JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

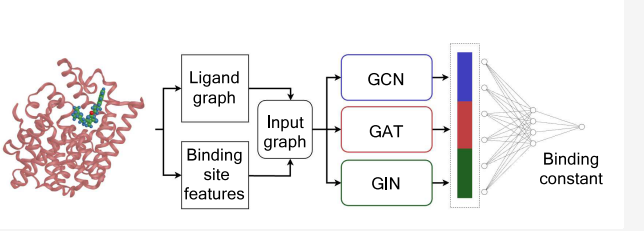
graphLambda: Fusion Graph Neural Networks for Binding Affinity Prediction

Ghaith Mqawass and Petr Popov*

Cite This: J. Chem. Inf. Model. 2024, 64, 2323−2330 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Predicting the binding affinity of protein−ligand complexes is crucial for computer-aided drug discovery (CADD) and the identification of potential drug candidates. The deep learning-based scoring functions have emerged as promising predictors of binding constants. Building on recent advancements in graph neural networks, we present graphLambda for protein−ligand binding affinity prediction, which utilizes graph convolutional, attention, and isomorphism blocks to enhance the predictive capabilities. The graphLambda model exhibits superior performance across **CASF16** and **CSAR HiQ NRC** benchmarks and demonstrates robustness with respect to different types of train-validation set partitions. The development of graphLambda underscores the potential of graph neural networks in advancing binding affinity prediction models, contributing to more effective CADD methodologies.



INTRODUCTION

Early stage drug discovery holds significant opportunities to improve the entire drug discovery cycle and reduce the associated expenses. Indeed, the pervasive 90% failure rate in clinical trials primarily stems from issues originating in early stage drug discovery, high attrition rates across target selection, hit identification, lead optimization, and clinical candidate selection.[1] To effectively search for hit candidates among ultralarge chemical libraries in silico, one may need a fast and accurate binding affinity scoring function.[2] However, developing highly accurate models for binding affinity prediction is still one of the main challenges in computer-aided drug discovery (CADD),[3] largely due to the lack of high-quality data sets and data uncertainty related to experimental conditions and measurement errors.[4] Typically, binding affinity predictors are integrated into the molecular docking pipelines or used to rescore or rerank the docking poses. Most recently, deep learning-based diffusion docking approaches were developed, where one generates, rather than combinatorially samples, docking poses.[5] Thus, binding-affinity-aware generation is a promising direction for scoring function development. The developed scoring functions to approximate the binding affinity can be roughly classified into four types: physics-based, empirical, statistical, and machine-learning scoring functions. We refer a reader to the comprehensive reviews of various types of scoring functions[6,7] and only briefly describe the differences between them below. Physics-based methods are mainly based on force fields[8] and can be adjusted with the solvation energy terms or other physicochemical characteristics.[9,10] Empirical scoring functions[11] combine different terms of various types, including energy, structure, sequence, and other factors that are related to the protein−ligand binding affinity.[12] Statistical scoring functions employ known properties of atoms, such as their size, charge, and hydrophobicity, as well as geometric and structural information about the macromolecular complexes, to derive atom pair interaction potentials from the data sets of protein−ligand complexes.[13] Finally, the machine learning approaches emerged as computationally cheap and effective alternatives to overcome the challenges faced by first-principle methods.[14]

Machine Learning-Based Scoring Functions. Machine learning (ML) approaches can be used to predict the binding affinity due to constantly accumulating biophysical and structural data for protein−ligand complexes. For example, one can calculate features characterizing intermolecular interactions and apply a classical machine learning approach like random forest (RF)[15] or other tree-based regressors[16,17] to derive the predictive models. Another classical machine-learning approach is also used for binding affinity prediction. For example, PESD-SVM[18] is a Support Vector Machine (SVM) model that was built using surface-based descriptors computed by encoding molecular shapes and property distributions on both protein and ligand surfaces as the input features. In another work,[19] the authors used two Gradient Boosting (GB) models for solving the scoring problem: the first model is to predict the binding affinity, and the second

Special Issue: Machine Learning in Bio-cheminformatics

Received: May 22, 2023
Revised: February 5, 2024
Accepted: February 6, 2024
Published: February 17, 2024

boosting model is to learn a confidence interval for a given drug-target pair. Feed-forward neural networks (FNNs) also can be classified as a classical machine learning approach. For example, AEScore uses fixed-length feature vectors that encode the local environment of each atom in a protein or a ligand; then the feature vectors are used as the input for atom-type specific FNNs to predict binding affinity as the output.[20]

Deep Learning-Based Scoring Functions. With the accumulation of binding affinity data, deep learning (DL) methods, including convolutional neural networks (CNNs) and graph neural networks (GNN), have proven to be superior to classical ML approaches. The CNN models typically differ in protein and/or ligand representation: sequence (1D), graph or contact matrices (2D), and structure (3D). For example, DeepDTA[21] is a 1D-convolutional neural network that predicts the binding affinity using protein amino acid sequence and SMILES (Simplified Molecular Input Line Entry System) of a ligand to represent protein−ligand complexes. The model uses two 1D CNN blocks: the first block extracts features from the protein sequence, while the second block extracts features from the SMILES input of the ligand; then, the learned representations are combined and fed to a fully connected layer. Another example is a 2D-CNN model that scores protein−ligand affinity based on the residue-atom contact shells represented as 2D images.[22] Other works rely on molecular representations for 3D CNN, where each atom, atom group, or atom superposition can be represented by a separate channel.[23,24] AK-Score[25] and Kdeep[26] 3D CNN models represent a protein−ligand complex as a voxel grid, where each voxel has several channels including hydrophobic, hydrogen-bond donor or acceptor, aromatic, positive or negative ionizable, and metallic and total excluded volume. However, CNN (especially 3D CNN) approaches have limitations with respect to the number of input channels due to large GPU memory consumption. On the other hand, decreasing the number of input channels may result in a poor representation of the atomic environment, hence, less powerful predicting models.

GNN is an alternative to CNN, which employs a permutation-invariant graph representation of protein−ligand complexes. For example, MGraphDTA[27] relies on multiscale GNN to capture local and global features from the input molecular graph for the ligands and amino acid sequences for the proteins. The extracted features are then combined and used to predict the binding affinity by means of fully connected layers. We previously developed graphDelta,[28] a message-passing neural network (MPNN) model as the binding affinity scoring function; graphDelta represents ligand as an unweighted graph and uses geometric functions that took into account distances and angles between atoms to represent the atomic environment of the protein binding site.

Although various graph neural network architectures have been developed and used in different molecular activity prediction tasks, the benefits of using their combinations have yet to be explored. Inspired by recent advances in graph neural networks, which brought new architecture blocks aiming to empower predictive models, in this work, we have further developed the graphLambda predictive model for the protein−ligand binding affinity using graph convolutional, attention, and isomorphism blocks. The graphLambda model demonstrated superior or on-par performance on various benchmarks and showed robustness with respect to different types of train-validation set partitions.

COMPUTATIONAL METHODS

Data Set. We used the PDBbind data set version 2020[29] comprising a general set of 23,496 experimentally measured binding affinity data (see Supporting Information Figure S1) for different types of biomolecular complexes deposited in the Protein Data Bank (PDB).[30] A refined set of 5316 high-quality protein−ligand complexes (see Supporting Information Figure S2) was used to train the models. A small subset from Comparative Assessment of Scoring Functions (CASF16)[31] of 285 protein−ligand complexes served as a core set for testing the models. Another benchmark data set CSAR HiQ_NRC,[32] comprising 2 subsets of 242 and 297 complexes, was also used for testing the models. To ensure a fair evaluation of the trained models, the test sets included only complexes that have a similarity <0.5 with respect to the training set.

Splitting. The protein−ligand complexes in a data set can be pairwisely similar in terms of protein structures, ligand structures, or protein−ligand interactions. Therefore, the random train-validation split could introduce performance bias, thus yielding overoptimistic results. In order to take the similarity issues into account and test the model generalization ability, we used a chemically aware splitting approach, where the pairwise similarities between the protein−ligand complexes are calculated, as it follows.

For the protein structure comparison, we used the TM-align software;[33] namely, it was used to evaluate protein-to-protein structural similarity (PPS) between any two given complexes in a data set. The corresponding similarity metric is the TM score, that belongs to the [0,1] range, where 1 indicates a perfect match between the two structures.

To calculate the structural similarity between the ligands (LLS), we used molecular fingerprints, which encode ligands by converting their structure into a bit vector, where each bit represents the presence or the absence of a certain chemical substructure or property.[34] In this work, ligand fingerprints were computed using RDKit Daylight-like fingerprint[35] with the default parameters.

Finally, to calculate the complex-to-complex similarity (CCS), we used protein−ligand interaction fingerprints SPLIF,[36] where all possible interaction types (e.g., $\pi-\pi$, CH$-\pi$, etc.) are encoded with bits. We used Open Drug Discovery Toolkit (ODDT)[37] to calculate SPLIF fingerprints.

For both RDKit and SPLIF fingerprints, we used the Tanimoto score[38] as the similarity metric. More precisely, given any two binary vectors, Z_1 and Z_2, the Tanimoto Coefficient (TC) can be calculated according to eq 1.

$$\mathrm{TC}(Z_1, Z_2) = \frac{|Z_1 \cap Z_2|}{|Z_1 \cup Z_2|} \tag{1}$$

It was shown in a previous study[39] that the Tanimoto coefficient is an appropriate choice for fingerprint-based similarity calculation.

Finally, given the similarity scores from a splitting approach, we calculated a similarity matrix S, and then it is converted to a distance matrix $D = 1 - S$. Then, we clustered the samples using an agglomerative clustering algorithm from Sklearn[40] with an average linkage. Note that choosing the similarity threshold is important,[41] and in this work, the threshold in the three chemically aware splits was set to 0.5, thus, imposing more difficult splits for testing the model's ability to generalize. The threshold value of 0.5 was selected based on the silhouette score curve to achieve better clustering (see Supporting

https://doi.org/10.1021/acs.jcim.3c00771
J. Chem. Inf. Model. 2024, 64, 2323−2330

Information Figure S3). The silhouette score is used to evaluate the quality of clustering; it ranges between [−1 and 1] where a value of 1 means that clusters are completely dense and separated, while a value of ≤0 means that clusters are overlapping or indicates incorrect labels for the clustered samples. We observed silhouette scores of 0.61 for PPS, 0.54 for LLS, and 0.36 for CCS splitting strategies. All clusters with more than 5 samples were randomly assigned to train and validation sets, and clusters containing less than 5 samples were assigned to the training set in a way to preserve the 4:1 ratio (see Supporting Information Table S1 for more details).

Descriptors. Here we compose a descriptor of the atomic environment of a binding site in a molecular complex relying on the work of Behler and Parrinello.[42] Importantly, the composed descriptor is invariant to permutation, rotation, and translation symmetries, and we previously showed the use of such a descriptor for the binding affinity prediction problem.[28] First, we define an energetically relevant local environment by using a cutoff function F_c (see eq 2), where r_{ij} is the distance between atoms i and j and R_c is the cutoff distance. The value of the function decreases with the distance increase and zeroes out contributions of atoms outside the local environment. In this study, we used R_c = 12 Å to include most of the atoms in the binding site.

$$F_c(r_{ij}) = \begin{cases} \tanh^3\left(1 - \frac{r_{ij}}{R_c}\right) & r_{ij} \le R_c \\ 0 & r_{ij} > R_c \end{cases} \tag{2}$$

Next, we use radial symmetry functions (BPS functions) to describe both pairwise distances between atoms (eq 3) and angles (eq 4) formed for each triplet of atoms i, j, and k.

$$G_i^2 = \sum_{i \neq j}^{\text{all}} e^{-\eta(r_{ij}-r_s)^2} F_c(r_{ij}) \tag{3}$$

$$G_i^3 = 2^{1-\zeta} \sum_{j,k \neq i}^{\text{all}} (1 + \lambda \cos \Theta_{ijk})^{\zeta} \times e^{-\eta(r_{ij}^2 + r_{jk}^2 + r_{ik}^2)} F_c(r_{ij}) F_c(r_{jk}) F_c(r_{ik}) \tag{4}$$

The parameter sets for r_s and η have 4 and 5 values, respectively, resulting in 20 descriptors obtained from G_i^2. Similarly, the computation of G_i^3 with the parameter sets of λ, ζ, and η having 2, 4, and 4 values, respectively, would result in 32 descriptors (see Supporting Information Table S2). As a result, the total number of values is 52 descriptors. For each ligand atom, 7 heavy protein atoms are taken into consideration: (C, N, O, P, S, M1, M2) where M1 represents single charged metal ions and M2 represents metal ions in the other charged state; hence, 7 × 52 = 364 values in the descriptor. Finally, the atomic composition of a ligand is represented by the one-hot encoding of 9 atom types (C, O, N, S, P, F, Cl, Br, I). Consequently, the final length of the initial node representation of the graph is 373.

Model. In this work, we use the graph neural network architecture, where the input is a graph representing the ligand molecule in the protein−ligand complex and the dimension of the input node features is 373. The proposed GNN is an intermediate fusion of 3 types of graph neural network architectures. Namely, a graph convolutional network,[43] graph attention network,[44] and a graph isomorphism network[45] are cascaded in parallel to process the input ligand graph with its nodes' features. Each one of these three networks has a different update function for node features during message passing according to eqs 5, 6, and 8, respectively. Each network includes 3 graph convolutional layers with 1D batch normalization layers in between followed by a Dropout layer and then by a multilayer perceptron (MLP) as a projection head to solve a graph-level regression task.

$$x_i' = \sum_{j \in N(i)} c_{ij} \Theta x_j \tag{5}$$

Here, x_i' is the updated feature vector, Θ is the learnable parameters matrix $c_{ij} = \frac{1}{\sqrt{|N(i)|\,|N(j)|}}$, and $|N(i)|$, $|N(j)|$are the sizes of neighborhoods of nodes i and j, respectively.

$$x' = \alpha_{i,i} \Theta x_i + \sum_{j \in N(i)} \alpha_{i,j} \Theta x_j \tag{6}$$

Here $\alpha_{i,j}$ represents the learned attention coefficients computed with eq 7

$$\alpha_{i,j} = \frac{\exp(\sigma(a^T[\Theta x_i \| \Theta x_j]))}{\sum_{k \in N(i) \cup \{i\}} \exp(\sigma(a^T[\Theta x_i \| \Theta x_k]))} \tag{7}$$

where $\sigma(.)$ is a nonlinear activation function ($LeakyReLU$) and a is a weight vector.

$$x' = h_\theta\left((1 + \epsilon) \cdot x_i + \sum_{j \in N(i)} x_j\right) \tag{8}$$

where h_θ denotes a neural network, that is, an MLP and ϵ is a parameter that can be learned or tuned.

In this work, we used the hyper-parameters for the BPS descriptors that were shown to be optimal for graphDelta in our previous work[28] (see Supporting Information Table S5). As for the model's hyper-parameters, we used the grid search with 5-fold cross-validation to find the optimal combination for the number of layers, the batch size, and the learning rate (see Supporting Information Table S4).

We use the mean squared error (MSE) (eq 9) as the loss function and the root mean squared error (RMSE) along with the Pearson correlation coefficient (R) (eq 10) as the evaluation metrics.

$$\text{MSE} = \sum_{i=1}^{n} (x_i - y_i)^2 \tag{9}$$

$$R = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{10}$$

Here n is the sample size, $\bar{x}$ is the mean of the ground truth values, and $\bar{y}$ is the mean of the predicted values.

RESULTS AND DISCUSSION

Overview of the Model. The developed model operates with a ligand represented as an unweighted-undirected graph, with atoms and chemical bonds corresponding to the nodes and the edges connecting these nodes. For each node in the molecular graph, the surrounding protein residues are encoded in the calculated descriptors and used as input node features. The protein−ligand complex representation is then fed to a fusion of graph neural networks, and Figure 1A illustrates the

https://doi.org/10.1021/acs.jcim.3c00771
J. Chem. Inf. Model. 2024, 64, 2323−2330

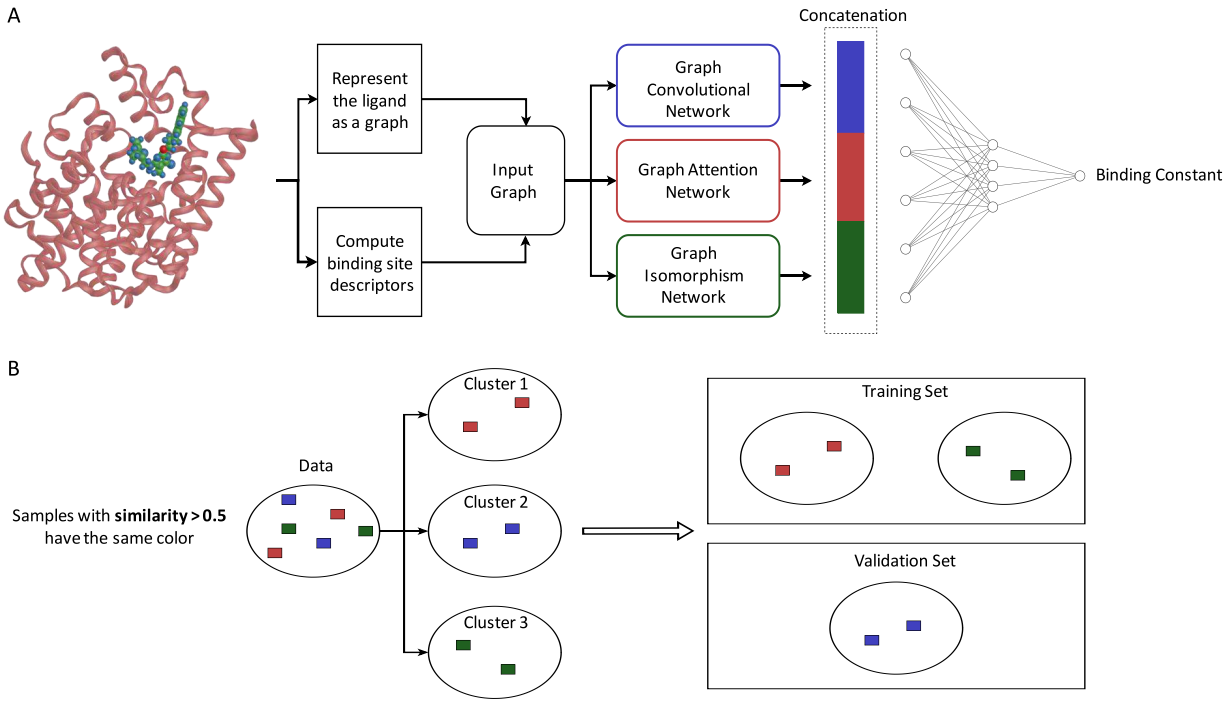


Figure 1. (A) General architecture of the graphLambda model. (B) Illustration of data clustering based on similarity between samples. Samples with similarity >0.5 have the same color.

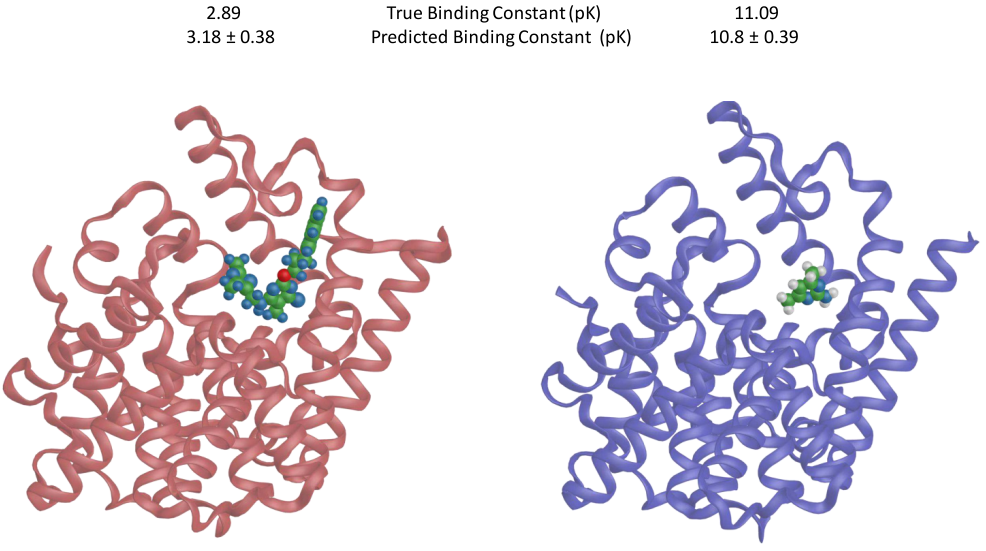


Figure 2. An example of graphLambda predictions. The protein complexes correspond to the same protein but different ligands. Ligands of the left and right complexes correspond to low and high binding affinity values, respectively. The predicted binding affinity value is averaged over the graphLambda models trained with different splitting strategies, not including the random split.

overall architecture of the model. The neural network architecture comprises a graph convolutional network, a graph attention network, and a graph isomorphism network. The aim of using this architecture is to infer the advantages of the corresponding graph blocks: the graph convolutional block is powerful in extracting local neighborhood features; the graph attention block infers a learnable attention mechanism to take into account that the presence of certain atoms can be more informative than others;[44] the graph isomorphism block can be helpful to improve the representational power of the model.[45] To the best of our knowledge, the proposed fusion method is the first attempt to combine three different graph-based representations for the task of protein−ligand binding affinity prediction.

To train the model, we used the refined subset of PDBbind,[46] as one of the most curated public data sets for binding affinity. However, using a random train-validation split for the derivation of predictive models will likely lead to overoptimistic results due to similar complexes shared between train and validation subsets. To avoid such a pitfall, we explored three different splitting strategies corresponding to different types of similarities: protein-to-protein structural similarity (PPS), ligand-to-ligand topological fingerprints similarity (LLS), and complex-to-complex interaction finger-

https://doi.org/10.1021/acs.jcim.3c00771
J. Chem. Inf. Model. 2024, 64, 2323−2330

Table 1. Comparing the Performance of Different GNNs Predicting the Binding Affinity on CASF16

	Splitting Approach							
	Random		PPS		LLS		CCS	
Model	*R*	RMSE	*R*	RMSE	*R*	RMSE	*R*	RMSE
GCN	0.9 (0.06)	0.93 (0.09)	0.7 (0.12)	1.58 (0.17)	0.92 (0.05)	0.836 (0.09)	0.93 (0.04)	0.84 (0.07)
GAT	**0.95** (0.03)	**0.63** (0.05)	0.65 (0.1)	1.65 (0.14)	**0.93** (0.04)	**0.78** (0.07)	0.86 (0.07)	0.965 (0.1)
GIN	0.91 (0.03)	0.88 (0.06)	0.69 (0.11)	1.58 (0.16)	0.88 (0.08)	0.998 (0.12)	0.91 (0.05)	0.88 (0.09)
GCN-GAT	0.93 (0.03)	0.811 (0.05)	0.82 (0.09)	1.27 (0.12)	0.88 (0.09)	0.994 (0.13)	0.91 (0.04)	0.88 (0.07)
GCN-GIN	0.91 (0.05)	0.886 (0.08)	0.81 (0.07)	1.27 (0.1)	0.88 (0.08)	1.034 (0.11)	0.9 (0.06)	0.939 (0.09)
GAT-GIN	0.93 (0.03)	0.788 (0.06)	0.85 (0.07)	1.12 (0.1)	0.91 (0.06)	0.932 (0.09)	0.88 (0.09)	0.895 (0.13)
GCN-GAT-GIN	0.93 (0.04)	0.784 (0.06)	**0.9** (0.05)	**0.927** (0.08)	0.91 (0.03)	0.9 (0.05)	**0.94** (0.04)	**0.76** (0.08)

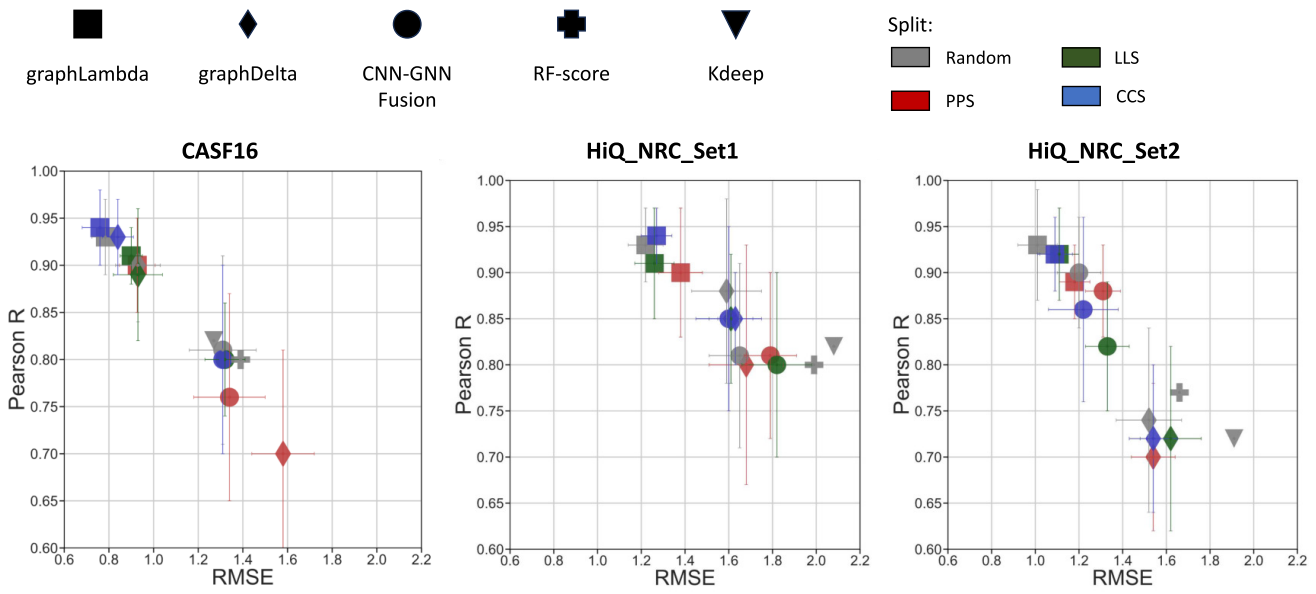


Figure 3. Performance metrics of graphLambda along with the state-of-the-art methods on the test benchmarks (**CASF16**, **HiQ_NRC_Set1**, **HiQ_NRC_Set1**). Each marker type refers to a prediction model: square, diamond, circle, cross, and triangle correspond to graphLambda, graphDelta, CNN-GNN Fusion, RF-score, and Kdeep, respectively. Each color represents the type of data split: gray, red, green, and blue correspond to random split, PPS, LLS, and CCS, respectively.

prints similarity (CCS). These three similarity metrics are used to cluster the data in the refined set and make training-validation subsets with a ratio of 4:1, as illustrated in Figure 1B. We trained the models using 4 different data splitting approaches (the random split and the 3 similarity-based splits described above) in a 5-fold cross-validation (CV) manner. Note that the random split is expected to yield a model with biased performance metrics because of similar complexes shared between train and validation subsets. The 5-fold CV yielded 5 instances of the trained models given the splitting strategy. The results on the validation set are R_{val} = 0.95 ± 0.02, and $RMSE_{val}$ = 0.71 ± 0.02. Figure 2 shows an example of the graphLambda's output when applied to the CGMP 3′,5′-cyclic phosphodiesterase protein in complex with low-affinity compound (4,6-dimethylpyrimidin-2-amine) and high-affinity compound (6-chloro-*N*-[(2,4-dimethyl-1,3-thiazol-5-yl)-methyl]-5-methyl-2-[3-(quinolin-2-yl)propoxy]pyrimidin-4-amine. As one can see graphLambda is able to correctly discriminate between the low- and high-affinity compounds. It is important to note that this particular protein complex is dissimilar to the training set (TM-align score, interaction fingerprint similarity, and the RDKit fingerprint similarity <0.5).

Ablation Study. In this work, the graphLambda model is a fusion of three different types of GNN (Figure 1A). We hypothesize that this combination boosts the predictive performance due to the complementarity of the underlying update functions utilized in these networks. More specifically, it was shown that (i) GCN extracts local neighborhood features around each atom in the molecular graph, (ii) GAT employs the attention mechanism that helps with generalization to unseen motifs, and (iii) GIN has better expressive power to learn a global function on graphs (see Methods). To verify our hypothesis, we conducted an ablation study as follows.

We composed the following subarchitectures to derive alternative GNN models: (GCN, GAT, GIN, GCN-GAT, GCN-GIN, GAT-GIN, and GCN-GAT-GIN). These models were trained and tested using the same protocol that was used to train and test the final graphLambda model. Table 1 demonstrates the obtained results; the best values related to *R* and RMSE are highlighted with a bold font. Although the GAT model achieved higher *R* and RMSE on both random and LLS splits, the GCN-GAT-GIN combination demonstrates the most robust results across all the splits with the difference between the best and the worst performances for the Pearson

https://doi.org/10.1021/acs.jcim.3c00771
J. Chem. Inf. Model. 2024, 64, 2323−2330

correlation coefficient and RMSE of 0.04 and 0.17, respectively (see Supporting Information Table S4). Such a margin for both metrics is at least two times better compared to any other combination, indicating that the fusion of different graph neural networks (GCN, GAT, and GIN) combines the advantages of the individual architecture types. It is worth noting that GCN, GAT, and GIN likely produce different embeddings of the input graph because of different aggregation functions. More precisely, the GCN aggregator sums the hidden states of neighboring nodes in a way that the importance of each hidden vector depends on the size of the neighborhood of the contributing nodes (see eq 5). Meanwhile, the GAT aggregator relies on the attention mechanism that learns the importance of nodes and their corresponding features (eqs 6 and 7). Finally, the GIN aggregator (eq 8) uses an MLP to update node representations resulting in more distinguishing power of non-isomorphic graphs.

Benchmarks. We evaluated graphLambda on CASF16 and CSAR_HiQ_NRC benchmarks and compared it with the graphDelta[28] and the CNN-GNN fusion model developed by Jones et al.[47] We chose those models for comparison because they achieve state-of-the-art (SOTA) results, and both of them are graph-based models that utilize structural information. Note that we retrained the SOTA models using the provided training protocols for the data set splits for a fair comparison. Figure 3 and Supporting Information Table S1 show the performance metrics obtained for the test benchmarks. As one can see, graphLambda outperforms both graphDelta and CNN-GNN fusion model on all benchmarks and for different splits with the average Pearson correlation coefficient and RMSE of 0.92 ± 0.01 and 1.07 ± 0.06, respectively. Note that graphLambda and graphDelta use similar descriptors of the atomic environment, suggesting the superior performance of graphLambda is due to the proposed fusion graph neural network architecture. It is worth noting that the graphLambda model ($RMSE_{avg}$ = 0.86 ± 0.07, R_{avg} = 0.92 ± 0.02) also outperforms the 3D CNN-based model Kdeep[26] ($RMSE_{avg}$ = 1.75 ± 0.35, R_{avg} = 0.73 ± 0.07) and random forest model RF-score[15] ($RMSE_{avg}$ = 1.68 ± 0.25 and R_{avg} = 0.72 ± 0.02) across the 3 test sets. In addition, we have tested another graph-based model, MGraphDTA, provided in the original work[27] on the benchmarks. We observed the superior performance of graphLambda compared with MGraphDTA, and the obtained results are listed in Supporting Information Table S1.

Virtual Ligand Screening. To demonstrate the broader applicability of graphLambda, we tested it in a virtual ligand screening application. We considered a data set of 895 compounds with measured IC50 ranging from low nM to low μM affinity from the CACHE4 challenge,[48] where the goal is to find high-affinity compounds for the TKB domain of the CBLB protein (PDB ID:8GCY). We docked the provided list of ligands using the smina software[49] with default parameters. It is important to note that the used docking protocol reproduced the cocrystallized binding pose (PDB ID:8GCY) (see Supporting Information Figure S4). We scored the obtained protein−ligand complexes using graphLambda as well as DeepDTA, graphDelta, and CNN-GNN Fusion models. To compute the accuracy, we checked if the predicted binding constant (i) belongs to the ground truth affinity interval and (ii) is correctly placed with respect to the 30 μM binding affinity threshold of the challenge. For the latter, we applied the 30 μM threshold to the ground truth IC50 ranges and predicted binding constants, thus obtaining ground truth and predicted binary labels. Note that we discarded 86 molecules, for which 30 μM belongs to the IC50 range, and the remaining 809 have their IC50 ranges below 30 μM and, thus, form only the positive class. To compose the negative class, we randomly selected ten sets of 809 molecules satisfying Lipinski's Rule of Five[50] from ChEMBL. Next, we created ten balanced sets containing 1618 molecules and calculated averaged performance metrics. The graphLambda model outperformed the other prediction models in the two evaluation settings, and the obtained results are listed in Supporting Information Table S3. More specifically, the average percentage accuracy of graphLambda using the 30 μM threshold is 54 ± 2.7, 96 ± 0.8, 75 ± 2.1, and 95 ± 1.2 for the random split, PPS, LLS, and CCS splits, respectively. As for the belonging to the ground truth IC50 interval, we obtained the following average percentage accuracy: 48 ± 2.1, 91 ± 1.3, 68 ± 1.2, and 88 ± 1.4 for the random split, PPS, LLS, and CCS splits, respectively. As one can see, the random split resulted in the worst performance, which again highlights the importance of data set splits for the scoring function development. To summarize, graphLambda is applicable for the virtual ligand screening campaigns and demonstrated superior performance in the TKB domain of the CBLB protein case study.

Conclusion. In this study, we presented a deep learning-based approach for binding affinity prediction that relies on different graph neural network architectures. We derived the graphLambda scoring function using graph convolutional, attention, and isomorphism blocks and showed that the fusion of these blocks resulted in the best performance. We have composed comprehensive train-validation splits taking into account protein−protein, ligand−ligand, and interface-interface similarities between the data set samples. We observed that graphLambda demonstrates robustness in terms of the performance metrics when trained on the composed training sets. Finally, graphLambda showed superior performance on the test benchmarks compared with the state-of-the-art approaches. graphLambda is available at https://github.com/i-Molecule/graphLambda.

■ ASSOCIATED CONTENT

Data Availability Statement

Additional details are available in the Supporting Information: distribution of the binding constants in PDBbind data, hyperparameter tuning for clustering, BPS functions parameters, details of training setup, and detailed results of graphLambda compared to the state-of-the-art models. graphLambda is available in the GitHub repository at https://github.com/i-Molecule/graphLambda.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.3c00771.

Distribution of the binding constants in PDBbind data, hyperparameter tuning for clustering, BPS functions parameters, details of training setup, and detailed results of graphLambda compared to the state-of-the-art models (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Petr Popov − *Tetra-d, Schaffhausen 8200, Switzerland; School of Science, Constructor University Bremen gGmbH,*

https://doi.org/10.1021/acs.jcim.3c00771
J. Chem. Inf. Model. 2024, 64, 2323−2330

Bremen 28759, Germany; orcid.org/0000-0003-3745-7154; Email: ppopov@constructor.university

Author

Ghaith Mqawass − *Faculty of Computer Science, University of Vienna, Vienna A-1090, Austria; UniVie Doctoral School Computer Science, University of Vienna, Vienna A-1090, Austria;* orcid.org/0009-0005-2514-5999

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.3c00771

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The work has been funded by the Vienna Science and Technology Fund (WWTF) [10.47379/VRG19009].

■ REFERENCES

(1) Sadybekov, A. V.; Katritch, V. Computational approaches streamlining drug discovery. *Nature* **2023**, *616*, 673−685.
(2) Smith, J.; Johnson, E. A Survey of Computer-Aided Drug Discovery Methods and Binding Affinity Prediction. *Journal of Chemical Informatics* **2020**, *12*, 789−804.
(3) Meli, R.; Morris, G. M.; Biggin, P. C. Scoring functions for protein-ligand binding affinity prediction using structure-based deep learning: A review. *Frontiers in bioinformatics* **2022**, *2*, 57.
(4) Tropsha, A. Best practices for QSAR model development, validation, and exploitation. *Molecular informatics* **2010**, *29*, 476−488.
(5) Corso, G.; Stärk, H.; Jing, B.; Barzilay, R.; Jaakkola, T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking*arXiv*. 2023. DOI: 10.48550/arXiv.2210.01776
(6) Jain, R.; Chen, H.; Zhu, X.; Chen, K.; Guo, J. Scoring Functions in Computer-Aided Drug Discovery: A Comprehensive Review. *Drug Discovery Today* **2021**, *26*, 1656−1668.
(7) Wang, Z.; Sun, H.; Yao, X.; Li, D.; Xu, L.; Li, Y. Scoring functions for protein-ligand interactions: a comprehensive survey. *J. Mol. Model.* **2019**, *25*, 1−25.
(8) Weiner, S. J.; Kollman, P. A.; Case, D. A.; Singh, U. C.; Ghio, C.; Alagona, G.; Profeta, S.; Weiner, P. A new force field for molecular mechanical simulation of nucleic acids and proteins. *J. Am. Chem. Soc.* **1984**, *106*, 765−784.
(9) Zou, X.; Yaxiong; Kuntz, I. D. Inclusion of Solvation in Ligand Binding Free Energy Calculations Using the Generalized-Born Model. *J. Am. Chem. Soc.* **1999**, *121*, 8033−8043.
(10) Gilson, M. K.; Given, J. A.; Head, M. S. A new class of models for computing receptor-ligand binding affinities. *Chem. Biol.* **1997**, *4*, 87−92.
(11) Eldridge, M. D.; Murray, C. W.; Auton, T. R.; Paolini, G. V.; Mee, R. P. Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *J. Comput. Aided. Mol. Des.* **1997**, *11*, 425−445.
(12) Yang, C.; Zhang, Y. Lin_F9: A Linear Empirical Scoring Function for Protein−Ligand Docking. *J. Chem. Inf. Model.* **2021**, *61*, 4630−4644. PMID: 34469692.
(13) Zheng, Z.; Merz, K. M. Development of the Knowledge-Based and Empirical Combined Scoring Algorithm (KECSA) To Score Protein−Ligand Interactions. *J. Chem. Inf. Model.* **2013**, *53*, 1073−1083. PMID: 23560465.
(14) Bitencourt-Ferreira, G.; Rizzotto, C.; de Azevedo Junior, W. F. Machine Learning-Based Scoring Functions, Development and Applications with SAnDReS. *Curr. Med. Chem.* **2021**, *28*, 1746−1756.
(15) Ballester, P. J.; Mitchell, J. B. A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics* **2010**, *26*, 1169−1175.
(16) Zilian, D.; Sotriffer, C. A. SFCscoreRF: A Random Forest-Based Scoring Function for Improved Affinity Prediction of Protein−Ligand Complexes. *J. Chem. Inf. Model.* **2013**, *53*, 1923−1933. PMID: 23705795.
(17) Sotriffer, C.; Sanschagrin, P.; Matter, H.; Klebe, G. SFCscore: Scoring functions for affinity prediction of protein−ligand complexes. *Proteins* **2008**, *73*, 395−419.
(18) Das, S.; Krein, M. P.; Breneman, C. M. Binding Affinity Prediction with Property-Encoded Shape Distribution Signatures. *J. Chem. Inf. Model.* **2010**, *50*, 298−308. PMID: 20095526.
(19) He, T.; Heidemeyer, M.; Ban, F.; Cherkasov, A.; Ester, M. SimBoost: a read-across approach for predicting drug−target binding affinities using gradient boosting machines. *J. Cheminform* **2017**, *9*. DOI: 10.1186/s13321-017-0209-z
(20) Meli, R.; Anighoro, A.; Bodkin, M.; Morris, G. M.; Biggin, P. C. Learning protein-ligand binding affinity with atomic environment vectors. *Journal of Cheminformatics* **2021**, *13*. DOI: 10.1186/s13321-021-00536-w
(21) Öztürk, H.; Özgür, A.; Ozkirimli, E. DeepDTA: deep drug−target binding affinity prediction. *Bioinformatics* **2018**, *34*, i821−i829.
(22) Wang, Z.; Zheng, L.; Liu, Y.; Qu, Y.; Li, Y.-Q.; Zhao, M.; Mu, Y.; Li, W. OnionNet-2: A Convolutional Neural Network Model for Predicting Protein-Ligand Binding Affinity Based on Residue-Atom Contacting Shells. *Frontiers in Chemistry* **2021**, *9*. DOI: 10.3389/fchem.2021.753002
(23) Golkov, V.; Skwark, M. J.; Mirchev, A.; Dikov, G.; Geanes, A. R.; Mendenhall, J.; Meiler, J.; Cremers, D. 3D Deep Learning for Biological Function Prediction from Physical Fields. *2020 International Conference on 3D Vision (3DV)* **2020**, 928−937.
(24) Sosnin, S.; Misin, M.; Palmer, D. S.; Fedorov, M. V. 3D matters! 3D-RISM and 3D convolutional neural network for accurate bioaccumulation prediction. *J. Phys.: Condens. Matter* **2018**, *30*, No. 32LT03.
(25) Kwon, Y.; Shin, W.-H.; Ko, J.; Lee, J. AK-Score: Accurate Protein-Ligand Binding Affinity Prediction Using an Ensemble of 3D-Convolutional Neural Networks. *International Journal of Molecular Sciences* **2020**, *21*, 8424.
(26) Jiménez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. K deep: protein-ligand absolute binding affinity prediction via 3d-convolutional neural networks. *J. Chem. Inf. Model.* **2018**, *58*, 287−296.
(27) Yang, Z.; Zhong, W.; Zhao, L.; Yu-Chian Chen, C. MGraphDTA: deep multiscale graph neural network for explainable drug−target binding affinity prediction. *Chem. Sci.* **2022**, *13*, 816−833.
(28) Karlov, D. S.; Sosnin, S.; Fedorov, M. V.; Popov, P. graphDelta: MPNN Scoring Function for the Affinity Prediction of Protein−Ligand Complexes. *ACS Omega* **2020**, *5*, 5150−5159. PMID: 32201802.
(29) Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Comparative Assessment of Scoring Functions: The CASF-2016 Update. *J. Chem. Inf. Model.* **2019**, *59*, 895−913.
(30) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic acids research* **2000**, *28*, 235−242.
(31) Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Comparative Assessment of Scoring Functions: The CASF-2016 Update. *J. Chem. Inf. Model.* **2019**, *59*, 895−913.
(32) Dunbar, J. B.; Smith, R. D.; Yang, C.-Y.; Ung, P. M.-U.; Lexa, K. W.; Khazanov, N. A.; Stuckey, J. A.; Wang, S.; Carlson, H. A. CSAR Benchmark Exercise of 2010: Selection of the Protein−Ligand Complexes. *J. Chem. Inf. Model.* **2011**, *51*, 2036−2046.
(33) Zhang, Y.; Skolnick, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* **2005**, *33*, 2302 PMID: 15849316.
(34) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983−996.
(35) RDKit Development Team RDKit: Open-Source Cheminformatics Software. RDKit website,2022; https://www.rdkit.org/.
(36) Da, C.; Kireev, D. Structural Protein−Ligand Interaction Fingerprints (SPLIF) for Structure-Based Virtual Screening: Method

and Benchmark Study. *J. Chem. Inf. Model.* **2014**, *54*, 2555−2561. PMID: 25116840.
(37) Wójcikowski, M.; Zielenkiewicz, P.; Siedlecki, P. Open Drug Discovery Toolkit (ODDT): a new open-source player in the drug discovery field. *Journal of Cheminformatics* **2015**, *7*. DOI: 10.1186/s13321-015-0078-2
(38) Tanimoto, T. *IBM Internal Report 17th* **1957**; Vol. *7*.
(39) Bjusz, D.; Rácz, A.; Héberger, K. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics* **2015**, *7*. DOI: 10.1186/s13321-015-0069-3
(40) Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; Layton, R.; VanderPlas, J.; Joly, A.; Holt, B.; Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* **2013**, 108−122.
(41) Sharma, A.; Lal, S. P. Tanimoto Based Similarity Measure for Intrusion Detection System. *Journal of Information Security* **2011**, *2*, 195−201.
(42) Behler, J.; Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **2007**, *98*. DOI: 10.1103/PhysRevLett.98.146401
(43) Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR* **2016**, *abs/1609.02907*.
(44) Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks, *arXiv*. 2017; https://arxiv.org/abs/1710.10903. DOI: 10.48550/arXiv.1710.10903
(45) Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? *arXiv*, 2018; https://arxiv.org/abs/1810.00826. DOI: 10.48550/arXiv.1810.00826
(46) Liu, Z.; Su, M.; Han, L.; Liu, J.; Yang, Q.; Li, Y.; Wang, R. Forging the Basis for Developing Protein−Ligand Interaction Scoring Functions. *Acc. Chem. Res.* **2017**, *50*, 302−309.
(47) Jones, D.; Kim, H.; Zhang, X.; Zemla, A.; Stevenson, G.; Bennett, W. F. D.; Kirshner, D.; Wong, S. E.; Lightstone, F. C.; Allen, J. E. Improved Protein−Ligand Binding Affinity Prediction with Structure-Based Deep Fusion Inference. *J. Chem. Inf. Model.* **2021**, *61*, 1583−1592. PMID: 33754707.
(48) Finding ligands targeting the TKB domain of CBLB https://cache-challenge.org/challenges/finding-ligands-targeting-the-tkb-domain-of-cblb; 2023.
(49) Koes, D. R.; Baumgartner, M. P.; Camacho, C. J. Lessons Learned in Empirical Scoring with smina from the CSAR 2011 Benchmarking Exercise. *J. Chem. Inf. Model.* **2013**, *53*, 1893−1904. PMID: 23379370.
(50) Lipinski, C. A. Lead- and drug-like compounds: the rule-of-five revolution. *Drug Discovery Today: Technologies* **2004**, *1*, 337−341.

https://doi.org/10.1021/acs.jcim.3c00771
J. Chem. Inf. Model. 2024, 64, 2323−2330