Encoding Protein-Ligand Interactions: Binding Affinity Prediction with Multigraph-based Modeling and Graph Convolutional Network.

Nedra Mekni (0000-0002-3490-1179)[*1,2,3], Hosein Fooladi (0000-0002-3124-2761)[1,3,4], Ugo Perricone (0000-0002-2181-2468)[2,§], Thierry Langer (0000-0002-5242-1240)[1,§]

[1] Department of Pharmaceutical Sciences, Division of Pharmaceutical Chemistry, University of Vienna, Josef-Holaubek-Platz 2, 1090 Vienna, Austria; [2]Molecular Informatics Group, Drug Discovery Unit, Fondazione Ri.MED, 90128 Palermo, Italy; [3]Vienna Doctoral School of Pharmaceutical, Nutritional and Sport Sciences (PhaNuSpo), University of Vienna, 1090 Vienna, Austria; [4]Christian Doppler Laboratory for Molecular Informatics in the Biosciences, Department of Pharmaceutical Sciences, University of Vienna, 1090 Vienna, Austria

***For correspondence:**
nmekni@fondazionerimed.com (N.M.)

[§]These authors contributed equally to this work

Abstract Machine learning models are employed to enhance the speed and provide novel insights in drug discovery due to their demonstrated effectiveness in predicting properties of small molecules like pKa, solubility, and binding affinity. These approaches accelerate drug discovery by helping researchers efficiently identify, prioritize, and optimize compounds. Nonetheless, when investigating properties that depend on the interaction between a ligand and its corresponding protein, a compelling need arises to incorporate the protein counterpart information within the models. Recently, graph neural networks (GNNs) have been developed to incorporate 3D structural information to improve our understanding of the underlying protein-ligand interactions. However, incorporating 3D information into GNNs is not always straightforward. To address the challenge, we introduce a model called InterGraph, which models the protein-ligand interaction as topological multigraphs. By leveraging a topological representation, InterGraph offers a comprehensive approach to a graph representation of the intricate spatial organization and connectivity patterns within protein-ligand systems. We introduce interaction spheres that assign varying edge densities, capturing the proximity-based influence of interactions. This approach enables us to capture the characteristics of the interaction network, filtering out the ones that are beyond 9 Å from the ligand since they are not considered relevant or established. Finally, we trained the model using a ligand binding dataset from PDBbind and tested it on a hold-out test set, achieving an RMSE value of 1.34. Our findings have demonstrated the power of the multigraph to encode the importance of close interactions, a factor that is relevant in the context of binding affinity. On average, our model accurately predicts binding affinity values for several protein-ligand complexes and exhibits higher accuracy for hydrolase, lyase, and families of proteins involved in mediating protein-protein interactions. Additionally, the Intergraph method displayed sensitivity to the binding mode when compared to a set of complexes that had undergone redocking calculations.

Introduction

Protein-ligand interactions are critical in numerous biological processes, and understanding how their binding influences biological processes is at the core of drug discovery and development. Binding affinity is a thermodynamic property that describes the strength of such interactions, providing valuable insights into

the likelihood of successful binding and the potency of a drug candidate [1]. Since experimental determination of the binding affinity is costly and time-consuming, computational methods are largely adopted because they can successfully prioritize drug candidates from extensive compound libraries more efficiently, accelerating drug discovery and development. Free energy calculations based on molecular dynamics are often employed; however, they come with a high computational cost, requiring substantial time to obtain reliable results [2, 3]. The greatest challenge lies in the intricate interplay between protein and ligand molecules, involving diverse molecular interactions and conformational changes, which limit the achievement of precise predictions [4]. Therefore, overcoming these challenges and developing efficient computational methods for accurate binding affinity prediction remains an active area of research [5]. Graph convolutional neural network (GCN) architectures have recently gained significant traction in the field [6–9]. Researchers have achieved exciting results by capitalizing on the inherent graph structure and the powerful learning capabilities of GCNs [10, 11]. This progress underscores the versatility and potential of GCN architectures in pushing the boundaries of predictive modeling in molecular sciences [7, 12–14]. Nevertheless, integrating the three-dimensional (3D) structure of the protein-ligand system into GCNs may not always be straightforward.

Present approaches characterize protein-ligand interactions using ligand molecular graphs, wherein each node encodes the contacts as features. These interactions are acquired by considering proximity-based contacts and applying a defined cutoff distance[7, 9]. This work presents a novel approach to model the intricate structure of a protein-ligand interaction occurring across various distances. Features of both the ligand and the protein are included, creating a topological representation of the interactions. Specifically, our approach uses a multigraph-based methodology, which proves to be highly advantageous in capturing the importance of close molecular interactions that determine binding affinity. The multigraph is constructed based on a PDB file where the atoms are represented as nodes, while the edges describe the Euclidean distances between the ligand and the protein atoms within three distance ranges [15, 16]. Consequently, this leads to forming three subgraphs within the main graph, with each subgraph defined by an area referred to as the "interaction sphere". The edge density within each interaction sphere captures the spatial proximity relationships and emphasizes the significance of closer interactions. Specifically, some spheres demonstrated a higher edge concentration, indicating an increased likelihood of interaction between protein atoms closer to the ligand. Conversely, a low edge density is observed in spheres that enclose atoms farther away, implying a reduced impact on the ligand's binding affinity.

As demonstrated, our approach involves incorporating distance features into multigraph reasoning models, which are inherently non-Euclidean [17]. These models primarily establish relations among graph elements and often overlook the significance of distance features, which, in contrast, play a crucial role in protein-ligand binding interactions. Nevertheless, graph convolutional neural networks (GCNs) are highly effective at capturing intricate patterns in the graph's structured data, and they showed remarkable results in predicting binding affinity [18, 19]. Therefore, we exploit the strengths of GCNs and incorporate the distance features to enhance their capabilities further.

The PDBbind dataset was used to train the model, specifically the PDBbind v.2020 refined set [20]. The refined set comprises non-covalent co-crystal structures with a resolution lower than 2.5 Å and an R-factor lower than 0.250. Binding affinity values are expressed as pKi and pKd. We trained and evaluated the model on a holdout set which showed a mean absolute error (MAE) of 1.34 and a Spearman correlation coefficient (S_ρ) of 0.66. We also performed a 10-fold cross-validation to assess the robustness and reliability of the results. Concluding our discussion, we present an analysis of the protein families where our model demonstrated higher accuracy.

This study aims to contribute to the research on binding affinity prediction using deep learning. We also proposed a novel approach to conceptualize protein-ligand interactions as multigraphs, promoting a valid approach that includes both entities (protein and ligand) involved in the binding process. Furthermore, the model can serve as pre-trained model that can be further fine-tuned as needed to cater to specific target proteins, enabling accurate and reliable predictions.

Results & Discussion

In this section, we present the results of our study aimed at predicting binding affinity from protein-ligand complexes. We will start by discussing our approach for modeling macromolecule complexes using graph representations. Then we will continue to describe the generation and optimization of the graph neural network model. Finally, we showcase the performance and robustness of the model on holdout sets.

Encoding close interactions in protein-ligand complexes using high edge density

As described in the method section, we construct a graph using pairwise atomic distances. Graphs were generated at the atomic level, representing protein and ligand atoms as nodes. Figure 1 illustrates the schematic representation. The distance thresholds essentially acted as guiding parameters. They defined a range around each ligand atom, delimiting which atoms fell within the proximity of the ligand atoms. This spatial region, defined by each threshold, creats what we refer to as an "interaction sphere." This interaction sphere conceptually represents the proximity within which potential molecular interactions are considered relevant and possible. This inherent variation in edge density (number of edges divided by the volume of the sphere) across different interaction spheres, becoming the focus of our analysis. The analysis of edge density was performed by calculating the number of edges present within each sphere for every ligand atom. These calculations offer valuable insights into the prevalence and distribution of connections occurring within distinct distance intervals around each ligand atom. Figure 2 illustrates the density analysis performed on the PDB ID 5E89. The ligand under investigation consists of 80 atoms, and for each atom, the edge density within its corresponding interaction sphere was computed. Notably, each ligand atom was associated with three interaction spheres aligned with a specific distance threshold. We can observe that the edge density decreases by increasing the threshold. For instance edge density for the 3Å threshold is higher than the 6Å threshold, and so on for all ligand atoms.

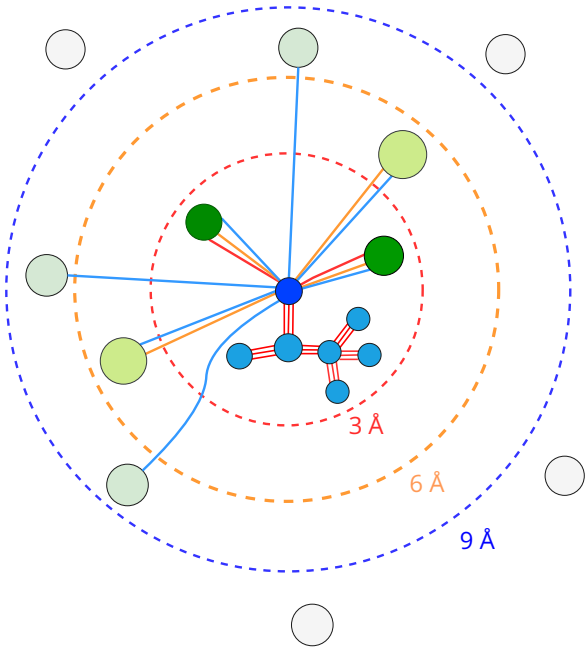
The radial distribution function of our analysis is discrete, with specific intervals defining the radii (0-3 Å, 3-6 Å, and 6-9 Å).

A remarkable aspect is how edge density acts as a structural feature, revealing the local environment of the protein-ligand complex. The closer the interaction (i.e., lower cut-off threshold), the higher the edge density and the higher the importance. Since closer interactions have a more profound impact on binding affinity, this information is embedded to help the network to focus on one of the most crucial aspects of the interaction process. Moreover, incorporating edge density in the graph representation gives the GCN spatial awareness. We intend to give the model a clearer picture of how interactions are distributed throughout the protein-ligand complex. By directly integrating this information into the graph representation, the GCN can learn to assign varying levels of importance, emphasizing edges that represent the denser and more influential interactions, especially those in closer proximity.

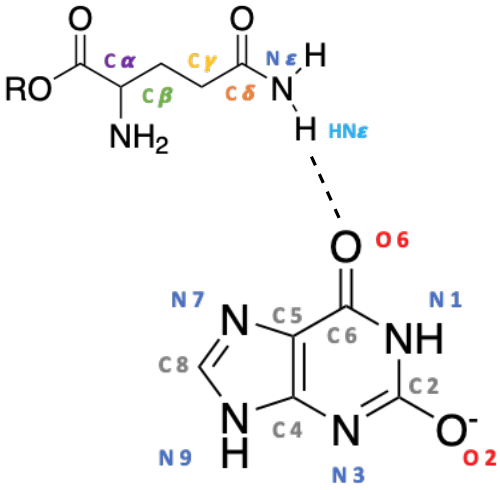
Protein-ligand interactions graph shows predictive power on an independent test set

To ensure improved model performance and generalization, we closely monitored the learning process on both the training and validation sets.

In Figure 3, we report the training and validation learning curve, which presents the loss (Y-axis) as a function of the training time (X-axis). The learning curve plot illustrates how the loss metric (MAE) evolves as the model undergoes training iterations. MAE was continuously monitored during the training process for both the training and validation sets. The plot indicates that the MAE on the training set decreases steadily throughout the training, as indicated by the blue line. Meanwhile, the orange line shows that the MAE on the validation set reaches the minimum value of 2.61. Upon examination, the plot indicates a good fit of the model, as evidenced by both the training and validation loss decreasing and stabilizing with a minimal difference between their values. Throughout the epochs, this close alignment of the training and validation lines reveals a consistent level of the model's performance and generalizability. Moreover, we observed that further training did not yield significant improvements, and the model did not overfit or underfit.



(a)



(b)

Figure 1. (a) This figure visually represents a protein-ligand topological interaction graph. The nodes (atoms) in the graph are depicted as filled circles, with blue circles representing the ligand atoms and green circles representing the protein atoms. The shading of the green circles reflects the proximity of the atoms to the ligand, with darker circles indicating closer atoms and lighter circles representing farther atoms. The connections between protein and ligand atoms (edges) are shown as lines, and distance thresholds of 3Å (red line), 6Å (orange line), and 9Å (blue line) are used to determine the proximity and to connect these atoms. The edges were drawn from a single reference atom shown in dark blue to simplify the overall representation, but the same concept was applied to each ligand atom to create the entire graph. This approach resulted in the creation of three interaction spheres for each ligand atom, represented by the dashed circles within the distance thresholds of 3Å (red), 6Å (orange), and 9Å (blue). Using multiple distance thresholds results in a multigraph with different edge densities. The nodes closer together have a higher edge density, and the nodes farther apart have a lower edge density. The full grey circles in the graph denote atoms excluded from the graph because they did not meet the set distance thresholds. **(b)** This example shows the interaction between a small molecule and the GLN amino acid within a protein. The atom type annotations next to each atom act as identifiers, giving essential details about their connectivity and chemical identity. In the case of protein atoms, it also indicates whether they belong to the amino acid's side chain or backbone. Specifically, the interaction involves a hydrogen bond between the protein's amide hydrogen (H) atom and the sixth ligand's carbon-oxygen double bond (C=O). On the protein side, the $N\epsilon$ notation specifies the nitrogen (N) atom in the amino acid's side chain. As for the ligand, the numbering system designates the positions of atoms within the molecule

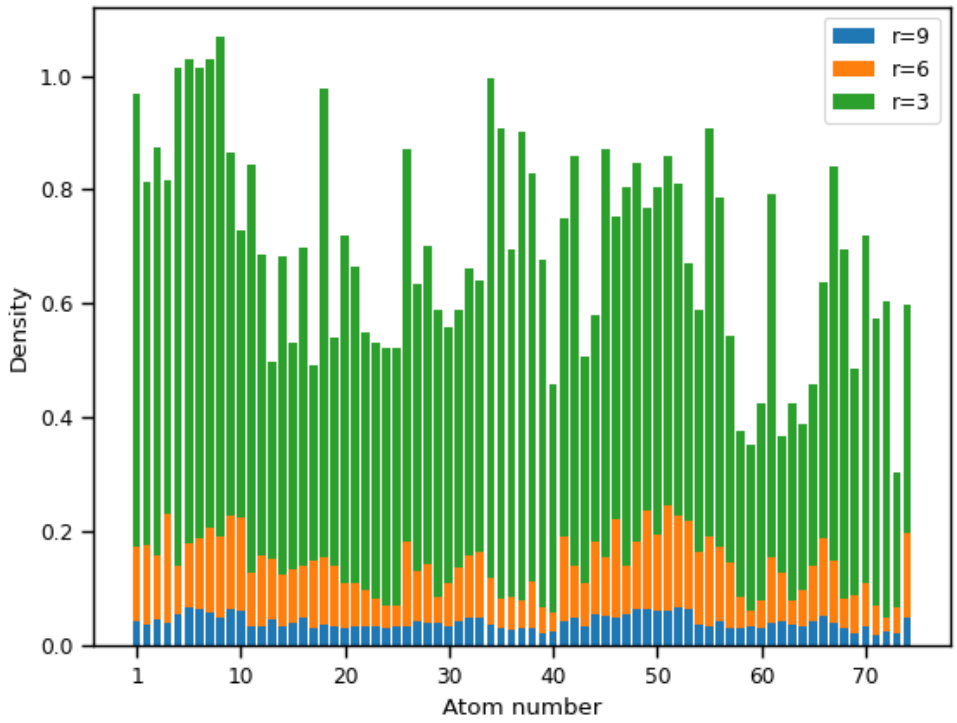
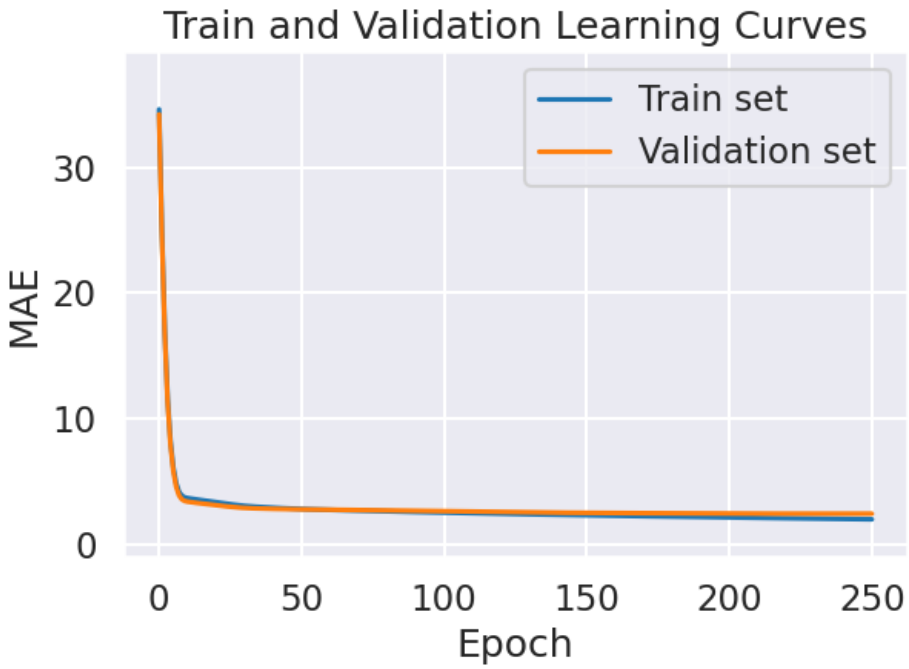


Figure 2. Discrete radial distribution function of edge density. This plot presents the edge density analysis of ligand atoms (PBD id 5E89). The bars represent calculated densities within interaction spheres of $r = 3$, $r = 6$ and $r = 9$ Å.



Training and validation performance: mean absolute error (MAE) on validation set

Figure 3. In the top panel of the graph, the model's performance on the training and validation sets is displayed over a range of epochs. The mean absolute error (MAE) metric evaluates the model's performance. The plot provides essential insights into the neural network's performance, indicating that further training may result in only minimal improvements in the model's performance on the validation set.

Investigating model generalization ability and rank capacity in predicting binding affinities

Evaluating the trained models is critical to assess accuracy and reliability beyond its training data. In the context of binding affinity prediction, the evaluation was conducted regarding the model's predictive accuracy and ability to rank ligands based on their affinities. We conducted this evaluation using a hold-out set (the test set), which contains data the model hasn't seen during training. The data distribution in the test and training sets (see SI Figure 2) exhibited a homogeneous representation. Mean absolute error (MAE) and Sperman's rank correlation coefficient (ρ) were used as metrics.

MAE indicates the average absolute difference between the predicted and true values and provides insight into the model's accuracy in predicting binding affinity values. The model showed a MAE value of 1.34 pK on the test set, with a 95% % Confidence Interval (CI) ranging from 1.2 to 1.48 pK. In Figure 4, we can observe the kernel density estimates (KDE) of the predicted and reference pK distribution. The peak of the distribution around 0 on the x-axis indicates a close agreement between the expected and calculated values, which indicates good model performance. The bell-shaped curve of the distribution also indicates that the data are uniformly distributed. The relatively narrow CI interval indicates that the model's predictions are consistently close to the true values. This outcome highlights the reliability of the model's predictions, motivating confidence in its performance. The ranking capacity is another desirable feature and was evaluated with the Spearman correlation coefficient (ρ). It evaluated how the predicted and true values are ranked similarly. The coefficient ranges from -1 to 1, where -1 indicates a perfect negative correlation, 0 indicates no correlation, and 1 indicates a perfect positive correlation (when one value increases, the other also increases) [21]. A higher (ρ) signifies that the model has a strong ranking capacity. The coefficient was calculated on the test with a value of 0.66. The encouraging ρ score indicates that the model has effectively learned relevant patterns from the training data, allowing it to provide accurate predictions [22, 23]. A more in-depth examination of the residuals (see Figure 5) reveals a consistent positive correlation. About

Distribution of the difference between the experimental and calculated binding affinity

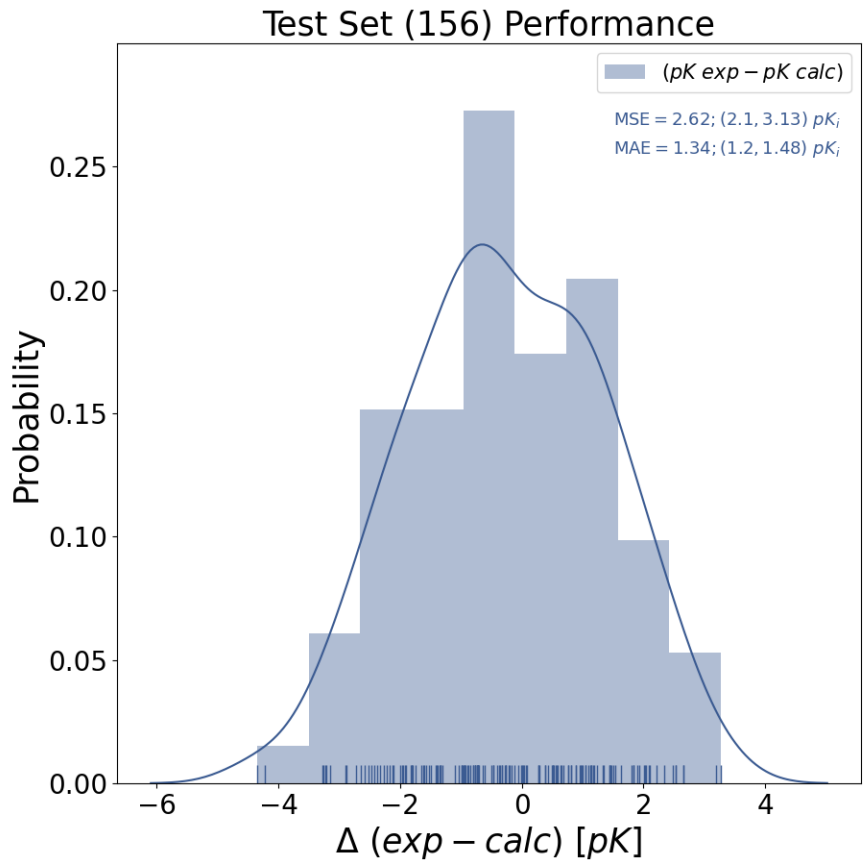


Figure 4. The plot in the figure provides a visual representation of the performance of a predictive model for a test set of 156 samples. The plot's x-axis shows the difference between the expected and calculated values, while the y-axis represents the probability density. This plot is labeled with the calculated mean squared error (MSE), mean absolute error (MAE), and the confidence interval (95%) of the model is given in square brackets.

46% of the points fall within the ±1 error range, and 30.7% of the points lie within the range $1 < x \leq 2$ and $-2 \leq x < -1$, where x represents the error. Nevertheless, although a substantial proportion of predictions fall within the ±1 error range and follow a linear trend, the residuals display some discrepancies from the ideal fit (shown by the diagonal line) in the lower range. This pattern might indicate that the model is lacking in capturing specific trends or relationships for some compounds with very strong binding affinity. The challenge of predicting low values can be attributed to the amount of low-value data in the training set (see SI2)

Exploring protein families with enhanced binding affinity prediction

The analysis of the residual plots on the test set showed deviations between the model's predictions and the actual binding affinity values for some data points. This observation prompted a more in-depth evaluation of the model's predictive abilities, specifically identifying whether these errors were associated with specific protein families. Additionally, we searched for which protein families the GCN demonstrated a higher accuracy.

The examination indicated that the hydrolase protein class is the predominant category within the dataset (Figure 6). Most predictions concerning this family remain within the error range of ±2, except for 7 data points in the tail of the plot. The tail also comprises predictions related to the ligand bound to sugar-binding

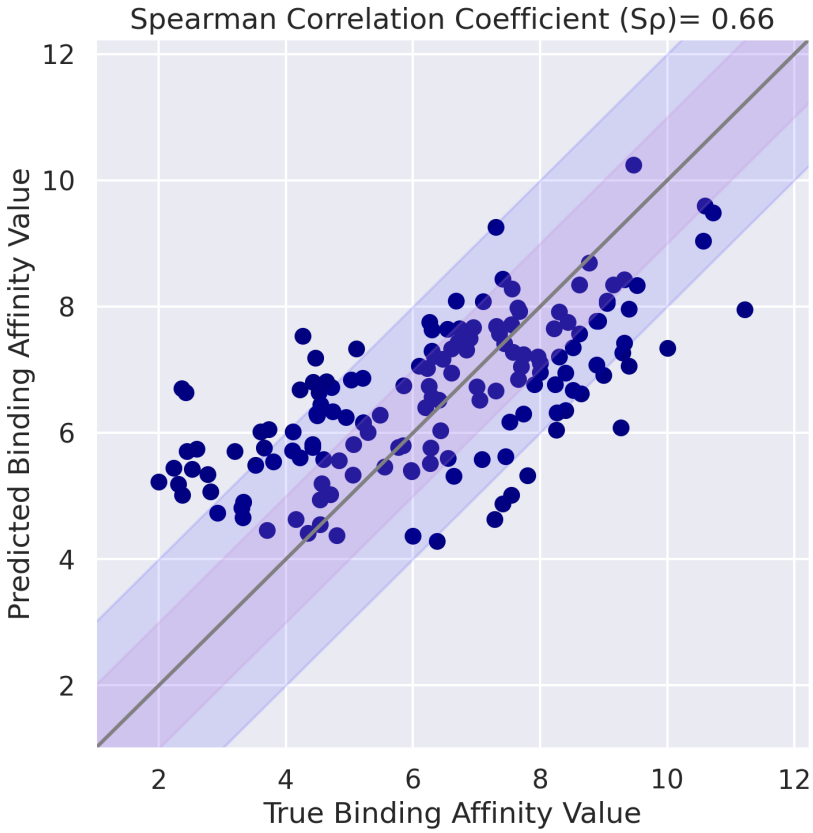


Figure 5. Scatter plot with error bands depicting the relationship between the true (X-axis) and predicted (Y-axis) binding affinity values (blue dots). The gray line represents the ideal linear relationship (y = x). The light purple-shaded region represents the ±1 standard deviation error band, while the lighter blue-shaded region represents the ±2 standard deviation error band. On top, the Spearman Correlation Coefficient ($S\rho$) value is reported.

proteins and transferases. In contrast to other proteins in the set, this group represents instances where the model's predictions showed the most substantial deviation from the actual values.

Taking a broader perspective, the model does not consistently perform poorly for any specific pattern or cluster of proteins. Nonetheless, we garnered insightful findings related to protein families, resulting in improved accuracy. Table 1 reports the number of entries of the most significant protein families, the count of predictions falling within a ±1 error range, and the corresponding percentage of predicted values within this error range[24]. Among the protein families analyzed, the hydrolase family is the predominant one for which the model exhibits good performance, with 26 out of 57 predictions (45.61%) accurately falling within the predefined ±1 error range. The Transferase family also displayed a promising outcome, with 15.30% of predictions falling within this error range. Furthermore, the Lyase family demonstrated high predictive accuracy, with 9 out of 16 predictions (56.25%) aligning within the desired error range. Similarly, the Chaperone and Transcription protein families achieved satisfactory results, with 33.34% and 50.00% of predictions within the ±1 error range, respectively. Included in SI Figure 4 is a comprehensive plot showcasing the data for all other families.

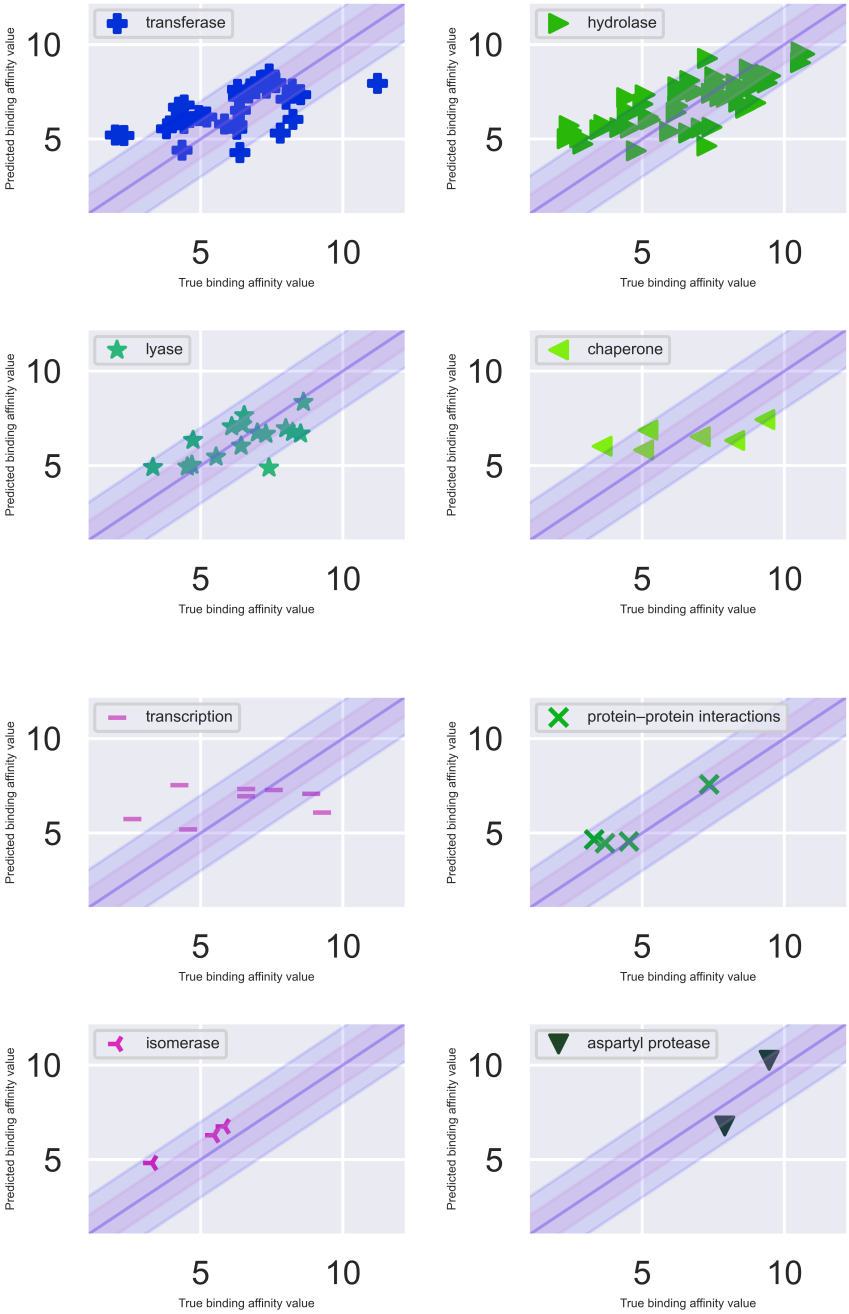


Figure 6. Comparative subplots are showing variation in binding affinity predictions across diverse protein families. Protein categories analyzed include Transferase, Hydrolase, Lyase, Chaperone, Transcription, Protein-protein interactions, Isomerase, and Aspartyl protease.

Table 1. Summary of protein family entries, predictions within ±1 error range

Protein family	N entries	Prediction within ±1 error	Percentage of predicted values in error range
Hydrolase	57	26	45.61
Transferase	30	15	50.00
Lyase	16	9	56.25
Transcription	8	4	50.00
Chaperone	6	2	33.34
Protein-protein interaction	4	3	75.00
Isomerase	3	2	67.67
Aspartyl protease	2	1	50.00

Cross-validation results: assessing model performance and robustness

We performed a 10-fold cross-validation to assess the robustness of our model, subjecting it to varying training and testing data splits. This involved dividing the training set into 10 distinct subsets. In each iteration, a different subset out of the 10 was designated as the test set, while the remaining 9 subsets were utilized for training. Figure 7 presents the mean absolute error (MAE) on the test set for each fold. The minimum and maximum values reported in the box plot are 1.20 and 1.32, respectively, while the mean value is 1.26. The median is 1.27, represented by the horizontal line in the box plot.

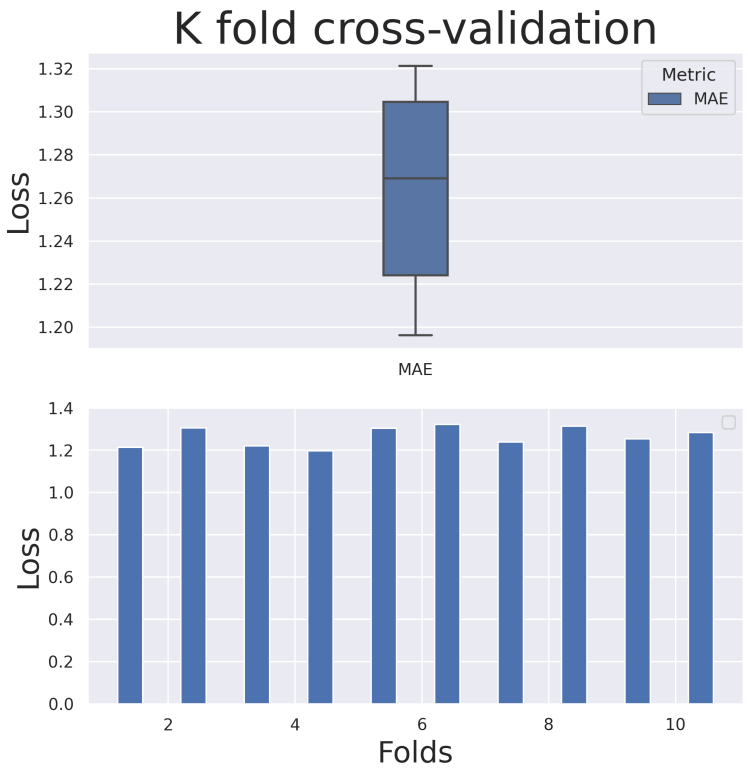


Figure 7. This figure presents two visual representations of the Mean Absolute Error (MAE) values obtained from a 10-fold cross-validation. On the top, a bar plot displays the MAE value for each fold in the cross-validation process. On the bottom, a box plot illustrates the distribution of MAE scores across all ten folds. The box represents the interquartile range (IQR), with a horizontal line inside denoting the median. The whiskers extend from the minimum value to the lower quartile (the start of the box) and from the upper quartile (the end of the box) to the maximum value.

In line with our strategy of predicting binding affinity through protein-ligand interactions, we also calculated the Root Mean Squared Error (RMSE) and Pearson correlation coefficient. This allowed us to bench-

mark our results against documented outcomes in the literature that employed the GCN architecture. Notably, the reference for comparison originates from the CASF-2016 set [25] and is referred here as GCNNET PLIG [7]. The comparison can be made because CASF-2016 is representative of the PDBBbind dataset [7, 20, 26]. Table 2 provides the results, computed as the mean of individual point predictions for each data point within the test set. The GCNNet-PLIG model [7] showcased a Pearson correlation coefficient of 0.76, indicating a meaningful and positive linear relationship between its predicted and actual values. In contrast, the InterGraph model demonstrated a slightly lower but comparable coefficient of 0.60. Although our model displayed a relatively weaker correlation (ρ), it is important to highlight that it is still positive and not random.

Table 2. Table compares the performance of two models, namely GCNNet-PLIGs and InterGraph. The evaluation metrics used for comparison are the Pearson correlation coefficient (ρ) and the Root Mean Square Error (RMSE).

Model	ρ	RMSE
GCNNet-PLIGs	0.76	0.80
InterGraph	0.60	1.59

Assess the robustness and sensitivity of our model when applied to docked poses

Our study contributes to the growing body of research aiming to develop deep-learning methods for predicting binding affinity. While classical structure-based methodologies have proven valuable tools in accelerating the drug discovery process, it is essential to acknowledge that such methods have inherent limitations. In particular, we refer to the constrained conformational sampling and reliance on approximated scoring functions, which can yield results that deviate from the biological reality [27].

Therefore, an accurate binding affinity prediction strictly depends on the correctness of the binding mode, and developing models that can distinguish the correct ones when predicting the binding affinity is required. Accordingly, we examined the model's sensitivity to the binding mode by analyzing the correlation between the model predictions and the pose accuracy across a series of docked complexes [28]. The investigation started with the cognate docking of ligands into their corresponding receptors (see Section Molecular docking calculations). As a first filter, we excluded docking outcomes with RMSD larger than 2 Å, a standard threshold for filtering out less accurate binding poses [29]. We calculated interaction multigraphs from the set of not filtered-out complexes, resulting in a total of 3784.

We applied the trained model to the docked dataset, resulting in a Mean Absolute Error (MAE) of 2.8 and a Spearman correlation coefficient (S_ρ) of 0.52. Model performance slightly decreased on the docked complexes compared to the experimental ones. This outcome can be foreseen, as slight interaction deviations can lead to observable effects.

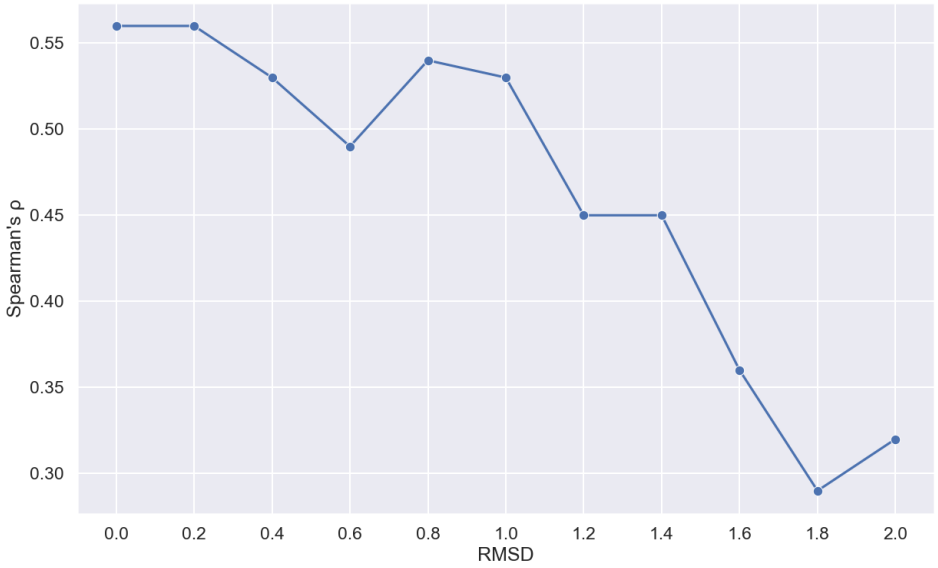


Figure 8. Relationship between the accuracy of the resulting pose (RMSD expressed in Å), and Spearman's rank correlation coefficient (Spearman's $S\rho$)

Our evaluation continued as a comparative analysis of our model's rank capacity and docking accuracy. We observed a correlation between the positive Spearman correlation coefficient ($S\rho$) and the accuracy of the docking outcomes. Our findings suggest that the model exhibits a heightened sensitivity to pose accuracy, as evidenced by the decreasing trend in Spearman correlation ($S\rho$) as the degree of error in pose prediction from docking increases (Figure 8).

Conclusion

Predicting accurate binding affinity is a significant challenge in the early drug discovery and development stages. Researchers possess a comprehensive understanding of both the capabilities and limitations of the existing tools. While they remain a powerful resource, there is a pressing need to develop more robust methods. As a result, in recent times, scientific literature has significantly expanded on AI-based approaches for binding affinity prediction. Such development has paved the way for new techniques that have the potential to address the limitations of traditional methods, whether on their own or combined. Inspired by the growing interest in including the protein counterpart in AI approaches, we developed the InterGraph model, a deep-learning approach to predict binding affinity from protein and ligand complexes. In our implementation, protein-ligand complexes are modeled as 3D topological multigraphs. The design of the protein-ligand multigraph involved a multi-threshold-based approach, which generated interaction spheres around each ligand atom, specifically to enclose proximity-based contacts. The resulting multigraphs were used to train a four-layer graph convolutional neuronal network. Notably, node features, as atom types, were retrieved directly from the PDB files to ensure a consistent and accurate representation of the molecular structure without relying on external atom typing references. Furthermore, InterGraph provides researchers with the flexibility to explore, allowing them to change distance thresholds and node attributes. This adaptability empowers researchers to tailor the model according to specific datasets or research objectives, enhancing its versatility. Our findings highlight the potential of InterGraph in capturing the complex relationships between proteins and ligands, leading to good predictions of binding affinities across multiple protein targets (MAE = 1.34; $S\rho$ = 0.64). Despite being trained on diverse protein families, it exhibited superior accuracy in categories such as Transferase, Hydrolase, Lyase, Chaperone, Transcription, Protein-protein interactions, and Isomerase. This makes it suitable as a pre-trained model that can facilitate reinforcement learning algo-

rithms in building upon its existing knowledge and fine-tuning it for the specific target categories mentioned above. Finally, we have shown that the model exhibits a remarkable sensitivity to the binding pose, highlighting its ability to discern slight changes in molecular interactions. This feature is of particular significance as it addresses a known limitation in conventional docking scoring functions, offering a more accurate binding affinity prediction. In conclusion, while binding affinity prediction remains an open challenge, InterGraph showed to be a valid approach that embraces the complexity of binding interactions and offers promising results. Moving forward, further advancements, including incorporating more diverse and comprehensive datasets and features, will likely contribute to the refinement and robustness of InterGraph.

Code and data availability

The code and data used in this study are openly available at the following link https://github.com/NedraMekni/InterGraph

Author Contributions

Conceptualization: NM ; Methodology: NM, HF ; Software: NM ; Investigation: NM ; Writing–Original Draft: NM ; Writing–Review&Editing: NM, HF, TL, UP; Funding Acquisition: UP ; Resources: TL ; Supervision: TL, UP.

Acknowledgments

We are grateful to Dr. Marcus Wieder for his support throughout the research process. His invaluable insights and guidance were crucial for the development of this work.

References

[1] **Perozzo R**, Folkers G, Scapozza L. Thermodynamics of Protein–Ligand Interactions: History, Presence, and Future Aspects. Journal of Receptors and Signal Transduction. 2004; 24(1-2):1–52. https://doi.org/10.1081/RRS-120037896, doi: 10.1081/RRS-120037896, pMID: 15344878.

[2] **Deflorian F**, Perez-Benito L, Lenselink EB, Congreve M, van Vlijmen HW, Mason JS, Graaf Cd, Tresadern G. Accurate prediction of GPCR ligand binding affinity with free energy perturbation. Journal of Chemical Information and Modeling. 2020; 60(11):5563–5579.

[3] **Wan S**, Sinclair RC, Coveney PV. Uncertainty quantification in classical molecular dynamics. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2021; 379(2197):20200082. https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2020.0082, doi: 10.1098/rsta.2020.0082.

[4] **Du X**, Li Y, Xia YL, Ai SM, Liang J, Sang P, Ji XL, Liu SQ. Insights into Protein–Ligand Interactions: Mechanisms, Models, and Methods. International Journal of Molecular Sciences. 2016; 17(2). https://www.mdpi.com/1422-0067/17/2/144, doi: 10.3390/ijms17020144.

[5] **Wang W**, Sun B, Liu D, Wang X, Zhang H. DPLA: prediction of protein-ligand binding affinity by integrating multi-level information. 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2021; p. 3428–3434.

[6] **Jiang D**, Wu Z, Hsieh CY, Chen G, Liao B, Wang Z, Shen C, Cao D, Wu J, Hou T. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. Journal of Cheminformatics. 2021; 13(1):12. https://doi.org/10.1186/s13321-020-00479-8, doi: 10.1186/s13321-020-00479-8.

[7] **Moesser MA**, Klein D, Boyles F, Deane CM, Baxter A, Morris GM. Protein-Ligand Interaction Graphs: Learning from Ligand-Shaped 3D Interaction Graphs to Improve Binding Affinity Prediction. bioRxiv. 2022; https://www.biorxiv.org/content/early/2022/03/07/2022.03.04.483012, doi: 10.1101/2022.03.04.483012.

[8] **Wang K**, Zhou R, Tang J, Li M. GraphscoreDTA: optimized graph neural network for protein–ligand binding affinity prediction. Bioinformatics. 2023 05; 39(6):btad340. https://doi.org/10.1093/bioinformatics/btad340, doi: 10.1093/bioinformatics/btad340.

[9] **Son J**, Kim D. Development of a graph convolutional neural network model for efficient prediction of protein-ligand binding affinities. PLOS ONE. 2021 04; 16(4):1–13. https://doi.org/10.1371/journal.pone.0249404, doi: 10.1371/journal.pone.0249404.

[10] **Alves LA**, Ferreira NCdS, Maricato V, Alberto AVP, Dias EA, Jose Aguiar Coelho N. Graph Neural Networks as a Potential Tool in Improving Virtual Screening Programs. Frontiers in Chemistry. 2022; 9. https://www.frontiersin.org/articles/10.3389/fchem.2021.787194, doi: 10.3389/fchem.2021.787194.

[11] **Zhang Z**, Chen L, Zhong F, Wang D, Jiang J, Zhang S, Jiang H, Zheng M, Li X. Graph neural network approaches for drug-target interactions. Current Opinion in Structural Biology. 2022; 73:102327. https://www.sciencedirect.com/science/article/pii/S0959440X2100169X, doi: https://doi.org/10.1016/j.sbi.2021.102327.

[12] **Zhang X**, Gao H, Wang H, Chen Z, Zhang Z, Chen X, Li Y, Qi Y, Wang R. PLANET: A Multi-Objective Graph Neural Network Model for Protein–Ligand Binding Affinity Prediction. bioRxiv. 2023; https://www.biorxiv.org/content/early/2023/02/03/2023.02.01.526585, doi: 10.1101/2023.02.01.526585.

[13] **Zhu F**, Zhang X, Allen JE, Jones D, Lightstone FC. Binding Affinity Prediction by Pairwise Function Based on Neural Network. Journal of Chemical Information and Modeling. 2020 06; 60(6):2766–2772. https://doi.org/10.1021/acs.jcim.0c00026, doi: 10.1021/acs.jcim.0c00026.

[14] **Wang K**, Zhou R, Tang J, Li M. GraphscoreDTA: optimized graph neural network for protein–ligand binding affinity prediction. Bioinformatics. 2023 05; 39(6). https://doi.org/10.1093/bioinformatics/btad340, doi: 10.1093/bioinformatics/btad340, btad340.

[15] **Gutman I**, Estrada E. Topological Indices Based on the Line Graph of the Molecular Graph. Journal of Chemical Information and Computer Sciences. 1996 01; 36(3):541–543. https://doi.org/10.1021/ci950143i, doi: 10.1021/ci950143i.

[16] **David L**, Thakkar A, Mercado R, Engkvist O. Molecular representations in AI-driven drug discovery: a review and practical guide. Journal of Cheminformatics. 2020; 12(1):56. https://doi.org/10.1186/s13321-020-00460-5, doi: 10.1186/s13321-020-00460-5.

[17] **Asif NA**, Sarker Y, Chakrabortty RK, Ryan MJ, Ahamed MH, Saha DK, Badal FR, Das SK, Ali MF, Moyeen SI, Islam MR, Tasneem Z. Graph Neural Network: A Comprehensive Review on Non-Euclidean Space. IEEE Access. 2021; 9:60588–60606. doi: 10.1109/ACCESS.2021.3071274.

[18] **Oloulade BM**, Gao J, Chen J, Lyu T, Al-Sabri R. Graph neural architecture search: A survey. Tsinghua Science and Technology. 2022; 27(4):692–708. doi: 10.26599/TST.2021.9010057.

[19] **Jiang D**, Hsieh CY, Wu Z, Kang Y, Wang J, Wang E, Liao B, Shen C, Xu L, Wu J, Cao D, Hou T. InteractionGraphNet: A Novel and Efficient Deep Graph Representation Learning Framework for Accurate Protein–Ligand Interaction Predictions. Journal of Medicinal Chemistry. 2021 12; 64(24):18209–18232. https://doi.org/10.1021/acs.jmedchem.1c01830, doi: 10.1021/acs.jmedchem.1c01830.

[20] **Su M**, Yang Q, Du Y, Feng G, Liu Z, Li Y, Wang R. Comparative Assessment of Scoring Functions: The CASF-2016 Update. Journal of Chemical Information and Modeling. 2019 02; 59(2):895–913. https://doi.org/10.1021/acs.jcim.8b00545, doi: 10.1021/acs.jcim.8b00545.

[21] **Wissler C**. The Spearman Correlation Formula. Science. 1905; 22(558):309–311. https://www.science.org/doi/abs/10.1126/science.22.558.309, doi: 10.1126/science.22.558.309.

[22] **Li L**, Wang B, Meroueh SO. Support Vector Regression Scoring of Receptor–Ligand Complexes for Rank-Ordering and Virtual Screening of Chemical Libraries. Journal of Chemical Information and Modeling. 2011 09; 51(9):2132–2138. https://doi.org/10.1021/ci200078f, doi: 10.1021/ci200078f.

[23] **Ballester PJ**, Mitchell JBO. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. Bioinformatics. 2010 03; 26(9):1169–1175. https://doi.org/10.1093/bioinformatics/btq112, doi: 10.1093/bioinformatics/btq112.

[24] **Cleves AE**, Johnson SR, Jain AN. Synergy and Complementarity between Focused Machine Learning and Physics-Based Simulation in Affinity Prediction. Journal of Chemical Information and Modeling. 2021 12; 61(12):5948–5966. https://doi.org/10.1021/acs.jcim.1c01382, doi: 10.1021/acs.jcim.1c01382.

[25] **Su M**, Yang Q, Du Y, Feng G, Liu Z, Li Y, Wang R. Comparative Assessment of Scoring Functions: The CASF-2016 Update. Journal of Chemical Information and Modeling. 2019 02; 59(2):895–913. https://doi.org/10.1021/acs.jcim.8b00545, doi: 10.1021/acs.jcim.8b00545.

[26] **Meli R**, Morris GM, Biggin PC. Scoring Functions for Protein-Ligand Binding Affinity Prediction Using Structure-based Deep Learning: A Review. Frontiers in Bioinformatics. 2022; 2. https://www.frontiersin.org/articles/10.3389/fbinf.2022.885983, doi: 10.3389/fbinf.2022.885983.

[27] **Meli R**, Morris GM, Biggin PC. Scoring Functions for Protein-Ligand Binding Affinity Prediction Using Structure-based Deep Learning: A Review. Frontiers in Bioinformatics. 2022; 2. https://www.frontiersin.org/articles/10.3389/fbinf.2022.885983, doi: 10.3389/fbinf.2022.885983.

[28] **Díaz-Rovira AM**, Martín H, Beuming T, Díaz L, Guallar V, Ray SS. Are Deep Learning Structural Models Sufficiently Accurate for Virtual Screening? Application of Docking Algorithms to AlphaFold2 Predicted Structures. Journal of Chemical Information and Modeling. 2023 03; 63(6):1668–1674. https://doi.org/10.1021/acs.jcim.2c01270, doi: 10.1021/acs.jcim.2c01270.

[29] **Castro-Alvarez A**, Costa AM, Vilarrasa J. The Performance of Several Docking Programs at Reproducing Protein–Macrolide-Like Crystal Structures. Molecules. 2017; 22(1). https://www.mdpi.com/1420-3049/22/1/136, doi: 10.3390/molecules22010136.

[30] **Paszke A**, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: *Advances in Neural Information Processing Systems 32* Curran Associates, Inc.; 2019.p. 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[31] **Eberhardt J**, Santos-Martins D, Tillack AF, Forli S. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. Journal of Chemical Information and Modeling. 2021 08; 61(8):3891–3898. https://doi.org/10.1021/acs.jcim.1c00203, doi: 10.1021/acs.jcim.1c00203.

[32] **Trott O**, Olson AJ. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of Computational Chemistry. 2010; 31(2):455–461. https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.21334, doi: https://doi.org/10.1002/jcc.21334.

[33] **O'Boyle NM**, Morley C, Hutchison GR. Pybel: a Python wrapper for the OpenBabel cheminformatics toolkit. Chemistry Central Journal. 2008; 2(1):5. https://doi.org/10.1186/1752-153X-2-5, doi: 10.1186/1752-153X-2-5.

Detailed methods

Data collection and preprocessing

The protein-ligand complexes used in this study were obtained from the PDBbind database, specifically, the refined set, where we obtained the PDB IDs and corresponding binding affinity values expressed as pKi and pKd. To retrieve the three-dimensional structures of these complexes, we utilized the Batch Download service provided by the RCSB Protein Data Bank (PDB) website (https://www.rcsb.org). The PDB files were processed to remove metal ions, co-factors, solvent molecules used for crystallization, and ions in the structures. Complexes containing multiple chains or multiple ligands were excluded from the dataset to ensure a focused and consistent analysis. Hydrogen atoms were added using the `pymolPy3`. The final processed dataset consists of 3,178 unique protein-ligand structures. Each structure is composed of a single chain.

Protein-ligand graph generation and proximity analysis

The implementation involved defining a graph, denoted by G, comprising a set of vertices, V (also called nodes), and a set of edges, E, where each edge connects a pair of vertices. The nodes of the graph represent atoms from either protein or ligand. For each ligand atom, we established connections (edges) with protein atoms located within predefined distance thresholds. The distance thresholds were set at discrete values, specifically 3, 6, and 9 Angstroms. These thresholds were instrumental in determining the extent of interactions between protein and ligand atoms.

Our approach involved a step-by-step creation of edges originating from ligand atoms. The algorithm 1 iterates through all ligand atoms and then checks if it encounters another atom within the specified thresholds of 3,6 and 9Å. If it does, an edge is formed. These thresholds defined the closeness of the interactions among protein-ligand atoms and simultaneously within the ligand atoms. The result of the algorithm is a graph structure that allows for multiple edges between nodes, as exemplified by the ligand atoms and their neighbors. This outcome led to a multigraph framework. Notably, the neural network can distinguish between protein and ligand atoms based on the attributes associated with the respective nodes. The constructed multigraph is three-dimensional (3D). Within this setup, the distance threshold acts as the radius of a sphere, encompassing edges located within the defined distance r from the reference ligand node (atom). Consequently, a distinct set of three interaction spheres emerges for each ligand atom. These interaction spheres vary in their edge density. We calculated the edge densities in each interaction sphere, which offers insights into the arrangement and distribution of interactions across diverse spatial extents. The methodology centers on calculating edge densities within three distinct spheres for each atom within the ligand. Around the origin atom o, we define three concentric spheres with radii of 3, 6, and 9 angstroms.

Definition 1 *Let $G = (V, E)$ be a multigraph where $E = (E', m)$ and*

$$E' \subseteq V \times V, m : E' \to \mathbb{N}$$

Given $o \in A_l$, let E_k^o be the set of edges such that $m(x) = k$ for all $x : (o, v) \in E', v \in V$:

$$E_3^o = \{x | x \in E', m(x) = 3\}$$
$$E_2^o = \{x | x \in E', m(x) = 2\}$$
$$E_1^o = \{x | x \in E', m(x) = 1\}$$

Let $D_{r\text{Å}}^o$ be the edge density within radius r:

$$D_{3\text{Å}}^o = \frac{|E_3^o|}{\frac{4}{3}\pi 3^3}$$
$$D_{2\text{Å}}^o = \frac{|E_2^o|}{\frac{4}{3}\pi 6^3}$$
$$D_{1\text{Å}}^o = \frac{|E_1^o|}{\frac{4}{3}\pi 9^3}$$

Each node in the graph is defined by a bit vector that captures atomic features obtained using the Biopython cheminformatics package. The set of features was implemented by concatenating one-hot-encoded

vectors where each dimension corresponds to a specific feature. In these vectors, the presence of a feature is represented by the value 1, while the absence of a feature is represented by the value 0.

The feature set included:

- **Protein or ligand atom**: This binary feature indicates whether the atom belongs to a protein or a ligand molecule.
- **Atom Type**: This feature captured the specific atom type. To ensure a complete representation, we collected atom type information from all the atom types in the PDB files in our dataset. The atom type column in PDB files serves to specify the particular classification or type of an atom based on its chemical environment and connectivity. As a result, this feature represented the most extended feature vector among the encoded features.
- **Element**: This feature represents the elemental composition of the atom. It allowed us to consider the specific elements present in the molecule, further enhancing our understanding of its chemical composition.
- **The number of Hydrogen atoms**: Hydrogen bonding plays a significant role in molecular interactions, particularly in binding affinity. By considering the number of hydrogen atoms bound to each atom, we aimed to capture the involvement of hydrogen bonding in the affinity towards binding partners

As part of our proposed method, we combined the above features to model protein-ligand interaction and capture relevant contributions to binding affinity. Finally, it has resulted in a 2450-dimensional node feature vector.

Memory optimization through online approach in graph generation

This study used a systematic and practical method to generate multigraphs for each protein-ligand entry. Because of multigraph dimensionality, considerable memory resources are required. The high dimensionality of the system can be attributed to two key factors: the extensive node feature set and the large number of nodes and edges. In our implementation, the node features are represented as vectors using one-hot encoding, resulting in a substantial length of 2450 dimensions. To ensure the accuracy and uniqueness of the feature set, all the features must be stored in memory throughout the graph construction process. Moreover, it is important to acknowledge that constructing these multigraphs is not a single-step process but rather an iterative one. The algorithm iterates through each structure, progressively incorporating features into the evolving graph representation. Due to the significant amount of data and structures involved, managing these multigraphs is memory-intensive. Furthermore, we encountered issues when attempting to save the generated graphs using `"torch.save."`. During the serialization process performed by torch, the graph data is temporarily stored in memory before being written to the disk. This intermediate storage adds an extra memory overhead, impacting overall memory usage.

Unlike a regular graph, there is no maximum limit for the size of the edges' multiset. In our case, an edge can be repeated at most three times within the multiset. This means that given the cardinality of nodes within a multigraph, the maximum size of the set of edges will be $3(n^2)$, indicating a space complexity of $O(n^2)$. In this context, we are examining multigraphs characterized by a node size on the order of 10^3 and a range of edge numbers spanning from 10^6 to 10^7. Intuitively, managing such graphs represents a challenge. We implemented an iterative approach to overcome this hardware limitation that involved constructing the graphs in batches. We began by building the multigraphs using the first batch of data. As we progressed to subsequent batches, we updated the one-hot encoded features of the previously processed $n-1$ batches whenever we encountered new features. This ensured that we maintained the up-to-date encoding of the previous batches while also generating new encoding for the current batch. Specifically, if the feature set of the new batch n (denoted as L) was an improper subset of the feature set from the previous $n-1$ batches (denoted as M), the node's feature vector is not expanded because it already includes the required features. However, if the feature set L is not an improper subset of M, it means that the new batch n introduces new features (denoted as $K = L \setminus M$), and the feature vectors from the previous $n-1$ batches are expanded to include them. Throughout this process, we carefully maintained the order of features to ensure accuracy and consistency across all nodes. We empirically observed that constructing graphs in batches of 200 PDBs at a

time allowed us to process a reasonable number of graphs while avoiding excessive memory consumption. In total, we had 23 batches. Finally, the resulting dataset was loaded from the disk into the main memory for the model training.

Algorithm 1 Graph Construction

Definition 0.1 *Graph* $G = (V, E)$

Input:

- $V = A_p \cup A_l$ where A_l set of ligand atoms, A_p set of protein atoms.
- $d : V \times V \to \mathbb{R}_0^+$ the euclidean distance.
- $d_1, d_2, d_3 \in \mathbb{R}_0^+$

Output:

- $(E, m) : E = \{(u, v) : u \in A_l, v \in V\}, m : E \to \mathbb{Z}^+$

Initialize

- $(E, m), E = \emptyset$

for $u \in A_l$ **do**
  for $v \in V$ **do**
    Compute the distance $d(u, v)$ between atoms u and v

    for $d_i \in \{d_1, d_2, d_3\}$ **do**
      if $d(u, v) \leq d_i$ **then**
        $(E, m) \cup \{(u, v)\}$
      end
    end
  end
end

Training and testing

Test and training were performed using `PyTorch geometric` [30]. The dataset was randomized and split (90:5:5) into training, validation, and test set. Mean squared error (MSE) was used as a cost function between predicted and experimental values. The training was done on the negative base-10 logarithm of Ki, Kd (pKi, pKd). We evaluated the model performance using the validation and test sets. The validation set was crucial in guiding model refinement and optimization, while the test set measured its overall performance and generalizability to new data. We employed two statistical metrics: the Spearman rank correlation coefficient ($S\rho$) and the Pearson correlation coefficient (ρ). These metrics offered distinct insights into different aspects of the model's performance. The Spearman rank correlation coefficient allowed us to assess the ranking of binding affinities, providing insights into the strength and direction of the monotonic relationship. By analyzing $S\rho$, we could evaluate whether the model captured the ranking of binding affinities. On the other hand, the Pearson correlation coefficient measured the linear relationship between the predicted and experimental values. ρ enabled us to evaluate the linear association's strength and direction and the linear trends' characterization in the model's binding affinity predictions.

Molecular docking calculations

We performed docking calculations using AutoDock Vina software [31, 32]. Ligand and protein structures were provided by the PDBbind dataset in SDF and PDB format, respectively.

As Vina works on PDBQT format, we used Open Babel v.2.3.1 [33] to process the PDB file format to PDBQT. Ligands were processed from SDF to PDBQT through the *mk_prepare_ligand.py* script implemented in the

Meeko package (https://github.com/forlilab/Meeko). The docking box was defined from the centroid of the ligand experimental coordinates and extended 25 Ångströms in each dimension (25x25x25). The centroid was calculated using ComputeCentroid, a specific function within rdkit.Chem.rdMolTransforms. Docking calculations were finally performed by setting the exhaustiveness parameter as 32, and for each ligand, the best 5 poses were stored. We employed RDKit's AllChem (http://www.rdkit.org) to evaluate and choose the most accurate docking result. AlignMol function was used to compute the Root Mean Square Deviation (RMSD) between the docked and co-crystallized ligands. Specifically, we retained the lowest docking poses with RMSD values below 2 Å, excluding 824 structures and a final set of 3784 docked complexes.

GCN architecture

The model was built using the PyTorch library with CUDA support and trained on a GPU for accelerated computation. The model's architecture consists of two graph convolutional layers (GCNConv) and a linear classifier that performs a linear transformation. In the first graph convolutional layer, the input features undergo a graph convolution operation, generating 32 output features. These output features are then normalized using a batch normalization layer. The second graph convolutional layer takes the normalized features from the previous layer and applies another graph convolution operation. Similar to the first layer, the output features are normalized using batch normalization. The rectified linear unit (ReLU) activation function is applied. This activation function ensures that negative values are set to zero while positive values remain unchanged. Following the graph convolutional layers and the ReLU activation, a global max pooling operation is performed. This operation extracts the maximum value across the feature dimension for each node in the graph. In the final step, the global max-pooled features are processed by a linear classifier that maps them to a single scalar value. It is important to note that the scalar value is always positive, with a lower bound of 0.

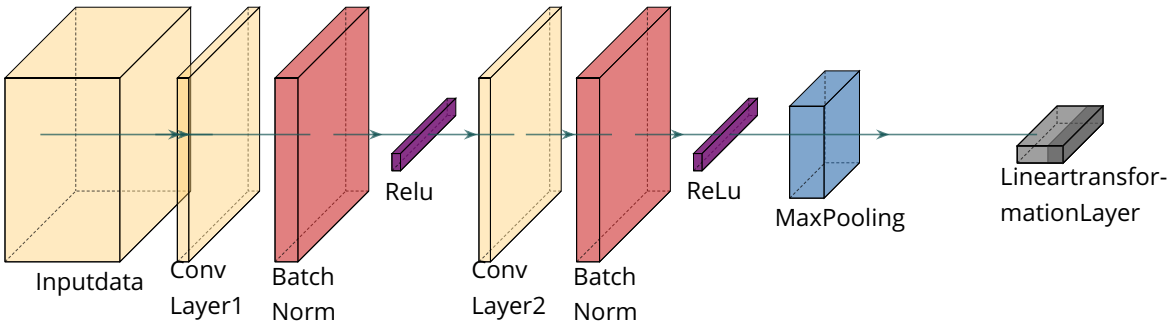


Figure 9. Graph Convolutional Neural Network (GCN) Architecture Scheme.

Cross Validation

To ensure the model's robustness and ability to generalize accurately to new data, we performed 10-fold cross-validation on the training set, constituting 10% of the dataset. Data was randomly partitioned in each fold into 10 subsets, with 9 subsets utilized for training and the remaining subset for validation. This procedure was repeated 10 times.