Towards *De Novo* Drug Design for the Coronavirus: A Drug-Target Interaction Prediction Approach using Atom-enhanced Graph Neural Network with Multi-hop Gating Mechanism

Duc Q. Nguyen, Khoan D. Le, Bach T. Ly, An D. Nguyen, Quang H. Nguyen, Tuan H. Nguyen, Tho T. Quan*
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University Ho Chi Minh City
Ho Chi Minh City, Vietnam

Cuong Quoc Duong, Phuong Thuy Viet Nguyen
University of Medicine and Pharmacy at Ho Chi Minh City
Ho Chi Minh City, Vietnam

Thanh N. Truong
University of Utah
Salt Lake City, Utah, United States

2022 9th NAFOSTED Conference on Information and Computer Science (NICS) | 978-1-6654-5422-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/NICS56915.2022.10013437

***Abstract*—For humans, the COVID-19 pandemic and Coronavirus have undeniably been a nightmare. Although there are effective vaccines, specific drugs are still urgent. Normally, to identify potential drugs, one needs to design and then test *interactions* between the drug and the virus in an *in silico* manner for determining candidates. This *Drug-Target Interaction* (DTI) process, can be done by *molecular docking*, which is too complicated and time-consuming for manual works. Therefore, it opens room for applying *Artificial Intelligence* (AI) techniques. In particular, *Graph Neural Network* (GNN) attracts recent attention since its high suitability for the nature of drug compounds and virus proteins. However, to introduce such a representation well-reflecting biological structures of biological compounds is not a trivial task. Moreover, since available datasets of Coronavirus are still not highly popular, the recently developed GNNs have been suffering from overfitting on them. We then address those issues by proposing a novel model known as *Atom-enhanced Graph Neural Network with Multi-hop Gating Mechanism*. On one hand, our model can learn more precise features of compounds and proteins. On the other hand, we introduce a new gating mechanism to create better atom representation from non-neighbor information. Once applying transfer learning from very large databanks, our model enjoys promising performance, especially when experimenting with Coronavirus.**

***Index Terms*—Coronavirus, Drug-Target Interaction, Graph Neural Networks, Multi-hop Gating Mechanism**

I. Introduction

Coronavirus (COVID-19) has caused deaths in the world in recent years. Besides vaccines, drugs are the keys for us to fight pandemic waves. Hence, drug design plays an essential role in the pandemic, especially when many virus variants were increasingly born. Basically, each type of virus has a specific *protein*, Hence, drug design is a process of *trial-and-error* that creates different *compounds* as illustrated in Figure 1. A compound can be represented as a graph whose vertices and edges are *atoms* and *bonds* respectively, and so is the virus. The designer can measure the effect of a designed compound by testing *interactions* between its atoms with those from the virus protein. If a compound introduces sufficiently many key interactions with the virus, it may effectively make the virus *inactivated*.

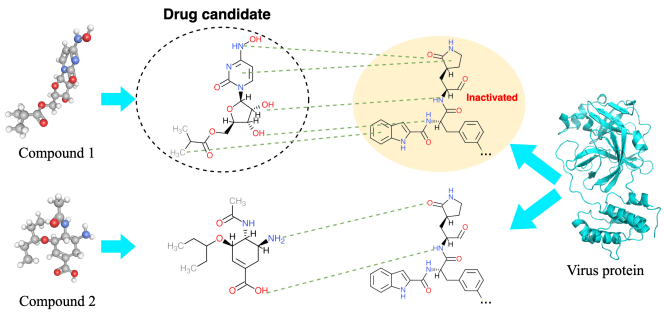

Fig. 1. The concept of interaction between drug and protein

To observe interactions between a compound and a virus, traditional *in silico* methods rely on *docking tools* such as AutoDock Vina, Smina, etc. However, as the docking process normally takes 3-5 minutes, it is hard for drug designers to explore large-scale of compounds to obtain the best designs [1]. With recent advancements in *Artificial Intelligence* (AI), AI-based models are used to predict the interactions between a compound and a protein to select only highly-interactive compounds for *in vitro* experiments. This prediction task is referred to as *Drug-Target Interaction* (DTI). Especially, *Graph Neural Networks* (GNNs) [2] emerged as the latest techniques for graph processing. Among state-of-the-art GNN-based models for DTI prediction, the study presented by [3] is the most remarkable as it does not only utilize GNNs but

*Corresponding author: qttho@hcmut.edu.vn

Authorized licensed use limited to: National University of Singapore. Downloaded on September 08,2025 at 03:04:53 UTC from IEEE Xplore. Restrictions apply.

also embeds the 3D structure of the compound and protein into graph features. This work serves as the baseline for our research. Moreover, in the COVID-19 pandemic, there are not many compounds being put in *in vitro* screening. The largest crowd-sourcing COVID-19 drug design system, COVID Moonshot [4], has experimented with approximately 2000 compounds. Among those compounds, only 379 ones have been captured co-crystal data i.e. the 3D structure by Fragalysis [5]. Thus, current drug-for-COVID datasets lack generalization for training AI models, making them suffer from *overfitting*. We also address this issue in our study by proposing a transfer learning strategy. In short, our contributions are two-fold as follows.

- We apply transfer learning to pretrain the baseline model on a suitable known-interaction variant of the dataset from Protein Data Bank [6].
- We propose a new model, known as *Atom-enhanced Graph Neural Network with Multi-hop Gating Mechanism*, extended from the baseline one introduced in [3] with three major improvements as follows: (i) *Enriched atom encoding*: we additionally encode atoms with more important attributes; (ii) *Total atom aggregation*: we enhance the baseline model by aggregating not only the atoms of compounds but also those of the proteins to produce more informative representation for the input; and (iii) *Multi-hop gating mechanism*: we modify the original gating mechanism to allow atoms affect other non-neighbor ones, earning improved interaction prediction performance.

II. Preliminaries and Related Works

A. Drug-Target Interaction problem

The Drug-Target Interaction problem is common in drug discovery and solved by methods from traditional to modern ones [7]. This problem takes a compound and a target protein as its inputs and uses an algorithm to predict if a compound interact (be pharmacologically active) with a protein or not [7]. A compound can be considered active or inactive based on a predefined threshold [8].

B. Modern DTI prediction techniques

Many techniques from *Machine Learning* (ML) to *Deep Learning* (DL) such as Support Vector Machine, Ensemble Learning, Transformer, GNN, etc. were applied to speed up the process of making vaccines and drugs and it makes no exception for the Drug-Target Interaction problem [9]. Conceptually, recent DTI models initially use suitable AI techniques to encode the compound and protein into feature vectors. Then, these models aggregate the two feature vectors together to feed into a classifier to predict if the compound is pharmacologically active with the protein or not [3], [10]–[12]. Some models utilize the attention mechanism to enhance the out extracted feature vectors [3], [10], [11]. Another approach to DTI problem is based on the "guilt-by-association" principle. Methods in this approach try to find similar proteins and compounds in the known-interaction datasets and then use them as evidence to predict the input compound-protein. The biggest shortcoming of this approach is unstable performance when working with unusual proteins or compounds, because there isn't much "evidence" guiding the model [7].

C. Graph Attention Network Layer

The *Graph Neural Networks* (GNN) contains multiple layers. The input graph after being passed through N GNN layers will be aggregated to form a representation vector. Then, that vector can be used in multiple downstream tasks. A conceptual view of a GNN model is illustrated in Figure 2.

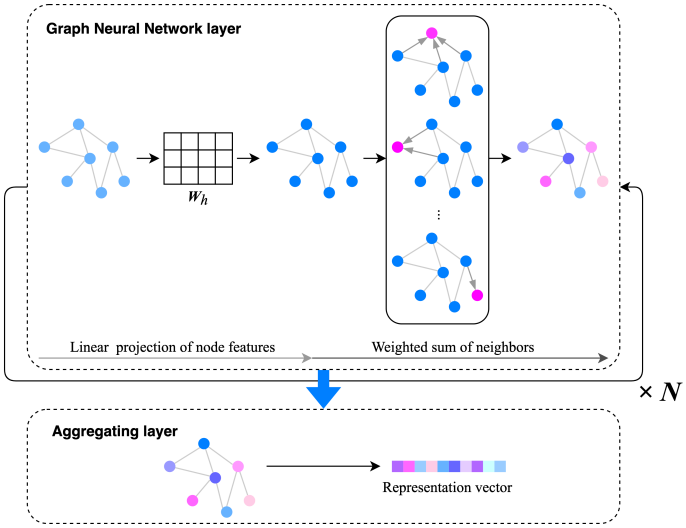

Fig. 2. Conceptualization of a GNN model

In this study, we utilize the power of *Graph Attention Networks* (GAT) [13], which is the most well-known variant of GNNs. Assuming we have a graph $\mathcal{G} = (V, E)$ where V is a set of nodes, E is a set of edges and each node has a fixed number of features F, the input for GAT contains an adjacency matrix $\boldsymbol{A}$ and a list of node feature vectors $\boldsymbol{X}$ defined as (1) and (2).

$$\boldsymbol{A}_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are connected by an edge} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$\boldsymbol{X} = \{x_1, x_2, \ldots, x_{|V|}\} \text{ with } x_i \in \mathbb{R}^F \tag{2}$$

Firstly, the GAT performs linear projection to put the node feature vectors into F'-dimensioned embedding space by (3). In (3), $\boldsymbol{W}_h \in \mathbb{R}^{F' \times F}$ is a learnable weight matrix, and x_i^h is the i-th node projected feature vector.

$$x_i^h = \boldsymbol{W}_h x_i, \ i = \overline{1, |V|} \tag{3}$$

Then the GAT calculates attention coefficients e_{ij} for all pairs of (i, j) nodes. These coefficients are then normalized by the *softmax* function to decrease the bias and the cost of computing. When normalizing for e_{ij}, only nodes which are connected to i-th node are considered. Finally, the higher

Authorized licensed use limited to: National University of Singapore. Downloaded on September 08,2025 at 03:04:53 UTC from IEEE Xplore. Restrictions apply.

representation of each node is produced by weighted sum of its neighbor nodes using attention coefficients. (4a), (4b) and (4c) are formal definitions of above operations.

$$\begin{cases} e_{ij} = (x_i^h)^T \boldsymbol{W}_a x_j^h + (x_j^h)^T \boldsymbol{W}_a x_i^h, \ i,j = \overline{1,|V|} & \text{(4a)} \\ a_{ij} = \dfrac{exp(e_{ij})}{\sum_{k \in C_i} exp(e_{ik})} \boldsymbol{A}_{ij}, \ i,j = \overline{1,|V|} & \text{(4b)} \\ {x'}_i = \sum\limits_{j \in C_i} a_{ij} x_j^h, \ i = \overline{1,|V|} & \text{(4c)} \end{cases}$$

In (4a), e_{ij} is the attention coefficient reflecting the importance of j-th atom to i-th atom and $\boldsymbol{W}_a \in \mathbb{R}^{F' \times F'}$ is a learnable weight matrix. In (4b), a_{ij} is the normalized attention coefficient corresponding to e_{ij} and C_i is the set of neighbor nodes of i-th node. In (4c), ${x'}_i$ is the higher representation of i-th node feature vector. Then the list $\boldsymbol{X'} = \{{x'}_i \in \mathbb{R}^{F'} | i = \overline{1,|V|}\}$ is the output of the GAT layer.

III. The Atom-enhanced Graph Neural Network Model with Multi-hop Gating Mechanism for Drug-Target Interaction Prediction

A. The baseline model

Our baseline model reuses the study of Lim [3]. Figure 3 illustrates the model overview.

1) Model input representation: Firstly, the model takes the input of a compound and a protein as a graph, which is presented by a matrix of atom features ($\boldsymbol{X}$) and two adjacency matrix ($\boldsymbol{A}^1$, $\boldsymbol{A}^2$) and (5), (6) and (7) show how to create these matrices, respectively. Figure 3 has visualized a conceptual view of matrix $\boldsymbol{X}$, $\boldsymbol{A}^1$ and $\boldsymbol{A}^2$.

$$\boldsymbol{X} = \{x_1, x_2, \ldots, x_M\} \text{ with } x_i \in \mathbb{R}^F \tag{5}$$

$$\boldsymbol{A}^1_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are connected by covalent bond or } i = j \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$\boldsymbol{A}^2_{ij} = \begin{cases} \boldsymbol{A}^1_{ij} & \text{if } i,j \in \text{protein or } i,j \in \text{compound} \\ e^{-(d_{ij}-\mu)^2/\sigma} & \text{if } i \in \text{protein and } j \in \text{compound,} \\ & \text{or if } i \in \text{compound and } j \in \text{protein} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

In (5), x_i is a feature vector of an atom which contains F features shown in Table I and M is the total atoms in the graph representing both compound and protein. In (6) and (7), i and j are the atom indexes with the same order as of $\boldsymbol{X}$. $\boldsymbol{A}^1_{ij}$ and $\boldsymbol{A}^2_{ij}$ are the elements at i-th row and j-th column in the $\boldsymbol{A}^1$ and $\boldsymbol{A}^2$ matrix, corresponding. In (7), d_{ij} is the distance between i-th atom and j-th atom and μ and σ are learnable parameters. To adapt this model for our chosen datasets, we have modified the required input by replacing the 8Å-radius atoms of protein by the atoms in the *protein pocket* as hinted by [14]. Therefore, using the protein cavity makes much more sense than the original method.

TABLE I
The list of atom features used in original study and in Improvement 1

Feature	Value
Original	
Atom type	C,N,O,S,F,P,Cl,Br,B,H (onehot)
Degree of atom	0, 1, 2, 3, 4, 5, 6 (onehot)
Number of H atoms attached	0, 1, 2, 3, 4 (onehot)
Implicit valence electrons	0, 1, 2, 3, 4, 5 (onehot)
In aromatic	0 or 1
Added in Improvement 1	
Hydrogen D/A	[is_donor, is_acceptor]
Pos/Neg Ionizable	[is_pos, is_neg]
In lumped hydrophobe	0 or 1

After all the inputs are prepared, they are then passed to the model for predicting compound-protein interaction.

2) Model architecture: In this baseline model, the GAT layer is used as the main layer for feature extraction. However, they modified the original GAT layer by adding a *gating mechanism* at the end of the layer to control how much feature information is passed through. In a formal form, Equation 4c of the original GAT is replaced by (8).

$$\begin{cases} x_i^{temp} = \sum_{j \in C_i} a_{ij} x_j^h, \ i = \overline{1,|V|} \\ z_i = \sigma(\boldsymbol{W}_o (x_i || x_i^{temp}) + b), \ i = \overline{1,|V|} \\ {x'}_i = z_i x_i + (1 - z_i) x_i^{temp}, \ i = \overline{1,|V|} \end{cases} \tag{8}$$

In (8), $\boldsymbol{W}_o \in \mathbb{R}^{1 \times 2{F'}_n}$ is a learnable weight matrix and '||' is the concatenation operator.

Let $\boldsymbol{GAT}()$ be the formal representation of all GAT layer formulations which are (3), (4a), (4b) and (8). With the input of ($\boldsymbol{X}$, $\boldsymbol{A}^1$, $\boldsymbol{A}^2$) created from the complex of protein and compound, we define a GAT block that takes these input and produces the higher representation for $\boldsymbol{X}$. Specifically, the GAT block separates the input into ($\boldsymbol{X}$, $\boldsymbol{A}^1$) and ($\boldsymbol{X}$, $\boldsymbol{A}^2$), passes them to GAT layer to get output ${\boldsymbol{X}'}_1$ and ${\boldsymbol{X}'}_2$ and perform subtraction ${\boldsymbol{X}'}_2 - {\boldsymbol{X}'}_1$ for model to learn the difference between the structure in a binding pose and the structure as separated. Equation 9 presents insights of a GAT block.

$$\begin{cases} {\boldsymbol{X}'}_1 = \boldsymbol{GAT}(\boldsymbol{X}, \boldsymbol{A}^1) \\ {\boldsymbol{X}'}_2 = \boldsymbol{GAT}(\boldsymbol{X}, \boldsymbol{A}^2) \\ {\boldsymbol{X}'}_{out} = {\boldsymbol{X}'}_2 - {\boldsymbol{X}'}_1 \end{cases} \tag{9}$$

In (9), ${\boldsymbol{X}'}_{out}$ is the output of GAT block. According to the original study, authors stacked N GAT block to achieve better feature representations. This can be done by using the output of the previous GAT block ${\boldsymbol{X}'}_{out}$ with two adjacency matrices $\boldsymbol{A}^1$, $\boldsymbol{A}^2$ as the input for the next GAT block. Please notice that the number of nodes in the GAT layer is equal to the total number of atoms of both protein and compound ($|V| = M$).

The output refined atom feature vectors of the last GAT block are aggregated in the next step to form a feature vector

Authorized licensed use limited to: National University of Singapore. Downloaded on September 08,2025 at 03:04:53 UTC from IEEE Xplore. Restrictions apply.

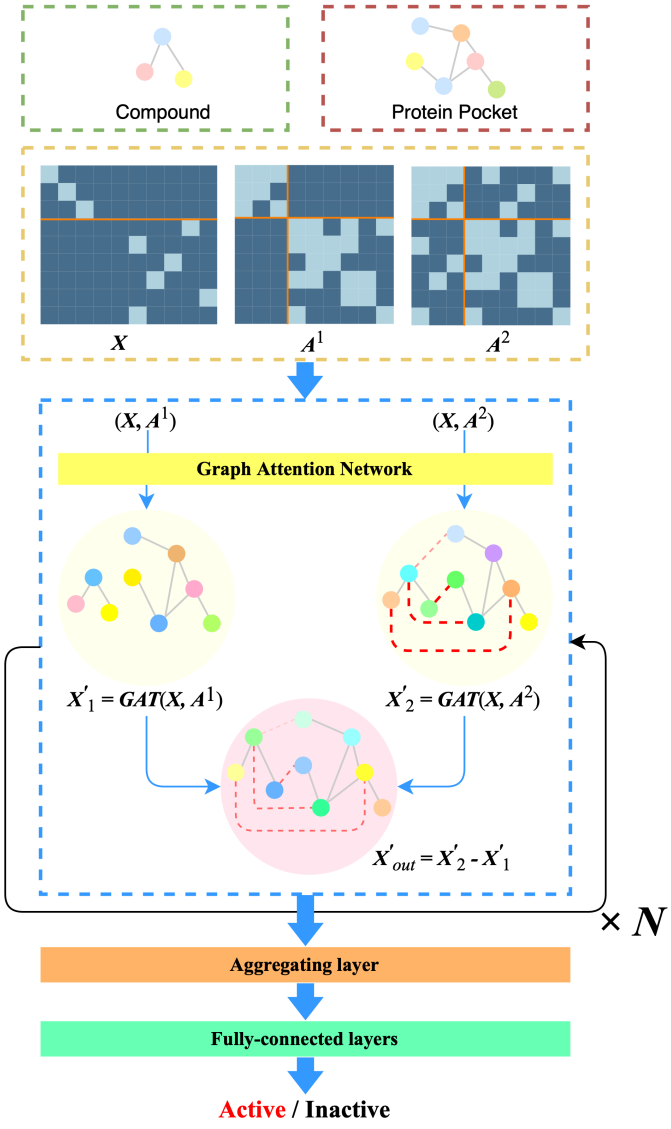

Fig. 3. The architecture of the baseline model

$x^{complex}$ representing the complex of the input protein and compound. Equation 10 gives the formulation for creating this vector. Finally, a classifier with multiple fully-connected layers is employed to decide if the input complex is active or not. A fully-connected layer is a non-linear transformation that is defined in (11).

$$x^{complex} = \sum_{i \in compound} x_i \tag{10}$$

$$y = \sigma(\boldsymbol{W}_c x + b) \tag{11}$$

In (11), x, and y are respectively the input and output fully-connected layer. The $\boldsymbol{W}_c$ is a learnable weight matrix and b is the bias. Each fully-connected layer in the classifier has its activation function σ as the $ReLU$ function except the $sigmoid$ function for the final one.

B. The transfer learning strategy

As mentioned above, to deal with insufficient drug data of Coronavirus's Mpro protein, we apply a transfer learning strategy for learning the general interaction rules before exploring specific rules of Mpro protein. Firstly, for each model with configured settings (e.g improvements), we perform pretraining using the PDBbind dataset [15] for the model to obtain the generalizability. After that, we finetune each model using the Fragalysis dataset. To demonstrate how transfer learning affects the model performance, we also train the scratch model with the Fragalysis dataset for comparison.

C. The Atom-enhanced GNN with Multi-hop Gating Mechanism

In this study, we introduce an improved model from the baseline one, known as *Atom-enhanced GNN with Multi-hop Gating Mechanism*[1], in which the following improvements are carried out.

1) Improvement 1: Enriched atom encoding: The first improvement is enriching atom encoding which adds more chemical features to the representation of each atom. The newly added features include atom degree of six, whether an atom is a hydrogen donor or hydrogen acceptor, whether an atom can be positive or negative ionizable, whether an atom is in any lumped hydrophobe, which are the prerequisites of the corresponding bonding type (a hydrogen bonding requires a hydrogen donor and a hydrogen acceptor atom, so on). Table I summarizes all features that are used in this improvement.

2) Improvement 2: Total atom aggregation: The second improvement basically bases on an assumption that interactions are only created if the protein and the compound match some interaction rules [9]. Therefore, we have modified the original aggregating layer which calculates the sum of all compound atom vectors to a combination of compound and protein representation vector. With our modification, any protein atoms that have a minimum distance to any compound atoms less than 5Å will be taken into consideration for interaction prediction. The mathematical formulations for our new aggregating layer are proposed in (12a)-(12c).

$$x^{complex} = (x^{compound} || x^{protein}) \tag{12a}$$

$$x^{compound} = \sum_{i \in compound} x_i \tag{12b}$$

$$\begin{cases} x^{protein} = \sum_{i \in P} x_i \\ P = \{x_p, p \in protein | \exists c \in compound : dist(p,c) < 5\text{Å}\} \end{cases} \tag{12c}$$

In (12a), '||' is the concatenation operator and in (12c), $dist(p, c)$ is the Euclidean distance between protein atom p and compound atom c.

[1]Our implementation is available at https://github.com/ViDok-BK/GMGM

Authorized licensed use limited to: National University of Singapore. Downloaded on September 08,2025 at 03:04:53 UTC from IEEE Xplore. Restrictions apply.

3) Improvement 3: Multi-hop gating mechanism: The third improvement is the multi-hop gating mechanism. In this improvement, we repeat the calculation of the gating mechanism multiple times. The reason for this improvement is to enlarge the receptive field of an atom, which is basically based on an assumption in chemistry that atoms having the same function (e.g. hydrophilic, hydrophobic, etc.) usually concentrate together and create a wide area of influence over non-neighbors. Figure 4 gives an intuitive view of this mechanism. This improvement is inspired by the attention diffusion mechanism proposed by Wang *et al.* [16]. The process of repeating the gating mechanism calculation exactly matches the process of approximate computation for attention diffusion except for the gating coefficient z_i which is computed from node feature vectors compared to manually input in Wang's study. Let K be the number of hops in the receptive field of an atom, the process of calculating multi-hop gating mechanism is presented in Algorithm 1. Please notice that in our proposed improvement, we follow Wang's study to use x_i^h for computing $x_i^{(k)}$, $k = \overline{1, K}$ instead of x_i as in original study of Lim.

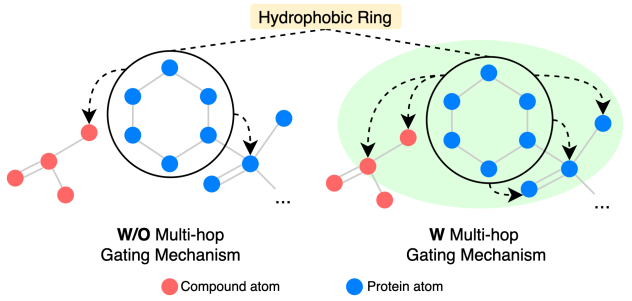

Fig. 4. The difference between models with and without multi-hop gating mechanism

Algorithm 1: Multi-hop gating mechanism

Input : Normalized attention coefficients a_{ij}, where $i, j = \overline{1, |V|}$
Atom feature vectors x_i^h, where $i = \overline{1, |V|}$
Number of hops K

Output: Refined atom feature vectors $x_i^{(K)}$, where $i = \overline{1, |V|}$

$x_i^{(0)} = x_i^h,\ i = \overline{1, |V|}$

for *k in Range*$(1 \dots K)$ **do**

$x_i^{temp} = \sum_{j \in C_i} a_{ij} x_j^{(k-1)},\ i = \overline{1, |V|}$

$z_i = \sigma(\boldsymbol{W}_o(x_i^{(0)} || x_i^{temp}) + b),\ i = \overline{1, |V|}$

$x_i^{(k)} = z_i x_i^{(0)} + (1 - z_i) x_i^{temp},\ i = \overline{1, |V|}$

return $\boldsymbol{X}^{(K)} = \{x_i^{(K)} | i = \overline{1, |V|}\}$

IV. Experiments

A. Datasets and configurations

First of all, we summarize two datasets that are used to train and test our model in Table II. The PDBbind dataset is splitted into training and testing sets with a ratio of 8:2 while the Fragalysis dataset is splitted with a ratio of 7:3. Table II indicates the number of active and inactive complexes (samples) in each dataset. A complex of protein and compound is labeled active if its IC50 is equal to or lower than $2.5\mu M$, otherwise, it is inactive. Due to the disproportion in the number of active and inactive samples, we implemented the undersampling technique to the class with a higher quantity to wipe out the bias in the training process.

TABLE II
THE NUMBER OF COMPOUND-PROTEIN COMPLEXES USED FOR TRAINING AND TESTING OF EACH DATASET

	PDBbind			**Fragalysis**		
	Active	Inactive	Total	Active	Inactive	Total
Training	10037	5237	15274	75	125	200
Testing	2530	1287	3817	35	54	89

TABLE III
AUC SCORES OF BASELINE MODEL COMPARED TO OTHER MODELS IN VARIOUS SETTINGS GROUPED BY MOLECULAR REPRESENTATION TYPE

Model with settings	**Directly trained on Fragalysis**	**Pretrained on PDBbind**	**Finetuned on Fragalysis**
String-based representation			
DeepDTA	0.870	**0.849**	0.862
String-based + Feature matrix representation			
DrugVQA	0.853	0.819	0.820
Graph-based + String-based representation			
GraphDTA-GINConvNet	0.885	0.838	0.874
GraphDTA-GATNet	**0.886**	0.814	0.890
GraphDTA-GCNNet	0.868	0.836	0.862
GraphDTA-GAT_GCN	0.874	0.835	0.874
Graph-based representation			
Baseline model	0.841	0.758	0.859
Baseline + Ipmt 1	0.865	0.787	0.896
Baseline + Ipmt 2	0.877	0.785	0.915
Baseline + Ipmt 3	0.870	0.793	0.936
Baseline + Ipmt 1,2	0.822	0.813	0.930
Baseline + Ipmt 1,2,3	0.868	0.820	**0.938**

To validate our model performance, we compare it with other deep learning-based ones. Particularly, we choose top-tier models which requires different input representations including DeepDTA (string-based) [12], DrugVQA (string-based for compound + feature matrix for protein) [10] and GraphDTA (graph-based for compound + string-based for protein) [11]. We use the metric of *Area Under the ROC Curve* (AUC score) to judge overall performance between models. Table III summarizes the AUC score of the above models and our model in different settings and scenarios.

B. The benefit of transfer learning strategy

In Table III, we can observe accuracy when our models have been pre-trained on the PDBbind compared to directly trained on the Fragalysis. Specifically, toward the baseline model, the AUC score increases from 0.841 to 0.859 (increased 0.018). With settings containing first two improvements, the difference is significant up to 0.108. This suggests that without pretraining, the models can not learn how compounds interact with proteins and seem overfitted when accompanying with

Authorized licensed use limited to: National University of Singapore. Downloaded on September 08,2025 at 03:04:53 UTC from IEEE Xplore. Restrictions apply.

our improvements. Although the results pretrained on the PDBbind dataset are not too high, the pretraining process helps our models increase their generalizability and learn the general interaction principles.

C. The effects of proposed improvements

According to Table III, when applying each improvement separately, the results have higher AUC scores than that of the baseline model in all scenarios. The most effective improvement belongs to the multi-hop gating mechanism. With this mechanism, finetuned model achieves 0.936 (improved 0.077 from the baseline). Moreover, when our improvements are combined together, they boost the model performance. Toward the best-achieved results, the setting including all three improvements reaches an outstanding score of 0.938 in the finetuning scenario. From the above observations, we can consider that the multi-hop gating mechanism is really effective in creating a larger influential field for an atom or group of atoms, which leads to outstanding results.

D. Comparison with other methods

When being trained directly on Fragalysis, the combination of the baseline model and Improvement 2 achieves 0.877 AUC score, which is better than that of DeepDTA and DrugVQA, at 0.870 and 0.853, respectively. In case being pretrained on PDBbind, our models show slightly lower performance than the others. In spite of the unsatisfied results on the PDBbind dataset, our models outperform the others when being finetuned on the Fragalysis dataset. In detail, all settings that have our improvements achieve AUC scores from 0.896 up to 0.938, which are strictly higher than the highest AUC score of GraphDTA (0.890), DeepDTA (0.862), and DrugVQA (0.820). These outstanding results suggest that our improved models can learn general interaction principles and thus, they achieve better results than other methods.

V. Conclusion

Drug-Target Interaction prediction has been an essential and last-long problem in drug discovery. Toward the COVID-19 situation, the more effective the method of solving this problem is, the quicker and cost-lower the drug development process is. With our proposed model together with the transfer learning strategy, we achieve a noticeable performance compared to the baseline model and other state-of-the-art ones. Therefore, our model can be applied in COVID-19 treatment research centers to boost their productivity. In the future, our model can be integrated some weighting functions for assessing the importance of both intra-molecular and inter-molecular interactions. Moreover, the loss function can be upgraded for evaluating the strength of pairwise interactions or the GAT layer can include edge features. In conclusion, this model has much room to improve including both the architecture and the optimization process.

Acknowledgements

This research is funded by Ho Chi Minh City University of Technology (HCMUT) under grant number SVKSTN-2021-KH&KTMT-38. We acknowledge the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

References

[1] S. Zev, K. Raz, R. Schwartz, R. Tarabeh, P. K. Gupta, and D. T. Major, "Benchmarking the Ability of Common Docking Programs to Correctly Reproduce and Score Binding Modes in SARS-CoV-2 Protease Mpro," *Journal of Chemical Information and Modeling*, vol. 61, no. 6, pp. 2957–2966, 2021.

[2] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[3] J. Lim, S. Ryu, K. Park, Y. J. Choe, J. Ham, and W. Y. Kim, "Predicting Drug–Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation," *Journal of Chemical Information and Modeling*, vol. 59, no. 9, pp. 3981–3988, 2019.

[4] J. Chodera, A. A. Lee, N. London, and F. von Delft, "Crowdsourcing drug discovery for pandemics," *Nature Chemistry*, vol. 12, no. 7, pp. 581–581, Jul 2020.

[5] Diamond, "Fragalysis," https://fragalysis.diamond.ac.uk, 2021, accessed on Oct. 2021.

[6] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 01 2000.

[7] M. Thafar, A. B. Raies, S. Albaradei, M. Essack, and V. B. Bajic, "Comparison Study of Computational Prediction Tools for Drug-Target Binding Affinities," *Frontiers in Chemistry*, vol. 7, 2019.

[8] K. Y. Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey, and P. Zhang, "Interpretable drug target prediction using deep neural representation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 3371–3377.

[9] S. Lim, Y. Lu, C. Y. Cho, I. Sung, J. Kim, Y. Kim, S. Park, and S. Kim, "A review on compound-protein interaction prediction methods: Data, format, representation and model," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 1541–1556, 2021.

[10] S. Zheng, Y. Li, S. Chen, J. Xu, and Y. Yang, "Predicting drug–protein interaction using quasi-visual question answering system," *Nature Machine Intelligence*, vol. 2, no. 2, pp. 134–140, Feb 2020.

[11] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh, "GraphDTA: predicting drug–target binding affinity with graph neural networks," *Bioinformatics*, vol. 37, no. 8, pp. 1140–1147, 10 2020.

[12] H. Öztürk, A. Özgür, and E. Ozkirimli, "DeepDTA: deep drug–target binding affinity prediction," *Bioinformatics*, vol. 34, no. 17, pp. i821–i829, 09 2018.

[13] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[14] D. K. Johnson and J. Karanicolas, "Druggable Protein Interaction Sites Are More Predisposed to Surface Pocket Formation than the Rest of the Protein Surface," *PLOS Computational Biology*, vol. 9, no. 3, pp. 1–10, 03 2013.

[15] R. Wang, X. Fang, Y. Lu, C.-Y. Yang, and S. Wang, "The PDBbind Database: Methodologies and Updates," *Journal of Medicinal Chemistry*, vol. 48, no. 12, pp. 4111–4119, 2005.

[16] G. Wang, R. Ying, J. Huang, and J. Leskovec, "Multi-hop attention graph neural networks," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 3089–3096, main Track.

Authorized licensed use limited to: National University of Singapore. Downloaded on September 08,2025 at 03:04:53 UTC from IEEE Xplore. Restrictions apply.