Design and application of deep learning methods to structure-based drug design

Mikhail Volkov

UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE DES SCIENCES CHIMIQUES

Laboratoire d'Innovation Thérapeutique – UMR7200

THÈSE présentée par :

Mikhail VOLKOV

soutenue le : **5 Juin 2023**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Chimie / Chémoinformatique

Design and application of deep learning methods to structure-based drug design

THÈSE dirigée par :

M. ROGNAN Didier	Directeur de recherche, CNRS

RAPPORTEURS :

M. BONNET Pascal	Professeur, Université d'Orléans
M. MONTES Matthieu	Professeur, CNAM - Paris

AUTRES MEMBRES DU JURY :

M. GASTON-MATHE Yann	Président, IKTOS S.A.
Mme SOPKOVA Jana	Professeure, Université de Caen
M. VARNEK Alexandre	Professeur, Université de Strasbourg

Contents

Acknowledgements

The last years, which I have spent at the Doctoral School in Strasbourg, have been a long and important journey for me, and almost everything I have achieved so far would have been hardly possible without my colleagues I have been working with during my PhD, and without people outside the lab, who have been with me, encouraging me to keep moving forward.

First of all, I would like to thank my supervisor Dr. Didier Rognan, who proposed this interesting interdisciplinary project and offered me great support in the course of my PhD. My scientific discussions with him have greatly transformed my vision of the domain I have been working in, and I believe that the skills I have gained as a result of working under his supervision will let me become a researcher with a much more integral and mature perspective on the methodology of research, which is crucial and often complicated in the fields, which undergo rapid development.

I would like to thank Iktos for the financial support of my project, as well as my colleagues from this company with whom I happened to work and who either provided scientific and technical support or participated in fruitful discussions, and I am particularly grateful to Joseph-André Turk, Nicolas Drizard and Hamza Tajmouati, who co-supervised different parts of my project on different stages of it. I would like to thank all members of our research group in the LIT I had a pleasure to work with, and participate in common discussions and lab events. My special thanks to Guillaume Bret for assistance with on-boarding and constant technical support and to Merveille Eguida, who not only helped me with important IChem modifications, but also guided me through all the bureaucracy after my arrival in France.

I also consider it my duty to thank Alexey Orlov, Dmitry Osolodkin, and Vladimir

Palyulin, who were my teachers and colleagues at the Moscow State University and who sparked my interest in computer-aided drug design, which was the reason of making a life-changing decision to start my PhD in cheminformatics in Strasbourg three years ago.

These three years have been an unusual time due to various disasters, which affected my habitual course of life in unexpected ways, for example caused the accelerated transition to remote mode of working and living. This is why I am so grateful to all my friends who kept in touch with me despite the distance, and to my family, most notably to my mom and my brother, for their constant support and faith in me.

Finally, I would like to thank the members of the Thesis Jury – the main reviewers Dr. Pascal Bonnet and Dr. Matthieu Montes as well as other members of the Committee – Dr. Yann Gaston-Mathé, Dr. Jana Sopkova and Dr. Alexandre Varnek for having agreed to review this work.

Abbreviations

2D, two-dimensional

3D, three-dimensional

ADP, adenosine diphosphate

AMP, adenosine monophosphate

ATP, adenosine triphosphate

AUC, area under curve

CNN, convolutional neural network

CPU, central processing unit

DNN, deep neural network

ECFP, extended connectivity fingerprint

FC, fully connected

FDA, U.S. Food and Drug Administration

GAT, graph attention network

GBT, gradient boosting trees

GIN, graph isomorphism network

GNN, graph neural network

GPU, graphics processing unit

IC_{50}, half maximal inhibitory concentration

IFP, interaction fingerprint

IPA, interacting pseudoatom

K_d, dissociation contant

K_i, inhibition constant

MLP, multilayer perceptrone

MPNN, message passing neural network

NN, neural network

PDB, protein data bank

PPA, protein pseudoatom

RF, random forest

RMSD, root-mean-square deviation

RMSE, root-mean-square error

ROC, receiver operating characteristics

R_p, Pearson correlation coefficient

SAM, S-adenosyl methionine

SF, scoring function

SVM, support vector machines

TPU, tensor processing unit

t-UMAP, Uniform Manifold Approximation and Projection

Thesis summary in French

UNIVERSITE DE STRASBOURG

ECOLE DOCTORALE DES SCIENCES CHIMIQUES

RESUME DE LA THESE DE DOCTORAT

Discipline : Chimie

Spécialité (facultative) : Chimie théorique, chimie informatique

Présentée par : VOLKOV Mikhail
(Nom Prénom du candidat)

Titre : APPLICATION DE RESEAUX DE NEURONES PROFONDS A DES STRATEGIES DE DRUG DESIGN BASEES SUR LA STRUCTURE DE PROTEINES CIBLES

Unité de Recherche : UMR 7200 – Laboratoire d'Innovation Thérapeutique
(N° et Nom de l'Unité)

Directeur de Thèse : Dr. ROGNAN Didier – Directeur de recherche, CNRS
(Nom Prénom – Grade)

Co-Directeur de Thèse (s'il y a lieu) :
(Nom Prénom – Grade)

Localisation : Faculté de Pharmacie, Université de Strasbourg

Thèse confidentielle : X *NON* ☐ *OUI*

1. Introduction

Malgré l'évolution rapide des méthodes de conception de candidat-médicaments, notamment celles basées sur la structure des protéines cibles, la prédiction exacte de l'énergie libre (affinité) de complexes protéine-ligand à partir de leur structure tridimensionnelle (3D) reste un problème important en chimie théorique. Les progrès récents amenés par diverses architectures de réseaux de neurones (NN) à la reconnaissance d'image, de texte ou de son, ont infusé le domaine pharmaceutique, et laissent un champ ouvert à de meilleures prédictions d'affinité de ligands pour leurs protéines cibles, particulièrement en raison de la facilité croissante avec laquelle des structures 3D de complexes protéine-ligand peuvent être obtenues.

Les NNs parviennent à résoudre avec succès des problèmes sur lesquels ont butés les modèles d'apprentissage automatique de la génération précédente, et suscitent donc un intérêt grandissant en chémoinformatique. Malgré le nombre croissant de modèles incorporant des architectures NN modernes visant à lier la prédiction d'affinité [1], il reste des questions ouvertes sur l'état de préparation de ces solutions actuellement disponibles à une application directe au monde réel (ex: criblage virtuel de chimiothèques) en raison de la quantité limitée de données expérimentales à partir desquels ces NNs ont été entrainés. Par ailleurs, le domaine d'applicabilité de tels modèles à usage général reste discutable [2].

Mon projet de thèse est dédié à l'étude de l'applicabilité des modèles NN de prédiction d'affinité de liaison entraînés sur l'ensemble des données structurales actuellement disponibles. Je me suis concentré sur des architectures de réseaux neuronaux de graphes (GNN) qui ont démontré leur applicabilité à la chimie théorique [3] et ont suscité un intérêt exceptionnel en raison de leur applicabilité directe aux données moléculaires représentées sous forme de graphes, tout en nécessitant une ingénierie limitée des descripteurs utilisés par le GNNs. Les interactions protéine-ligand peuvent être naturellement représentées sous forme de graphes moléculaires, ce qui rend les approches GNN applicables au problème de prédiction de l'affinité de liaison.

2. Résultats et discussion

2.1. Développement de GNNs et prédiction d'énergies libres de liaison.

Nous avons utilisé la base de données PDBbind [4] comme source de données de structures de complexes protéine-ligand d'affinité connues. PDBbind est un sous-ensemble de la Protein Data Bank (PDB), pour lequel à la fois les structures 3D et l'affinité du ligand (K_d, K_i, IC_{50}) pour sa cible ont été déterminés expérimentalement. Ce jeu de données a été divisé en trois sous-ensembles: (i) un jeu d'entrainement se limitant au données publiés avant le 01/01/2016 (10565 entrées), (ii) un jeu de test externe ("core set", 257 entrées) classiquement utilisé à des fins de prédiction, (iii) un second jeu externe étendu ("2019 hold-out") à l'ensemble des complexes publiés après le 01/01/2017 (3386 entrées) et mimant ainsi un scénario réaliste ou des données nouvelles sont prédites par un modèle entrainé sur des données plus anciennes. [5]

Afin de développer une architecture de GNN, j'ai mis au point la représentation d'objets moléculaires (protéine, ligand, interactions protéine-ligand) sous forme de graphes (Figure 1). Pour ce faire, les interactions moléculaires ont été détectées à la volée au moyen du logiciel IChem précédemment développé au laboratoire [6] et converties en un premier graphe. Un second graphe décrivant le site de liaison de la protéine cible a été développé ad-hoc, en positionnant les nœuds du graphe sur des pseudoatomes caractéristiques de l'acide aminé en interaction avec le ligand, puis en reliant ces nœuds par des arêtes selon des critères de distance. Enfin, un troisième graphe est construit de manière classique à partir de la structure 2D du ligand en plaçant les nœuds sur les atomes et les arêtes sur les liaisons (Figure 1).

L'architecture GNN utilisée pour lire ces graphes descripteurs est basé sur un réseau de neurones à transmission de message (MPNN), tel que publié récemment par Gilmer et al. [3]. Ce type de GNN peut être appliqué à nos trois graphes déconnectés qui sont lus simultanément. Nous avons donc pu estimer la contribution exacte de chacun des trois graphes ligand (L), protéine (P) et interactions (I), ainsi que leur combinaison afin de prédire l'affinité de chaque ligand pour sa protéine cible.

La première surprise a été de constater que les graphes protéine et ligand, utilisés seuls, sont suffisants pour atteindre de bonnes prédictions, et que le graphe interaction, supposé être le plus pertinent, n'apporte aucune valeur ajoutée à la précision de modèles où les graphes sont utilisés simultanément. Ces résultats laissent entrevoir un biais sérieux dans le jeu de données PDBbind. Afin de le définir, j'ai donc mis au point

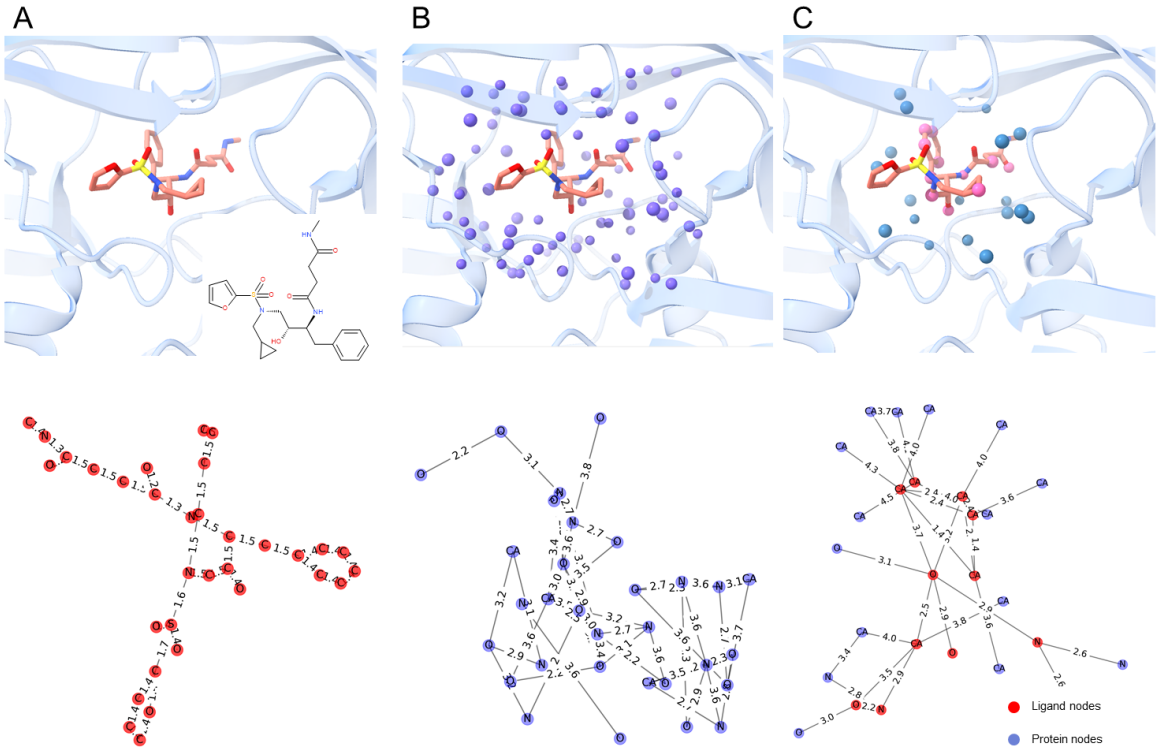


Figure 1. Graphes décrivant le ligand (A), la protéine cible (B) et les interactions protéine-ligand (C) (PDB ID 2PSV).

des modèles simples de mémorisation dans lesquels l'affinité du complexe protéine-ligand à prédire est déduite de la moyenne des affinités enregistrées pour le ligand le plus similaire ou la protéine la plus similaire. Ces modèles simples sont presque aussi précis que les modèles MPNN, notamment celui basé sur la simple similarité des ligands, et attestent que ces derniers n'ont pas capturé les détails physicochimiques fins d'une interaction moléculaire, mais fonctionnent par simple mémorisation et rappel des données connues pour les objets les plus similaires (Table 1).

Modèle	Jeu de test "2016 core set"		Jeu de test "2019 hold-out set"	
	R_p[a]	RMSE[b]	R_p	RMSE
PLI MPNN[c]	0.813	1.511	0.652	1.481
Similarité Ligand[d]	0.663	1.624	0.509	1.641
Similarité Protéine[e]	0.547	1.765	0.310	1.794

Table 1: a) Coefficient de correlation de Pearson, b) erreur quadratique moyenne de prediction (unite de pK_i), c) Modèle de MPNN à trois composantes (protéine, ligand, interactions, d) prédiction égale à la moyenne de l'affinité des cinq complexes du jeu d'entrainement avec les ligands les plus similaires de celui du jeu de test (exprimée par le coefficient de Tanimoto calculé sur des empreintes circulaires ECFP4), e) prédiction égale à la moyenne de l'affinité des cinq complexes du jeu d'entrainement avec les protéines les plus similaires de celui du jeu de test (exprimée par la distance Euclidienne calculé entre empreintes de cavité)

J'ai essayé de déterminer l'origine de ces biais en sous-échantillonnant l'ensemble d'apprentissage via l'élimination séquentielle de paires protéine-ligand dont l'affinité est facilement prévisible, ou en développant des modèles spécifiques tenant compte de l'enfouissement des ligands dans leur protéine cible. Aucune de ces approches n'a pu conduire à l'élimination de la dépendance aux ligands et aux protéines. Comme approche alternative, un nouvel ensemble de données PDBbind de faible parcimonie a été conçu. Nous avons remarqué que les mises à jour annuelles de l'ensemble de données PDBbind n'entraînent pas d'améliorations visibles de la qualité des modèles de prédiction d'affinité de liaison entraînés sur l'ensemble de données, ce qui peut être lié à la faible parcimonie de la matrice des paires protéine-ligand possibles. En raison du faible nombre de cibles pour lesquelles l'affinité de liaison de leurs complexes

avec plusieurs ligands est mesurée, nous en avons construit un nouveau sous-ensemble avec un nombre limité de protéines uniques (10 protéines les plus représentées, 2030 complexes) pour lesquelles plusieurs complexes avec des ligands différents d'affinité différente est connue. Une augmentation significative de la qualité du modèle d'interaction a été observée pour le modèle GNN entraîné sur cet ensemble de données à faible parcimonie par rapport à celui entraîné sur l'ensemble complet. Afin de débiaiser de manière définitive nos modèles, nous avons enfin supprimé les graphes P et L et conservés uniquement les graphes I en les complexifiant par prise en compte d'interactions moléculaires non plus à 4 mais à 6 Å de distance, ce qui a amélioré grandement la précision des modèles, sans biais induits par la prise en compte des graphes P et L qui ne sont plus lus par le MPNN (Fig. 2).

2.2. Application de modèles GNN au criblage virtuel

L'application directe des modèles de prédiction d'affinité à partir de structures cristallographiques (ou de cryo-microscopie électronique) est limitée par plusieurs facteurs:

- la flexibilité du ligand peut lui permettre d'être amarré dans différentes conformations d'enthalpie de liaison similaire à celle observée dans la structure aux rayons X, alors qu'il n'est pas garanti que l'ensemble des interactions soit préservé.

- l' absence d'échantillons négatifs (de faible affinité ou de mode de liaison incorrect) dans l'ensemble d'apprentissage, ce qui entraîne une incertitude dans la prédiction de l'affinité pour les modes de liaison sous-optimaux et l'impossibilité de filtrer les ligands de faible affinité dans un criblage virtuel.

Nous supposons que les interactions protéine-ligand représentées sous la forme d'un graphe d'interaction peuvent être appliquées à des données de criblage virtuel obtenu par docking moléculaire, à deux conditions: (i) disposer d'un modèle de prédiction de pertinence d'une pose de docking (classifieur binaire), (ii) disposer d'un modèle spécifique de prédiction d'affinité à partir de poses de docking préalablement retenues par le classifieur.

J'ai donc développé un modèle de prédiction d'affinité à partir de poses de docking en deux temps. Dans un premier temps, j'ai mis un point un classifieur binaire basé

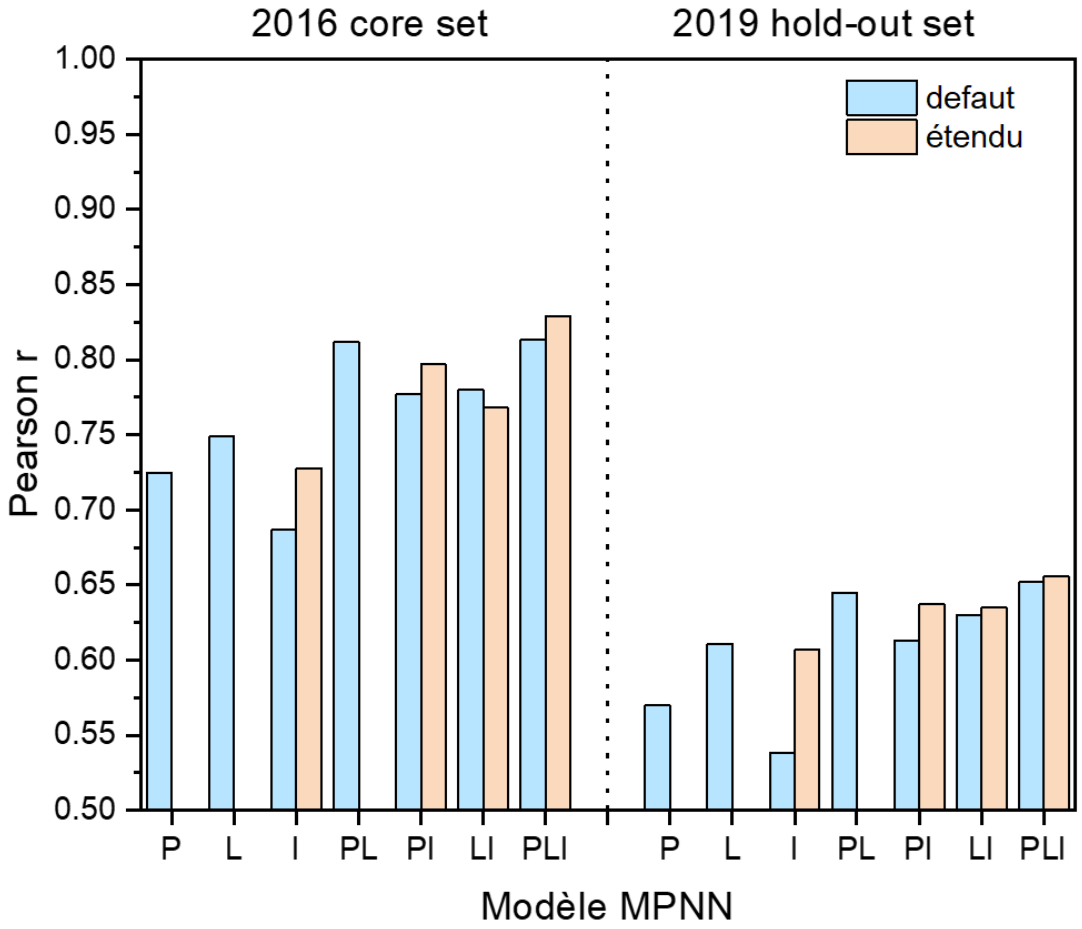


Figure 2. Performance (coefficient de Pearson) de modèles MPNN de prédiction d'affinité à partir de graphes de protéine (P), de graphe de ligand (L) et de graphes d'interactions (I) utilisés seuls ou en combinaisons. Par défaut, les interactions sont calculées dans une limite de distance de 4 Å. Le modèle dit 'étendu" enregistre les interactions non covalentes protéine-ligand jusqu'à 6 Å de distance. Les prédictions sont réalisées sur les deux jeux externe "core set" et "2019 hold-out.

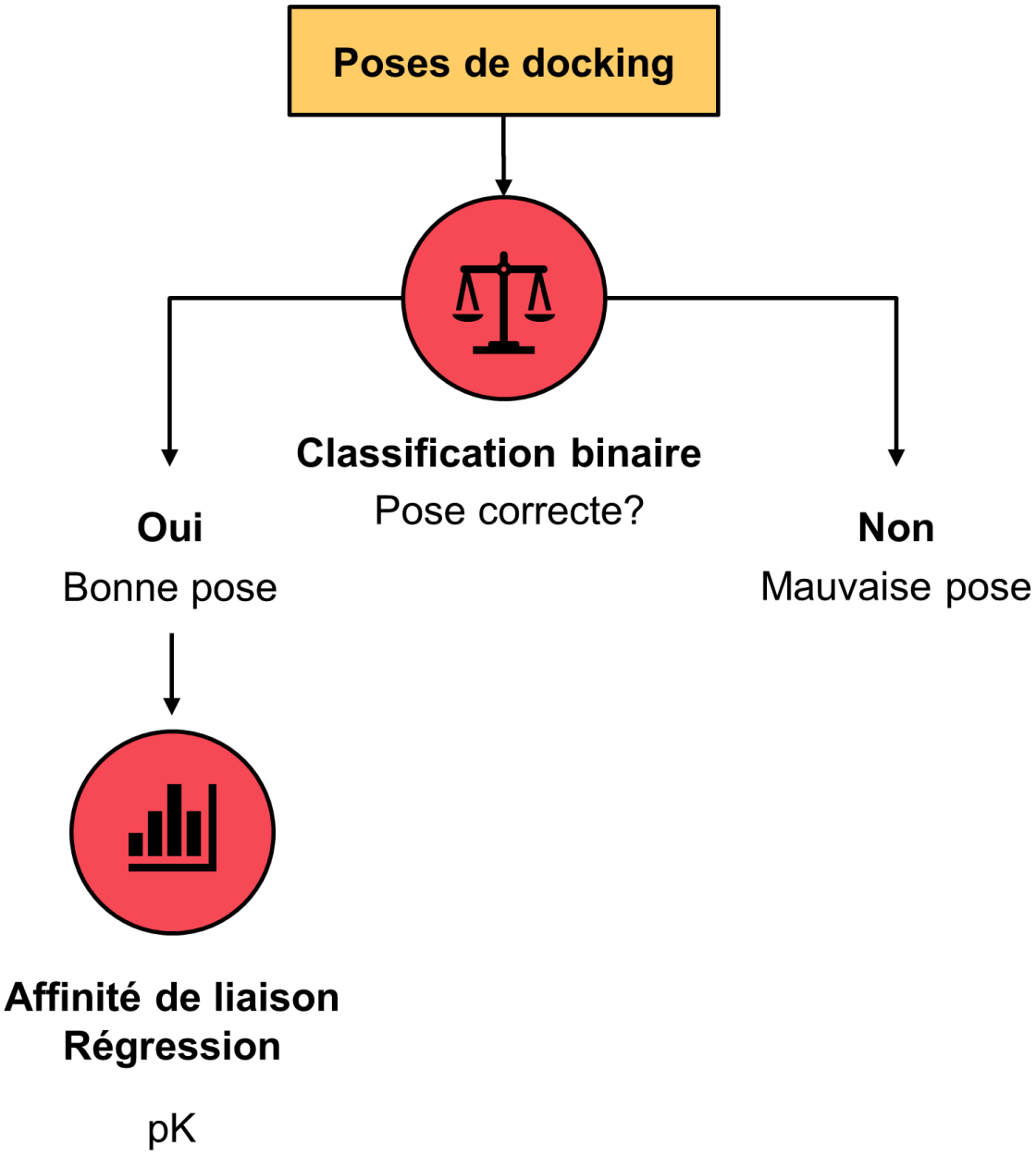


Figure 3. Prédiction d'affinité en deux étapes à partir de poses de docking.

sur la même architecture MPNN que notre prédicteur d'affinité de liaison qui peut être appliqué au graphe d'interaction pour le pré-filtrage visant à la sélection des bonnes poses de docking, qui peuvent être ensuite traités avec un modèle de régression pour en prédire l'affinité [Fig. 3]. L'ensemble de données augmentée pour cette étude a été préparé via le docking des ligands de la base de données PDBbind 2019 à leur protéine cible au moyen de trois logiciels de docking différents (Surflex-dock, Plants, Dock 6).

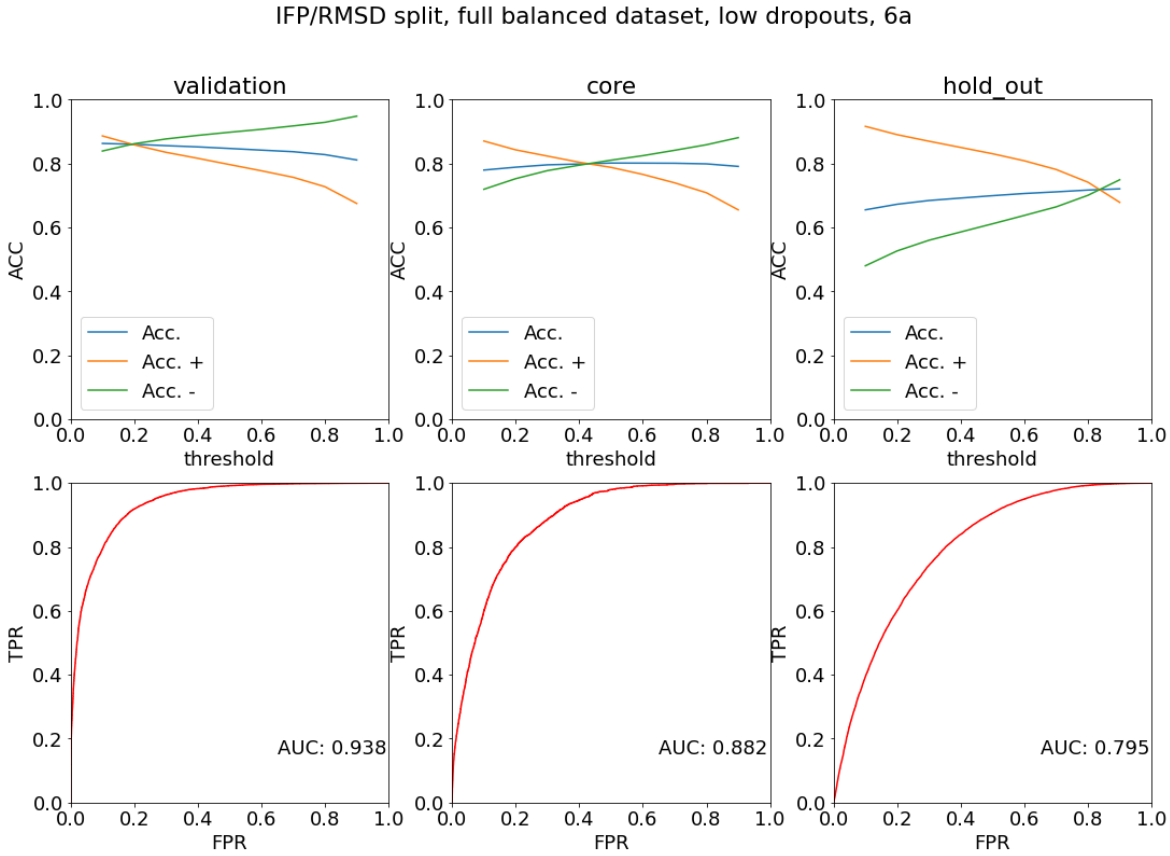


Figure 4. Précision de modèles MPNN de classification binaire de poses de docking, entrainés sur des graphes d'interaction protéine-ligand enregistrés à une distance maximale de 6 Å (B). Les valeurs de précision dont données pour toutes les poses (Acc), les vrais positifs (poses correctes, Acc+) et les vrais négatifs (poses incorrectes, Acc-). Les courbes ROC sont définies à partir des valeurs de probabilités de pose correcte émises par le modèle pour chaque pose.

L'ensemble des 4,5 millions de poses obtenues (au maximum 100 poses/ligand/logiciel) a ensuite été divisé en sous-ensembles de poses "correctes" et "incorrectes" en fonction de l'écart quadratique moyen (RMSD) des coordonnées atomiques par rapport à la

pose cristallographique du ligand (correct: RMSD<2 Å; incorrect: RMSD>2 Å) ainsi que la similarité des interactions protéine-ligand comparant pose de docking et pose cristallographique (correct: similarité>0.6). La première étape de classification binaire d'une pose de docking comme "correcte" ou "incorrecte", est satisfaisante avec une précision de l'ordre de 80% pour les deux labels (Figure 4). Les valeurs d'aires sous la courbe ROC calculés sur les deux jeux de tests sont supérieures à 0.80 et illustrent donc une bonne capacité du modèle de classification à distinguer les bonnes poses des mauvaises.

La seconde étape de prédiction d'affinité par régression, à partir de seules poses prédites correctes, permet de comparer des modèles de prédiction basées sur des structures cristallographiques ou sur des poses de docking. Afin de déterminer si un modèle entrainé sur un type d'objet moléculaire (structure cristallographique) peut être appliqué à l'autre (pose de docking), nous avons envisagé trois scénarios: (i) modèle entrainé et appliqué à des structures cristallographique, (ii) modèle entrainé sur des structures cristallographiques et appliqué à des poses de docking, (iii) modèle entrainé et appliqué à des poses de docking (Table 2).

Jeu	Scenario 1 [a]			Scenario 2 [b]			Scenario 3 [c]		
	Inc[d]	R^{2e}	R_p[f]	Inc	R^2	R_p	Inc	R^2	R_p
Core	0.222	0.521	0.750	0.557	0.386	0.706	0.556	0.612	0.803
Hold-out	0.316	0.306	0.564	0.428	0.103	0.485	0.428	0.399	0.644

Table 2. Performance de modèles MPNN de prédiction d'affinité selon trois scénarios après élimination des structures d'entrainement par le classifieur de poses. [a] Entrainement et test sur des structures cristallographiques, [b] Entrainement sur des structures cristallographiques et test sur des poses de docking, [c] Entrainement et test sur des poses de docking, [d] Inc: Fraction de poses prédites incorrectes, [e] R^2: Coefficient de détermination, [f] R_p coefficient de correlation de Pearson.

Les résultats obtenus montrent clairement qu'il est préférable d'entrainer et de tester les MPNNs sur les mêmes objets moléculaires (comparer les scénarios 1 et 3 au scénario 2). L'augmentation des données d'apprentissage à partir des poses de docking semble avoir un effet bénéfique sur la précision des poses de docking (Table 2). Plus le

jeu d'apprentissage est différent du jeu d'entrainement (jeu hold-hold), plus le modèle a des difficultés à prédire les valeurs d'affinité.

La taille relativement grande de l'ensemble des jeux de tests nous permet d'évaluer les performances de prédiction de l'affinité de liaison sur les sous-ensembles de celui-ci (Figure 5), consistant en des complexes formés par des ligands et des protéines, qui étaient soit nouveaux soit non nouveaux par rapport à l'ensemble d'apprentissage. L'évaluation sur quatre ensembles de données (nouvelle protéine et nouveau ligand, nouvelle protéine et ligand connu, protéine connue et nouveau ligand, protéine connue et ligand connu) démontre la généralisation insuffisante des modèle MPNNs à l'ensemble de données composé de protéines et de ligands complètement nouveaux.

En général, nos observations correspondent aux résultats obtenus pour le problème de régression, avec des prédictions trop optimistes pour le jeu de test core (facile car très similaire au jeu d'entrainement) et des performances inférieures sur un jeu de données plus complexe mais plus réaliste (hold-out). La tendance du modèle de graphe d'interaction étendu à produire des prédictions plus précises est également restée cohérente avec nos résultats précédents. Cette tendance est observée pour les trois scénarios, parmi lesquels le troisième, qui impliquait un prédicteur d'affinité entraîné sur les poses d'amarrage, a clairement démontré les meilleures performances sur les graphes d'interaction calculés jusqu'à 6 Å, montrant l'avantage de l'augmentation des données dans ce cas d'utilisation.

3. Conclusions générales

Au cours de mon projet de thèse, j'ai utilisé des réseaux de neurones profonds de type GNN pour développer des modèles de prédiction d'affinité de complexes protéine-ligand à partir de leur structure 3D ainsi que des modèles de classification de poses de docking, fonctionnant sur la même représentation en graphes que les modèles de régression. Ces modèles sont utilisables avec précaution, pour analyser des données de criblage virtuel et enrichir une sélection de touches virtuelles en ligands actifs.

Néanmoins, il a été démontré que la parcimonie de données d'entrainement reste un obstacle important pour le développement de modèles de prédiction d'affinité de liaison à usage général, non seulement en raison du manque de données, mais également

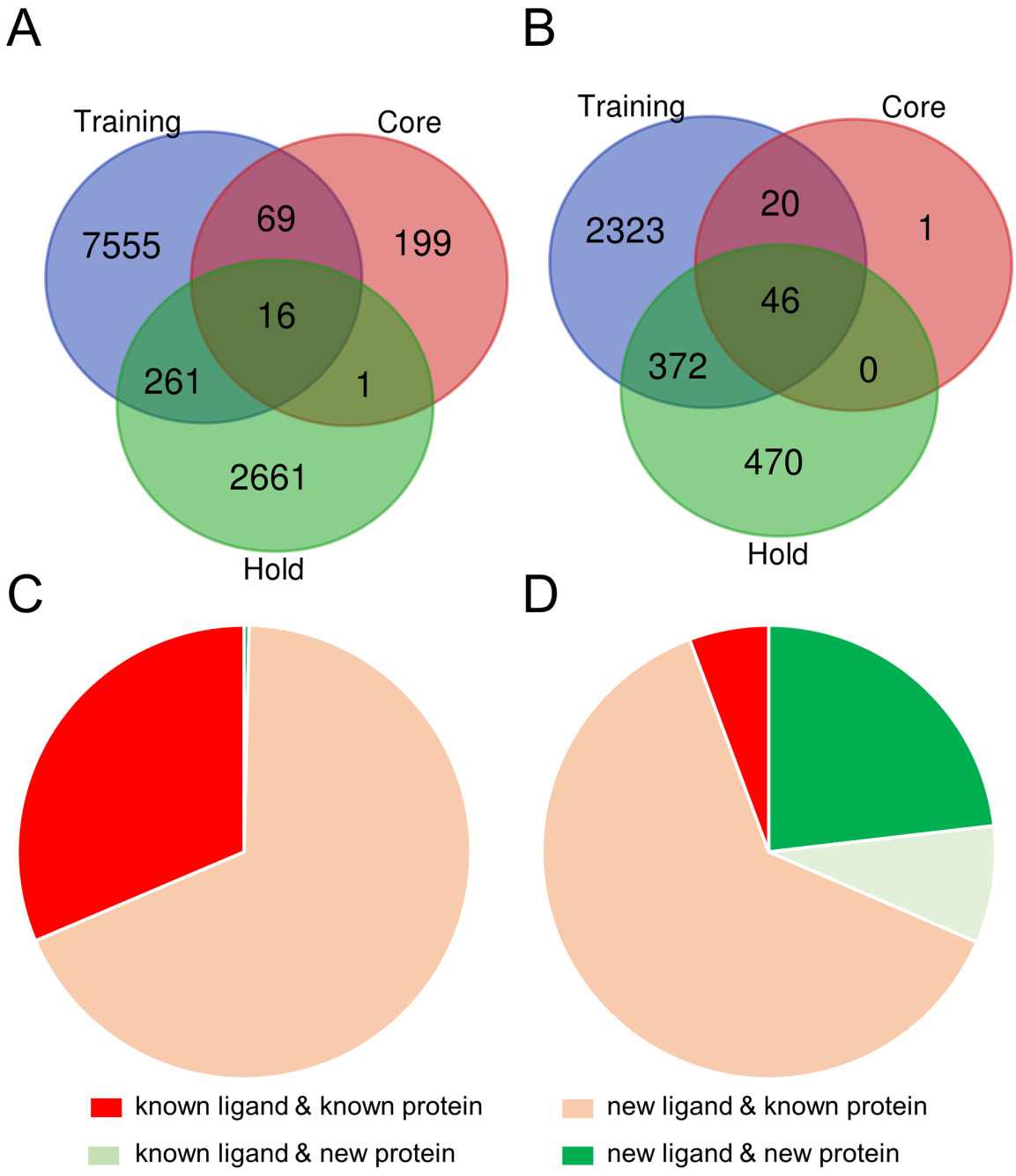


Figure 5. Croisement des entrées des jeux tests "core 2016" (287 entrées) et "hold-out 2019" (3386 entrées) avec le jeu d'entrainement (11820 entrées). A) Nombre de ligands identiques, B) nombre de protéines identiques, C) Composition comparée des jeux core et d'entrainement, D) Composition comparée des jeux hold-out et d'entrainement.

en raison de biais cachés dans l'ensemble de données, qui complexifient l'extraction de données pertinentes. Un effort coordonné de la communauté scientifique ainsi que des agences de financement sera nécessaire afin de disposer de matrices protéine-ligand plus denses et pour lesquelles à la fois la structure 3D et l'affinité auront pu être déterminés de manière expérimentale. Les méthodes d'apprentissage développées ici-même pourront dès lors exprimer tout leur potentiel et accélérer l'identification précoce de molécules bioactives.

4. References

(1) Bajorath, J. *Artificial Intelligence in the Life Sciences* **2022**, *2*, 100037.

(2) Yang, J.; Shen, C.; Huang, N. *Frontiers in pharmacology* **2020**, *11*, 69.

(3) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. In *International conference on machine learning*, 2017, pp 1263–1272.

(4) Wang, R.; Fang, X.; Lu, Y.; Wang, S. *Journal of medicinal chemistry* **2004**, *47*, 2977–2980.

(5) Sheridan, R. P. *Journal of chemical information and modeling* **2013**, *53*, 783–790.

(6) Da Silva, F.; Desaphy, J.; Rognan, D. *ChemMedChem* **2018**, *13*, 507–510.

Liste des présentations

Internes

Volkov, M. Application of graph neural networks to binding affinity prediction in protein-ligand complexes. Seminaire de l'UMR 7200, Obernai, 08/04/2022, *communication orale.*

Nationales et internationales

Volkov, M., Turk, J. A., Drizard, N., Martin, N., Hoffmann, B., Gaston-Mathé, Y., & Rognan, D. Applicability of graph neural networks to predict binding affinities from protein-ligand structures, GGMM- SFCi (Group of Graphism and Molecular Modeling & Société Française de Chémoinformatique) workshop, Lille, 29/09-01/10/2021, *présentation par affiche.*

Volkov, M., **Rognan, D.** On the frustration to predict binding affinities from protein-ligand structures with deep neural networks. 2nd Annual Iktos Conference on AI for de novo drug design, Paris, 03-04/05/2022, Paris, *communication orale.*

Volkov, M., Turk, J. A., Drizard, N., Martin, N., Hoffmann, B., Gaston-Mathé, Y., & Rognan, D. Applicability of graph neural networks to binding affinities prediction from protein-ligand structures, 17th German Conference on Cheminformatics 2022 GCC2022, 08-10/05/2022, Garmisch-Partenkirchen, Allemagne, *présentation par affiche.*

Volkov, M., Turk, J. A., Drizard, N., Martin, N., Hoffmann, B., Gaston-Mathé, Y., & Rognan, D. Applicability of graph neural networks to binding affinities prediction from protein-ligand structures, Chemoinformatics Strasbourg Summer School 2022, 27/06-01/07/2022, *présentation par affiche.*

Liste des publications

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. *Journal of Medicinal Chemistry* **2022**, 65, 7946–7958. doi: https://doi.org/10.1021

Introduction

The thesis manuscript covers the work carried out by the PhD candidate during studies at the ED 222 of the University of Strasbourg, which was devoted to the investigation of applicability of machine-learning based scoring functions to the problem of binding affinity prediction in protein-ligand complexes with the main focus on graph neural network architectures. The thesis manuscript consists of three main chapters.

In the first chapter, the state-of-the-art in the field of machine-learning based scoring functions is given. The most frequently used neural network architectures, which are applied in the design of binding affinity predictors are introduced, then, the main part of the review discusses the architectural and performance details of existing deep neural network-based scoring functions.

The second chapter is devoted to the development and application of graphs neural network (GNN) models for binding affinity prediction from protein-ligand X-ray structures. We propose an architecture that enables training on protein, ligand, or interaction input and the combinations of them allowing the user to evaluate the performance of models trained using different input representations, as well as the compatible graph representations of all types of inputs. Then, the performance of GNN models trained on the PDBbind dataset and evaluated on its benchmarking subsets is evaluated. The potential sources of biases to protein and ligand inputs are discussed. In order to determine them, a series of additional computational experiments, investigating the effects of ligand buriedness, the complexity of the interaction graph, as well and the sparsity of the matrix of protein-ligand combinations are performed. The contribution of the training set memorization into the performance of the final model is estimated with the help of simple memorization models we introduce in this chapter. Finally, the general conclusions on the applicability of the best models obtained in the course of the current study are given, and the possible directions of further research in the field are discussed.

The third Chapter is dedicated to the development of a method relying on previously developed binding affinity prediction models for possible virtual screening applications. Their major limitation was the inclusion of only X-ray structural data of complexes with true binders into the training set, while in a real use-case scenario the results of docking of not necessarily truly active small molecules are supposed to be

used. We therefore proposed a pipeline, containing a pose classifier, which filters out presumably incorrect docking results, and a binding affinity predictor, which is applied only to those poses, which have been previously classified as good. As the training and validation sets for this study, we used the results obtained by re-docking all PDBbind ligands into their protein targets, obtaining a set of docking poses of different quality. We assessed the performance of an architecture, based on two separate models for classification and regression, and the multitarget architecture, which performs two predictions simultaneously. The role of the novelty of structures (ligand, protein) in the time-split test set is evaluated and discussed. It appears that models trained on X-ray structures are poorly applicable to docking poses and vice-versa. Even with up-to-date GNN architectures and all available structural data from PDBbind, the best performing models still generalize poorly and are not widely usable for daily virtual screening applications.

CHAPTER 1

Application of neural network models for binding affinity prediction in protein-ligand complexes

1.1 Introduction

State-of-the-art drug design is a difficult and expensive problem due to both the complexity of the object of interest, which is a living organism, the incompleteness of information about the processes involved into the development of a certain pathology, as well as the attention paid to the security of substances, which are supposed to get a regulator's approval, which requires multistage assessment of efficacy and safety of the substance, boosting the development costs, reaching $868M to $1,241M, according to the FDA estimates [1], while the failure rate of clinical trials remains very high (more than 90% [2]). Thus any contribution to the simplification of the drug design pipeline, which can potentially lead to the higher success rate of the lead compounds accepted for preclinical and clinical trials is valuable for drug design projects. With the breakthrough in the development of computational hardware in the recent decades and the accompanying advancements of chemoinformatics and molecular modeling techniques and software, the computational methods in drug design methods have become an integral part of research pipelines in the pharmaceutical industry. The success of computer-aided drug design techniques has attracted significant investments not only to the companies, which perform the full cycle of drug design related research, but also to developers of mainly software solutions drug design research can benefit from.

A significant percent of approved drugs are small molecules, which selectively bind with their macromolecular targets in a human body, thus affecting the biological function of these targets. A successful medicine in this case is a molecule, which can strongly and selectively bind to the appropriate site on the surface of the macromolecule. The strength of binding can be characterized by the Gibbs free binding energy $\Delta_f G$, which can be experimentally determined by measuring the dissociation constant of the corresponding protein-ligand complex (Eq. 1.1).

$$\Delta_f G = RTlnK_d \tag{1.1}$$

where $K_d = \frac{[Prot][Lig]}{[Complex]}$in a reaction $Complex \rightleftharpoons Prot + Lig$

Determination of binding affinity of different ligands to a particular macromolecule is important in order to rank compounds from a certain set (compound li-

brary) and select those, which bind stronger, thus being able to selectively interact with this target in a human organism. While a rigorous calculation of binding affinity in a protein-ligand complex via quantum chemistry methods is hardly feasible in practical applications due to the extreme complexity of the system considered, methods, applying classical molecular mechanics approaches such as MM/GBSA proved their reliability in computation of binding energy in protein-ligand complexes [3]. In structure-based drug-design, though, that the pipeline of hit identification involves so-called "high-throughput screening" stage, on which molecules from a large library of compounds are tested in an assay aimed at determination of "activity" towards the target of interest. The aim of high-throughput virtual screening campaigns, though, is to either emulate the in vitro high-throughput screening stage in order to select "hit" molecules for further experimental investigation, or to simply reduce the number of compounds to be tested by filtering out structures from a large library of small molecules. Virtual screening campaigns often involve estimation of binding affinities of a huge number of protein-ligand complexes, which makes estimations even with classical molecular dynamics methods too expensive to be applicable in a real use-case scenario.

One of the methods often used in virtual screening in structure-based drug design (a type of drug design campaign, in which the structure of a macromolecular target of interest is known) is molecular docking. It is a computational procedure aimed at the determination of the energetically favorable conformations of molecules in which they can bind with each other. It can be broken down into two subsequent procedures – conformation sampling (generation of possible mutually compatible conformations of two molecules) and conformation scoring (estimation of binding energy of a conformation yielded on the previous stage). Molecular docking procedures can be classified by the types of molecules, which are fitted to each other e.g. protein and small molecule (protein ligand docking), two proteins (protein-protein docking), protein and nuclear acid. In the score of the current work only the former is discussed. Another possible classification encounters the consideration of flexibility (usually, rotations around rotatable bonds) of two molecules: in rigid docking neither receptor nor ligand flexibility in the course of docking is considered, in semi-rigid docking the receptor is treated as rigid and the ligand as flexible, in flexible docking both molecules are treated as flexible

(thus the docking procedure can emulate the induced fit mechanism).

A scoring function (SF) associates a value with a particular protein-ligand complex conformation, which, ideally, should correlate with the binding affinity of this particular complex. The scoring function quality can be determined according to their ability to solve problems occurring in drug design tasks such as [4, 5]:

- Scoring: ability to predict binding affinities that have a linear correlation with experimental data. The scoring power is usually measured by the Pearson's correlation coefficient R_p (see Evaluation metrics in section 1.4)

- Ranking: ability to rank binding affinities of known ligands of the same target protein. Unlike in scoring, where assessment is performed on a series of protein-ligand complexes of different ligands and proteins, in ranking the ligands of the same target molecule are examined. While the correct order of ligands scored by an SF, the linearity of score correlation with binding affinity is irrelevant for ranking. Thus, SFs with high-ranking power are more suitable for virtual screening campaigns. Ranking power is measured by computing the rate of correct identification of the best binder in a set of complexes. Alternatively, Kendall's tau and Spearman coefficient can also be used.

- Docking: ability to identify the "correct" binding conformations, or those conformations of a ligand which are close to it, from poses generated by docking software. Docking power is measured a the success rate of identifiying correct poses among the top ranked ones. A general appreciation of correcteness is provided by the root-mean-square deviation (RMSD) of heavy atoms to the native X-ray pose, that should be lower than 2.0 Å.

- Screening: ability to discriminate a target protein's true binders from presumed inactive decoy ligands. Metrics used to quantify the screening power is the enrichment factor in true actives among the 1% top-raking ligands.

Alternatively, scoring functions can be characterized according to the methodology behind binding affinity determination, a common classification [6] names the following scoring function categories:

- physics-based scoring functions– scoring based on force fields, solvation models, quantum mechanics models (e.g. DOCK)
- empirical scoring functions – score is based on contribution of different types of interactions, steric clashes, etc. These scoring functions benefit from low computational cost, but they perform worse in determining the exact binding affinity. Construction of a representative training set is crucial in development of these scoring functions.
- Knowledge-based scoring functions – scoring based on pairwise potentials
- Machine-learning based scoring functions – a family of scoring functions relying on machine learning algorithms (SVM, RF, NN, DNN).

In the scope of the current review we will focus on the discussion of machine learning based scoring functions using deep neural network architectures. The review consists of three parts in which we discuss the basic architectures, which are being used as building blocks in the design of protein-ligand affinity prediction; the available public datasets on which training and evaluation of DNN-based scoring functions is usually performed; then we perform a review of existing models, discussing their unique features, strengths, and weaknesses.

1.2 Deep neural network architectures commonly used in binding affinity prediction models

1.2.1 Convolutional neural networks (CNN)

First proposed in 1988 [7], the CNN architecture revolutionized image recognition field in the recent decade due to development of powerful hardware solutions for large-scale parallelism (GPUs, TPUs). In 2013, AlexNet [8] showed the best performance in the image classification challenge on ImageNet. This result was beaten by the ResNet model, introduced by Microsoft in 2015 [9], causing the outbreak of interest to neural network applications in various branches of science and technology.

The basic convolutional neural network includes three types of layers: convolu-

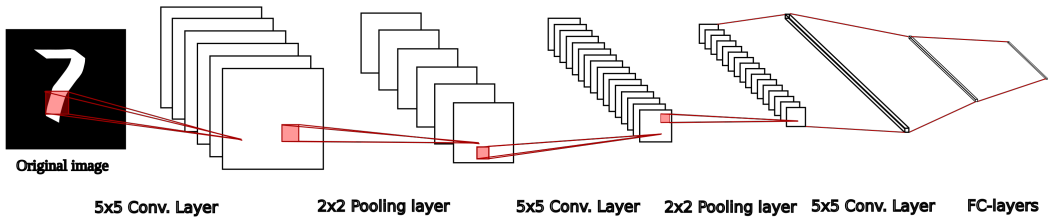


Figure 1.1 LeNet-5 – an example of a CNN architecture for image classification

tional, pooling layer, and fully-connected layers or multilayer perceptrons (MLPs) [10]. The most characteristic part of a convolutional architecture are convolutional layers, which extract characteristic features of an image or another euclidean object. The convolutional layer consists of kernels, which "move" along the image retrieving information from different parts of an image, thus the model becomes capable of identifying a pattern, which occurs in different parts of the object. The output of a convolutional layer is a feature map, each neuron of which is connected to an area (receptive field) of an input image. A feature map is produced via a convolution operation with a kernel and an application of a non-linear activation function. For a 2D convolution a feature map value $z^l_{i,j,k}$ with the coordinates (i, j) is expressed in the Eq. 1.2:

$$z^l_{i,j,k} = W^{lT}_k x^l_{i,j} + b^l_k \tag{1.2}$$

(1) , where l is the layer index, k is a feature map number in a current layer, W^l_k is a learnable kernel tensor, b^l_k is a bias term, $x^l_{i,j}$ is a input fragment with a center at (i, j). The kernel weights are shared for the current feature map. The feature map value after an activation is computed as in Eq. 1.3.:

$$a^l_{i,j,k} = a(z^l_{i,j,k}) \tag{1.3}$$

where a is a non-linear activation function e.g. sigmoid, tanh, ReLU.

The pooling layer in a CNN serves for preservation of invariance to shifts via reduction of the feature map resolution. The action of a pooling layer is shown at the Eq. 1.4:

$$y^l_{i,j,k} = pool(a^l_{i,j,k}), \forall (m, n) \in R_{i,j} \tag{1.4}$$

where $R_{i,j}$ is a local neighborhood of (i, j).

The most frequent pooling operations are average or max pooling. In the former, the average value of pixels in a local neighborhood is taken. In the latter, the maximal value in a local neighborhood.

A deep convolutional network consists of multiple convolutional layers with pooling, the feature maps of the first layers detect low-level features (such as simple shapes), while subsequent feature maps recognize more complex patterns.

The output of a stack of convolutional layers with pooling is further processed with fully connected layers which take a latent representation of the CNN block as an input and perform "high-level reasoning" producing the desired output of the entire model – a vector of probabilities (logits) for a classification problem, or a single value for a regression problem. Despite their advantages such as easier explainability, easiness to apply techniques developed for image processing tasks, convolutional neural networks still possess some drawbacks, which make alternative solutions more preferable in some applications. CNNs are invariant to translations, but the correct treatment of rotations by this architecture requires data augmentation. The input grid feature tensor is also sparse – only a small fraction of the volume, containing a protein-ligand complex, contains interacting atoms and functional groups thus being relevant for proper determination of binding energy. This redundancy leads to computational inefficiency such as increased memory requirements and necessity to perform more costly computations.

1.2.2 Graph neural networks (GNN)

Graph neural networks are a novel emerging methodology in machine learning, which shows great results when applied to train neural network models for processing of data, which are more naturally represented in a form of graphs, rather than regular grids. Graph irregularity does not allow feature extraction via convolution with CNN-like filters; thus several alternative approaches were developed. A major breakthrough in the development of GNN architectures took place in 2016-2017 when several independent research groups introduced graph convolution architectures, which remain the state-of-the-art in machine learning on chemical data, namely, graph convolutional neural networks (GCN) [11] and message passing neural networks (MPNN) [12], followed by

Graph Attention Networks (GAT) [13], which resolve some of the issues of the original Kipf's GCN. Despite scalability issues (the dependency of the node representation on its neighborhood of potentially very high complexity leads to increased computation costs for large and/or highly connected graphs), this family of architectures causes particular interest due to their applicability in many chemoinformatics-related tasks.

Convolutional neural networks (GCN)

Despite all GNN architectures, which are discussed in this review, technically belong to the family of models with a graph convolutional operator [14], the term "graph convolutional network" (GCN) is commonly used to describe an approach proposed by Kipf et al. in 2016, is a modification of a spectral graph convolution approach that stems from graph signal processing methods [11]. The convolution operation in this architecture is performed on a graph equipped with self-loops for every node, which prevents considering feature vectors of neighboring edges only but not the current node itself on the convolution stage. The graph structure is represented as an adjacency matrix , the graph nodes are equipped with node feature vectors stored in a matrix , where N is the number of nodes, D is the node feature length of an input graph. The output of the convolution layer is defined by the equation 1.5:

$$H^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{1.5}$$

where l is the layer number, $\tilde{A}$ is the adjacency matrix of G, $\tilde{D}$ is a degree matrix, $W^{(l)}$ is a weight matrix, $H^{(l)}$ is the matrix of activation of the current layer ($H^{(0)} = X$), σ is the activation function (e.g. ReLU). The $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ multiplication here performs symmetric normalization of node features, which prevents gradient explosion or vanishing.

Message passing neural networks (MPNN)

The message passing neural networks were first introduced in 2017 in [12]. In this framework the graph $G = (V, E)$ is represented by the set of nodes equipped with node feature vectors h_v connected with edges equipped with edge feature vectors e_{vw}. Each message passing step consists of two operations: message computation and node feature

update, expressed by the message function and update function. The third operation — "readout" — is performed afterwards. All three functions should be differentiable and should be invariant to node permutations.

The message function (usually, the sum) computes messages on graph edges taking the feature vector of this edge and feature vectors of nodes this edge connects as arguments. Then for each node the message it receives is computed via summation of messages coming from all outbound/inbound edges as shown in the Eq. 1.6:

$$m_v^{(l+1)} = \sum_{w \in N(v)}^{(l+1)} M_t(h_v^l, h_w^l, e_{vw}) \tag{1.6}$$

where M_t is the sum of all messages the node receives, h_v^t is the feature vector of node v, h_w^t is the feature vector of a neighbor node w, e_{vw} is the feature of edge between nodes v and w.

The update function changes the node feature vectors considering the incoming message. The update function (Eq. 1.7) is usually the average between the message the node receives and its previous feature vector.

$$h_v^{(l+1)} = U_t(h_v^t, m_v^{(t+1)})) \tag{1.7}$$

where U_t is the update function, $m_v^{(t+1)}$ is the message received by the node v.

The message passing step can be repeated multiple times, the number of steps plays a role of one on the model hyperparameters.

The final graph representation after message passing should further be transformed into a tensor of a regular shape, so that it can be further processed by other neural network layers. This is achieved by applying a readout function to the graph (e.g. sum or average of node feature vectors) (Eq. 1.8).

$$\hat{y} = R(h_v^T | v \in G) \tag{1.8}$$

where $\hat{y}$ is the output tensor, R is the readout function, T is the number of message passing steps.

The capability of processing edge information is one of the advantages of this method in comparison with GCN, but at the same time the necessity to store not only

node features, but also edge messages can be a limiting factor of application of this method.

Graph isomorphism networks (GIN)

Despite their ability to demonstrate the breakthrough in graph learning with respect to their ability to linearly scale with the number of nodes in the graph, the GCN framework still suffers from limited capability of distinguishing isomorphic and non-isomorphic graphs. In the Graph isomorphism architecture (GIN), first proposed by Xu et al. [15], node feature update is performed according to the Eq. 1.9:

$$h_v^{(k)} = MLP^{(k)}((1 + \epsilon^{(k)}) \cdot h_v^{(k-1)} + \sum_{u \in N(v)} h_u^{(k-1)}) \tag{1.9}$$

where h_v is the node feature vector, h_u are the feature vectors of neighbor nodes, k is the layer index, MLP is the multilayer perception, ϵ is a parameter, which can be learnable or static (NNs with a static ϵ are unable to distinguish some non-isomorphic graphs, although being almost as powerful as the NNs with learnable ϵ on real benchmark datasets).

Graph attention networks (GAT)

The graph attention architecture (GAT) proposed by Veličković et. al. [13] in 2017 was developed with an aim to make implicit attribution of different weights to different nodes in a neighborhood of a certain node possible without performing expensive matrix operations or without a prior knowledge of a graph structure. The latter is a drawback of spectral approaches and their successors, such as GCNs, which require a computation of a graph Laplacian, that depends on graph connectivity. This condition restricts the applicability of these methods to graphs having a structure, which differs from the training set samples.

Attention mechanisms in the state-of-the-art machine learning are a standard technique to resolve problems involving operations on sequences such as machine translation. One of the benefits brought by this technique is an ability to operate on inputs of variable length.

In the GAT architecture the latent representations are computed for each node of the graph using self-attention, assigning neighbor nodes different weights based on their importance. This operation can be performed without a prior knowledge of the global graph structure, the graph also can be either directed or undirected.

As an input, the attention layer uses a set of node features $h = \{\vec{h_1}, \vec{h_2}, ..., \vec{h_N}\}, \vec{h_i} \in \mathbb{R}^F$, where N is the number of nodes, F is the number of node features. As an output the layer produces a set $h' = \{\vec{h'_1}, \vec{h'_2}, ..., \vec{h'_N}\}, \vec{h'_i} \in \mathbb{R}^{F'}$, where the number of node features can be different.

A self-attention operation on the nodes (Eq. 1.10) yields an attention coefficient, which corresponds to the importance of node j to node i. These coefficients e_{ij} are computed only for nodes, which belong to a neighborhood of node i, and normalized using a softmax function (Eq. 1.11).

$$e_{ij} = a(W\vec{h_i}, W\vec{h_j}) \tag{1.10}$$

where W is a learnable weight matrix $W \in \mathbb{R}^{(F' \times F)}$

$$a_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{jk})} \tag{1.11}$$

These values are further used as coefficients of corresponding node features in a linear combination of node feature vectors, which is used to compute new node features after applying a nonlinear function σ.

$$\vec{h'_i} = \sigma(\sum_{j \in N_i} \alpha_{ij} W\vec{h_j}) \tag{1.12}$$

A multi-head attention approach, which can be used to stabilize learning, supposes that several independent attention procedures with their own W matrices are applied, and their outputs are concatenated into new node feature vectors h'. If multi-head attention is performed in a final neural network layer, averaging of outputs can be used instead of concatenation.

1.3 Datasets

1.3.1 Structural datasets

PDBbind

In the scope of the current work we investigated predictability of binding affinity in protein-ligand complexes from their structures. In order to perform training of such models, a dataset of protein-ligand 3D structures annotated with binding affinities of the corresponding complexes is needed. X-ray crystallography and more recently cryo-electron microscopy (cryo-EM) [16] remain the main source of high-quality structural data in molecular biology nowadays, and the Protein Data Bank (PDB) serves as the main public library of resolved X-ray structures of biomolecules. Thus, structural datasets, containing information about protein ligand binding are usually being developed as curated derivatives of the PDB —- duplicates of the same complex are discarded, the information on binding constants is retrieved from literature. The most commonly used dataset of this type is PDBBind, developed and maintained by R. Wang's group [17, 18]. It contains protein-ligand, protein-nucleic acid nucleic acid-ligand, the inhibition constant (K_i), the half maximal inhibitory concentration (IC_{50}) [18]. In case if multiple types of binding affinity measurements are available, the priority is given to K_d over K_i over IC_{50} values. If multiple binding affinity measurements are available, only those, which were performed at room temperature and neutral pH or at the closest conditions to these standard ones were considered. The dataset is regularly updated, its most recent version was released in 2020 [19]. Its 2016 version contained more than 13300 complexes. 4057 of which belong to the so-called "refined set" – a subset of structures of higher quality. In order to be admitted to this data set the complex must satisfy the resolution criteria (resolution $<$2.5 Å and R-factor $<$0.25). The available affinity value should be either K_d or K_i, the ligand should not be bound covalently, etc.

The first release of PDBbind took place in 2004 and was based on current state of the PDB database by 2003 (release #103) [17]. Afterwards, the dataset was updated annually, the most recent release is PDBbind v.2020, which contains data on 19443 protein-ligand, 2852 protein-protein, 1052 protein-nucleic acid, and 149 nucleic acid-

ligand complexes [19].

CASF Benchmark sets

The CASF benchmark is a popular benchmark for assessment of structure-based binding affinity prediction models evaluating their performance in docking, ranking, screening and scoring tasks. One of the ideas behind the creation of CASF was decoupling scoring benchmarks from docking benchmarks in order to be able to evaluate scoring quality independently from sampling quality [5], the dataset thus contains high-quality structures of protein-ligand complexes from the "core set" of PDBbind. Its first release took place in 2007 (derived from the PDBbind set of the same year), with subsequent updates in 2013 and 2016. The "core set" is designed on the basis of the "refined set". Complexes, which make it up, are picked from clusters of similar protein targets so that each cluster is represented by three structures with high, medium, and low binding affinity.

The core set construction principles for the CASF 2016 core set [20] are listed below:

1. All refined set complexes are clustered by protein sequence similarity with a 90% sequence similarity cutoff, so that each cluster normally corresponds to a single protein target.

2. Cluster less than six members are discarded. From the remaining clusters five representative complexes are chosen – the one with the highest binding affinity, the one with the lowest one, and three complexes with intermediate affinity values. The range between binding affinities for complexes with max K_a and min K_a should be at least two logarithmic units, the intermediate affinities are picked so that they evenly cover the binding affinity range with at least one logarithmic unit gap between affinity values.

3. The quality of the electron density map is examined to guarantee the correctness of the 3D structure of a complex in PDB.

The core set of PDBbind v.2016 (CASF 2016) consists of 285 protein-ligand complexes belonging to 57 clusters of protein sequences (5 complexes per cluster). The

core sets used in previous CASF versions included 195 complexes (65 protein clusters, 3 complexes per cluster) for CASF-2013 and CASF-2007.

Binding MOAD

Binding MoAD [21–24] is an alternative dataset of protein-ligand structured with provided binding affinity annotations, which was developed with an intention to address the redundancy issue in datasets such as PDBbind, which is inherited from PDB database that can contain multiple structures of the same complex.

Complexes included into the database have resolution better than 2.5 Å, and the ligands in structure files are additionally validated. Accepted ligands are biologically relevant small molecules, including peptides with amino-acid chain length of 10 or less as well as oligonucleotides with four or less nucleic acid residues. Covalently bound molecules are discarded as well as small molecules serving as additives in crystallization.

The developers of Binding MOAD analyzed sequence similarity of protein targets and clustered them by 90% sequence identity cutoff, for each cluster a ligand with the highest affinity was picked.

Binding MOAD is also updated on an annual basis, its first release in 2005 [21] contained 5331 protein-ligand complexes, while the most recent update (http://www.bindingmoad.org/) contains 41409 structures, 15223 (36.8%) of which have binding affinity annotations (K_a, K_d, K_i, or IC_{50}).

Astex Diverse Set

The Astex diverse set [25] was developed as a validation set for molecular docking constructed according to the following quality criteria for a benchmarking set proposed by the authors: 1) Protein-ligand complexes should be relevant for drug discovery; 2) The dataset should contain diverse ligands and protein targets; 3) Crystal structures in the dataset should have very high quality; 4) Ligand should not form contacts with multiple protein subunits; 5) The dataset should be sufficiently large; 6) It should include recently resolved structures; 7) It should be freely available. The authors analyzed electron densities of complexes and considered only those for which the electron density corresponded to the ligand geometry in a PDB structure. The dataset was

composed based on the protein sequence clustering, protein clusters of little interest to drug design or agrochemistry were not considered, non-drug-like ligands were also discarded. Electron density maps were examined on order to exclude ambiguous structures. The resulting set is composed of 85 diverse protein-ligand complexes, for which binding affinity is given in either K_i, K_m, K_d, or IC_{50} units. Approximately 90% of protein targets in the dataset are targets in drug-discovery or agrochemical campaigns. Among the ligands, 23 are approved drugs and six participated in clinical trials.

1.3.2 Non-structural datasets

Metz

The dataset proposed by Metz et. al. contains screening data for 172 different kinases and more than 3000 kinase ligands [26]. The binding affinity is provided for 42.1% [27] of possible drug-target pairs in pK_i units.

Davis

The Davis dataset [28] includes data on 72 known kinase inhibitors and 442 target kinases, representing >80% of the human protein kinome. The results of a screening assay are provided, K_d values are given for protein-ligand pairs with measurable binding affinity (K_d <10 μM) on a primary screen. The pK_d values in the Davis dataset lie in the interval from 5.0 to 10.8 [29].

KIBA

The KIBA dataset [30] is another public dataset of kinase inhibitor bioactivity data. It was originally developed with an aim to obtain a database with comparable binding affinity values obtained from multiple sources under an assumption that the conditions of measurement of the corresponding affinity values were the same. The authors collected drug-target affinity data experimentally measured in K_i, K_d, and IC_{50} units from multiple databases, and transformed these values into "KIBA scores" in order to achieve the uniformity of them. KIBA score is computed according to the following rules:

$$KIBA = \begin{cases} K_i\ adj,\ if\ IC_{50},\ K_i\ are\ available \\ K_d\ adj,\ if\ IC_{50},\ K_i,\ K_d\ are\ available \\ (K_i\ adj + K_d\ adj)\ /\ 2,\ if\ IC_{50},\ K_i,\ K_d\ are\ available \end{cases} \tag{1.13}$$

where

$$K_i\ adj = \frac{IC_{50}}{1 + L_d(IC_{50}/K_i)} \tag{1.14}$$

$$K_d\ adj = \frac{IC_{50}}{1 + L_d(IC_{50}/K_d)} \tag{1.15}$$

The pairwise matrix of KIBA scores is provided for 52498 compounds (with given ChEMBL IDs) and 467 kinase targets (with given Uniprot IDs). The score values cover the range from 0.0 to 17.2.

In evaluation of binding affinity prediction models a modified version of the KIBA dataset is often used, in which drugs and targets for which less than 10 measurements of affinity were available, are removed. The resulting dataset contains 2116 unique drug molecules and 229 target molecules and achieves the protein-ligand matrix coverage of 24.4% [27].

1.4 DNN-Based Binding Affinity Prediction Models

1.4.1 Binding Affinity Prediction models classified by model architectures

While most of the existing DNN-based binding affinity predictors are using the NN architectures described in the previous part of the current review, a significant difference between models, which rely on the same architecture may be in the way, a protein-ligand complex is represented for a subsequent treatment by a NN model. In this part

of the review, we describe various binding affinity prediction models, paying attention to both the model architecture and the input featurization applied. The summaries of model performance of different benchmarking sets are given in Tables 1.1-1.5.

Prediction of protein-ligand binding affinities solely from ligand and protein string representation.

DeepDTA [31] was one of the first DNNs, outperforming previous baseline models KronRLS (linear regression on a similarity matrix), and SimBoost (gradient boosting trees) on Davis and KIBA datasets. In its best performing version, protein and ligand embeddings learned from SMILES strings of ligands and protein sequences are used as inputs for two 1D-CNN blocks, which generate latent representations of a protein and a ligand. These latent vectors are then concatenated, and used as an input of a three-layer MLP, which predicted the target affinity value. A WideDTA model released by the same research group one year later uses a similar architecture with its CNN part extended to four CNN blocks. Each of these blocks processes Ligand SMILES string, ligand maximum common substructure, protein sequence, protein motifs and domains. The learned latent representations of these four inputs are then processed similarly to as it is performed in DeepDTA.

An innovation in the AttentionDTA architecture, derived from DeepDTA, was the inclusion of an attention mechanism to correlate protein and ligand information in the latent representations generated by the CNN blocks [32]. The resulting model demonstrated the level of performance of the WideDTA model on KIBA and Davis datasets without more complex feature engineering and inclusion of additional convolutional blocks.

The next substantial improvement in binding affinity prediction from protein and ligand information for kinase datasets took place with the release of GNN models such as DgraphDTA in 2020 [33]. The general framework of the architecture being used inherited the one used in DeepDTA, but unlike in the case of its predecessor, the CNN module was replaced by Kipf's GCN layers. This modification required changes in the input preparation — instead of plain text strings ligand and protein were represented in a form of graphs —- a 2D molecular graph for the former and a contact map-based

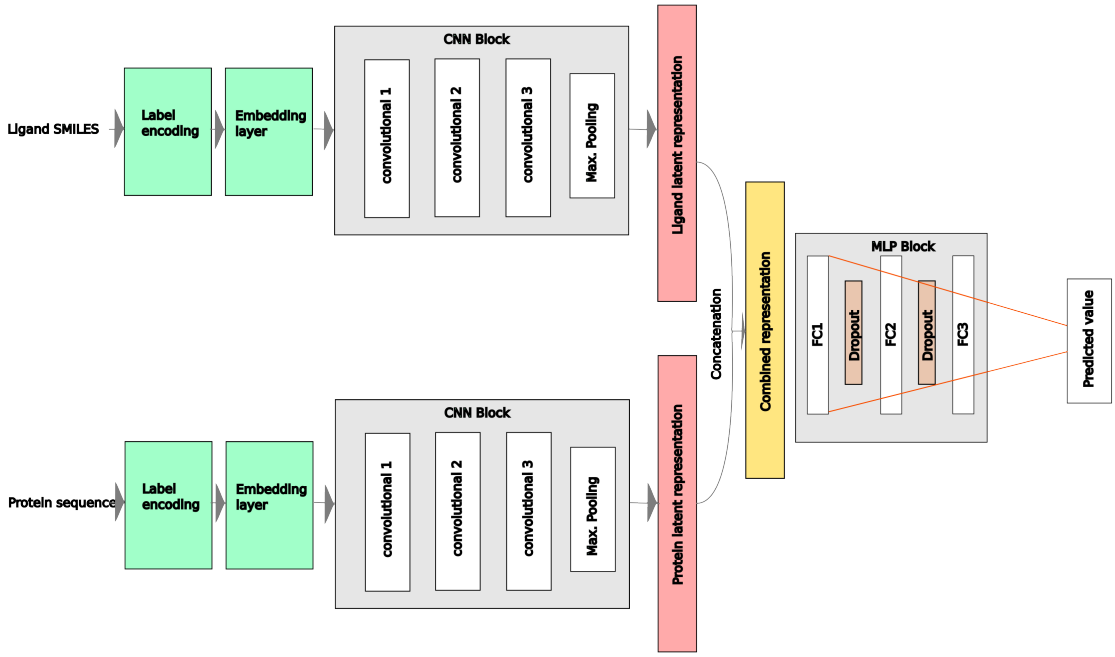


Figure 1.2 A flowchart representation of the DeepDTA architecture.

representation for the latter.

Unlike in DgraphDTA, in GraphDTA the CNN-based learning on protein sequences is kept, while the GNN module is introduced for ligand information processing only. The authors of the corresponding study [34] examined the performance of binding affinity predictors using several GNN architectures, namely, GIN, GCN, GAT, and a combination of GAT and GCN. Testing on the Davis dataset showed superior performance of GAT and GIN with a concordance index (CI) of 0.892 and 0.893. On the KIBA dataset, all models except GAT demonstrated approximately equal performance of CI 0.89, still standing behind DgraphDTA. A similar level of performance was achieved with a MATT-DTI model [35], in which a CNN-block for ligand information was used. The difference from AttentionDTA was the multihead attention operation to combine ligand and protein latent representations as well as a self-attention layer applied to ligand embeddings prior to convolutional layers.

SimCNN-DTA [36] is a method that applies a 2D CNN to the outer products between column vectors of Tanimoto similarity matrix of the drugs and column vectors of Smith–Waterman similarity matrix of the targets for continuous DTA prediction. The product of two vectors – a 2D matrix – is used as an input for the neural network, consisting of two convolutional layers with max pooling and two fully connected layers.

A FingerDTA [37] model uses two representations of both drug and target. 1D-fingerprints (extended connectivity fingerprints for ligand representations and word2vec-based encoder of amino acid sequences for target representations) are processed by FC-blocks, while two-dimensional one-hot encoded amino-acid sequence and SMILES string representations serve as inputs for 1D-CNN layers. Afterwards, the vector products of outputs of FC-blocks and CNN blocks for both protein and ligand are computed. This operation plays a role of an attention mechanism, introducing global information generated by CNNs. The resulting vectors get concatenated and serve as an input of the final regression FC-block, that generates binding affinity prediction.

The best self-reported* benchmarking results on the Davis dataset (CI=0.907) among the models, which are in the scope of the current review, is demonstrated by the DTITR model released in 2022 [38]. In this architecture protein and ligand embeddings

are processed by transformer-encoder blocks followed by cross-attention transformer-encoder blocks, enabling information exchange between protein and ligand branches of the networks. The latent representations given by both branches are concatenated and processed by an MLP as it was implemented in the most basic DeepDTA architecture.

A similar level of performance was achieved by GSAML-DTA model, in which the ligand graph and protein graph (constructed from a contact map) are processed by GAT and GCN layers in parallel with a subsequent self-attention operation. The resulting vectors are concatenated to form learned ligand and protein feature vectors, which are concatenated with each other before being processed by an MLP. Denoising of learned features is performed using the mutual information principle [39].

There are several other recently published DNN architectures which are applied in the same domain, and which demonstrate performance close to the previously described ones, using various architectural innovations. The MGraphDTA model uses multiscale graph convolutions, conceptually similar to residual CNNs for image processing, where skip-connections allow to propagate information from prior layers passing by some subsequent layers. This technique allows to mitigate the vanishing gradient problem (the weights of the network remain unchanged as the derivative vanishes, so the network cannot learn) as well as to learn both global and local features of the graph [40]. In FusionDTA the fusion layer, which consists of several multi-head attention blocks, applied to protein sequence, ligand SMILES string and a combined latent representation of them [41]. The ELECTRA-DTA is an example of an alternative approach to the problem, in which the embeddings for protein and ligand string representations were generated by the ELECTRA language model, which demonstrated better efficiency then BERT for small model sizes. Before being used for binding affinity predictions, ELECTRA is pre-trained on SMILES and sequences corpora extracted from PubChem and Uniprot. The latter part of the network consists of separate CNN branches with SE modules for protein and ligand [42]. In a method proposed by Ma et. al. in [43] the protein features are extracted by the SAGE network, while the ligand graph is processed by a two-layer Simple Graph Convolution model (SGC), which is a variant of a GCN without non-linearity between GCN layers. This technique allows to simplify the network, keeping its performance comparable to regular GCNs.

3D-CNN models for structure-based affinity prediction

Three-dimensional CNN architectures are naturally applicable to three-dimensional chemical data such as the 3D structures of protein-ligand complexes. One of the first implementations of a CNN network able to operate on chemical structural data and applied to the binding affinity prediction problem, was Pafnucy [44] released in 2017. The model uses the voxelized representation of the complex as a 4D tensor, where the first three dimensions correspond to the coordinates in a Cartesian space, the 4th dimension being a vector of 19 features, encoding atom types, atom hybridization, number of bonds with heavy atoms and heteroatoms, binary properties such as hydrophobicity, aromaticity, electron pair donor or acceptor and being a part of a ring. The neural network consists of three convolutional layers with cubic filters, the flattened output of the final layer is processed by a 3-layer fully connected network, the final layer of which outputs a single value.

A major weakness of a convolutional architecture is its vulnerability to affine transformations, which may lead to a significant difference in predicted values for an input and its affine-transformed (e.g. rotated) copy. Thus, data augmentation with transformed inputs is used to minimize the influence of this effect. In the Pafnucy model, 24 different orientations of a complex (all combinations of 90° rotations) are used.

The KDEEP model [45] also relies on a slightly alternative approach to grid featurization of a 3D structure of the complex. The 3D atomic features (hydrophobic, hydrogen-bond donor or acceptor, aromatic, positive or negative ionizable, metallic and total excluded volume) are generated for protein and ligand atoms. Each grid point feature vector of length 16 (concatenated vector for protein and ligand features) accumulates the feature values from all atoms inside the grid box with respect to its distance from the contributing atom, the contribution dependency on distance being similar to the VdW term.

Repurposing of previously published architectures due to their universality became a common practice in machine-learning. For instance, KDEEP relies on a SqueezeNet network [46] — a model initialy developed for image classification, which

was simplified (less convolutional layers used) and adoped to processing of a multi-channel 3D-grid. The convolutional modules in this architecture consist of "expand" and "squeeze" modules. The former performs convolutions with 1x1 and 3x3 kernels, the outputs of which are further merged, forming a tensor with a x8 higher number of filters then in the input. The "squeeze" module performs convolution with 1x1 kernels.

The DeepAtom model [47] is an adaptation of another convolutional architecture for image processing — ShuffleNet [48]. The 3D grid of a protein ligand complex prepared similarly to the previous approaches is first processed by 1x1 convolutional filters, followed by shuffle blocks, producing a 1024x2x2x2 tensor as an output. The last module – a "global affinity regressor" splits this tensor into 8 single-dimensional vectors of length 1024. Because of previously applied shuffle modules, the receptive field of all eight vectors fully covers the input grid box, therefore it can be processed by an MLP module, which yields the final prediction used for loss computation.

A more sophisticated approach to protein-ligand binding affinity prediction was proposed by Gomes et. al. [49] and implemented in atomic convolutional neural networks (ACNN). This method models the thermodynamic cycle, predicting Gibbs free energies of formation for protein, ligand, and protein-ligand complex and computes the target value — $\Delta_f G$ of the complex from apo-protein and unbound ligand according to the Eq. 1.16:

$$\Delta_f G = G_{complex} - G_{protein} - G_{ligand} \tag{1.16}$$

All three terms on the right side are predicted by identical NNs with shared weights. The loss function value used for backpropagation is computed for the $\Delta_f G$. The thermodynamic cycle is thus directly included into model optimization.

Unlike in previously discussed CNNs in the ACNN model an alternative featurization of input is applied. Instead of working with a 4D tensor analogous to a 3D image with multiple color channels, the authors represent a protein-ligand complex in a form of two 2D matrices —- a distance matrix and an atom type matrix. The atom-type convolution is performed on these two matrices yielding a 3D tensor of a shape (N, M, A), where N equals to the number of atoms in a complex, M is equal to the

number of neighbors considered in a proximal environment of each atom, A corresponds to the number of unique atom types. A radial pooling layer is further applied to the atom type expansion tensor. The output has shape (N, N_{at}, N_r), where N_r is a number of radial filters. It is flattened thus yielding a tensor $E(N, N_{at} * N_r)$, which is further processed by a fully connected network, which treats rows corresponding to different atoms separately, that produces an output of shape $(N, 1)$, where E_i is an energy of the i-th atom. The total energy of the molecule is computed as a sum of individual atomic contributions and is thus invariant to atom permutation. This architecture, the fully connected part of which does not depend on the size of the system (nor the number of atoms due to the atom-wise application of the neural network, nor the volume of it due to the featurization approach, which is used) does not have technical limitations on applicability to systems of various size contrary to 3D grid convolution methods.

In 2017 the better performance of a CNN-based Gnina scoring function [50] in comparison with the Autodock Vina scoring function was demonstrated in a D3R 2017 challenge [51].

1D and 2D CNN models for structure-based affinity prediction

The TopologyNet [52] model introduced in 2017 used a multichannel 1D topological invariants of protein-ligand complexes, which are processed by 1D-convolutional network. The TopDP-DL model [53] takes three inputs: a 2D topological map of a protein-ligand complex and two 1D barcode vectors representing interatomic distances and charges. The outputs of convolution blocks are therefore flattened and concatenated before being further processed by fully connected layers.

In OnionNet [54] a relatively lightweight model consisting of 3 2D convolutional layers followed by three fully connected layers was used for binding affinity prediction from structure, showing state-of-the art results on PDBbind core set benchmarks. A 2D-matrix of a shape (60x64), where y coordinate corresponded to the interactions in all possible combinations of atom types in ligand and in protein and x coordinate corresponded to the index of a shell around a certain atom in which an interaction may take place (in total 60 shells with 0.5 Å width each starting from 1 Å distance from the atom center and covering the range of 30.5 Å around the atom of interest) –

a modified approach first introduced for the RF-score model [55].

An improved version of OnionNet called SE-OnionNet [56], which was published in 2021, uses the same featurization approach as the original model, but the CNN architecture used there is modified —- between CNN layers 1 and 2, 2 and 3 the squeeze-and-exitation blocks, which were first proposed in SE-Net [57] are inserted. These blocks allow the model to learn relationships between different feature maps produced by convolutional layers. On PDBbind 2016 core set, the SE-Onionnet model demonstrated an increase of R_p by 0.037 (R_p=0.853 vs R_p=0.816 for the original model).

Graph Neural Network Architectures for structure-based affinity prediction

Although unlike convolutional architectures, graph neural network architectures make direct learning on graph data possible, engineering of node and edge features still plays an important role in the model development as well as the choice of how to represent graph-like structures, which, unlike organic small molecules, do not have a standard definition or a standard form of representation.

In GraphDelta [58] — a graph neural network model based on the MPNN architecture, the input consists of ligand graphs, node features of which included Behler-Parinello-like symmetric functions, capturing the information of the protein environment of a given ligand atom. A particularity of the GNN architecture used in GraphDelta was the readout implementation —- instead of more common sum or average of node features a fuzzy histogram approach indroduced by Kearnes et. al. [59] was used to capture the distribution of node feature values and prevent information loss at the readout stage.

A GraphBAR [60] model is an example of utilization of a GCN architecture for binding affinity prediction. The graph representation of a complex applied in this approach considers interatomic distances of different range between all atoms in the binding site, which are represented in a form of several adjacency matrices for various distance cutoffs, graph convolutions on all of them independently in different graph convolutional blocks.

An important class of binding affinity prediction models are the architectures,

in which protein, ligand, and interaction inputs are processed in independent clocks of the network, which makes the examination of a contribution of each of the inputs into the final prediction. One of the examples of this approach is a DEELIG model [61]. In the corresponding study, two versions of the model were examined —- the one, that included only the 3D-convolutional block, which took a protein-ligand feature grid as an input similarly to previously discussed 3D-CNN models; and the "composite" model, which includes an additional fully connected block, which processes a ligand fingerprint. The study shows superiority of the composite model over the basic one ($\Delta R_p = 0.03$). In InteractionGraphNet all three inputs are represented as graphs. First, the intramolecular convolutions are performed on ligand and protein graphs, then the intermolecular graph convolution block performs message passing over edges in the interaction graph [62].

In order to combine the strengths of 3D-CNNs and GNNs, an architecture consisting of two independent GNN blocks and one 3D-CNN block was suggested in HAC-Net [63]. The CNN part includes SE modules, the GNN part is based on graph attentional aggregation similar to the mechanism used in GGNNs.

The SS-GNN [64] model used a single undirected graph of protein-ligand interactions as an input, the core of which consists of the nodes corresponding to ligand atoms connected via edges, corresponding to ligand covalent bonds. Protein nodes are included into the graph, if the corresponding atoms are located closer to certain ligand atoms, then the preset distance cutoff. The neural network architecture include two GIN layers for node feature extraction and a three layer MLP for edge feature aggregation. The node features generated by GIN and edge feature generated by MLP are concatenated, producing a resulting edge-level set of features, which are then processed by another MLP before the final pooling operation (the sum of all edge features), which becomes a final output value generated by the model.

The PIGNet [65] model relies on a similar initial representation of the protein-ligand complex. Two adjacency matrices representing covalent and non-covalent bonds are used to discriminate intramolecular bonds and intermolecular interactions. The Gated GAT module performing node feature update using the adjacency matrix of covalent interactions followed by the interaction network, which considers a feature

matrix of non-covalent intermolecular interactions and generates node-level latent representations, from which energy terms are computed. These terms correspond to VdW energy, H-bonds energy, hydrophobic contacts and metal chelatation. The energy contribution of each interaction is computed as a sum of these terms divided by the rotor penalty in order to consider an entropy contribution into the protein-ligand binding energy. The total energy of a complex is calculated as a sum of affinity contributions of protein-ligand interactions. The loss function implemented in this study, in addition to the MSE loss of energy prediction, contains two additional terms – a derivative loss to learn the shape of the potential energy curve and a loss term, which considers the contribution of data augmentation to the predicted binding energy.

In a study by Moesser et. al. [66], several combined model architectures were analyzed. The authors used a combination of an MLP operating on ligand fingerprints and a GNN block operating on a graph representation of a ligand. The third optional block was a 1D CNN processing the protein sequence. Two alternative graph representations were used — a ligand graph with atom-level features generated by RDKit, and another representation, in which a feature vector that reflected the presence of each of 22 possible protein atom types in the environment of a ligand atom, was concatenated with the RDKit atomic feature vector (the PLIG model). The list of examined GNN architectures included GAT, GCN, GIN, SGC, SageNet and GAT+GCN combinations. The architecture based on GAT with PLIG graph representation demonstrated the highest performance on PDBbind 2016 core set.

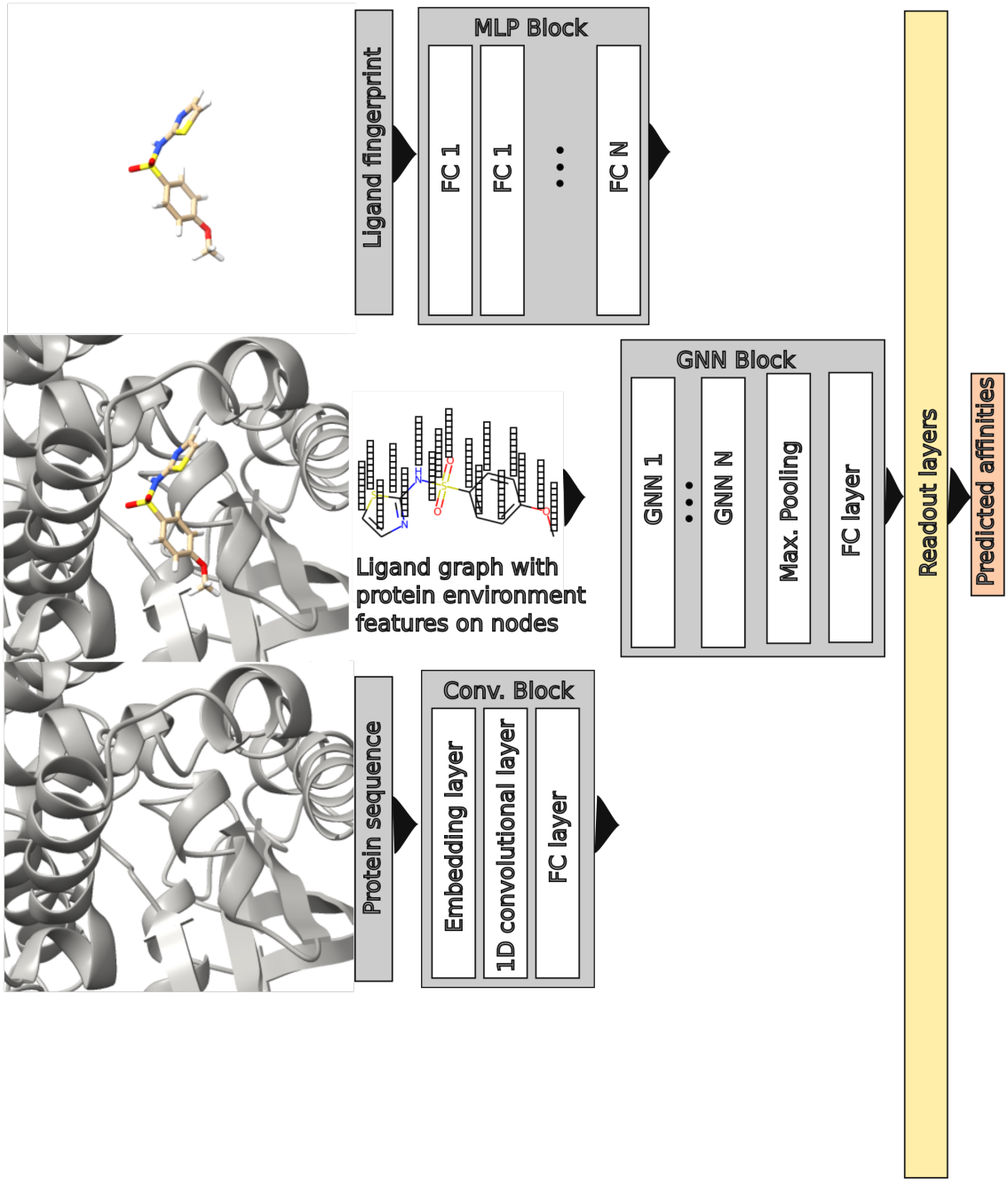


Figure 1.3 An example of a NN architecture for binding affinity prediction (Moesser et al. [66]) that combines protein, ligand, and interaction inputs.

1.4.2 Training, Validation, and Test Set Composition

The construction of training, validation, and test sets plays a crucial role in machine learning and determines the applicability domain of the resulting model. Moreover, for a rigorous comparison of ML-models solving the same problem, a uniformity of test sets used by developers of the corresponding methods is necessary to make a comparison of self-reported results possible. In publications, discussing the development of machine-learning models for binding affinity prediction from protein and ligand information only using the non-structural datasets of kinase inhibitors activity, a common approach used for train-test splitting is the one published be He et al. [27] introducing a GBT-based SimBoost architecture. Training and validation of a model there is performed using a five-fold cross validation, the folds are constructed in such a way, that each protein target is included at least into two folds, to ensure, that it its used both for training and testing. The cross-validation is performed ten times, and the reported values include the mean and the SD of each metric. The procedure is independent for each of the datasets used in the study [27]. In order to verify the generalization capabilities of a model, the developers of DeepDTA modified this approach, splitting the dataset into six equal parts. Five parts are therefore used for model training and hyperparameter optimization (four of five subsets are used for training, the fifth one becomes the validation set), the remaining one plays a role of an independent test set. The reported metric values are measured on the test set using predictions given by five models trained previously.

For structure-based binding affinity prediction models, the PDBbind core set usually serves as the benchmarking set, which allows to compare different ML-based scoring functions. At the same time, the core set reportedly possesses biases to protein and ligand structures, which makes it possible to get relatively high scores on the benchmark with the models, predicting binding affinity exclusively from protein or ligand structure [81, 82]. In order to better examine the generalization capabilities of a resulting model, additional more complex datasets for validations are often constructed, but the absence of an acknowledged benchmarking set complexifies the comparison of affinity predictors on more advanced datasets.

Another issue the developers of structure-based affinity prediction models have

Table 1.1. Evaluation of binding affinity prediction models on PDBbind 2016 core set

Model	Ref	Year	Architecture	MSE	RMSE	R^2	R_p	Spearman	SD	MAE
HAC-Net	[63]	2022	GCN+GNN	0.971	1.205	0.690	0.846	0.843		
SS-GNN	[64]	2022	GIN		1.181		0.853			
PIGNet	[65]	2022	GAT				0.760			
PointTransformer	[67]	2022	GAT		1.190		0.853			0.923
PLIG	[66]	2022	GAT		1.220		0.840			
SFCNN	[68]	2022	CNN		1.326		0.793		1.325	1.027
SPE-MONN-UniRep	[69]	2022	GCN+CNN		1.272		0.823			
InteractionGraphNet	[62]	2021	MPNN		1.220	0.690	0.837			0.940
SIGN	[67]	2021	GAT		1.316		0.797		1.312	1.027
HPC-HWPC	[70]	2021	HPC		1.307		0.831			
FAST	[71]	2021	CNN+SG_CNN		1.871		0.712	0.693		1.498
BAPA	[72]	2021	CNN+ATT		1.308		0.819	0.819	1.247	1.021
GraphBar	[60]	2021	GCN		1.413		0.778		1.371	1.144
DEELIG	[61]	2021	GCN+MLP				0.889			
ECIF6_LD_GBT	[73]	2021	GBT		1.169		0.866			
SE-OnionNet	[56]	2021	CNN		1.592		0.853		1.253	0.912
LigityScore3D	[74]	2021	CNN				0.725		1.500	
S-MAN	[75]	2020	GAT		1.359		0.786		1.347	1.093
GraphDelta	[58]	2020	MPNN		1.050		0.870			
Gnina	[50]	2020	CNN		1.370		0.800			
DeepAtom	[47]	2019	CNN		1.318		0.807		1.286	1.039
PLEC-nn	[76]	2019	MLP				0.817		1.256	
OnionNet	[54]	2019	CNN		1.278		0.816		1.257	0.984

Table 1.2. Evaluation of binding affinity prediction models on PDBbind 2013 core set

Model	Ref	Year	Architecture	MSE	RMSE	R^2	R_p	Spearman
SS-GNN	[64]	2022	GIN	1.347	0.816			
SFCNN	[68]	2022	CNN	1.452	0.795		1.417	1.114
SPE-MONN-UniRep	[69]	2022	GCN+CNN	1.412	0.797			
HPC-HWPC	[70]	2021	HPC	1.483	0.784			
BAPA	[72]	2021	CNN+ATT	1.457	0.771	0.774	1.443	1.170
GraphBar	[60]	2021	GCN	1.688	0.670		1.669	1.367
DEELIG	[61]	2021	GCN+MLP		0.894			
SE-OnionNet	[56]	2021	CNN	1.692	0.812		1.423	1.323
LigityScore3D	[74]	2021	CNN		0.713		1.580	
PLEC-nn	[76]	2019	MLP		0.774		1.426	
OnionNet	[54]	2019	CNN	1.503	0.782		1.445	1.208
Pafnucy	[44]	2018	CNN		0.700		1.610	

Table 1.3. Evaluation of binding affinity prediction models on PDBbind 2007 core set

Model	Ref	Year	Architecture	RMSE	R^2	R_p	Spearman
SPE-MONN-UniRe	[69]	2022	GCN+CNN	1.429		0.822	
HPC-HWPC	[70]	2021	HPC	1.403		0.829	
PotentialNet	[77]	22018	GCNN		0.668	0.822	0.826
1D2D	[53]	2018	CNN	1.950		0.806	
TopologyNet	[52]	2017	CNN	1.370		0.826	

Table 1.4. Evaluation of binding affinity prediction models on Davis dataset

Model	Ref	Year	Architecture	MSE	RMSE	CI	R^2	R_p	Spearman	AUPR
S2GC+SAGE	[43]	2023	GCN	0.23		0.895		0.85		
FingerDTA	[37]	2023	CNN	0.23		0.895				
DTITR	[38]	2022	Transformer	0.19	0.438	0.907	0.77		0.712	
GSAML-DTA	[39]	2022	GAT	0.2			0.72			
MgraphDTA	[40]	2022	MPNN+CNN	0.21		0.900				
FusionDTA	[41]	2022	LSTM+Att	0.21		0.913				
Affinity2Vec	[29]	2022	XGBoost	0.24		0.886				
ELECTRA_DT	[42]	2022	ELECTRA+CNN	0.24		0.897		0.84		0.70
GraphDTA	[34]	2021	GCN	0.25		0.880				
GraphDTA	[34]	2021	GAT_GCN	0.25		0.881				
GraphDTA	[34]	2021	GAT	0.23		0.892				
GraphDTA	[34]	2021	GIN	0.23		0.893				
MATT_DTI	[35]	2021	CNN+Att	0.23		0.890				
MDeePred	[78]	2021	CNN		0.505	0.886			0.690	0.74
DeepAffinity	[79]	2021	RNN+RNN		0.503	0.900				
DeepAffinity	[79]	2021	RNN+GCN		0.511	0.881				
DeepAffinity	[79]	2021	CNN_GCN		0.811	0.737				
DeepAffinity	[79]	2021	HRNN_GCN		0.502	0.881				
DeepAffinity	[79]	2021	HRNN_GIN		0.660	0.822				
DGraphDTA	[33]	2020	GCN	0.20		0.904	0.70	0.87		
WideDTA	[80]	2019	CNN	0.26		0.886		0.82		

Table 1.4. Evaluation of binding affinity prediction models on Davis dataset

Model	Ref	Year	Architecture	MSE	RMSE	CI	R^2	R_p	Spearman	AUPR
Attention1DTA	[32]	2019	GAT	0.22		0.893	0.68			0.78
Attention2DTA	[32]	2019	GAT	0.22		0.886	0.68			0.78
DeepDTA	[31]	2018	CNN	0.26		0.878	0.63			0.71

Table 1.5. Evaluation of binding affinity prediction models on KIBA dataset

Model	Ref.	Year	Architecture	MSE	RMSE	CI	R^2	R_p	Spearman	AUPR
S2GC+SAGE	[43]	2023	GCN	0.127		0.903		0.887		
FingerDTA	[37]	2023	CNN	0.150		0.885				
GSAML-DTA	[39]	2022	GAT	0.132			0.800			
MgraphDTA	[40]	2022	MPNN+CNN	0.128		0.902				
FusionDTA	[41]	2022	LSTM+Att	0.13		0.906				
Affinity2Vec	[29]	2022	XGBoost	0.124		0.905				
ELECTRA_DTA	[42]	2022	ELECTRA+CNN	0.162		0.889		0.879		0.795
GraphDTA	[34]	2021	GCN	0.179		0.866				
GraphDTA	[34]	2021	GAT_GCN	0.147		0.882				
GraphDTA	[34]	2021	GAT	0.139		0.889				
GraphDTA	[34]	2021	GIN	0.139		0.891				
MATT_DTI	[35]	2021	CNN+Att	0.150		0.889				
DeepAffinity	[79]	2021			0.4335	0.842				
DeepAffinity	[79]	2021			0.5367	0.796				
DeepAffinity	[79]	2021			0.8244	0.576				
DeepAffinity	[79]	2021			0.448	0.842				
DeepAffinity	[79]	2021			0.6669	0.689				
DGraphDTA	[33]	2020	GCN	0.126		0.904	0.786	0.903		
WideDTA	[80]	2019	CNN	0.179		0.875		0.856		
Attention1DTA	[32]	2019	GAT	0.155		0.882	0.755			0.829
Attention2DTA	[32]	2019	GAT	0.162		0.88	0.738			0.811
DeepDTA	[31]	2018	CNN	0.194		0.863	0.673			0.673

to deal with is the limited size of available datasets of experimentally obtained protein-ligand structures, which even causes concerns about the applicability of these data for training of general purpose machine learning models for binding affinity prediction [83]. The most basic approaches, which do not incorporate data augmentation (besides augmentation with rotations of a grid box, which is an irreplaceable procedure in 3D-CNNs), involve training on the general + refined ([62]) or only the refined set of PDBbind [44, 45], with the core set excluded, a part of the dataset is taken out and served as a validation set (e.g. 1000 randomly selected complexes in [44]). Among the data augmentation approaches the procedure implemented in [65], consisting of docking augmentation, random screening augmentation and cross screening augmentation deserves mentioning. In this study each of the augmentation techniques generated more than 100 000 complexes, which were used for model training. Despite the observed increase of docking success rate and screening power of a model trained on an augmented dataset, the scoring benchmark showed lower performance in comparison with a model trained exclusively on PDBbind. No performance increase of models trained on docking pose augmented datasets is also reported in [54, 60, 84].

1.5 Evaluation metrics

Concordance index – CI

The CI of two sets of data is equal to the probability of two random pairs with different label values to be predicted in the correct order.

$$CI = 1/Z \sum_{dx>dy} h(bx - by) \tag{1.17}$$

where bx and by are predicted values, for higher and lower affinities (dx, dy), Z is the normalization constant, $h(x)$ is the Heavyside function

$$h(x) = \begin{cases} 1, \, if \, x \, > \, 0 \\ 0.5, \, if \, x \, = \, 0 \\ 0, \, if \, x \, < \, 0 \end{cases} \tag{1.18}$$

Pearson correlation coefficient – $\mathbf{R_p}$

$$R_p = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \overline{y})^2}} \tag{1.19}$$

where N is the sample size, x_i the experimental value, y_i the predicted value, $\overline{x}$ the mean of experimental values, and $\overline{y}$ the mean of predicted values.

Root mean squared error – RMSE

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - y_{pred})^2}{N}} \tag{1.20}$$

where N is the sample size, y_i the experimental value, y_{pred} the predicted value.

Determination coefficient – $\mathbf{R^2}$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - y_{pred})^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2} \tag{1.21}$$

where $\overline{y}$ is the mean value of y: $\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$, x_i — the experimental value, y_i — the predicted value.

Standard deviation – SD

$$SD = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \overline{y})^2}{N-1}} \tag{1.22}$$

where N is the total sample size, $\overline{y}$ is the mean value of y in a sample, y_i — the predicted value.

Mean squared error - MSE

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_{pred})^2 \tag{1.23}$$

where N is the total sample size, y_i — the experimental value, y_{pred} — the predicted value.

Mean absolute error – MAE

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - y_{pred}| \tag{1.24}$$

where N is the total sample size, y_i — the experimental value, y_{pred} — the predicted value.

Spearman's coefficient

$$\rho = \frac{\frac{1}{N}\sum_{i=1}^{N}(R(y_i)) - \overline{R(y)}) \cdot (R(\hat{y}_i) - \overline{R(\hat{y})})}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(R(y_i) - \overline{R(y)})^2 \cdot \sum_{i=1}^{N}(R(\hat{y}_i) - \overline{R(\hat{y})})^2}} \tag{1.25}$$

where $\overline{R(\hat{y})}$ is the predicted value rank, $R(y_i$ is the real value rank, $\overline{R(\hat{y})})$ is the mean of value ranks predicted by the model, $\overline{R(y)}$ is the mean of the real ranks [38].

1.6 Conclusions

Deep learning application to drug design is a rapidly developing field of research, and the development of DNN-based binding affinity prediction models is one of the topics the research community is focused on. At the moment, the leaderboard of both structure based and sequence/text-based models for this purpose consists of multiple models, relying on different architectures, although the inclusion of GNN architectures and attention mechanisms becomes more and more common in the most recent publications. For structure-based affinity prediction the low amount of available experimental data remains an actual problem, which causes interest in the implementation of data augmentation techniques. Despite the progress made in the field of development of structure-based DNN models for binding affinity prediction, the biases to protein and ligand information, which are found in popular benchmarking sets, cause concerns about potential practical applicability of such models for completely new data.

This review covers a broad range of emerging field of research, thus for deeper review of details, which are out of the scope of the current chapter, the author can recommend several reviews of adjacent topics. A deeper review of graph neural architectures with respect to their genealogy as well as to their applications can be found in a publication by Zhou et. at. [14], similarly, for a more complete review of convolutional neural network architectures we refer to Gu et al. [10]. A publication by Meli et. al. [85] and an older paper by Li et al. [86] offer a broader discussion of structure-based ML models for binding affinity prediction.

1.6.1 References

(1) Ciociola, A. A.; Cohen, L. B.; Kulkarni, P.; Kefalas, C.; Buchman, A.; Burke, C.; Cain, T.; Connor, J.; Ehrenpreis, E. D.; Fang, J., et al. *Official journal of the American College of Gastroenterology— ACG* **2014**, *109*, 620–623.

(2) Sun, D.; Gao, W.; Hu, H.; Zhou, S. *Acta Pharmaceutica Sinica B* **2022**, *12*, 3049–3062.

(3) Genheden, S.; Ryde, U. *Expert opinion on drug discovery* **2015**, *10*, 449–461.

(4) Nguyen, D. D.; Wei, G.-W. *Journal of chemical information and modeling* **2019**, *59*, 3291–3304.

(5) Liu, Z.; Su, M.; Han, L.; Liu, J.; Yang, Q.; Li, Y.; Wang, R. *Accounts of chemical research* **2017**, *50*, 302–309.

(6) Li, J.; Fu, A.; Zhang, L. *Interdisciplinary Sciences: Computational Life Sciences* **2019**, *11*, 320–328.

(7) Atlas, L.; Homma, T.; Marks, R. *Neural Information Processing Systems* **1987**, ed. by Anderson, D., 31–40.

(8) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. *Communications of the ACM* **2017**, *60*, 84–90.

(9) He, K.; Zhang, X.; Ren, S.; Sun, J. *Proceedings of the IEEE conference on computer vision and pattern recognition* **2016**, 770–778.

(10) Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J., et al. *Pattern recognition* **2018**, *77*, 354–377.

(11) Kipf, T. N.; Welling, M. *arXiv preprint arXiv:1609.02907* **2016**.

(12) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. In *International conference on machine learning*, 2017, pp 1263–1272.

(13) Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. *arXiv preprint arXiv:1710.10903* **2017**.

(14) Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. *AI open* **2020**, *1*, 57–81.

(15) Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. *arXiv preprint arXiv:1810.00826* **2018**.

(16) Renaud, J.; Chari, A.; Ciferri, C. *Nature reviews Drug discovery* **2018**, *17*, 471–492.

(17) Wang, R.; Fang, X.; Lu, Y.; Wang, S. *Journal of medicinal chemistry* **2004**, *47*, 2977–2980.

(18) Liu, Z.; Li, Y.; Han, L.; Li, J.; Liu, J.; Zhao, Z.; Nie, W.; Liu, Y.; Wang, R. *Bioinformatics* **2015**, *31*, 405–412.

(19) PDBBind, `http://www.pdbbind.org.cn/`, Accessed: 2022-03-12.

(20) Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. *Journal of chemical information and modeling* **2018**, *59*, 895–913.

(21) Hu, L.; Benson, M. L.; Smith, R. D.; Lerner, M. G.; Carlson, H. A. *Proteins: Structure, Function, and Bioinformatics* **2005**, *60*, 333–340.

(22) Smith, R. D.; Clark, J. J.; Ahmed, A.; Orban, Z. J.; Dunbar Jr, J. B.; Carlson, H. A. *Journal of molecular biology* **2019**, *431*, 2423–2433.

(23) Ahmed, A.; Smith, R. D.; Clark, J. J.; Dunbar Jr, J. B.; Carlson, H. A. *Nucleic acids research* **2015**, *43*, D465–D469.

(24) Benson, M. L.; Smith, R. D.; Khazanov, N. A.; Dimcheff, B.; Beaver, J.; Dresslar, P.; Nerothin, J.; Carlson, H. A. *Nucleic acids research* **2007**, *36*, D674–D678.

(25) Hartshorn, M. J.; Verdonk, M. L.; Chessari, G.; Brewerton, S. C.; Mooij, W. T.; Mortenson, P. N.; Murray, C. W. *Journal of medicinal chemistry* **2007**, *50*, 726–741.

(26) Metz, J. T.; Johnson, E. F.; Soni, N. B.; Merta, P. J.; Kifle, L.; Hajduk, P. J. *Nature chemical biology* **2011**, *7*, 200–202.

(27) He, T.; Heidemeyer, M.; Ban, F.; Cherkasov, A.; Ester, M. *Journal of cheminformatics* **2017**, *9*, 1–14.

(28) Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; Zarrinkar, P. P. *Nature biotechnology* **2011**, *29*, 1046–1051.

(29) Thafar, M. A.; Alshahrani, M.; Albaradei, S.; Gojobori, T.; Essack, M.; Gao, X. *Scientific reports* **2022**, *12*, 4751.

(30) Tang, J.; Szwajda, A.; Shakyawar, S.; Xu, T.; Hintsanen, P.; Wennerberg, K.; Aittokallio, T. *Journal of Chemical Information and Modeling* **2014**, *54*, 735–743.

(31) Öztürk, H.; Özgür, A.; Ozkirimli, E. *Bioinformatics* **2018**, *34*, i821–i829.

(32) Zhao, Q.; Xiao, F.; Yang, M.; Li, Y.; Wang, J. *2019 IEEE international conference on bioinformatics and biomedicine (BIBM)* **2019**, 64–69.

(33) Jiang, M.; Li, Z.; Zhang, S.; Wang, S.; Wang, X.; Yuan, Q.; Wei, Z. *RSC advances* **2020**, *10*, 20701–20712.

(34) Nguyen, T.; Le, H.; Quinn, T. P.; Nguyen, T.; Le, T. D.; Venkatesh, S. *Bioinformatics* **2021**, *37*, 1140–1147.

(35) Zeng, Y.; Chen, X.; Luo, Y.; Li, X.; Peng, D. *Briefings in bioinformatics* **2021**, *22*, bbab117.

(36) Janfaza, V.; Weston, K.; Razavi, M.; Mandal, S.; Muzahid, A. *arXiv e-prints* **2021**, arXiv–2110.

(37) Zhu, X.; Liu, J.; Zhang, J.; Yang, Z.; Yang, F.; Zhang, X. *Big Data Mining and Analytics* **2022**, *6*, 1–10.

(38) Monteiro, N. R.; Oliveira, J. L.; Arrais, J. P. *Computers in Biology and Medicine* **2022**, *147*, 105772.

(39) Liao, J.; Chen, H.; Wei, L.; Wei, L. *Computers in Biology and Medicine* **2022**, *150*, 106145.

(40) Yang, C.; Chen, E. A.; Zhang, Y. *Molecules* **2022**, *27*, 4568.

(41) Yuan, W.; Chen, G.; Chen, C. Y.-C. *Briefings in Bioinformatics* **2022**, *23*, bbab506.

(42) Wang, J.; Wen, N.; Wang, C.; Zhao, L.; Cheng, L. *Journal of cheminformatics* **2022**, *14*, 1–14.

(43) Ma, D.; Li, S.; Chen, Z. *Mathematical Biosciences and Engineering* **2023**, *20*, 269–282.

(44) Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P. *Bioinformatics* **2018**, *34*, 3666–3674.

(45) Jiménez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. *Journal of chemical information and modeling* **2018**, *58*, 287–296.

(46) Iandola, F. N.; Han, S.; Moskewicz, M. W.; Ashraf, K.; Dally, W. J.; Keutzer, K. *arXiv preprint arXiv:1602.07360* **2016**.

(47) Li, Y.; Rezaei, M. A.; Li, C.; Li, X. *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* **2019**, 303–310.

(48) Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. *Proceedings of the European conference on computer vision (ECCV)* **2018**, 116–131.

(49) Gomes, J.; Ramsundar, B.; Feinberg, E. N.; Pande, V. S. *arXiv preprint arXiv:1703.10603* **2017**.

(50) Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. *Journal of chemical information and modeling* **2017**, *57*, 942–957.

(51) Sunseri, J.; King, J. E.; Francoeur, P. G.; Koes, D. R. *Journal of computer-aided molecular design* **2019**, *33*, 19–34.

(52) Cang, Z.; Wei, G.-W. *PLoS computational biology* **2017**, *13*, e1005690.

(53) Cang, Z.; Mu, L.; Wei, G.-W. *PLoS computational biology* **2018**, *14*, e1005929.

(54) Zheng, L.; Fan, J.; Mu, Y. *ACS omega* **2019**, *4*, 15956–15965.

(55) Ballester, P. J.; Mitchell, J. B. *Bioinformatics* **2010**, *26*, 1169–1175.

(56) Wang, S.; Liu, D.; Ding, M.; Du, Z.; Zhong, Y.; Song, T.; Zhu, J.; Zhao, R. *Frontiers in Genetics* **2021**, *11*, 607824.

(57) Hu, J.; Shen, L.; Sun, G. *Proceedings of the IEEE conference on computer vision and pattern recognition* **2018**, 7132–7141.

(58) Karlov, D. S.; Sosnin, S.; Fedorov, M. V.; Popov, P. *ACS omega* **2020**, *5*, 5150–5159.

(59) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. *Journal of computer-aided molecular design* **2016**, *30*, 595–608.

(60) Son, J.; Kim, D. *PloS one* **2021**, *16*, e0249404.

(61) Ahmed, A.; Mam, B.; Sowdhamini, R. *Bioinformatics and Biology Insights* **2021**, *15*, 1–9.

(62) Jiang, D.; Hsieh, C.-Y.; Wu, Z.; Kang, Y.; Wang, J.; Wang, E.; Liao, B.; Shen, C.; Xu, L.; Wu, J., et al. *Journal of medicinal chemistry* **2021**, *64*, 18209–18232.

(63) Kyro, G. W.; Brent, R. I.; Batista, V. S. *arXiv preprint arXiv:2212.12440* **2022**.

(64) Zhang, S.; Jin, Y.; Liu, T.; Wang, Q.; Zhang, Z.; Zhao, S.; Shan, B. *arXiv preprint arXiv:2206.07015* **2022**.

(65) Moon, S.; Zhung, W.; Yang, S.; Lim, J.; Kim, W. Y. *Chemical Science* **2022**, *13*, 3661–3673.

(66) Moesser, M. A.; Klein, D.; Boyles, F.; Deane, C. M.; Baxter, A.; Morris, G. M. *bioRxiv* **2022**, 2022–03.

(67) Wang, Y.; Wu, S.; Duan, Y.; Huang, Y. *Briefings in Bioinformatics* **2022**, *23*, bbab474.

(68) Wang, Y.; Wei, Z.; Xi, L. *BMC bioinformatics* **2022**, *23*, 222.

(69) Wu, J.; Liu, Z.; Yang, X.; Lin, Z. *BMC bioinformatics* **2022**, *23*, 1–12.

(70) Liu, X.; Wang, X.; Wu, J.; Xia, K. *Briefings in Bioinformatics* **2021**, *22*, bbaa411.

(71) Jones, D.; Kim, H.; Zhang, X.; Zemla, A.; Stevenson, G.; Bennett, W. D.; Kirshner, D.; Wong, S. E.; Lightstone, F. C.; Allen, J. E. *Journal of chemical information and modeling* **2021**, *61*, 1583–1592.

(72) Seo, S.; Choi, J.; Park, S.; Ahn, J. *BMC bioinformatics* **2021**, *22*, 1–15.

(73) Sánchez-Cruz, N.; Medina-Franco, J. L.; Mestres, J.; Barril, X. *Bioinformatics* **2021**, *37*, 1376–1382.

(74) Azzopardi, J.; Ebejer, J.-P. *Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2021* **2021**, *3: Bioinformatics*, 38–49.

(75) Zhou, J.; Li, S.; Huang, L.; Xiong, H.; Wang, F.; Xu, T.; Xiong, H.; Dou, D. *arXiv preprint arXiv:2012.09624* **2020**.

(76) Wójcikowski, M.; Kukiełka, M.; Stepniewska-Dziubinska, M. M.; Siedlecki, P. *Bioinformatics* **2019**, *35*, 1334–1341.

(77) Feinberg, E. N.; Sur, D.; Wu, Z.; Husic, B. E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V. S. *ACS central science* **2018**, *4*, 1520–1530.

(78) Rifaioglu, A. S.; Cetin Atalay, R.; Cansen Kahraman, D.; Doğan, T.; Martin, M.; Atalay, V. *Bioinformatics* **2021**, *37*, 693–704.

(79) Karimi, M.; Wu, D.; Wang, Z.; Shen, Y. *Bioinformatics* **2019**, *35*, 3329–3338.

(80) Öztürk, H.; Ozkirimli, E.; Özgür, A. *arXiv preprint arXiv:1902.04166* **2019**.

(81) Li, Y.; Yang, J. *Journal of chemical information and modeling* **2017**, *57*, 1007–1012.

(82) Yang, J.; Shen, C.; Huang, N. *Frontiers in pharmacology* **2020**, *11*, 69.

(83) Rognan, D. *Systems Medicine: Integrative, Qualitative and Computational Approaches* **2021**, *2*, ed. by Wolkenhauer, O., 163–173.

(84) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. *Journal of chemical information and modeling* **2020**, *60*, 4200–4215.

(85) Meli, R.; Morris, G. M.; Biggin, P. C. *Frontiers in bioinformatics* **2022**, *2*, 57.

(86) Li, H.; Sze, K.-H.; Lu, G.; Ballester, P. J. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2020**, *10*, e1465.

CHAPTER 2

Binding affinity prediction with GNN models

published in Volkov et al., J. Med.Chem., 2022, 65, 7946-7958

2.1 Introduction

Predicting absolute binding free energies (affinities) from three-dimensional atomic coordinates of protein–ligand complexes remains one of the grand challenges of computational chemistry. [1] For example, drug discovery would immediately benefit from key advances in this topic, by better triaging potentially interesting molecules among virtual screening hits [2, 3] and proposing viable analogs in emerging ultra-large chemical spaces [4] for hit to lead optimization. With the ever increasing amount of high-resolution experimentally determined protein–ligand structures, [5] binding affinity prediction algorithms have switched from physics-based [6] to empirical scoring functions, [7] and in the last years to machine learning [8] and deep learning methods. [9, 10] The latter category of descriptor-based scoring functions has notably led to numerous protein–ligand affinity models [11–38] (see a non-exhaustive list Table 2.1), notably because deep learning does not require explicit descriptor engineering and is ideally suited to find hidden nonlinear relationships between 3D protein–ligand structures and binding affinity. The first deep neural networks (DNNs) to predict binding affinities were convolutional neural networks (CNNs) reading a protein–ligand complex as an ensemble of grid-based voxels with multiple channels corresponding to pharmacophoric properties. [22, 34, 35] The CNN architecture is relatively inefficient from a computational point of view because most of the voxels do not carry any relevant information. Moreover, the search for the best possible hyperparameters is very demanding with respect to memory usage and CPU time. Last, the same object must be presented in multiple orientations in a 3D grid to remove the dependency on the initial atomic coordinates. To overcome these issues and speed-up the training process, most recently developed DNNs read inputs in the form of a molecular graph [39] where nodes are represented by atoms and edges by bonds and/or non-covalent intra and intermolecular interactions. Atoms and edges are embedded with user-defined atomic and/or pharmacophoric properties, enabling all graph components to be updated according to their surroundings all along the network during the training phase.

A gold standard dataset to probe DNN models is the PDBbind database, developed by Wang et al. [40] and updated on a regular basis. [41] In its last version (v.2020), it stores 19443 protein–ligand X-ray structures of known binding affinity ex-

pressed as either inhibition constant (K_i), dissociation constant (K_d) or half-maximum inhibition concentrations (IC_{50}). The general set which encompasses all data is further split in a refined set (5316 entries in the v.2020 release) containing high-quality X-ray structures and the most reliable affinity data (K_i and K_d only), and a core set (290 entries) made of a set of 58 proteins cocrystallized with five different ligands of various affinities. Despite several warnings on the composition [21] and completeness [42] of the PDBbind archive, it remains the largest resource to train machine learning models for structure-based prediction of binding affinities. Many graph neural networks (GNNs), used as end-to-end standalone architecture, [12, 16, 31, 33, 38] in cascade [30] or in combination with CNNs, [26] have been described recently. None of them significantly outperforms first-generation CNNs, most models presenting rather similar accuracies (Pearson correlation coefficient in the 0.80-0.85 range; root-mean square error (RMSE) around 1.2-1.3 pK unit) in predicting affinities for the PDBbind core set (Table 2.1) but significantly lower accuracies for true external test sets. [24, 26, 28]

Despite the strong commitment of data scientists, we believe that drug discovery has not really benefited from the already described models for the major reasons that machine (deep) learning scoring functions still generalize poorly and are not readily applicable to virtual screening of large compound libraries. [25] This major discrepancy does not prevent computer scientists to propose novel deep learning models, almost on a monthly basis, usually focusing on the novelty of the DNN architecture but often omitting to answer three questions: (i) Is the apparent performance biased by either the chosen descriptors, [43, 44] or the protein–ligand training space? [21, 45] (ii) Does the model generalize well to external test sets? (iii) Has the model captured the physics of intermolecular interactions and does it achieve good predictions for meaningful reasons?

A first warning has been raised by several groups noticing that CNNs trained on voxelized protein–ligand complexes or graphs do not really learn the physics of protein–ligand recognition because ligand-only or protein-only models exhibit performances quite similar to those reached by protein–ligand reading models. [21, 28, 33, 44, 46] Comparison of the performance of 24 recently-published DNNs [11–38] reveals that the model accuracy is independent of the size of the training set (e.g. PDBbind general

vs. refined set; Table 2.3), contradicting the general idea that more high-quality input protein–ligand structures are required to generate better models. Data augmentation strategies consisting of adding high-quality docking poses to PDBbind X-ray structures also lead to contradictory results. [14, 20, 26, 31] Although very few attempts to predict a true thermodynamic cycle, considering proteins and ligands in their free and bound states have been reported; [21, 22, 28] it remains counter-intuitive that the best models are not obtained with architectures explicitly taking into account the three bound/unbound species. Moreover, there is no relationship between the complexity of protein (sequence vs. structure) and ligand (SMILES strings vs. 2D graphs vs. 3D structures) descriptors and the accuracy of the resulting DNN models. [39, 47, 48] Simple models even omitting to consider the protein–ligand bound state are equally good at predicting binding affinities. [24, 42, 47, 48] It is therefore tempting to speculate that DNNs just memorize hidden patterns in either the ligand or protein spaces on which the models have been trained. As a consequence, modifications of protocols used to split input data into training, validation and test sets have a major impact on the accuracy and applicability domain of obtained models. [21, 22]

Since the publicly available training set is limited to the world of PDBbind protein–ligand complexes, there is a need for better identifying still hidden biases in the PDBbind archive, as well as to remove probable redundancies in the choice of descriptors. In the current study, we present a critical evaluation of a modular message passing graph neural network architecture to predict binding affinities from three independent graphs describing proteins, ligands and their complexes. The modularity of the DNN architecture enables depicting the true contribution of each state (free vs bound) of the two partners and to clearly evidence serious biases in both the ligand and protein composition of the PDBbind space. The current study suggests that descriptors focusing on non-covalent interactions with no additional ligand/protein information are the most suited to unbiased learning.

Table 2.1. Structure-based deep neural networks to predict protein-ligand binding affinities.

Model	Type	Objects	Descriptor	Training set	Test set	Split	R*p*	RMSE	Reference
TNet-BP	CNN	PL	Topological fingerprints	PDBbind 2016 refined (n=3767)	PDBbind 2016 core (n=290)	PDBBind original	0.826	1.37	11
ACNN	CNN	P, L, PL	Atom type-labelled distances (Nat*25 atom types*12 closest neighbors)	PDBbind 2015 refined (n=2965)	PDBbind 2015 refined (n=741)	Temporal	0.727	-	12
Brendan	CNN	PL	3D grid (21*21*21 Å) * 256 bit SPLIF vector	PDBbind 2016 general (10000)	PDBbind 2016 general (1500)	Random	0.704	-	13
PotentialNet	GNN	P, L, PL	Protein-ligand graph	PDBbind 2007 refined (n=1095)	PDBbind 2007 core (n=195)	PDBBind original	0.822	1.39	14
K_{DEEP}	CNN	L, PL	1-Å 3D grid (25*25*25 Å) * 16 features	PDBbind 2016 refined (n=3767)	PDBbind 2016 core (n=290)	PDBbind original	0.820	1.27	15
Pafnucy	CNN	L, PL	1-Å 3D grid (21*21*21 Å) * 19 features	PDBbind 2016 general (11906)	PDBbind 2016 core (n=290)	PDBBind original	0.780	1.42	16
DeepATom	CNN	PL	1-Å 3D grid (25*25*25 Å) * 24 features	PDBbind 2016 refined (n=3390)	PDBbind 2016 core (n=290)	PDBbind original	0.807	1.32	18
DeepBindRG	CNN	PL	ligand (84) * protein (41) atom pair distances < 4 Å	PDBbind 2018 general (n=13500)	PDBbind 2018 general (n=925)	Random	0.593	1.50	21
OnionNet	CNN	PL	Atom type-labelled distances (Nat*25 atom types*12 closest neighbors)	PDBbind 2016 general (n=11906)	PDBbind 2016 core (n=290)	PDBbind original	0.816	1.28	22
RosENet	CNN	PL	voxelized Rosetta interaction energies + pharmacophoric descriptors	PDBbind 2016/2018 refined (n=4463)	PDBbind 2016 core (n=290)	PDBBind original	0.820	1.24	23
graphDelta	GNN	L, PL	One-hot encoded ligand atoms + protein environmental descriptors (373)	PDBbind 2018 general (n=8766)	PDBbind 2016 core (n=285)	PDBbind original	0.870	1.05	24
AK-Score	CNN	PL	id Kdeep	PDBbind 2016 refined (n=3772)	PDBbind 2016 core (n=285)	PDBBind original	0.827	1.22	25
SE-OnionNet	CNN	PL	1-Å grid (21*21*21)* 64 protein-ligand element distance counts	PDBbind 2018 general (n=11663)	PDBbind 2018 refined (n=463)	Random	0.853	1.59	27
Progessive multitask network	CNN	P, L, PL	ligand ECFP + Protein ECFP + Protein-Ligand SPLIF	PDBbind 2016 refined (n=3568)	PDBbind 2016 core (n=290)	PDBbind original	0.740	0.98	28
ACNN	CNN	P, L, PL	Atom type-labelled distances (Nat*25 atom types*12 closest neighbors)	PDBbind 2015 refined (n=3706)	PDBbind 2015 core (n=195)	PDBBind original	0.730	-	29
Pair	CNN	PL	protein-ligand distance pairs	PDBBbind 2018 refined (n=2675)	PDBBbind 2018 refined (n=891)	Random split	0.660	1.61	30
DEELIG	CNN	L, PL	Atomic model: 3D grid (10*10*10 Å) * 19 bits (atomic model); Composite model: 3D grid (10*10*10 Å) * 44 bits (pocket) + 14716 bits (ligand)	in-house set (n=4041)	PDBbind 2016 core (n=290)	Random 80/10/10	0.889	-	31
Interaction GraphNet	GNN	P, L, PL	independent GNN for intra and inter-molecular interactions	PDBbind 2016 general (n=10366)	PDBbind 2016 core (n=290)	PDBBind original	0.837	1.22	32
midlevel fusion	CNN+ GNN	PL	CNN: 1-Å grid (48*48*48)* 19 atomic features; GNN: covalent (d < 1.5 Å) and non-covalent edges (1.5 < d < 4.5 Å)	Pdbbind 2016 general+refined (13283)	PDB2016 core set (n=290)	PDBBind original	0.810	1.31	33
SMPLIP	RF+ CNN	L, PL	IFP (140) + interaction distances (140) + SMF descriptors (2282)	Pdbbind 2016 general+refined (13283)	PDB2016 core set (n=290)	PDBBind original	0.770	1.51	34
OctSurf	CNN	PL	1-Å 3D grid (64*64*64 Å) * 24 features/octant	PDBbind 2018 general (n=16126)	PDBbind 2016 core (n=285)	PDBBind original	0.793	1.45	35
BAPA	CNN	PL	Protein-ligand interaction descriptors + 6 Vina terms	PDBbind 2016 refined (n=3689)	PDBbind 2016 core (n=285)	PDBbind original	0.819	1.31	36
APMNet	GNN+ GNN	P, L	75 DeepChem atomic features	PDBbind 2016 general (n=11844)	PDBbind 2016 core (n=290)	PDBBind original	0.815	1.27	37
GraphBAR	GNN	PL	13 features * 200 protein-ligand atoms	PDBbind 2016 general (n=11146)	PDBbind 2016 core (n=290)	PDBbind original	0.764	1.44	38

2.2 Results and discussion

Describing ligands, proteins and protein–ligand complexes as graphs

Ligand graphs were generated from PDBbind mol2 input files, defining atoms as nodes and bonds as edges. Each node was annotated by the corresponding atom element, whereas each edge was annotated by the corresponding bond length (Figure 2.2 A).

Protein graphs were described from ligand-binding sites, defined as any amino acid, ion or water molecule for which one heavy atom is less than 4 Å away from any ligand heavy atom (Figure 2.2 B). In the protein graph, nodes correspond to protein pseudoatoms (PPA), as previously defined by Schmitt et al., [49] and placed at key main chain/side chain positions and annotated by the molecular interaction properties of the corresponding residue (Figure 2.1). A total of six properties were used to annotate protein nodes with the following labels and interaction properties: CA, aliphatic (hydrophobic interactions); O, hydrogen-bond acceptor (hydrogen bond); CZ, aromatic (π-π interaction); OG, hydrogen-bond acceptor and donor (hydrogen bond); N, hydrogen-bond donor (hydrogen bond); ZN, metal (metal chelation). To avoid keeping protein residues whose side chains are pointing outward the ligand-binding cavity, a residue-based filtering was performed based on the angle between the ligand center of mass, the residue c-alpha atom and all residue-specific PPAs. PPAs of amino acid side chains, for which the corresponding angle was higher than 90°were removed from the binding site definition. Finally, edges were added between final protein nodes distant by less than 4.0 Å and further annotated according to the distance between the corresponding PPAs.

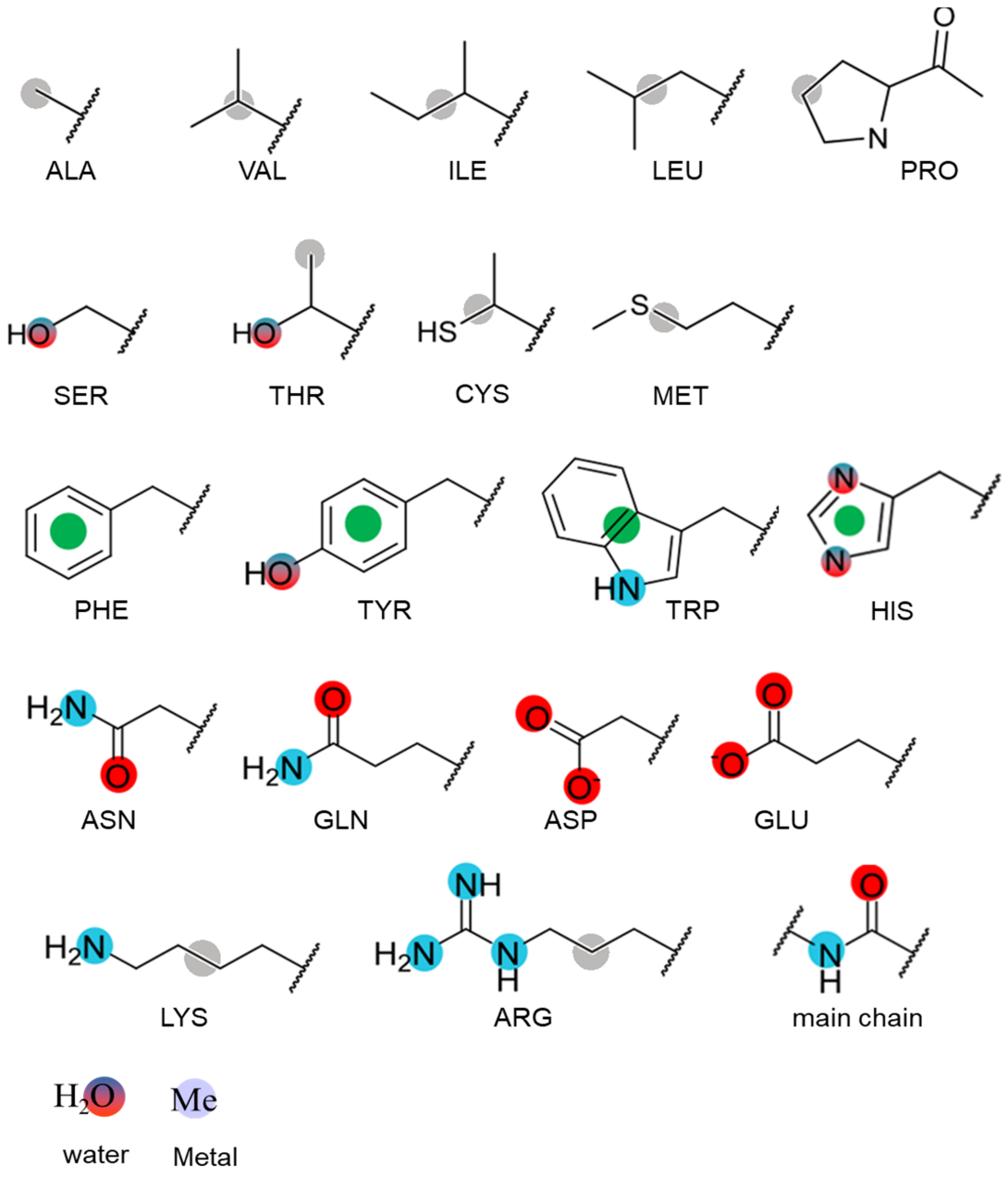


Figure 2.1 Location and pharmacophoric properties of protein pseudoatoms (grey, aliphatic; red, hydrogen-bond acceptor; cyan, hydrogen-bond donor; green, aromatic; metal-chelating, steel blue)

Table 2.2. Geometric rules to define protein-ligand non-covalent interactions.

Interaction	Rule 1[a]	Rule 2[b]
H-bond	$\|\|\overrightarrow{DA}\|\| \leq 3.5$Å	$< \overrightarrow{DH}, \overrightarrow{HA} > \in [\frac{-\pi}{4}, \frac{\pi}{4}]$
Ionic	$\|\|\overrightarrow{+-}\|\| \leq 4.0$Å	
Hydrophobe	$\|\|\overrightarrow{Y_1Y_2}\|\| \leq 4.5$Å	
Aromatic (Face to face)	$\|\|\overrightarrow{ac_1ac_2}\|\| \leq 4.0$Å	$< \overrightarrow{n_1}, \overrightarrow{n_2} > \in [\frac{-\pi}{6}, \frac{\pi}{6}]$
Aromatic (Edge to face)	$\|\|\overrightarrow{ac_1ac_2}\|\| \leq 4.0$Å	$< \overrightarrow{n_1}, \overrightarrow{n_2} > \in [\frac{\pi}{6}, \frac{5\pi}{6}]$
π-cation	$\|\|\overrightarrow{ac-}\|\| \leq 4.0$Å	$< \overrightarrow{n}, \overrightarrow{ac+} > \in [\frac{-\pi}{6}, \frac{\pi}{6}]$
Metal	$\|\|\overrightarrow{MA}\|\| \leq 2.8$Å	

[a] D: H-bond donor; A: H-bond acceptor; +: cation; -:anion; Y: hydrophobe; ac: geometric center of an aromatic ring; M: metal.

[b] H: hydrogen; n: normal to the aromatic ring.

Noncovalent interactions (hydrophobic, aromatic, hydrogen bonds, ionic bonds, metal chelation; see details in Table 2.2) between protein and ligands were computed on the fly with the GRIM routine of the IChem v5.2.9 package. [50]

For each interaction, IPAs are placed at the two atoms of the interacting pair (Figure 2.2 C). The resulting representation was converted to a graph where nodes represent either protein or ligand-interacting atoms. Edges between nodes were added in two consecutive steps. First, the principal edges were added between interacting IPAs. Then, secondary edges were added between noninteracting IPAs under the conditions that the corresponding IPAs originate from the same molecule (protein or ligand) and that their distance is less than 4 Å. Each node was annotated by one of the following labels, according to the nature of the corresponding noncovalent interaction: CA, hydrophobic; NZ; ionic (the interacting protein atom is positively charged); N, hydrogen-bond (the interacting protein atom is a donor); OG, hydrogen-bond (the interacting protein atom is both an acceptor and donor); O, hydrogen-bond (the interacting protein atom is an acceptor); CZ, aromatic; OD1, ionic (the interacting protein atom is negatively charged); ZN: metal coordination. An additional binary label was added to nodes to account for their belonging to either the protein or the ligand. The only

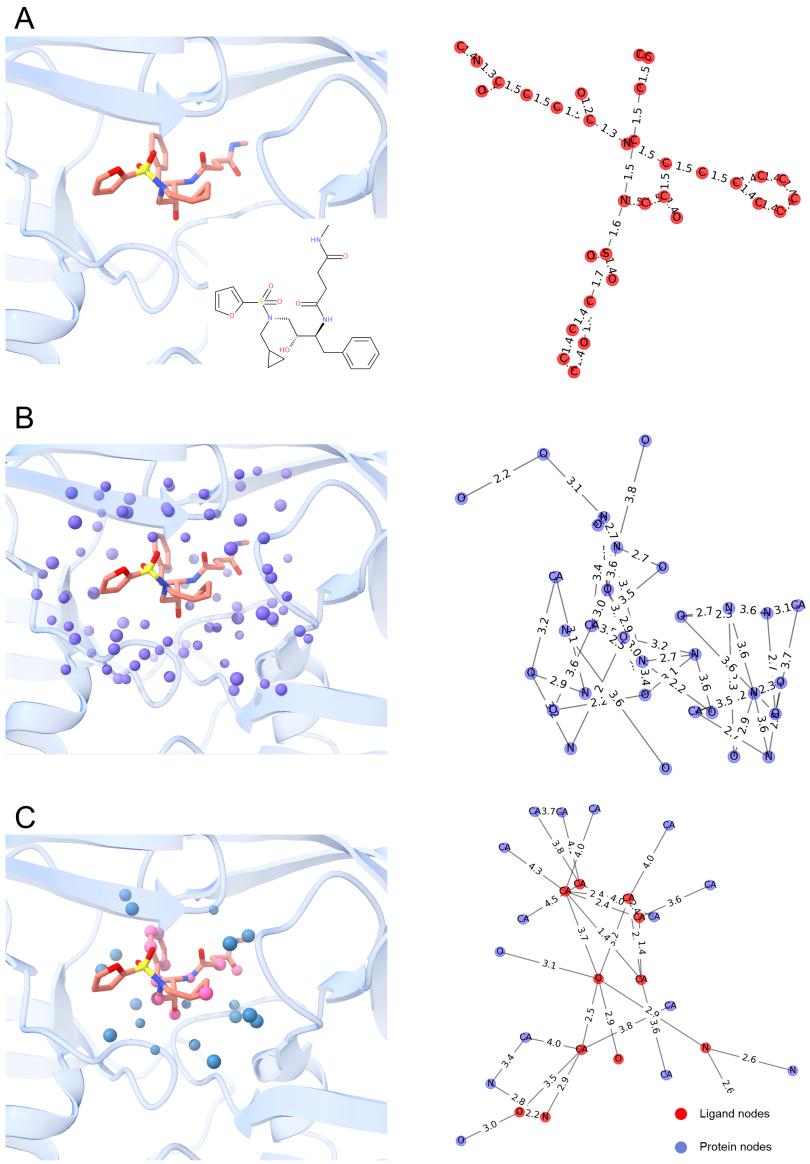


Figure 2.2 Encoding protein, ligand and protein–ligand structures (PDB ID 2PSV) in graphs. A) Nodes are set at ligand atomic coordinates, and labelled by atomic element. Edges represent bonds, annotated by bond length. B) Proteins are represented by ligand-binding site pseudoatoms (slate blue spheres) placed at amino acid-specific positions. Nodes are set at protein pseudoatom coordinates and annotated by pharmacophoric properties. Edges link two nodes distant by less than 4.0 Å. C) protein–ligand interactions are represented by interaction pseudoatoms (pink and blue spheres) set at protein and ligand-interacting atoms. Edges are placed between two nodes (protein, blue; ligand, red) in direct interaction, or between protein or ligand notes if distant by less than 4.0 Å. Each edge is annotated by the distance between the corresponding nodes.

edge feature is the distance between pseudoatoms corresponding to the graph nodes (edge length). Therefore, the information on the spatial structure of the binding site was partially preserved, while the representation remained invariant to binding site rotations and node numbering.

DNN architecture

We used a graph CNN architecture that belongs to the family of message passing neural networks (MPNNs), recently shown to exhibit excellent performance in predicting quantum chemical properties. [51] The MPNN is here applied to an undirected graph G with node features x_v and edge features e_{vw}. In an MPNN, each node v in the graph has a hidden state h_v^t (feature vector). For each node v, a function of hidden states and edges of all neighboring nodes is aggregated. The hidden state of the node V_t is then updated with the obtained message m_v^{t+1} and its previous hidden state. Three main equations characterize the MPNN on graphs. First, the message m_v^{t+1} obtained from all neighboring nodes $N(v)$ is given by equation 2.1:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \tag{2.1}$$

where M_t is the aggregation function applied at step t, h_v^t the hidden state of node v, h_t^w the hidden state of the neighboring node w, e_v^w is the feature of edge between v and w. The hidden state h_v^{t+1} of the node v is then updated according to equation 2.2:

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{2.2}$$

where U_t, the update function, is another neural network used to update the hidden state by taking into account both the sum of all previous messages and the previous hidden state. The message passing algorithm is repeated a user-defined number of times until the readout phase generates a final feature vector $\hat{y}$ describing the entire graph G according to equation 2.3:

$$\hat{y} = R(h_v^T | v \in G) \tag{2.3}$$

where R is the readout function, T is the number time steps.

The message functions M_v, node update function U_v and readout function R are all learned differentiable functions. The complete architecture of the graph convolutional network (Figure 2.3 A) includes an MPNN module with a customizable hidden size and a two-layer dense module with a top layer size of hidden size / 4. The

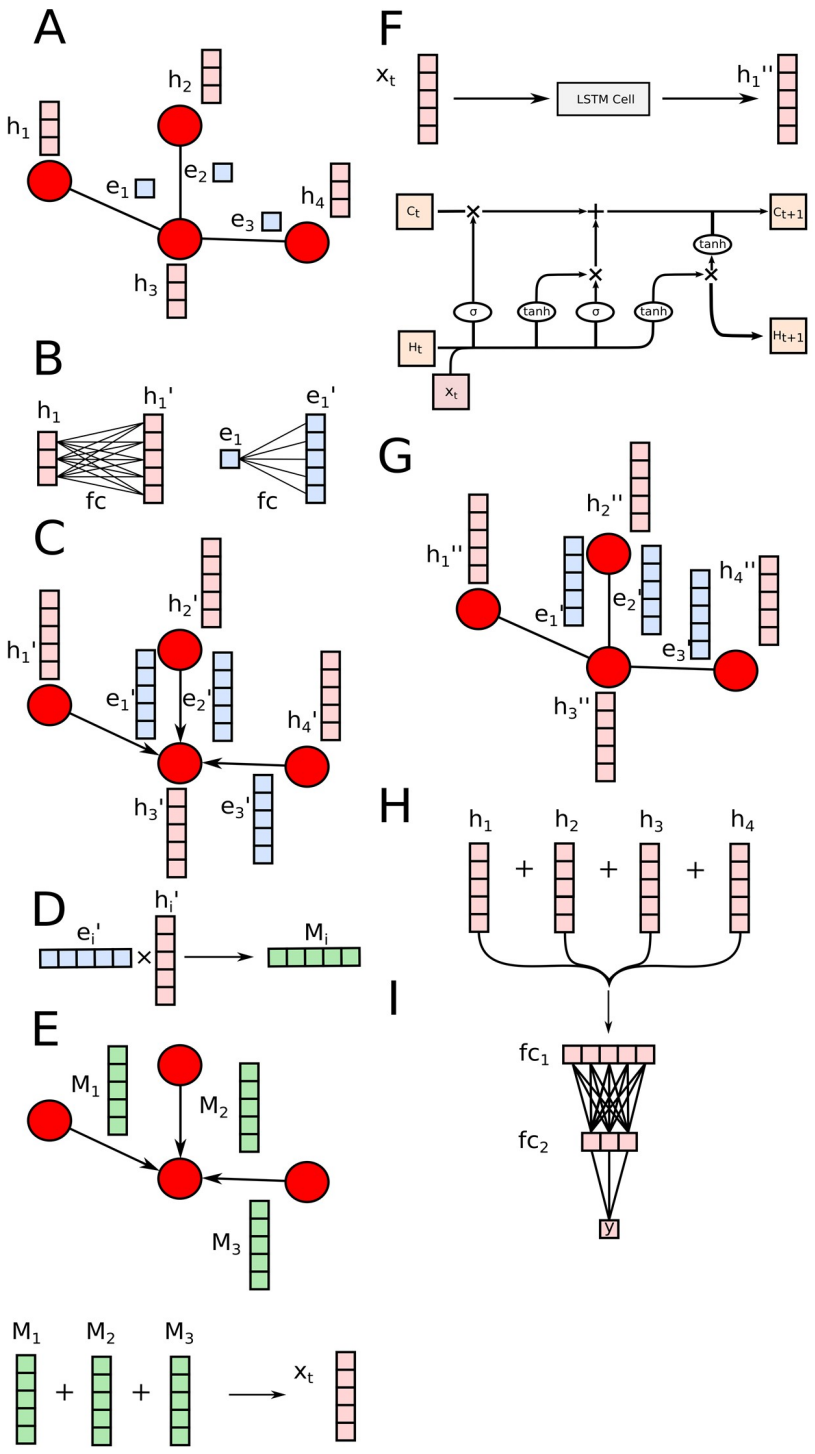


Figure 2.3 General architecture of a MPNN with two message passing steps. A) Initial graph with node and edge labels. B) Transformation of node and edge feature vectors with fully connected layers (fc) C) Application of linear layers to node and edge feature vectors. D) Message generation. E) Message passing. F) Node features update using a standard LSTM cell architecture. G) Graph with updated node features. H) Readout. I) Fully connected (fc) layers.

invariance of the MPNN readout function to node and edge re-enumeration enables applying MPNNs to a merged input consisting of multiple disconnected graphs describing protein, ligand, and protein–ligand interactions without modifying the network architecture. In the current study, MPNN models have been derived from graphs describing the two molecular species (protein and ligand) in both their liganded and unliganded states, thereby enabling to evaluate the exact contribution of each state. To ascertain the fairest possible comparison, all models were trained on the same training/validation set using exactly the same input graphs.

DNN models are heavily biased by ligand and protein features

Starting from three possible input graphs describing the protein, the ligand and their noncovalent interactions, seven combinations (one graph, two graphs, three graphs) were first tested as baselines with two objectives: (i) benchmark the performance of MPNN in predicting binding affinities with respect to other DNN architectures, [11–24, 26–38] (ii) analyze the contribution of each input graph and assess their potential synergistic use (Table 2.3).

Despite our customized protocol to process PDBbind entries, we were able to reproduce the performance of the native Pafnucy model, [35] estimated by the Pearson's correlation coefficient R_p in predicting experimentally derived affinities for samples of the PDBbind 2016 core set ($R_p = 0.777$; Table 2.3). Our seven MPNN models exhibit various performances with Rp values ranging from 0.687 to 0.813. Intuitively, one would have expected that a model trained on protein–ligand interactions (I model) achieves better performance than models trained solely on either the ligands (L model) or the proteins (P model). However, the P and L models exhibit a better performance than the I model (Table 2.3). Out of the one-component models, the ligand-based model is clearly the one leading to the best results ($R_p = 0.749$, RMSE = 1.567). Combining two graph inputs increases the accuracy of the corresponding predictions, with a clear advantage to the PL model ($R_p = 0.812$, RMSE = 1.553) omitting protein–ligand interaction features. The most sophisticated model, taking into account the three graph inputs (PLI model), does not provide any clear advantage compared to the PL model, suggesting that explicitly defined molecular interactions are not required to predict binding affinities of the core set sample. Applying the models to a much

larger (n=3386) and more difficult hold-out set, obtained by temporal splitting of the PDBbind dataset (hold-out 2019 set) illustrates a moderate generalization capacity, with the R_p value decreasing by ca. 0.15 unit for all models (Table 2.3).

Table 2.3. Performance of modular MPNN models in predicting affinities for the external 2016 core set and the 2019 hold-out set.

model[a]	2016 core set		2019 hold-out set	
	R_p	RMSE	R_p	RMSE[b]
P	0.725	1.569	0.570	1.528
L	0.749	1.567	0.611	1.455
I	0.687	1.605	0.538	1.563
PL	0.812	1.553	0.645	1.512
PI	0.777	1.462	0.613	1.485
LI	0.780	1.477	0.630	1.425
PLI	0.813	1.511	0.652	1.481
Pafnucy[c]	0.773	1.429	0.456[d]	1.642

a) P: protein graph, L: ligand graph, I: interaction graph; PL: merged protein and ligand graphs, PI: merged protein and interaction graph; LI: merged ligand and interaction graph; PLI: merged protein, ligand and interaction graph. b) Root-mean square error, in pK unit. c) In-house Pafnucy prediction (R_p = 0.78 in the original paper). [35] d) Predictions failed for 29 entries.

From a pure statistical point of view, the performance of four out of the seven MPNN models is superior to that achieved with the CNN Pafnucy model, when applied to the 2016 external core set (Table 2.3). Extending predictions to the challenging 2019 hold-out set suggests that all models outperform Pafnucy. Assuming that a Pearson R_p threshold value of 0.600 is commonly used in pharmaceutical industrial settings to qualify a good predictive QSAR model, [52] five out of the seven MPNN models could be considered as satisfactory. However, these models remain enigmatic from a physicochemical point of view since ligand-only and protein-only models still outperform the interaction model. Moreover, the impact on model predictive performance of the explicit consideration of protein–ligand interactions in the two or three-component

models remains very limited (Table 2.1). Noteworthy, focusing the analysis on three target classes for which enough samples are present in the hold-out set (GPCRs, 47 samples; kinases, 572 samples; nuclear receptors, 106 samples) did not change the above observations (Figure 2.4).

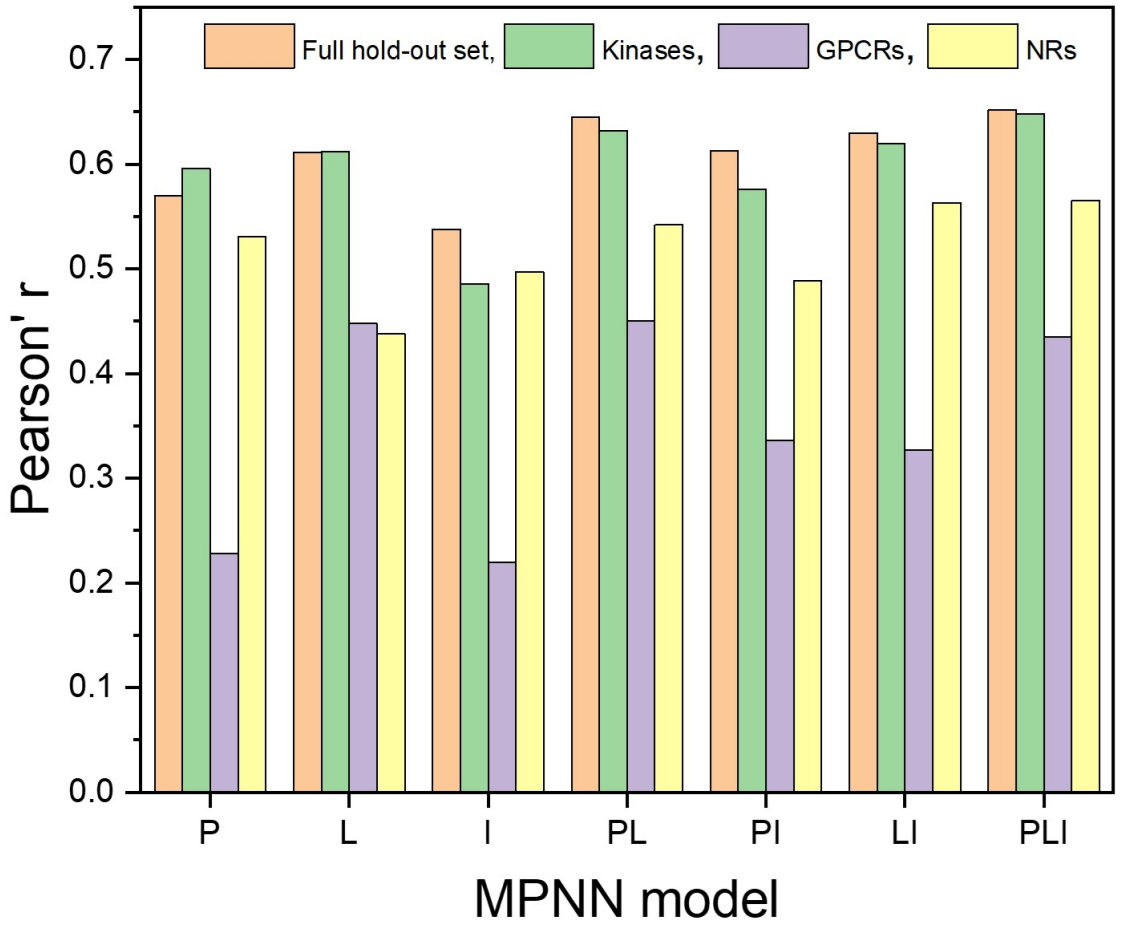


Figure 2.4 Performance of modular MPNN models in predicting affinities for specific target classes of the 2019 hold-out set. Mapping of protein target classes (GPCRs, G-protein-coupled receptor; NRs, Nuclear hormone receptors) from the Pharos database (https://pharos.nih.gov/) to PDB entries was performed using the Pharos-to-PDB code (https://github.com/ravila4/Pharos-to-PDB).

Several conclusions can be drawn from these results. First, the herein implemented MPNN architecture provides a lower accuracy to previously reported CNN and GNN models, when just protein–ligand interactions are taken as input. Pafnucy, used here as a state-of-the-art CNN achieves a better accuracy than the MPNN I

model (Table 2.3). Second, protein–ligand binding affinities of the 2016 core set can apparently be predicted from sole protein or ligand structures. Third, the explicit description of protein–ligand interactions does not provide any clear advantage compared to the corresponding interaction-agnostic models (e.g. compare P to PI, L to LI, and PL to PLI models, Table 2.3). Fourth, all models exhibit a decreased accuracy when applied to a hold-out set of newly described complexes, suggesting a probable overtraining. Most of these observations are counter-intuitive and cannot been rationally explained by first-principle physics. They evidence, to our viewpoint, potential biases in the composition of the PDBbind training/test sets suggesting that the derived models have partly memorized input data but not learned the physics of protein–ligand non-covalent interactions. This phenomenon has already been described for many ligand-based machine learning models and frequently happens when training and test sets exhibit significant redundancies. [53] Another alert, that we already mentioned for both machine learning and DNNs, [43, 54] is their propensity to predict binding affinities with apparently satisfactory performance metrics (R_p, RMSE), but where the predicted values are in fact contained within a very tiny range centered on the mean value of training samples. This tendency is again observed for the current predictions of all MPNN models, whatever the chosen input graph(s) and external test set (Figure 2.5).

Whereas experimental affinities of the two external test sets are spread over 10 pk units, MPNN and Pafnucy predictions are restricted to ca. 6 pk units. Considering only the 25th and 75th percentiles of the distributions (boxes in Figure 2.5), 50 % of the predicted data are centered on a mean value ±1.5 pK unit, Pafnucy predictions lying even in a narrower range for 2019 hold-out set predictions (Figure 2.5). The prediction error is statistically minored if the output value is close to the mean of trained samples. This may be a reason why machine learning models tend to yield narrow distribution of predicted values. This phenomenon might be even amplified in machine learning models for which the loss function aims at minimizing the root-mean-square error. Altogether, we suspect significant biases in the ligand and protein composition of the PDBbind archive which, to our viewpoint, should prevent the blind usage of DNN models in prospective applications.

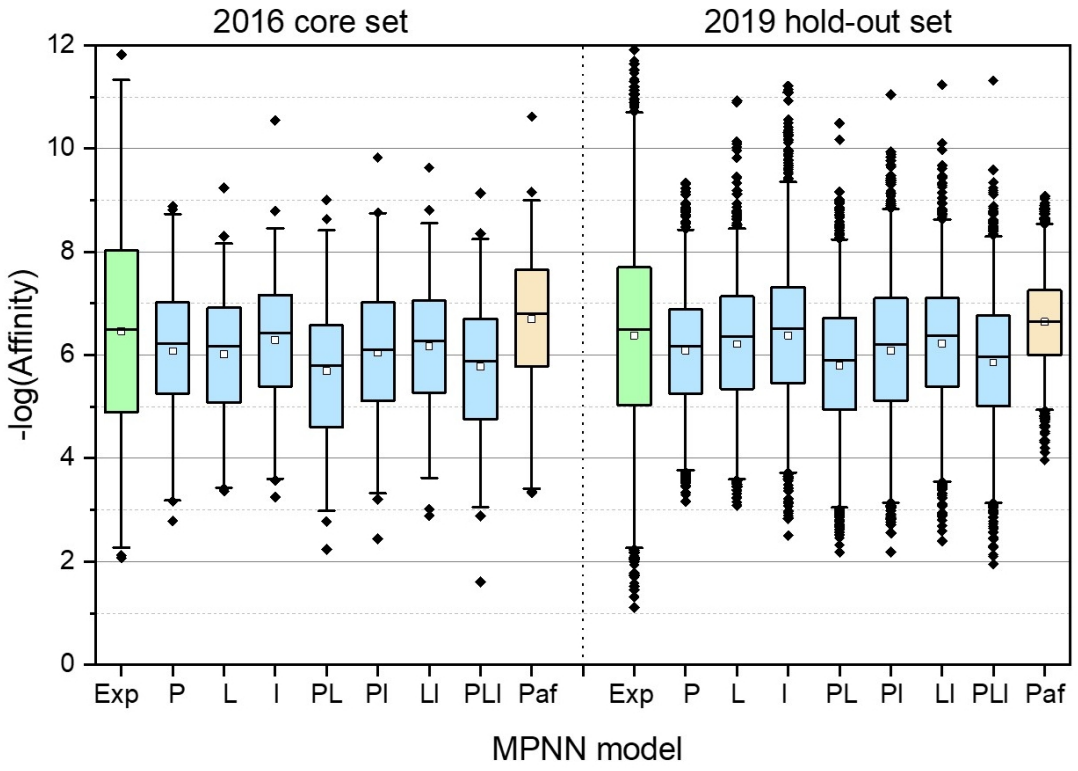


Figure 2.5 Distribution of experimental and predicted affinities for the 2016 core set (n=257) and the 2019 hold-out set (n=3386). Exp: experimental affinity; P, L, I, PL, PI, LI, PLI: predicted by MPNN models using protein(P), ligand(L) and protein–ligand interaction (I) graphs used alone or in combinations; Paf: predicted by the Pafnucy model. The boxes delimit the 25th and 75th percentiles, and the whiskers delimit the 1st and 99th percentiles. The median and mean values are indicated by a horizontal line and a filled square in the box, respectively. Outliers are indicated by a diamond.

Simple memorization models suggest that ligand and protein neighborhoods contribute massively to MPNN predictions

To estimate the relative contribution of simple memorization vs true learning when applying MPNNs to predict affinities for PDBbind samples, we generated simple memorization baseline models in which the predicted affinity of a test sample was just inferred by ligand or protein similarity to the five closest training samples (Table 2.4). Of course, such memorization models are meaningless and just define baselines to quantify the amount of biases in the training data set.

Table 2.4. Performance of simple memorizing models in predicting affinities for the external 2016 core set and the 2019 hold-out set.

model	2016 core set		2019 hold-out set	
	R_p	RMSE	R_p	RMSE
PLI MPNN[a]	0.813	1.511	0.652	1.481
ligand similarity[b]	0.663	1.624	0.509	1.641
protein similarity[c]	0.547	1.765	0.310	1.794

a) Three-component (protein, ligand, and protein–ligand interactions) MPNN model of Table 2.3. b) Prediction is equal to the average affinity of the five training samples with the most similar ligands, similarity being expressed by a Tanimoto coefficient on ECFP4 circular fingerprints (see the Experimental Section). c) Prediction is equal to the average affinity of the five training samples with the most similar proteins, similarity being expressed by an Euclidean distance on protein cavity fingerprints (see the Experimental Section).

Given its simplicity, the ligand memorization model performs remarkably well on the two external test sets (Table 2.4) and is almost equivalent in accuracy to the protein–ligand interaction MPNN model (I model, Table 2.3). The protein similarity model exhibits a decreased but still noticeable performance. The observed dependency was relatively insensitive to the number of closest training samples (ligands, proteins) used to infer average affinity values for prediction (Figure 2.6). We can therefore conclude that simple memorization probably accounts for a large part of the excellent performance of the MPNN model using ligand, protein and protein–ligand interaction graphs as input (Table 2.4).

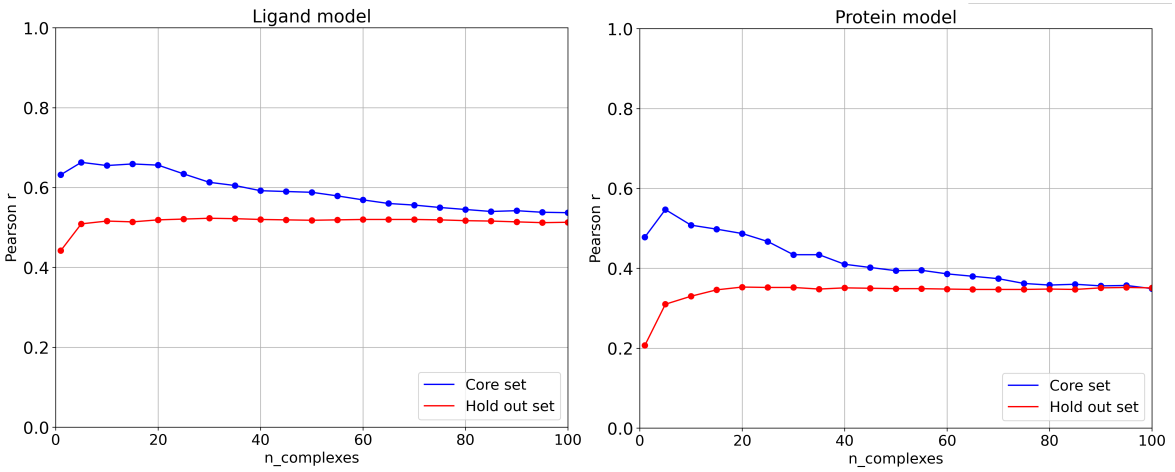


Figure 2.6 Influence of the number of closest ligands or proteins used to average binding affinities in the performance of simple memorization models, assessed by the Pearson r correlation coefficient.

Undersampling the training set does not remove ligand and protein biases

The goal of this procedure was to reduce the bias originating from the sampling of proteins and ligands present in the PDBbind data set. Thus, we undersampled the PDBbind training set by removing progressively the protein–ligand pairs which are easily predictable if we rely solely on protein or ligand graphs, while ignoring the interaction graphs. Intuitively, those are probably the most biased datapoints. As a first approach to remove potential ligand and protein biases in the training set, we filtered out all training samples whose affinities were easily predicted by ligand-only or protein-only five-fold cross-validation MPNN models. The protocol was repeated for batches of 50 samples to get a good tradeoff between speed and precision of the unbiasing algorithm.

Undersampling reduced the size of the training set from 9662 to 4635 samples, but marginally affected the accuracy of all MPNN models, whatever the graphs used as inputs (Figure 2.7). Interestingly, decreasing the size of the training set by 50 % did not alter the quality of the predictions for both external sets. However, the same obvious biases (good performance of ligand-only and protein-only models, no benefit of explicitly considering protein–ligand interactions) were found again, suggesting that the hidden biases reported above are still present in the undersampled training set.

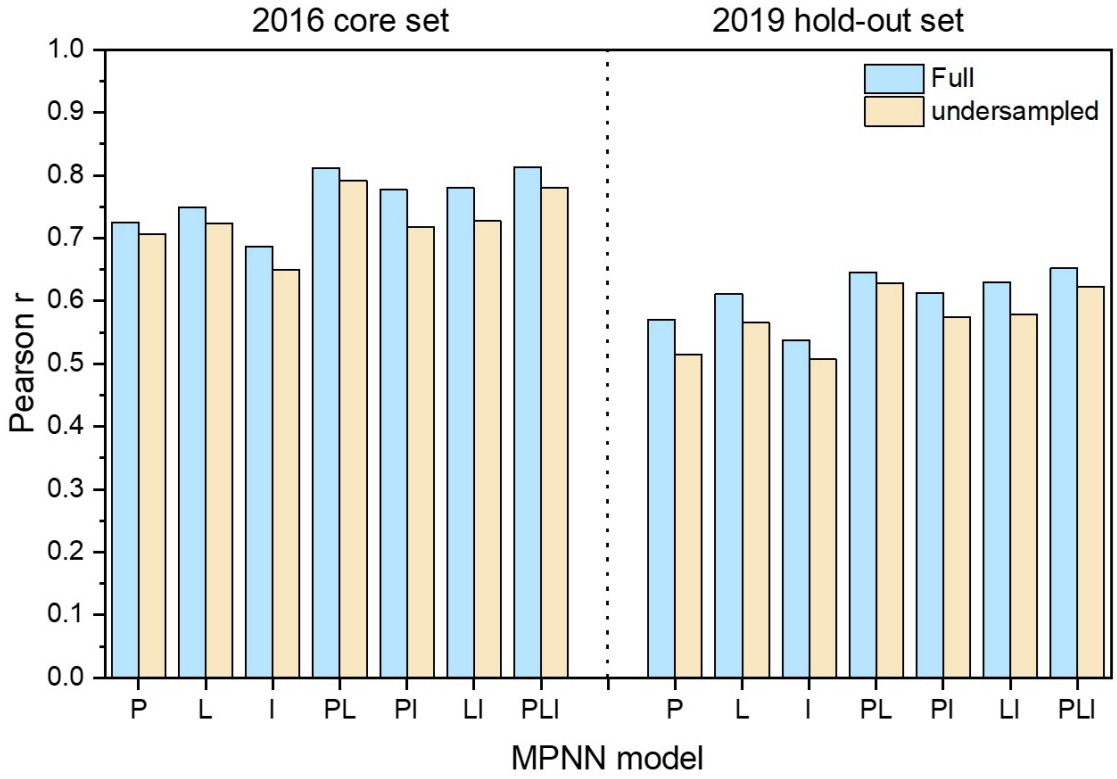


Figure 2.7 Effect of undersampling the PDBbind training set on the on the scoring power of MPNN models in predicting binding affinities for the 2016 core set and the 2019 hold-out set. Default models were trained on the full set (9653 entries) whereas undersampled models were trained only on 4658 samples. P: protein graph model, L: ligand graph model, I: interaction graph model; PL: merged protein and ligand graphs model, PI: merged protein and interaction graphs model; LI: merged ligand and interaction graphs model; PLI: merged protein, ligand and interaction graphs model.

Influence of ligand buriedness

In a second approach, we looked whether the buriedness of the protein-bound ligands in the training and external sets may be a source of potential biases. Indeed, a fully buried ligand would generate quite complementary protein and ligand graphs that implicitly encode all possible non-covalent protein–ligand interactions. In such cases, it might be conceivable to predict albeit with a moderate accuracy the binding affinity of the corresponding complex from sole ligand or protein graphs.

Computing the buried surface area of all PDBbind ligands in their bound state shows a similar distribution for the three sets (training, 2016 core set, 2019 hold-out set) centered on a mean value close to 60-65 % (Figure 2.8 A).

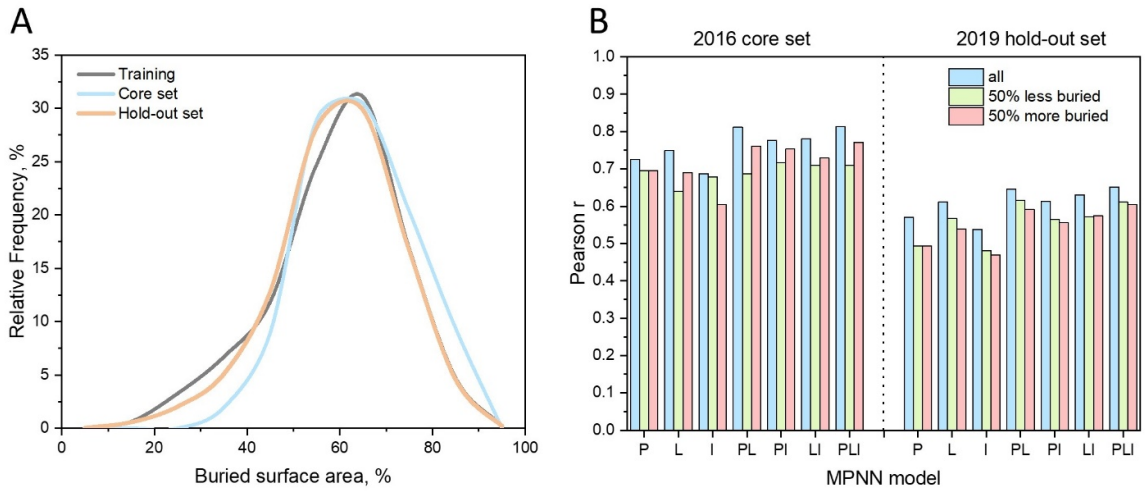


Figure 2.8 Effect of ligand buriedness on MPNN predictions. (A) Distribution of the buried surface area of protein-bound PDBbind ligands. (B) Influence of the protein-bound ligand buriedness on the scoring power of MPNN models in predicting binding affinities for the core set and the 2019 hold-out set. P: protein graph model, L: ligand graph model, I: interaction graph model ; PL: merged protein and ligand graphs model, PI: merged protein and interaction graphs model; LI: merged ligand and interaction graphs model; PLI: merged protein, ligand and interaction graphs model.

We then trained novel MPNN models on two subsets of the PDBbind training set defined by ligand buriedness. The first subset contained the samples with the 50% less buried ligands, whereas the second subset encompassed complexes with the 50 % more buried ligands. Using these new MPNN models to predict the binding affinities

of samples from the two external sets gave disappointing results (Figure 2.8 B). First, all new models were less accurate than the former models trained on the full training set. Second, neither the ligand nor the protein dependency was removed in the new models because novel ligand-only (I models) and protein-only models (P models) were still able to predict binding affinities of both external test samples (Figure 2.8 B). We can therefore safely conclude that ligand buriedness is not the cause of protein and ligand biases in the PDBbind data set.

Complexity of the protein–ligand interaction descriptors

As a third approach, we made the hypothesis that the importance of protein and ligand descriptors with respect to the interaction descriptors may originate from the different complexity level of the input graphs. Indeed, interaction graphs computed in IChem are far simpler than the cognate protein and ligand graphs, when considering the number of nodes, edges and the graph density. By default, protein–ligand interactions have been computed using strict geometrical rules (distances and angles), [55] notably interaction-specific upper distance thresholds (hydrogen bond: 3.5 Å, aromatic π-π interactions: 4.0 Å, ionic bonds: 4.0 Å, hydrophobic interactions: 4.5 Å), leading to relative simple graphs with respect to the number of nodes and edges (Figure 2.9). To increase the importance of protein–ligand interactions in our MPNN models, we therefore increased the complexity of interaction graphs by registering non-covalent interactions up to of 6.0 Å. The new interaction graphs ("int6" label) contain much more nodes and edges, are definitely denser, and are now comparable with protein and ligand graphs (Figure 2.9).

Using the new interaction graphs as input to MPNN models increased significantly the scoring power of the interaction-only I model for the two external test sets (core set, $R_p = 0.728$; hold-out set, $R_p = 0.607$; Figure 2.10). Interestingly, this modification did not increase the accuracy of two-component and three-component models (Figure 2.10). Given the marginal benefit of combining the new interaction graph with either protein and/or ligand graphs, using the single new interaction graph definition appears as the best possible compromise between prediction accuracy, model applicability, and lower risk of memorization effects.

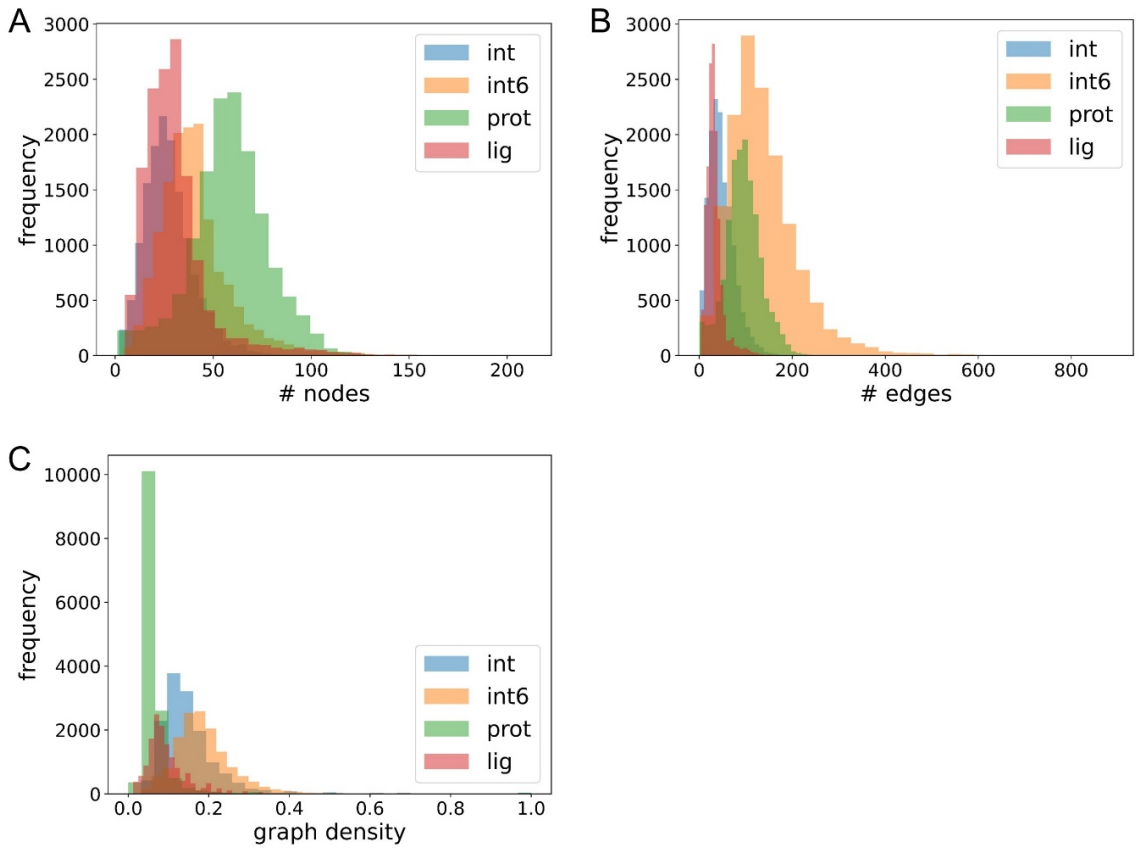


Figure 2.9 Distribution of the number of nodes (A), number of edges (B) and density (C) for interaction (int), protein (prot) and ligand graphs derived from PDBbind protein–ligand complexes (n=14 215). The graph density is defined as $density = \frac{N_{edges}}{N_{nodes}(N_{nodes}-1)}$ where N_{edges} is the number of edges and N_{nodes} is the number of nodes. By default, protein–ligand interactions are computed using interaction-specific upper distance thresholds (hydrogen bond: 3.5 Å, aromatic π-π interactions: 4.0 Å, ionic bonds: 4.0 Å, hydrophobic interactions: 4.5 Å). In the extended mode (int6), a larger distance cut-off of 6.0 Å is applied to all non-covalent interactions.

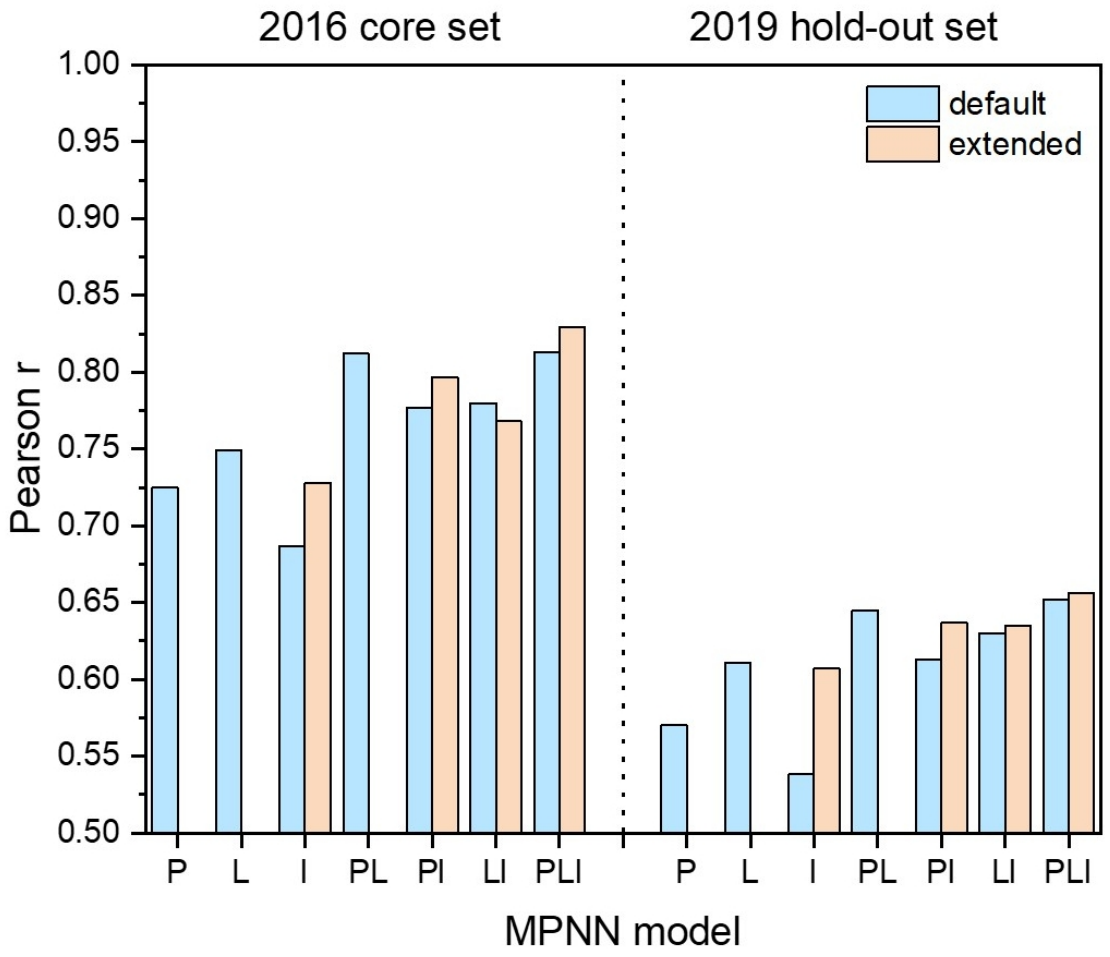


Figure 2.10 Influence of the interaction graph complexity on the on the scoring power of MPNN models in predicting binding affinities for the 2016 core set and the 2019 hold-out set. By default, (blue bars), protein–ligand interactions are computed using interaction-specific upper distance thresholds (hydrogen bond: 3.5 Å, aromatic π- π interactions: 4.0 Å, ionic bonds: 4.0 Å, hydrophobic interactions: 4.5 Å). In the extended mode (tan bars), a larger distance cut-off of 6.0 Å is applied to all non-covalent interactions. P: protein graph model, L: ligand graph model, I: interaction graph model; PL: merged protein and ligand graphs model, PI: merged protein and interaction graphs model; LI: merged ligand and interaction graphs model; PLI: merged protein, ligand and interaction graphs model.

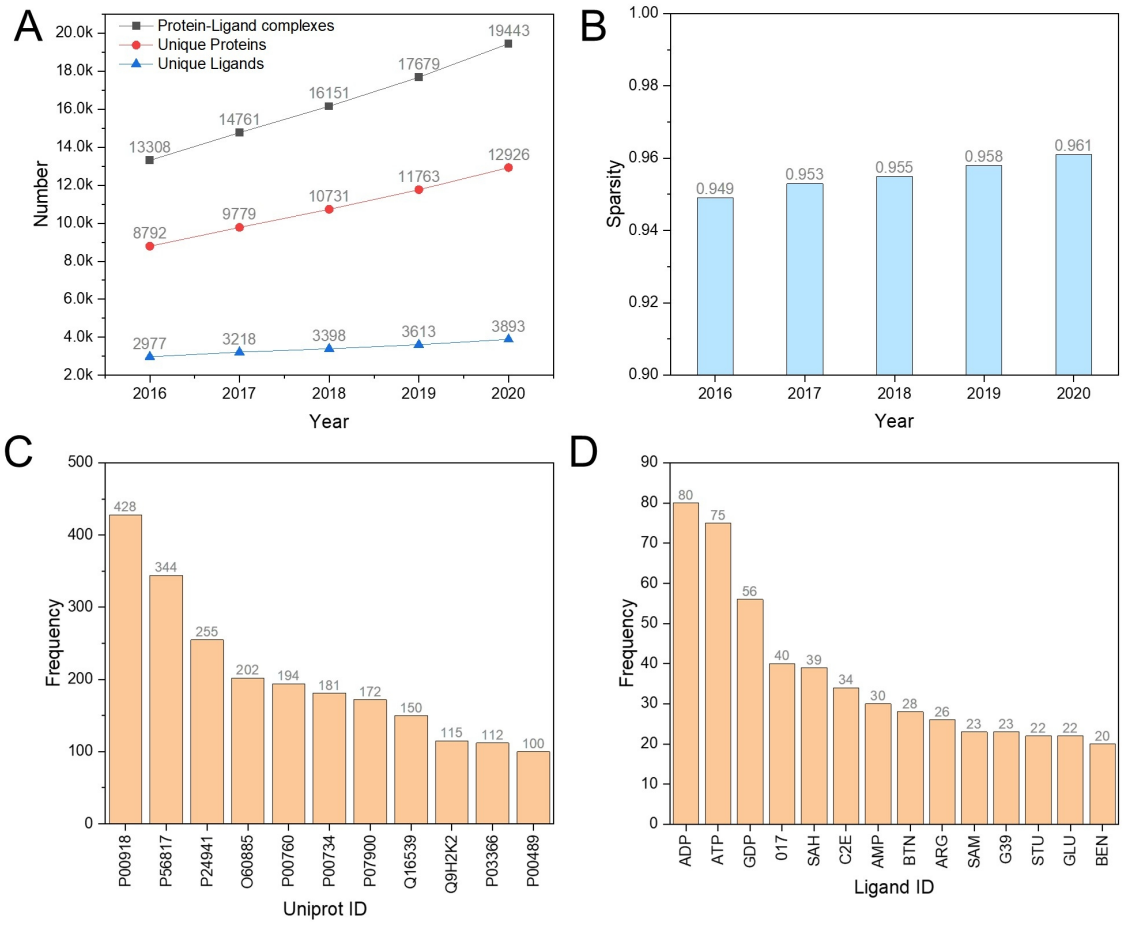


Figure 2.11 Yearly evolution of the PDBbind dataset. A) Number of unique entries (protein–ligand complexes, proteins, ligands), B) Sparsity of the protein–ligand matrix, C) Ten most frequent proteins (PDBbind 2020 release) labelled by their UniProt identifier, D) Ten most frequent ligands (2020 release) labelled by their PDB ligand identifier.

Sparsity of the training protein–ligand matrix

Despite a regular increase in the number of entries in PDBbind (Fig. 8A), the accuracy of machine learning models in predicting binding affinities has reached a plateau (R_p = 0.80 ± 0.05), whatever the DNN architecture, the chosen descriptors and the size of the training set (Table 2.1, Table 2.3). Higher accuracies are not necessarily required, given the experimental error associated with heterogeneous binding assays use to collect PDBbind affinities. However, better models are still desirable, notably to achieve accurate and stable predictions when applied to external test sets. Looking at the yearly increase in the number of PDBbind samples, it appears that the number of unique complexes grows faster that the number of unique proteins, the latter increasing faster than the number of unique ligands (Fig. 8A).

Considering a matrix of x proteins, y ligands and z protein–ligand complexes of known structure, the sparsity S of the PDBbind matrix is defined by the following equation:

$$S = 1 - \frac{z}{x \cdot y} \tag{2.4}$$

In other words, the sparsity index describes the fraction of the overall matrix with a missing value (here a protein–ligand complex of known structure and binding affinity). The sparsity S value is very high for the PDBbind dataset (ca. 0.95) and even tends to slightly increase with time (Figure 2.11 B). By comparison with high-performance QSAR models, that rely on a minimal number of compound annotations per assay (usually >200), and now reach the accuracy of four-concentration IC_{50} determinations, [52] the sparsity of the corresponding protein–ligand matrices may reach values as low as 0.65. [52, 56, 57]

The PDBbind matrix contains very few targets annotated by multiple ligands (Figure 2.11 C). The number of single ligands annotated by multiple proteins is even lower and mostly concerns target-permissive cofactors and nucleotides (e.g., ATP, ADP, AMP, SAM; Figure 2.11 D). To check the influence of the training matrix sparsity, we selected the 2030 PDBbind entries from the ten most frequent proteins (Figure 2.11 D) to design novel training (n=1505), validation (n=147), and external test sets (core 2016,

n=49; hold-out 2019, n=329). Importantly, the set membership (training, evaluation, core, hold-out) of selected entries was kept unchanged, as well as the distribution of experimental affinities (Figure 2.12). The previously described extended interaction model (int6) was here used to describe noncovalent interactions. Altogether, the new subset contains only 10 unique proteins and 1777 unique ligands, thereby achieves a lower sparsity ($S = 0.885$) with respect to the full PDBbind 2019 dataset ($S = 0.958$).

The performance of the MPNN models on the new subset is higher than that obtained on the full set (Figure 2.13). Unfortunately, neither protein nor ligand dependencies have been removed when predicting affinities for the two external test sets still focusing of the 10 most frequent proteins. The protein-only and ligand-only models remain very accurate, notably for predicting affinities of core set samples. Interestingly, the interaction model is the only one for which the performance is significantly increased for the two external test sets (core set, $R_p = 0.852$, RMSE = 1.256; hold-out set, $R_p = 0.605$, RMSE = 1.363; Figure 2.13). The I model appears again as a reasonable choice for predicting affinities of novel protein–ligand complexes. The current study suggests that increasing the density of the training protein–ligand matrix is an attractive path to increase the accuracy of affinity prediction models. From a practical point of view, it will necessitate a coordinated effort from the drug design community and research financing agencies to solve a wide array of protein–ligand structures in which the same target is repeatedly pictured with different ligands of various affinities, and vice-versa.

2.3 Experimental section

Dataset preparation

The index files of the PDBbind 2019 release were downloaded from the PDBbind website. [41] For each registered protein–ligand complex, the corresponding atomic coordinates (PDB format) were retrieved from the RCSB Protein Data Bank [58] and processed with Protoss v.4.0 [59] to generate atomic coordinates of hydrogen atoms while optimizing the protonation and ionizable states of both ligand and protein amino acids. Each structure was then postprocessed using an in-house script to keep only

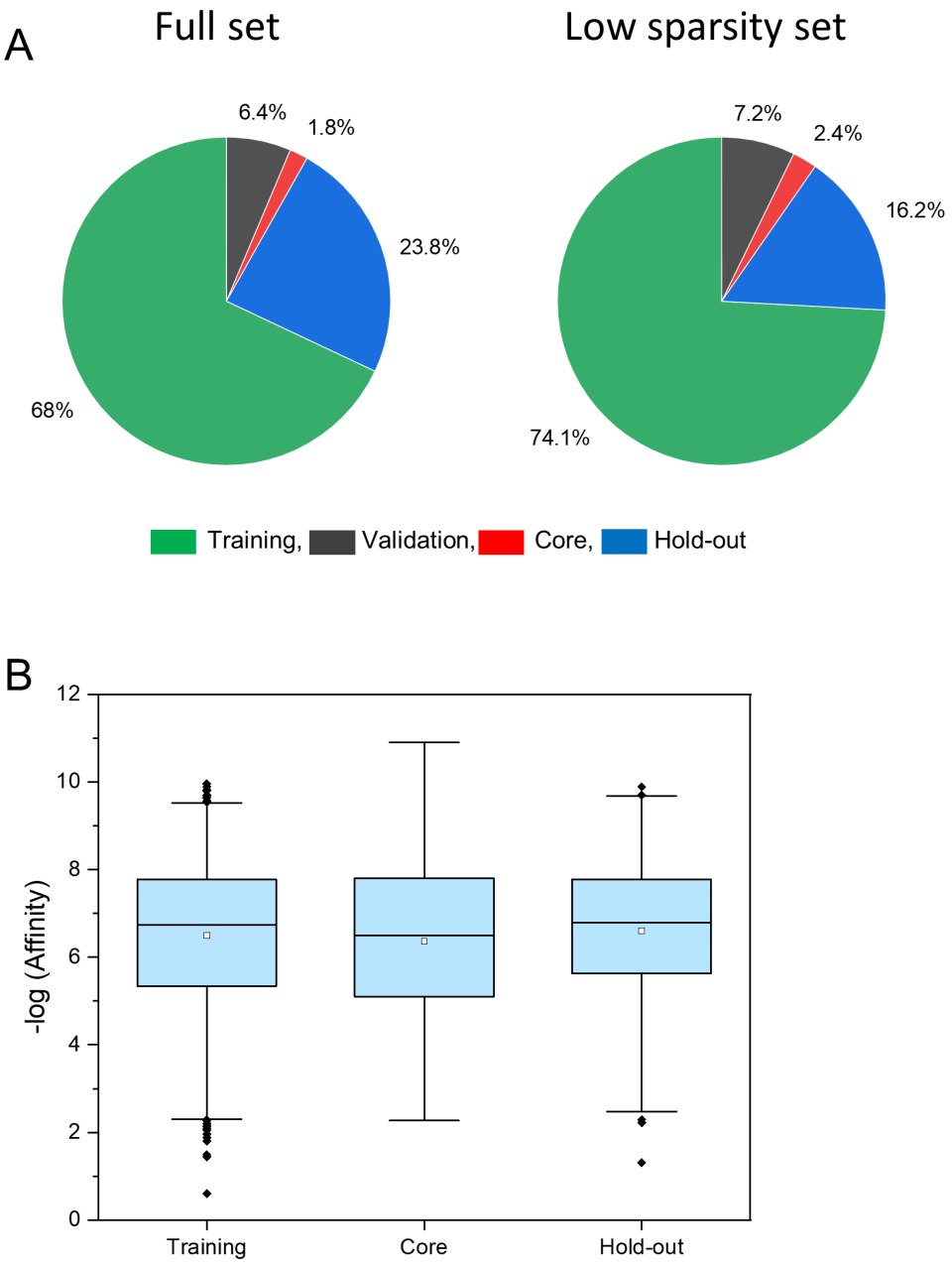


Figure 2.12 PDBbind low sparsity subset. **A)** Split into four subsets for training, validation and external test sets (core 2016, hold-out 2019), **B)** Distribution of experimental affinities for the training (n=1,505), core (n=49) and hold-out sets (n=329). The boxes delimit the 25th and 75th percentiles, and the whiskers delimit the 1st and 99th percentiles. The median and mean values are indicated by a horizontal line and a filled square in the box, respectively. Outliers are indicated by a diamond.

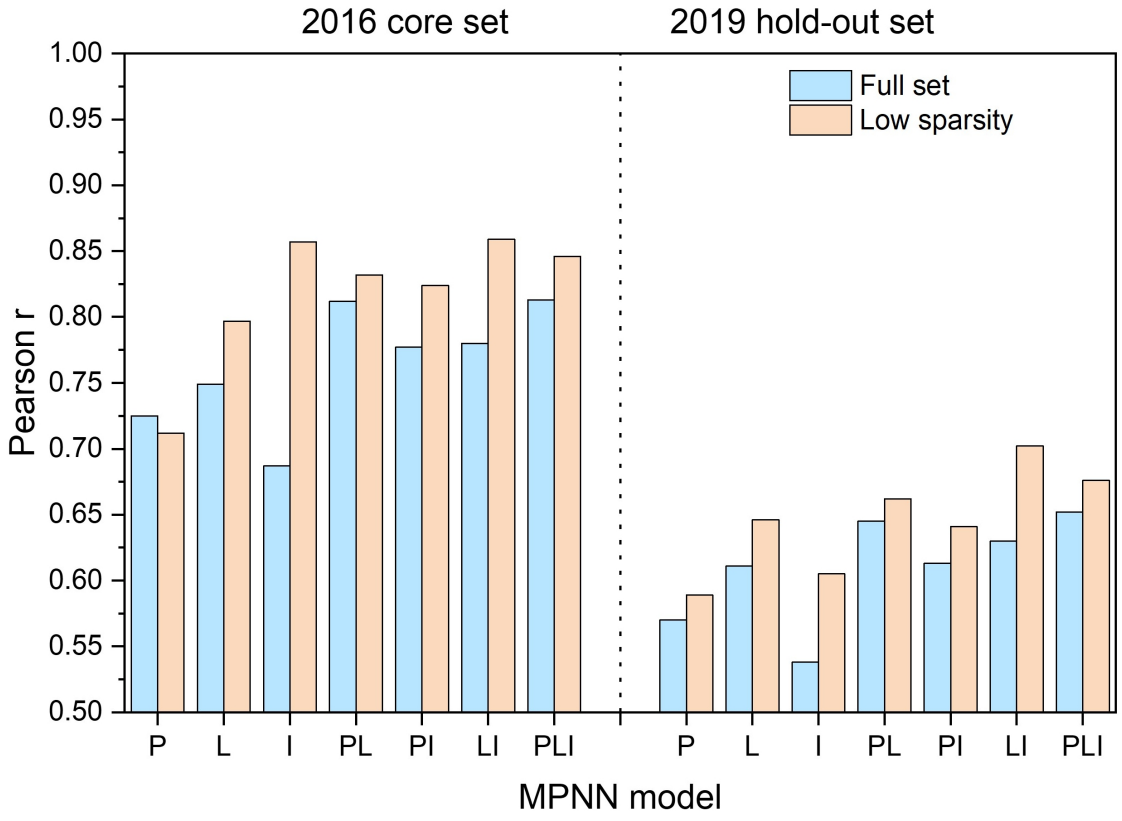


Figure 2.13 Increasing the density of training protein–ligand matrices to predict binding affinities for the 2016 core set and the 2019 hold-out set. P: protein graph model, L: ligand graph model, I: interaction graph model; PL: merged protein and ligand graphs model, PI: merged protein and interaction graphs model; LI: merged ligand and interaction graphs model; PLI: merged protein, ligand and interaction graphs model.

water molecules exhibiting , according to IChem [50] rules, at least two hydrogen bonds to either protein or ligand atoms. Entries with covalently bound ligands were excluded. Remaining protonated ligand and protein (including all remaining bound water molecules, cofactors, prosthetic groups and ions) were saved separately in mol2 file format. A curated set of 14215 complexes, for which graphs generation succeeded without any failure, was further split in two parts according to the release date (part 1: until 2016-12-31, part 2: after 2017-01-01). Part 1 complexes, corresponding to the general and refined 2016 sets, were divided into training (9662 entries), validation (903 entries) and test (257 entries) as previously described. [35] Part 2 (3386 entries) was saved as an external hold-out set, mimicking a real temporal split scenario in which binding affinities for newly released structures are predicted by a model trained on past structural data. Analyzing the distribution of pairwise ligand similarities evidence a large scaffold diversity of each set (training set, 2016 core set, 2019 hold-out set) as well as the absence of obvious similarity biases when comparing the training set to the two external sets. The pairwise similarity and UMAP [60] plots of all PDBbind ligands are provided on (Figure 2.14).

Molecular descriptors

Proteins, ligands, and protein–ligand interactions were represented as graphs using in-house scripts and the IChem package. [50] The graph processing pipeline was implemented using the Networkx framework v.2.5. [61]

Message passing neural networks

The neural network models were implemented using PyTorch v.1.6.0 [62]and PyTorch Lightning v.1.5.1. [63] The graph convolution procedure was implemented with a Deep Graph Library framework v.0.5.0. [64] Two approaches were tested in order to consider the three molecular graphs. In the 'merged approach' the feature vectors of the three input graphs are simply merged. An alternative architecture (parallel approach) was tested, which included separate MPNNs for each input, yielding parallel hidden vectors, which were concatenated before applying fully connected layers to them. Preliminary trials indicated that the parallel architecture had higher memory requirements and demanded longer computational time, while having an accuracy close to that obtained

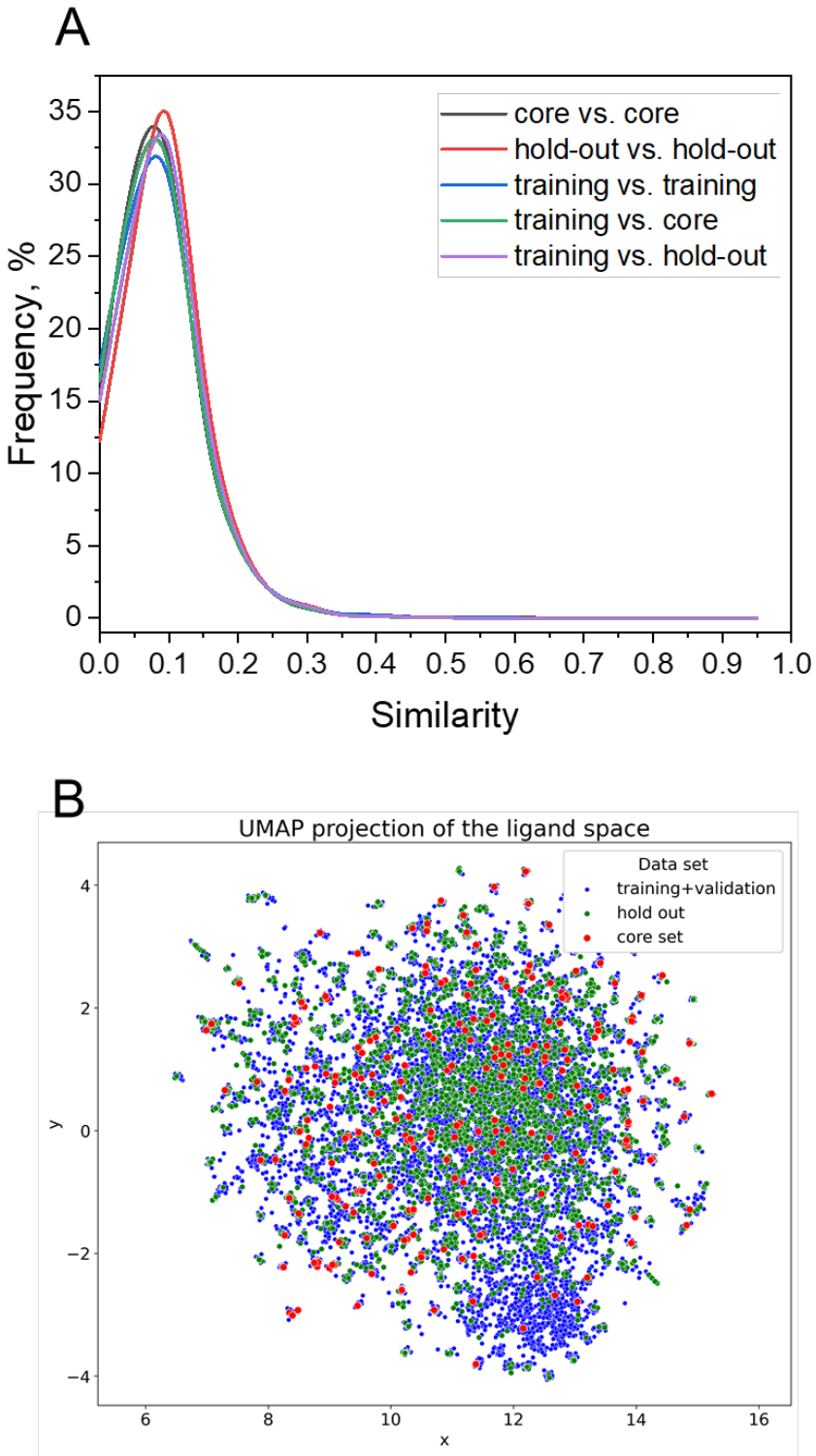


Figure 2.14 Chemical diversity of PDBbind ligands. **A)** Pairwise similarity of Murcko superstructures for ligands from the training, 2016 core and 2019 hold-out sets. The similarity is expressed by the Tanimoto coefficient computed from ECFP4 fingerprints. **B)** Uniform Manifold Approximation and Projection (UMAP)of PDBbind ligands, performed in umap-learn 0.5.3 with a number of neighbours of 30 and a dice distance metric. The Morgan fingerprints of radius 2 (nBits=1024) were computed in rdkit v.2020.09.1.

with the merged approach. Thus, the graph merging was selected as the preferable procedure of multiple graph inputs. The parameter optimization aimed to increase the determination coefficient R^2 in predicting binding affinities using a stochastic gradient descent approach with the ADAM optimizer. The learning rate (lr) was changed over time by the factor of 0.9 after 20 epochs with no improvement for the first lr modification and after 40 epochs for the subsequent lr modifications. The weight decay and dropout rate were set to values of 0.001 and 0.2, respectively. Other hyperparameters (batch size, size of hidden layers, number of message passing steps) were systematically optimized by a grid search as follows:

Batch size: search space [32 , 64, 128, 256], final value 256

size of hidden layers: search space [256 , 512, 1024, 2054], final value 2054

message passing steps: search space [1, 2], final value 1

Data undersampling

Data undersampling was performed using an iterative fivefold cross-validation approach on the whole PDBbind 2016 training set. At each iteration, ligand-only and protein-only MPNN models were trained using one fold as a test set and the remaining folds as a training set. Binding affinity was predicted for all test complexes with both models. At each iteration, training samples with the lowest sum of binding affinity prediction errors given by the two protein and ligand models were removed from the dataset. One hundred iterations of undersampling were performed and 50 complexes were removed at each iteration. The final undersampled training set contains 4635 protein–ligand complexes.

Prediction of binding affinities with Pafnucy [35]

The package was downloaded from the Pafnucy website. [65] In a first step, 3D grids were prepared for each protein–ligand complex in mol2 file format, to create an HDF file with atoms' coordinates and features. In the second step, the recommended model (batch5-2017-06-05T07:58:47-best) was used to rescore each protein–ligand complex, expressing results in pK_d unit.

Estimation of ligand buriedness

Ligand buriedness was computed with IChem v5.2.9 [50] using bound states of protein and ligand in separate mol2 files.

Ligand and protein pairwise similarity

Pairwise ligand similarities were computed from circular ECFP4 fingerprints [66] determined in PipelinePilot v.2019 (Dassault Systèmes Biovia Corp., San Diego, U.S.A). Protein similarities were estimated from the Euclidean distance of 89 cavity descriptors generated by IChem v5.2.9. [50]

Evaluation metrics

The scoring power of the different DNN models was evaluated using the Pearson's correlation coefficient (R_p; equation 2.5) and the root-mean square error metric (RMSE, equation 2.6).

$$R_p = \frac{\sum_{i=1}^{n}(X_i - X)(Y_i - Y)}{\sqrt{\sum_{i=1}^{n}(X_i - X)^2}\sqrt{\sum_{i=1}^{n}(Y_i - Y)^2}} \tag{2.5}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_i - Y_i)^2}{n}} \tag{2.6}$$

2.4 Conclusions

Predicting binding affinities of protein–ligand complexes by considering both the corresponding free and bound states appears frustrating because the explicit description of non-covalent intermolecular interactions does not provide any statistical advantage with respect to simpler approximations omitting fine details of protein–ligand interactions. The current study confirms the protein and ligand biases already observed in several studies using DUD-E and PDBbind data sets as sources of three-dimensional information. [21, 22, 25, 26, 28, 32, 33, 44, 46] However, important controversies still remain regarding the interpretation of these observations. On one side, many computer scientists are not alerted and keep focusing on a pure metrics-based analysis which usually shows that adding descriptors of protein–ligand interactions indeed

produce prediction models with slightly better performance metrics (Pearson R correlation, RMSE). [22, 25, 28, 32, 33] On the other side, several warnings have been raised by a few groups [21, 44, 46] arguing that a machine learning model must be interpretable from a physicochemical ground. We totally agree with the latter studies, but we were unable to find obvious ways to remove hidden protein and ligand biases in the PDBbind archive of protein–ligand complexes. Neither undersampling, nor considering ligand buriedness and sparsity of the protein–ligand training matrix could remove the observed tendency of deep neural models to accurately predict binding affinities from sole ligand or protein descriptors. The approach proposed by Yang et al. [21] to split the data set according to ligand scaffold and protein sequence/structure similarity is efficient in reducing protein and ligand biases but remains artificial and not satisfactory for daily practice where affinity data have to be predicted for new proteins bound to "old ligands" (repurposing), "old proteins" bound to new ligands (hit to lead optimization) and new proteins bound to new ligands (virtual screening). In the current study, we therefore privileged a temporal splitting protocol in which affinities for novel protein–ligand complexes are predicted from a model trained on past structural data. The sparsity of the protein–ligand training matrix appears to be the most important parameter, notably for models trained only on protein–ligand interactions. To avoid building models relying on ligand-specific and protein-specific features, we disfavor annotating the non-covalent interactions with explicit ligand and protein descriptors, as often seen in GNNs with attention procedures to annotate graph nodes with ligand and binding pocket connectivity atomic tables. [18, 25, 33, 38] As a conclusion, we recommend training DNN models on pure interaction descriptors in order to reduce the risk of overfitting. Only the latter models appear robust enough to be used for prospective applications.

2.5 Supporting information

Location and pharmacophoric properties of protein pseudoatoms; Performance of modular MPNN models in predicting affinities for specific target classes of the 2019 holdout set; Influence of the number of closest ligands or proteins used to average binding affinities in the performance of simple memorization models; PDBbind low sparsity

subset; Chemical diversity of PDBbind ligands; Structure-based deep neural networks to predict protein–ligand binding affinities; Geometric rules to define protein–ligand non-covalent interactions (PDF). This material is available free of charge via the Internet at http://pubs.acs.org

Data availability

Data. Input files (curated mol2 input files for PDBbind samples; ligand, protein and interaction graphs; training, validation and test set membership) are freely available at http://bioinfo-pharma.u-strasbg.fr/labwebsite/downloads/pdbbind.tgz. Software. Pafnucy version 1.0 was downloaded from https://gitlab.com/cheminfIBB/pafnucy, and used with default settings. Rescoring was performed using the recommended model batch5-2017-06-05T07:58:47-best. IChem (version 5.2.9) was downloaded from http://bioinfo-pharma.u-strasbg.fr/labwebsite/download.html. IChem is freely available for non-profit academic research and subjected to moderate license fees for companies.

Acknowledgements

The Calculation Center of the IN2P3 (CNRS, Villeurbanne, France) is acknowledged for the allocation of computing time and excellent support. We sincerely thank Prof. M. Rarey (University of Hamburg, Germany) for providing an executable version of Protoss. This study was funded by a PhD grant to M.V from Iktos SAS.

2.6 References

(1) Mobley, D. L.; Gilson, M. K. *Annual review of biophysics* **2017**, *46*, 531.

(2) Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K., et al. *Nature* **2019**, *566*, 224–229.

(3) Gorgulla, C.; Boeszoermenyi, A.; Wang, Z.-F.; Fischer, P. D.; Coote, P. W.; Padmanabha Das, K. M.; Malets, Y. S.; Radchenko, D. S.; Moroz, Y. S.; Scott, D. A., et al. *Nature* **2020**, *580*, 663–668.

(4) Hoffmann, T.; Gastreich, M. *Drug discovery today* **2019**, *24*, 1148–1156.

(5) Burley, S. K.; Bhikadiya, C.; Bi, C.; Bittrich, S.; Chen, L.; Crichlow, G. V.; Duarte, J. M.; Dutta, S.; Fayazi, M.; Feng, Z., et al. *Protein Science* **2022**, *31*, 187–208.

(6) Kollman, P. *Chemical reviews* **1993**, *93*, 2395–2417.

(7) Guedes, I. A.; Pereira, F. S.; Dardenne, L. E. *Frontiers in pharmacology* **2018**, *9*, 1089.

(8) Ain, Q. U.; Aleksandrova, A.; Roessler, F. D.; Ballester, P. J. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2015**, *5*, 405–424.

(9) Kim, J.; Park, S.; Min, D.; Kim, W. *International Journal of Molecular Sciences* **2021**, *22*, 9983.

(10) Kimber, T. B.; Chen, Y.; Volkamer, A. *International Journal of Molecular Sciences* **2021**, *22*, 4435.

(11) Cang, Z.; Wei, G.-W. *PLoS computational biology* **2017**, *13*, e1005690.

(12) Torng, W.; Altman, R. B. *Journal of chemical information and modeling* **2019**, *59*, 4131–4149.

(13) Zhang, H.; Liao, L.; Saravanan, K. M.; Yin, P.; Wei, Y. *PeerJ* **2019**, *7*, e7362.

(14) Zheng, L.; Fan, J.; Mu, Y. *ACS omega* **2019**, *4*, 15956–15965.

(15) Hassan-Harrirou, H.; Zhang, C.; Lemmin, T. *Journal of chemical information and modeling* **2020**, *60*, 2791–2802.

(16) Karlov, D. S.; Sosnin, S.; Fedorov, M. V.; Popov, P. *ACS omega* **2020**, *5*, 5150–5159.

(17) Kwon, Y.; Shin, W.-H.; Ko, J.; Lee, J. *International journal of molecular sciences* **2020**, *21*, 8424.

(18) Li, S.; Wan, F.; Shu, H.; Jiang, T.; Zhao, D.; Zeng, J. *Cell Systems* **2020**, *10*, 308–322.

(19) Wang, S.; Liu, D.; Ding, M.; Du, Z.; Zhong, Y.; Song, T.; Zhu, J.; Zhao, R. *Frontiers in Genetics* **2021**, *11*, 607824.

(20) Xie, L.; Xu, L.; Chang, S.; Xu, X.; Meng, L. *Chemical Biology & Drug Design* **2020**, *96*, 973–983.

(21) Yang, J.; Shen, C.; Huang, N. *Frontiers in pharmacology* **2020**, *11*, 69.

(22) Gomes, J.; Ramsundar, B.; Feinberg, E. N.; Pande, V. S. *arXiv preprint arXiv:1703.10603* **2017**.

(23) Zhu, F.; Zhang, X.; Allen, J. E.; Jones, D.; Lightstone, F. C. *Journal of chemical information and modeling* **2020**, *60*, 2766–2772.

(24) Ahmed, A.; Mam, B.; Sowdhamini, R. *Bioinformatics and Biology Insights* **2021**, *15*, 11779322211030364.

(25) Jiang, D.; Hsieh, C.-Y.; Wu, Z.; Kang, Y.; Wang, J.; Wang, E.; Liao, B.; Shen, C.; Xu, L.; Wu, J., et al. *Journal of medicinal chemistry* **2021**, *64*, 18209–18232.

(26) Jones, D.; Kim, H.; Zhang, X.; Zemla, A.; Stevenson, G.; Bennett, W. D.; Kirshner, D.; Wong, S. E.; Lightstone, F. C.; Allen, J. E. *Journal of chemical information and modeling* **2021**, *61*, 1583–1592.

(27) Kumar, S.; Kim, M.-h. *Journal of cheminformatics* **2021**, *13*, 1–17.

(28) Liu, Q.; Wang, P.-S.; Zhu, C.; Gaines, B. B.; Zhu, T.; Bi, J.; Song, M. *Journal of Molecular Graphics and Modelling* **2021**, *105*, 107865.

(29) Seo, S.; Choi, J.; Park, S.; Ahn, J. *BMC bioinformatics* **2021**, *22*, 1–15.

(30) Shen, H.; Zhang, Y.; Zheng, C.; Wang, B.; Chen, P. *International journal of molecular sciences* **2021**, *22*, 4023.

(31) Son, J.; Kim, D. *PloS one* **2021**, *16*, e0249404.

(32) Lau, T.; Dror, R. **2017**.

(33) Feinberg, E. N.; Sur, D.; Wu, Z.; Husic, B. E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V. S. *ACS central science* **2018**, *4*, 1520–1530.

(34) Jiménez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. *Journal of chemical information and modeling* **2018**, *58*, 287–296.

(35) Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P. *Bioinformatics* **2018**, *34*, 3666–3674.

(36) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. *Chemical science* **2018**, *9*, 513–530.

(37) Li, Y.; Rezaei, M. A.; Li, C.; Li, X. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2019, pp 303–310.

(38) Lim, J.; Ryu, S.; Park, K.; Choe, Y. J.; Ham, J.; Kim, W. Y. *Journal of chemical information and modeling* **2019**, *59*, 3981–3988.

(39) Xiong, J.; Xiong, Z.; Chen, K.; Jiang, H.; Zheng, M. *Drug Discovery Today* **2021**, *26*, 1382–1393.

(40) Wang, R.; Fang, X.; Lu, Y.; Wang, S. *Journal of medicinal chemistry* **2004**, *47*, 2977–2980.

(41) PDBBind, `http://www.pdbbind.org.cn/`, Accessed: 2022-03-12.

(42) Wang, J.; Dokholyan, N. V. *Journal of Chemical Information and Modeling* **2022**, *62*, 463–471.

(43) Gabel, J.; Desaphy, J.; Rognan, D. *Journal of chemical information and modeling* **2014**, *54*, 2807–2815.

(44) Sieg, J.; Flachsenberg, F.; Rarey, M. *Journal of chemical information and modeling* **2019**, *59*, 947–961.

(45) Shen, C.; Hu, Y.; Wang, Z.; Zhang, X.; Pang, J.; Wang, G.; Zhong, H.; Xu, L.; Cao, D.; Hou, T. *Briefings in Bioinformatics* **2021**, *22*, bbaa070.

(46) Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C. J.; Duca, J. S.; Hornak, V.; Koes, D. R.; Kurtzman, T. *PloS one* **2019**, *14*, e0220113.

(47) Öztürk, H.; Özgür, A.; Ozkirimli, E. *Bioinformatics* **2018**, *34*, i821–i829.

(48) Nguyen, T.; Le, H.; Quinn, T. P.; Nguyen, T.; Le, T. D.; Venkatesh, S. *Bioinformatics* **2021**, *37*, 1140–1147.

(49) Schmitt, S.; Kuhn, D.; Klebe, G. *Journal of molecular biology* **2002**, *323*, 387–406.

(50) Da Silva, F.; Desaphy, J.; Rognan, D. *ChemMedChem* **2018**, *13*, 507–510.

(51) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. In *International conference on machine learning*, 2017, pp 1263–1272.

(52) Martin, E. J.; Polyakov, V. R.; Zhu, X.-W.; Tian, L.; Mukherjee, P.; Liu, X. *Journal of chemical information and modeling* **2019**, *59*, 4450–4459.

(53) Wallach, I.; Heifets, A. *Journal of chemical information and modeling* **2018**, *58*, 916–932.

(54) Tran-Nguyen, V.-K.; Bret, G.; Rognan, D. *Journal of Chemical Information and Modeling* **2021**, *61*, 2788–2797.

(55) Marcou, G.; Rognan, D. *Journal of chemical information and modeling* **2007**, *47*, 195–207.

(56) Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; Zarrinkar, P. P. *Nature biotechnology* **2011**, *29*, 1046–1051.

(57) Metz, J. T.; Johnson, E. F.; Soni, N. B.; Merta, P. J.; Kifle, L.; Hajduk, P. J. *Nature chemical biology* **2011**, *7*, 200–202.

(58) Protein Data Bank, `https://www.rcsb.org`, Accessed: 2022-12-03.

(59) Bietz, S.; Urbaczek, S.; Schulz, B.; Rarey, M. *Journal of cheminformatics* **2014**, *6*, 1–12.

(60) McInnes, L.; Healy, J.; Melville, J. *arXiv preprint arXiv:1802.03426* **2018**.

(61) Networkx- Network Analysis in Python, `https://networkx.org`, Accessed: 2022-03-12.

(62) Pytorch, `https://pytorch.org/`, Accessed: 2022-03-12.

(63) Pytorch Lightning, `https://www.pytorchlightning.ai/`, Accessed: 2022-03-12.

(64) Deep Graph Library, `https://www.dgl.ai/`, Accessed: 2022-03-12.

(65) Pafnucy Website, `https://gitlab.com/cheminfibb/pafnucy`, Accessed: 2022-12-03.

(66) Rogers, D.; Hahn, M. *Journal of chemical information and modeling* **2010**, *50*, 742–754.

Journal of Medicinal Chemistry

pubs.acs.org/jmc

Article

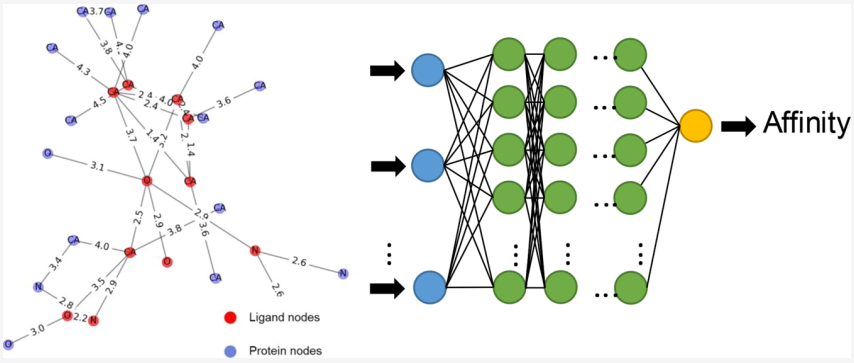
On the Frustration to Predict Binding Affinities from Protein−Ligand Structures with Deep Neural Networks

Mikhail Volkov, Joseph-André Turk, Nicolas Drizard, Nicolas Martin, Brice Hoffmann, Yann Gaston-Mathé, and Didier Rognan*

Cite This: https://doi.org/10.1021/acs.jmedchem.2c00487

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



ABSTRACT: Accurate prediction of binding affinities from protein−ligand atomic coordinates remains a major challenge in early stages of drug discovery. Using modular message passing graph neural networks describing both the ligand and the protein in their free and bound states, we unambiguously evidence that an explicit description of protein−ligand noncovalent interactions does not provide any advantage with respect to ligand or protein descriptors. Simple models, inferring binding affinities of test samples from that of the closest ligands or proteins in the training set, already exhibit good performances, suggesting that memorization largely dominates true learning in the deep neural networks. The current study suggests considering only noncovalent interactions while omitting their protein and ligand atomic environments. Removing all hidden biases probably requires much denser protein−ligand training matrices and a coordinated effort of the drug design community to solve the necessary protein−ligand structures.

INTRODUCTION

Predicting absolute binding free energies (affinities) from three-dimensional (3D) atomic coordinates of protein−ligand complexes remains one of the grand challenges of computational chemistry.[1] For example, drug discovery would immediately benefit from key advances in this topic, by better triaging potentially interesting molecules among virtual screening hits[2,3] and proposing viable analogues in emerging ultra-large chemical spaces[4] for hit to lead optimization. With the ever increasing amount of high-resolution experimentally determined protein−ligand structures,[5] binding affinity prediction algorithms have switched from physics-based[6] to empirical scoring functions,[7] and in the last few years to machine learning[8] and deep learning methods.[9,10] The latter category of descriptor-based scoring functions has notably led to numerous protein−ligand affinity models[11−38] (see a nonexhaustive list Table S1) notably because deep learning does not require explicit descriptor engineering and is ideally suited to find hidden nonlinear relationships between 3D protein−ligand structures and binding affinity. The first deep neural networks (DNNs) to predict binding affinities were convolutional neural networks (CNNs) reading a protein−ligand complex as an ensemble of grid-based voxels with multiple channels corresponding to pharmacophoric properties.[12,15,16] The CNN architecture is relatively inefficient from a computational point of view because most of the voxels do not carry any relevant information. Moreover, the search for the best possible hyperparameters is very demanding with respect to memory usage and CPU time. Finally, the same object must be presented in multiple orientations in a 3D grid to remove the dependency on the initial atomic coordinates. To overcome these issues and speed-up the training process, most recently developed DNNs read inputs in the form of a molecular graph[39] where nodes are represented by atoms and edges by bonds and/or noncovalent intra and intermolecular

Received: March 28, 2022

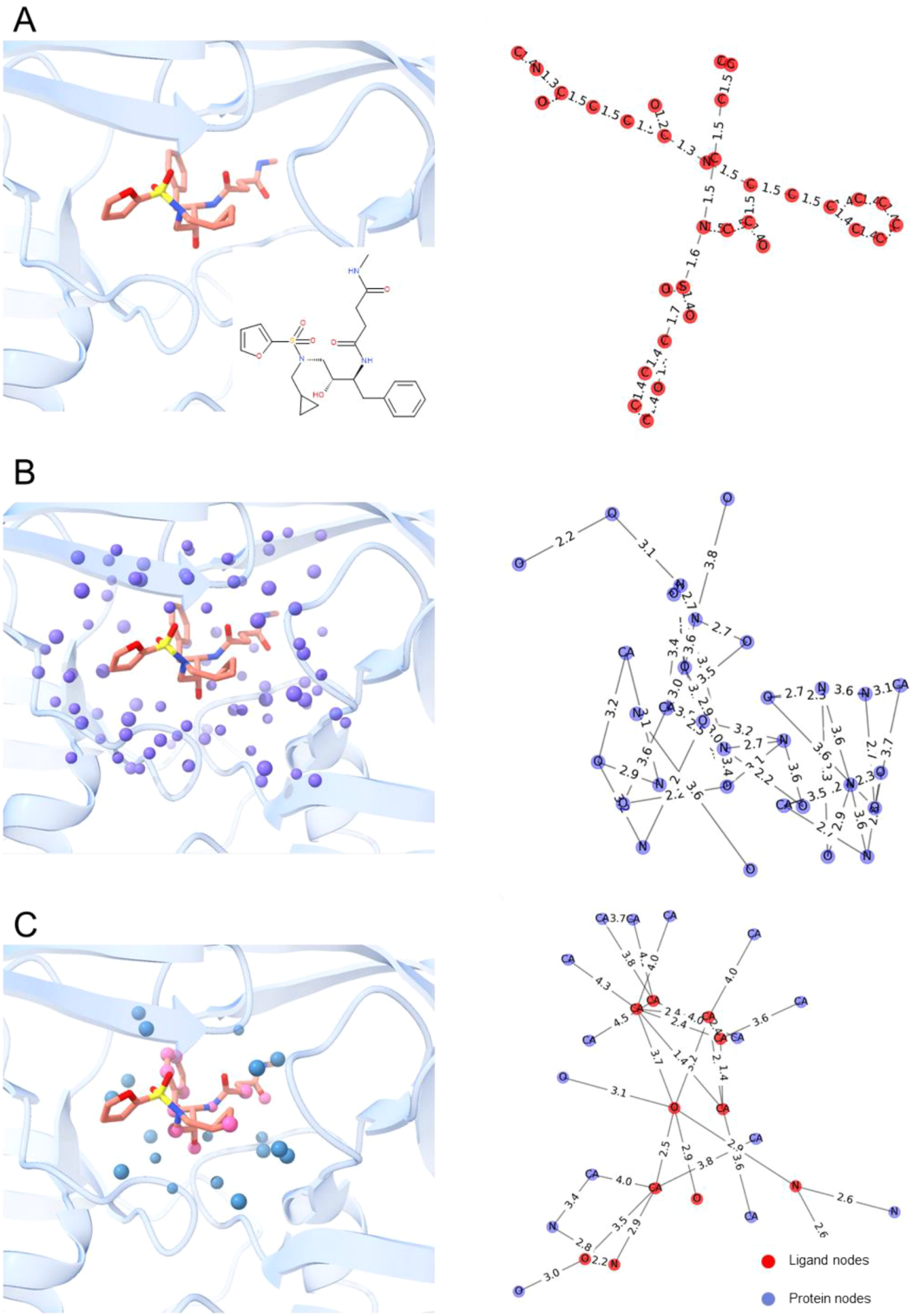


Figure 1. Encoding protein, ligand, and protein−ligand structures (PDB ID 2PSV) in graphs. (A) Nodes are set at ligand atomic coordinates (2D sketch in the inset), and labeled by atomic elements. Edges represent bonds, annotated by the bond length. (B) Proteins are represented by ligand-binding site pseudoatoms (slate blue spheres) placed at amino acid-specific positions. Nodes are set at PPA coordinates and annotated by pharmacophoric properties. Edges link two nodes distant by less than 4.0 Å. (C) Protein−ligand interactions are represented by interaction pseudoatoms (IPAs) (pink and blue spheres) set at protein and ligand-interacting atoms. Edges are placed between two nodes (protein, blue; ligand, red) in direct interaction, or between protein and ligand notes if distant by less than 4.0 Å. Each edge is annotated by the distance between the corresponding nodes.

interactions. Atoms and edges are embedded with user-defined atomic and/or pharmacophoric properties, enabling all graph components to be updated according to their surroundings all along the network during the training phase.

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

A gold standard data set to probe DNN models is the PDBbind database, developed by Wang et al.[40] and updated on a regular basis.[41] In its last version (v.2020), it stores 19443 protein−ligand X-ray structures of known binding affinity expressed as either the inhibition constant (K_i), dissociation constant (K_d), or half-maximum inhibition concentrations (IC_{50}). The general set which encompasses all data is further split in a refined set (5316 entries in the v.2020 release) containing high-quality X-ray structures and the most reliable affinity data (K_i and K_d only) and a core set (290 entries) made of a set of 58 proteins cocrystallized with five different ligands of various affinities. Despite several warnings on the composition[29] and completeness[42] of the PDBbind archive, it remains the largest resource to train machine learning models for structure-based prediction of binding affinities. Many graph neural networks (GNNs), used as end-to-end standalone architecture,[14,19,20,24,38] in cascade[37] or in combination with CNNs,[33] have been described recently. None of them significantly outperforms first-generation CNNs, most models presenting rather similar accuracies (Pearson correlation coefficient in the 0.80−0.85 range; root-mean square error (RMSE) around 1.2−1.3 pK unit) in predicting affinities for the PDBbind core set (Table S1) but significantly lower accuracies for true external test sets.[31,33,35]

Despite the strong commitment of data scientists, we believe that drug discovery has not really benefited from the already described models for the major reasons that machine (deep) learning scoring functions still generalize poorly and are not readily applicable to virtual screening of large compound libraries.[32] This major discrepancy does not prevent computer scientists to propose novel deep learning models, almost on a monthly basis, usually focusing on the novelty of the DNN architecture but often omitting to answer three questions: (i) Is the apparent performance biased by either the chosen descriptors[43,44] or the protein−ligand training space?[29,45] (ii) Does the model generalize well to external test sets? (iii) Has the model captured the physics of intermolecular interactions and does it achieve good predictions for meaningful reasons?

A first warning has been raised by several groups noticing that CNNs trained on voxelized protein−ligand complexes or graphs do not really learn the physics of protein−ligand recognition because ligand-only or protein-only models exhibit performances quite similar to those reached by protein−ligand reading models.[14,46,44,29,35] Comparison of the performance of 24 recently published DNNs[11−38] reveals that the model accuracy is independent of the size of the training set (e.g., PDBbind general vs refined set; Table S1), contradicting the general idea that more high-quality input protein−ligand structures are required to generate better models. Data augmentation strategies consisting of adding high-quality docking poses to PDBbind X-ray structures also lead to contradictory results.[22,28,33,38] Although very few attempts to predict a true thermodynamic cycle considering proteins and ligands in their free and bound states have been reported,[12,29,35] it remains counter-intuitive that the best models are not obtained with architectures explicitly taking into account the three bound/unbound species. Moreover, there is no relationship between the complexity of protein (sequence vs structure) and ligand (SMILES strings vs 2D graphs vs 3D structures) descriptors and the accuracy of the resulting DNN models.[39,47,48] Simple models even omitting to consider the protein−ligand bound state are equally good at predicting binding affinities.[31,42,47,48] It is therefore tempting to speculate that DNNs just memorize hidden patterns in either the ligand or protein spaces on which the models have been trained. As a consequence, modifications of protocols used to split input data into training, validation, and test sets have a major impact on the accuracy and applicability domain of obtained models.[12,29]

Because the publicly available training set is limited to the world of PDBbind protein−ligand complexes, there is a need for better identifying still hidden biases in the PDBbind archive, as well as to remove probable redundancies in the choice of descriptors. In the current study, we present a critical evaluation of a modular message passing graph neural network architecture to predict binding affinities from three independent graphs describing proteins, ligands, and their complexes. The modularity of the DNN architecture enables depicting the true contribution of each state (free vs bound) of the two partners and to clearly evidence serious biases in both the ligand and protein compositions of the PDBbind space. The current study suggests that descriptors focusing on noncovalent interactions with no additional ligand/protein information are the most suited to unbiased learning.

RESULTS AND DISCUSSION

Describing Ligands, Proteins, and Protein−Ligand Complexes as Graphs. Ligand graphs were generated from PDBbind mol2 input files, defining atoms as nodes and bonds as edges. Each node was annotated by the corresponding atom element, whereas each edge was annotated by the corresponding bond length (Figure 1A).

Protein graphs were described from ligand-binding sites, defined as any amino acid, ion, or water molecule for which one heavy atom is less than 4 Å away from any ligand heavy atom (Figure 1B). In the protein graph, nodes correspond to protein pseudoatoms (PPAs), as previously defined by Schmitt et al.,[49] and are placed at key main chain/side chain positions and annotated by the molecular interaction properties of the corresponding residue (Figure S1). A total of six properties were used to annotate protein nodes with the following labels and interaction properties: CA, aliphatic (hydrophobic interactions); O, hydrogen-bond acceptor (hydrogen bond); CZ, aromatic ($\pi-\pi$ interaction); OG, hydrogen-bond acceptor and donor (hydrogen bond); N, hydrogen-bond donor (hydrogen bond); ZN, metal (metal chelation). To avoid keeping protein residues whose side chains are pointing outward the ligand-binding cavity, a residue-based filtering was performed based on the angle between the ligand center of mass, the residue c-alpha atom, and all residue-specific PPAs. PPAs of amino acid side chains, for which the corresponding angle was higher than 90°, were removed from the binding site definition. Finally, edges were added between final protein nodes distant by less than 4.0 Å and further annotated according to the distance between the corresponding PPAs.

Noncovalent interactions (hydrophobic, aromatic, hydrogen bonds, ionic bonds, metal chelation; see details in Table S2) between protein and ligands were computed on the fly with the GRIM routine of the IChem v5.2.9 package.[50]

For each interaction, IPAs are placed at the two atoms of the interacting pair (Figure 1C). The resulting representation was converted to a graph where nodes represent either protein or ligand-interacting atoms. Edges between nodes were added in two consecutive steps. First, the principal edges were added between interacting IPAs. Then, secondary edges were added between noninteracting IPAs under the conditions that the

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

corresponding IPAs originate from the same molecule (protein or ligand) and that their distance is less than 4 Å. Each node was annotated by one of the following labels, according to the nature of the corresponding noncovalent interaction: CA, hydrophobic; NZ; ionic (the interacting protein atom is positively charged); N, hydrogen-bond (the interacting protein atom is a donor); OG, hydrogen-bond (the interacting protein atom is both an acceptor and donor); O, hydrogen-bond (the interacting protein atom is an acceptor); CZ, aromatic; OD1, ionic (the interacting protein atom is negatively charged); ZN: metal coordination. An additional binary label was added to nodes to account for their belonging to either the protein or the ligand. The only edge feature is the distance between pseudoatoms corresponding to the graph nodes (edge length). Therefore, the information on the spatial structure of the binding site was partially preserved, while the representation remained invariant to binding site rotations and node numbering.

DNN Architecture. We used a graph CNN architecture that belongs to the family of message passing neural networks (MPNNs), recently shown to exhibit excellent performance in predicting quantum chemical properties.[51] The MPNN is here applied to an undirected graph G with node features x_v and edge features e_{vw}. In an MPNN, each node v in the graph has a hidden state h_v^t (feature vector). For each node v, a function of hidden states and edges of all neighboring nodes is aggregated. The hidden state of the node Vt is then updated with the obtained message $m_v^{t+1}$and its previous hidden state. Three main equations characterize the MPNN on graphs. First, the message m_v^{t+1} obtained from all neighboring nodes $N(v)$ is given by eq 1:

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

where M_t is the aggregation function applied at step t, h_v^t the hidden state of node v, h_w^t the hidden state of the neighboring node w, e_{vw} is the feature of edge between v and w.

The hidden state h_v^{t+1} of the node v is then updated according to eq 2:

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \quad (2)$$

where U_t, the update function, is another neural network used to update the hidden state by taking into account both the sum of all previous messages and the previous hidden state.

The message passing algorithm is repeated a user-defined number of times until the readout phase generates a final feature vector $\hat{y}$ describing the entire graph G according to eq 3:

$$\hat{y} = R(\{h_v^T \mid v \in G\}) \quad (3)$$

where R is the readout function, T is the number of time steps.

The message functions M_t, node update function U_t, and readout function R are all learned differentiable functions. The complete architecture of the graph convolutional network (Figure 2A) includes an MPNN module with a customizable hidden size and a two-layer dense module with a top layer size of hidden size/4. The invariance of the MPNN readout function to node and edge re-enumeration enables applying MPNNs to a merged input consisting of multiple disconnected graphs describing protein, ligand, and protein−ligand interactions without modifying the network architecture. In the current study, MPNN models have been derived from graphs

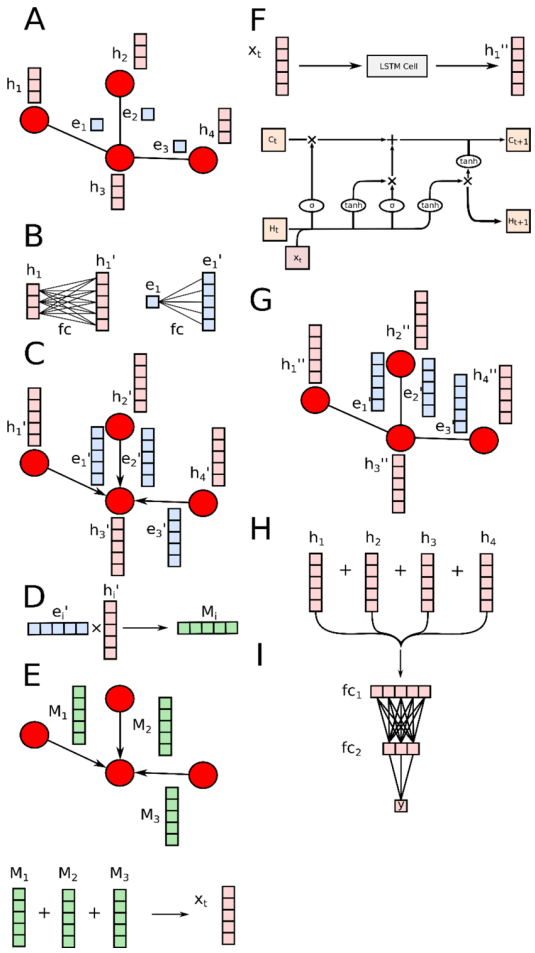


Figure 2. General architecture of an MPNN with two message passing steps. (A) Initial graph with node and edge labels. (B) Transformation of node and edge feature vectors with fully connected layers (fc). (C) Application of linear layers to node and edge feature vectors. (D) Message generation. (E) Message passing. (F) Node feature update using a standard LSTM cell architecture. (G) Graph with updated node features. (H) Readout. (I) Fully connected (fc) layers.

describing the two molecular species (protein and ligand) in both their liganded and unliganded states, thereby enabling to evaluate the exact contribution of each state. To ascertain the fairest possible comparison, all models were trained on the same training/validation set using exactly the same input graphs.

DNN Models Are Heavily Biased by Ligand and Protein Features. Starting from three possible input graphs describing the protein, the ligand, and their noncovalent interactions, seven combinations (one graph, two graphs, and three graphs) were first tested as baselines with two objectives: (i) benchmark the performance of the MPNN in predicting binding affinities with respect to other DNN architectures[11−31,33−38] and (ii) analyze the contribution of each input graph and assess their potential synergistic use (Table 1).

Despite our customized protocol to process PDBbind entries, we were able to reproduce the performance of the native Pafnucy model,[16] estimated by the Pearson's correlation coefficient Rp in predicting experimentally derived affinities for samples of the PDBBind 2016 core set (Rp = 0.777; Table 1). Our seven MPNN models exhibit various performances with

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

Table 1. Performance of Modular MPNN Models in Predicting Affinities for the External 2016 Core Set and the 2019 Hold-Out Set

model[a]	2016 core set		2019 hold-out set	
	Rp	RMSE[b]	Rp	RMSE
P	0.725	1.569	0.570	1.528
L	0.749	1.567	0.611	1.455
I	0.687	1.605	0.538	1.563
PL	0.812	1.553	0.645	1.512
PI	0.777	1.462	0.613	1.485
LI	0.780	1.477	0.630	1.425
PLI	0.813	1.511	0.652	1.481
Pafnucy[c]	0.773	1.429	0.456[d]	1.642

[a]P: protein graph, L: ligand graph, I: interaction graph; PL: merged protein and ligand graphs, PI: merged protein and interaction graph; LI: merged ligand and interaction graph; PLI: merged protein, ligand and interaction graph. [b]Root-mean square error, in pK unit. [c]In-house Pafnucy prediction (Rp = 0.78 in the original paper).[16] [d]Predictions failed for 29 entries.

Rp values ranging from 0.687 to 0.813. Intuitively, one would have expected that a model trained on protein−ligand interactions (I model) achieves better performance than models trained solely on either the ligands (L model) or the proteins (P model). However, the P and L models exhibit a better performance than the I model (Table 1). Out of the one-component models, the ligand-based model is clearly the one leading to the best results (Rp = 0.749 and RMSE = 1.567). Combining two graph inputs increases the accuracy of the corresponding predictions, with a clear advantage to the PL model (Rp = 0.812 and RMSE = 1.553) omitting protein−ligand interaction features. The most sophisticated model, taking into account the three graph inputs (PLI model), does not provide any clear advantage compared to the PL model, suggesting that explicitly defined molecular interactions are not required to predict binding affinities of the core set sample. Applying the models to a much larger (n = 3386) and more difficult hold-out set obtained by temporal splitting of the PDBbind data set (hold-out 2019 set) illustrates a moderate generalization capacity, with the Rp value decreasing by ca. 0.15 unit for all models (Table 1).

From a pure statistical point of view, the performance of four out of the seven MPNN models is superior to that achieved with the CNN Pafnucy model, when applied to the 2016 external core set (Table 1). Extending predictions to the challenging 2019 hold-out set suggests that all models outperform Pafnucy. Assuming that a Pearson Rp threshold value of 0.600 is commonly used in pharmaceutical industrial settings to qualify a good predictive QSAR model,[52] five out of the seven MPNN models could be considered as satisfactory. However, these models remain enigmatic from a physicochemical point of view because ligand-only and protein-only models still outperform the interaction model. Moreover, the impact on model predictive performance of the explicit consideration of protein−ligand interactions in the two or three-component models remains very limited (Table 1). Noteworthy, focusing the analysis on three target classes for which enough samples are present in the hold-out set (GPCRs, 47 samples; kinases, 572 samples; nuclear receptors, 106 samples) did not change the above observations (Figure S2).

Several conclusions can be drawn from these results. First, the herein implemented MPNN architecture provides a lower accuracy to previously reported CNN and GNN models, when just protein−ligand interactions are taken as input. Pafnucy, used here as a state-of-the-art CNN, achieves a better accuracy than the MPNN I model (Table 1). Second, protein−ligand binding affinities of the 2016 core set can apparently be predicted from sole protein or ligand structures. Third, the explicit description of protein−ligand interactions does not provide any clear advantage compared to the corresponding interaction-agnostic models (e.g., compare P to PI, L to LI, and PL to PLI models, Table 1). Fourth, all models exhibit a decreased accuracy when applied to a hold-out set of newly described complexes, suggesting a probable overtraining. Most of these observations are counter-intuitive and cannot be rationally explained by first-principles physics. They evidence, to our viewpoint, potential biases in the composition of the PDBbind training/test sets suggesting that the derived models have partly memorized input data but did not learn the physics of protein−ligand noncovalent interactions. This phenomenon has already been described for many ligand-based machine learning models and frequently happens when training and test sets exhibit significant redundancies.[53] Another alert, that we already mentioned for both machine learning and DNNs,[43,54] is their propensity to predict binding affinities with apparently satisfactory performance metrics (Rp, RMSE), but where the predicted values are in fact contained within a very tiny range centered on the mean value of training samples. This tendency is again observed for the current predictions of all MPNN models, whatever the chosen input graph(s) and external test set (Figure 3).

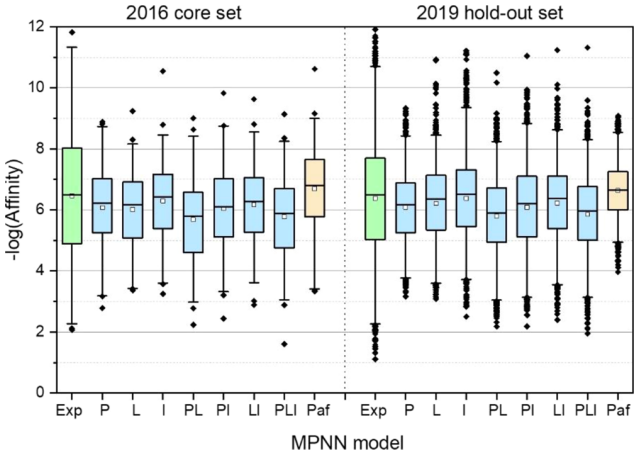


Figure 3. Distribution of experimental and predicted affinities for the 2016 core set (n = 257) and the 2019 hold-out set (n = 3386). Exp: experimental affinity; P, L, I, PL, PI, LI, PLI: predicted by MPNN models using protein (P), ligand (L), and protein−ligand interaction (I) graphs used alone or in combinations; Paf: predicted by the Pafnucy model. The boxes delimit the 25th and 75th percentiles, and the whiskers delimit the 1st and 99th percentiles. The median and mean values are indicated by a horizontal line and a filled square in the box, respectively. Outliers are indicated by a diamond.

Whereas experimental affinities of the two external test sets are spread over 10 pK units, MPNN and Pafnucy predictions are restricted to ca. 6 pK units. Considering only the 25th and 75th percentiles of the distributions (boxes in Figure 3), 50% of the predicted data are centered on a mean value ±1.5 pK unit, Pafnucy predictions lying even in a narrower range for 2019 hold-out set predictions (Figure 3). The prediction error

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

is statistically minored if the output value is close to the mean of trained samples. This may be a reason why machine learning models tend to yield narrow distribution of predicted values. This phenomenon might be even amplified in machine learning models for which the loss function aims at minimizing the root-mean-square error. Altogether, we suspect significant biases in the ligand and protein composition of the PDBbind archive which, to our viewpoint, should prevent the blind usage of DNN models in prospective applications.

Simple Memorization Models Suggest that Ligand and Protein Neighborhoods Contribute Massively to MPNN Predictions. To estimate the relative contribution of simple memorization vs true learning when applying MPNNs to predict affinities for PDBbind samples, we generated simple memorization baseline models in which the predicted affinity of a test sample was just inferred by ligand or protein similarity to the five closest training samples (Table 2). Of course, such memorization models are meaningless and just define baselines to quantify the amount of biases in the training data set.

Table 2. Performance of Simple Memorizing Models in Predicting Affinities for the External 2016 Core Set and the 2019 Hold-Out Set

model	2016 core set		2019 hold-out set	
	Rp	RMSE	Rp	RMSE
PLI MPNN[a]	0.813	1.511	0.652	1.481
ligand similarity[b]	0.663	1.624	0.509	1.641
protein similarity[c]	0.547	1.765	0.310	1.794

[a]Three-component (protein, ligand, and protein−ligand interactions) MPNN model of Table 1. [b]Prediction is equal to the average affinity of the five training samples with the most similar ligands, similarity being expressed by a Tanimoto coefficient on ECFP4 circular fingerprints (see the Experimental Section). [c]Prediction is equal to the average affinity of the five training samples with the most similar proteins, similarity being expressed by an Euclidean distance on protein cavity fingerprints (see the Experimental Section).

Given its simplicity, the ligand memorization model performs remarkably well on the two external test sets (Table 2) and is almost equivalent in accuracy to the protein−ligand interaction MPNN model (I model, Table 1). The protein similarity model exhibits a decreased but still noticeable performance. The observed dependency was relatively insensitive to the number of closest training samples (ligands, proteins) used to infer average affinity values for prediction (Figure S3). We can therefore conclude that simple memorization probably accounts for a large part of the excellent performance of the MPNN model using ligand, protein, and protein−ligand interaction graphs as input (Table 2).

Undersampling the Training Set Does Not Remove Ligand and Protein Biases. The goal of this procedure was to reduce the bias originating from the sampling of proteins and ligands present in the PDBBind data set. Thus, we undersampled the PDBbind training set by removing progressively the protein−ligand pairs which are easily predictable if we rely solely on protein or ligand graphs, while ignoring the interaction graphs. Intuitively, those are probably the most biased datapoints. As a first approach to remove potential ligand and protein biases in the training set, we filtered out all training samples whose affinities were easily predicted by ligand-only or protein-only fivefold cross-validation MPNN models. The protocol was repeated for batches of 50 samples to get a good tradeoff between speed and precision of the unbiasing algorithm.

Undersampling reduced the size of the training set from 9662 to 4635 samples but marginally affected the accuracy of all MPNN models, whatever the graphs used as inputs (Figure 4). Interestingly, decreasing the size of the training set by 50%

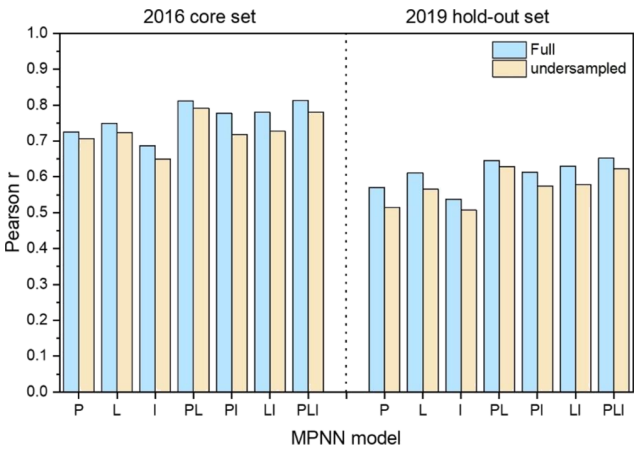


Figure 4. Effect of undersampling the PDBbind training set on the scoring power of MPNN models in predicting binding affinities for the 2016 core set and the 2019 hold-out set. Default models were trained on the full set (9662 entries), whereas undersampled models were trained only on 4635 samples. P: protein graph model, L: ligand graph model, I: interaction graph model; PL: merged protein and ligand graph model, PI: merged protein and interaction graph model; LI: merged ligand and interaction graph model; PLI: merged protein, ligand, and interaction graph model.

did not alter the quality of the predictions for both external sets. However, the same obvious biases (good performance of ligand-only and protein-only models, no benefit of explicitly considering protein−ligand interactions) were found again, suggesting that the hidden biases reported above are still present in the undersampled training set.

Influence of Ligand Buriedness. In a second approach, we looked whether the buriedness of the protein-bound ligands in the training and external sets may be a source of potential biases. Indeed, a fully buried ligand would generate quite complementary protein and ligand graphs that implicitly encode all possible noncovalent protein−ligand interactions. In such cases, it might be conceivable to predict albeit with a moderate accuracy the binding affinity of the corresponding complex from sole ligand or protein graphs.

Computing the buried surface area of all PDBbind ligands in their bound state shows a similar distribution for the three sets (training, 2016 core set, and 2019 hold-out set) centered on a mean value close to 60−65% (Figure 5A).

We then trained novel MPNN models on two subsets of the PDBbind training set defined by ligand buriedness. The first subset contained the samples with the 50% less buried ligands, whereas the second subset encompassed complexes with the 50% more buried ligands. Using these new MPNN models to predict the binding affinities of samples from the two external sets gave disappointing results (Figure 5B). First, all new models were less accurate than the former models trained on the full training set. Second, neither the ligand nor the protein dependency was removed in the new models because novel ligand-only (I models) and protein-only models (P models)

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

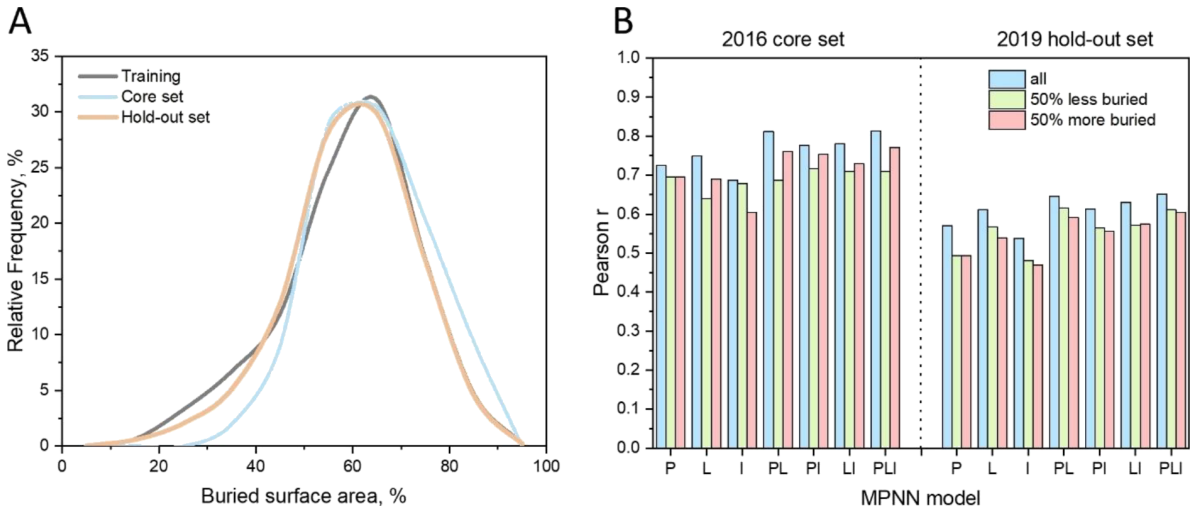


Figure 5. Effect of ligand buriedness on MPNN predictions. (A) Distribution of the buried surface area of protein-bound PDBbind ligands. (B) Influence of the protein-bound ligand buriedness on the scoring power of MPNN models in predicting binding affinities for the core set and the 2019 hold-out set. P: protein graph model, L: ligand graph model, I: interaction graph model; PL: merged protein and ligand graph model, PI: merged protein and interaction graph model; LI: merged ligand and interaction graph model; PLI: merged protein, ligand, and interaction graph model.

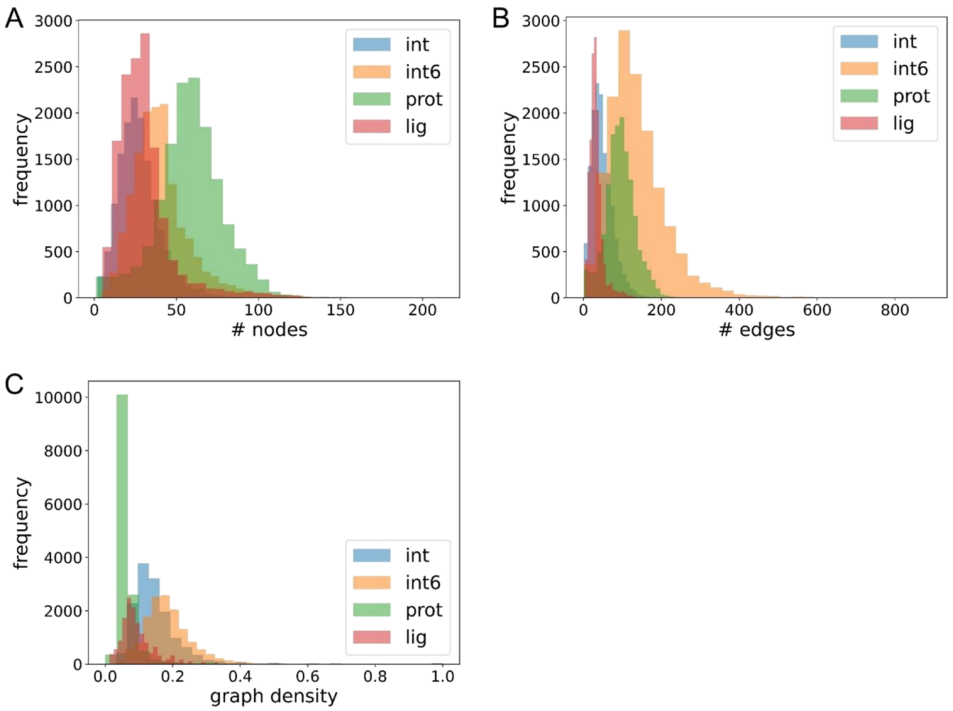


Figure 6. Distribution of the number of nodes (A), number of edges (B), and density (C) for interaction (int), protein (prot), and ligand graphs derived from PDBbind protein–ligand complexes (n = 14215). The graph density is defined as density $= \frac{N_{edges}}{N_{nodes}(N_{nodes}-1)}$, where N_{edges} is the number of edges and N_{nodes} is the number of nodes. By default, protein–ligand interactions are computed using interaction-specific upper distance thresholds (hydrogen bond: 3.5 Å, aromatic π–π interactions: 4.0 Å, ionic bonds: 4.0 Å, hydrophobic interactions: 4.5 Å). In the extended mode (int6), a larger distance cut-off of 6.0 Å is applied to all noncovalent interactions.

were still able to predict binding affinities of both external test samples (Figure 5B). We can therefore safely conclude that ligand buriedness is not the cause of protein and ligand biases in the PDBbind data set.

Complexity of the Protein–Ligand Interaction Descriptors. As a third approach, we made the hypothesis that the importance of protein and ligand descriptors with respect to the interaction descriptors may originate from the different complexity level of the input graphs. Indeed, interaction graphs computed in IChem are far simpler than the cognate protein and ligand graphs, when considering the number of nodes, edges, and the graph density. By default, protein–ligand interactions have been computed using strict geometrical rules (distances and angles),[55] notably interaction-specific upper

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX–XXX

distance thresholds (hydrogen bond: 3.5 Å, aromatic $\pi-\pi$ interactions: 4.0 Å, ionic bonds: 4.0 Å, and hydrophobic interactions: 4.5 Å), leading to relatively simple graphs with respect to the number of nodes and edges (Figure 6). To increase the importance of protein−ligand interactions in our MPNN models, we therefore increased the complexity of interaction graphs by registering noncovalent interactions up to of 6.0 Å. The new interaction graphs ("int6" label) contain much more nodes and edges, are definitely denser, and are now comparable with protein and ligand graphs (Figure 6).

Using the new interaction graphs as input to MPNN models increased significantly the scoring power of the interaction-only I model for the two external test sets (core set, Rp = 0.728; hold-out set, Rp = 0.607; Figure 7). Interestingly, this

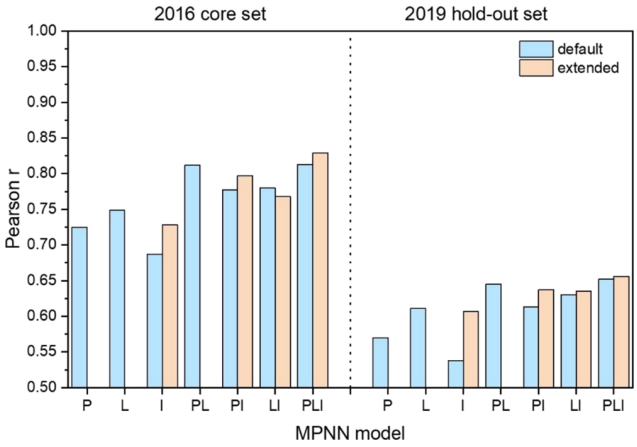


Figure 7. Influence of the interaction graph complexity on the scoring power of MPNN models in predicting binding affinities for the 2016 core set and the 2019 hold-out set. By default, (blue bars), protein−ligand interactions are computed using interaction-specific upper distance thresholds (hydrogen bond: 3.5 Å, aromatic $\pi-\pi$ interactions: 4.0 Å, ionic bonds: 4.0 Å, and hydrophobic interactions: 4.5 Å). In the extended mode (tan bars), a larger distance cut-off of 6.0 Å is applied to all noncovalent interactions. P: protein graph model, L: ligand graph model, I: interaction graph model; PL: merged protein and ligand graph model, PI: merged protein and interaction graph model; LI: merged ligand and interaction graph model; PLI: merged protein, ligand, and interaction graph model.

modification did not increase the accuracy of two-component and three-component models (Figure 7). Given the marginal benefit of combining the new interaction graph with either protein and/or ligand graphs, using the single new interaction graph definition appears as the best possible compromise between prediction accuracy, model applicability, and lower risk of memorization effects.

Sparsity of the Training Protein−Ligand Matrix. Despite a regular increase in the number of PDBbind entries (Figure 8A), the accuracy of machine learning models in predicting binding affinities has reached a plateau (Rp = 0.80 ± 0.05), whatever the DNN architecture, the chosen descriptors, and the size of the training set (Tables 1 and S1). Higher accuracies are not necessarily required, given the experimental error associated with heterogeneous binding assays use to collect PDBbind affinities. However, better models are still desirable, notably to achieve accurate and stable predictions when applied to external test sets. Looking at the yearly increase in the number of PDBbind samples, it appears that the number of unique complexes grows faster than the number of unique proteins, the latter increasing faster than the number of unique ligands (Figure 8A).

Considering a matrix of x proteins, y ligands, and z protein−ligand complexes of known structure, the sparsity S of the PDBbind matrix is defined by the following equation:

$$S = 1 - \frac{z}{x \cdot y} \tag{4}$$

In other words, the sparsity index describes the fraction of the overall matrix with a missing value (here a protein−ligand complex of known structure and binding affinity). The sparsity S value is very high for the PDBbind data set (ca. 0.95) and even tends to slightly increase with time (Figure 8B). By comparison with high-performance QSAR models that rely on a minimal number of compound annotations per assay (usually >200) and now reach the accuracy of four-concentration IC_{50} determinations,[52] the sparsity of the corresponding protein−ligand matrices may reach values as low as 0.65.[52,56,57]

The PDBbind matrix contains very few targets annotated by multiple ligands (Figure 8C). The number of single ligands annotated by multiple proteins is even lower and mostly concerns target-permissive cofactors and nucleotides (e.g., ATP, ADP, AMP, SAM; Figure 8D). To check the influence of the training matrix sparsity, we selected the 2030 PDBbind entries from the 10 most frequent proteins (Figure 8D) to design novel training (n = 1505), validation (n = 147), and external test sets (core 2016, n = 49; hold-out 2019, n = 329). Importantly, the set membership (training, evaluation, core, hold-out) of selected entries was kept unchanged, as well as the distribution of experimental affinities (Figure S4). The previously described extended interaction model (int6) was here used to describe noncovalent interactions. Altogether, the new subset contains only 10 unique proteins and 1777 unique ligands and thereby achieves a lower sparsity (S = 0.885) with respect to the full PDBbind 2019 data set (S = 0.958).

The performance of the MPNN models on the new subset is higher than that obtained on the full set (Figure 9). Unfortunately, neither protein nor ligand dependencies have been removed when predicting affinities for the two external test sets still focusing on the 10 most frequent proteins. The protein-only and ligand-only models remain very accurate, notably for predicting affinities of core set samples. Interestingly, the interaction model is the only one for which the performance is significantly increased for the two external test sets (core set, Rp = 0.852, RMSE = 1.256; hold-out set, Rp = 0.605, RMSE = 1.363; Figure 9). The I model appears again as a reasonable choice for predicting affinities of novel protein−ligand complexes. The current study suggests that increasing the density of the training protein−ligand matrix is an attractive path to increase the accuracy of affinity prediction models. From a practical point of view, it will necessitate a coordinated effort from the drug design community and research financing agencies to solve a wide array of protein−ligand structures in which the same target is repeatedly pictured with different ligands of various affinities, and vice-versa.

CONCLUSIONS

Predicting binding affinities of protein−ligand complexes by considering both the corresponding free and bound states appears frustrating because the explicit description of non-covalent intermolecular interactions does not provide any statistical advantage with respect to simpler approximations

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

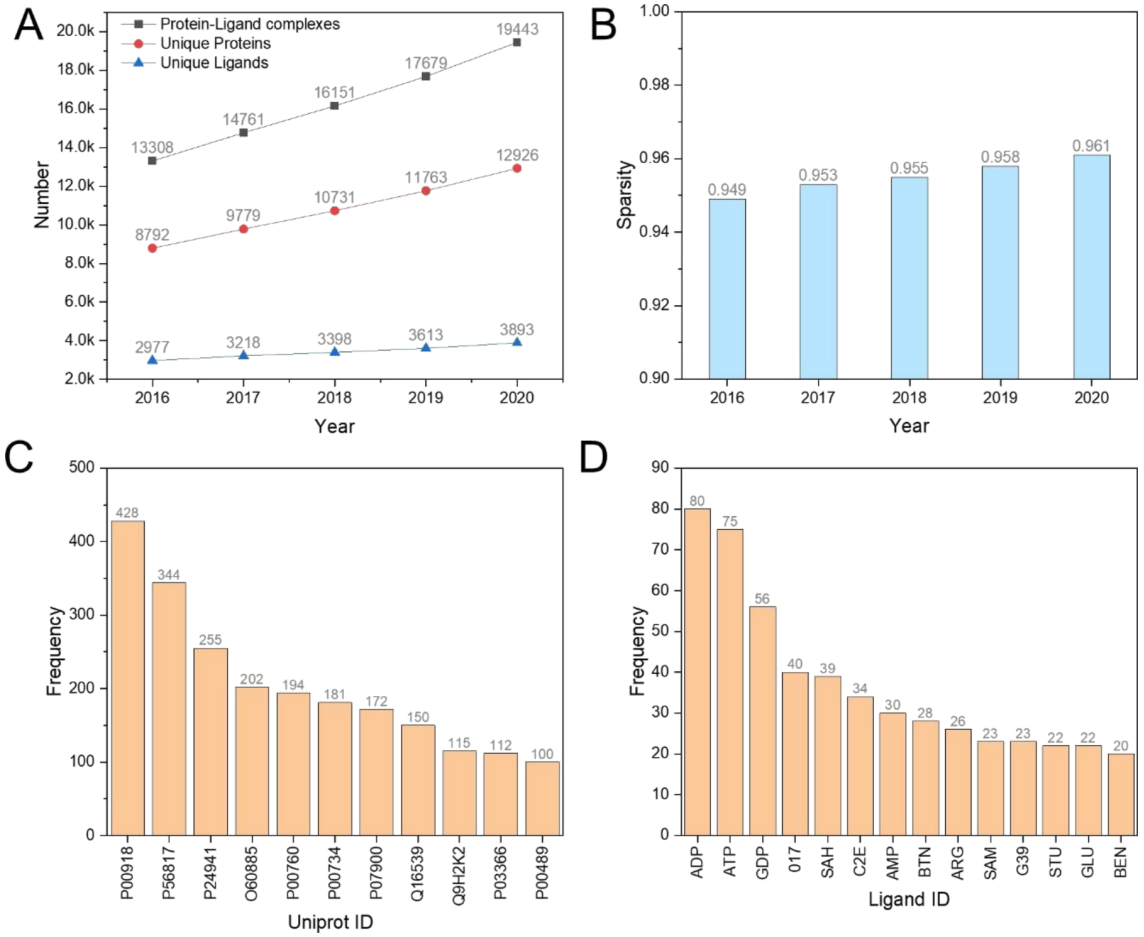


Figure 8. Yearly evolution of the PDBbind data set. (A) Number of unique entries (protein−ligand complexes, proteins, and ligands), (B) sparsity of the protein−ligand matrix, (C) ten most frequent proteins (PDBbind 2020 release) labeled by their UniProt identifier, and (D) ten most frequent ligands (2020 release) labeled by their PDB ligand identifier.

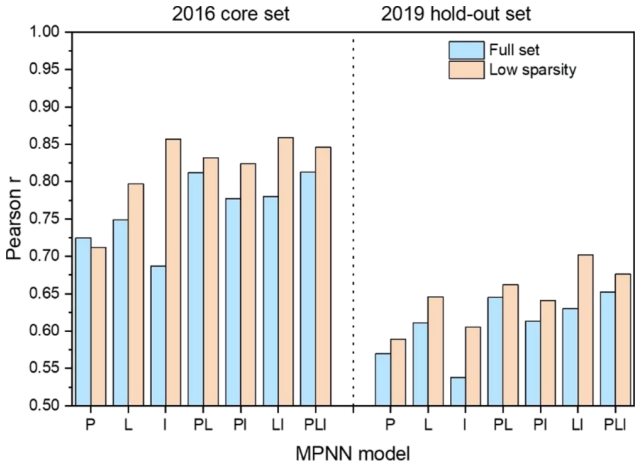


Figure 9. Increasing the density of training protein−ligand matrices to predict binding affinities for the 2016 core set and the 2019 hold-out set. P: protein graph model, L: ligand graph model, I: interaction graph model; PL: merged protein and ligand graph model, PI: merged protein and interaction graph model; LI: merged ligand and interaction graph model; PLI: merged protein, ligand, and interaction graph model.

omitting fine details of protein−ligand interactions. The current study confirms the protein and ligand biases already observed in several studies using DUD-E and PDBbind data sets as sources of three-dimensional information.[12−14,29,32,33,35,46,44] However, important controversies still remain regarding the interpretation of these observations. On one side, many computer scientists are not alerted and keep focusing on a pure metrics-based analysis which usually shows that adding descriptors of protein−ligand interactions indeed produce prediction models with slightly better performance metrics (Pearson R correlation, RMSE).[12−14,32,35] On the other side, several warnings have been raised by a few groups[29,46,44] arguing that a machine learning model must be interpretable from a physicochemical ground. We totally agree with the latter studies, but we were unable to find obvious ways to remove hidden protein and ligand biases in the PDBbind archive of protein−ligand complexes. Neither undersampling nor considering ligand buriedness and sparsity of the protein−ligand training matrix could remove the observed tendency of deep neural models to accurately predict binding affinities from sole ligand or protein descriptors. The approach proposed by Yang et al.[29] to split the data set according to ligand scaffolds and protein sequence/structure similarity is efficient in reducing protein and ligand biases but remains artificial and not satisfactory for daily practice where affinity data have to be predicted for new proteins bound to "old ligands" (repurposing), "old proteins" bound to new ligands (hit to lead optimization) and new

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

proteins bound to new ligands (virtual screening). In the current study, we therefore privileged a temporal splitting protocol in which affinities for novel protein−ligand complexes are predicted from a model trained on past structural data. The sparsity of the protein−ligand training matrix appears to be the most important parameter, notably for models trained only on protein−ligand interactions. To avoid building models relying on ligand-specific and protein-specific features, we disfavor annotating the noncovalent interactions with explicit ligand and protein descriptors, as often seen in GNNs with attention procedures to annotate graph nodes with ligand and binding pocket connectivity atomic tables.[14,19,26,32] As a conclusion, we recommend training DNN models on pure interaction descriptors to reduce the risk of overfitting. Only the latter models appear robust enough to be used for prospective applications.

EXPERIMENTAL SECTION

Data set Preparation. The index files of the PDBbind 2019 release were downloaded from the PDBbind website.[41] For each registered protein−ligand complex, the corresponding atomic coordinates (PDB format) were retrieved from the RCSB Protein Data Bank[58] and processed with Protoss v.4.0[59] to generate atomic coordinates of hydrogen atoms while optimizing the protonation and ionizable states of both ligand and protein amino acids. Each structure was then postprocessed using an in-house script to keep only water molecules exhibiting, according to IChem[50] rules, at least two hydrogen bonds to either protein or ligand atoms. Entries with covalently bound ligands were excluded. Remaining protonated ligand and protein (including all remaining bound water molecules, cofactors, prosthetic groups and ions) were saved separately in mol2 file format. A curated set of 14215 complexes, for which graph generation succeeded without any failure, was further split in two parts according to the release date (part 1: until 2016-12-31, part 2: after 2017-01-01). Part 1 complexes, corresponding to the general and refined 2016 sets, were divided into training (9662 entries), validation (903 entries) and test (257 entries) as previously described.[16] Part 2 (3386 entries) was saved as an external hold-out set, mimicking a real temporal split scenario in which binding affinities for newly released structures are predicted by a model trained on past structural data. Analyzing the distribution of pairwise ligand similarities evidences a large scaffold diversity of each set (training set, 2016 core set, 2019 hold-out set) as well as the absence of obvious similarity biases when comparing the training set to the two external sets. The pairwise similarity and UMAP[60] plots of all PDBbind ligands are provided in the Supporting Information (Figure S5).

Molecular Descriptors. Proteins, ligands, and protein−ligand interactions were represented as graphs using in-house scripts and the IChem package.[50] The graph processing pipeline was implemented using the Networkx framework v.2.5.[61]

Message Passing Neural Networks. The neural network models were implemented using PyTorch v.1.6.0[62] and PyTorch Lightning v.1.5.1.[63] The graph convolution procedure was implemented with a Deep Graph Library framework v.0.5.0.[64]

Two approaches were tested to consider the three molecular graphs. In the 'merged approach' the feature vectors of the three input graphs are simply merged. An alternative architecture (parallel approach) was tested, which included separate MPNNs for each input, yielding parallel hidden vectors, which were concatenated before applying fully connected layers to them. Preliminary trials indicated that the parallel architecture had higher memory requirements and demanded longer computational time, while having an accuracy close to that obtained with the merged approach. Thus, the graph merging was selected as the preferable procedure of multiple graph inputs. The parameter optimization aimed to increase the determination coefficient R^2 in predicting binding affinities using a stochastic gradient descent approach with the ADAM optimizer. The learning rate (lr) was changed over time by the factor of 0.9 after 20 epochs with no improvement for the first lr modification and after 40 epochs for the subsequent lr modifications. The weight decay and dropout rate were set to values of 0.001 and 0.2, respectively. Other hyperparameters (batch size, size of hidden layers, and number of message passing steps) were systematically optimized by a grid search as follows:

Batch size: search space [32, 64, 128, 256], final value 256.

size of hidden layers: search space [256, 512, 1024, 2054], final value 2054.

message passing steps: search space [1,2], final value 1.

Data Undersampling. Data undersampling was performed using an iterative fivefold cross-validation approach on the whole PDBbind 2016 training set. At each iteration, ligand-only and protein-only MPNN models were trained using one fold as a test set and the remaining folds as a training set. Binding affinity was predicted for all test complexes with both models. At each iteration, training samples with the lowest sum of binding affinity prediction errors given by the two protein and ligand models were removed from the data set. One hundred iterations of undersampling were performed and 50 complexes were removed at each iteration. The final undersampled training set contains 4635 protein−ligand complexes.

Prediction of Binding Affinities with Pafnucy.[16] The package was downloaded from the Pafnucy website.[65] In a first step, 3D grids were prepared for each protein−ligand complex in mol2 file format, to create an HDF file with atoms' coordinates and features. In the second step, the recommended model (batch5-2017-06-05T07:58:47-best) was used to rescore each protein−ligand complex, expressing results in pK_d unit.

Estimation of Ligand Buriedness. Ligand buriedness was computed with IChem v5.2.9[50] using bound states of protein and ligand in separate mol2 files.

Ligand and Protein Pairwise Similarity. Pairwise ligand similarities were computed from circular ECFP4 fingerprints[66] determined in PipelinePilot v.2019 (Dassault Systèmes Biovia Corp., San Diego, USA). Protein similarities were estimated from the Euclidean distance of 89 cavity descriptors generated by IChem v5.2.9.[50]

Evaluation Metrics. The scoring power of the different DNN models was evaluated using Pearson's correlation coefficient (Rp; eq 5) and the root-mean square error metric (RMSE, eq 6).

$$\mathrm{Rp} = \frac{\sum_{i=1}^{n}(X_i - X_\cdot)(Y_i - Y_\cdot)}{\sqrt{\sum_{i=1}^{n}(X_i - X_\cdot)^2}\sqrt{\sum_{i=1}^{n}(Y_i - Y_\cdot)^2}} \tag{5}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(X_i - Y_i)^2}{n}} \tag{6}$$

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jmedchem.2c00487.

Location and pharmacophoric properties of PPAs; performance of modular MPNN models in predicting affinities for specific target classes of the 2019 hold-out set; influence of the number of closest ligands or proteins used to average binding affinities in the performance of simple memorization models; PDBbind low sparsity subset; chemical diversity of PDBbind ligands; structure-based DNNs to predict protein−ligand binding affinities; geometric rules to define protein−ligand noncovalent interactions (PDF)

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

AUTHOR INFORMATION

Corresponding Author

Didier Rognan − *Laboratoire d'innovation thérapeutique, UMR7200 CNRS-Université de Strasbourg, Illkirch 67400, France;* orcid.org/0000-0002-0577-641X; Phone: +33 3 68 85 42 35; Email: rognan@unistra.fr; Fax: +33 3 68 85 43 10

Authors

Mikhail Volkov − *Laboratoire d'innovation thérapeutique, UMR7200 CNRS-Université de Strasbourg, Illkirch 67400, France;* orcid.org/0000-0002-4974-6079

Joseph-André Turk − *Iktos, Paris 75017, France*

Nicolas Drizard − *Iktos, Paris 75017, France*

Nicolas Martin − *Iktos, Paris 75017, France*

Brice Hoffmann − *Iktos, Paris 75017, France*

Yann Gaston-Mathé − *Iktos, Paris 75017, France*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jmedchem.2c00487

Notes

The authors declare no competing financial interest.

Data. Input files (curated mol2 input files for PDBbind samples; ligand, protein and interaction graphs; training, validation and test set membership) are freely available at http://bioinfo-pharma.u-strasbg.fr/labwebsite/downloads/pdbbind.tgz. *Software*. Pafnucy version 1.0 was downloaded from https://gitlab.com/cheminfIBB/pafnucy, and used with default settings. Rescoring was performed using the recommended model batch5-2017-06-05T07:58:47-best. IChem (version 5.2.9) was downloaded from http://bioinfo-pharma.u-strasbg.fr/labwebsite/download.html. IChem is freely available for non-profit academic research and subjected to moderate license fees for companies.

ACKNOWLEDGMENTS

The Calculation Center of the IN2P3 (CNRS, Villeurbanne, France) is acknowledged for the allocation of computing time and excellent support. We sincerely thank Prof. M. Rarey (University of Hamburg, Germany) for providing an executable version of Protoss. This study was funded by a PhD grant to M.V. from Iktos SAS.

ABBREVIATIONS

2D, two-dimensional; 3D, three-dimensional; ADP, adenosine diphosphate; AMP, adenosine monophosphate; ATP, adenosine triphosphate; CNN, convolutional neural network; CPU, central processing unit; DNN, deep neural network; ECFP, extended connectivity fingerprint; IC50, half maximal inhibitory concentration; IPA, interacting pseudoatom; kd, dissociation constant; Ki, inhibition constant; MPNN, message passing neural network; PDB, protein data bank; PPA, protein pseudoatom; Rp, Pearson correlation coefficient; RMSE, root-mean-square error; SAM, S-adenosyl methionine; UMAP, Uniform Manifold Approximation and Projection.

REFERENCES

(1) Mobley, D. L.; Gilson, M. K. Predicting Binding Free Energies: Frontiers and Benchmarks. *Annu. Rev. Biophys.* **2017**, *46*, 531−558.

(2) 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*, 224−229.

(3) Gorgulla, C.; Boeszoermenyi, A.; Wang, Z. F.; Fischer, P. D.; Coote, P. W.; Padmanabha Das, K. M.; Malets, Y. S.; Radchenko, D. S.; Moroz, Y. S.; Scott, D. A.; Fackeldey, K.; Hoffmann, M.; Iavniuk, I.; Wagner, G.; Arthanari, H. An Open-Source Drug Discovery Platform Enables Ultra-Large Virtual Screens. *Nature* **2020**, *580*, 663−668.

(4) Hoffmann, T.; Gastreich, M. The Next Level in Chemical Space Navigation: Going Far Beyond Enumerable Compound Libraries. *Drug Discovery Today* **2019**, *24*, 1148−1156.

(5) Burley, S. K.; Bhikadiya, C.; Bi, C.; Bittrich, S.; Chen, L.; Crichlow, G. V.; Duarte, J. M.; Dutta, S.; Fayazi, M.; Feng, Z.; Flatt, J. W.; Ganesan, S. J.; Goodsell, D. S.; Ghosh, S.; Kramer Green, R.; Guranovic, V.; Henry, J.; Hudson, B. P.; Lawson, C. L.; Liang, Y.; Lowe, R.; Peisach, E.; Persikova, I.; Piehl, D. W.; Rose, Y.; Sali, A.; Segura, J.; Sekharan, M.; Shao, C.; Vallat, B.; Voigt, M.; Westbrook, J. D.; Whetstone, S.; Young, J. Y.; Zardecki, C. RCSB Protein Data Bank: Celebrating 50 Years of the PDB with New Tools for Understanding and Visualizing Biological Macromolecules in 3D. *Protein Sci.* **2022**, *31*, 187−208.

(6) Kollman, P. A. Free Energy Calculations: Applications to Chemical and Biochemical Phenomena. *Chem. Rev.* **1993**, *93*, 2395−2417.

(7) Guedes, I. A.; Pereira, F. S. S.; Dardenne, L. E. Empirical Scoring Functions for Structure-Based Virtual Screening: Applications, Critical Aspects, and Challenges. *Front. Pharmacol.* **2018**, *9*, 1089.

(8) Ain, Q. U.; Aleksandrova, A.; Roessler, F. D.; Ballester, P. J. Machine-Learning Scoring Functions to Improve Structure-Based Binding Affinity Prediction and Virtual Screening. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2015**, *5*, 405−424.

(9) Kim, J.; Park, S.; Min, D.; Kim, W. Y. Comprehensive Survey of Recent Drug Discovery Using Deep Learning. *Int. J. Mol. Sci.* **2021**, *22*, 9983.

(10) Kimber, T. B.; Chen, Y.; Volkamer, A. Deep Learning in Virtual Screening: Recent Applications and Developments. *Int. J. Mol. Sci.* **2021**, *22*, 4435.

(11) Cang, Z.; Wei, G. W. TopologyNet: Topology Based Deep Convolutional and Multi-Task Neural Networks for Biomolecular Property Predictions. *PLoS Comput. Biol.* **2017**, *13*, No. e1005690.

(12) Gomes, J.; Ramsundar, B.; Feinberg, E. N.; Pande, V. S. *Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity*, 2017, arXiv:1703.10603 (accessed March 12, 2022).

(13) Lau, T.; Dror, R. *Brendan-A Deep Convolutional Network for Representing Latent Features of Protein-Ligand Binding Poses*, 2017, http://cs231n.stanford.edu/reports/2017/pdfs/2531.pdf (accessed March 12, 2022).

(14) 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*, 1520−1530.

(15) Jimenez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. KDEEP: Protein-Ligand Absolute Binding Affinity Prediction Via 3D-Convolutional Neural Networks. *J. Chem. Inf. Model.* **2018**, *58*, 287−296.

(16) 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*, 3666−3674.

(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*, 513−530.

(18) Li, Y.; Rezaei, M.; Li, C.; Li, X.; Wu, D. *DeepAtom: A Framework for Protein-Ligand Bidning Affinity Prediction*, 2019, arXiv:1912.00318v1.

(19) Lim, J.; Ryu, S.; Park, K.; Choe, Y. J.; Ham, J.; Kim, W. Y. Predicting Drug-Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. *J. Chem. Inf. Model.* **2019**, *59*, 3981−3988.

(20) Torng, W.; Altman, R. B. Graph Convolutional Neural Networks for Predicting Drug-Target Interactions. *J. Chem. Inf. Model.* **2019**, *59*, 4131−4149.

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

(21) Zhang, H.; Liao, L.; Saravanan, K. M.; Yin, P.; Wei, Y. DeepBindRG: A Deep Learning Based Method for Estimating Effective Protein-Ligand Affinity. *PeerJ* **2019**, *7*, No. e7362.
(22) 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*, 15956−15965.
(23) Hassan-Harrirou, H.; Zhang, C.; Lemmin, T. RoseNet: Improving Binding Affinity Prediction by Leveraging Molecular Mechanics Energies with an Ensemble of 3D Convolutional Neural Networks. *J. Chem. Inf. Model.* **2020**, *60*, 2791−2802.
(24) Karlov, D. S.; Sosnin, S.; Fedorov, M. V.; Popov, P. GraphDelta: MPNN Scoring Function for the Affinity Prediction of Protein-Ligand Complexes. *ACS Omega* **2020**, *5*, 5150−5159.
(25) Kwon, Y.; Shin, W. H.; Ko, J.; Lee, J. AK-Score: Accurate Protein-Ligand Binding Affinity Prediction Using an Ensemble of 3D-Convolutional Neural Networks. *Int. J. Mol. Sci.* **2020**, *21*, 8424.
(26) Li, S.; Wan, F.; Shu, H.; Jiang, T.; Zhao, S.; Zeng, J. MONN: A Multi-Objetcive Neural Network for Predicting Compound-Protein Interactions and Affinities. *Cell Syst.* **2020**, *10*, 308−322.
(27) Wang, S.; Liu, D.; Ding, M.; Du, Z.; Zhong, Y.; Song, T.; Zhu, J.; Zhao, R. SE-OnionNet: A Convolution Neural Network for Protein-Ligand Binding Affinity Prediction. *Front. Genet.* **2020**, *11*, No. 607824.
(28) Xie, L.; Xu, L.; Chang, S.; Xu, X.; Meng, L. Multitask Deep Networks with Grid Featurization Achieve Improved Scoring Performance for Protein-Ligand Binding. *Chem. Biol. Drug Des.* **2020**, *96*, 973−983.
(29) Yang, J.; Shen, C.; Huang, N. Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets. *Front. Pharmacol.* **2020**, *11*, 69.
(30) Zhu, F.; Zhang, X.; Allen, J. E.; Jones, D.; Lightstone, F. C. Binding Affinity Prediction by Pairwise Function Based on Neural Network. *J. Chem. Inf. Model.* **2020**, *60*, 2766−2772.
(31) Ahmed, A.; Mam, B.; Sowdhamini, R. DEELIG: A Deep Learning Approach to Predict Protein-Ligand Binding Affinity. *Bioinf. Biol. Insights* **2021**, *15*, No. 11779322211030364.
(32) Jiang, D.; Hsieh, C. Y.; Wu, Z.; Kang, Y.; Wang, J.; Wang, E.; Liao, B.; Shen, C.; Xu, L.; Wu, J.; Cao, D.; Hou, T. InteractionGraphNet: A Novel and Efficient Deep Graph Representation Learning Framework for Accurate Protein-Ligand Interaction Predictions. *J. Med. Chem.* **2021**, *64*, 18209−18232.
(33) Jones, D.; Kim, H.; Zhang, X.; Zemla, A.; Stevenson, G.; Bennett, W. F. D.; Kirshner, D.; Wong, S. E.; Lightstone, F. C.; Allen, J. E. Improved Protein-Ligand Binding Affinity Prediction with Structure-Based Deep Fusion Inference. *J. Chem. Inf. Model.* **2021**, *61*, 1583−1592.
(34) Kumar, S.; Kim, M. H. SMPLIP-Score: Predicting Ligand Binding Affinity from Simple and Interpretable on-the-Fly Interaction Fingerprint Pattern Descriptors. *J. Cheminf.* **2021**, *13*, 28.
(35) Liu, Q.; Wang, P. S.; Zhu, C.; Gaines, B. B.; Zhu, T.; Bi, J.; Song, M. OctSurf: Efficient Hierarchical Voxel-Based Molecular Surface Representation for Protein-Ligand Affinity Prediction. *J. Mol. Graphics Modell.* **2021**, *105*, No. 107865.
(36) Seo, S.; Choi, J.; Park, S.; Ahn, J. Binding Affinity Prediction for Protein-Ligand Complex Using Deep Attention Mechanism Based on Intermolecular Interactions. *BMC Bioinf.* **2021**, *22*, 542.
(37) Shen, H.; Zhang, Y.; Zheng, C.; Wang, B.; Chen, P. A Cascade Graph Convolutional Network for Predicting Protein-Ligand Binding Affinity. *Int. J. Mol. Sci.* **2021**, *22*, 4023.
(38) Son, J.; Kim, D. Development of a Graph Convolutional Neural Network Model for Efficient Prediction of Protein-Ligand Binding Affinities. *PLoS One* **2021**, *16*, No. e0249404.
(39) Xiong, J.; Xiong, Z.; Chen, K.; Jiang, H.; Zheng, M. Graph Neural Networks for Automated De Novo Drug Design. *Drug Discovery Today* **2021**, *26*, 1382−1393.
(40) 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*, 2977−2980.
(41) PDBbind, http://www.pdbbind.org.cn/ (accessed 2022-03-12).
(42) Wang, J.; Dokholyan, N. V. Yuel: Improving the Generalizability of Structure-Free Compound-Protein Interaction Prediction. *J. Chem. Inf. Model.* **2022**, *62*, 463−471.
(43) Gabel, J.; Desaphy, J.; Rognan, D. Beware of Machine Learning-Based Scoring Functions-on the Danger of Developing Black Boxes. *J. Chem. Inf. Model.* **2014**, *54*, 2807−2815.
(44) Sieg, J.; Flachsenberg, F.; Rarey, M. In Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2019**, *59*, 947−961.
(45) Shen, C.; Hu, Y.; Wang, Z.; Zhang, X.; Pang, J.; Wang, G.; Zhong, H.; Xu, L.; Cao, D.; Hou, T. Beware of the Generic Machine Learning-Based Scoring Functions in Structure-Based Virtual Screening. *Briefings Bioinf.* **2021**, *22*, No. bbaa070.
(46) Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C. J.; Duca, J. S.; Hornak, V.; Koes, D. R.; Kurtzman, T. Hidden Bias in the DUD-E Dataset Leads to Misleading Performance of Deep Learning in Structure-Based Virtual Screening. *PLoS One* **2019**, *14*, No. e0220113.
(47) Ozturk, H.; Ozgur, A.; Ozkirimli, E. DeepDTA: Deep Drug-Target Binding Affinity Prediction. *Bioinformatics* **2018**, *34*, i821−i829.
(48) Nguyen, T.; Le, H.; Quinn, T. P.; Nguyen, T.; Le, T. D.; Venkatesh, S. GraphDTA: Predicting Drug-Target Binding Affinity with Graph Neural Networks. *Bioinformatics* **2021**, *37*, 1140−1147.
(49) Schmitt, S.; Kuhn, D.; Klebe, G. A New Method to Detect Related Function among Proteins Independent of Sequence and Fold Homology. *J. Mol. Biol.* **2002**, *323*, 387−406.
(50) Da Silva, F.; Desaphy, J.; Rognan, D. IChem: A Versatile Toolkit for Detecting, Comparing, and Predicting Protein-Ligand Interactions. *ChemMedChem* **2018**, *13*, 507−510.
(51) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. *Neural Message Passing for Quantum Chemistry*, 2017, arXiv:1704.01212
(52) Martin, E. J.; Polyakov, V. R.; Zhu, X. W.; Tian, L.; Mukherjee, P.; Liu, X. All-Assay-Max2 PQSAR: Activity Predictions as Accurate as Four-Concentration IC50s for 8558 Novartis Assays. *J. Chem. Inf. Model.* **2019**, *59*, 4450−4459.
(53) Wallach, I.; Heifets, A. Most Ligand-Based Classification Benchmarks Reward Memorization Rather Than Generalization. *J. Chem. Inf. Model.* **2018**, *58*, 916−932.
(54) Tran-Nguyen, V. K.; Bret, G.; Rognan, D. True Accuracy of Fast Scoring Functions to Predict High-Throughput Screening Data from Docking Poses: The Simpler the Better. *J. Chem. Inf. Model.* **2021**, *61*, 2788−2797.
(55) Marcou, G.; Rognan, D. Optimizing Fragment and Scaffold Docking by Use of Molecular Interaction Fingerprints. *J. Chem. Inf. Model.* **2007**, *47*, 195−207.
(56) Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; Zarrinkar, P. P. Comprehensive Analysis of Kinase Inhibitor Selectivity. *Nat. Biotechnol.* **2011**, *29*, 1046−1051.
(57) Metz, J. T.; Johnson, E. F.; Soni, N. B.; Merta, P. J.; Kifle, L.; Hajduk, P. J. Navigating the Kinome. *Nat. Chem. Biol.* **2011**, *7*, 200−202.
(58) Protein Data Bank, https://www.rcsb.org (accessed 2022-12-03).
(59) Bietz, S.; Urbaczek, S.; Schulz, B.; Rarey, M. Protoss: A Holistic Approach to Predict Tautomers and Protonation States in Protein-Ligand Complexes. *Aust. J. Chem.* **2014**, *6*, 12.
(60) McInnes, L.; Healy, J.; Melville, J., Umap: Uniform Manifold Approximation and Projection for Dimension Reduction., 2021, arXiv:1802.03426v3.
(61) Networkx- Network Analysis in Python, https://networkx.org (accessed 2022-03-12).
(62) Pytorch, https://pytorch.org/ (accessed 2022-03-12).
(63) Lightning, https://www.pytorchlightning.ai/ (accessed 2022-03-12).

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

(64) Deep Graph Library, https://www.dgl.ai/ (accessed 2022-03-12).
(65) Pafnucy Website, https://gitlab.com/cheminfibb/pafnucy (accessed 2022-12-03).
(66) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

https://doi.org/10.1021/acs.jmedchem.2c00487
J. Med. Chem. XXXX, XXX, XXX−XXX

2.7 Supplementary materials

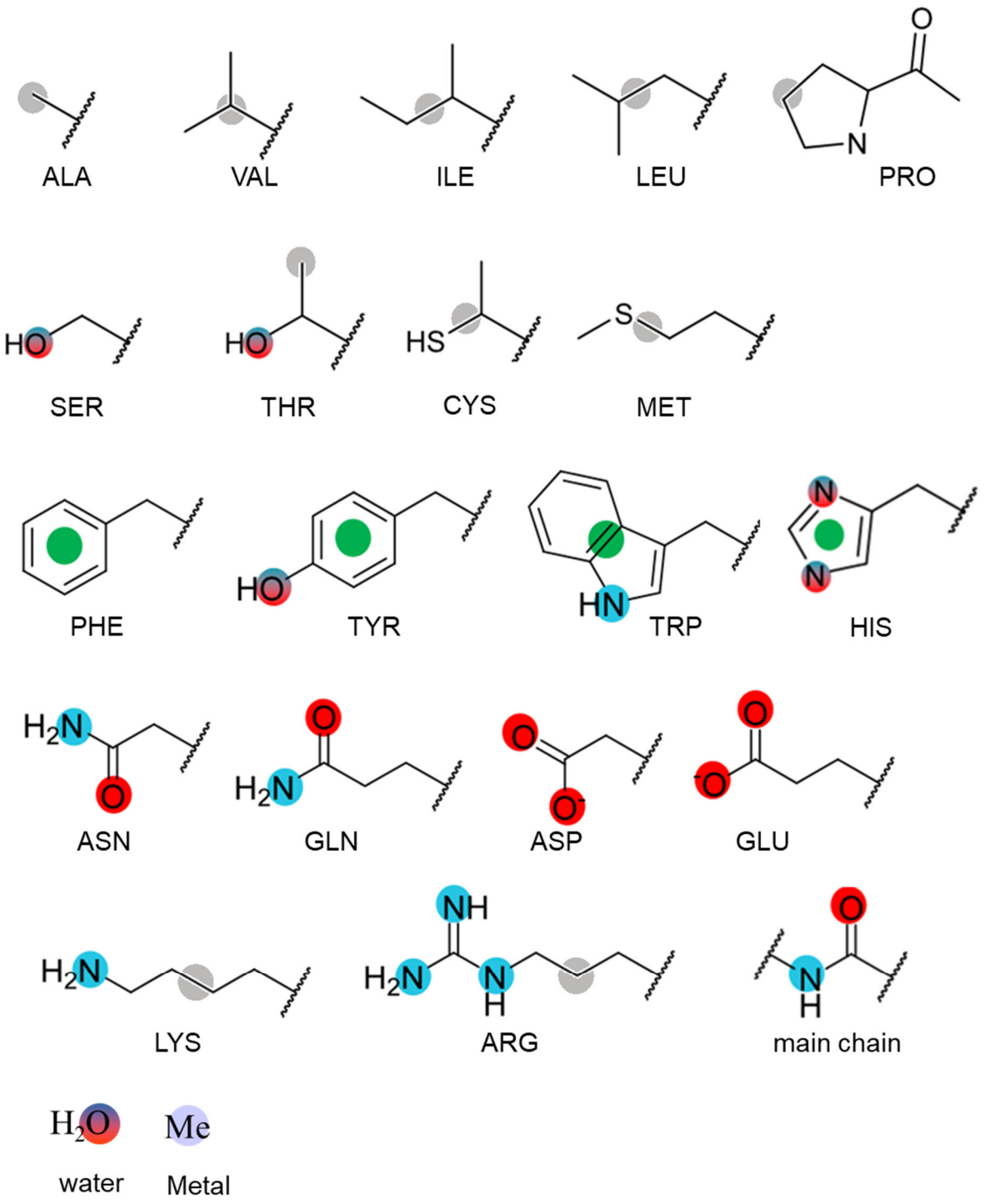


Figure S1. Location and pharmacophoric properties of protein pseudoatoms (grey, aliphatic; red, hydrogen-bond acceptor; cyan, hydrogen-bond donor; green, aromatic; metal-chelating, steel blue)

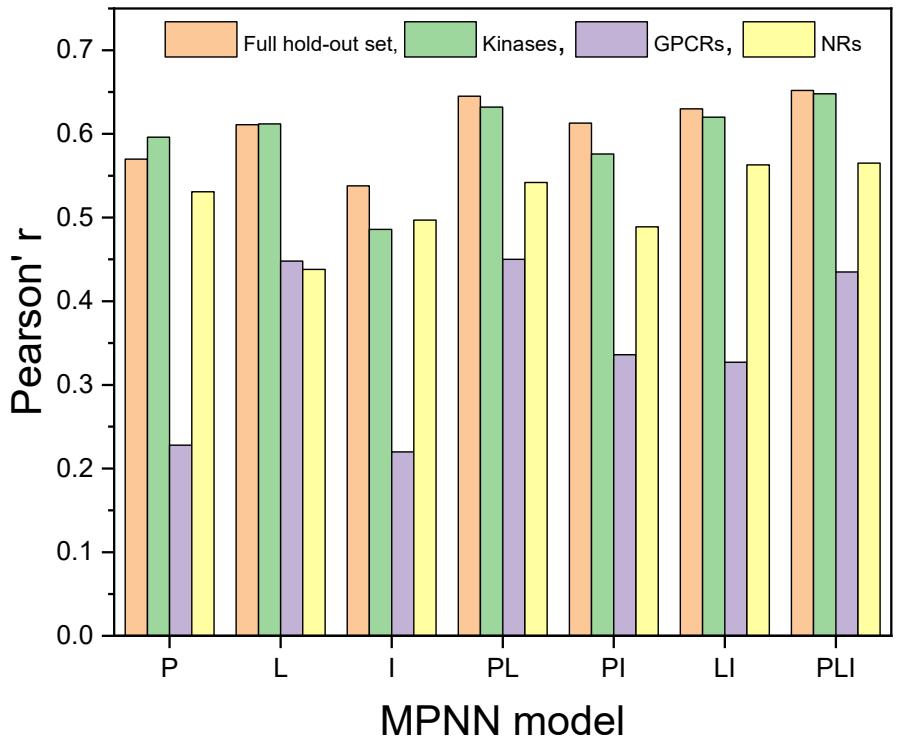


Figure S2. Performance of modular MPNN models in predicting affinities for specific target classes of the 2019 hold-out set. Mapping of protein target classes (GPCRs, G-protein-coupled receptor; NRs, Nuclear hormone receptors) from the Pharos database (https://pharos.nih.gov/) to PDB entries was performed using the Pharos-to-PDB code (https://github.com/ravila4/Pharos-to-PDB).

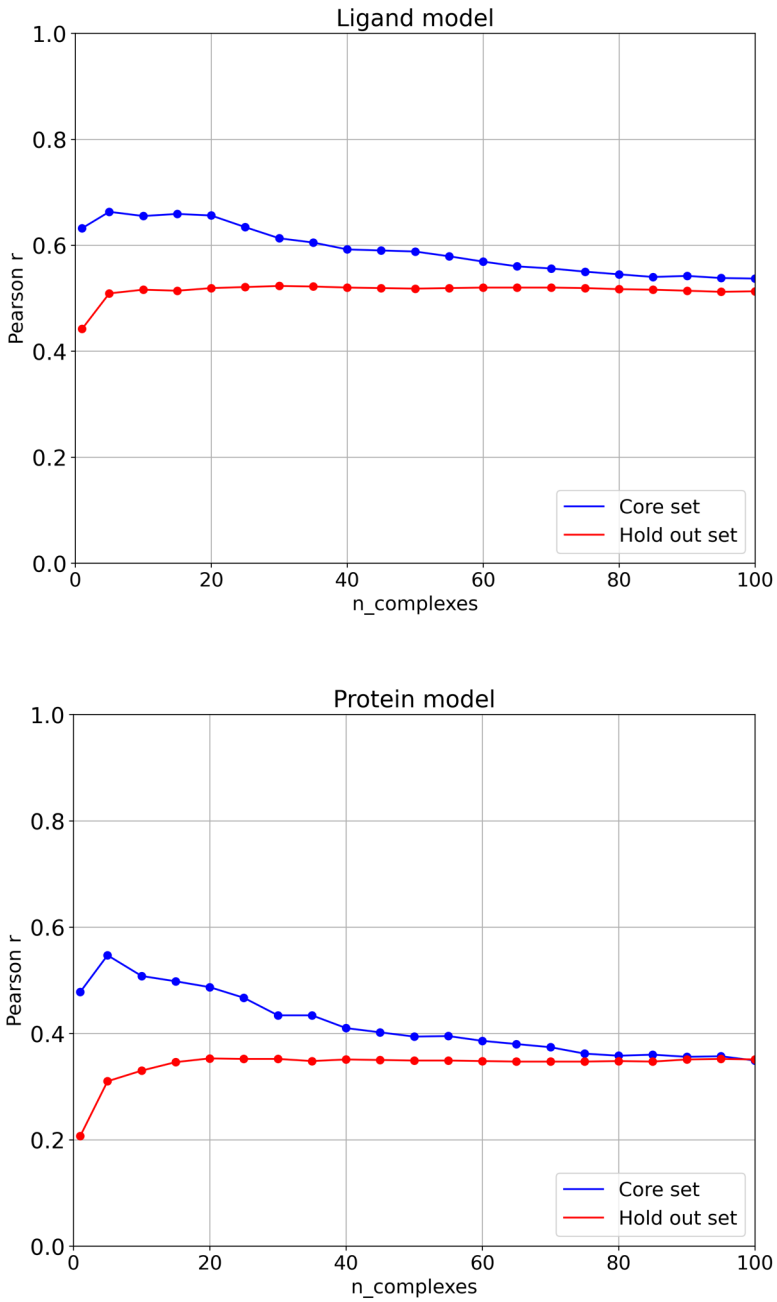
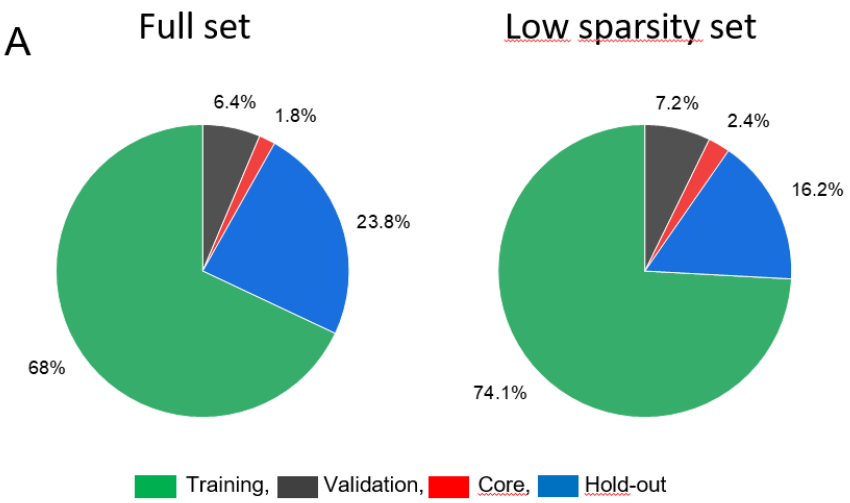


Figure S3. Influence of the number of closest ligands or proteins used to average binding affinities in the performance of simple memorization models, assessed by the Pearson r correlation coefficient.



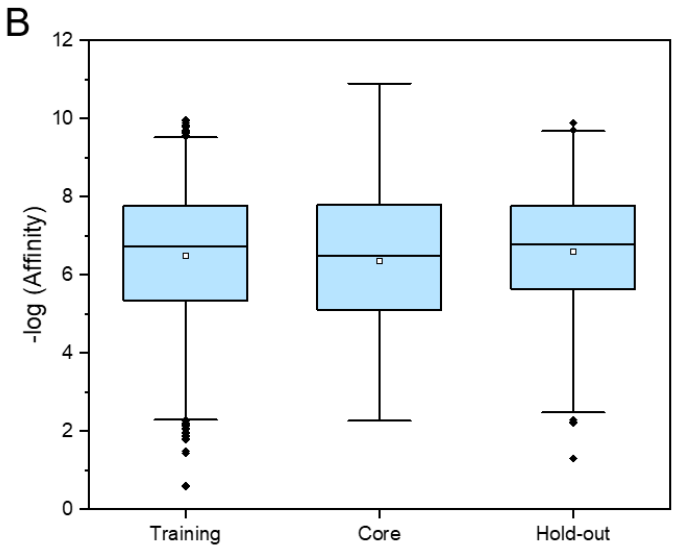


Figure S4. PDBbind low sparsity subset. **A)** Split into four subsets for training, validation and external test sets (core 2016, hold-out 2019), **B)** Distribution of experimental affinities for the training (n=1,505), core (n=49) and hold-out sets (n=329). The boxes delimit the 25th and 75th percentiles, and the whiskers delimit the 1st and 99th percentiles. The median and mean values are indicated by a horizontal line and a filled square in the box, respectively. Outliers are indicated by a diamond.

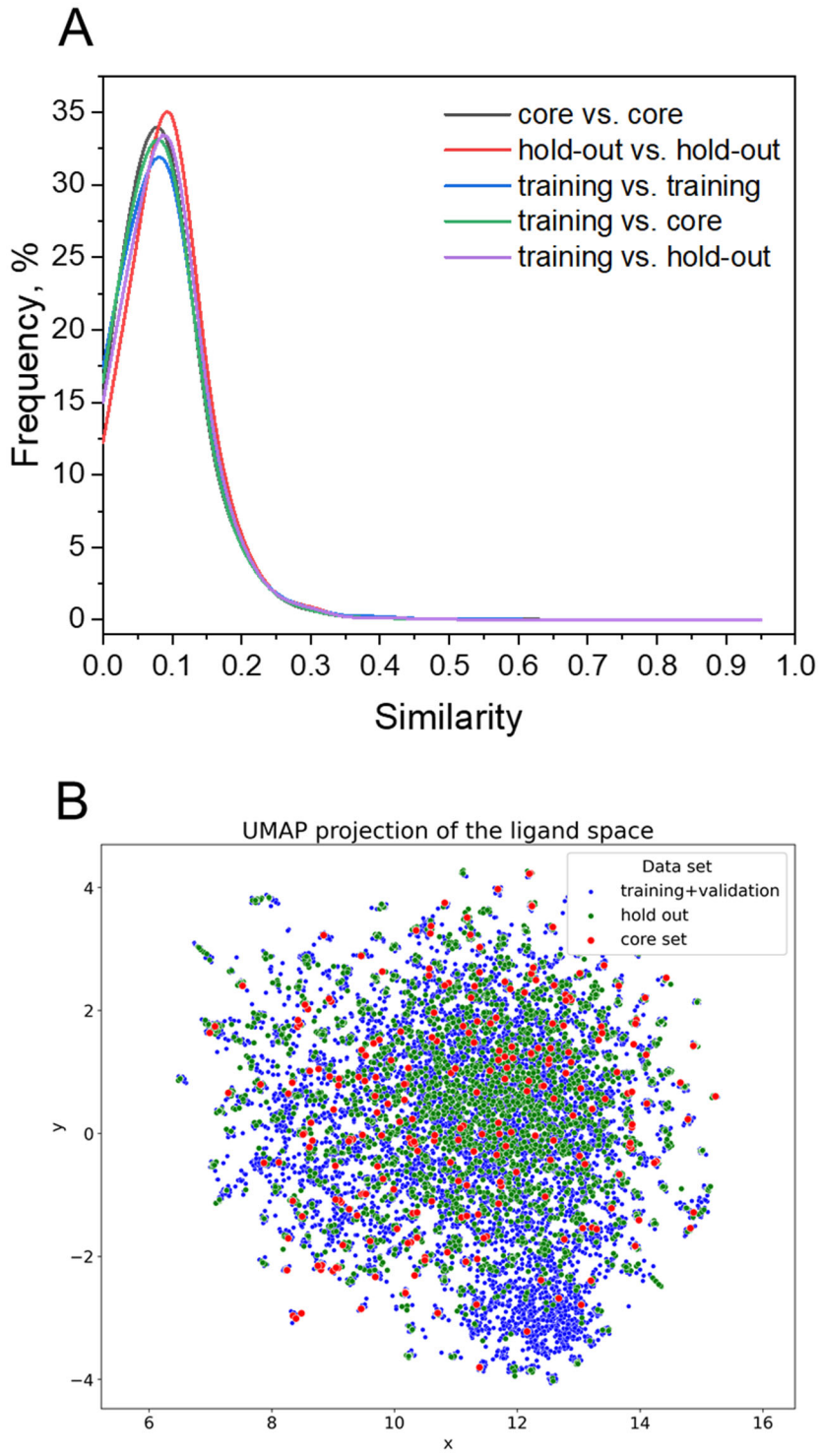


Figure S5. Chemical diversity of PDBbind ligands. **A)** Pairwise similarity of Murcko superstructures for ligands from the training, 2016 core and 2019 hold-out sets. The similarity is expressed by the Tanimoto coefficient computed from ECFP4 fingerprints. **B)** Uniform Manifold Approximation and Projection (UMAP)of PDBbind ligands, performed in umap-learn 0.5.3 with a number of neighbours of 30 and a dice distance metric. The Morgan fingerprints of radius 2 (nBits=1024) were computed in rdkit v.2020.09.1.

Table S1. Structure-based deep neural networks to predict protein-ligand binding affinities.

Model	Type	Objects	Descriptor	Training set	Test set	Split	R*p*	RMSE	Reference
TNet-BP	CNN	PL	Topological fingerprints	PDBbind 2016 refined (n=3767)	PDBbind 2016 core (n=290)	PDBBind original	0.826	1.37	11
ACNN	CNN	P, L, PL	Atom type-labelled distances (Nat*25 atom types*12 closest neighbors)	PDBbind 2015 refined (n=2965)	PDBbind 2015 refined (n=741)	Temporal	0.727	-	12
Brendan	CNN	PL	3D grid (21*21*21 Å) * 256 bit SPLIF vector	PDBbind 2016 general (10000)	PDBbind 2016 general (1500)	Random	0.704	-	13
PotentialNet	GNN	P, L, PL	Protein-ligand graph	PDBbind 2007 refined (n=1095)	PDBbind 2007 core (n=195)	PDBBind original	0.822	1.39	14
K_{DEEP}	CNN	L, PL	1-Å 3D grid (25*25*25 Å) * 16 features	PDBbind 2016 refined (n=3767)	PDBbind 2016 core (n=290)	PDBbind original	0.820	1.27	15
Pafnucy	CNN	L, PL	1-Å 3D grid (21*21*21 Å) * 19 features	PDBbind 2016 general (11906)	PDBbind 2016 core (n=290)	PDBBind original	0.780	1.42	16
DeepATom	CNN	PL	1-Å 3D grid (25*25*25 Å) * 24 features	PDBbind 2016 refined (n=3390)	PDBbind 2016 core (n=290)	PDBbind original	0.807	1.32	18
DeepBindRG	CNN	PL	ligand (84) * protein (41) atom pair distances < 4 Å	PDBbind 2018 general (n=13500)	PDBbind 2018 general (n=925)	Random	0.593	1.50	21
OnionNet	CNN	PL	Atom type-labelled distances (Nat*25 atom types*12 closest neighbors)	PDBbind 2016 general (n=11906)	PDBbind 2016 core (n=290)	PDBbind original	0.816	1.28	22
RosENet	CNN	PL	voxelized Rosetta interaction energies + pharmacophoric descriptors	PDBbind 2016/2018 refined (n=4463)	PDBbind 2016 core (n=290)	PDBBind original	0.820	1.24	23
graphDelta	GNN	L, PL	One-hot encoded ligand atoms + protein environmental descriptors (373)	PDBbind 2018 general (n=8766)	PDBbind 2016 core (n=285)	PDBbind original	0.870	1.05	24
AK-Score	CNN	PL	id Kdeep	PDBbind 2016 refined (n=3772)	PDBbind 2016 core (n=285)	PDBBind original	0.827	1.22	25
SE-OnionNet	CNN	PL	1-Å grid (21*21*21)* 64 protein-ligand element distance counts	PDBbind 2018 general (n=11663)	PDBbind 2018 refined (n=463)	Random	0.853	1.59	27
Progessive multitask network	CNN	P, L, PL	ligand ECFP + Protein ECFP + Protein-Ligand SPLIF	PDBbind 2016 refined (n=3568)	PDBbind 2016 core (n=290)	PDBbind original	0.740	0.98	28
ACNN	CNN	P, L, PL	Atom type-labelled distances (Nat*25 atom types*12 closest neighbors)	PDBbind 2015 refined (n=3706)	PDBbind 2015 core (n=195)	PDBBind original	0.730	-	29
Pair	CNN	PL	protein-ligand distance pairs	PDBBbind 2018 refined (n=2675)	PDBBbind 2018 refined (n=891)	Random split	0.660	1.61	30
DEELIG	CNN	L, PL	Atomic model: 3D grid (10*10*10 Å) * 19 bits (atomic model); Composite model: 3D grid (10*10*10 Å) * 44 bits (pocket) + 14716 bits (ligand)	in-house set (n=4041)	PDBbind 2016 core (n=290)	Random 80/10/10	0.889	-	31
Interaction GraphNet	GNN	P, L, PL	independent GNN for intra and inter-molecular interactions	PDBbind 2016 general (n=10366)	PDBbind 2016 core (n=290)	PDBBind original	0.837	1.22	32

midlevel fusion	CNN+GNN	PL	CNN: 1-Å grid (48*48*48)* 19 atomic features; GNN: covalent (d < 1.5 Å) and non-covalent edges (1.5 < d < 4.5 Å)	Pdbbind 2016 general+refined (13283)	PDB2016 core set (n=290)	PDBBind original	0.810	1.31	33
SMPLIP	RF+ CNN	L, PL	IFP (140) + interaction distances (140) + SMF descriptors (2282)	Pdbbind 2016 general+refined (13283)	PDB2016 core set (n=290)	PDBBind original	0.770	1.51	34
OctSurf	CNN	PL	1-Å 3D grid (64*64*64 Å) * 24 features/octant	PDBbind 2018 general (n=16126)	PDBbind 2016 core (n=285)	PDBBind original	0.793	1.45	35
BAPA	CNN	PL	Protein-ligand interaction descriptors + 6 Vina terms	PDBbind 2016 refined (n=3689)	PDBbind 2016 core (n=285)	PDBBind original	0.819	1.31	36
APMNet	GNN+GNN	P, L	75 DeepChem atomic features	PDBbind 2016 general (n=11844)	PDBbind 2016 core (n=290)	PDBBind original	0.815	1.27	37
GraphBAR	GNN	PL	13 features * 200 protein-ligand atoms	PDBbind 2016 general (n=11146)	PDBbind 2016 core (n=290)	PDBBind original	0.764	1.44	38

Table S2. Geometric rules to define protein-ligand non-covalent interactions.

Interaction	Rule 1[a]	Rule 2[b]
H-bond	$\left\|\overrightarrow{DA}\right\| \le 3.5\mathring{A}$	$\left\langle \overrightarrow{DH}, \overrightarrow{HA} \right\rangle \in \left[\frac{-\pi}{4}, \frac{\pi}{4} \right]$
Ionic	$\left\|\overrightarrow{+-}\right\| \le 4.0\mathring{A}$	
Hydrophobe	$\left\|\overrightarrow{Y_1 Y_2}\right\| \le 4.5\mathring{A}$	
Aromatic (Face to face)	$\left\|\overrightarrow{ac_1 ac_2}\right\| \le 4.0\mathring{A}$	$\left\langle \overrightarrow{n_1}, \overrightarrow{n_2} \right\rangle \in \left[\frac{-\pi}{6}, \frac{\pi}{6} \right]$
Aromatic (Edge to face)	$\left\|\overrightarrow{ac_1 ac_2}\right\| \le 4.0\mathring{A}$	$\left\langle \overrightarrow{n_1}, \overrightarrow{n_2} \right\rangle \in \left[\frac{\pi}{6}, \frac{5\pi}{6} \right]$
pi-cation	$\left\|\overrightarrow{ac+}\right\| \le 4.0\mathring{A}$	$\left\langle \overrightarrow{n}, \overrightarrow{ac+} \right\rangle \in \left[\frac{-\pi}{6}, \frac{\pi}{6} \right]$
Metal	$\left\|\overrightarrow{MA}\right\| \le 2.8\mathring{A}$	

[a] D: H-bond donor; A: H-bond acceptor; +: cation; -:anion; Y: hydrophobe; ac: geometric center of an aromatic ring; M: metal.

[b] H: hydrogen; n: normal to the aromatic ring.

CHAPTER 3

Binding affinity prediction from docking poses

3.1 Introduction

One major challenge of structure-based drug design is the ability to predict, preferably at a high throughput, the affinity (absolute binding free energy) of small molecules to macromolecular targets (proteins, nucleic acids) from their corresponding three-dimensional (3D) structures [1]. To achieve this goal, a mathematical scoring function [2] is asked to iteratively solve three related but slightly different problems: (i) discriminating native from irrelevant binding modes (docking accuracy), (ii) predicting the absolute binding free energy of the selected solution (scoring accuracy), (iii) discriminate true from false binders upon virtually screening a compound library (screening accuracy). Despite the advent of many benchmarking initiatives involving hundreds of research groups [2–9], and the development of orthogonal approaches (empirical, force-fields, potential of mean forces, machine learning) [10] , we must admit that no robust solution has yet been found to this crucial issue. At best, binding free energy differences of congeneric compounds may be reproduced satisfactorily with free energy perturbation (FEP) methods [11], at the cost of a computational burden that is unfortunately incompatible with high-throughput screening of large compound libraries. This bitter statement is even more frustrating at the light of spectacular developments in structural biology [12] and automated synthesis of drug-like compounds from multibillion compounds spaces [13] opening novel avenues in computer-driven drug discovery.Unreasonable hopes have been raised the last decade with the application of machine learning (ML) approaches [14] ranging from simplistic Random Forest modelling of protein-ligand atom pair distributions [15] to physics-informed graph neural networks [16]. In brief, each of the above-cited three problems (docking, scoring, screening) have found suitable solutions. Predicting the binding mode of a small molecular compound to a protein can be achieved with accuracies up to 90% but the same models cannot rank compounds by decreasing affinities [17] . Predicting affinities from protein-ligand X-ray structures is achievable within 1.5 pK unit but does not generalize to unseen complexes (other X-ray/cryo-EM structures, 3D models) and cannot therefore be used for virtual hit identification [18] . Since, the affinity of a ligand to its target protein is usually known far before solving the corresponding 3D structure, such scoring functions are of very limited use in daily drug design scenarios. Binding

affinity and binding mode predictions are two sides of the same problem. The fact that a single model cannot solve simultaneously both issues [17, 19] is an explicit acknowledgement of weaknesses due to many sources. Some are easily preventable, e.g. the lack of self-criticism in the interpretation of results (just focusing on statistics and not on their physicochemical meaning [20]), the frequently observed disconnection between academic studies and real drug design needs [21], the lack of scientific rigor, or the blind usage of datasets exhibiting major biases [22–26]. Some other are harder to solve, for example the unavailability of sufficiently dense and diverse protein-ligand matrices of known affinities and structures [18].

From the simple ascertainment that docking poses are not X-ray structures (nor zebra are horses), we reasoned that predicting binding affinities from machine learning models should be done on the same molecular objects (docking poses) than those to which they are supposed to be applied next. This approach enables a clear augmentation of input data (there are many more docking poses than X-ray structures) in favor of machine learning, but requires first to select the "good" docking poses for a proper labeling of input instances. In other words a first ML model dedicated to predict the suitability of a binding pose should be applied prior to a binding affinity regressor. If ML/DNN models are indeed able to discriminate with 80-90% accuracies good from erroneous docking poses [17] , obtained results on binding affinity prediction and virtual screening accuracies remain controversial. Whereas some deep neural networks (DNNs) supplemented with docking poses exhibit indeed better docking, ranking or virtual screening accuracies [16, 27–29] , some others reported the opposite behavior [30–32] or no significant effects [33]. A fair analysis of these studies is impossible because of the different level of noise and biases that significantly impact observed results. Hence, many studies report the use of datasets (DUD [34] , DUDE-E [35], PDBbind [36]) with significant biases [33], notably stemming from too close chemical neighborhoods between training and test samples [20] . Therefore, any ML model explicitly describing test proteins and ligands at the atomic level will memorize the later information more than really learning protein-ligand interactions [18] . This observation explains why adding ligand descriptors to protein-ligand interaction feature helps in better predicting binding affinities [32].

It is therefore still unknown whether truly unbiased affinity prediction ML models trained on X-ray structures may be applied to docking poses or if docking pose-specific models are required for this task. If yes, would the accuracy of these predictions be higher and would docking-based models better generalize to unseen complexes?

To address the above questions, we here report a follow-up study of a previous contribution [18] reporting that sophisticated message-passing graph neural networks (MPNNs) aimed at predicting binding affinity from protein-ligand X-ray structures mostly memorize but do not learn protein-ligand interactions, because of severe biases arising from simple protein and ligand descriptors. Moreover, we reported a temporal splitting procedure of the PDBbind gold standard set, enabling to test MPNNs on a hold-out set of 3,386 entries that is much larger and diverse than the commonly used core set of 290 entries [18]. Herein, we investigate the usage of ligand and protein-debiased DNN architectures fed with molecular objects (X-ray structures, docking poses) and check whether such models may be applied to objects they have not been trained on. We further verified if augmenting input instances by switching X-ray poses to docking poses leads to better binding affinity predictions and wider generalization.

3.2 Results and discussion

A temporal splitting procedure of the PDBbind set enables to define a more realistic hold-out set for challenging ML models.

The PDBbind dataset [36] of protein-ligand complexes of known structure and affinity has become a gold standard for predicting affinities from 3D structures, training on either the general or the refined set and testing on a small and diverse external core set. Many studies reported that the core set is a far too easy external test set since alternative splitting protocols based on similarity clustering (protein sequence or ligand scaffold) significantly reduced the models' performance [26, 32] . We therefore recently proposed a novel temporal splitting procedure [18] in which the model is trained on data released until Dec. 31st 2016 and tested on a novel hold-out set consisting in 3,386 entries released from year 2017 on (Figure 3.1).

A Venn diagram analysis of shared ligands and proteins between the three sets

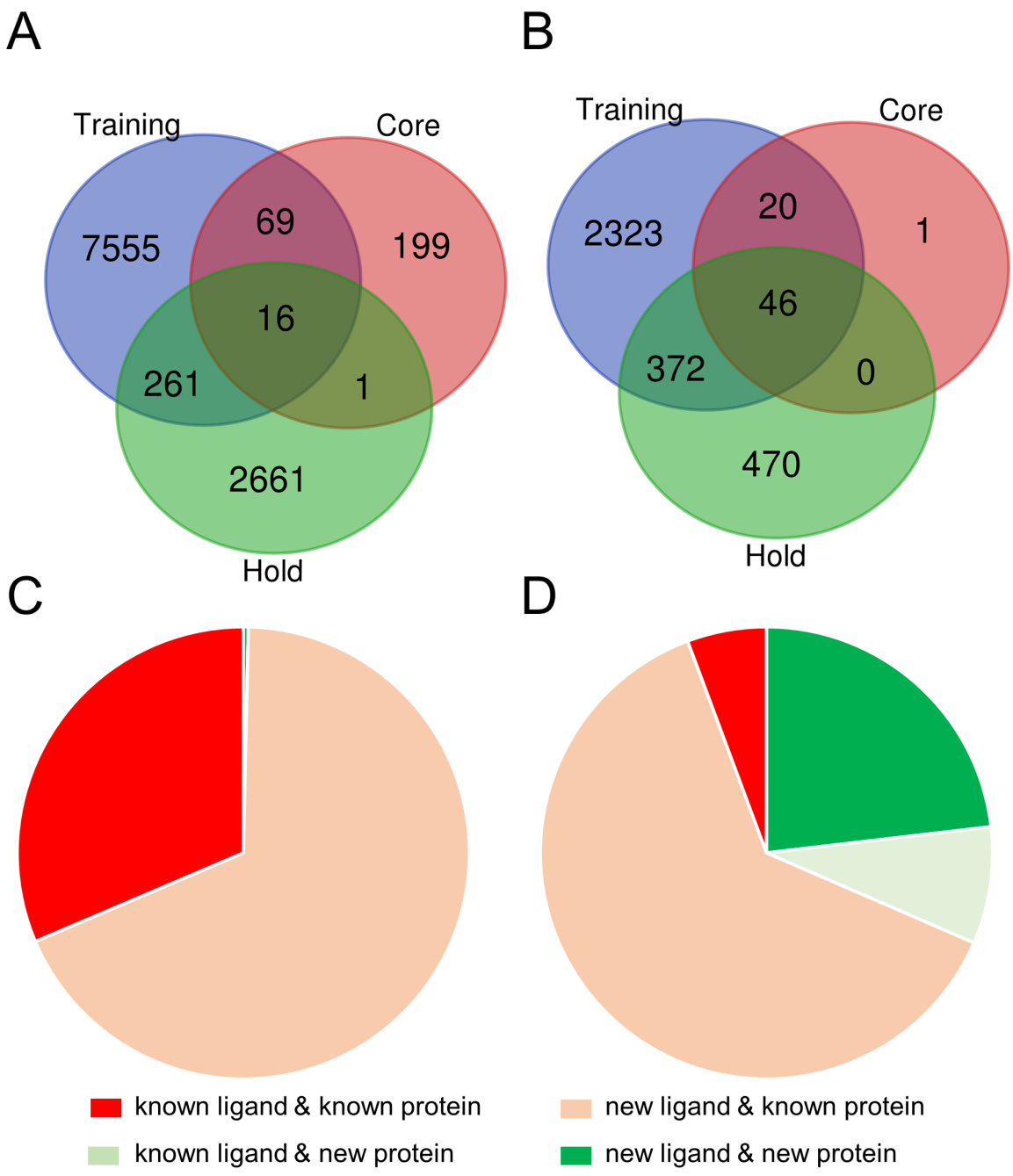


Figure 3.1 Overlap between core (287 entries), hold-out (3,386 entries) and training-validation sets (11,820 entries). A) Count of shared ligands, B) Count of shared proteins, C) Core set vs. training set composition, D) Hold-out set vs. training set composition.

(Figure 3.1) clearly shows that the herein proposed hold-out set if much more diverse in terms of ligand (PDB heteromer three-letter code) and protein (SwissProt accession number) composition, than the core set used in all previous studies. The hold-out set features more realistic scenarios in which new data have to be predicted from the past: i) both the protein and the ligand are unknown to the model (hit identification), ii) only the ligand is unknown (hit to lead optimization), iii) only the protein is unknown (hit profiling), iv) neither the protein nor the ligand are unknown (drug repurposing). We therefore believe that predictions on the novel hold-out set will be much more representative of the true generalizability of ML models.

Setting-up a data set of docking poses of varying accuracy by self-docking PDBbind ligands into their cognate targets.

In order to generate a dataset of possible binding modes of different correctness, we redocked each PDBbind ligand to its corresponding protein target using three state-of-the art docking utilities relying on different sampling algorithms, namely PLANTS (ant colony optimization) [37] , Surflex (incremental construction) [38] and DOCK 6 (genetic algorithm) [39]. Up to 200 poses were saved for each entry and each docking tools, yielding 4.5 million registered docking poses that were next labeled as “good” or ”bad” according to the root-mean square deviation (RMSD) of their heavy atoms to the X-ray pose (good: RMSD<2.0 Å, bad: RMSD>4.0 Å), and split into training, validation and test sets (core, hold-out) as previously described [18] . Since the splitting protocol is based on the PDB identifier of the corresponding protein-ligand complexes, any pose of one protein-ligand pair never appears in both training and/or validation/test sets. While the RMSD distribution of docked poses generated with PLANTS and Surflex was shifted to the range of high-quality poses (Figure 3.2), DOCK 6 poses generated by our settings (Table S1) were clearly of lower quality (mean RMSD = 7.5 Å, Figure 3.2) thus serving as the main source of poses of lower quality in the current study. Altogether, ca. 1.23 million poses were annotated as good (Figure 3.3), therefore requiring to undersample the set of bad poses to achieve the equity of the number of good and bad poses in training, validation, and test sets (Figure 3.2).

Since the DNN models are reading protein-ligand interaction graphs, we reasoned that an additional metric focusing also on pairwise interactions, should be used to

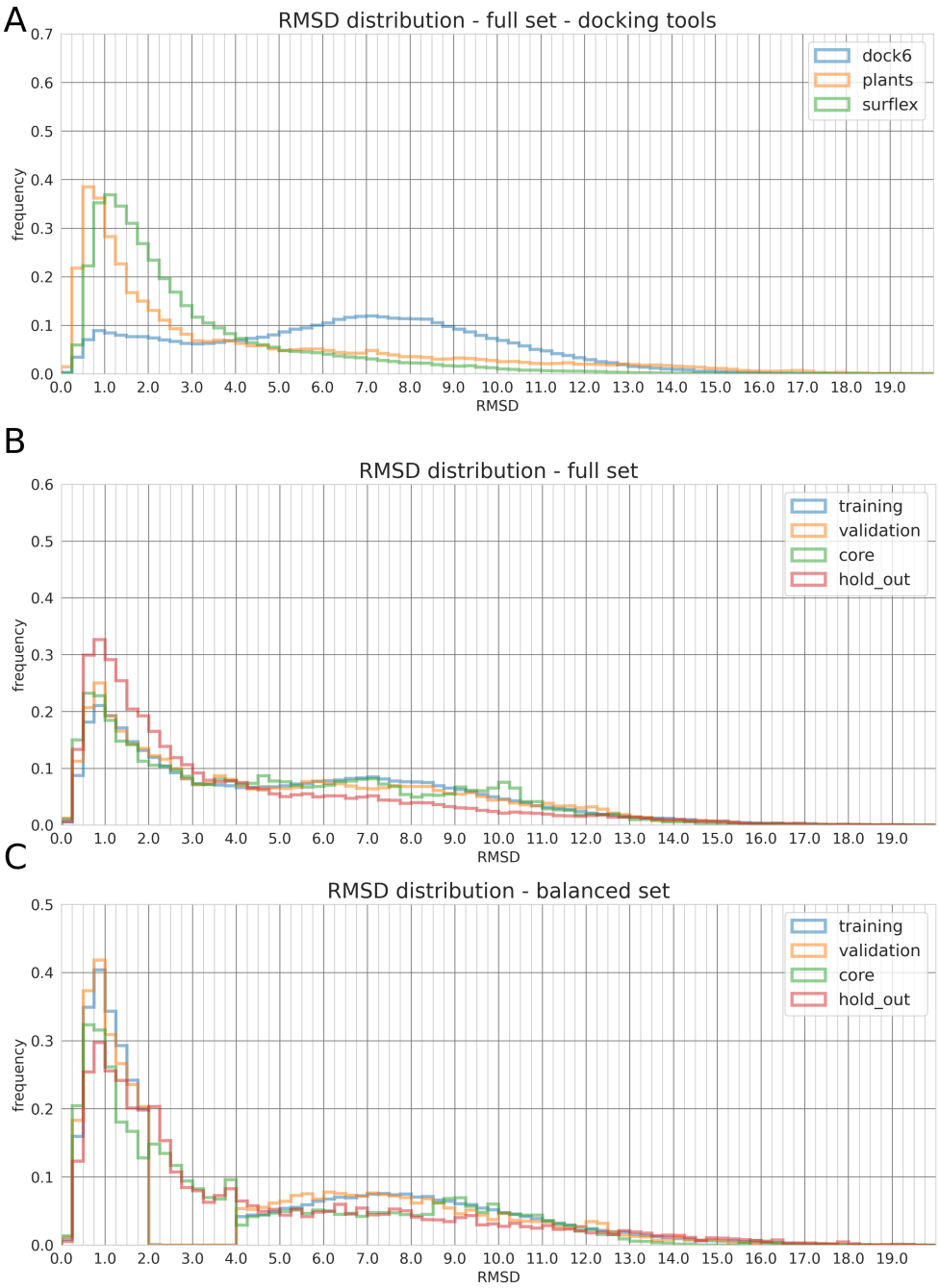


Figure 3.2 Distribution of RMSD of docked poses to the X-ray structure of PDBbind ligands. A) full set, B) training, validation, core and hold-out sets (no balancing of good vs. bad poses), C) training, validation, core and hold-out sets (balancing of good vs. bad poses).

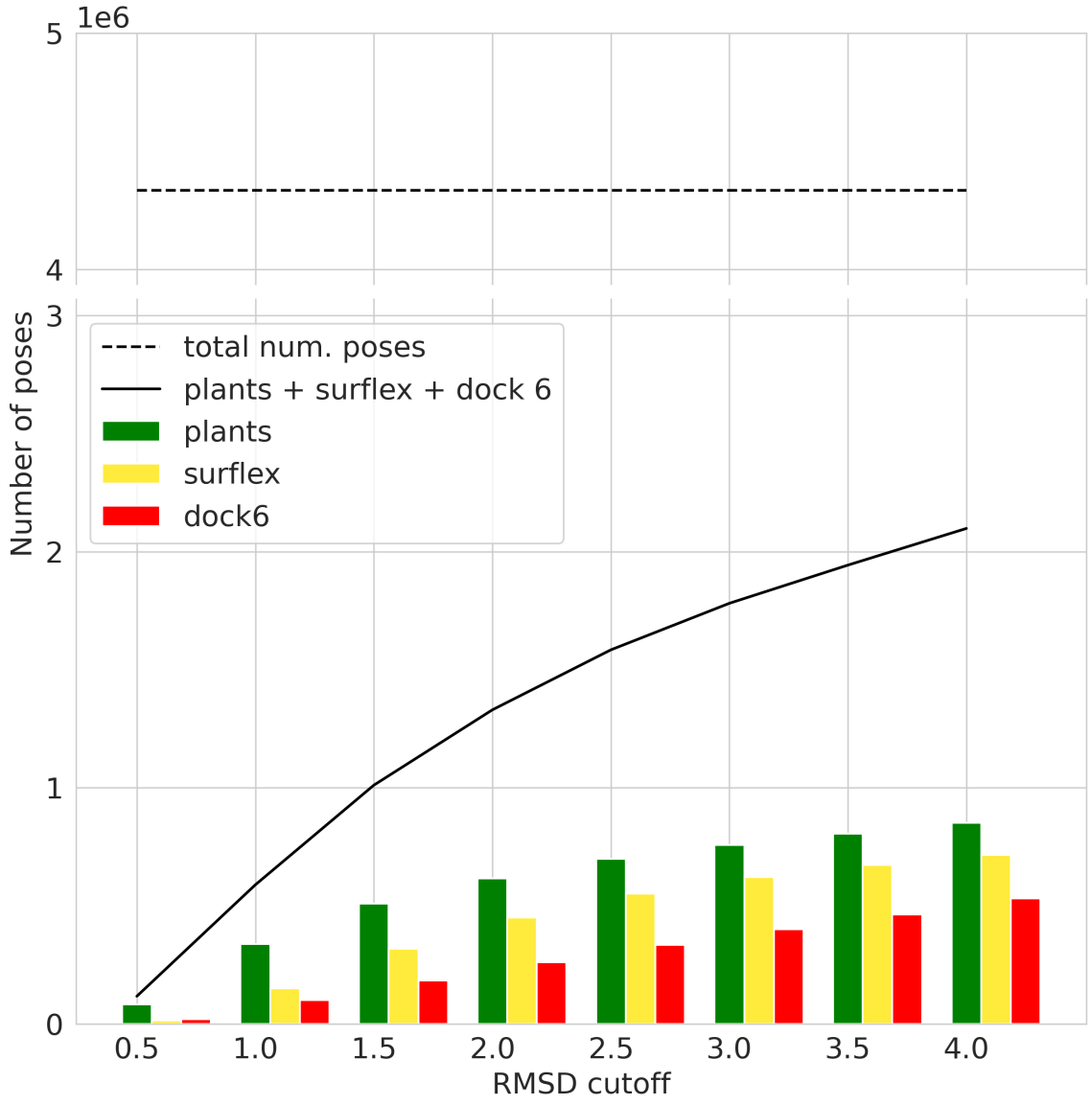


Figure 3.3 Cumulative enrichment in good poses for the three docking tools.

qualify a good pose. We thus calculated the similarity of protein-ligand interaction fingerprints (IFPs) [40] between docked and X-ray poses. Although RMSD values are inversely proportional to IFP similarities (Figure 3.4), using a double selection criterion (good pose: RMSD<2.0 Å AND IFP similarity >0.6; bad pose: RMSD>4.0 Å AND IFP similarity <0.4) permits to refine the pose labeling procedure with good only poses being now defined by a RMSD lower than 2 Å while the IFP similarity is higher than 0.6. This procedure prevents erroneous labeling of many docking poses considered correct by the RMSD criterion but incorrect by the IFP similarity criterion (Figure 3.4). For the test set, the class of inaccurate poses was augmented by the factor of 2 via addition of the equal number of poses of intermediate quality (2.0 Å <RMSD<4.0 Å) to the set of "bad" poses.

The final balanced set of docking poses, used for model training, validation and test comprises 272,839 good and 289,519 bad poses, picked among the three docking tools' proposals (Table 3.1).

Table 3.1. Statistics on docking poses of PDBbind ligands

Docking tool	Number of good poses[a]		Number of bad poses[b]	
	Full set	Balanced set	Full set	Balanced set
PLANTS	414,301	106,037	275,104	53,005
Surflex	268,327	94,129	77,684	21,213
DOCK 6	151,947	72,673	1,002,372	215,301
Total	834,575	272,839	1,355,160	289,519

[a] RMSD <2.0 Å and IFP similarity >0.60; [b] RMSD $\geq$ 2.0 Å

Binary classification of docking poses with MPNN models operating on interaction graphs.

The lack of structural data on complexes with weak binders as well as the high sparsity of the protein-ligand training matrix leads to a limited applicability of machine learning based models to real virtual screening problems, in which the model should either attribute low scores to weak binders or discard them in another way. In order to apply a binding affinity regressor to docking poses, a binary classifier using exactly the same

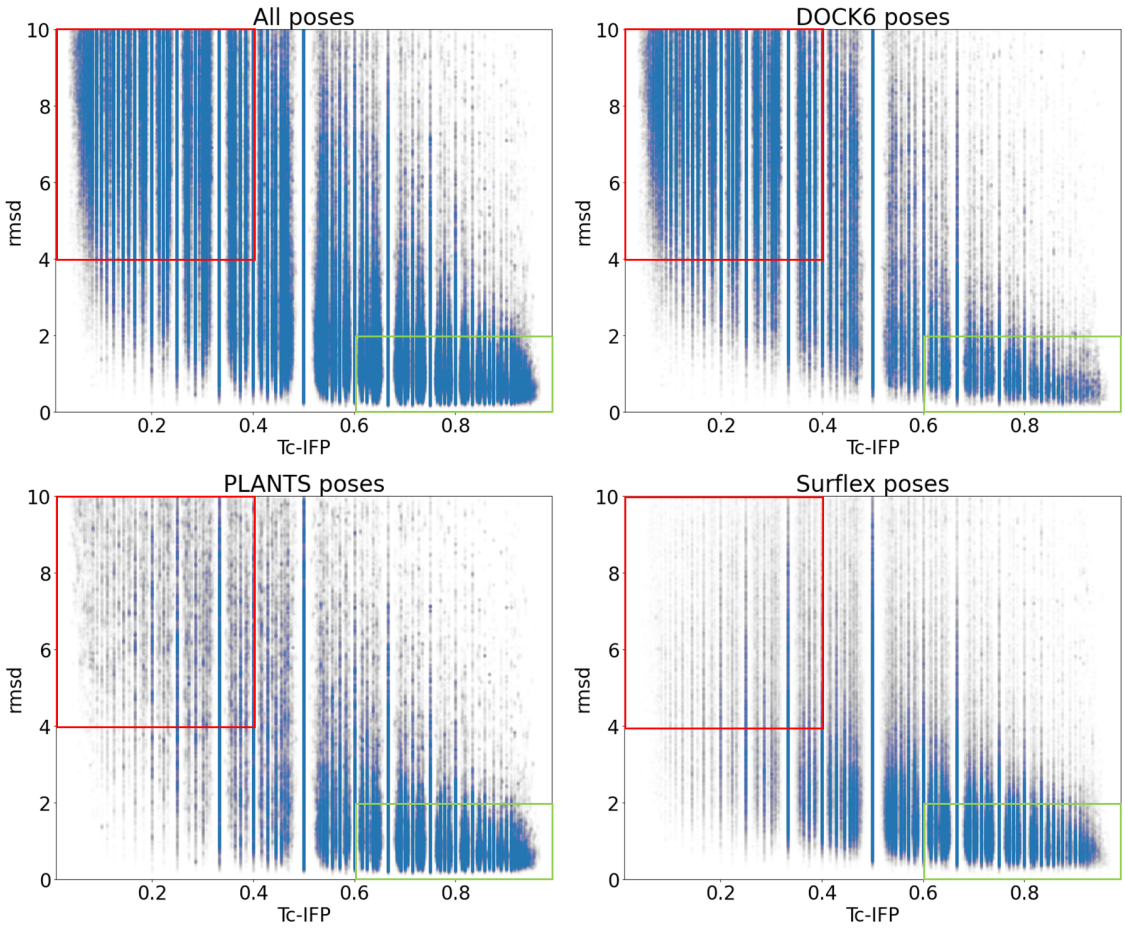


Figure 3.4 Variation of RMSD with respect to the similarity of protein-ligand interactions, upon comparing docking and X-ray poses of the full set of PDBbind ligands. Docking poses are labelled good if located in the lower right green quadrant, and bad if located in the upper left quadrant or anywhere else outside the green quadrant.

protein-ligand interaction graph input as the affinity prediction model, needs to be developed to properly select the predicted good poses that are then passed to a next affinity prediction model. The same MPNN architecture as that previously described for affinity regression [18] to train a model reading protein-ligand interaction graphs generated by IChem [41] and predict the pose label (good vs. bad).

Two binary classification models were trained using two either 4 or 6 Å pairwise distance cut-offs to register non covalent interactions (Figure 3.5). Both models exhibit and excellent classification accuracy with area under the receiver-operating curve (ROC AUC) above 0.80 for almost all evaluated sets (Figure 3.5). The herein obtained accuracies are in line with previous reports, attesting that our MPNN models compete with state-of-the-art pose classification models [17, 18, 33]. In agreement with the previously reported affinity prediction model trained on X-ray structures [18] , interaction graphs using the 6 Å pairwise distance led to better pose classifications than the shorter 4 Å distance range (Figure 3.4). As to be expected, the ROC AUC value slightly decreases when switching from the ligand-biased core set (0.849 and 0.882 for the 4 Å and 6 Å graphs, respectively) to the diverse and larger hold-out set (0.796 and 0.795 Å, respectively; Figure 3.5). At this high level of accuracy, the MPNN models were equally accurate to assign good labels to the native docking poses (Figure 3.5) although the classification accuracy varies with the MPNN probability output used as a threshold to score good and bad labels. As previously reported [19], the usually considered threshold of 0.5 for good/bad pose labelling does not correspond with the optimal classification of the three investigated set (Figure 3.5). Since the optimal threshold is clearly test set-dependent, we further used the regular 0.50 threshold in the next studies. It ensures an accuracy of ca. 80% on the two test sets investigated here.

Predicting binding affinities from docking poses with prior filtering of predicted incorrect binding modes.

We implemented a binding affinity prediction pipeline consisting of two DDN models, a first pose classifier and a second binding affinity regressor. In a first stage, an interaction graph was generated for each protein-ligand complex. Then the class label (good, bad) was first predicted with the pose classification model and only predicted good poses

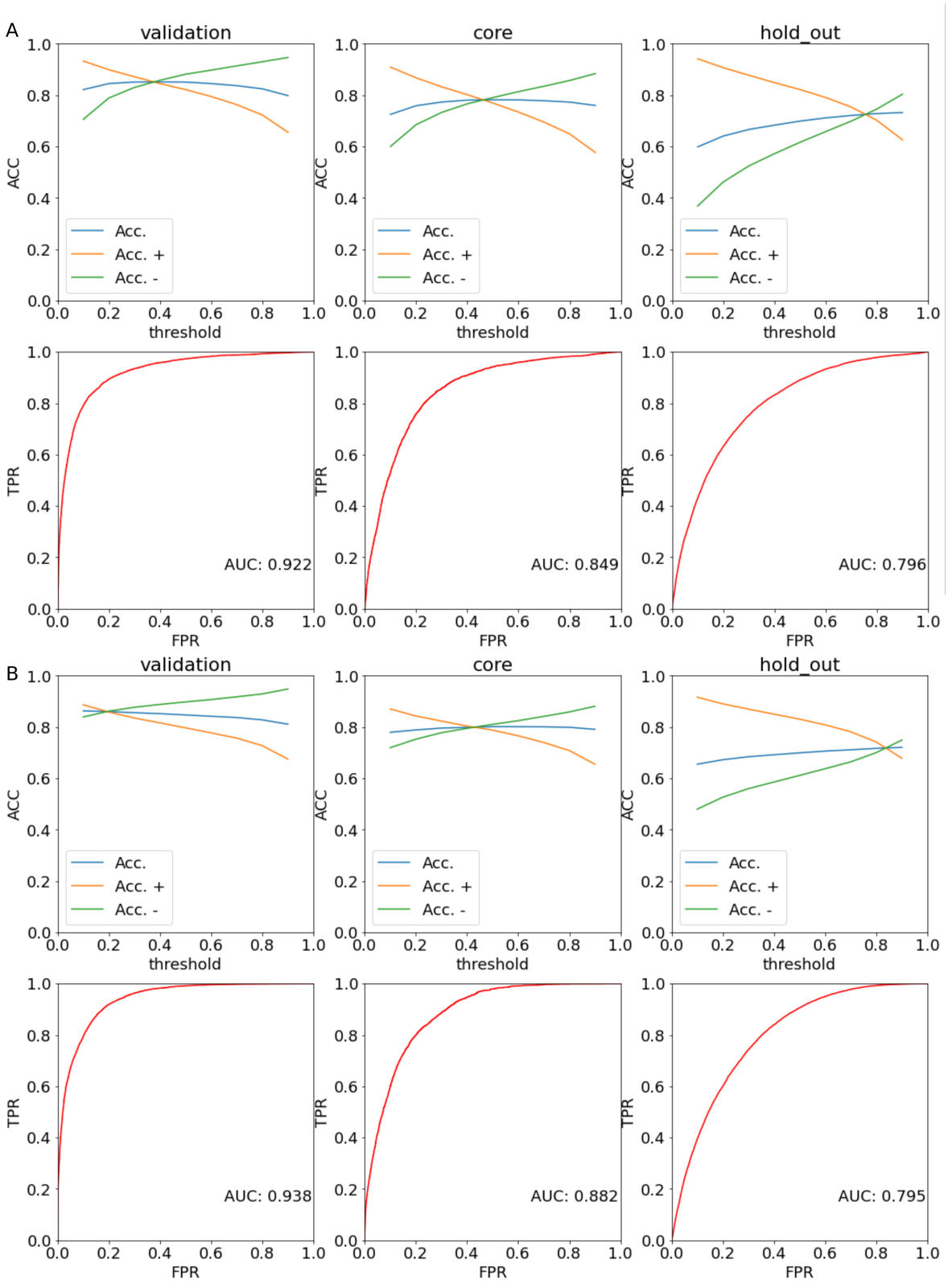


Figure 3.5 Binary classification accuracy of MPNN models trained on protein-ligand interaction graphs defined from protein-ligand interactions registered at 4 Å (A) or 6 Å cut-off distances (B). Accuracies are given for all poses (Acc), positive samples (good poses, Acc+) and negative samples (bad poses, Acc-). ROC curves are defined from the model probability output values generated for every docking pose.

are next passed to a binding affinity regression model predicting the binding affinity in pK unit (Figure 3. 6).

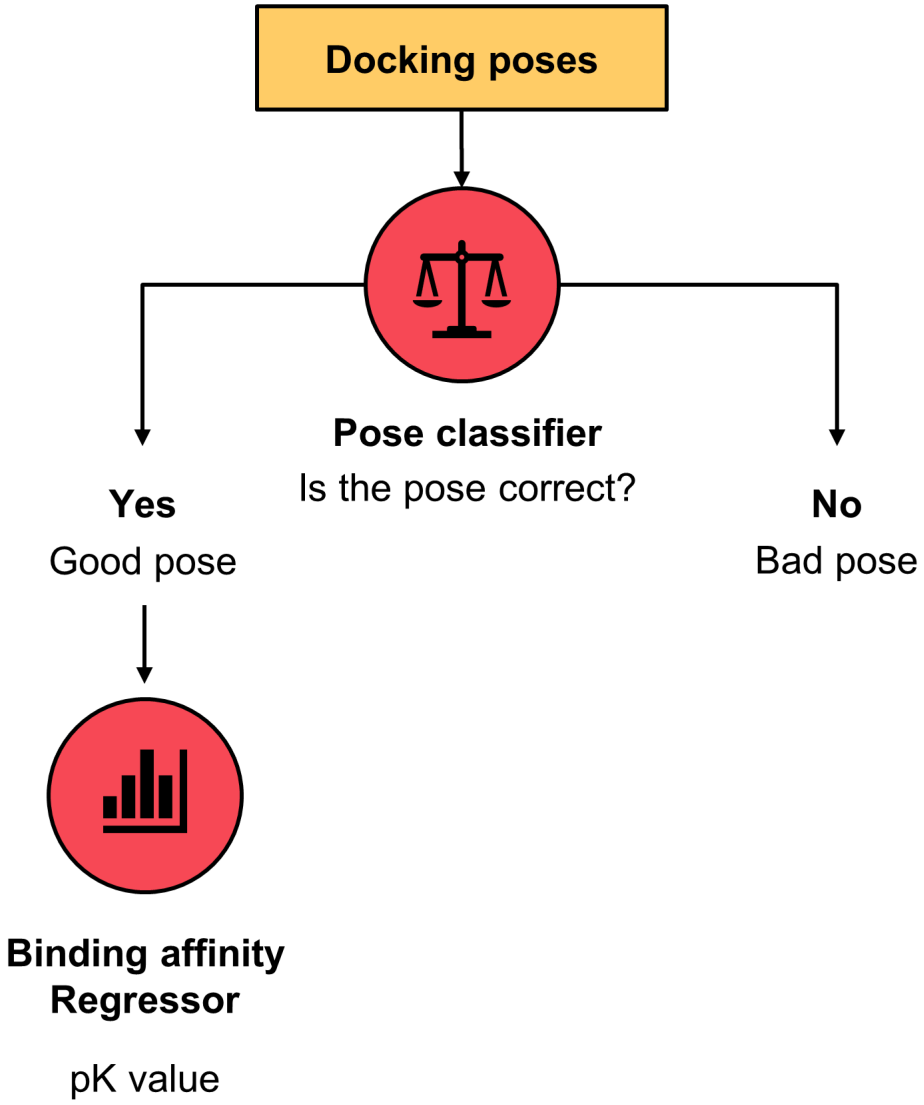


Figure 3.6 A flowchart of the binding affinity prediction pipeline with prior pose filtering

In the current study, two different pose classification models trained on the balanced dataset of docking poses were used from protein-ligand interactions detected at either 6 or 4 Å pairwise distances. Importantly, the same interaction graph has been used for pose classification and binding affinity regression models. To ascertain the applicability of the affinity prediction models, they were trained and applied to either X-ray structures or docking poses, thereby enabling to test whether training and testing on different molecular objects is desirable or not. In the first scenario, the pose

classifier and affinity regressors were applied on X-ray structures. Surprisingly, the classifier previously trained on docking poses failed to predict as "good" true X-ray structures in 22 to 31% of test cases (Table 3.2), suggesting that the pose classifier is very sensitive to the origin of the test samples. This discrepancy probably arises from the different treatment, notably with respect to encountered force-fields, of X-ray structures and docking poses. Harmonization of this treatment was recently shown to diminish the sensitivity to the origin of atomic coordinates [31]. Affinity predictions on the remaining X-ray structures were of good quality as estimated by the R^2 determination coefficient and the Pearson R_p coefficient of the regression applied to core set samples (R^2=0.521, R_p=0.750). As to be expected, the quality of the prediction significantly decreased when applied to the more difficult hold-out set (R^2=0.316, R_p=0.564). It is important to notice that the erroneous pre-filtering of X-ray structures by the pose classifier had a minor impact on the affinity predictions since the regression statistics on the non-filtered test samples were almost equivalent, as reported in our original study (core set: R_p=0.728, hold-out set, R_p=0.607) [18] .

Table 3.2. Performance of binding affinity prediction models with prior filtering of incorrect poses. Protein-ligand interactions are detected within a 6 Å distance range.

Set	Scenario 1[a]			Scenario 2[b]			Scenario 3[c]		
	Inc[d]	R^{2e}	R_p[f]	inc	R^2	R_p	inc	R^2	R_p
Core	0.222	0.521	0.750	0.557	0.386	0.706	0.556	0.612	0.803
Hold-out	0.316	0.306	0.564	0.428	0.103	0.485	0.428	0.399	0.644

[a] regressor trained and applied on X-ray stuctures, [b] regressor trained on X-ray structures and applied to docking poses, [c] trained and applied to docking poses, [d] Inc: frequency of bad pose classification, [e] R^2: determination coefficient, [f] R_p Pearson r coefficient.

In the second scenario, the affinity regressor trained on X-ray structures was applied to good docking poses, a situation mirroring most previous studies. Unfortunately, this translational application is operated at the cost of the quality of the predictions since the R^2 and R_p coefficients significantly decreased for both core and hold-out test samples (Table 3.2). In the last scenario, both the classifier and the re-

gressor operated in the same set of docking poses, which brings the affinity prediction back to statistical values (core set: R_p=0.803, hold-out set, R_p=0.644) even higher than obtained in the first scenario where classifier and regressor was operating on X-ray structures. Reassuringly, the failure rate of the pose classifier to properly label docking poses is now close to 50%, a value expected from the balanced nature of good vs. bad poses on which the classifier has been trained on.

Again, we observed better results when testing core set samples with respect to hold-out samples. The obtained value for the core set (R_p = 0.803) is undoubtedly overoptimistic due to inherent ligand and protein biases previously observed for this set. The accuracy obtained on the hold-out set (R_p = 0.644) is more representative of the true accuracy of our DNN models in predicting binding affinities for a realistic set of novel entries (recall Figure 3.1). Although augmenting input data with docking poses clearly help making better affinity predictions (compare scenarios 2 and 3), the obtained model still meets difficulties in generalizing to unseen complexes. We last verified whether the previous observation [18] that more complex interaction graphs registered at a 6 Å pairwise distance, leads to better prediction with respect to simpler graphs registered at a maximal 4 Å distance (Table 3.3). The same behavior was observed again, whatever the scenario (Table 3.3). In all cases, observed R^2 and R_p coefficients were significantly lower, than those derived from 6 Å interaction graphs.

Table 3.3. Performance of binding affinity prediction models with prior filtering of incorrect poses. Protein-ligand interactions are detected within a 4 Å distance range.

Set	Scenario 1[a]			Scenario 2[b]			Scenario 3[c]		
	Inc[d]	R^{2e}	R_p[f]	inc	R^2	R_p	inc	R^2	R_p
Core	0.128	0.435	0.660	0.568	-0.275	0.381	0.568	0.316	0.568
Hold	0.196	0.087	0.445	0.450	-0.739	0.250	0.450	0.136	0.460

[a] regressor trained and applied on X-ray stuctures, [b] regressor trained on X-ray structures and applied to docking poses, [c] trained and applied to docking poses, [d] Inc: frequency of bad pose classification, [e] R2: determination coefficient, [f] R_p Pearson r coefficient.

The two DNN models trained and applied on the same objects (scenario 1: X-

ray poses, scenario 3: docking poses) are clearly different and yielded different affinity predictions notably when the test samples were more distant from the training samples (Figure 3.7).

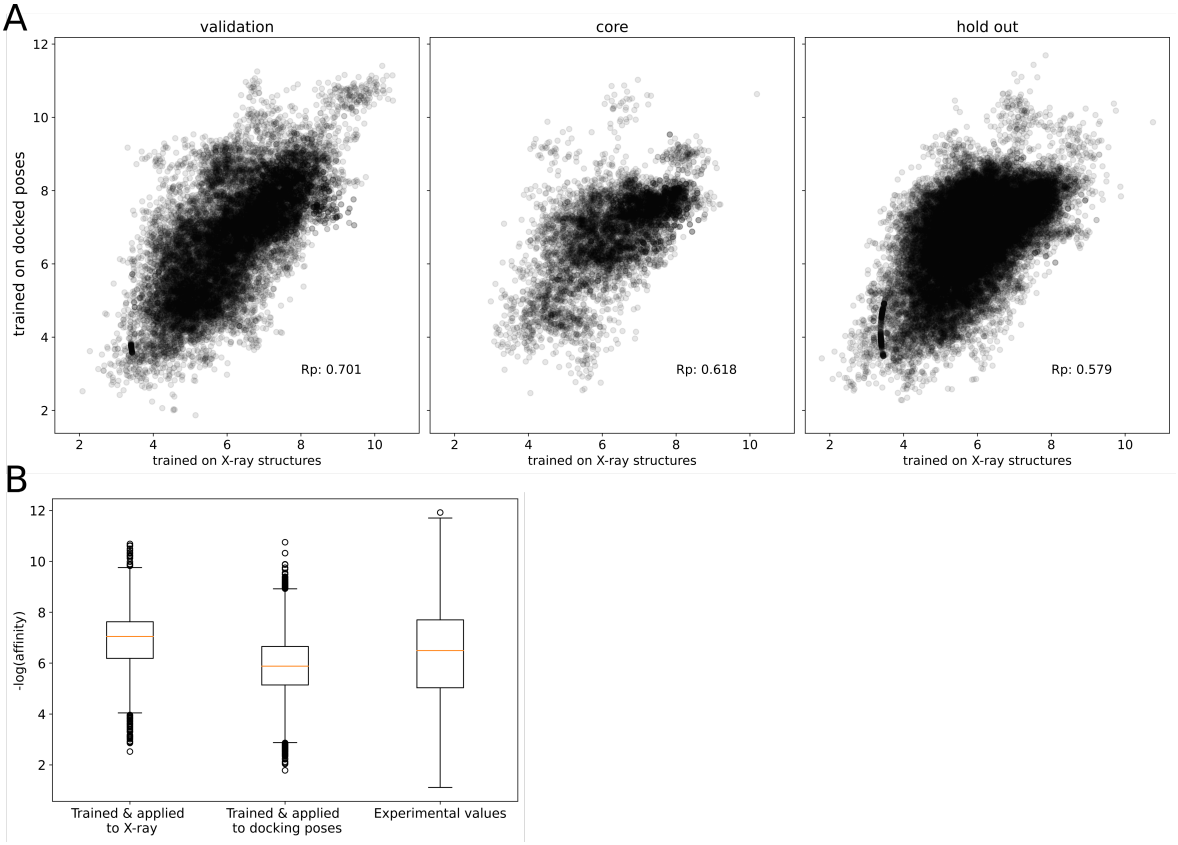


Figure 3.7 Comparison of affinity predictions for models trained and applied on the same set of molecular objects (X-ray structures, scenario 1 ; docking poses, scenario 3). A) Correlation of binding affinity predictions with a regression model trained on X-ray structures and on docked poses (6 Å interaction detection range), B) Box-and-whisker plot of predicted affinity values. The boxes delimit the 25th and 75th percentile, and the whiskers delimit the 1st and 99th percentiles. The median values are indicated by a horizontal line in the box, respectively. Outliers are indicated by a circle.

We notably verified whether the previously noticed tendency of affinity prediction ML models, trained and on X-ray structures [18, 23, 42] to output affinity values within a narrow range centered on the mean of value of training samples, is still verified when the ML model has been specifically trained and applied to an augmented set of docking poses (Figure 3.7). The distribution of predicted affinities is significantly shifted to lower values in case docking poses are used for training an application, still within a

narrow range as observed when X-ray structures are used (Figure 3.7).

The best possible DNN model operating on an augmented set of docking poses still hardly generalize.

Since the composition of the hold-out set in terms of ligand and protein novelty with respect to the training data was previously assessed (Figure 3.1), we decomposed affinity prediction values obtained after training and testing docking poses (scenario 3, Table 3.3) for the four possible categories of hold-out complexes (Table 3.4). Please notice than the notion of "novelty" of ligand and protein partners was here very conservative as ligand and protein identity, according to PDB HET codes and SwissProt accession numbers, were considered.

Table 3.4. Generalizability of binding affinity predictions with respect to the novelty of protein and ligand composition of the hold-out test set.

Ligand	Protein	R^2	R_p	RMSE
known	Known	0.437	0.664	1.329
known	Unknown	0.503	0.713	1.472
unknown	Known	0.393	0.639	1.371
unknown	unknown	0.302	0.597	1.506

[a] regressor trained and applied on X-ray stuctures, [b] regressor trained on X-ray structures and applied to docking poses, [c] trained and applied to docking poses, [d] Inc: frequency of bad pose classification, [e] R^2: determination coefficient, [f] R_p Pearson r coefficient.

As to be expected, the quality of the predictions decreases with the level of uncertainty with respect to ligand and protein composition of the external test set (Table 3.4.) Despite being trained on a large set of docking poses, the MPNN model still does not generalize well to complexes between new ligands and new proteins. Given the tendency of the model to predict binding affinities in a narrow range (Figure 7), this observation suggests that memorization still dominates true learning of protein-ligand interactions in the herein developed DNN models.

Multitask classification and regression models do not compete with the step-by-step pipeline.

As an alternative version to the iterative workflow beginning with pose classification and ending with binding affinity prediction, we implemented a multi-task model combining both tasks simultaneously. In this model the protein-ligand interaction graph is processed first with a unique MPNN, its readout vector being then treated separately by two fully connected networks aimed at pose classification and binding affinity prediction. Despite being computationally more efficient, these models demonstrated worse performance than their counterpart pipelines of two single-purpose models (Table 3.5).

Table 3.5. Performance of a multi-task binding affinity prediction model.

Set	6 Å interaction graph			4 Å interaction graph		
	Inc[a]	R^{2b}	R_p[c]	Inc[a]	R^{2b}	R_p[c]
Core	0.424	0.264	0.686	0.386	0.254	0.511
Hold	0.309	-0.255	0.500	0.266	0.064	0.415

[a] Inc: frequency of bad pose classification, [b] R^2: determination coefficient, [c] R_p Pearson r coefficient.

One of the possible causes can be the difficulty of the model to optimize two loss functions simultaneously. The classification loss overfitting starts much earlier than the convergence of the regression loss is achieved(Figure 3.8).

3.3 Experimental section

PDBbind dataset.

The dataset composition, preparation and and splitting was identical to that reported in our previous study [18]. We filtered out PDB ids, for which the 6 Å interaction graph construction failed, thus obtaining the training, validation, core, and hold out set sizes of 4508, 463, 143, and 1326, respectively.

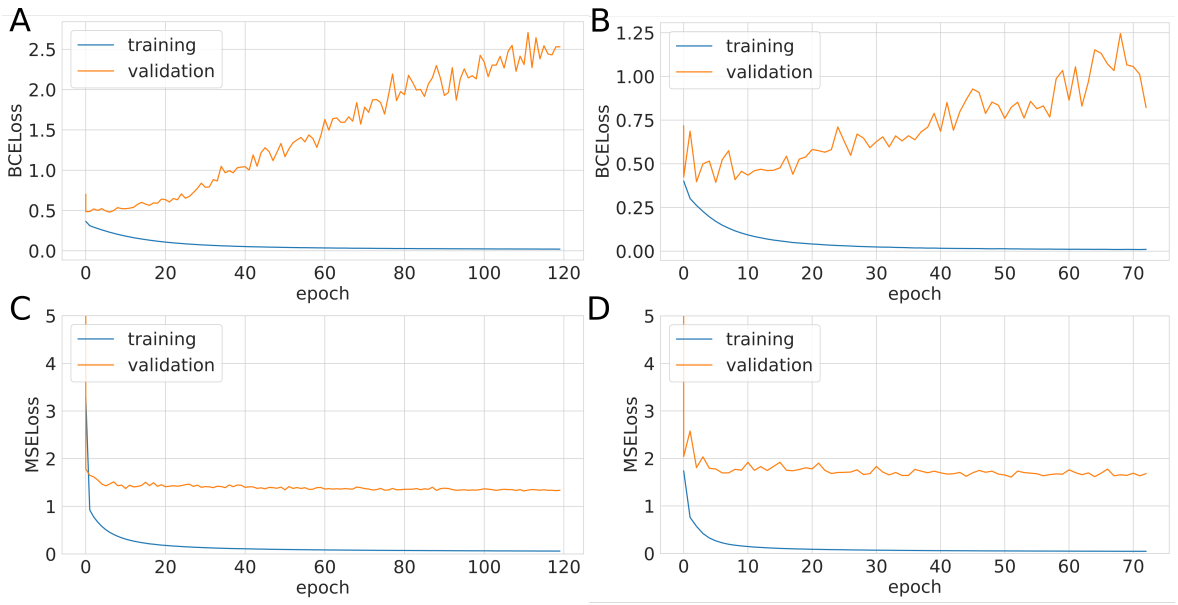


Figure 3.8 Delayed convergence of the regression loss and early overfitting of classification loss in a multi-task model. Model training on 4 Å graphs (A —- classification loss, C —- regression loss) and 6 Å graphs (B — classification loss, D — regression loss)

Self-docking of PDBbind entries.

PLANTS docking. Molecular docking was performed in PLANTS v.1.2 [37] using the chemplp scoring function and a search speed of 1. The center of mass used for delimiting the cavity was defined as the center of mass of the PDBbind ligand. The radius for the search space was set as the largest distance between the center of mass and any binding site heavy atom as provided by PDBbind in the pocket.mol2 file. Sampled conformers were clustered with an RMSD of 0.25 Å, a maximum of 100 poses being finally saved. An example configuration file is provided in Table S1.

Surflex-dock docking. Molecular docking was performed in Surflex v.4543 [38]. The protomol was generated from the X-ray pose of the ligand using standard parameters. Docking was performed with -pgeom and +macrocyc ligand preparation parameters. A maximum of 100 docking solutions were saved.

DOCK doking. Protein surface was computed with the WriteDMS script implemented in UCSF Chimera [43]. Sphere selection was performed with a 4.0 Å cutoff. An example parameter file is provided in Table S3. Clustering of conformations was performed with an RMSD threshold of 1.0 Å, a maximum of 100 poses were finally

saved.

The RMSD of heavy atoms between all sampled conformers and the original X-ray structure of a ligand was computed with the sf-dock rms utility of Surflex-Dock.

Preparation of interaction graphs.

Protein-ligand interactions were computed in IChem v.5.2.9. [41], using either a 4.0 Å or a 6.0 Å maximal cut-off distance and saved in a json format, as previously described [18].

Interaction fingerprints.

Protein-ligand interaction fingerprints were calculated with the IFP module of the IChem v.5.2.9 package.

Docking pose labelling.

A binary labeling (good vs. wrong) of all docking poses was done according to the RMSD of heavy atoms to the X-ray pose, and the similarity of protein-ligand interaction fingerprints, measured by a Tanimoto coefficient (Tc-IFP). The pose was labelled as "good" if the RMSD was less or equal to 2 Å and the Tc-IFP value higher than 0.6. The pose was labelled as "wrong" if the RMSD was higher or equal to 6 Å and Tc-IFP lower or equal to 0.4. Poses not labelled "good" or "wrong" were nor retained further. To achieve an equal number of good and poses poses for each PDBbind entry, poses of the overrepresented class were randomly removed until their number reached that of the underrepresented class. Afterwards, the randomly sampled poses of intermediate quality (RMSD between 2.0 Å and 4.0 Å) were added to the core set and hold out being labeled as "wrong poses". Altogether, the final dataset comprises 272 839 "good" poses and the 289 519 "wrong" poses (the sizes of classes for training and validation sets are identical).

Binding affinity regression models.

The MPNN architecture recently described for predicting binding affinities from X-ray structures [18] was used herein. The node and edge feature vectors were transformed

by linear layers, then the latent node feature vectors were iteratively updated with the message passing neural network. The readout vector was further used as an input of a two-layer fully connected network. Hyperparameters were optimized with ADAM using MSELoss as a loss function. The dropout rate for the graph convolution subnetwork was set to 0.2, the weight decay to 0.001. The size of the latent node feature vector was 2048, the number of message-passing steps was set to 2, the initial learning rate was $2 \cdot 10^{-4}$ with subsequent multiplications by 0.9 after the first 20 epochs with no validation loss decrease, and then multiplied by 0.9 every 40 epochs.

Binary classification models.

The MPNN for binary classification was based on the previous regression MPNN with a sigmoid function applied to the output of the last linear layer. The learning rate of $1 \cdot 10^{-5}$ was constant. Hyperparameters were optimized with ADAM using BCELoss as a loss function.

Binding affinity prediction with prior classification.

In order to apply pose classification and binding affinity prediction models in a pipeline, the interaction graph representation was generated first. Then, separate prediction of binding affinity and pose class was performed with regression and classification models. The predicted value of a binding affinity logarithm was returned only for those poses, which were characterized as "correct" ones by a classifier (otherwise returning None).

Multitask classification and regression models.

The architecture of a mixed predictor was based on that used for classification and regression models. After performing a graph convolution operation with the MPNN module, the resulting latent vector was processed by two independent fully connected networks, working as a regressor and as a classifier. The loss function was set as a sum of classification and validation losses (Eq. 3.1)

$$Loss = CLoss(target, prediction_{class}) + \alpha \cdot RLoss(target, prediction_{regr}) \qquad (3.1)$$

where α=0 if prediction <positive class cutoff, else $\alpha = 1$.

Evaluation metrics.

The quality of classification models was determined using ROC AUC, accuracy (Acc), positive (PPV) and negative (NPV) predictive values as follows (Eq. 3.2–3.4)

$$PPV = \frac{TP}{TP + FP} \tag{3.2}$$

$$NPV = \frac{TN}{TN + FN} \tag{3.3}$$

$$Acc = \frac{N_{corr.pred.}}{N_{pred}} \tag{3.4}$$

where TP are true positives, FP false positives, TN true negatives, FN false negatives, $N_{corr.pred.}$ — the number of correct predictions, and N_{pred} the number of predictions.

The quality of the binary affinity regression models was determined using the Pearson's correlation coefficient (R_p) and the root-mean square error (RMSE) of the prediction as follows (Eq. 3.5–3.6)

$$R_p = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \tag{3.5}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_i - Y_i)^2}{n}} \tag{3.6}$$

where n is the number of samples, X_i the experimental affinity value, Y_i the predicted affinity value, $\overline{X}$ the mean of experimental values, and $\overline{Y}$ the mean of predicted values.

3.4 Conclusions

Predicting binding affinities from 3D coordinates of protein-ligand complexes remains a major challenge in drug discovery. Despite apparent advances in retrospective analyses, machine learning models still suffer from poor generalizability mainly because of the limited set of diverse 3D structures they are trained on. We believe that most previous studies are unfortunately biased by significant flaws concerning the respective contribution of ligand and protein instances in training and external test sets. Given that all previous studies rely on the PDBbind dataset of protein-ligand X-ray structures of known affinity, we here evidenced that the PDBbind core set, usually taken

as an external test set to evaluate the predictive power of ML models, is too close to the general training set and therefore leads to overoptimistic results where memorization dominates true learning. This observation explains why adding explicit ligand descriptors to protein-ligand attributes always enhance the apparent predictivity of ML models. We therefore recommend abandoning the usage of the core set for evaluating binding affinity ML models and propose a much more realistic temporal split to define a truly diverse hold-out split for external validation.

Having a debiased test set in our hands, we then asked the question whether augmenting input data with docking poses could help define more accurate and generalizable models. This requires a first selection of near-native docking poses by a classifier trained to distinguish good from bad poses. Again, we noticed that the common practice of using as a single criterion the RMSD of docking poses to the X-ray structure is not optimal since RMSD only partially correlates with the conservation of protein-ligand interactions that ML models reads as input. We herein propose to additionally use the similarity of protein-ligand interactions to refine the pose labelling and to prevent selecting, as native, a significant proportion of incorrect poses.

With a debiased training set and a refined labeling of docking poses, we unambiguously shown that our MPNN models should be applied to the same molecular objects than those they have been trained on. In other words, models trained on X-ray structures should be applied to X-ray structure but not to docking poses. Accordingly, models trained on docking poses should be tested on docking poses only. Whatever the source of input structures (X-ray diffraction, docking), the best ML models in our hands exhibit a comparable accuracy (R_p = 0.55–0.56) in predicting affinities of the hold-out samples. The observed accuracy is much lower than that reported in most biased retrospective studies challenging the core set, and unfortunately not sufficient for their wide application to real life cases where proteins and ligands are usually unknown to the training samples.

3.5 References

(1) Yang, C.; Chen, E. A.; Zhang, Y. *Molecules* **2022**, *27*, 4568.

(2) Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. *Journal of chemical information and modeling* **2018**, *59*, 895–913.

(3) Wagner, J. R.; Churas, C. P.; Liu, S.; Swift, R. V.; Chiu, M.; Shao, C.; Feher, V. A.; Burley, S. K.; Gilson, M. K.; Amaro, R. E. *Structure* **2019**, *27*, 1326–1335.

(4) Dunbar Jr, J. B.; Smith, R. D.; Yang, C.-Y.; Ung, P. M.-U.; Lexa, K. W.; Khazanov, N. A.; Stuckey, J. A.; Wang, S.; Carlson, H. A. *Journal of chemical information and modeling* **2011**, *51*, 2036–2046.

(5) Cheng, T.; Li, X.; Li, Y.; Liu, Z.; Wang, R. *Journal of chemical information and modeling* **2009**, *49*, 1079–1093.

(6) Gathiaka, S.; Liu, S.; Chiu, M.; Yang, H.; Stuckey, J. A.; Kang, Y. N.; Delproposto, J.; Kubish, G.; Dunbar, J. B.; Carlson, H. A., et al. *Journal of computer-aided molecular design* **2016**, *30*, 651–668.

(7) Gaieb, Z.; Liu, S.; Gathiaka, S.; Chiu, M.; Yang, H.; Shao, C.; Feher, V. A.; Walters, W. P.; Kuhn, B.; Rudolph, M. G., et al. *Journal of computer-aided molecular design* **2018**, *32*, 1–20.

(8) Gaieb, Z.; Parks, C. D.; Chiu, M.; Yang, H.; Shao, C.; Walters, W. P.; Lambert, M. H.; Nevins, N.; Bembenek, S. D.; Ameriks, M. K., et al. *Journal of computer-aided molecular design* **2019**, *33*, 1–18.

(9) Parks, C. D.; Gaieb, Z.; Chiu, M.; Yang, H.; Shao, C.; Walters, W. P.; Jansen, J. M.; McGaughey, G.; Lewis, R. A.; Bembenek, S. D., et al. *Journal of computer-aided molecular design* **2020**, *34*, 99–119.

(10) Guedes, I. A.; Pereira, F. S.; Dardenne, L. E. *Frontiers in Pharmacology* **2018**, *9*, 1089.

(11) Wang, L.; Wu, Y.; Deng, Y.; Kim, B.; Pierce, L.; Krilov, G.; Lupyan, D.; Robinson, S.; Dahlgren, M. K.; Greenwood, J., et al. *Journal of the American Chemical Society* **2015**, *137*, 2695–2703.

(12) Renaud, J.; Chari, A.; Ciferri, C. *Nature reviews Drug discovery* **2018**, *17*, 471–492.

(13) Sadybekov, A. A.; Sadybekov, A. V.; Liu, Y.; Iliopoulos-Tsoutsouvas, C.; Huang, X.-P.; Pickett, J.; Houser, B.; Patel, N.; Tran, N. K.; Tong, F., et al. *Nature* **2022**, *601*, 452–459.

(14) Kim, J.; Park, S.; Min, D.; Kim, W. *International Journal of Molecular Sciences* **2021**, *22*, 9983.

(15) Ballester, P. J.; Schreyer, A.; Blundell, T. L. *Journal of chemical information and modeling* **2014**, *54*, 944–955.

(16) Moon, S.; Zhung, W.; Yang, S.; Lim, J.; Kim, W. Y. *Chemical Science* **2022**, *13*, 3661–3673.

(17) Ashtawy, H. M.; Mahapatra, N. R. *Journal of chemical information and modeling* **2018**, *58*, 119–133.

(18) Volkov, M.; Turk, J.-A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D. *Journal of medicinal chemistry* **2022**, *65*, 7946–7958.

(19) Morrone, J. A.; Weber, J. K.; Huynh, T.; Luo, H.; Cornell, W. D. *Journal of chemical information and modeling* **2020**, *60*, 4170–4179.

(20) Janela, T.; Bajorath, J. *Nature Machine Intelligence* **2022**, 1–10.

(21) Volkamer, A.; Riniker, S.; Nittinger, E.; Lanini, J.; Grisoni, F.; Evertsson, E.; Rodrıguez-Pérez, R.; Schneider, N. *Artificial Intelligence in the Life Sciences* **2023**, 100056.

(22) Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C. J.; Duca, J. S.; Hornak, V.; Koes, D. R.; Kurtzman, T. *PloS one* **2019**, *14*, e0220113.

(23) Gabel, J.; Desaphy, J.; Rognan, D. *Journal of chemical information and modeling* **2014**, *54*, 2807–2815.

(24) Shen, C.; Hu, Y.; Wang, Z.; Zhang, X.; Pang, J.; Wang, G.; Zhong, H.; Xu, L.; Cao, D.; Hou, T. *Briefings in Bioinformatics* **2021**, *22*, bbaa070.

(25) Sieg, J.; Flachsenberg, F.; Rarey, M. *Journal of chemical information and modeling* **2019**, *59*, 947–961.

(26) Yang, J.; Shen, C.; Huang, N. *Frontiers in pharmacology* **2020**, *11*, 69.

(27) Pereira, J. C.; Caffarena, E. R.; Dos Santos, C. N. *Journal of chemical information and modeling* **2016**, *56*, 2495–2506.

(28) Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. *Journal of chemical information and modeling* **2017**, *57*, 942–957.

(29) Xie, L.; Xu, L.; Chang, S.; Xu, X.; Meng, L. *Chemical Biology & Drug Design* **2020**, *96*, 973–983.

(30) Zheng, L.; Fan, J.; Mu, Y. *ACS omega* **2019**, *4*, 15956–15965.

(31) Son, J.; Kim, D. *PloS one* **2021**, *16*, e0249404.

(32) Boyles, F.; Deane, C. M.; Morris, G. M. *Journal of Chemical Information and Modeling* **2021**, *62*, 5329–5341.

(33) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. *Journal of chemical information and modeling* **2020**, *60*, 4200–4215.

(34) Huang, N.; Shoichet, B. K.; Irwin, J. J. *Journal of medicinal chemistry* **2006**, *49*, 6789–6801.

(35) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. *Journal of medicinal chemistry* **2012**, *55*, 6582–6594.

(36) Wang, R.; Fang, X.; Lu, Y.; Wang, S. *Journal of medicinal chemistry* **2004**, *47*, 2977–2980.

(37) Korb, O.; Stutzle, T.; Exner, T. E. *Journal of chemical information and modeling* **2009**, *49*, 84–96.

(38) Jain, A. N. *Journal of computer-aided molecular design* **2007**, *21*, 281–306.

(39) Allen, W. J.; Balius, T. E.; Mukherjee, S.; Brozell, S. R.; Moustakas, D. T.; Lang, P. T.; Case, D. A.; Kuntz, I. D.; Rizzo, R. C. *Journal of computational chemistry* **2015**, *36*, 1132–1156.

(40) Marcou, G.; Rognan, D. *Journal of chemical information and modeling* **2007**, *47*, 195–207.

(41) Da Silva, F.; Desaphy, J.; Rognan, D. *ChemMedChem* **2018**, *13*, 507–510.

(42) Tran-Nguyen, V.-K.; Bret, G.; Rognan, D. *Journal of Chemical Information and Modeling* **2021**, *61*, 2788–2797.

(43) Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E. *Journal of computational chemistry* **2004**, *25*, 1605–1612.

3.6 Supplementary materials

Table S1. DOCK 6 configuration files

conformer_search_type flex

write_fragment_libraries no

user_specified_anchor no

limit_max_anchors no

min_anchor_size 6

pruning_use_clustering yes

pruning_max_orients 500

pruning_clustering_cutoff 100

pruning_conformer_score_cutoff 100.0

pruning_conformer_score_scaling_factor 1.0

use_clash_overlap no

write_growth_tree no

use_internal_energy yes

internal_energy_rep_exp 12

internal_energy_cutoff 100.0

ligand_atom_file ligand.mol2# ligand to dock (mol2 file format)

limit_max_ligands no

skip_molecule no

read_mol_solvation no

calculate_rmsd no

use_database_filter no

orient_ligand yes

automated_matching yes

receptor_site_file selected_spheres.sph

max_orientations 500

critical_points no

chemical_matching no

use_ligand_spheres no

bump_filter yes

bump_grid_prefix grid

max_bumps_anchor 10

max_bumps_growth 10

score_molecules yes

contact_score_primary no

contact_score_secondary no

grid_score_primary yes

grid_score_secondary no

grid_score_rep_rad_scale 1

grid_score_vdw_scale 1

grid_score_es_scale 1

grid_score_grid_prefix grid

multigrid_score_secondary no

dock3.5_score_secondary no

continuous_score_secondary no

footprint_similarity_score_secondary no

pharmacophore_score_secondary no

descriptor_score_secondary no

gbsa_zou_score_secondary no

gbsa_hawkins_score_secondary no

SASA_score_secondary no

amber_score_secondary no

minimize_ligand yes

minimize_ancho yes

minimize_flexible_growth yes

use_advanced_simplex_parameters no

simplex_max_cycles 5

simplex_score_converge 0.1

simplex_cycle_converge 1.0

simplex_trans_step 1.0

simplex_rot_step 0.1

simplex_tors_step 10.0

simplex_anchor_max_iterations 50

simplex_grow_max_iterations 50

simplex_grow_tors_premin_iterations 5

simplex_random_seed 42

simplex_restraint_min yes

simplex_coefficient_restraint 20

atom_model all

vdw_defn_file vdw_AMBER_parm99.defn

flex_defn_file flex.defn

flex_drive_file flex_drive.tbl

ligand_outfile_prefix dock6_res

write_orientations no

num_scored_conformers 1000

write_conformations yes

cluster_conformations yes

cluster_rmsd_threshold 1.0

rank_ligands yes

max_ranked_ligands 1

Table S2. PLANTS configuration files

scoring function and search settings

scoring_function chemplp

search_speed speed1

input

protein_file protein.mol2 #protein file

ligand_file ligand_random.mol2 #ligand file (random starting orientation)

```
# output

output_dir ./results

write_protein_conformations 0

# write single mol2 files (e.g. for RMSD calculation)

write_multi_mol2 0

# binding site definition

bindingsite_center x y z

# center of mass of residues-lining cavity

bindingsite_radius r

# cluster algorithm

cluster_structures 100

cluster_rmsd 0.25
```

General conclusions

The recent progress of machine learning using deep neural architectures has raised exceptional interest in the possible application of these techniques in many multidisciplinary fields, including drug design, where development of scoring functions for molecular docking still remains an actual problem. In addition to that, the development of a lightweight and accurate method for binding affinity prediction in protein-ligand complexes, would be beneficial for both virtual screening campaigns and lead optimization stages of drug development. As a result of the grown interest of the community in neural network approaches, a variety of neural network-based models aimed at binding affinity prediction from protein-ligand structures emerged in the recent years, which rely on different architectures and achieve remarkable performance on common benchmarking sets.

Due to their relevance in structure-based drug design campaign and their theoretical capability of generalization, structure-based binding affinity prediction models have caused our exceptional interest. The work presented in the scope of this doctoral thesis included the development of a binding affinity prediction model, relying of one of the most prominent architectures applied in this field – message passing neural networks (MPNN), operating on graph representation of the protein-ligand complex structure. We performed the assessment of its performance on commonly used and custom benchmarking sets and investigated the potential generalization problems coming from the training set memorization. As a result, we propose a binding affinity prediction model, which is comparable with the state-of-the-art in terms of performance on the PDBbind 2016 test set, stressing its potential limitations. In order to obtain a model suitable for virtual screening, we introduce a pipeline combining a docking pose classifier with a binding affinity predictor. In the perspective of further applications, the models, developed in the course of the current project, still needs a more complete assessment of ranking and screening performance, ideally of the datasets, developed with a consideration of previously reported biases in broadly used test sets such as the one, which is a part of the CASF benchmark.

In addition to that, it is important to stress, that for structure based binding affinity prediction with neural network model, the scarcity of data available for training remains an important issue. The possible directions of research, aimed at

overcoming of this limitation, can lie in the domain of development of new neural network architectures with higher generalization capability, but also in further expansion of experimentally retrieved structural datasets, not only in absolute values, but also in terms of increased density of coverage of potential protein-ligand combinations for promiscuous ligands, and the implementation of new data augmentation strategies for the improvement of model scoring performance.

Mikhail VOLKOV

Design and application of deep learning methods to structure-based drug design

Résumé

Prédire l'affinité d'un ligand pour sa protéine à partir d'informations structurales est un enjeu majeur pour le développement de médicaments. Malgré les récentes avancées dues à l'utilisation de machines d'apprentissage, de nombreux biais restent à résoudre quant à la capacité de ces modèles à extrapoler à des complexes nouveaux. Dans cette étude, nous avons utilisé une architecture de réseaux de neurones profonds à transmission de message (MPNN) qui lit directement les graphes moléculaires du ligand, de la protéine et des interactions protéine-ligand. Les biais des divers modèles développés ont été évalués, concernant notamment la composition des jeux d'entrainement et de test, la densité des matrices d'apprentissage et la complexité des graphes d'interaction. Nous avons enfin appliqué des modèles non biaisés à la prédiction d'affinité à partir de poses de docking en couplant un modèle binaire de pertinence de pose de docking et un modèle de prédiction d'affinité pour des applications de criblage virtuel.

Mots clés: affinité, fonctions de scores, in silico, machine d'apprentissage, réseaux de neurones de graphes, docking, biais.

Résumé en anglais

The accurate binding affinity prediction of protein-ligand complexes from structural data remains a relevant problem in drug discovery. Despite the recent progress in the development of machine learning-based scoring functions relying on novel neural network architectures, there are multiple evidences of potential biases in commonly used training and test sets, which lead to limited generalization capabilities of these models. In the current study we propose an MPNN-based neural network architecture operating on graph inputs of protein, ligand, and interactions derived from the structural data, as well as on their combinations. The potential biases of our models are evaluated with consideration of factors such as training set composition and sparsity, training sample memorization, ligand buriedness and graph complexity are also evaluated. We also introduce a pipeline consisting of a docking pose classifier and binding affinity predictor for potential virtual screening applications.

Keywords: binding affinity prediction, scoring functions, *in silico*, machine learning, graph neural networks, docking, bias