A Re-implementation of the Hierarchical Graph Pooling Model Variants

Ali Naqvi
McMaster University
Milton, Ontario
naqvia18@mcmaster.ca

ABSTRACT

In this short essay, we re-implement and compare the results from the paper, "Hierarchical Graph Model with Structured Learning" [8]. We utilized six distinct datasets, similar to the original paper, with variations of the proposed model for comparison. This includes four modifications made to the proposed model. The margins of accuracy in this comparison indicate issues with newer versions of the utilized libraries, specifically concerning the structured learning layer.

PVLDB Reference Format:
Ali Naqvi. A Re-implementation of the Hierarchical Graph Pooling Model Variants.

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/ali1naqvi/HGP.

1 INTRODUCTION

In the realm of machine learning, exploring non-Euclidean data with Graph Neural Networks (GNNs) signifies a substantial advance beyond traditional deep learning techniques. GNNs examine this diverse data in the form of graphs, evident across various fields such as social network analysis, molecular chemistry, and bioinformatics. To manage the complexity and variability of graph data, GNNs employ convolution and pooling operations, fundamental to conventional neural networks. Pooling operations, in particular, are crucial for summarizing graph-level features from individual node and edge attributes.

This short essay focuses on the re-implementation of a novel graph pooling operator, HGP-SL, originally proposed to overcome the inherent limitations of hierarchical representation learning in GNNs. The re-implementation not only aims to validate the original findings but also to update the implementation with newer library versions. This comparative analysis is aimed at assessing whether the remarkable results previously achieved maintain their high accuracy with current technology.

2 EXPERIMENTAL SETUP

2.1 Datasets

For the testing comparison, we used the six datasets presented in the paper. Table 1 displays the key features of each dataset. The first dataset, **ENZYMES** [1], contains protein tertiary structures, and each enzyme belongs to one of the 6 top-level enzyme classes. **PROTEINS** and **D&D** [2] are two protein graph datasets. The nodes represent the amino acids and the connection between two nodes is established with an edge if they are less than 6 Angstroms apart. **NCI1** and **NCI109** [6] are two biological datasets screened for activity against non-small cell lung and ovarian cancer cell lines. The graphs are chemical compounds with nodes and edges representing atoms and chemical bonds, respectively. Finally, **Mutagenicity** [3], is a chemical compound dataset of drugs and have two categorizations of classes: mutagen and non-mutagen.

Datasets	$\#\|\mathcal{G}\|$	$\#\|\mathcal{V}\|$	Avg.$\|\mathcal{V}\|$	Avg.$\|\mathcal{E}\|$	$\#\|\mathcal{C}\|$
ENZYMES	600	19,580	32.63	62.14	6
PROTEINS	1,113	43,471	39.06	72.82	2
D&D	1,178	334,925	284.32	715.66	2
NCI1	4,110	122,747	29.87	32.30	2
NCI109	4,127	122,494	29.68	32.13	2
Mutagenicity	4,337	131,488	30.32	30.77	2

Table 1: Statistics of the datasets used.

2.2 Baselines

The baseline models we used to compare are the inclusion and variants of the HGP-SL model. These models are:

1. HGP-SL_{NSL} (No Structure Learning), where we remove the structure learning layer.
2. HGP-SL_{HOP}, where we remove the structure learning layer and connect the nodes within its h-hops
3. HGP-SL_{DEN} where we utilize the structure learning layer to learn a dense graph structure using a softmax function [5].
4. HGP-SL, the proposed model which utilizes sparse-max function [4] to learn a sparse graph structure.

2.3 Evaluation Metric

In our re-implementation, we adopt accuracy as the primary metric to assess the performance of our graph classification model. Accuracy is defined as the ratio of correctly predicted graph labels to the total number of graphs evaluated, expressed as a percentage. Mathematically, it can be represented as:

$$\text{Accuracy} = \left(\frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \right) \times 100\%$$

This metric provides and indication how often the model predicts the correct label for a given input graph.

Categories	Models	*ENZYMES*	*PROTEINS*	*D&D*	*NCI*1	*NCI*109	*Mutagenicity*
Paper Results	HGP-SL$_{NSL}$	60.18 ± 2.43	81.51 ± 1.69	77.24 ± 1.09	76.33 ± 1.43	76.32 ± 1.22	79.42 ± 0.58
	HGP-SL$_{HOP}$	62.16 ± 2.11	83.03 ± 1.74	78.42 ± 1.37	77.72 ± 1.54	78.78 ± 1.09	79.88 ± 1.09
	HGP-SL$_{DEN}$	63.51 ± 2.64	83.12 ± 0.84	78.11 ± 1.35	77.42 ± 1.23	78.76 ± 0.61	81.07 ± 1.02
	HGP-SL	**68.79 ± 2.11**	**84.91 ± 1.62**	**80.96 ± 1.26**	**78.45 ± 0.77**	**80.67 ± 1.16**	**82.15 ± 0.58**
Re-implementation	HGP-SL$_{NSL}$	44.44±4.89	75.00±0.72	77.56±1.12	75.09±1.32	75.05±1.42	78.04±1.61
	HGP-SL$_{HOP}$	42.50±3.94	76.49±1.49	79.20±1.06	75.75±1.52	77.10±1.05	78.38±0.93
	HGP-SL$_{DEN}$	32.22±1.85	76.12±1.12	74.54±2.52	76.56±1.65	71.91±0.88	77.67±1.59
	HGP-SL	29.44±1.91	78.84±1.16	75.66±1.30	77.31±1.60	73.75±1.41	78.13±1.34

Table 2: Graph classification in terms of accuracy with standard deviation (in percentage). We use bold to indicate winnings from Paper Results and underline for winnings from Re-implementation.

2.4 Experiment and Parameter Settings

To ensure a fair comparison, we employ a similar evaluation to the paper where a 10-fold cross-validation approach is done. Specifically, the dataset is divided randomly into three parts: 80% for the training set, 10% for the validation set, and finally, 10% for the testing of the model's generalization to unseen data. The hyperparameters used for each dataset were tuned accordingly to the original paper and can be seen in Table 3.

Datasets	*lr*	*wd*	*bs*	*pr*	*do*	*nl*
ENZYMES	0.001	0.001	128	0.8	0.0	2
PROTEINS	0.001	0.001	512	0.5	0.0	3
D&D	0.0001	0.001	64	0.3	0.5	2
NCI1	0.001	0.001	512	0.8	0.0	3
NCI109	0.001	0.001	512	0.8	0.0	3
Mutagenicity	0.001	0.001	512	0.8	0.0	3

Table 3: Best Hyper-Parameters for each Dataset. Abbreviations: *lr* **- Learning Rate,** *wd* **- Weight Decay,** *bs* **- Batch Size,** *pr* **- Pool Ratio,** *do* **- Dropout,** *nl* **- Net Layers**

3 EXPERIMENTS AND ANALYSIS

For the final reported accuracy, we utilized the mean accuracy obtained over 10 experimental runs, as shown in Table 2, including the standard deviation. The original author did not report the computational cost for each variant; however, we observed a significantly longer time to compute the final prediction accuracy for variants that use the structured learning layer compared to those that do not.

From Table 2, gaps between the predictions from the literature and its re-implementation are evident. Variants employing the structured learning layer, specifically HGP-SL$_{DEN}$ and HGP-SL, exhibit significantly lower prediction accuracy on the Enzymes dataset and are generally outperformed in the other datasets. This decrease in accuracy for these specific variants can be attributed to the structured learning layer; newer versions of Torch-Sparse lack autograd support because obtaining gradients for sparse-sparse matrix multiplication is challenging. Similar to this short essay, recent attempts to replicate the paper's results have been unsuccessful. However, based on the implementation of HGP-SL variants, we can conclude that the structured learning layer is the root cause of the proposed model's underperformance. In the future, we aim to extend the original paper's limited analysis of the computational expense of the structured learning layer. Additionally, we plan to explore expanding the architecture to incorporate more recent innovations, such as including the Graph Isomorphism Network in their ablation study [7].

4 CONCLUSIONS

With the re-implementation of the Hierarchical Graph Pooling with Structured Learning model (HGP-SL) and its variants, we identified discrepancies in the originally reported prediction accuracy. These discrepancies are likely due to the deprecation of features previously offered by the versions of the libraries used, such as the absence of autograd support in Torch-Sparse. Given that the paper is over four years old, an updated approach incorporating these considerations, along with recent innovations, would be appropriate.

REFERENCES

[1] KM Borgwardt, underlineCS, Stefan Schönauer, SVN Vishwanathan, Alexander Smola, and Peer Kröger. 2005. Protein function prediction via graph kernels. *Bioinformatics* 21 Suppl 1 (01 2005), i47–56.

[2] Paul D. Dobson and Andrew J. Doig. 2003. Distinguishing Enzyme Structures from Non-enzymes Without Alignments. *Journal of Molecular Biology* 330, 4 (2003), 771–783. https://doi.org/10.1016/S0022-2836(03)00628-4

[3] Jeroen Kazius, Ross McGuire, and Roberta Bursi. 2005. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of medicinal chemistry* 48 1 (2005), 312–20. https://api.semanticscholar.org/CorpusID:21873061

[4] André F. T. Martins and Ramón Fernandez Astudillo. 2016. From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. arXiv:1602.02068 [cs.CL]

[5] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. 2018. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. arXiv:1811.03378 [cs.LG]

[6] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. 2011. Weisfeiler-Lehman Graph Kernels. *J. Mach. Learn. Res.* 12 (2011), 2539–2561. https://api.semanticscholar.org/CorpusID:1797579

[7] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks? *ArXiv* abs/1810.00826 (2018). https://api.semanticscholar.org/CorpusID:52895589

[8] Zhen Zhang, Jiajun Bu, Martin Ester, Jianfeng Zhang, Chengwei Yao, Zhi Yu, and Can Wang. 2019. Hierarchical Graph Pooling with Structure Learning. *arXiv preprint arXiv:1911.05954* (2019).