Proximity Graph Networks: Predicting Ligand Affinity with Message Passing Neural Networks

Zachary J. Gale-Day[1,3], Laura Shub[1,2,3,4], Kangway V. Chuang[1,2,3,4] and Michael J. Keiser[1.2,3,4]*

1. Department of Pharmaceutical Chemistry, University of California, San Francisco, CA, 94158, USA
2. Department of Bioengineering and Therapeutic Sciences, University of California, San Francisco, CA, 94158, USA
3. Institute for Neurodegenerative Diseases, University of California, San Francisco, San Francisco, CA, 94158, USA
4. Bakar Computational Health Sciences Institute, University of California, San Francisco, CA, 94158, USA

*To whom correspondence should be addressed, keiser@keiserlab.org

Abstract

Message-passing neural networks (MPNNs) on molecular graphs generate continuous and differentiable encodings of small molecules with state-of-the-art performance on protein-ligand complex scoring tasks. Here, we describe the Protein-Graph Network (PGN) package, an open-source toolkit that constructs ligand-receptor graphs based on atom proximity and allows users to rapidly apply and evaluate MPNN architectures for a broad range of tasks. We demonstrate the utility of PGN by introducing benchmarks for affinity and docking score prediction tasks. Graph networks generalize better than fingerprint-based models and perform strongly for the docking score prediction task. Overall, MPNNs with Proximity Graph data structures augment the prediction of ligand-receptor complex properties when ligand-receptor data are available.

Introduction

Computational and machine learning (ML)-based approaches to predicting binding affinity are critical research directions in drug discovery.[1–4] A strong predictor of a ligand affinity is desirable both for hit identification in virtual screening and for computationally evaluating structure-activity relationships for hit expansion and hit-to-lead optimization. These approaches include ligand-based methods that exclusively use 2-dimensional (2D) molecular representations of known ligands to infer binding based on molecular similarity and structure-based approaches that encode three-dimensional (3D) protein-ligand interactions. Despite recent advances in these approaches, predicting ligand affinity remains a critical challenge in computational drug design, particularly for generalizing to novel chemotypes.[5] Developing strong computational predictors would enable the computationally assisted medicinal chemist to evaluate many more compounds in less time and cost than a purely experimental approach.

Recent reports show the power of learnable molecular representations by using Message Passing Neural Networks (MPNNs) to accept the raw molecular graph as input.[6–16] Importantly, these learnable representations are tuned to each prediction task, allowing for a richer and more task-specific encoding of the molecular graph, and have been shown to outperform hash-based encodings in head-to-head comparisons in some cases.[10,17] Beyond molecule-autonomous applications, early reports suggest that MPNNs can predict important properties of ligand-receptor complexes.[7,18] Unlike methods based on ligand structure only, these networks either explicitly or implicitly encode 3D structural information about the ligand-receptor complex.

This paper introduces Proximity Graph Networks (PGNs), an open-source toolkit that allows for simple extension of multiple MPNN architectures to ligand-receptor graphs. This software package enables information to pass between ligand and protein atoms during learning, which we show can greatly affect model performance. Tunable ligand-receptor encodings offer performance advantages in predicting ligand-receptor affinities. We also highlight MPNN's modularity, allowing us to implement new encoder architectures with minimal changes. We find that different MPNN architectures are suited to different tasks, highlighting the importance of a modular framework for easy evaluation of MPNN architectures. PGNs showed strong performance compared to other published approaches on PDBbind datasets.[19–21] Additionally, the PGNs improved generalization performance on ligands bound to receptors not seen in the training set. We also evaluated our models for a fine-grained, single protein, D4 Dopamine Receptor docking-score prediction task.[22] Strong performance on the docking-score prediction task can aid in hit picking and improving the application of deep learning to streamline docking workflows.[23,24] These results indicate that PGNs could be a powerful tool to learn the properties of ligand-receptor complexes.

Background

Developing accurate computational scoring functions (SFs) for assessing protein-ligand binding affinity is an ongoing challenge, with approaches ranging from molecular fingerprints to docking. These approaches include fingerprint and atom-pair expert encodings (PLEC[25], LUNA[26] and ECIF[27] and docking-based SFs (Glide[28], RF-score[29], NN-score[30]). In contrast, new deep-learning methods are based on graph encoders.[31,32] The application of graph encoders to cheminformatics tasks shows promise at improving existing scoring functions.[6–16] Early research using deep learning models as SFs includes TopologyNet[33] and several traditional convolutional models that use voxel-based representations of ligand-receptor complexes as inputs.[34–37] The following paragraphs discuss several early applications using MPNNs to learn SFs.[7,18]

Our approach to generating ligand-receptor graphs resembles PLEC's implementation of proximity-based implicit graphs during fingerprint (FP) generation.[25] Beyond this, works like ours focus on directly applying message-passing neural networks to the ligand-receptor graphs. Notably, *Feinberg et al.* present PotentialNet for various molecular applications, including PDBbind.[18] Unfortunately, the limited information about implementation and the unavailability of their codebase makes direct comparison with PotentialNet challenging. Additionally, a recent report by *Cho et al*.[7] has further explored this approach, emphasizing a single architecture, only the PDBbind Refined dataset, and a slightly different graph generation procedure than the one used by PotentialNet or herein. Finally, two recent approaches have applied novel attention-based architectures with good effect on this task.[38,39]

In contrast to these previous works, this paper introduces PGN for optimized message-passing schemes. It provides an open-source implementation of MPNN architectures based on Gated-Graph Neural Networks[31] and Directed MPNNs[10] to predict ligand-receptor properties (**Figure 1**). Our experimentation with message-passing parameters using our open-source codebase may be a guide and tool for scientists interested in applying MPNNs to their tasks. The modular nature of our package will also allow for simple testing of different graph generation schemes and new MPNN architectures.

Methods

We summarize generalized Message Passing Neural Networks as defined by Gilmer *et al*,[8] and extend this framework to the PFP architecture implemented in our PGN software package. Please see the Supplementary Methods section for descriptions of previously reported architectures (Gated Graph Neural Networks[31] and Directed Message Passing Neural Networks[10]).

Message Passing Neural Networks. We describe the MPNN formalism for an undirected graph (G) with node features (x_v) and edge features (e_{vw}). MPNN comprises two distinct steps: the message passing phase that spreads node information to neighbors and the readout phase, which transforms node representations into graph-level representations.

The message passing phase of T time steps (also commonly referred to as D depth) has two operators: the message function M_t and a vertex update function U_t. Message passing transforms the node features into a new hidden representation h_v^tat each time step. The initial node representation h_v^0 can either be the raw node features ($h_v^t = x_v$) or a transformed version of the initial features ($h_v^t = NN(x_v)$), where NN is a simple neural network. Subsequent time steps update the hidden values of the nodes according to:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t\left(h_v^t, h_w^t, e_{vw}\right)$$

$$h_v^{t+1} = U_t\left(h_v^t, m_v^{t+1}\right),$$

where $N(v)$ is the set of neighboring nodes adjacent to v in graph G and m_v^{t+1} is the sum of all messages from nodes in the set $N(v)$. After all T steps of message passing, h_v^T contains the final node representations. We aggregate the node representations to yield a graph-level representation for further learning using the readout function:

$$\hat{y} = R\left(h_v^T \mid v \in \{G\}\right).$$

The readout function must have several properties to guarantee invariance to graph isomorphism (e.g., the ordering of the nodes cannot affect the network output).[8] The graph level representation $\hat{y}$ is generally further transformed by a fully connected neural network to perform a regression of classification task. The functions above must be differentiable to learn graph representations from data.

Proximity Fingerprint Network. The Proximity Fingerprint (PFP) Network model simplifies the GGNET architecture.[31] We tuned PFP to perform better in low-data situations, drawing inspiration from Duvenaud *et al*'s early fingerprint-like MPNNs.[9]

Message Passing: PFP's message is defined identically to GGNET:

$$M_t\left(h_v, h_w, e_{vw}\right) = A_{e_{vw}} h_{w.}$$

Above, $A_{e_{vw}}$ corresponds to a learnable weight matrix. To simplify the model, PFP replaces the GRU-based update function by simply aggregating messages through a rectified linear unit:

$$U_t = U = ReLU\left(m_v^{t+1}\right).$$

Importantly, the updated hidden states of the nodes (h_v^{t+1}) do not depend upon the previous hidden state directly. We account for this lack of state memory through the readout function described below.

Readout**:** We use residual connections at each message passing timestep t to ensure that all levels contribute to the output representation $\hat{y}$,. Therefore, we define R_t as the readout at timestep t and R as the overall readout function. During each timestep, a neural network transforms the node's hidden state to the desired output dimension, followed by a simple add pool to guarantee node order invariance. Finally, we use a $LogSoftmax$ layer to yield the final output:

$$R_t = LogSoftmax\left(\sum_{v\in G} NN\left(h_v^t\right)\right).$$

The final readout is then a simple linear combination of the readouts R_t:

$$\hat{y} = \sum_t R_t.$$

Graph Construction. We used covalent bond edges and proximity-based virtual edges to build the ligand-receptor proximity graphs. This approach mirrors virtual graph construction in PLEC fingerprints.[25] We started building the graph representation from the ligand atoms. From each heavy atom, we added the protein atoms within a sphere of 4.5 Å radius (**Figure 1a**). Next, we added virtual "proximity" edges connecting the ligand and proximal protein atoms (**Figure**

1b) to allow information to flow between the ligand and the protein during message passing. Finally, to ensure all nodes were reachable, we added all atoms within five bonds of the proximal protein atoms and edges for all interconnecting bonds. **Figure 1c** shows the final set of nodes (atoms) and edges (covalent and proximity) as well as the completed proximity graph for another protein-ligand pair with more extensive contacts. We applied a simple featurization of atoms (**Table 1**) and bonds (**Table 2**).

Hyperparameter Optimization.

We performed Bayesian Optimization to optimize model hyperparameters. We used Hyperopt[40] in Python to tune model depth (i.e., number of message passing steps), dropout rate, number of fully connected layers in the regressor, and hidden dimension (size of the fully connected layer in the regressor or size of the $\hat{y}$ vector, depending on the model).

Implementation.

We implemented all models in PyTorch[41] and PyTorch Geometric.[42] Molecular data processing used Open Babel,[43] RDKit, and ODDT.[44] For basic graph data structures, we used NetworkX.[45] We adapted the DMPNN code from the chemprop repository[10] to work with proximity graph data structures. We visualized structures with Chimera.[46]

Experiments

Data.

We tested our models on the PDBbind 2019 Refined, PDBbind 2019 General, and D4 Diverse Docking datasets:

PDBbind Refined Set. The refined set is a subset of the general set that we filtered to include only the highest-quality ligand-receptor complexes. The filtering pipeline is described in Wang *et al.*[21] The final dataset consists of 4,852 ligand-receptor complexes.

PDBbind General Set. The general set includes all 21,382 structures in the PDBbind database.[20,47] However, we only included Protein-Ligand complexes for this paper, thus narrowing the set to 17,679 structures. We employed no extra filtering or other manipulation.

D4 Diverse Docking Set. This paper introduces a docking score prediction task for the Dopamine D4 Receptor. For this task, our dataset includes 86,452 ligands from the ZINC database[48] docked in the Dopamine D4 Receptor from an Ultra-Large Library[22,49] (ULL) docking campaign. The compounds all represent different structural scaffolds as determined by Bemis-Murcko Scaffold Splitting.[50] The structures were annotated with the docking score.

D4 Experimental Dataset. This dataset contains the subset of 510 ligands with both docked structures to the Dopamine D4 Receptor and experimental binding data. The structures all result from the same ULL docking campaign as the D4 Diverse Docking Set. We used this dataset for classification; docked structures were either binders or non-binders. Additionally, we used the D4 Experimental Dataset for our metric learning task discussed below.

Experimental Procedure.

Cross-validation and Hyperparameter Optimization. Each architecture type was optimized individually for each dataset and training/test split strategy type (discussed in the following subsection). We ran five iterations of hyperparameter optimization for each parameter. Model performance was the average validation loss over 5-fold cross-validation with a given set of hyperparameters. Due to significantly longer training times than other architectures, we ran DMPNN hyperparameter optimization for the PDBbind General and D4 Diverse datasets with 3-fold cross-validation. Before cross-validation, we held out a test set of approximately 10% of examples. Final evaluation of all models was done with optimal hyperparameters on five randomly seeded initializations using the test set selected before model selection. The same test set was used for each set of hyperparameter values to minimize data contamination.

Random Split. In the random split, examples were randomly sorted into test and training sets. The training set was then further split into cross-validation folds, without replacement, when hyperparameter optimization was performed.

Protein Split (PDBbind). The complexes were grouped by the annotated UniProt ID in the PDBbind database. Groups were then shuffled and added to the test set until the size of the test set exceeded the desired test percent. Groups that would make up more than 10% of the test set were automatically assigned to the training set. For cross-validation, the training set was split such that no more than one group of structures was in both the training and validation set. We expected this split to be a more rigorous test of model generalization than random splitting.

Similarity Split (D4 Diverse). For the D4 dataset, a standard scaffold split yields no clusters due to how the dataset was constructed. Therefore, to maximize the difference between the ligands in the training and test sets, Butina clustering[51] of RDKit fingerprints (size 1024) with a similarity threshold of 0.7 was performed. Groups were segregated (as discussed above) for the protein split methodology. Due to the long hyperparameter optimization times, the model parameters from random splitting experiments were used to evaluate model performance.

Metrics. For the PDBbind Refined, PDBbind General, and D4 Diverse datasets both the RMSE and the Pearson Correlation Coefficient were used for evaluation, in line with CASF[19,52,53] recommendations for standard PDBbind evaluation.

Baseline. Given the similarity of the PLEC implicit graph to the proximity graph data structure, we used PLEC as the primary baseline model for the study. We used the same feed-forward network and hyperparameter optimization scheme (as discussed above) to evaluate PLEC performance for comparability. We also evaluated the models on the CASF 2016 splits to provide a more extensive comparison to published models.[19]

Controls for shortcut learning. We used three separate adversarial controls to stress-test each model: frozen MPNN, proximity-edge ablation, and ligand only. We froze encoder weights in the ‘frozen MPNN’ control before training the model to yield a non-learnable, arbitrary graph representation. The proximity-edge ablation control removed the proximity edges from the graph to test how important message passing between the ligand and protein nodes was to model performance. Finally, the ligand-only control removed all non-ligand atoms and edges.

Results and Discussion

In this section, we performed experiments to evaluate 1) whether MPNNs would offer advantages over fingerprint and other baseline methods, 2) assess the importance of proximity features for protein-ligand models, and 3) whether these models could be effective predictors of experimental binding affinity. To answer these questions, we systematically benchmarked multiple approaches across PDBBind and D4 Dock Datasets and evaluated performance based on RMSE and Pearson Correlation Coefficient (PCC). PCC is more relevant for rank ordering compounds for testing, whereas RMSE is more relevant for predicting raw binding energies in isolation and therefore the inclusion of both gives a better picture of usefulness than either alone. Additionally these metrics are the standard for the PDBBind dataset[47] which will allow for easy comparison with past and future work. Error bars shown in the text represent the standard error of the mean for the given experiment.

Unless otherwise stated, all results below use the optimal hyperparameters (see **Tables S1-S14**) from the hyperparameter search. All results reported for graph models use the complete Proximity Graph data structure unless explicitly stated otherwise.

Performance on PDBbind Datasets. First, we evaluated each model on the PDBbind Refined and General datasets. We tested all models with both random splitting and protein splitting.

PDBbind Refined dataset. **Figure 2** shows the results of model evaluation on the refined dataset. The PFP encoder-based models performed similarly to or significantly better than PLEC in all cases. All other MPNN approaches underperformed on this dataset.

Interestingly, the PFP model trained with random splitting consistently outperformed the baseline by PCC metric but had a similar RMSE. This discrepancy suggests that the baseline better fits the data distribution, while PFP produced a stronger linear correlation. When considering the protein splitting performance, PFP significantly outperformed the baseline, suggesting that the graph model has better generalization performance. In addition, when we applied our best-performing model to the CASF-2016 benchmark, the PFP encoder outperformed all methods aside from deltaVinaRF20 (**Table S15**).

PDBbind General dataset. In this case, the baseline significantly outperformed all MPNNs on the random splits, while the PFP Network significantly outperformed the baseline on the more challenging protein split task (**Figure 3**). Interestingly, the DMPNN architecture performed better on this task, displaying comparable performance to PFP with random splitting. Once again, PFP had a significant performance advantage when using protein splitting.

Performance on D4 Diverse Dataset. Next, we evaluated the performance of the different models on the D4 Diverse dataset, which is a diverse set of molecules docked into the D4 Dopamine receptor (**Figure 4a-b**). We tested all models using random and similarity splitting, as described above. This dataset offers a different task type than the PDBbind dataset, which attempts to capture the general features of protein-ligand structures labeled with coarse-grained experimental affinities. The D4 diverse task focuses on learning a specialized representation where the model must differentiate a diverse set of molecules at a homogeneous interface. This specific binding model, therefore, requires the ability to differentiate similar binding modes and accurately predict a sensitive readout of protein-ligand complementarity.

When assessed against the D4 Diverse dataset, all graph models outperformed the baseline by a large margin (**Figure 4c-e**). In contrast to the results observed for PDBbind datasets, the DMPNN model performed best, followed by GGNET and PFP. The similarity split data resulted in models that performed no worse than the random split, likely due to the large diversity already seen within the dataset.

Next, we artificially limited the training set size and evaluated model performance using the full test set to understand if dataset size would strongly affect relative model performance (**Figure 4f**). All MPNN architectures outperformed the baseline, regardless of the dataset size. This result suggests that a learnable representation was particularly advantageous for this task. We were next interested in seeing if this performance carried over to the more complex task of experimental binding prediction using a dataset of 589 docked structures with empirical binding data (see supporting methods). The first approach was to evaluate the experimental ligand with the trained PDBbind refined model; however, we saw no ability for the model to predict experimental binding affinity (**Figure S1-2**). Interestingly, we saw that finetuning from the D4 dataset to PDBbind dataset also was not helpful, suggesting that the molecular representations learned for each task are likely quite distinct (**Figure S3**). Additionally, when we evaluated the experimental set using our optimized model from the D4 docking score prediction task we saw no discrimination between binders and non-binders; this is not surprising given the binders and non-binders had generally similar docking score distributions. Finally, we tried to use a metric learning approach to distinguish binders from non-binders. Although we saw improvements compared to PLEC (**Figure S4-5**), proximity edges did not appear to improve performance compared to the ligand autonomous graph networks.

Controls for shortcut learning. Three different experiments explored the importance of (i) having a learnable representation, (ii) message passing between the protein and ligand, and (iii) adding receptor information to the graph. To address (i), we used a frozen MPNN control, where encoder weights were set randomly before training. To investigate (ii), a Proximity Graph was stripped of all proximity edges (i.e., only the ligand and protein covalent bonds associated edges remained in the graph). Finally, to investigate (iii), the proximity graph was stripped of protein and proximity data. All adversarial controls for shortcut learning,[54] aside from the GGNET ligand-only and proximity-edge ablation studies for PDBbind datasets, significantly negatively affected performance, as intended (**Figure 5**). Additionally, we see no clear systematic bias relating error to the number/type of interactions or proximity-graph complexity (**SI Figure 6**).

The GGNET models on PDBbind performed poorly, so we do not believe the failure of these controls to be relevant in general. Removing proximity edges had a similar effect as the ligand-only control, suggesting that proximity edges were crucial for model performance.

Conclusions and Future Work

MPNN-based molecular encodings promise a tunable representation that can suit any task through gradient descent. This allure has spurred much interest in applying graph models to various computational chemistry tasks. In this work, we show certain datasets are much more suited to MPNN-based models than others and that encoder architectures can have variable performance based on the character of the dataset used for training. In particular, we introduce the D4 Dopamine dock score prediction task, consisting of diverse ligands bound to one receptor. This task benefits from the MPNNs' tunable representation more than the common PDBbind task used in most previous work developing MPNN scoring functions (SFs). We believe this is due to the accuracy of the labels, the abundance of diverse data, and the need to identify and discriminate fined-grained differences between many seemingly similar binding surfaces.

In addition, we show that incorporating proximity information to conventional MPNN architectures offers significant performance advantages. In all but one case, PDBbind General with Random Splitting, one of the MPNN models performed as good or better than the conventional fingerprint baseline. The performance improvement was particularly significant for the D4 Diverse Docking dataset. Despite strong performance as an ensemble, the graph networks were not suited to all tasks equally, showing the importance of diverse message-passing architectures for optimal performance on multiple applications.

Despite our extensive evaluation of this approach, there are several opportunities for future work. The most obvious area for improvement of the PGN package would be including more diverse graph convolutional methods, such as Deep Tensor Networks and other related architectures that represent distance and angle information more natively.[15,55] Additionally, exploring different data augmentation techniques to improve model performance in low-data situations would be advantageous. Beyond simple improvements to the PGN package, we also envision that PGN could facilitate analyses of molecular dynamics simulations and assist virtual screen approaches.[23,24] Additionally, the strong generalization of our PGN models makes model fine-tuning for low-data situations another potential application that could aid drug discovery.[9,56]

Acknowledgments

This work was supported by CZI grant DAF2018-191905 (DOI 10.37921/550142lkcjzw) from the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation (funder DOI 10.13039/100014989) (M.J.K.) and by the UCSF Program for Breakthrough Biomedical Research, funded in part by the Sandler Foundation (M.J.K.). We would like to thank Jessica McKinley for help with project administration, Matt O'Meara for data curation support, Alexandre Fassio for sharing data curation resources, Jason E. Gestwicki for mentoring, and Jennifer Huber for her support writing this manuscript.

Author Contributions

Zachary J. Gale-Day: Conceptualization; Methodology; Software; Data Curation; Writing - Original Draft; Visualization. **Laura Shub**: Software; Validation. **Kangway Chuang**: Resources; Methodology. **Michael J. Keiser**: Conceptualization; Writing - Review & Editing; Supervision; Project Administration; Funding acquisition.

Data and Software Availability

The complete source code and fully trained models are available at: https://github.com/keiserlab/pgn. This repository includes all package components and several scripts for common tasks and usage for PGN.

Competing Interests

The authors declare no competing interests.

Tables

Table 1. Atom features.

Feature	**Description**	**Size**
Atomic #	Atom type, indexed by atomic number	100
Isotope id	Type of isotope	1
Degree	Number of non-hydrogen neighbors	1
Formal charge	The formal charge of the atom	1
Is ring	0/1 atom is in ring	1
Is aromatic	0/1 atom in aromatic ring	1
Group	0 from receptor/1 ligand	1

Table 2. Bond features.

Feature	**Description**	**Size**
Bond Length	Distance between connecting nodes	1
Bond type	One-hot encoding of bond order	3
Is aromatic	0/1 aromatic bond	1
Is Proximity	0/1 proximity edge	1

Figure Legends

Figure 1. The process used to create Proximity Graphs and learn using the PGN architecture. The top portion of each panel (a-c) shows the processing of a simple example ligand-receptor complex (5OU2). The bottom panel shows a simplified example on a single atom. (a) A 4.5 Å radius sphere (translucent green) around the ligand (dark green) was used to filter the protein for proximal atoms. (b) The proximal protein atoms (dark blue) and the connecting bonds were added to the graph and proximity edges were added to the graph (gray). (c) Proteins atoms within 5 bonds of proximal protein atoms were added to the graph. An additional example of a more complex protein ligand complex (6MNF) is shown below. (d) Once the proximity graph is constructed a simple featurization is applied to the atoms and edges. This completed proximity graph data structure is fed into one of the included MPNN architectures to encode the graphs before being passed to the FC network to produce the desired regression or classification output. (e) The PDBbind (left) and D4 dock-score (right) regression tasks are summarized. PDBbind contains the experimental pKis paired with X-ray diffraction structures for many proteins. In contrast, the dock-score prediction task pairs the protein structure from 5WIU with a number of predicted protein-ligand complexes with molecules in the ZINC database.

Figure 2. Performance of the different models on the PDBBind Refined dataset. (a) RMSE of each model with random splitting (blue) and protein splitting (orange),where lower is better. (b) PCC of each model with random splitting (blue) and protein splitting (orange), where higher is better. (c) Table or errors and correlation values for each model and split with RMSE and PCC. Best scoring model is bolded.

Figure 3. Performance of the different models on the PDBBind General dataset. (a) RMSE of each model with random splitting (blue) and protein splitting (orange), where lower is better. (b) PCC of each model with random splitting (blue) and protein splitting (orange), where higher is better. (c) Table or errors and correlation values for each model and split with RMSE and PCC. Best scoring model is bolded.

Figure 4. Overview of the construction of the D4 Diverse Dataset and performance of the different models on the D4 Diverse dataset. (a) Overview of the original protein-ligand complex used to construct the docking model. The inset shows the proximity graph of the experimental

ligand bound into the pocket used for docking. (b) Heatmap showing the similarity of 1000 random ligands selected from the whole dock run and the D4 Diverse dataset used in this manuscript. (c) Table or errors and correlation values for each model and split with RMSE and PCC. Best scoring model is bolded. (d) RMSE of each model with random splitting (blue) and protein splitting (orange); lower is better. (e) PCC of each model with random splitting (blue) and protein splitting (orange; higher is better. (f) RMSE of models for various dataset sizes; lower is better.

Figure 5. Performance of the different models and the various controls on the PDBbind Refined, PDBbind Genera,l and D4 Diverse datasets. All control RMSEs were normalized to the performance of the best model using the full Proximity Graph as input. For RMSE, a lower value is better. Each panel is the full set of controls for each MPNN architecture: (a) PFP, (b) GGNET, and (c) DMPNN.

Figures

Figure 1

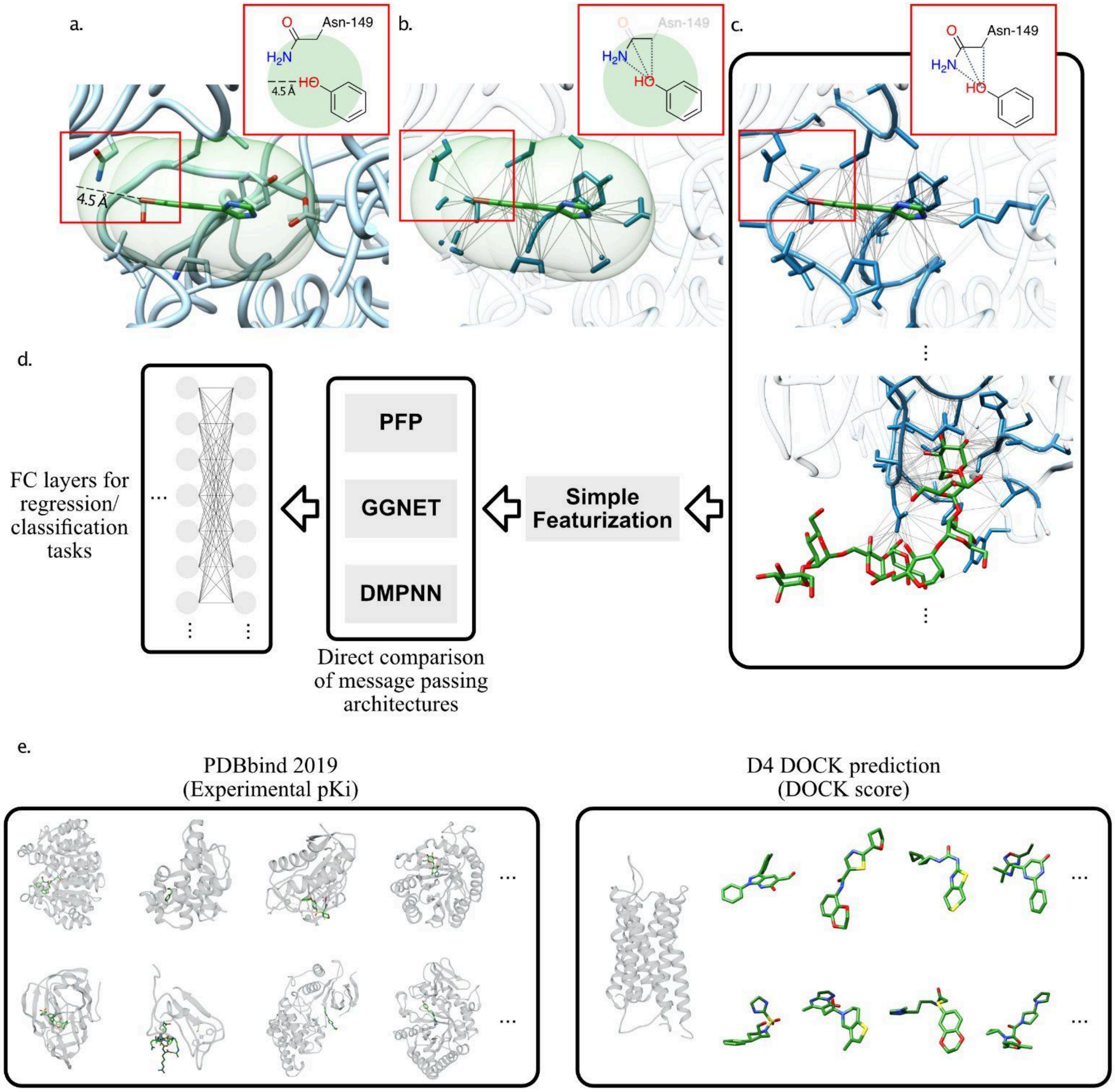
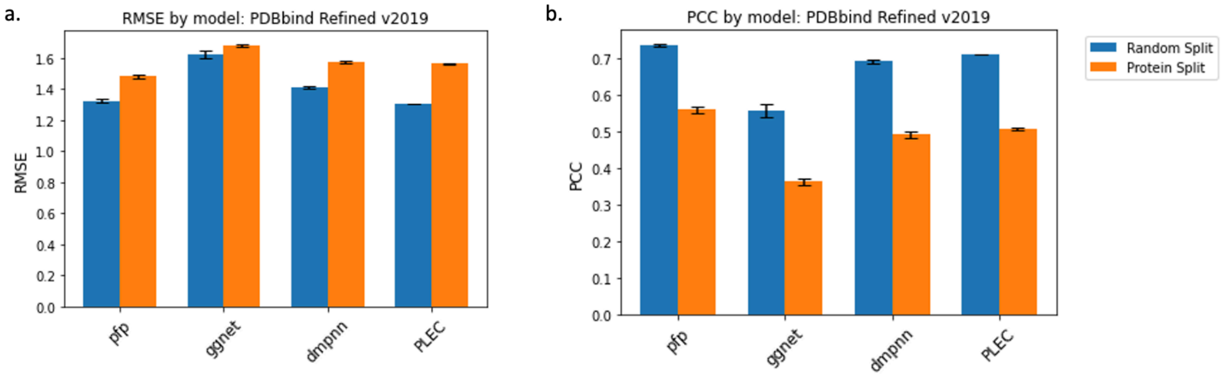

a.
Asn-149
O
H_2N
4.5 Å
HO
b.
Asn-149
c.
Asn-149
d.
FC layers for regression/ classification tasks
PFP
GGNET
DMPNN
Simple Featurization
Direct comparison of message passing architectures
e.
PDBbind 2019 (Experimental pKi)
D4 DOCK prediction (DOCK score)

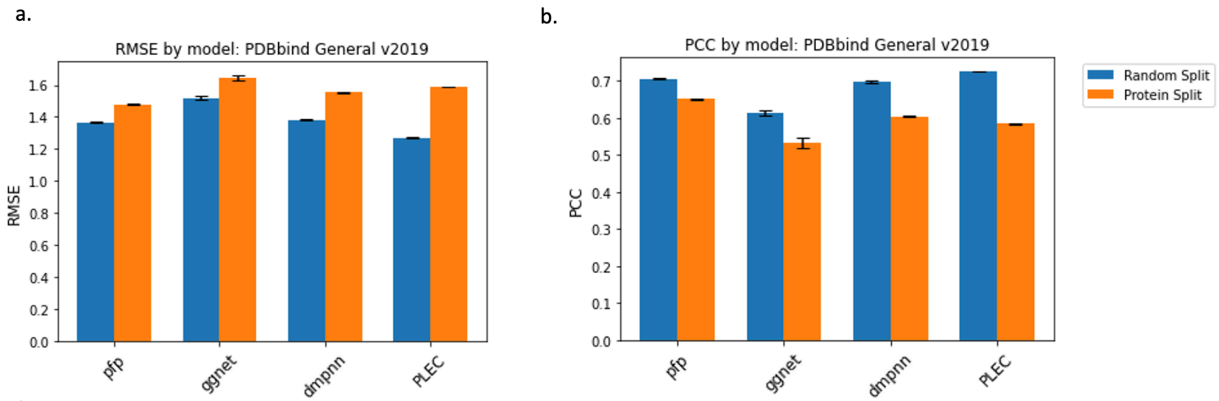
Figure 2



c.

Model	Random Split (RMSE)	Protein Split (RMSE)	Random Split (PCC)	Protein Split (PCC)
PFP	1.323 ± 0.010	**1.480 ± 0.010**	**0.735 ± 0.005**	**0.559 ± 0.008**
GGNET	1.617 ± 0.024	1.688 ± 0.010	0.562 ± 0.018	0.351 ± 0.008
DMPNN	1.412 ± 0.001	1.572 ± 0.009	0.691 ± 0.005	0.491 ± 0.009
PLEC	**1.304 ± 0.003**	1.562 ± 0.004	0.711 ± 0.001	0.507 ± 0.004

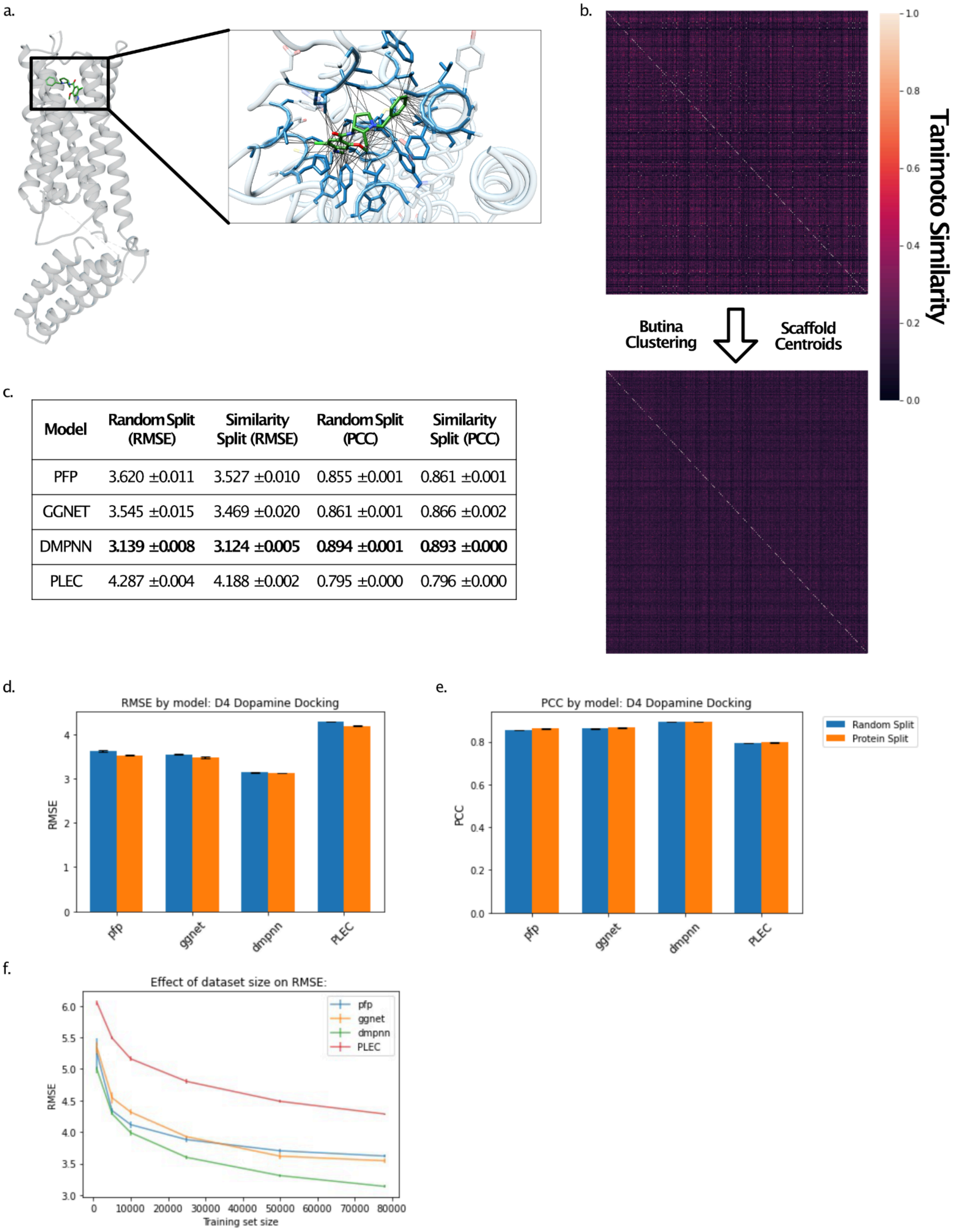
Figure 3



c.

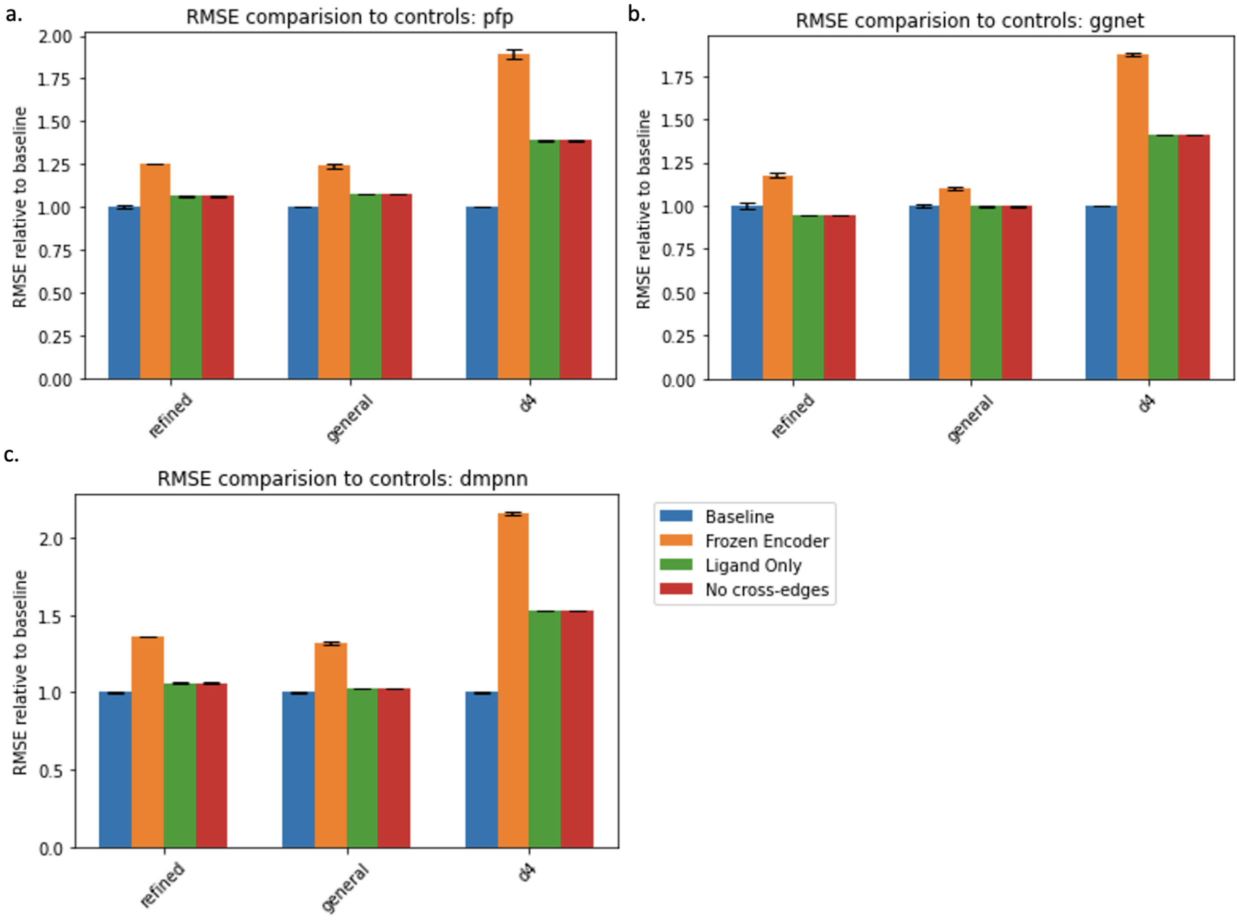
Model	Random Split (RMSE)	Protein Split (RMSE)	Random Split (PCC)	Protein Split (PCC)
PFP	1.364 ± 0.004	**1.480 ± 0.004**	0.706 ± 0.002	**0.651 ± 0.001**
GGNET	1.488 ± 0.012	1.647 ± 0.016	0.631 ± 0.008	0.532 ± 0.013
DMPNN	1.383 ± 0.005	1.554 ± 0.003	0.699 ± 0.003	0.605 ± 0.003
PLEC	**1.270 ± 0.003**	1.587 ± 0.003	**0.726 ± 0.001**	0.584 ± 0.002

Figure 4



Model	Random Split (RMSE)	Similarity Split (RMSE)	Random Split (PCC)	Similarity Split (PCC)
PFP	3.620 ±0.011	3.527 ±0.010	0.855 ±0.001	0.861 ±0.001
GGNET	3.545 ±0.015	3.469 ±0.020	0.861 ±0.001	0.866 ±0.002
DMPNN	**3.139 ±0.008**	**3.124 ±0.005**	**0.894 ±0.001**	**0.893 ±0.000**
PLEC	4.287 ±0.004	4.188 ±0.002	0.795 ±0.000	0.796 ±0.000

Figure 5



References

(1) D'Souza, S.; Prema, K. V.; Balaji, S. Machine Learning Models for Drug-Target Interactions: Current Knowledge and Future Directions. *Drug Discov. Today* **2020**, *25* (4), 748–756.

(2) Askr, H.; Elgeldawi, E.; Aboul Ella, H.; Elshaier, Y. A. M. M.; Gomaa, M. M.; Hassanien, A. E. Deep Learning in Drug Discovery: An Integrative Review and Future Challenges. *Artif Intell Rev* **2023**, *56* (7), 5975–6037.

(3) Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; Zhao, S. Applications of Machine Learning in Drug Discovery and Development. *Nat. Rev. Drug Discov.* **2019**, *18* (6), 463–477.

(4) Dara, S.; Dhamercherla, S.; Jadav, S. S.; Babu, C. M.; Ahsan, M. J. Machine Learning in Drug Discovery: A Review. *Artif Intell Rev* **2022**, *55* (3), 1947–1999.

(5) Volkov, M.; Turk, J.-A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D. On the Frustration to Predict Binding Affinities from Protein-Ligand Structures with Deep Neural Networks. *J. Med. Chem.* **2022**, *65* (11), 7946–7958.

(6) Stokes, J. M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N. M.; MacNair, C. R.; French, S.; Carfrae, L. A.; Bloom-Ackerman, Z.; Tran, V. M.; Chiappino-Pepe, A.; Badran, A. H.;

Andrews, I. W.; Chory, E. J.; Church, G. M.; Brown, E. D.; Jaakkola, T. S.; Barzilay, R.; Collins, J. J. A Deep Learning Approach to Antibiotic Discovery. *Cell* **2020**, *180* (4), 688–702.e13.

(7) Cho, H.; Lee, E. K.; Choi, I. S. Layer ‑ Wise Relevance Propagation of InteractionNet Explains Protein – Ligand Interactions at the Atom Level. *Sci. Rep.* **2020**, 19–23.

(8) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. *34th International Conference on Machine Learning, ICML 2017* **2017**, *3*, 2053–2070.

(9) Aguilera-Iparraguirre, D. D. D. M.; Rafael Gómez-Bombarelli, Timothy Hirzel, Al´an Aspuru-Guzik, Ryan P Adams Harvard. Convolutional Networks on Graphs for Learning Molecular Fingerprints. *J. Chem. Inf. Model.* **2016**, *56* (2), 399–411.

(10) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **2019**, *59* (8), 3370–3388.

(11) Kearnes, S.; Mccloskey, K.; Berndl, M.; Pande, V.; Riley, P. *Molecular Graph Convolutions: Moving Beyond Fingerprints*. https://arxiv.org/pdf/1603.00856.pdf.

(12) Altae-Tran, H.; Ramsundar, B.; Pappu, A. S.; Pande, V. Low Data Drug Discovery with One-Shot Learning. *ACS Central Science* **2017**, *3* (4), 283–293.

(13) Ramsundar, B.; Kearnes, S.; Riley, P.; Webster, D.; Konerding, D.; Pande, V. Massively Multitask Networks for Drug Discovery. *ICML* **2015**.

(14) Feinberg, E. N.; Joshi, E.; Pande, V. S.; Cheng, A. C. Improvement in ADMET Prediction with Multitask Deep Featurization. *J. Med. Chem.* **2020**, *63* (16), 8835–8848.

(15) Klicpera, J.; Groß, J.; Günnemann, S. Directional Message Passing for Molecular Graphs. *ILCR* **2020**, 1–13.

(16) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science* **2018**, *4* (2), 268–276.

(17) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.* **2018**, *9* (2), 513–530.

(18) Feinberg, E. N.; Sur, D.; Wu, Z.; Husic, B. E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V. S. PotentialNet for Molecular Property Prediction. **2018**. https://doi.org/10.1021/acscentsci.8b00507.

(19) 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* (2), 895–913.

(20) Wang, R.; Fang, X.; Lu, Y.; Wang, S. The PDBbind Database: Collection of Binding Affinities for Protein-Ligand Complexes with Known Three-Dimensional Structures. *J. Med. Chem.* **2004**, *47* (12), 2977–2980.

(21) Wang, R.; Fang, X.; Lu, Y.; Yang, C. Y.; Wang, S. The PDBbind Database: Methodologies and Updates. *J. Med. Chem.* **2005**, *48* (12), 4111–4119.

(22) Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K.; Tolmachev, A. A.; Shoichet, B. K.; Roth, B. L.; Irwin, J. J. Ultra-Large Library Docking for Discovering New Chemotypes. *Nature* **2019**, *566* (7743), 224–229.

(23) Gentile, F.; Agrawal, V.; Hsing, M.; Ton, A. T.; Ban, F.; Norinder, U.; Gleave, M. E.; Cherkasov, A. Deep Docking: A Deep Learning Platform for Augmentation of Structure Based Drug Discovery. *ACS Central Science* **2020**, *6* (6), 939–949.

(24) Yang, Y.; Yao, K.; Repasky, M. P.; Leswing, K.; Abel, R.; Shoichet, B. K.; Jerome, S. V. Efficient Exploration of Chemical Space with Docking and Deep Learning. *J. Chem. Theory Comput.* **2021**, *17* (11), 7106–7119.

(25) Wójcikowski, M.; Kukiełka, M.; Stepniewska-Dziubinska, M. M.; Siedlecki, P. Development of a Protein-Ligand Extended Connectivity (PLEC) Fingerprint and Its Application for Binding Affinity Predictions. *Bioinformatics* **2019**, *35* (8), 1334–1341.

(26) Fassio, A. V.; Shub, L.; Ponzoni, L.; McKinley, J.; O'Meara, M. J.; Ferreira, R. S.; Keiser, M. J.; de Melo Minardi, R. C. Prioritizing Virtual Screening with Interpretable Interaction Fingerprints. *J. Chem. Inf. Model.* **2022**, *62* (18), 4300–4318.

(27) Sánchez-Cruz, N.; Medina-Franco, J. L.; Mestres, J.; Barril, X. Extended Connectivity Interaction Features: Improving Binding Affinity Prediction through Chemical Description. *Bioinformatics* **2020**, 1–7.

(28) Halgren, T. A.; Murphy, R. B.; Friesner, R. A.; Beard, H. S.; Frye, L. L.; Pollard, W. T.; Banks, J. L. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 2. Enrichment Factors in Database Screening. *J. Med. Chem.* **2004**, *47* (7), 1750–1759.

(29) Ballester, P. J.; Mitchell, J. B. O. A Machine Learning Approach to Predicting Protein-Ligand Binding Affinity with Applications to Molecular Docking. *Bioinformatics* **2010**, *26* (9), 1169–1175.

(30) Durrant, J. D.; Mccammon, J. A. NNScore 2 . 0 : A Neural-Network Receptor À Ligand Scoring Function. **2011**, 2897–2903.

(31) Li, Y.; Zemel, R.; Brockschmidt, M.; Tarlow, D. Gated Graph Sequence Neural Networks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings* **2016**, No. 1, 1–20.

(32) Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* **2017**, 1–14.

(33) Cang, Z.; Wei, G. TopologyNet: Topology Based Deep Convolutional and Multi-Task Neural Networks for Biomolecular Property Predictions. *PLoS Comput. Biol.* **2017**, *13* (7), 1–27.

(34) Li, Y.; Rezaei, M. A.; Li, C.; Li, X. DeepAtom: A Framework for Protein-Ligand Binding Affinity Prediction. *Proceedings - 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019* **2019**, 303–310.

(35) Zheng, L.; Fan, J.; Mu, Y. OnionNet: A Multiple-Layer Intermolecular-Contact-Based Convolutional Neural Network for Protein-Ligand Binding Affinity Prediction. *ACS Omega* **2019**, *4* (14), 15956–15965.

(36) Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P. Development and Evaluation of a Deep Learning Model for Protein–ligand Binding Affinity Prediction. *Bioinformatics* **2018**, *34* (21), 3666–3674.

(37) Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein-Ligand Scoring with Convolutional Neural Networks. *J. Chem. Inf. Model.* **2017**, *57* (4), 942–957.

(38) Li, S.; Zhou, J.; Xu, T.; Huang, L.; Wang, F.; Xiong, H.; Huang, W.; Dou, D.; Xiong, H. *Structure-Aware Interactive Graph Neural Networks for the Prediction of Protein-Ligand Binding Affinity*; ACM, 2021.

(39) Knutson, C.; Bontha, M.; Bilbrey, J. A.; Kumar, N. Decoding the Protein–ligand Interactions Using Parallel Graph Neural Networks. *Sci. Rep.* **2022**, *12* (1), 7624.

(40) Bergstra, J.; Yamins, D.; Cox, D. D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *Proc. of the 30th International Conference on Machine Learning ICML* **2013**, I115–I123.

(41) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv* **2019**, No. NeurIPS.

(42) Fey, M.; Lenssen, J. E. Fast Graph Representation Learning with Pytorch Geometric. *arXiv* **2019**, No. 1, 1–9.

(43) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminform.* **2011**, *3* (10), 1–14.

(44) Wójcikowski, M.; Zielenkiewicz, P.; Siedlecki, P. Open Drug Discovery Toolkit (ODDT): A New Open-Source Player in the Drug Discovery Field. *J. Cheminform.* **2015**, *7* (1), 1–6.

(45) Hagberg, A. A.; Schult, D. A.; Swart, P. J. Exploring Network Structure, Dynamics, and Function

Using NetworkX. *7th Python in Science Conference (SciPy 2008)* **2008**, No. SciPy, 11–15.

(46) Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E. UCSF Chimera - A Visualization System for Exploratory Research and Analysis. *J. Comput. Chem.* **2004**, *25* (13), 1605–1612.

(47) 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* (2), 302–309.

(48) Sterling, T.; Irwin, J. J. ZINC 15 - Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55* (11), 2324–2337.

(49) Coleman, R. G.; Carchia, M.; Sterling, T.; Irwin, J. J.; Shoichet, B. K. Ligand Pose and Orientational Sampling in Molecular Docking. *PLoS One* **2013**, *8* (10), e75992.

(50) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39* (15), 2887–2893.

(51) Butina, D. Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way to Cluster Small and Large Data Sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39* (4), 747–750.

(52) Li, Y.; Han, L.; Liu, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 2. Evaluation Methods and General Results. *J. Chem. Inf. Model.* **2014**, *54* (6), 1717–1736.

(53) Cheng, T.; Li, X.; Li, Y.; Liu, Z.; Wang, R. Comparative Assessment of Scoring Functions on a Diverse Test Set. *J. Chem. Inf. Model.* **2009**, *49* (4), 1079–1093.

(54) Chuang, K. V.; Keiser, M. J. Adversarial Controls for Scientific Machine Learning. *ACS Chem. Biol.* **2018**, *13* (10), 2819–2821.

(55) Schütt, K. T.; Kindermans, P. J.; Sauceda, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K. R. SchNet: A Continuous-Filter Convolutional Neural Network for Modeling Quantum Interactions. *Adv. Neural Inf. Process. Syst.* **2017**, *2017-Decem* (1), 992–1002.

(56) Navarin, N.; Tran, D. V.; Sperduti, A. Pre-Training Graph Neural Networks with Kernels. *arXiv* **2018**, No. Nips.