Title

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

Journal

Journal of chemical information and computer sciences, 64(14)

Authors

Gale-Day, Zachary
Shub, Laura
Chuang, Kangway
et al.

Publication Date

2024-07-22

DOI

10.1021/acs.jcim.4c00311

Peer reviewed

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

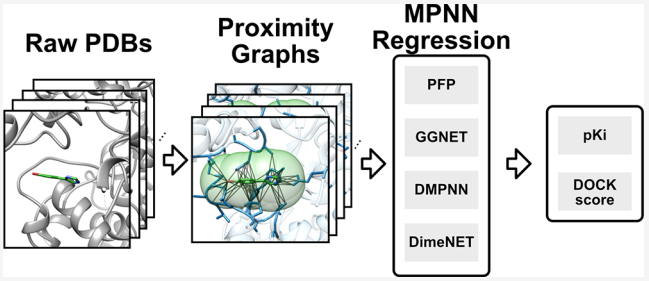
pubs.acs.org/jcim
Article

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

Zachary J. Gale-Day, Laura Shub, Kangway V. Chuang, and Michael J. Keiser*

Cite This: J. Chem. Inf. Model. 2024, 64, 5439−5450

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 proximity 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 two-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 study introduces proximity graph networks (PGNs), an open-source toolkit (https://github.com/keiserlab/torch_pgn) that allows for the 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 data sets.[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

Received: February 22, 2024
Revised: June 4, 2024
Accepted: June 24, 2024
Published: July 2, 2024

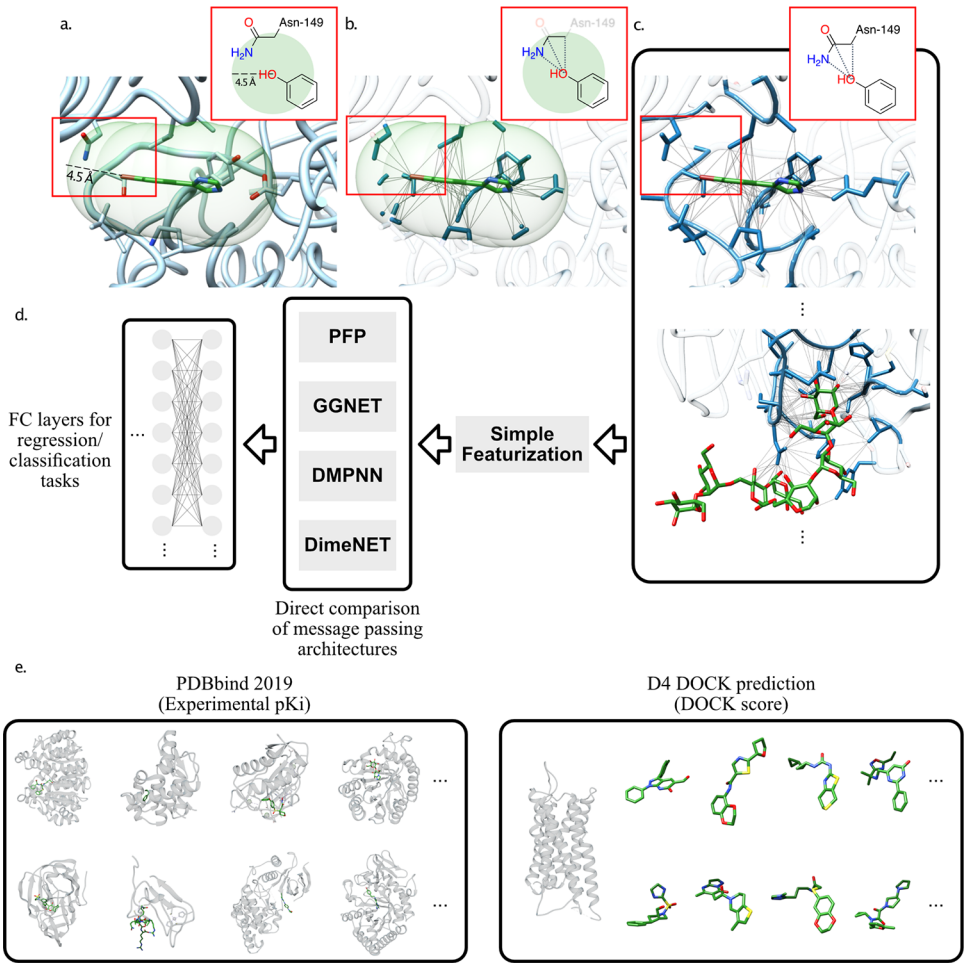


Figure 1. Process used to create Proximity Graphs and learn using the PGN architecture. The top portion of each panel (a–c) shows the processing of a simple example ligand–receptor complex (5OU2). The bottom panel shows a simplified example on a single atom. (a) 4.5 Å radius sphere (translucent green) around the ligand (dark green) was used to filter the protein for proximal atoms. (b) 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 fully connected (FC) network to produce the desired regression or classification output. (e) 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.

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 chem-informatics 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 the implicit graph constructed in the protein–ligand extended connectivity (PLEC) implementation of 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 data set, 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 studies, this study introduces PGN for optimized graph neural networks. It provides an open-source implementation of MPNN architectures based on

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450

gated-graph neural networks (GGNET),[31] directed MPNNs (DMPNN),[10] and the equivariant graph network DimeNET++[40] 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 proximity fingerprint (PFP) architecture implemented in our PGN software package. Please see the Supplementary Methods section for descriptions of previously reported architectures GGNET[31] and DMPNN[10]). For a description of DimeNET++, please refer to the study of Gasteiger et al.[40]

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 which 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 called 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^t at 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

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

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

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:

$$\hat{y} = R(h_v^T | v \in \{G\})$$

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 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:

$$M_t(h_v, h_w, e_{vw}) = A_{e_{vw}} h_w$$

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:

$$U_t = U = \mathrm{ReLU}(m_v^{t+1})$$

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 time step t to ensure that all levels contribute to the output representation $\hat{y}$. Therefore, we define R_t as the readout at time step t and R as the overall readout function. During each time step, 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:

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

The final output ($\hat{y}$) 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. We added the protein atoms within a sphere of 4.5 Å radius from each heavy atom to match the optimal parameters for PDBBind determined with PLEC (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 four 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[41] 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

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

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450

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

$\hat{y}$ vector, depending on the model) for PFP, GGNET, DMPNN, and PLEC models. Due to its slower training speed, DimeNET++ models were optimized using Optuna[42] with trials running in parallel on multiple GPUs.

Implementation. We implemented all models in PyTorch[43] and PyTorch Geometric.[44] Molecular data processing used Open Babel,[45] RDKit, and ODDT.[46] For basic graph data structures, we used NetworkX.[47] We adapted the DMPNN code from the chemprop repository[10] and DimeNET++ code from the PyTorch Geometric implementation to work with proximity graph data structures. We visualized structures with Chimera.[48]

EXPERIMENTS

Data. We tested our models on the PDBbind 2019 Refined, PDBbind 2019 General, and D4 Diverse Docking data sets:

PDBbind Refined Data 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 data set consists of 4852 ligand−receptor complexes.

PDBbind General Data Set. The general set includes all 21,382 structures in the PDBbind database.[20,49] However, we only included protein−ligand complexes for this study, thus narrowing the set to 17,679 structures. We employed no extra filtering or other manipulation.

D4 Diverse Docking Data Set. We introduce a docking score prediction task for the Dopamine D4 receptor. For this task, our data set includes 86,452 ligands from the ZINC database[50] docked in the Dopamine D4 Receptor from an ultra-large library[22,51] (ULL) docking campaign. The compounds all represent different structural scaffolds as determined by Bemis-Murcko scaffold splitting.[52] The structures were annotated with the docking score.

D4 Experimental Data Set. This data set 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 data set for classification; docked structures were either binders or nonbinders. Additionally, we used the D4 Experimental data set for our metric learning task discussed below.

Experimental Procedure. *Cross-Validation and Hyperparameter Optimization.* Each architecture type was optimized individually for each data set 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 five-fold cross-validation with a given set of hyperparameters. Due to significantly longer training times than other architectures, we ran hyperparameter optimization for the PDBbind General and D4 Diverse data sets with three-fold cross-validation for DMPNN and all data sets for DimeNET++. Before cross-validation, we held out a test set of approximately 10% of examples. The 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. The random split randomly sorted examples 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 data set, a standard scaffold split yields no clusters due to how the data set was constructed. Therefore, to maximize the difference between the ligands in the training and test sets, Butina clustering[53] of Morgan 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 data sets both the root-mean-square error (RMSE) and the Pearson correlation coefficient (PCC) were used for evaluation, in line with CASF[19,54,55] recommendations for standard PDBbind evaluation.

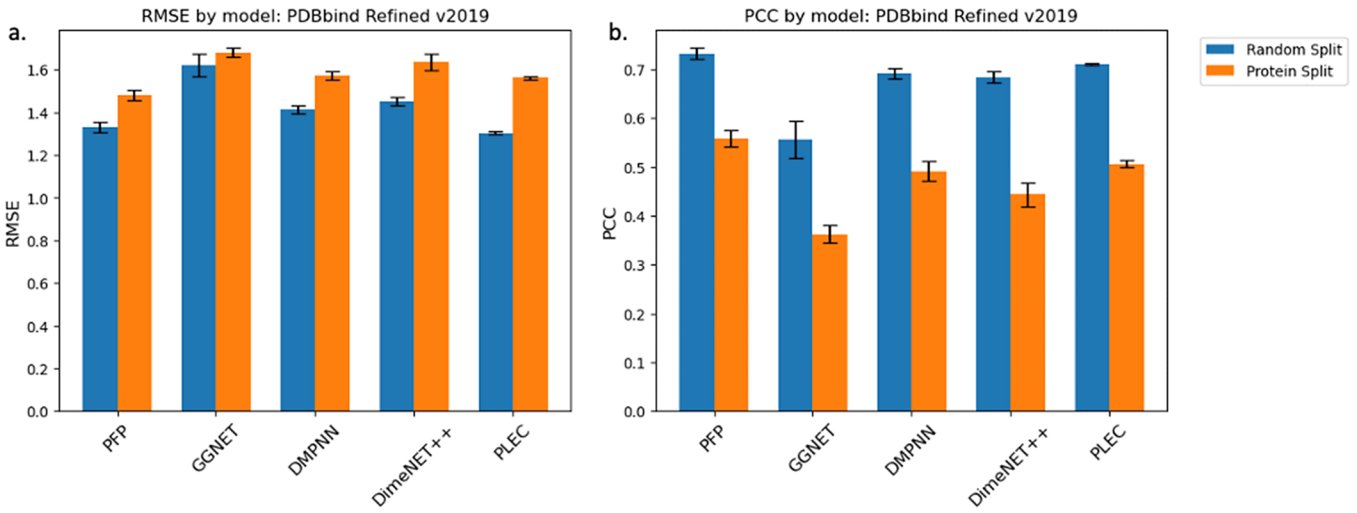
Baseline. PLEC uses the graph generation procedure outlined above; however, instead of creating an explicit proximity graph data structure as outlined in this work, PLEC passes the results through a hash function to create a static fingerprint data structure. Given the similarity of PLEC's implicit graph to the proximity graph data structure, we used PLEC as the primary baseline model to determine how a differentiable and explicit molecular interface graph representation contributes to model performance. 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 nonlearnable, 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 nonligand atoms and edges.

RESULTS AND DISCUSSION

In this section, we evaluate the methods contained in the PGN package (PFP, GGNET, DMPNN, and DimeNET++) by performing experiments to evaluate (1) whether MPNNs would offer advantages over fingerprint and other baseline

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450



c.

Model	Random Split (RMSE)	Protein Split (RMSE)	Random Split (PCC)	Protein Split (PCC)
PFP	1.323 ± 0.024	**1.480 ± 0.022**	**0.735 ± 0.011**	**0.559 ± 0.017**
GGNET	1.617 ± 0.051	1.688 ± 0.020	0.562 ± 0.038	0.351 ± 0.017
DMPNN	1.412 ± 0.020	1.572 ± 0.019	0.691 ± 0.010	0.491 ± 0.020
DimeNET++	1.453 ± 0.019	1.635 ± 0.038	0.684 ±0.011	0.444 ± 0.024
PLEC	**1.304 ± 0.006**	1.562 ± 0.008	0.711 ± 0.002	0.507 ± 0.008

Figure 2. Performance of the different models on the PDBBind Refined data set. (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.

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 Docking data sets and evaluated performance based on RMSE and PCC. PCC is more relevant for rank-ordering compounds for testing. In contrast, RMSE is more relevant for predicting raw binding energies in isolation; therefore, including both gives a better picture of usefulness than either alone. Additionally, these metrics are the standard for the PDBBind data set,[47] allowing for easy comparison with past and future work. Error bars shown in the text represent the 95% confidence interval for the given experiment.

Unless otherwise stated, all results below use the optimal hyperparameters (see Tables S1−S18) from the hyperparameter search. All results reported for graph models use the complete Proximity Graph data structure, unless explicitly stated otherwise.

Performance on PDBbind Data Sets. First, we evaluated each model on the PDBbind Refined and General data sets. We tested all models with both random splitting and protein splitting.

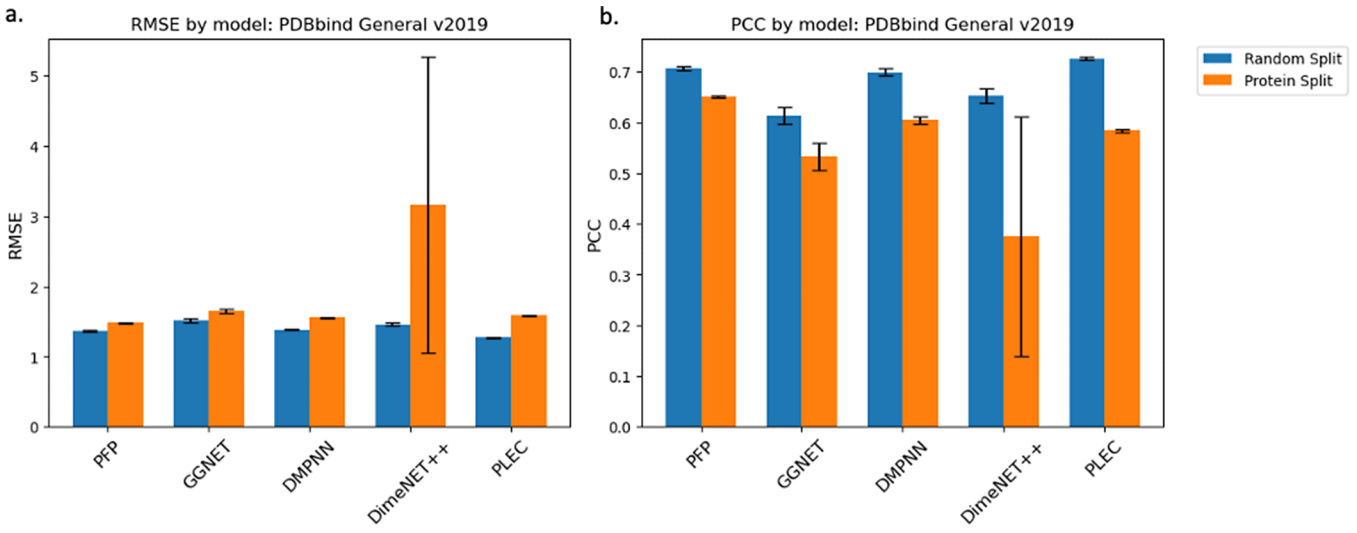
PDBbind Refined Data Set. Figure 2 shows the results of model evaluation on the refined data set. The PFP encoder-based models performed similarly to or significantly better than PLEC in all cases. All other MPNN approaches underperformed on this task. This suggests that although a differentiable representation can be useful, the specific encoder architecture is an important consideration, highlighting the usefulness of easy access to multiple encoder choices in the PGN package.

Interestingly, the PFP model trained with random splitting consistently outperformed the baseline by PCC metric but had a similar RMSE to PLEC. 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 S19).

PDBbind General Data Set. 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, while DimeNET++ showed a high level of instability in predictions, suggesting that overfitting occurs early in training.

Performance on the D4 Diverse Data Set. Next, we evaluated the performance of the different models on the D4 Diverse data set, 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

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450



c.

Model	Random Split (RMSE)	Protein Split (RMSE)	Random Split (PCC)	Protein Split (PCC)
PFP	1.364 ± 0.009	**1.480 ± 0.008**	0.706 ± 0.004	**0.651 ± 0.003**
GGNET	1.488 ± 0.026	1.647 ± 0.033	0.631 ± 0.017	0.532 ± 0.027
DMPNN	1.383 ± 0.010	1.554 ± 0.007	0.699 ± 0.007	0.605 ± 0.007
DimeNET++	1.458 ± 0.024	3.158 ± 2.105	0.653 ± 0.015	0.375 ± 0.236
PLEC	**1.270 ± 0.006**	1.587 ± 0.003	**0.726 ± 0.003**	0.584 ± 0.002

Figure 3. Performance of the different models on the PDBBind General data set. (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.

above. This data set offers a different task type than the PDBbind data set, 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 to produce a sensitive readout of protein−ligand complementarity. This task is analogous to "deep docking" where a neural network predicts dock score (usually based on fingerprint representations), which is typically far less computationally expensive than traditional docking algorithms.[23,24] Using this approximate dock score, the larger library is then accessed using the neural network to allow for downstream applications.

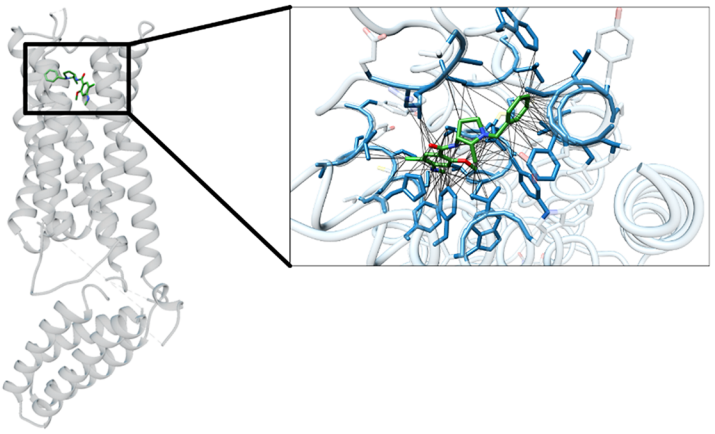
When assessed against the D4 Diverse data set, all graph models outperformed the baseline by a large margin (Figure 4c−e). In contrast to the PDBbind data sets, the DimeNET++ model performed best followed closely by DMPNN. GGNET and PFP lagged behind the other graph models significantly in this task. The similarity-split data resulted in models that performed no worse than the random split, likely due to the large ligand diversity already seen within the data set.

Next, we artificially limited the training set size and evaluated model performance using the full test set to understand if data set size would strongly affect relative model performance (Figure 4f). All MPNN architectures outperformed the baseline, regardless of the data set size. This result suggests that a learnable representation was particularly advantageous for this task.

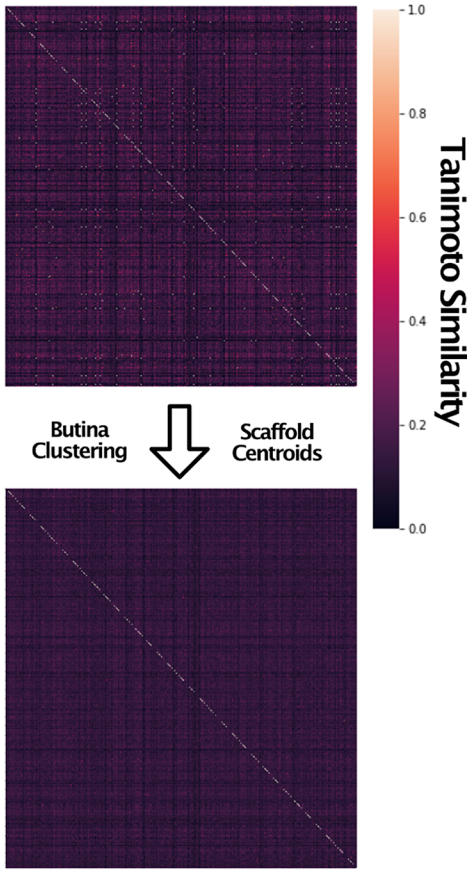
We were next interested in seeing if this performance carried over to the more complex task of experimental binding prediction using a data set of 589 docked structures with empirical binding data (see Supporting Methods). Although docking is effective at prioritizing binders in the better-scored poses, the extreme size of modern docking libraries makes filtering an important problem, and therefore a computational solution would be valuable.[22] The first approach was to evaluate the experimental ligands with the trained PDBbind refined model; however, we saw no ability for the model to predict experimental binding affinity (Figures S1 and S2). Interestingly, we saw that fine-tuning from the D4 data set to the PDBbind data set 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 nonbinders; this is not surprising, given the binders and nonbinders had generally similar docking score distributions. Given the difficulty of predicting experimental affinity, we explored simplifying the task to discriminate between binders and nonbinders using metric learning. Although we saw improvements compared to PLEC (Figures S4 and S5), proximity edges did not appear to improve performance compared to the ligand autonomous graph networks.

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450

a.



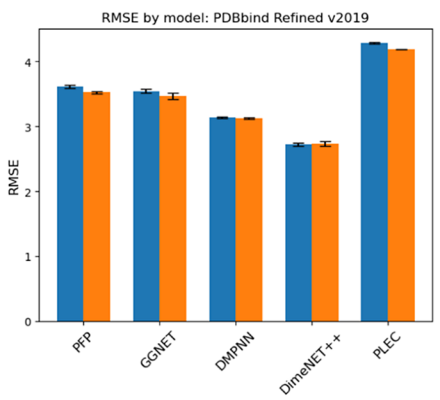
b.



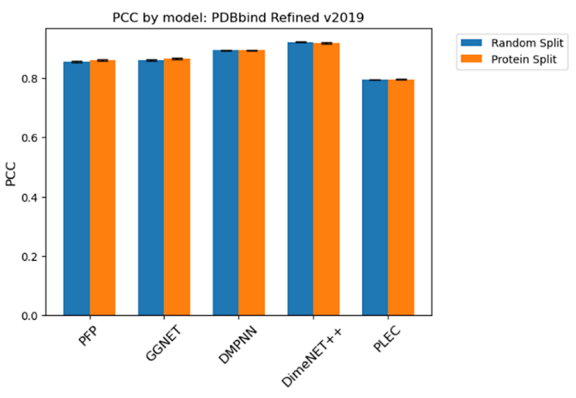
c.

Model	Random Split (RMSE)	Similarity Split (RMSE)	Random Split (PCC)	Similarity Split (PCC)
PFP	3.620 ±0.024	3.527 ±0.021	0.855 ±0.002	0.861 ±0.002
GGNET	3.545 ±0.032	3.469 ±0.044	0.861 ±0.003	0.866 ±0.004
DMPNN	3.139 ±0.017	3.124 ±0.010	0.894 ±0.001	0.893 ±0.001
DimeNET++	**2.721 ±0.025**	**2.733 ±0.037**	**0.921 ±0.001**	**0.919 ±0.003**
PLEC	4.287 ±0.008	4.188 ±0.005	0.795 ±0.000	0.796 ±0.000

d.



e.



f.

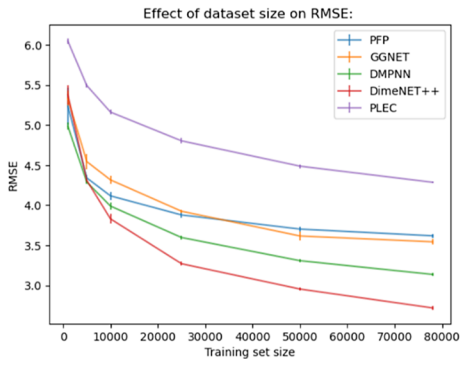


Figure 4. Overview of the construction of the D4 Diverse Data set and performance of the different models on the D4 Diverse data set. (a) Overview of the original protein−ligand complex used to construct the docking model. The inset shows the proximity graph of the experimental ligand bound into the pocket used for docking. (b) Heatmap showing the similarity of 1000 random ligands selected from the whole dock run and the D4 Diverse data set 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 data set sizes; lower is better.

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450

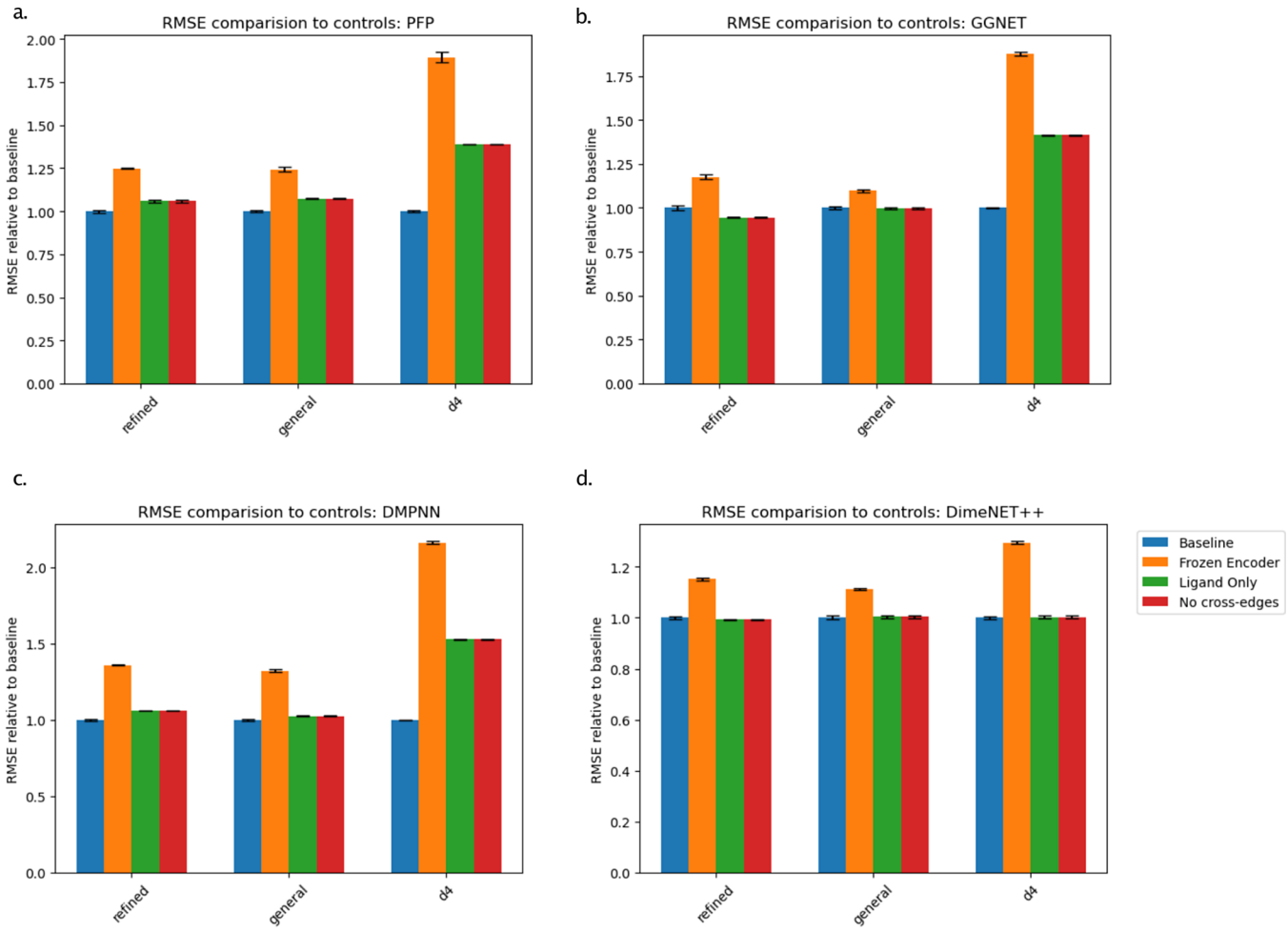


Figure 5. Performance of the different models and the various controls on the PDBbind Refined, PDBbind General, and D4 Diverse data sets. All control RMSEs were normalized to the performance of the best model using the full Proximity Graph as input. For RMSE, a lower value is better. Each panel is the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

Controls for Shortcut Learning. Three different experiments explored the importance of (i) using 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 with encoder weights randomly set before training. To investigate (ii), we stripped a proximity graph of all proximity edges (i.e., only the ligand and protein covalent bonds associated edges remained in the graph). Finally, to investigate (iii), we stripped the proximity graph of protein and proximity data. All adversarial controls for shortcut learning,[56] aside from the GGNET ligand-only and proximity-edge ablation studies for PDBbind data sets, significantly negatively affected performance for the 2D-MPNN architectures (PFP, DMPNN, GGNET) (Figure 5 and Table S20). Additionally, we saw no clear systematic bias relating error to the number or type of interactions or the proximity-graph complexity (SI Figure 6). On the other hand, the DimeNET++ ligand-only and proximity edge ablation studies performed as well as the full model on all data sets, suggesting that proximal residues do not contribute to its predictions. This is potentially due to the weighting of the DimeNET++ architecture based on distance, meaning ligand−ligand interactions could potentially dominate performance. This suggests that modifications to this architecture may be required to fully take advantage of receptor information.

Importantly, the frozen encoder controls in all cases severely impaired performance, suggesting that a learnable graph representation is crucial for model performance. Also, the ligand-only and proximity-edge stripped controls performed similarly to each other in all cases, suggesting that message-passing between the receptor and ligand is required for proximal protein atoms to contribute to the model.

Analysis of Proximity Graph Construction. We next sought to understand how proximity graph parameter choice affected the best-performing MPNN architectures that benefited from proximity graphs (PFP for PDBBind Refined and DMPNN for D4 Medium Diverse). We found that receptor depth minimally affected the PDBBind Refined architecture, which had a modest but significant increase in performance with increasing proximity radius that plateaued above 4.5 Å (Figure 6a, b). On the other hand, both parameters strongly affected D4 Medium Diverse data set performance, achieving a maximum when receptor depth was greater than three and radius was equal to 4.5 Å (Figure 6a, b). The large jump in performance at receptor depth 3 is interesting, suggesting that this depth potentially changes the graph's character by reconstituting fuller side chains or allowing more extensive message passing. These parameters are fully adjustable in the PGN package using the *proximity_radius* and *receptor_depth* arguments.

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450

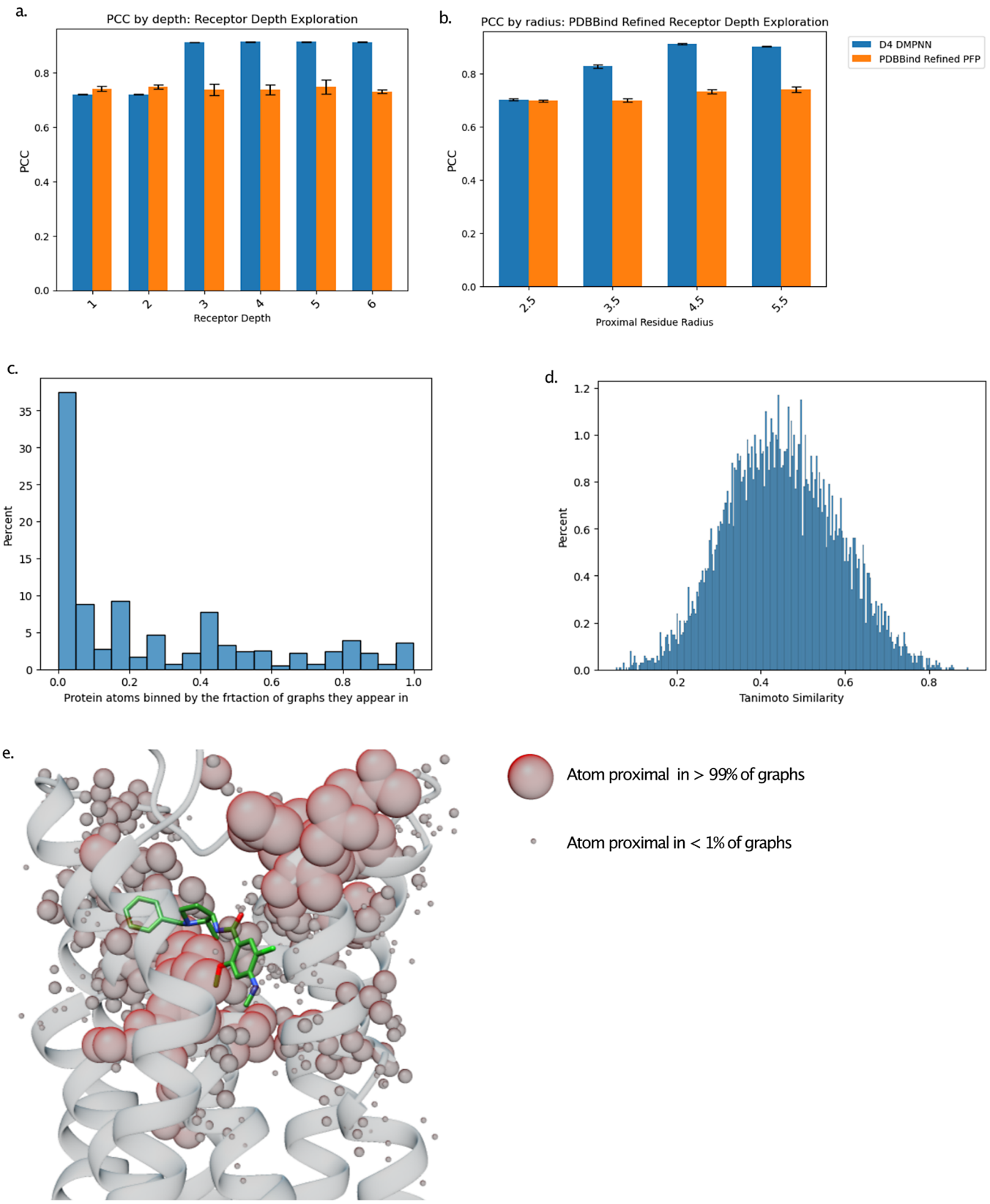


Figure 6. Analysis of proximity graph parameters and properties. Performance of the best 2D MPNN networks on the D4 Medium Diverse and PDBBind Refined 2019 data sets when the (a) proximity radius or (b) receptor depth vary. (c) Histogram showing the frequency of protein atoms included in the proximity graph during D4 Medium Diverse graph construction. (d) Histogram showing the Tanimoto Similarity of included protein atoms across a sample of 10,000 D4 Medium Diverse graphs. (e) Schematic showing the atoms included in the proximity graph overlaid onto the crystal structure. The size of the sphere centered on each atom is proportional to the percentage of graphs in which a given atom is included.

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450

Given this strong performance, we sought to better understand the nature of the D4 Medium Diverse proximity graphs. We found that about 37% of protein atoms occur in less than 5% of the proximity graphs. On the other hand, a subset of about 5% of atoms appear in almost all proximity graphs (Figure 6c). To further explore how unique the protein atom signature was for the graphs, we calculated the Tanimoto similarity between protein atoms for a random sample of 10,000 proximity graphs (Figure 6d). We found that the mean Tanimoto similarity was about 0.45, suggesting a meaningful protein-environment fingerprint overlap. To better understand how the seemingly paradoxical high degree of atomic uniqueness and overall similarity could coexist, we overlaid atom frequency with the D4 PDB used for docking (Figure 6e). It appears that there is a high degree of overlap in atoms within the proximity radius (as represented by the larger spheres near the binding pocket); however, there are many less-frequent atoms at the periphery, likely resulting in the addition of atoms using the receptor depth attribute. Given the sensitivity of performance to this attribute, it is likely that these atoms contribute significantly to the discriminative power of the models.

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 data sets are much more suited to MPNN-based models than others and that encoder architectures can have variable performance based on the character of the data set 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 data set. 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. Additionally, exploring different data augmentation techniques to improve model performance in low-data situations would be advantageous. Another area of future improvement will likely come from richer more engineered featurization of bonds and atoms and more physics-aware graph construction methods. 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] The strong generalization of our PGN models makes model fine-tuning for low-data situations another potential application that could aid drug discovery.[9,57]

ASSOCIATED CONTENT

Data Availability Statement

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

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.4c00311.

Result of hyperparameter optimization for the PFP Encoder; result of hyperparameter optimization for the GGNET Encoder; result of hyperparameter optimization for the DMPNN Encoder; result of hyperparameter optimization for the DimeNET++ Encoder on the PDBBind refined dataset with random splitting; other tables; comparison of the performance of the optimized PFP encoder PGN model; evaluation of the D4 Amber Experimental Dataset using the model training; fine-tuning results using the optimized D4 Medium Diverse models; Box plots; metric spaces; non-scaled values used to create the graphs; and mean error distributions (PDF)

AUTHOR INFORMATION

Corresponding Author

Michael J. Keiser − *Department of Pharmaceutical Chemistry, Department of Bioengineering and Therapeutic Sciences, Institute for Neurodegenerative Diseases, and Bakar Computational Health Sciences Institute, University of California, San Francisco, San Francisco, California 94158, United States;* orcid.org/0000-0002-1240-2192; Email: keiser@keiserlab.org

Authors

Zachary J. Gale-Day − *Department of Pharmaceutical Chemistry and Institute for Neurodegenerative Diseases, University of California, San Francisco, San Francisco, California 94158, United States;* orcid.org/0000-0002-8598-7048

Laura Shub − *Department of Pharmaceutical Chemistry, Department of Bioengineering and Therapeutic Sciences, Institute for Neurodegenerative Diseases, and Bakar Computational Health Sciences Institute, University of California, San Francisco, San Francisco, California 94158, United States;* orcid.org/0000-0003-0211-0396

Kangway V. Chuang − *Department of Pharmaceutical Chemistry, Department of Bioengineering and Therapeutic Sciences, Institute for Neurodegenerative Diseases, and Bakar Computational Health Sciences Institute, University of California, San Francisco, San Francisco, California 94158, United States;* orcid.org/0000-0002-0652-8071

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c00311

Author Contributions

Z.J.G.-D.: Conceptualization; Methodology; Software; Data Curation; Writing - Original Draft; Visualization. L.S.: Software; Validation. K.V.C.: Resources; Methodology.

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450

M.J.K.: Conceptualization; Writing - Review and Editing; Supervision; Project Administration; Funding acquisition.

Notes

The authors declare no competing financial interest.

■ 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.

■ REFERENCES

(1) D'Souza, S.; Prema, K. V.; Balaji, S. Machine Learning Models for Drug-Target Interactions: Current Knowledge and Future Directions. *Drug Discovery 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 Discovery* **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**, *10*, 19−23.

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

(9) Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P.Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems*; 2015; Vol. 2, pp 2224−2232.

(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. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 595−608.

(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. In *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. In *ILCR*; 2020; pp 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. *ACS Cent. Sci.* **2018**, *4*, 1520.

(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* **2021**, *37*, 1376−1382.

(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. *J. Chem. Inf. Model.* **2011**, *51*, 2897−2903.

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

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450

(32) Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*; 2017; pp 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), No. e1005690.
(34) Li, Y.; Rezaei, M. A.; Li, C.; Li, X. DeepAtom: A Framework for Protein-Ligand Binding Affinity Prediction. In *Proceedings - 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019*; 2019; pp 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. In *Proc. of the 27th ACM SIGKDD conf. on knowledge discovery & data mining*; 2021; pp 975−985.
(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) Gasteiger, J.; Giri, S.; Margraf, J. T.; Günnemann, S. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules. *arXiv [cs.LG]*, 2020. http://arxiv.org/abs/2011.14115.
(41) Bergstra, J.; Yamins, D.; Cox, D. D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proc. of the 30th International Conference on Machine Learning ICML*; 2013; pp I115−I123.
(42) Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-Generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; KDD'19*; Association for Computing Machinery: New York, NY, USA, 2019; pp 2623−2631.
(43) 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. In *NeurIPS*; 2019.
(44) Fey, M.; Lenssen, J. E. Fast Graph Representation Learning with Pytorch Geometric. *arXiv* 2019; No. *1*, 1−9.
(45) 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.
(46) 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.
(47) Hagberg, A. A.; Schult, D. A.; Swart, P. J. Exploring Network Structure, Dynamics, and Function Using NetworkX. In *7th Python in Science Conference (SciPy 2008)*; 2008; No. *SciPy*, pp 11−15.
(48) 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.
(49) 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.
(50) Sterling, T.; Irwin, J. J. ZINC 15 - Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55* (11), 2324−2337.
(51) 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), No. e75992.
(52) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39* (15), 2887−2893.
(53) 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.
(54) 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.
(55) 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.
(56) Chuang, K. V.; Keiser, M. J. Adversarial Controls for Scientific Machine Learning. *ACS Chem. Biol.* **2018**, *13* (10), 2819−2821.
(57) Navarin, N.; Tran, D. V.; Sperduti, A. Pre-Training Graph Neural Networks with Kernels. In *Nips*; 2018.

https://doi.org/10.1021/acs.jcim.4c00311
J. Chem. Inf. Model. 2024, 64, 5439−5450