivity

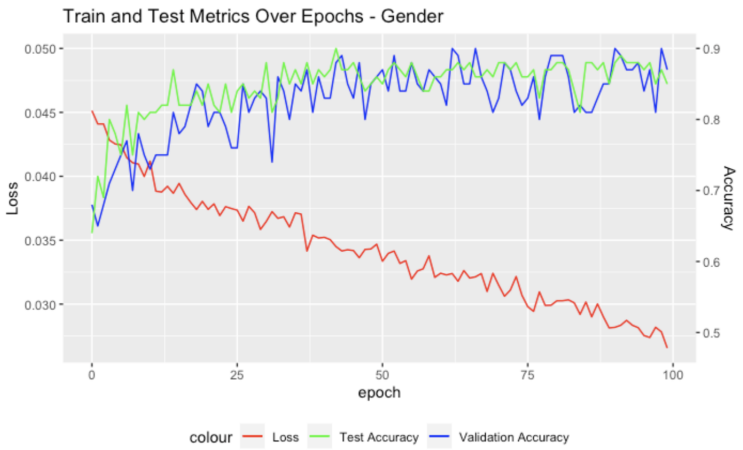

Figure 7: Train and test accuracy by epoch on HCPGender

Summarized below are the aggregate results of the 5 trials of the model on the HCP-Activity and HCP-Gender task:

Table 3: GAM performance results

Model	Dataset	Accuracy μ	Accuracy σ^2
GAM	HCP-Activity	97.24	0.68
GAM	HCP-Gender	84.91	3.51

Comparing to table 2 again, we see the GAM model preforms quite well compared to the NeuroGraph benchmarks. In HCP-Activity, GAM is only surpassed by NN and GNN*. GAM does even better relative to the baselines in the HCP-Gender task where its 84.54 mean accuracy, though still less than NN and GNN*, differentiates itself from the other general baseline models. Since the original model was trained on a binary identification task, it is possible that this is a consequence of the recommended parameters provided. Overall the results certainly suggest that a GAM model is appropriate for this task and it actually slightly out preforms GAT, the other attention based model. This is likely due to the sparsity of the graph as GAT's attention is limited to k-hop neighbors, while the random walk approach of GAM is not as sensitive to node degrees.

5 Conclusion

The NeuroGraph dataset and the performance of associated graph methods are a prime example of how synthesized graph data from other modalities can improve performance on machine learning tasks. We propose a novel adaptive message passing GNN for graph classification, AirGC, that outperforms several of the benchmark methods demonstrated by NeuroGraph. This proposed architecture does not use permutation invariant aggregation, though, so it is only narrowly applicable to datasets like this one where node order is consistent. The adaptive residual connections classifier failed to achieve a better node representation for graph classification than the standard residual architecture, but outperformed other architectures without residual connections. We also propose a random walk based graph attention model within the residual architecture of NeuroGraph. We show that an attention based approach also performs relatively well on the classification task, and that this modified version of GAM actually slightly out preforms a more traditional k-hop neighborhood attention model.

A significant body of further work remains in this domain. Alternative connectome representations and data pipelines could improve performance of graph methods on tasks not suited to traditional ML. This paper focuses on static graph representations, but dynamic graphs provide increased information for potentially improved performance—alternative methods should absolutely be considered in that domain as well. New research is going on every day in the field of graph-based deep learning, and such methods should absolutely be tested across the vast field of bioinformatics. This paper provides a roadmap for the design and implementation of new models based on newly created methods in the human connectome domain.

Our repository can be found here: https://github.com/bengoldstein19/NeuroGraph/tree/main

References

[1] Xiao-Meng Zhang, Li Liang, Lin Liu, and Ming-Jing Tang. Graph neural networks and their current applications in bioinformatics. *Frontiers in Genetics*, 12, 2021. ISSN 1664-8021. doi: 10.3389/fgene.2021.690049. URL https://www.frontiersin.org/articles/10.3389/fgene.2021.690049.

[2] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, and Yong Li. A survey of graph neural networks for recommender systems: Challenges, methods, and directions, 2023.

[3] Ivana Balazevic, Carl Allen, and Timothy Hospedales. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019. doi: 10.18653/v1/d19-1522. URL http://dx.doi.org/10.18653/v1/D19-1522.

[4] Anwar Said, Roza G. Bayrak, Tyler Derr, Mudassir Shabbir, Daniel Moyer, Catie Chang, and Xenofon Koutsoukos. Neurograph: Benchmarks for graph machine learning in brain connectomics, 2023.

[5] Xiaorui Liu, Jiayuan Ding, Wei Jin, Han Xu, Yao Ma, Zitao Liu, and Jiliang Tang. Graph neural networks with adaptive residual. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 9720–9733. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/50abc3e730e36b387ca8e02c26dc0a22-Paper.pdf.

[6] Xiangnan Kong John Boza Lee, Ryan Rossi. *Graph Classification using Structural Attention*. KDD, 2018.

[7] Olaf Sporns, Giulio Tononi, and Rolf Kötter. The human connectome: A structural description of the human brain. *PLOS Computational Biology*, 1(4):null, 09 2005. doi: 10.1371/journal.pcbi.0010042. URL https://doi.org/10.1371/journal.pcbi.0010042.

[8] Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20. ACM, August 2020. doi: 10.1145/3394486.3403076. URL http://dx.doi.org/10.1145/3394486.3403076.

[9] T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks, 2023.

[10] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Lukasz Kaiser Illia Polosukhin Ashish Vaswani, Noam Shazeer. Attention is all you need. 2017.

[11] Arantxa Casanova Adriana Romero Pietro Liò Yoshua Bengio Petar Veličković, Guillem Cucurull. Graph attention networks. *ICLR*, 2018.

[12] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.