KATNet: A Dynamic Protein-Ligand Binding Affinity Prediction Model Using Cross-Attention Based 3D-CNN

Kathan Joshi*, Hyuntae Na†, *Member, IEEE*
School of Science, Engineering and Technology, Penn State Harrisburg, United States
Email:*†{ksj5289, hzn17}@psu.edu

***Abstract*—Predicting protein-ligand binding affinities is an essential problem in computational biochemistry. It has been envisioned to boost drug optimization and development research substantially. To achieve precise and efficient affinity prediction, many deep learning methods have been employed in recent years. The affinity prediction largely depends on the structural properties of the considered protein and ligand, respectively. These structures of protein and ligand are assumed to be stable or non-dynamic. However, in reality, both structures are present in an unstable form, i.e., their structure is dynamic or keeps changing in nature. To this aim, we try to investigate the role of the structural change of the protein molecule in the formation of protein-ligand bond. Henceforth, we propose a novel deep-learning architecture, namely, KATNet, to accurately predict the binding affinities while contemplating the structural change of the protein molecule into the system. Additionally, we have also provided a new feature representation matrix that contemplates and encodes the structural change in protein molecules with the Anisotropic Network Model (ANM) method at its core.**

***Index Terms*—Protein structure, 3D-Convolution, Drug Discovery, Ligand, Binding Affinity, Cross-Attention, Convolution Neural Network, Normal Model Analysis, Anisotropic Network Model, Deep Learning.**

I. Introduction

Proteins are naturally occurring complex biomolecules responsible for various biological tasks in all living organisms. All human life activities largely depend on different types of proteins present in the human body, with up to three billion proteins in the human body. Nonetheless, these protein molecules cannot conduct these life activities independently. They need to interact with other molecules known as ligands [1]. For instance, mostly all the kinase enzymes present in the human body need to interact with adenosine triphosphate (ATP) molecules to achieve its functionality [2]. Mutation in these proteins could lead to various life-threatening diseases such as cancer and Parkinson's [3] [4] [5]. Nevertheless, if we could develop small organic drug molecules or ligands that target the kinase protein forms bond with them, and regulate its functionality it could cure many diseases. However, not every candidate drug molecule or ligand is compatible to form a bond with the target proteins, i.e., their affinity largely varies. Henceforth, predicting accurately the protein-ligand binding affinity is an important field of computational biochemistry.

Moreover, discovering and developing novel drugs is a very difficult and expensive task. In reality, only a few new drugs get approved every year which is very diminutive in comparison to the extent of possible chemical configurations available and the billions of dollars in funding available in drug development campaigns. This is because the affinity largely depends on the structural and physicochemical properties of the considered target protein and candidate ligand. Due to the complexity of the problem, computer-aided solutions are essential to optimize the development process.

Many traditional techniques such as AutoDock [6], GOLD [7], X-score [8], etc., have been developed to analyze the protein-ligand interaction. However, these methods involve the use of complex algorithms and require specialized expertise in relevant fields to implement them, which makes them time-consuming and arduous to utilize. To this aim, contemporary research activities focus on the application of deep-learning methods in predicting protein-ligand binding affinities. Nonetheless, these deep learning methods can further be divided into interaction-free and interaction-based models [9]. The difference is the type of training data used, one uses a bonded protein-ligand structure and the other takes in two distinct structures as input. Moreover, the scope of this paper is limited to the study of interaction-free models.

A. Previous Related Work

In general, two wide categories of deep learning methods: convolutional neural networks (CNN), and graph neural networks (GNN) are used to predict the binding affinities. However, convolutional models are more widely used for this problem. This method uses the structural information of both protein and ligand to predict the affinity. For instance, [10] introduced a topology net, which represented the 3D structural information into one dimension and then calculated the binding affinity by applying a 1D convolutional neural network. Although this reduced the computational complexity, the information loss was huge for this method. Henceforth, 3D convolutional neural networks are more widely used to limit information loss. [11], [12], [13], and [14] are some really good 3D CNN based models. However, each uses different feature representations (but still voxel) for their models. Currently, [13] holds the benchmark performance for the PDBbind (v.2016) dataset. Some authors [15] also use GNN to predict the affinities, where they encode the structural information into a graph and feed it into a network. In [16], authors have tried to fuse the 3D-CNN with GNN to predict the affinity. Authors in [17] use

Transformers to extract the sequence information of the protein and graph attention network (GAT) to predict the affinity. However, they still face the information loss problem since they are not considering the 3D structure information. Nevertheless, no previous literature contemplates the fluctuations of protein molecules while predicting the binding affinity.

B. Motivation

The key motivation behind this work is to analyze a more realistic scenario of a protein-ligand binding process. As discussed above, to date, many deep-learning techniques have been proposed to predict the protein-ligand binding affinities accurately. However, to the best of our knowledge, none of them study the effect of dynamic protein structure in the system. Hence, in this work, we analyze a more realistic scenario by considering the protein to be dynamic in nature, meaning the structure or the protein molecule is fluctuating with respect to its stable structure, as reviewed in [18] [19] [20] and references therein. Furthermore, we consider the structure of the ligand molecule to be stable in its natural form.

The main principle behind considering such assumptions is that almost all proteins never exist in a single conformation state. They tend to fluctuate a lot and each protein's functionality largely depends on how they fluctuate. For instance, as discussed in [21], Heme is located in the active sites of protein Hemoglobin and Myoglobin. The protein fluctuations lead to the distortion of these heme porphyrin molecules which greatly affects the various biological functions such as electron transfer, oxygen transport, and/or storage. Furthermore, each fluctuated conformation of the protein molecule is notably different from its stable form. Hence, it is important to consider the structural fluctuations of protein molecules when predicting the protein-ligand binding affinities. In particular, we apply the elastic network model analysis to the protein structure to reproduce the fluctuations in their native conformations.

Additionally, we consider the structure of the ligand molecules stable (in their native conformation). Since ligand molecules are considerably smaller compared to the protein molecules. We assume the ligand fluctuation is negligible compared to that of proteins.

C. Contribution

In particular, the main contributions of the paper are as follows:

1) We propose a novel deep learning (DL) architecture, namely KAT-Net, to predict the binding affinity between an unstable (dynamic) protein structure and a stable ligand structure.
2) We also proposed a new input representations matrix to feed into the proposed DL architecture. The described representation matrix encodes the structural change of the protein molecule with ANM model at its core.

The remainder of this work is organized as follows. In Section II, we introduce the dataset used for training and benchmarking purposes. Section III includes the description of the proposed feature extraction method and KATNet model. In Section IV, we include the results of this work and in Section V we draw the main conclusions resulting from this work and suggest some ideas for future works.

II. Materials

In this section, we discuss in detail the data used for both training and benchmarking purposes. In particular, we have used the PDBbind (v.2020) database, containing 19,433 protein-ligand complexes and their equivalent experimentally determining binding affinities collected from different works of literature in terms of the dissociation constant (K_d), inhibition constant (K_i) or half-concentration constant (IC_{50}). Note higher values of these constants correspond to weaker bonds between protein and ligand. A refined subset containing 3,849, i.e., $N_r = 3849$ protein-ligand structures was extracted from it. The extraction is based on the quality of structural resolution and experimental precision of the binding affinities values.

In general, we follow the protocol described in [11] to extract our refined dataset. Specifically, we excluded every complex with a resolution higher than 2.5 Å, or R-factor higher than 0.25. We also exclude the complexes if ligands are bounded through covalent bonds or if the experimental affinity is not reported in K_d or K_i or if $K_d < 1pM$. An even smaller subset of the refined set was made, namely, the core set consisting of 284, i.e., $n_c = 284$ protein-ligand complexes for the benchmarking (testing) purpose.

Many literatures [22] [23] [24], have discussed that during the training of a model, the standard train and test split of the PDBbind database data tend to yield overly optimistic results. To avoid this many authors train their respective deep-learning models on different data sets. However, for the sake of simplicity, the scope of this paper is only limited to the PDBbind database.

III. Methods

A. Input Representation

In particular, we have three inputs to our proposed deep learning architecture as follows:

1) Protein Structure (.PDBQT file)
2) Ligand Structure (.PDBQT file)
3) Protein-Pocket Structure (.PDB)

For the protein and ligand structure, we have adopted a set of descriptors from [25] to represent them in 3D voxel input representation. At first, a 24 $\mathring{A}^3$ 3-D grid is constructed which contains the protein and ligand. After, the relative coordinates of protein and ligands are computed with the centroid of ligand coordinates as the origin. Now, for the feature construction, we only consider those atoms of protein and ligand which are inside this grid. Nevertheless, we follow the same procedure to extract the protein and ligand voxel representation, but with the aim of describing it lucidly, we only describe the procedure for the protein molecules. The whole 24 $\mathring{A}^3$ 3-D grid is further discretized into 1 $\mathring{A}^3$ smaller grids (or voxel) and each voxel occupancy is determined on two things, firstly, depending on the volume of the protein atom (atom occupancy) inside the smaller voxel volume and on other based on the seven physicochemical properties atoms. Moreover, we find the contribution of each atom to each grid point, i.e., atom occupancies by using the van der Waals radius for each atom

TABLE I: Rules defined for computing voxel occupencies

Property	**Rule**
Hydrophobic	Aliphatic or aromatic C
Aromatic	Aromatic C
Hydrogen bond acceptor	Acceptor 1 H-bond or S spherical N; Acceptor 2 H-bond or S spherical O; Acceptor 2 H-bonds S
Hydrogen bond donor	Donor 1 H-bond or Donor S spherical H with either O or N partner
Positive ionizable	Gasteiger positive charge
Negative ionizable	Gasteiger negative charge
Metallic	Mg, Zn, Mn, Ca, or Fe
Atom Occupancy	calculated based on eq. 1

type based on the following formula:

$$c(r) = 1 - exp\left(-\left(\frac{r_{vdw}}{r}\right)^{12}\right) \tag{1}$$

where, r is the Euclidean distance between the atom and the grid point. To be specific, each voxel occupancy is calculated based on the rules described in Table I. Now, since each voxel representation is categorized in 8 different properties (or channels) our input feature shape for protein molecule is of $(8, 24, 24, 24)$. To visualize these channels of a voxel, it can be viewed as pixel values of red, green, and blue of an RGB image. As mentioned previously we use a similar procedure to obtain the voxel representation of the ligand molecule and the input representation for the ligand molecule will also have the shape $(8, 24, 24, 24)$. However, instead of feeding these input representations separately, we concatenate the ligand features and protein features for the first dimension since we want to capture the combined information on how the structural properties of protein and ligand affect their binding affinity. Therefore, our new combined feature becomes of shape $(16, 24, 24, 24)$. To extract the voxel representations specific voxelization routines are available in the HTMD Python framework.

To capture the dynamic fluctuations of the protein molecule we use the anisotropic network model (ANM) elastic model, described in paper [26] analysis at its base. Firstly, We compute a $3N \times 3N$ Hessian matrix H, where N represents the number of C atoms in the protein molecule's pocket. Here, we have only applied the ANM analysis to the pocket of the protein molecule. H is basically comprised of $N \times N$ super-elements of size 3×3, as follows:

$$H = \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1N} \\ H_{21} & & & H_{2N} \\ \vdots & & & \vdots \\ H_{N1} & & & H_{NN} \end{bmatrix} \tag{2}$$

where, the ij^{th} super-element of H $(i \neq j)$ is:

$$H_{ij} = \begin{bmatrix} \frac{\partial^2 V}{\partial X_i \partial X_j} & \frac{\partial^2 V}{\partial X_i \partial Y_j} & \frac{\partial^2 V}{\partial X_i \partial Z_j} \\ \frac{\partial^2 V}{\partial Y_i \partial X_j} & \frac{\partial^2 V}{\partial Y_i \partial Y_j} & \frac{\partial^2 V}{\partial Y_i \partial Z_j} \\ \frac{\partial^2 V}{\partial Z_i \partial X_j} & \frac{\partial^2 V}{\partial Z_i \partial Y_j} & \frac{\partial^2 V}{\partial Z_i \partial Z_j} \end{bmatrix} \tag{3}$$

here, V represents the potential energy of the elastic system. The following equation represents the elements of the H_{ij} matrix for the ANM.

$$\frac{\partial^2 V}{\partial X_i \partial Y_j} = -\gamma (X_j - X_i)(Y_j - Y_i)/d_{ij}^2 \tag{4}$$

and elements of the diagonal super-element H_{ii} can be given by the following equations:

$$\frac{\partial^2 V}{\partial X_i^2} = -\gamma \sum_j (X_j - X_i)^2/d_{ij}^2 \tag{5}$$

$$\frac{\partial^2 V}{\partial X_i \partial Y_i} = -\gamma \sum_j (X_j - X_i)(Y_j - Y_i)/d_{ij}^2 \tag{6}$$

Here, d_{ij} is the Euclidean distance between C_i and C_j atoms, γ is a constant assumed as 1, and X_i, Y_i, Z_i represents the x, y, and z coordinates of the i^{th} C atom. Note, that we use the cut-off distance as $18\mathring{A}$ while computing the Hessian matrix. To simplify. cut-off distance basically signifies that if d_{ij} is greater than the cut-off distance we ignore the existence of the elastic spring in between those C atoms.

After computing the Hessian matrix, we calculate the eigenvalues and eigenvectors of the Hessian matrix. Moreover, here, each eigenvector represents one normal mode or at a rudimentary level simply one fluctuation into a different structure. Now, we only take the eigenvector corresponding to the maximum eigenvalues since that eigenvector corresponds to the maximum fluctuation in the stable protein structure. Additionally, we also multiply that eigenvector with the corresponding eigenvalue to scale it. Finally, we truncate or pad the obtained eigenvector for each pocket structure in the training set to make the input shape $(1, 256)$ before feeding it to the proposed architecture.

B. Proposed Architecture

The base version of the proposed deep-learning architecture is described in Fig. 1. As shown, the proposed architecture consists of four major elements, 1) 3D Convolution (part of CNN), 2) Cross-Attention mechanism [27], 3) Expand block, and 4) Fully connected (FC) neural network.

Several layers of neural networks can fit very highly complex nonlinear functions. Nevertheless, these networks cannot scale much when the input is of high dimensions. Henceforth, We have used 3D convolution to capture high-dimensionality in our input. The CNN architecture, however, may differ based on many things. Henceforth. it is a good practice to adopt architectures that have been proven to perform well in various applications, such as Resnet [28], VGGnet [29], Squeezenet [30]. Since in [11] it has been discussed that squeeze net architecture tends to perform better for this task, we use the squeeze net architecture for our practice. The expand block shown the Fig. 1 is based on the squeeze net architecture mentioned above. The details of the expand block are explained in Fig. 2. Although, it has been proven that squeeze net architecture tends to perform better in this task, expand block helps to capture the inter-related features of the protein and ligand structures. Since we are feeding both the protein features and ligand features individually to the architecture, the expand block helps to capture the codependency of each structure.

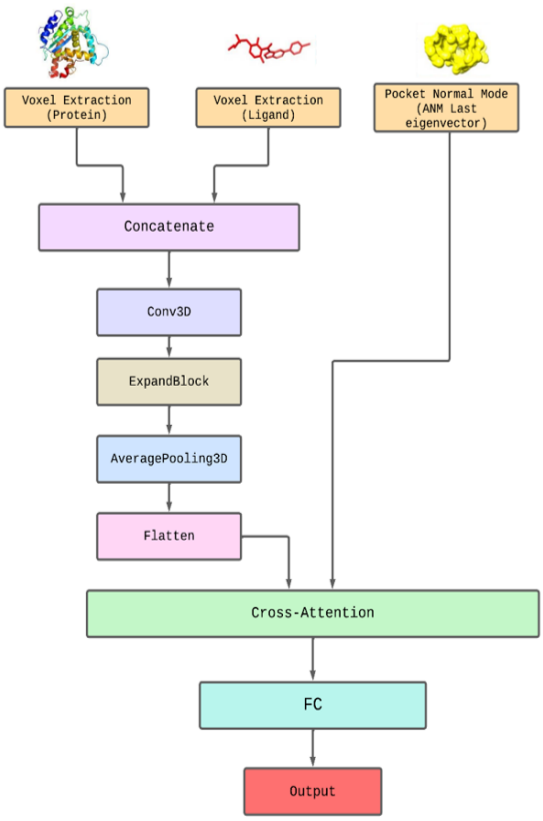

Fig. 1: Proposed deep learning architecture to capture the dynamicity of protein structure during binding with ligands.

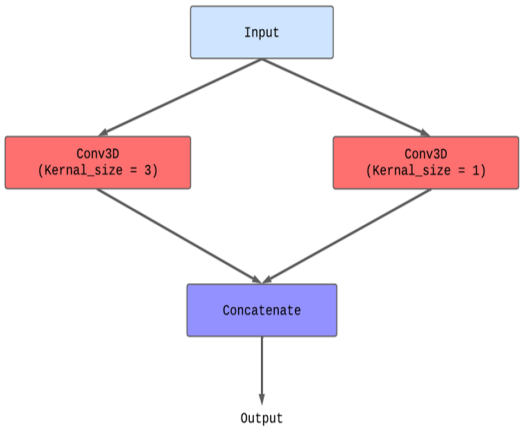

Fig. 2: Expand block description.

To capture, the structural dynamicity of the pocket structure with the existing protein-ligand combined features obtained after the flatten block we apply the cross-attention mechanism. This technique enables the model to learn the effect of maximum structural change of the pocket and capture how it affects the overall binding affinity. Cross-attention mechanism is explained in Fig. 3. For further Details of the attention mechanism please refer to [27]. However, in brief, the vectors Query (Q), Key (K), and value (V) are basically, the weights learned by the cross-attention layer during the training phase.

Finally, the output of the attention block is followed by a

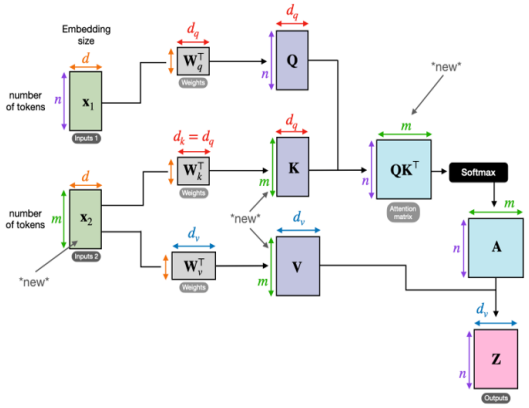

Fig. 3: Cross-Attention block description.

fully connected. This layer serves the purpose of learning the non-linear complex relationship between the extracted feature and the target variable. This helps the model to learn more intricate patterns and make more accurate predictions. Note, in general, we have used the Relu activation function for all the layers except the output layer, for which we have used linear activation since predicting the binding affinity is a regression problem.

IV. Results

In this section, we discuss the results of our model. Like previous studies, we use the Root Mean Square Error (RMSE) and Pearson's correlation coefficient (R) to describe our model accuracy. Both RMSE and R are computed based on the following equations:

$$RMSE(y,\hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \tag{7}$$

$$R(y,\hat{y}) = \frac{\sum_{i=1}^{n}(y_i-\bar{y})(\hat{y}_i-\bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i-\bar{\hat{y}})^2}} \tag{8}$$

where y_i and $\hat{y}_i$ corresponds to the experimental and predicted affinity i. The RMSE represents the ability of the model to correctly predict a small enough prediction range where experimental affinities lie, and R measures the performance of the method when ranking complexes linearly.

In general, we have trained two variants of our proposed models. The only difference is the number of hidden layers in the fully connected network. In the first variant (V_1), the total number of trainable parameters is 755,789 and in the second variant (V_2) the total number of trainable parameters is 1,851,761. The Figs. 5, 6, and 7 corresponds to the V_2 while Fig.4 represents the results for V_1.

Fig. 4 represents the correlation plot for actual (experimental) v/s predicted binding affinity. For V_1 the obtained RMSE and R are 2.468, and 0.2949, respectively. As shown, the red line represents the linear relation between the actual and predicted values. In general, the closer this red line is to the diagonal line $y = x$, the better the model performance. As the $R \to 1$, the red line becomes more aligned to the diagonal. Additionally, to the

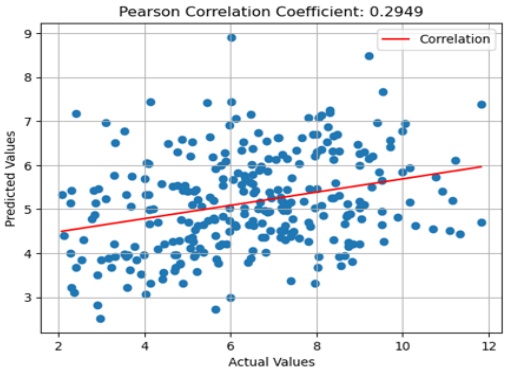

Fig. 4: Correlation plot for predicted v/s actual values for variant V_1

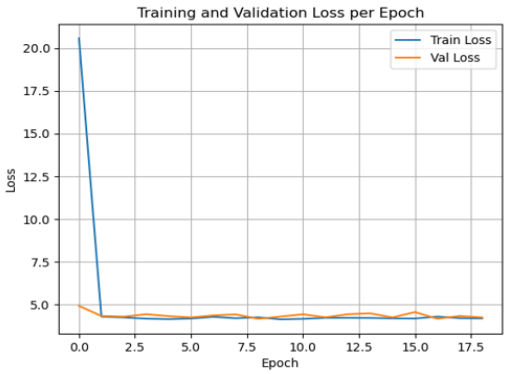

Fig. 5: Training v/s Validation loss per epoch

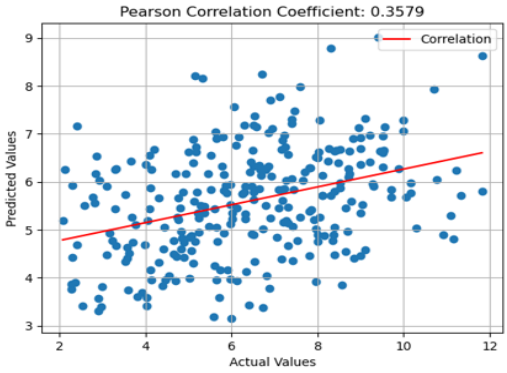

Fig. 6: Correlation plot for predicted v/s actual values for variant V_2 with optimizer amsgrad = False

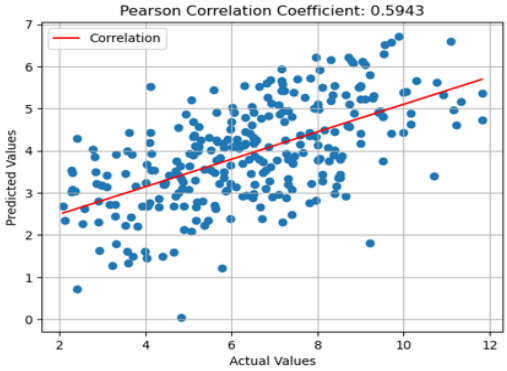

Fig. 7: Correlation plot for predicted v/s actual values for variant V_2 with optimizer amsgrad = True

RMSE and R, we have also computed the Mean absolute error (MAE), and Max error to observe the performance of our model on different scales. For V_1, we got the MAE = 1.9741 and Max error = 7.1239. To increase the performance of the model V_1, we increased the number of trainable parameters in variant V_2. Note, that V_1 is the model obtained in the first iteration of hyper-parameter tuning, so in general always a model needs to be tuned a little to achieve better performance.

In Fig. 5, we have plotted the training v/s validation loss per epoch for V_2. As shown, the model was trained for 18 epochs with the Adam optimizer. Note that while training the amsgrad [31] was kept true, this enables the model to converge more quickly compared to normal convergence. Also, as can be seen in Fig. 5, the minimum observed training and validation losses are 4.1252 and 4.1619, respectively. As both the training and validation loss are almost similar after the final epochs, it means that the model complexity or in general number of trainable parameters could be increased a lot. Additionally, the training loss remains almost constant after two epochs as well, implying that the optimizer might end up in local minima. So that issue also needs to be addressed.

Both Fig. 6, and Fig. 7, are results for V_2. However, the only difference between them is the Adam optimizer's amsgrad value, which is False and True for Fig. 6, and Fig. 7, respectively. Note, that with the change in just the optimizer, we got considerably better results. The RMSE and R obtained when the optimizer amsgrad is false are 2.268 and 0.3579, respectively. The calculated MAE is 1.7464 and the Max error is 6.3347. As seen, there is a significant improvement in every performance measure for V_2 compared to V_1. With amsgrad = True, we are getting the best results for our model, i.e., RMSE = 2.059 and R = 0.5943, with other measures being MAE = 1.5324, and Max error = 6.1332. Table II describes the overall performance of our model compared to other models existing in the literature. Note that the training data used to train our model and others is different. The table just shows the further potential of our model, with some hyperparameter tuning and more training data.

V. Conclusion and Future Work

This work has presented a novel deep learning architecture, KATNet, and a new feature representation technique based on the anisotropic network model (ANM) elastic model. The model provides support to a more realistic scenario as compared to previous research, which does not consider fluctuations of protein molecules while calculating the protein-ligand binding affinity value. Nonetheless, the proposed method is generic and can also be applied with any other elastic model analysis instead of ANM, which could better capture the protein fluctuations; during the feature extraction phase. Through experiments on the PDBbind dataset, we verify the effectiveness of our model. However, we still need to work on our model architecture to improve its performance, so as to accelerate the development

TABLE II: Our model comparison v/s the existing DL models in the literature

Model	**Correlation R**	**RMSE**
K_{deep} [11]	0.82	1.27
RF-score [32]	0.80	1.39
X-score [8]	0.66	1.71
cyScore [33]	0.65	4.13
saCNN [13]	0.865	1.117
Ours	0.5943	2.059

of new drugs.

The future extension of this work could be the incorporation of a bi-directional LSTM layer after the cross-attention layer because as for the current model, we are observing some information loss in between the cross-attention and fully connected layer, as the fully connected layer is not able to capture the spatial dependency of the feature. Adding an LSTM layer subsequently connected with a fully connected layer will solve that problem Additionally, we would like to train our model on more data. Since, as of now, we are only training our model on 3,849 complexes, training on more data could also help a lot in boosting the model's performance. Furthermore, the model complexity could also be increased more as described in the previous section. It is important to study and expand this research area since it has direct application in boosting the efficiency and rate of the drug-development process.

REFERENCES

[1] E. Kobayashi, A. Jin, H. Hamana, K. Shitaoka, K. Tajiri, S. Kusano, S. Yokoyama, T. Ozawa, T. Obata, A. Muraguchi *et al.*, "Rapid cloning of antigen-specific t-cell receptors by leveraging the cis activation of t cells," *Nature Biomedical Engineering*, vol. 6, no. 7, pp. 806–818, 2022.

[2] A. Volkamer, S. Eid, S. Turk, S. Jaeger, F. Rippmann, and S. Fulle, "Pocketome of human kinases: prioritizing the atp binding sites of (yet) untapped protein kinases for drug discovery," *Journal of chemical information and modeling*, vol. 55, no. 3, pp. 538–549, 2015.

[3] A. A. Zarrin, K. Bao, P. Lupardus, and D. Vucic, "Kinase inhibition in autoimmunity and inflammation," *Nature Reviews Drug Discovery*, vol. 20, no. 1, pp. 39–63, 2021.

[4] G. Manning, D. B. Whyte, R. Martinez, T. Hunter, and S. Sudarsanam, "The protein kinase complement of the human genome," *Science*, vol. 298, no. 5600, pp. 1912–1934, 2002.

[5] P. A. Jänne, N. Gray, and J. Settleman, "Factors underlying sensitivity of cancers to small-molecule kinase inhibitors," *Nature reviews Drug discovery*, vol. 8, no. 9, pp. 709–723, 2009.

[6] D. S. Goodsell and A. J. Olson, "Automated docking of substrates to proteins by simulated annealing," *Proteins: Structure, Function, and Bioinformatics*, vol. 8, no. 3, pp. 195–202, 1990.

[7] G. Jones, P. Willett, R. C. Glen, A. R. Leach, and R. Taylor, "Development and validation of a genetic algorithm for flexible docking," *Journal of molecular biology*, vol. 267, no. 3, pp. 727–748, 1997.

[8] R. Wang, L. Lai, and S. Wang, "Further development and validation of empirical scoring functions for structure-based binding affinity prediction," *Journal of computer-aided molecular design*, vol. 16, pp. 11–26, 2002.

[9] H. Wang, "Prediction of protein–ligand binding affinity via deep learning models," *Briefings in Bioinformatics*, vol. 25, no. 2, p. bbae081, 2024.

[10] Z. Cang and G.-W. Wei, "Topologynet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions," *PLoS computational biology*, vol. 13, no. 7, p. e1005690, 2017.

[11] J. Jiménez, M. Skalic, G. Martinez-Rosell, and G. De Fabritiis, "K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks," *Journal of chemical information and modeling*, vol. 58, no. 2, pp. 287–296, 2018.

[12] M. M. Stepniewska-Dziubinska, P. Zielenkiewicz, and P. Siedlecki, "Development and evaluation of a deep learning model for protein–ligand binding affinity prediction," *Bioinformatics*, vol. 34, no. 21, pp. 3666–3674, 2018.

[13] Y. Wang, Z. Qiu, Q. Jiao, C. Chen, Z. Meng, and X. Cui, "Structure-based protein-drug affinity prediction with spatial attention mechanisms," in *2021 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE, 2021, pp. 92–97.

[14] Y. Li, M. A. Rezaei, C. Li, and X. Li, "Deepatom: A framework for protein-ligand binding affinity prediction," in *2019 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE, 2019, pp. 303–310.

[15] H. E. Manoochehri, A. Pillai, and M. Nourani, "Graph convolutional networks for predicting drug-protein interactions," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019, pp. 1223–1225.

[16] D. Jones, H. Kim, X. Zhang, A. Zemla, G. Stevenson, W. D. Bennett, D. Kirshner, S. E. Wong, F. C. Lightstone, and J. E. Allen, "Improved protein–ligand binding affinity prediction with structure-based deep fusion inference," *Journal of chemical information and modeling*, vol. 61, no. 4, pp. 1583–1592, 2021.

[17] F. Hu, Y. Hu, J. Zhang, D. Wang, and P. Yin, "Structure enhanced protein-drug interaction prediction using transformer and graph embedding," in *2020 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE, 2020, pp. 1010–1014.

[18] H. Na, T.-L. Lin, and G. Song, "Generalized spring tensor models for protein fluctuation dynamics and conformation changes," *Protein Conformational Dynamics*, pp. 107–135, 2014.

[19] J. R. López-Blanco and P. Chacón, "New generation of elastic network models," *Current opinion in structural biology*, vol. 37, pp. 46–53, 2016.

[20] B. H. Lee, S. Seo, M. H. Kim, Y. Kim, S. Jo, M.-k. Choi, H. Lee, J. B. Choi, and M. K. Kim, "Normal mode-guided transition pathway generation in proteins," *PloS one*, vol. 12, no. 10, p. e0185658, 2017.

[21] H. X. Kondo and Y. Takano, "Analysis of fluctuation in the heme-binding pocket and heme distortion in hemoglobin and myoglobin," *Life*, vol. 12, no. 2, p. 210, 2022.

[22] J. Gabel, J. Desaphy, and D. Rognan, "Beware of machine learning-based scoring functions on the danger of developing black boxes," *Journal of chemical information and modeling*, vol. 54, no. 10, pp. 2807–2815, 2014.

[23] C. Kramer and P. Gedeck, "Leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets," *Journal of chemical information and modeling*, vol. 50, no. 11, pp. 1961–1969, 2010.

[24] P. J. Ballester and J. B. Mitchell, "Comments on "leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets": Significance for the validation of scoring functions," pp. 1739–1741, 2011.

[25] J. Jiménez, S. Doerr, G. Martínez-Rosell, A. S. Rose, and G. De Fabritiis, "Deepsite: protein-binding site predictor using 3d-convolutional neural networks," *Bioinformatics*, vol. 33, no. 19, pp. 3036–3042, 2017.

[26] A. R. Atilgan, S. Durell, R. L. Jernigan, M. C. Demirel, O. Keskin, and I. Bahar, "Anisotropy of fluctuation dynamics of proteins with an elastic network model," *Biophysical journal*, vol. 80, no. 1, pp. 505–515, 2001.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[30] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[31] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.

[32] P. J. Ballester and J. B. Mitchell, "A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking," *Bioinformatics*, vol. 26, no. 9, pp. 1169–1175, 2010.

[33] Y. Cao and L. Li, "Improved protein–ligand binding affinity prediction by using a curvature-dependent surface-area model," *Bioinformatics*, vol. 30, no. 12, pp. 1674–1680, 2014.