Effectiveness of Graph Neural Network Operators in Virulence Classification of Mouse-Influenza A Protein Interactions

Liu Jiale[a], **Zhu Luyao**[a], **Kwoh Chee Keong**[a], **Shamima Rashid**[a]

[a] *School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798*
jliu051@e.ntu.edu.sg , luyao001@e.ntu.edu.sg ,asckkwoh@ntu.edu.sg , sham0012@e.ntu.edu.sg

1. Introduction

This study develops a graph neural network (GNN) to predict the virulence of Influenza A Virus (IAV) infections by incorporating protein-protein interactions between the virus and its mouse host. Since direct human studies are unethical, mouse models are used to infer disease properties [1].
GNN have previously been used in protein applications. SolPredictor [2] predicts the solubility of molecules using a graph representation with nodes representing atoms and edges representing connections between atoms. The Residue-Based Graph Attention and Convolutional Network (RGN) uses both graph convolution and graph attention networks to predict interaction sites of a protein structure [3].

Here, a GNN architecture is proposed, that employs four different graph operators to classify mouse-IAV protein-protein interactions using interaction scores predicted with the Deep Sequence Contact Residue Interaction Prediction Transfer (D-SCRIPT) software [4]. The goal is to better understand the role of these interactions in viral severity and disease progression, aiding in the development of effective treatments and vaccines. The code and datasets developed here are available at `https://github.com/liujiale-study/MSAIProj_HP-PPI/tree/main`.

2. Methods

The proteins in the dataset are primarily characterized by embeddings from ProstT5 [5], while the interactions are predicted using D-SCRIPT. ProstT5 is a model for the creation and translation of sequence and structural embeddings from amino acid sequences and 3Di tokens, respectively.

2.1 Dataset

Previous work had curated infection records with lethal dose values for several mice strains and assigned them into three-class virulence labels [6, 7]. Here, the dataset is further processed and expanded to include UniProt IDs from the UniProt [8] database for strains BALB/cJ and DBA/2J.
Additionally, the protein-protein interaction data for C57BL/6, BALB/cJ and DBA/2J were computed using D-SCRIPT, which provides predicted probabilities of interactions. A protein pair was considered as interacting if its D-SCRIPT score was higher than a threshold of 0.01 (1%). Next, the virulence labels were assigned to each interacting edge, according to its information in the virulence record.

The final dataset consists of a total of (i) 53,266 mouse proteins, (ii) 659 IAV proteins and (iii) 365,926 mouse-IAV interactions.

2.2 GNN Architecture and Evaluation

The data was converted into a bipartite graph format with heterogenous node representation for the mouse and viral protein nodes, as shown in Figure 1.

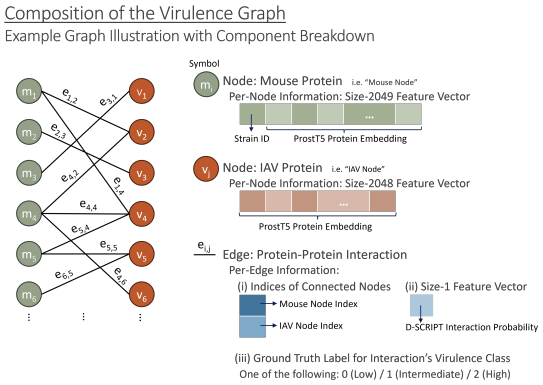

Fig. 1: Virulence Graph Composition

A node corresponds to a protein while an edge represents an interaction between 2 proteins. Only interactions between mouse and IAV nodes are considered. The node embeddings (of length 2048) were generated using ProstT5[5].The D-SCRIPT predicted interaction probability was used as an edge attribute.

Figure 2 shows the proposed GNN architecture.
It consists of a linear layer that transforms the input to fixed-size embeddings, followed by a 'block unit' consisting of a (i) GNN operator layer, (ii) batch normalization layer, (iii) ReLU layer and (iv) dropout layer. The layers (ii)-(iv) have the effect of regularizing the network and stabilizing its training. This 'block unit' structure is repeated (Figure2(c)). The intermediate representations of mouse and viral proteins produced by the block unit are further concatenated to form the supervision edge representation (Figure2(d)). The final linear layer transforms the supervision edge representation into 3 predicted classes corresponding to each virulence category *Low/Intermediate/High*, encoded as $0, 1, 2$ respectively.

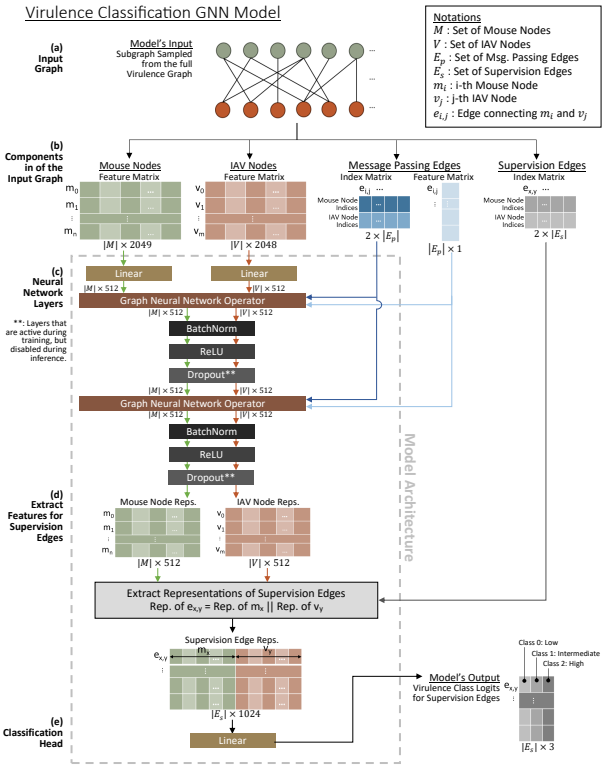

Fig. 2: GNN Architecture

The GNN operator layer consists of one of four operators – Residual Gated Graph Convolutional Network (RGGCN) [9], Graph Attention Network (GAT) [10], Graph Transformer (GTR) [11] and Graph Isomorphism Network with Edge Features (GINE) [12]. The RGGCN operator is given as:

$$x_i' = W_1 x_i + \Big(\sum_{j \in N(i)} \eta_{i,j} \odot W_2 x_j \Big) + b \qquad (1)$$

where the edge gate $\eta_{i,j}$ is given by:

$$\eta_{i,j} = \sigma(W_3(x_i || \varepsilon_{i,j}) + W_4(x_j || \varepsilon_{i,j})) \qquad (2)$$

The model can switch between one of the four aforementioned operators and only one operator may be in use in the proposed GNN at any given time. The edges were split into train, validation and test sets at a ratio of 70:20:10. Section B gives more training settings.

3. Results

The proposed GNN was evaluated on each of the four graph operators – RGGCN, GAT, GTR and GINE, using the measures of F1, overall accuracy and Matthew's Correlation Coefficient (MCC).

The results in Table 1 indicate that the RGGCN and GTR operators provide the best performance. Although these two operators have the highest scores in different metrics, their overall performance across all metrics is < 0.01. The RGGCN has an overall accuracy of 0.887516, closely followed by the GTR at 0.886915. However, the GTR has a better MCC score of 0.831264, 0.001354 higher than RGGCN. Overall, the RGGCN layer provides the best performance with greater efficiency due to significantly shorter training times than needed by the GTR. This may be attributed to the higher complexity and number of weights in the GTR operator. The GTR operator was also found to have more fluctuations in its training loss compared to the RGGCN, indicating that further architectural regularization might be needed.

Table 1: GNN Performance using Operators

Operator	RGGCN	GAT	GTR	GINE
F1: *Low*	0.903292	0.672532	**0.907952**	0.886930
F1: *Intermediate*	**0.880099**	0.588587	0.871150	0.852913
F1 *High*	0.873618	0.604683	**0.877721**	0.854077
Mean F1	**0.885670**	0.621934	0.885608	0.864640
Overall Accuracy	**0.887516**	0.628744	0.886915	0.866801
MCC	0.829910	0.436294	**0.831264**	0.798548
Best Fit Epoch	130	143	104	113
Total Training Time	00:40:56	10:36:27	15:03:56	00:32:32

GTR obtained the highest MCC score, higher by 0.032716 compared to the GINE operator. As MCC is a metric that is independent of class distribution, the difference in performance may indicate that GINE is slightly more sensitive to the distribution of classes and is less accurate when the balance of classes is accounted for. Finally, the GAT operator showed the lowest performance amongst all four operators.

4. Conclusion

This work presents a dataset of predicted protein-protein interactions and their associated virulence class labels. A GNN model with a choice of four graph operators is also proposed for the classification of the predicted interactions according to their assigned virulence levels. After experimentation, the RGGCN and GTR operators were found to perform similarly well, with the RGGCN being more efficient in terms of training time and stable training loss.

Future work includes more extensive interaction features such as those obtained from protein-protein interfaces or from directly incorporating protein 3D structure data, to improve the model for enhanced virulence classification.

Acknowledgments

This work was supported by the grants MOE2019-T2-2-175 and RG21/23, from the Ministry of Education, Singapore.

References

[1] Robert L. Perlman. Mouse models of human disease: An evolutionary perspective. *Evolution, Medicine, and Public Health*, 2016(1):170–176, 2016.

[2] Waqar Ahmad, Hilal Tayara, HyunJoo Shim, and Kil To Chong. SolPredictor: Predicting solubility with residual gated graph neural network. *International Journal of Molecular Sciences*, 25(2):715, 2024.

[3] Shuang Wang, Wenqi Chen, Peifu Han, Xue Li, and Tao Song. RGN: Residue-based graph attention and convolutional network for protein–protein interaction site prediction. *Journal of Chemical Information and Modeling*, 62(23):5961–5974, 2022.

[4] Samuel Sledzieski, Rohit Singh, Lenore Cowen, and Bonnie Berger. D-SCRIPT translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions. *Cell Systems*, 12(10):969–982, 2021.

[5] Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Milot Mirdita, Martin Steinegger, and Burkhard Rost. Bilingual language model for protein sequence and structure. *NAR Genomics and Bioinformatics*, 6(4):lqae150, December 2024.

[6] F. X. Ivan and C. K. Kwoh. Rule-based meta-analysis reveals the major role of PB2 in influencing influenza A virus virulence in mice. *BMC Genomics*, 20(Suppl 9):973, 2019. Type: Journal Article.

[7] Teng Ann Ng, Shamima Rashid, and Chee Keong Kwoh. Virulence network of interacting domains of influenza a and mouse proteins. *Frontiers in Bioinformatics*, 3:1123993, 2023.

[8] UniProt: The universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023.

[9] Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

[10] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[11] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

[12] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[14] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[15] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Appendix A. Dataset Creation

In [7], the records of [6] were further annotated and data for the C57BL/6 mouse strain was developed into a network of interacting mouse and IAV protein domains.

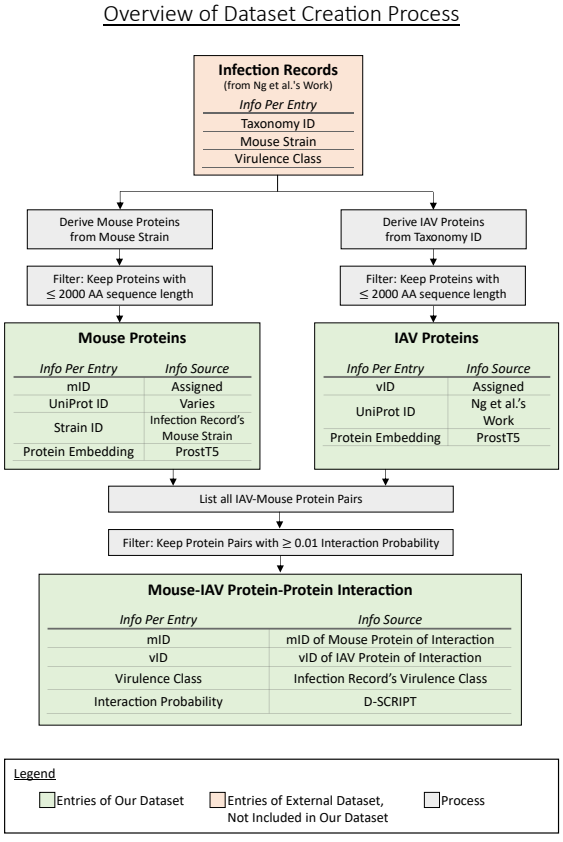

Fig. A1: Dataset Creation Process

Two assumptions have been made in this work to reduce computational complexity and to address

lack of data concerns. First, only pairwise interactions between mouse and IAV proteins have been considered, while in nature proteins function in complexes which may consist of more than 2 proteins. For example, multiple host proteins may bind to an IAV protein,as the viral infection progresses. Such types of interactions are not considered in this work to reduce the model complexity.

Second, the infection records' virulence labels were projected onto corresponding mouse-IAV protein-protein interactions and each interaction was treated as a training sample. This was done to provide increased training samples as there were insufficient virulence records for a graph-level classification task. Neural networks typically require thousands of training sample in order to generalise well on unseen data. However, in nature, not all protein interactions in an IAV infected mouse strains could be considered to have the same virulence level. F

Appendix B. Training Settings

The model was implemented with the PyTorch [13] and PyTorch Geometric [14] libraries.

Due to GNNs requiring large amounts of memory, a sub sampling approach [15] was applied on the input graphs to reduce the computational complexity.

The Adam optimizer was used with $\alpha = 0.001$ with a weight decay of 0.01. The cross-entropy loss was applied here. Under these settings, the model was trained for 150 epochs with a batch size of 1024 for each of the four graph operators - RGGCN, GAT, GTR and GINE. The best fit model was selected based on the lowest mean validation loss.