# Graph Attention Neural Networks Improving Molecular Docking Rank with Protein-Ligand Contact Maps

**Glauco E. Lima**[1], **Simone Q. Pantaleão**[1], **Isabelle A. Pereira**[1], **Ana L. Scott** [1]

[1]Computational Biology and Biophiysics Laboratory – Universidade Federal do ABC (UFABC) – Postal code 09280-560 – Santo André – SP – Brazil

***Abstract.*** *Predicting the binding mode and affinity of small molecules to proteins is key to understanding their interaction. Empirical scoring functions are commonly used by docking programs, but accurately predicting them remains challenging. Docking programs can generate ligand conformations similar to crystallographic structures, yet scoring functions often struggle to identify the correct pose. This study employs Graph Attention Networks (GAT) to learn ligand-protein contact information and re-rank docking poses. Using PDBbind-core data, docking calculations with AutoDock Vina generate binding poses, evaluated by contacts and RMSD. Close contacts are mapped using BINANA, and bipartite graphs are created with atomic descriptors using RDKit.*

## 1. Introduction

Research in the area of protein-ligand docking explores possible binding positions of the ligand on a specific molecular target and also attempts to predict the binding affinity. That can help to understand the molecular interaction between these macromolecules and small ligands (substrate, drugs, natural compounds), which is crucial for scientists involved in drug design and discovery nowadays. Identifying new drug candidates requires an understanding of the most important chemical elements that guide ligand-protein interactions in relevant biological targets [Meng et al. 2011]. The docking scoring function is used to rank the best binding orientations, but the highest-ranked solution is not always the position that reproduces the best orientation [Ramírez and Caballero 2018].

Therefore, as effective as they may be, docking software has considerable limitations such as a lack of confidence in the ability of scoring functions to provide accurate binding energies. This stems from the fact that some intermolecular interaction terms are difficult to predict accurately, such as the solvation effect and entropy change. Moreover, some intermolecular interactions are rarely considered in scoring functions, despite being proven to be significant [Meng et al. 2011]. Among the various computational tools used in bioinformatics, machine learning techniques have proven to be especially useful for the analysis and interpretation of biological data [Greener et al. 2022]. It is a computational approach that allows a program to learn from data without being explicitly programmed to do so. Recent studies have demonstrated the effectiveness of deep learning methods, including Graph attention neural networks (GATs) [Velickovic et al. 2017], in improving the accuracy of docking predictions. For instance, researchers [Yuan et al. 2021] utilized GATs to model the affinity between proteins and ligands. This study aims to develop a tool for reclassifying docking poses generated by docking algorithms [Dias et al. 2008]. The tool utilizes a GAT network to process bipartite graphs representing the contact map between a ligand and a receptor [Imambi et al. 2021]. Atoms in these graphs are represented as nodes with associated atomic descriptors, allowing the network to

evaluate the probability of correctness for each pose. To achieve this, we use the PDBbind-core data [Su et al. 2018] and generate docking poses with AutoDock Vina [Eberhardt et al. 2021a]. These poses are then evaluated based on contacts and RMSD. Close contacts are mapped using BINANA [Durrant and McCammon 2011], and bipartite graphs are constructed with atomic descriptors obtained from RDKit [Landrum 2013].

## 2. Methods and details

In Figure 1, we present a flow diagram with the steps of the methodology proposed. The dataset used is PDBbind [Liu et al. 2015], which provides experimentally validated binding structures of protein-ligand complexes. Each sample in the PDBbind core dataset is prepared automatically using PDBFixer to correct protein structures [Eastman et al. 2017]. It identifies and fixes missing residues and atoms, adds hydrogens at pH 7.0, standardizes non-standard residues, and removes unwanted heterogeneous molecules. The box dimensions were determined based on the center of mass of each ligand, with a 10 Å buffer for each dimension. The exhaustiveness was set to 50 [Agarwal and Smith 2023]. Then docking calculations were performed to generate possible binding poses of the protein-ligand complex using the AutoDock Vina software [Eberhardt et al. 2021b]. When building the dataset, the number of docking poses generated for each complex can vary. However, the more poses generated per complex, the greater the imbalance in the distribution of labels. For consistency with existing literature, we opted to employ ten poses for each complex [Plewczynski et al. 2011]. Nevertheless, it is noteworthy that the model, after training, possesses the capability to evaluate any number of poses.

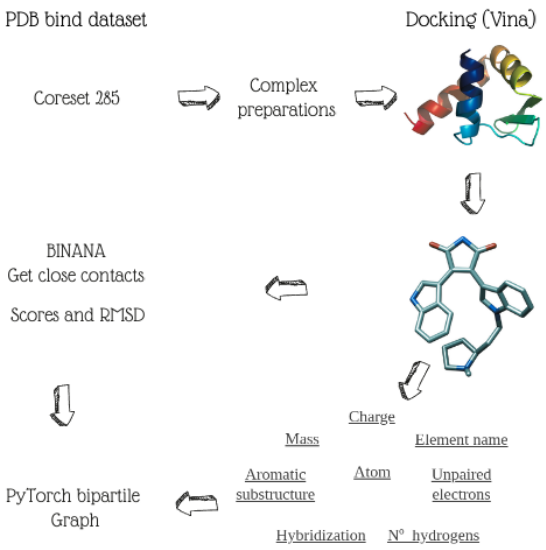



**Figure 1. Workflow to generate the training data**

### 2.1. Dataset

The dataset used for training this model was derived from the PDBind core set, specifically the 2016 release [Su et al. 2018]. This release includes 285 protein-ligand complexes selected from the refined set (v.2016) by applying the following criteria. To reduce

redundancy, proteins exhibiting over 90% sequence similarity were grouped together, from which five representative complexes were selected according to their binding affinity (BA) values. This selection included the complex with the highest BA, the one with the lowest, and three others that represented evenly spaced BA values. Furthermore, each complex's electron density map was examined for quality assurance, and any identical ligands or stereoisomers were omitted from the final dataset [Su et al. 2018]. The variation in electron density map quality between the datasets resulted in a significant difference in results for the proposed model architecture. Consequently, the model exhibits good performance when trained and tested on the core set, but the same does not happen for the refined set, even though the latter possesses higher redundancy. Therefore, the results presented in this study are derived exclusively from the core set.

Table 1 shows the amount of data that we started with and the amount that remained to feed the model. We performed docking for all 285 complexes. During the processing pipeline, several factors reduced the amount of data (number of poses) available. For example:

- Vina Autodock did not generate all 10 expected poses for some complexes.
- Complexes not processed correctly by the libraries used, such as RDKit.
- Descriptors were not properly assigned to the complexes.
- Errors in RMSD calculation (for docking results that were too far off).

The labelling process is rigid, meaning that, in most cases, only one pose per complex is labeled as positive, therefore it's also unbalanced dataset problem [Ganganwar 2012].

**Table 1. Model Performance Metrics**

| | |
|---|---|
| Qty. pdbs docked | 285 |
| Successfully processed graphs | 1948 |
| Class distribution | Incorrect: 1802 (92.5%), Correct: 146 (7.5%) |

## 2.2. Labelling

The data generation process involves a re-docking experiment, in which the protein-ligand complexes have known ligand binding positions. The ligands were removed from their respective binding sites, and docking simulations were subsequently performed to predict the ligand positions. Following that, an evaluation step was undertaken to assess and classify the outcomes into good or poor predictions for each pose [Morrone et al. 2020]. Despite the limited amount of data, we opted to automatically define positive and negative classes. This approach is crucial for ensuring that the labeling technique remains independent of the dataset size. To evaluate redocking results, it's a common practice to use the RMSD, which compares the predicted pose with the experimental result. However, relying solely on RMSD can be insufficient for a few reasons. A predicted pose might show a low RMSD yet form interactions with the protein that differ greatly from those seen in experiments. Conversely, a high RMSD might conceal an accurate binding mode if it preserves the critical interactions but includes a flexible ligand region that is incorrectly positioned, distorting the RMSD [Baber et al. 2009]. Therefore, we adopted a consensus

approach based on the RMSD, calculated using the CalcRMS function available in RDKit [Landrum 2013], along with a graph similarity metric that we developed. To evaluate this similarity between docking results $G_{\text{Dock}} = \{(u_1, v_1), \ldots, (u_n, v_n)\}$ and experimental data $G_{\text{Exp}} = \{(u_1, v_1), \ldots, (u_m, v_m)\}$, where, $(u_i, v_i)$ denotes pairs of atoms, with $u_i$ as an atom from the protein and $v_i$ as an atom from the ligand, such that these atoms are located less than 4 Å apart. We use the following equation (1) to denote it.

$$\text{Similarity Index} = \frac{|\{(u_i, v_i) \in G_{\text{Exp}} \mid (u_i, v_i) \in G_{\text{Dock}}\}|}{|G_{\text{Exp}}|} \tag{1}$$

Here, $|\{(u_i, v_i) \in G_{\text{Exp}} \mid (u_i, v_i) \in G_{\text{Dock}}\}|$ denotes the number of common pairs between $G_{\text{Exp}}$ and $G_{\text{Dock}}$, and $|G_{\text{Exp}}|$ represents the total number of pairs in $G_{\text{Exp}}$.

### 2.3. Data preparation

After obtaining the docking results, the next step was to identify the nearby contacts between the protein and ligand for each complex, using the software BINANA (Bind Analyzer Tool) [Durrant and McCammon 2011]. For each pose: the atom index, atom name, and distance between atoms were collected. Using the RDKit library, several properties of each atom were extracted: i) the atom's formal charge, ii) the element's symbol, iii) the hybridization state, iv) the total number of hydrogen atoms bonded to the atom, v) the number of unpaired electrons, vi) whether the atom is part of an aromatic substructure and the vii) mass [Morrone et al. 2020].

### 2.4. Graph Representation

The interaction within the contact region of the complex protein-ligand was mapped as a graph [Morrone et al. 2020]. Using BINANA, we identified close contacts between receptor and ligand atoms, defined as those within 4 angstroms of each other, according to BINANA's criteria for close contacts [Durrant and McCammon 2011].

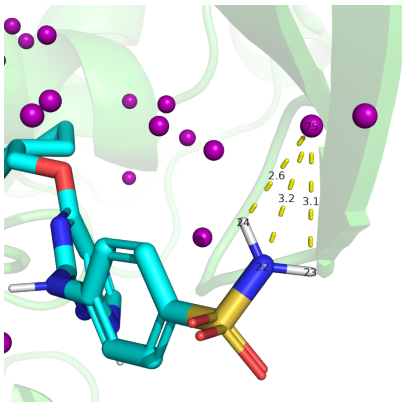



**Figure 2. The BINANA software and a Python script were used to detect and extract close-contact atoms information, where close contact refers to all atoms within 4 Å of each other.**

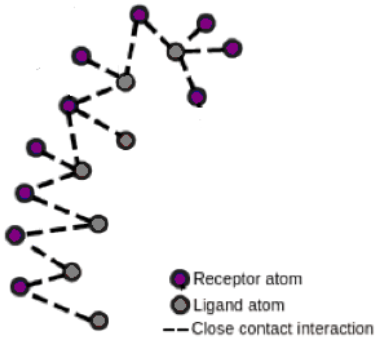



**Figure 3. Using PyTorch, a bipartite graph is created, where atoms serve as the nodes, defined by the atomic descriptors. The distance between atoms is also computed, and the edges represent the close-contact.**

For example, in Figure 2, we have Pose 1 of the 4EOR complex. The purple dots represent receptor atoms that are close to the ligand. When examining the atom with index 75, it appears three times due to the presence of ligand atoms 22, 23, and 24, which are within 4 angstroms of atom 75. With these contacts, a bipartite graph is created in which the atoms are the nodes defined by the atomic descriptors generated using the RDKit.

### 2.5. Model architecture

Molecular data can be represented in three-dimensional spaces as: 3D graphs, 3D surfaces, and 3D voxels [Liu et al. 2023]. However, the latter two approaches exhibit sensitivity to the spatial orientation of molecular complexes and frequently fail to accurately capture the intricate details of atomic bonding. Since molecular complexes consist of protein and ligand molecules, they can naturally be modeled as graphs, where atoms serve as nodes and bonds as edges. Graph neural networks (GNNs) provide an ideal framework for representing such molecular structures, with the advantage of being inherently invariant to changes in orientation of the entry data [Réau et al. 2023].

Attention mechanisms have shown great success across a range of sequence processing tasks, including natural language processing, speech recognition and others [Niu et al. 2021]. On graphs, attention-based models have been developed to generalize the attention operator, assigning different importance to neighboring nodes, which helps to improve predictive performance. Recognizing the strengths of this approach, we developed a model using the graph attention network (GAT) architecture.

The deep learning architecture is represented in figure 4 and begins with a GAT layer, which employs attention mechanisms to process node features and capture relationships. After the initial convolution, the receptor atoms have their representations updated to include information about the neighboring ligand atoms [Velickovic et al. 2017]. The new representation is aggregated using global mean pooling to create a representation that feeds into the fully connected layer [Imambi et al. 2021], which produces a single output representing the probability of correctness of the pose [Morrone et al. 2020].

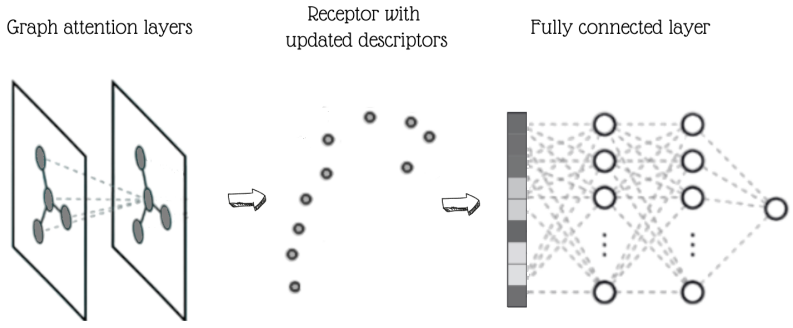



**Figure 4. The bipartite object is fed into the Graph Attention Network (GAT) architecture. The final output is the probability of correctness of the pose, that was represented as a graph.**

The architecture of the model is also illustrated in Table 2, which depicts the input data and the changes in dimensions that occur at each layer.

**Table 2. Model Architecture and Data Flow**

| Layer/Operation | Input Dimensions | Output Dimensions |
|---|---|---|
| **GAT Layer (Conv1)** | $x_s$ and $x_t$ : $[N_{\text{atoms}} \times 7]$ | $[N_{\text{atoms}} \times 100]$ |
| **ReLU Activation** | $[N_{\text{atoms}} \times 100]$ | $[N_{\text{atoms}} \times 100]$ |
| **Global Mean Pooling** | $[N_{\text{atoms}} \times 100]$ | [Batch size $\times$ 100] |
| **Fully Connected Layer** | [Batch size $\times$ 100] | [Batch size $\times$ 1] |

**Layer Descriptions:**

- **GAT Layer (Conv1):** Processes the input features of ligand and receptor atoms, where each atom is represented by 7 features. The output is a 100-dimensional feature vector for each atom, capturing complex interactions.
- **ReLU Activation:** Applies the Rectified Linear Unit activation function element-wise to the output from the GAT layer, introducing non-linearity into the model.
- **Global Mean Pooling:** This layer combines the features of individual atoms to create a single representation for each graph (Batch = 250 graphs). It does this by averaging the feature vectors of all atoms within each graph, resulting in one summarized feature vector per graph.
- **Fully Connected Layer:** Reduces the graph-level features to a single scalar output per graph, which predicts the correctness of the docking pose.

The first layer is a GAT, the input features are represented as matrices $x_s$ and $x_t$, corresponding to the ligand and receptor, respectively. Both matrices have dimensions of $[N_{\text{atoms}} \times 7]$, where $N_{\text{atoms}}$ denotes the number of atoms in each entry and 7 the number of features. Additionally, the first convolutional layer (conv1) includes a hidden layer with a dimensionality of 7. The attention mechanism in the GAT enhances local relationships within the graph, improving node embeddings. To address class imbalance, we adopt a custom loss function, BalancedBCEWithLogitsLoss, which applies a positive class weight of 13.5, this weighted loss helps the model prioritize learning from the minority class during training. The Adam optimizer is used with a learning rate of 0.05 and a weight decay of 0.01 to introduce regularization and avoid overfitting. A PyTorch DataLoader is utilized to efficiently organize the data into mini-batches, streamlining the training process [Imambi et al. 2021]. The argument follow_batch=['x_s', 'x_t'] is employed to ensure that specific node feature sets, remain consistent across batches.

## 3. Results

The dataset is randomly divided into training and validation sets [Yang et al. 2023], aiming to develop a model capable of operating effectively with proteins that exhibit high similarity to those present in the training set, the data were split as follows: 77% allocated for training and 23% reserved for testing. The results presented below include the metrics obtained from evaluating the models on a test set consisting of 448 poses.

**Table 3. Model Performance Metrics**

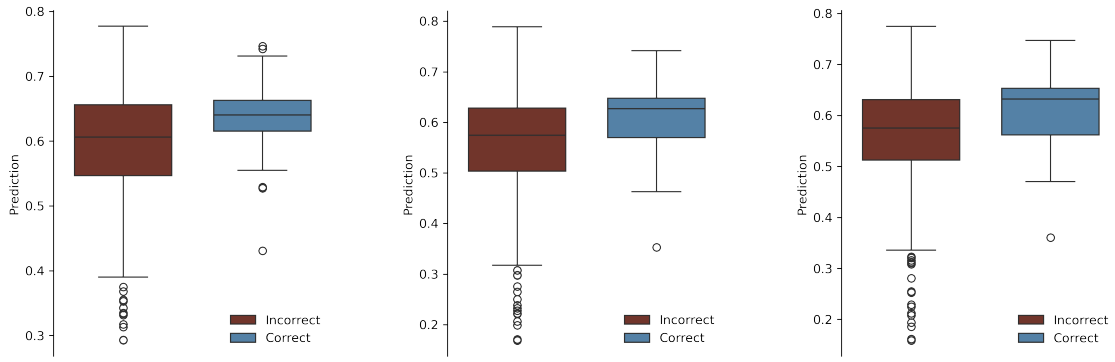
| Model name | Precision | Recall | Auc pr | Auc roc | f1 | Batch | Opt. param |
|---|---|---|---|---|---|---|---|
| model_188 | 0.11 | 0.67 | 0.11 | 0.63 | 0.19 | 200 | f1 |
| model_174 | 0.14 | 0.55 | 0.13 | 0.66 | 0.22 | 250 | f1 |
| model_180 | 0.13 | 0.52 | 0.15 | 0.67 | 0.2 | 250 | recall |
| model_177 | 0.12 | 0.48 | 0.13 | 0.66 | 0.2 | 250 | f1 |
| model_183 | 0.14 | 0.48 | 0.14 | 0.67 | 0.21 | 250 | recall |
| model_182 | 0.13 | 0.42 | 0.13 | 0.66 | 0.2 | 250 | recall |
| model_175 | 0.12 | 0.36 | 0.11 | 0.62 | 0.18 | 250 | f1 |
| model_179 | 0.09 | 0.36 | 0.1 | 0.62 | 0.15 | 250 | recall |
| model_189 | 0.15 | 0.33 | 0.14 | 0.67 | 0.21 | 250 | f1 |
| model_184 | 0.17 | 0.3 | 0.15 | 0.67 | 0.22 | 500 | f1 |

Due to the pronounced class imbalance, the model's effectiveness is significantly influenced by the configuration of the loss function parameters [Li et al. 2021]. In this work, we adapt the Binary Cross Entropy with Logits Loss function [Ruby and Yendapalli 2020], to adjust the class weights and reflect the disparity between classes using the parameter pos_weight [Xiong et al. 2021]. The table 3 shows the ten best results obtained in the experiments, ranked by recall. Each line represents an experiment with a different set of parameters. If a certain type of parameter is not represented in the columns, it is because it is set the same for all models. The F1 score was employed as the criterion for selecting the epoch in which the model should be saved [Yang 2001]. The model was stored whenever there was an improvement in it. We opted not to use the AUC-ROC as the primary metric for model selection, as it is well-documented that this metric can lead to misleading conclusions when applied to imbalanced datasets. Additionally, we also evaluated the AUC-PR (Precision-Recall curve), which is more suitable for evaluating models on imbalanced datasets, as it focuses on the performance of the minority class without being skewed by the majority class's performance. Considering the operational principles of this metric and the observed data distribution, which comprises 1802 incorrect (92.5%) and 146 correct samples (7.5%), any AUC-PR value exceeding the baseline of 0.075 indicates that the model has acquired meaningful patterns from the data [Sofaer et al. 2019]. The results demonstrate a capacity for learning across various trials, though there is some variation. Thus, real performance should be evaluated cautiously considering all results.

When analyzing the results of the best ranked experiments in figure 5, an immediate difference is observed in the distribution of scores assigned by the model to the classes identified as positive and negative. As expected, the model assigned higher probabilities to classes known to be positive. The second relevant aspect is that the model shows superior performance in predicting positive instances while struggling with negative instances, the distribution of negative classes has a high median influenced by the parameter that adjusts for the unbalanced data [Rezaei-Dastjerdehei et al. 2020].

The model that obtained the most satisfactory metrics so far is named $model_{188}$, due to its results for F1 and recall metrics. The loss curve for this model is presented in the Appendix A. The recall metric received special attention, aiming to maximize the model's ability to correctly identify positive cases and minimize the rejection of these

cases.




A - Model 188 B - Model 174 C - Model 180

**Figure 5. Box plots of different models.**

The figure 6 illustrates the probability density profile for both classes. In figure 7, we identify the optimal threshold, which corresponds to the point where Youden's J statistic is maximized at 0.62, balancing precision and recall [Youden 1950]. However, we opted to prioritize higher recall.

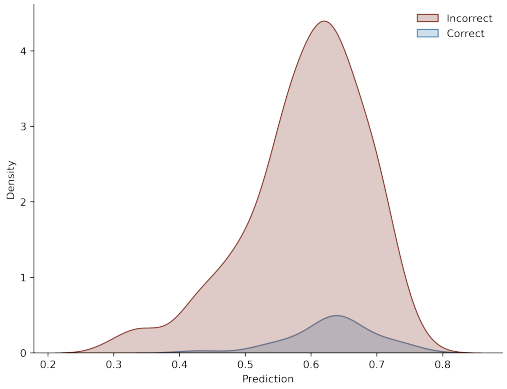



**Figure 6. Distribution of probabilistic predictions made for classes known to be positive and negative.**

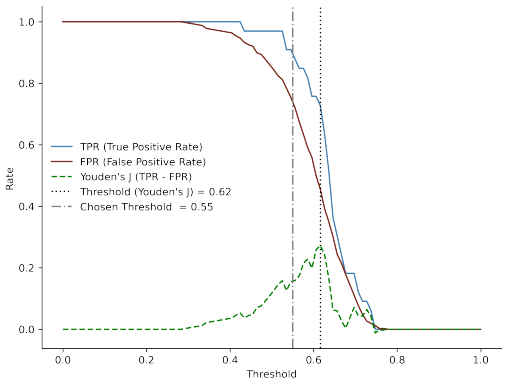



**Figure 7. TTPR (True Positive Rate) and FPR (False Positive Rate) with cut-off Point at 0.55 and Youden's J Value.**

Using the box plot in figure 5-A as a reference, we selected the lower limit of the positive class as our threshold. A visual inspection of the model's box plot indicates that this threshold effectively removes approximately 25% of incorrect data, with only three outliers from the positive class being discarded. As shown in figure 7, the threshold of 0.55 results in an acceptable reduction of false negatives while maintaining a low number of false negatives [Frederick and Bowden 2009]. For comparison, it is visually evident from the box plot in figure 5-A that the median of the negative class is lower than the first quartile of the positive class. This suggests that approximately 50% of the negative cases could potentially be removed if we were willing to sacrifice around 25% of the positive cases, a good result for an unbalanced set.

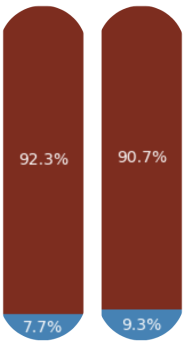



**Figure 8-A**

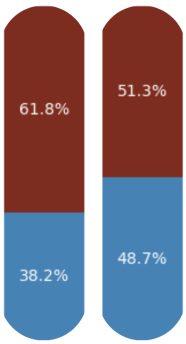



**Figure 8-B**

Figure 8-A illustrates the class distribution after applying the model as a filter to the testing dataset, using a threshold of 0.55. The proportion of positive classes increases from 7.7% to 9.3%, this change is not visually significant due to the class imbalance present in the dataset. However, in figure 8-B, we see that when applying the filter to a subset containing only the poses ranked first by AutoDock Vina, the increase becomes more pronounced, as this subset represents a less imbalanced dataset, increasing from 38.2% to 48.7%.

## 4. Conclusion

The experiments conducted throughout this study provided insights into the performance and challenges associated with the use of Graph Attention Networks (GAT) in the analysis of protein-ligand contact maps represented as bipartite graphs. The developed model showed promising capabilities for interpreting information in the graphs with the core set dataset. When applied to the PDBbind refined set, the model did not achieve the expected performance, even after the selection of subsets closer to the core set profile. An analysis of the literature revealed that the difference was not related to the quality of the proteins' resolution or binding affinity, but rather to the electron density map, which stood out as a crucial parameter for differentiating the datasets [Su et al. 2018]. This result highlights the importance of the electron density map in the analysis and selection of data for model training. Given the limited dataset, calculating useful ranking metrics was not feasible. Additionally, the evaluation of the importance of the "pose rank" descriptor, which is the rank generated by AutoDock vina, revealed a significant limitation in the current model, which showed a tendency to prioritize this descriptor exclusively, ignoring other relevant factors. This challenge was addressed by opting for a model that does not include the "rank" information.

The choice of model was primarily based on the F1 and recall, indicating that the model named $model_{188}$ has achieved the best results so far. The tests suggest that the model functions as an effective filter for eliminating negative cases, saving scientists time. However, to better validate the model's ranking capability, a larger dataset is needed. The results provide a solid foundation for the development of more robust and accurate models, with an emphasis on improving data quality by comprehensively considering descriptors. The findings highlight the necessity of adaptive approaches and balancing techniques to tackle the challenges associated with class imbalance and to enhance the overall performance of the model. We will enhance the model by incorporating atomic

reactivity descriptors derived from Density Functional Theory (DFT) [Orio et al. 2009], which are expected to address the limitations related to electron density maps.

## Data and Software Availability

All the protein-ligand data are available through the PDBbind website at http://www.pdbbind.org.cn. The project model and pipeline are available on GitHub: https://github.com/glauco-endrigo/BindRanker.

## Corresponding Author Information

Email: glauco.endrigo@hotmail.com
ORCID: https://orcid.org/0009-0006-5102-7280

## Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

## References

Agarwal, R. and Smith, J. C. (2023). Speed vs accuracy: effect on ligand pose accuracy of varying box size and exhaustiveness in autodock vina. *Molecular Informatics*, 42(2):2200188.

Baber, J. C., Thompson, D. C., Cross, J. B., and Humblet, C. (2009). Gard: a generally applicable replacement for rmsd. *Journal of Chemical Information and Modeling*, 49(8):1889–1900.

Dias, R., de Azevedo, J., and Walter, F. (2008). Molecular docking algorithms. *Current drug targets*, 9(12):1040–1047.

Durrant, J. D. and McCammon, J. A. (2011). Binana: a novel algorithm for ligand-binding characterization. *Journal of Molecular Graphics and Modelling*, 29(6):888–893.

Eastman, P., Swails, J., Chodera, J. D., McGibbon, R. T., Zhao, Y., Beauchamp, K. A., Wang, L.-P., Simmonett, A. C., Harrigan, M. P., Stern, C. D., et al. (2017). Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology*, 13(7):e1005659.

Eberhardt, J., Santos-Martins, D., Tillack, A. F., and Forli, S. (2021a). Autodock vina 1.2. 0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling*, 61(8):3891–3898.

Eberhardt, J., Santos-Martins, D., Tillack, A. F., and Forli, S. (2021b). Autodock vina 1.2. 0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling*, 61(8):3891–3898.

Frederick, R. I. and Bowden, S. C. (2009). The test validation summary. *Assessment*, 16(3):215–236.

Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47.

Greener, J. G., Kandathil, S. M., Moffat, L., and Jones, D. T. (2022). A guide to machine learning for biologists. *Nature reviews Molecular cell biology*, 23(1):40–55.

Imambi, S., Prakash, K. B., and Kanagachidambaresan, G. (2021). pytorch. *Programming with TensorFlow: solution for edge computing applications*, pages 87–104.

Landrum, G. (2013). Rdkit documentation. *Release*, 1(1-79):4.

Li, M., Zhang, X., Thrampoulidis, C., Chen, J., and Oymak, S. (2021). Autobalance: Optimized loss functions for imbalanced data. *Advances in Neural Information Processing Systems*, 34:3163–3177.

Liu, M., Li, C., Chen, R., Cao, D., and Zeng, X. (2023). Geometric deep learning for drug discovery. *Expert Systems with Applications*, page 122498.

Liu, Z., Li, Y., Han, L., Li, J., Liu, J., Zhao, Z., Nie, W., Liu, Y., and Wang, R. (2015). Pdb-wide collection of binding data: current status of the pdbbind database. *Bioinformatics*, 31(3):405–412.

Meng, X.-Y., Zhang, H.-X., Mezei, M., and Cui, M. (2011). Molecular docking: a powerful approach for structure-based drug discovery. *Current computer-aided drug design*, 7(2):146–157.

Morrone, J. A., Weber, J. K., Huynh, T., Luo, H., and Cornell, W. D. (2020). Combining docking pose rank and structure with deep learning improves protein–ligand binding mode prediction over a baseline docking approach. *Journal of chemical information and modeling*, 60(9):4170–4179.

Niu, Z., Zhong, G., and Yu, H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62.

Orio, M., Pantazis, D. A., and Neese, F. (2009). Density functional theory. *Photosynthesis research*, 102:443–453.

Plewczynski, D., Łażniewski, M., Grotthuss, M. V., Rychlewski, L., and Ginalski, K. (2011). Votedock: consensus docking method for prediction of protein–ligand interactions. *Journal of computational chemistry*, 32(4):568–581.

Ramírez, D. and Caballero, J. (2018). Is it reliable to take the molecular docking top scoring position as the best solution without considering available structural data? *Molecules*, 23(5):1038.

Réau, M., Renaud, N., Xue, L. C., and Bonvin, A. M. (2023). Deeprank-gnn: a graph neural network framework to learn patterns in protein–protein interfaces. *Bioinformatics*, 39(1):btac759.

Rezaei-Dastjerdehei, M. R., Mijani, A., and Fatemizadeh, E. (2020). Addressing imbalance in multi-label classification using weighted cross entropy loss function. In *2020 27th national and 5th international iranian conference on biomedical engineering (ICBME)*, pages 333–338. IEEE.

Ruby, U. and Yendapalli, V. (2020). Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(10).

Schafer, R. W. (2011). What is a savitzky-golay filter?[lecture notes]. *IEEE Signal pro-*

*cessing magazine*, 28(4):111–117.

Sofaer, H. R., Hoeting, J. A., and Jarnevich, C. S. (2019). The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*, 10(4):565–577.

Su, M., Yang, Q., Du, Y., Feng, G., Liu, Z., Li, Y., and Wang, R. (2018). Comparative assessment of scoring functions: the casf-2016 update. *Journal of chemical information and modeling*, 59(2):895–913.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al. (2017). Graph attention networks. *stat*, 1050(20):10–48550.

Xiong, G., Wu, Z., Yi, J., Fu, L., Yang, Z., Hsieh, C., Yin, M., Zeng, X., Wu, C., Lu, A., et al. (2021). Admetlab 2.0: an integrated online platform for accurate and comprehensive predictions of admet properties. *Nucleic acids research*, 49(W1):W5–W14.

Yang, Y. (2001). A study of thresholding strategies for text categorization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 137–145.

Yang, Z., Zhong, W., Lv, Q., Dong, T., and Yu-Chian Chen, C. (2023). Geometric interaction graph neural network for predicting protein–ligand binding affinities from 3d structures (gign). *The journal of physical chemistry letters*, 14(8):2020–2033.

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer*, 3(1):32–35.

Yuan, H., Huang, J., and Li, J. (2021). Protein-ligand binding affinity prediction model based on graph attention network. *Math. Biosci. Eng*, 18(6):9148–9162.

# Appendix A

The loss curve for the $model_{188}$ is illustrated in Figure 9. The red line represents the loss curve that has been smoothed using the Savitzky-Golay filter [Schafer 2011].

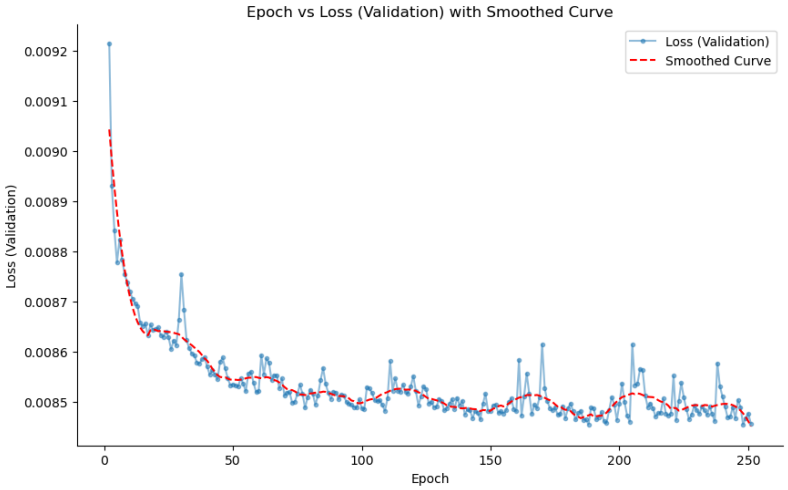



**Figure 9: Loss curve for model 188**