Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Accurate Protein-Ligand Complex Structure Prediction using Geometric Deep Learning

Junfeng Zhang (jfzhang@nju.edu.cn)
Nanjing University
Kelei He
Nanjing University https://orcid.org/0000-0002-7264-9437
Tiejun Dong
Nanjing University https://orcid.org/0000-0003-3434-3370
Jinhui wu
Nanjing University https://orcid.org/0000-0001-8809-1290

Biological Sciences - Article

Keywords:

Posted Date: March 21st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1454132/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Accurate Protein-Ligand Complex Structure Prediction using Geometric Deep Learning

Kelei He[1,2†], Tiejun Dong[2,3†], Jinhui Wu[1,3*] and Junfeng Zhang[1,2,3*]

[1]State Key Laboratory of Pharmaceutical Biotechnology, Medical School of Nanjing University, Nanjing, P. R. China.
[2]National Institute of Healthcare Data Science, Nanjing University, Nanjing, P. R. China.
[3]School of Life Sciences, Nanjing University, Nanjing, P. R. China.

*Corresponding author(s). E-mail(s): jfzhang@nju.edu.cn; wuj@nju.edu.cn;
Contributing authors: hkl@nju.edu.cn; tiejundong@smail.nju.edu.cn;
†These authors contributed equally to this work.

Abstract

Understanding the structure of the protein-ligand complex is crucial to drug development. However, existing virtual structure measurement methods are mainly docking and its derived methods combined with deep learning, which have restricted performance and efficiency due to their sampling and scoring methodology. Here we show the complex structure can be directly predicted using our proposed LigPose based on geometric deep learning in an end-to-end manner. By representing the ligand and the protein as a complete graph, LigPose optimizes the 3-D structure of the complexes with their atom coordinates in the Euclidean space. LigPose achieved state-of-the-art performance on two major tasks in drug development, i.e., complex structure prediction and affinity estimation, indicating a promising paradigm of predicting the protein-ligand complex structures in drug development, with improved capacity far beyond popular docking tools.

1 Introduction

Small organic molecule (SOM) plays an important role in clinical treatment, accounting for about 70% FDA-approved drugs in 2015 ∼ 2020 [1]. Its efficacy is achieved by binding to the target (usually a protein) as a ligand, to produce a protein-ligand complex. Understanding the structural details of the protein-ligand complexes at the atomic level reveals the bioactivities of the proteins and ligands, thus guiding the structure-based drug development, *e.g.*, drug screening [2, 3, 4] and lead optimization [4, 3, 5]. Conventional methods use experimental measurements (*e.g.*, X-ray diffraction [6] and cryo-electron microscopy [7]) to analyze novel protein-ligand complexes, however, are time and resource expensive.

To alleviate this problem, virtual measurement by molecular docking [8, 2, 9, 4] has been widely adopted in the past decades to predict the native-like ligand-binding conformations with the respective protein-binding sites. Typically, a docking tool first samples a set of binding conformations (namely poses), and then ranks them using a scoring function to select a top-scored pose [9] (Fig. 1**b**). However, popular docking tools generated the native-like poses with accuracies from approximately 40% to 60% in terms of success rate [10], which were far from satisfactory. Since deep learning has impacted the field of drug development, many researchers attempted to build hybrid methods by regarding it as a more expressive scoring function to rank the poses sampled by conventional docking tools [11, 12, 13, 14, 15, 16]. Nevertheless, the performance of these hybrid methods has a constrained upper bound, as limited by the sampling time and space of docking tools [15].

Very recently, several deep learning methods, *i.e.*, AlphaFold [17] and RoseTTAFold [18], have shown great capacity to predict protein structures, that outperformed previous methods by a large margin. These methods provided a novel computational approach to generate protein structures directly from their chemical sequences, rethinking the structure prediction paradigm in a data-driven perspective without following the conventional force-field assumption [19]. Recent progresses [20, 21, 22, 23] demonstrated that using these methods as the protein/peptide structure estimator can predict precise structures for amino acid-based complexes such as the protein-protein and protein-peptide complexes, showing their generalizability to the downstream tasks. However, these methods are not specifically designed for SOM ligands, which inevitably restricted the related applications such as protein-ligand complex structure prediction and affinity estimation. As inspired by these observations, developing a novel deep learning method to directly predict the protein-ligand complex structure may overcome the aforementioned limitation of existing methods.

To this end, we introduce LigPose, a novel docking-free deep learning method to accurately predict the protein-ligand complex structures (Fig. 1**a**). LigPose is capable of predicting the native-like conformation of ligands with their corresponding protein targets in an end-to-end manner. Specifically, for a given protein and ligand pair, the ligand and its binding target are jointly

represented as a complete undirected graph, with each node denoting an atom, and all nodes are mutually connected. Then, their 3-D structures are directly optimized by the atom coordinates in the Euclidean space (Fig. 1**b**).

LigPose achieved state-of-the-art performance on two major tasks in drug development, *i.e.*, structure prediction and affinity estimation. Notably, LigPose outperformed 12 existing docking tools with a significant improvement of 15% in terms of success rate, with an up to 265x faster inference speed, accelerating by only one ordinary GPU device. In addition, the extensive analysis showed that LigPose has the interpretability to learn non-covalent interactions even without explicitly incorporating the physical or chemical priors, *e.g.*, the energies, and the atom interactions, which are usually regarded as the core idea of building structure prediction methods. The superior performance of LigPose suggests a novel promising paradigm for structure-based drug design using deep learning. We hope this work can facilitate the research of drug development.

2 Results

2.1 LigPose pipeline

LigPose predicts the 3-D structure of the ligand-binding conformation with its protein target in an end-to-end manner. As shown in Fig. 1**c**, a given protein and ligand are jointly represented by a complete undirected graph, where each atom was denoted as a node, representing its chemical feature and coordinate, and all nodes are mutually connected. Since the entire protein-ligand graph is large, LigPose (Fig. 1**d**) first adopts a sampling and recycling strategy, to predict with multiple cycles, where each cycle updates a randomly sampled sub-graph. The features and coordinates within the sub-graph are then processed by the proposed feature and coordinate update blocks. These two blocks are built on the graph neural network, and stacked 8 times with unshared weights. The key design of these two blocks is leveraging the inter-atom distances during network forwarding to maintain the spatial information, making the predicted atom coordinates are insensitive to their initial positions, but influenced by the inter-atom correlations. Finally, a symmetric-aware loss is proposed to optimize the network, combined with a stochastic coordinate initialization strategy for the ligands, to enable the method of distinguishing the ligand atoms with the same chemical features. Please refer to Methods, Suppl. Methods and Fig. 1**e-j** for more details of the method.

2.2 Predicting accurate complex structures

In experiments, we first focus on the flexible-ligand prediction with the determined protein pockets, a widely used setting in structure-based drug development [25], to demonstrate its effectiveness. We compare LigPose with 12 popular docking tools on the refined set of PDBbind database [26, 10]. The

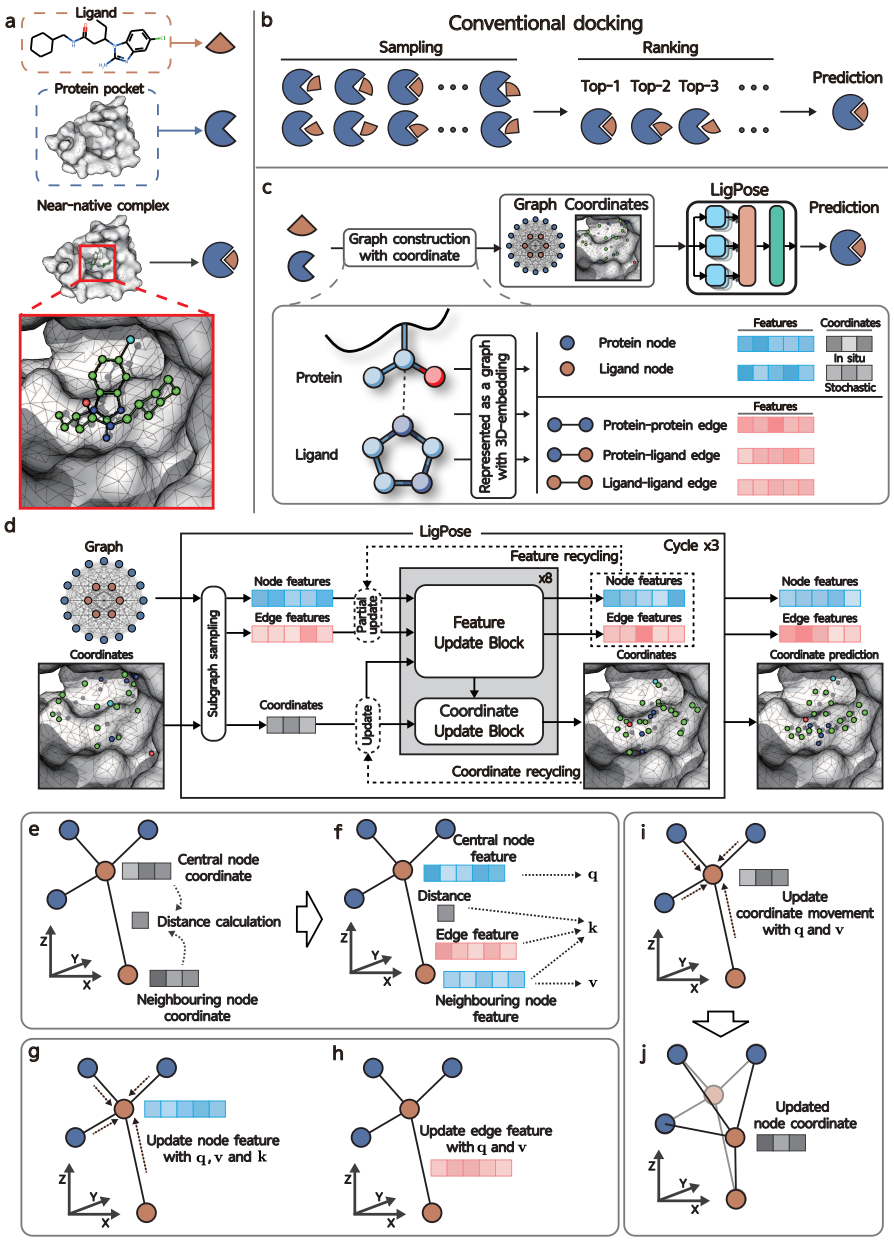
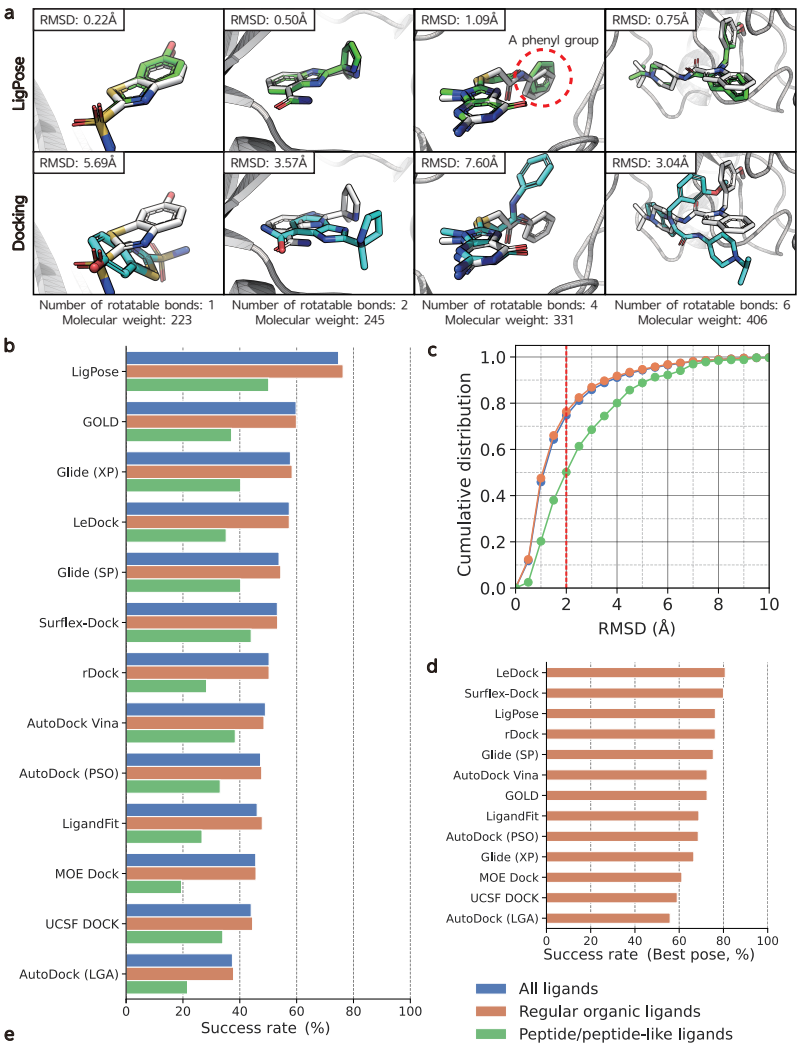


Fig. 1 LigPose predicts ligand-protein complex structure using geometric deep learning in an end-to-end manner, compared with the conventional docking method. (**a**) Notations of a ligand, a protein pocket and their complex. (**b**) Pipeline of the conventional docking tool, consisting of two stages, *i.e.*, sampling and ranking [9]. (**c**) Pipeline of our method. (**d**) Architecture of LigPose. (**e-h**) Updating features of the central node and its related edges by feature update block. (**e**) Calculating pair-wise distances. (**f**) Calculating attention weights in feature update block. Arrows show the information flow. (**g**) Updating central node features with message aggregation. (**h**) Updating edge features. (**i-j**) Updating coordinates by coordinate update block. (**i**) Updating central node coordinates. (**j**) The updated coordinate of the central node.



e

Methods	Success rate (%)	Description
Smina [24]	67.72	Docking tool
Autodock Vina [14]	67.90	Docking tool
Glide SP [11]	78.86	Docking tool
InteractionGraphNet [11]	74.25	Docking tool + deep learning-based scoring function
DeepBSP [14]	79.70	Docking tool + deep learning-based scoring function
DeepDock [12]	62.11	Global optimization + deep learning-based distance likelihood
LigPose^{score}-Core	77.54	Scoring function version of LigPose
LigPose	**82.46**	End-to-end prediction

Fig. 2 Performance of LigPose on ligand-binding conformation prediction compared with popular molecular docking tools and hybrid deep learning methods. (**a**) Visualization of the generated poses of LigPose and a docking tool (Smina [24]) for four ligands with various weights and number of rotatable bonds (PDB codes: 1ZFQ, 3KJD, 5V79, 2BQ7). The native poses, predictions of LigPose and docking tools are denoted as white, green and cyan backbones, respectively. Within the predictions, the oxygen and nitrogen atoms are denoted as red and blue colors, respectively. (**b**) Quantitative comparison of success rate between LigPose and the top-scored poses generated by 12 docking tools on the refined set of PDBbind. (**c**) Cumulative distribution of RMSD of LigPose. The red dashed line indicates the RMSD threshold of 2Å. (**d**) Quantitative comparison of success rate between LigPose and the best poses generated by the docking tools. Blue, orange, and green colors denote all ligands, the regular organic ligands, and the peptide/peptide-like ligands, respectively. (**e**) Quantitative comparison of success rate between LigPose, the docking tools and the hybrid deep learning methods on the core set of PDBbind.

PDBbind database collects a large set ($N = 19443$) of 3-D biomolecular complex structures from PDB database. The refined set ($N = 5316$) is derived from PDBbind with high data quality regarding the crystal structures, binding data, and the nature of complexes [27].

The visualization of the predicted poses in Fig. 2**a** shows LigPose is able to predict accurate poses for ligands of various flexibilities on those are failed by docking tools. It is worth noting that the symmetric structures (*i.e.*, atoms in a phenyl group) are predicted in the right place. The quantitative comparison of the success rate between LigPose and 12 popular docking tools with top-scored poses is shown in Fig. 2**b**. The predicted poses with Root Mean Square Deviation (RMSD) less than 2Å to their native poses are regarded as successful predictions [10]. As suggested by the figure, LigPose obtains a high success rate of 74.7%, making a significant improvement of 14.9% in comparison to the best record docking tool (GOLD [28] with 59.8%). The other docking tools have unsatisfied success rates, ranging from 37.4% (AutoDock (LGA) [29]) to 57.8% (Glide (XP) [30]). In addition, we observe the peptide/peptide-like molecules with more rotatable bonds and weights are more difficult to predict than the regular organic ligands, leading to a performance decrease for both LigPose and docking tools. Considering the predicted poses with RMSD < 4Å(Fig. 2**c**), LigPose achieves the success rate of above 90% on regular organic molecules, shown effectiveness on near-native structure prediction for conformations.

We also report the performance of LigPose on regular organic molecules compared with the docking tools with best poses in Fig. 2**d**. Herein, the best pose denotes the pose of the lowest RMSD among all sampled poses, and reveals the performance upper bound of the docking tools. The figure shows the success rate of LigPose (76.3%) was slightly lower than the best-performed docking tool (LeDock [31] with 80.8%). It is worth noting that LigPose outperformed 10 docking tools, indicating that these docking tools could not have a chance to outperform LigPose by improving the scoring functions [15], since they cannot produce corrected pose candidates in limited sampling steps for some ligands that are correctly predicted by LigPose. Additionally, LigPose outperformed 8 docking tools for peptide/peptide-like molecules (Suppl. Fig. A2).

To conduct a fair comparison with the deep learning-based methods, we performed LigPose on the core set ($N = 285$), a well-designed but smaller subset of PDBbind. Here we summarized the success rates of LigPose with diversified methods, including docking tools and state-of-the-art deep learning-based methods [14, 11, 12] in Fig. 2**e**, where LigPose obtained the success rate of 82.5%, which is higher than the best-performed method DeepBSP (= 79.7%), indicating the high performance of the proposed end-to-end paradigm.

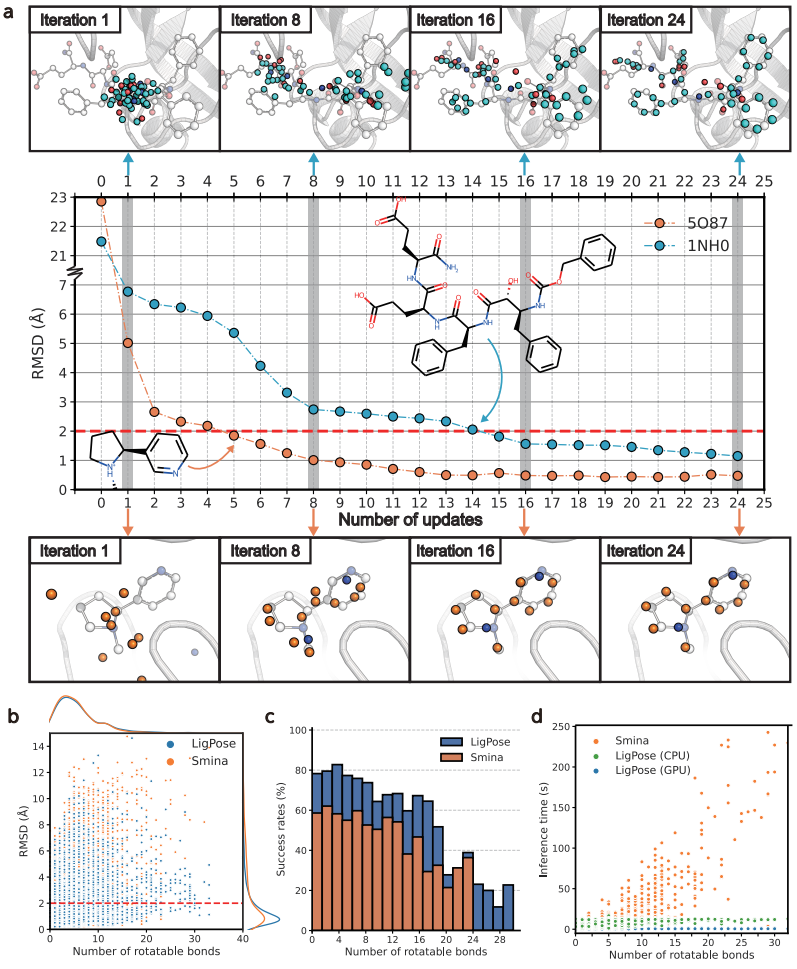


Fig. 3 Performance of LigPose with respect to the ligand flexibility. (**a**) RMSD trajectory of two representative samples (PDB codes: 5O87 and 1NH0). LigPose updates a given ligand 24 times using 3 cycles, with each cycle updating the ligand 8 times. The ligand atoms in the $\{1, 8, 16, 24\}$th updates are visualized at the top and the bottom of the panel. The predicted atoms are denoted with orange and cyan colors for 5O87 and 1NH0, respectively. The oxygen and nitrogen atoms are denoted with red and blue colors, respectively. The native poses are placed in the background with a grey color. (**b-c**) RMSD (**b**) and success rate (**c**) for ligands with respect to the number of rotatable bonds on the refined set of PDBbind. The red dashed line indicates the RMSD threshold of 2Å. (**d**) Inference time of LigPose and Smina on 1000 randomly selected samples in the refined set.

2.3 Ligands with various flexibilities

To solve the ligands with high flexibilities (*i.e.*, with many rotatable bonds), the strategies of sub-graph sampling and recycling are combined to progressively refine the predictions. We visualize the ligand atoms and plot the RMSD

trajectories for two representative samples in Fig. 3**a**, *i.e.*, one is heavier with more rotatable bonds (PDB code: 1NH0) than the other (PDB code: 5O87). As a hard example, the prediction of 1NH0 reaches the threshold of 2Å after 15 updates, which is much slower than that of 5O87 (with 5 updates), indicating the effectiveness of the proposed recycling strategy.

To quantify the performance of LigPose with respect to the number of rotatable bonds, we plot the results of LigPose and Smina on the refined set in terms of RMSD (**b**) and success rate (**c**) in Fig. 3. One can observe is the difficulty of prediction is positively correlated with the number of rotatable bonds. We suppose the reason is that more rotatable bonds indicate higher flexibility of the ligand. Although the success rates of LigPose and Smina are both decreased when predicting ligands with many rotatable bonds, LigPose performed consistently better than Smina through the dataset. Moreover, Smina totally failed to predict with more than 24 rotatable bonds, indicating the limited processing power of the docking methods. By contrast, LigPose correctly predicted 20% of them, showing the merit of our learning-based methodology. Similar findings were discovered from the core set, as shown in Suppl. Fig. A3.

The efficiency is also a core factor for drug development [32], caused by a large number of drug-like molecules for screening and measurement. We then plot the inference time per ligand with respect to the number of ligand rotatable bonds for LigPose and Smina in Fig. 3**d**. For a fair comparison, we use the same device to implement Smina and LigPose (CPU), with only an additional common GPU device (Nvidia GTX 2080Ti) for the GPU version (LigPose (GPU)). From the figure, we observed the inference time of LigPose is constant, in contrast to that of Smina, which is positively related to the number of rotatable bonds. On average, LigPose inferences 6 (CPU) / 111 (GPU) times faster than Smina, for ligands with more than 10 rotatable bonds, and up to 265x faster for ligands with more than 20 rotatable bonds.

2.4 Using as a scoring function

Here, we show an extended application of LigPose, *i.e.*, treating it as a scoring function to build LigPose$^{\text{score}}$, by directly choosing the pose candidate that best matches the prediction of LigPose using the metric of RMSD (see a brief illustration in Fig. 4**a**). Thus, LigPose$^{\text{score}}$ can be composed with docking, and also evaluated on the Comparative Assessment of Scoring Functions (CASF-2016) benchmark [33] derived from the core set. The benchmark is carefully designed to evaluate the capacity of scoring functions, with pose candidates being the clustering centers of docking sampled poses, using RMSD and conformation similarity as the clustering index.

As suggested by Fig. 4**b**, our method (denote as LigPose$^{\text{score}}$-CASF) with a success rate of 86.0% outperformed 34 conventional scoring functions (with success rates ranging from 26.0% to 84.9%), and was also comparable to the state-of-the-art deep learning-based scoring functions, *i.e.*, DeepBSP (87.2%) [14], DeepDock (87.0%) [12], and PIGNet (87.0%) [34]. We conclude the slightly lower performance of LigPose$^{\text{score}}$ for two reasons. First,

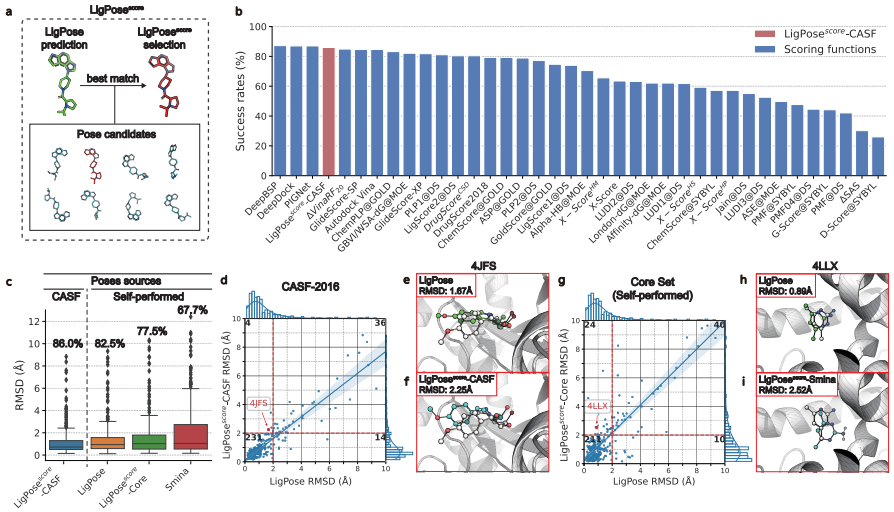


Fig. 4 LigPose used as a scoring function. (**a**) Brief illustration of LigPose^{score}, which simply chooses a pose candidate that best matches the prediction of LigPose as the top-scored pose. (**b**) Comparison of success rates between LigPose, LigPose^{score}-CASF, and 37 well-established scoring functions on CASF-2016 benchmark. (**c**) RMSD distribution and success rate of Smina, LigPose, and LigPose^{score}-Core, and LigPose^{score}-CASF. (**d,g**) Correlation plots of RMSD between predictions of LigPose and LigPose^{score} on CASF-2016 (**d**) and self-performed poses (**g**), respectively. Each of the figure is partitioned into four regions by two red dashed lines at RMSD of 2Å, with the sample count marked in the corner. (**e,f,h,i**) Visualization of two representative samples as pointed out in **d** and **g**. Prediction of LigPose (**e,h**), LigPose^{score} (**f,i**), and the native poses (**e,f,h,i**) are denoted as green, cyan and white color, respectively. The oxygen and nitrogen atoms are denoted as red and blue color, respectively.

LigPose^{score} still needs to generate a near-native pose, instead of explicitly scoring and ranking the pose candidates, which improves the difficulty of the task. Second, the CASF-2016 benchmark is inadequate to assess the structure prediction ability for real production purposes. Since the RMSD is inaccessible in real-world applications, the native complex structure could not be predetermined during virtual measurement. As previously discussed for Fig. 2**e**, several works use self-performed poses to partially solve the problem. We adopt this setting and evaluate LigPose^{score} on self-performed poses, by performing Smina on the core set (denote as LigPose^{score}-Core). The RMSD distribution of Smina, LigPose^{score}-Core, and LigPose are shown in Fig. 4**c**, with LigPose^{score}-CASF as the baseline. As shown in Fig. 2**e** and Fig. 4**c**, without selective poses, the success rates of all methods are decreased compared with those in CASF-2016, ranging from 62.1% to 82.5%. LigPose^{score}-Core showed a greatly improved success rate (77.5%) compared with Smina (67.7%), and was also comparable to the best performing scoring function (*i.e.*, DeepBSP (79.7%) [11]). Moreover, LigPose (82.5%) outperformed DeepBSP and LigPose^{score}-Core with the notable higher success rates of 2.8% and 5%, respectively,

indicating that the end-to-end learning paradigm of LigPose is more effective in real applications without pre-designing the pose candidates.

Furthermore, we plotted the correlation of RMSD in Fig. 4, between LigPose and LigPose^{score}-CASF (**d**) / -Core (**g**), each with a typical sample visualized by the predicted structures aside (**e,f,h,i**). We observed from the figure that docking may mislead the model. For some cases that can be correctly predicted by LigPose (*i.e.*, with RMSD < 2), LigPose^{score} reshaped them to wrong structures, especially on self-performed poses (with more samples in the upper-left regions in **g** compared with **d**). These results also indicate that docking-based sampling and scoring have inevitably ignored limitations for structure prediction in real practice.

2.5 Affinity estimation

Table 1 Performance comparison for affinity estimation on the core set of PDBbind. Methods that require native complex structures were listed in the second set of rows. Methods without using the native complex structures were listed in the last set of rows. The term "Native" indicates the methods use the native complex structures for affinity estimation, "Docking" indicates they use the docking-sampled complex structures, and "none" indicates they use none of the above complex structures.

Models	MAE	RMSE	Pearson R	Using Complex Structure		
				Native	Docked	None
Pafnucy [35]	1.13	1.42	0.78	✓		
TopologyNet [36]	-	1.34	0.81	✓		
FAST [37]	1.019	1.308	0.810	✓		
OnionNet[38]	0.984	1.278	0.816	✓		
InteractionGraphNet [39]	0.940	1.220	0.837	✓		
PointTransformer[40]	0.91	1.19	0.852	✓		
FAST [37]	1.498	1.871	0.712		✓	
OnionNet [38]	-	1.523	0.755		✓	
InteractionGraphNet [39]	1.150	1.503	0.757		✓	
APMNet [41]	0.998	1.268	0.815			✓
LigPose^{affnity}	0.896	1.198	0.835			✓

To further investigate the scalability of LigPose, we also apply it to the task of protein-ligand affinity estimation (denote as LigPose^{affnity}), another core function of docking. For disease-related proteins, ligands with strong binding interactions (*i.e.*, affinity) can be selected as drug candidates. LigPose is feasible to adapt to this task without leveraging native complex structures, by simply adding several task-specific output layers (see Suppl. Sec. A.2.4 for details). The comparisons of LigPose^{affnity} with several recent structure-based deep learning methods [37, 39, 35, 36, 41, 38, 40] on affinity estimation are summarized in Table 1, where LigPose^{affnity} achieved the best performance among the competing methods which do not use native structures, with mean square error (MAE) of 0.896, root mean square error (RMSE) of 1.198 and

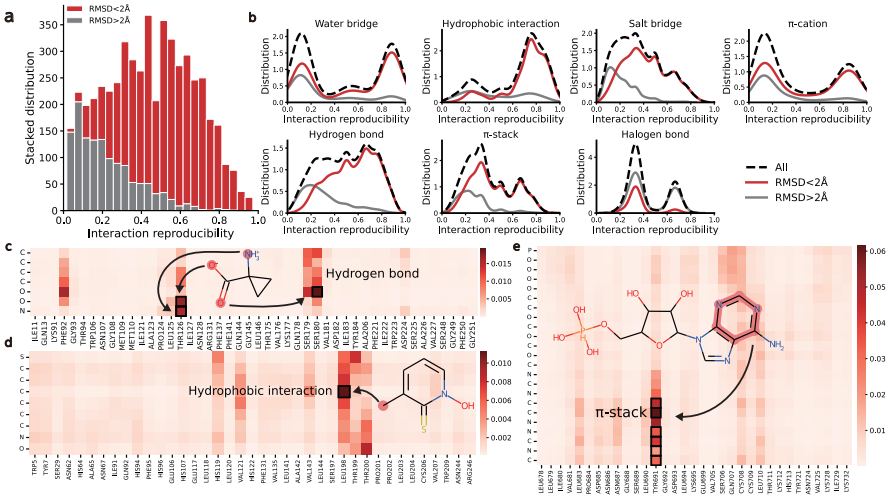


Fig. 5 LigPose implicitly learned non-covalent interaction patterns. (**a**) Stacked distribution of interaction reproducibility for all non-covalent interactions on the refined set. The red and grey colors denote RMSD<2Å and RMSD≥2Å, respectively. (**b**) Distribution of the interaction reproducibility for seven specific non-covalent interactions. Distribution of all samples, the samples of RMSD<2Å and RMSD≥2Å are shown in black, red and grey dashed lines, respectively. (**c-e**) Visualization of attention weights in LigPose for three representative protein-ligand complexes. The attention weights with the red color from light to deep indicate the probability of non-covalent interaction between a ligand atom and a protein residue from low to high. Non-covalent interactions of native structures are denoted by black boxes in the attention maps for hydrogen bonds (PDB code: 1Y20) (**c**), hydrophobic interactions (PDB code: 4Q7P) (**d**), and π-stack (PDB code: 4G0Y) (**e**). Attention weights are visualized in atom-level for ligands and residue-level for proteins.

Pearson R of 0.835. Compared to the methods that use native structures, LigPose$^{\text{affnity}}$ is also comparable or better than the best performing method PointTransformer (MAE: 0.91, RMSE: 1.19, Pearson R: 0.852), indicating its strong ability to learn native-like atom interactions. Therefore, without using native complex structures, LigPose$^{\text{affnity}}$ can achieve impressive results that largely outperformed other methods.

2.6 Learning non-covalent interactions

In addition to the effectiveness and efficiency, LigPose also showed good interpretability by investigating its ability to reconstruct the non-covalent interactions. We used a metric, namely interaction reproducibility, to quantify this ability, following the work in [42] (see Methods for details).

From Fig. 5**a**, we observed the interactions are well reconstructed by LigPose on predictions with RMSD<2Å, and have lower scores on predictions with RMSD≥2Å. With respect to each type of interaction (Fig. 5**b**), LigPose performed best for salt bridge and hydrogen bond, and showed insensitive to halogen bond. We suppose the reason is the lack of respective ligand samples. We further compared the performance of LigPose with Smina (Suppl. Figs. A4

and A5), and found that LigPose showed consistently better performance for all the seven types of interactions.

In addition to the high quantitative scores, we further inspected the attention weights, by visualizing three representative complexes with respect to three non-covalent interactions (*i.e.*, hydrogen bonds, hydrophobic interactions and π-stack) in Fig. 5(**c-e**). We observed from the figure that LigPose can capture some potential interactions between ligands and proteins, without explicitly using related physical or chemical prior knowledge.

3 Discussion

In this work, we propose LigPose, to predict protein-ligand complex structure using geometric deep learning. The complex is first represented by one complete undirected graph, then, LigPose with a graph transformer network backbone, directly predicting the complex structure in 3-D space. To precisely predict the complex structure with atomic accuracy, LigPose is equipped with three innovations. To be specific, 1) to handle a large number of protein nodes, we use a sampling and recycling strategy to optimally learn the protein information by multiple cycles while reducing the computation burden. Then, 2) in each cycle, the feature update block is used to extract latent features for graph nodes and edges, and the coordinate update block updates the coordinates of all atoms (nodes) to their near-native 3-D position. Besides, 3) we design a specific atom initialization strategy and a symmetric-aware loss function for symmetric structures of the ligands. These advances obtained a notable improvement for the tasks of structure prediction and affinity estimation.

LigPose correctly predicted 3945 (75%) ligand structures (with the RMSD < 2) on the refined set of PDBbind, which significantly outperformed 12 popular docking tools by 15%. LigPose also performed better than recent hybrid deep learning methods on the core set. Many researches in this field focused on building hybrid methods [13, 15, 14, 11, 43, 16]. However, as we demonstrated in experiments (Fig. 2**e**), these methods are limited by the sampling power of the docking tools [15]. Indeed, exhaustively exploring the sampling space dramatically increases the runtime, making it impractical in real-world drug screening. Recently, Oscar *et al.*[12] proposed DeepDock, which used geometric deep learning to build a docking-like method. Interestingly, this method could be regarded as a deep learning-based scoring function from a global perspective. However, this method in a docking-like fashion will also face the same challenges to docking tools in both effectiveness and efficiency. For instance, they reported that 60 out of 285 samples were failed to find a minimum, and only 62% (177) out of 285 samples were successful predicted on the core set, leading to unsatisfied performance even worse than conventional docking methods (Fig. 2**e**). By contrast, LigPose overcomes these limitations by directly optimizing the atom coordinates without sampling and energy-based minimizing, suggesting a novel computational paradigm for the prediction of the protein-ligand complex structures, coincident with AlphaFold in the task

of protein structure prediction [17]. Moreover, LigPose enjoys other characteristics in drug development, *e.g.*, the state-of-the-art affinity estimation ability without leveraging the native complex structures, showing its feasibility to real applications.

In addition to the high performance, LigPose also inferences considerably faster than docking. In real-world drug development, methods are expected to accurately screen a tremendous number (*i.e.*, up to 10^{60} [44]) of drug-like molecule candidates [2, 45]. Unfortunately, the docking-based methods with an iteratively sampling strategy have been suffered from the efficiency problem for a long time. Many studies tried to solve the problem by implementing CPU/GPU parallel acceleration algorithms in favor of engineering [46, 47, 48]. Besides, some works combined deep learning with the conventional docking tools to boost the inference speed [49, 50]. We suggest that LigPose with an efficient end-to-end learning framework could be an alternative choice for docking to accelerate drug development.

Building an accurate automatic drug development pipeline has encouraged decades of exploration [51, 19]. Recent advances demonstrated that protein structures can be accurately predicted by AlphaFold and RoseTTAFold [17, 18]. Therefore, predicting the structure of biomolecular complexes could be naturally considered as a subsequent application, as also suggested by Baek *et al.*[19]. In addition, the task of identifying the protein pocket has been well-addressed by existing works [52, 53, 54, 55]. Inspired by these successes, integrating LigPose with the aforementioned methods is obviously appropriate, carrying out a promising end-to-end pipeline of predicting a protein-ligand complex structure directly from the protein sequence and the ligand chemical formula.

Moreover, in the field of structural biology, recent researches have extended the applicable scenario of artificial intelligence to design the protein structures [56, 57]. The similar idea could also be applied to drug development. However, existing methods proposed to design (or generate) novel drug candidates neglected to consider either the 3-D structures or the protein-drug interactions [58, 59, 60, 61, 62, 63, 64, 65, 66]. Here, we suggest that LigPose has the potential to build novel end-to-end drug-generation methods with both refined structural details and confident bioactivity prediction, to boost de novo drug design, which can be regarded as the future work.

4 Methods

4.1 Dataset

The *general set* derived from PDBbind database (version 2020) [26] is adopted by this work, consisting of 19443 protein-ligand complexes. Within the general set, two subsets are used for evaluation, *i.e.*, the *refined set* and the *core set* that consists of 5316 and 285 complexes, respectively. These two subsets are of better data quality for evaluating docking tools and scoring functions in existing works. Additionally, the Comparative Assessment of Scoring Functions 2016 (CASF-2016) benchmark based on the core set, with a careful design of poses, is used to assess the performance of LigPose and the scoring functions. For a fair comparison, we remove the native poses in this dataset.

We train LigPose with the general set, except the protein-ligand pairs included in the refined set and the core set, which are used for evaluation. In other words, the samples for training and evaluation are non-overlapped. In addition, the peptide-like molecules are identified with the Biologically Interesting Molecule Reference Dictionary (BIRD) [67]. Only a minor proportion of the data ($152, < 1\%$) in the general set is not successfully processed by RDKit [68], and is removed from the training set.

4.2 Pre-processing

In this work, the protein pocket was adopted as the binding target, since obtaining the atomically refined structure is the major concern in drug development [69, 4]. Besides, many methods have been successfully developed to predict the ligand-binding pockets with high accuracies [52, 53, 54, 55, 70, 71]. The protein pockets are the amino acids surrounding the ligands in 3-D space, as predefined by the PDBbind database. For a given protein-ligand complex, a complete undirected graph is firstly constructed using RDKit [68], with each node representing an atom, and all nodes (including themselves) are mutually connected. Additionally, for each ligand, we divide it into several small fragments at rotatable chemical bonds.

We initialize the graph by features listed in Suppl. Table A2. In particular, the nodes are initialized as vectors with a length of 79 for a protein, and 45 for a ligand. Edges are also initialized as vectors with a length of 7, in which one dimension is used to encode the distances between the two connected nodes. The distances are initialized by the following three rules. 1) For edges connecting the protein and ligand nodes, the distances are initialized with -1. 2) For edges in the same fragment or chemical bond within the ligands, the distances are initialized with their actual distances. 3) For edges connecting different fragments, the distances are initialized with -1. To enable the ability of position equivalence for the method, and also augment the data, we implement a stochastic strategy to initialize the coordinates of the ligands. In this case, the protein pocket nodes are kept in their original positions, and ligand nodes are randomly placed inside a 4nm cubic box at the center of the pocket.

4.3 LigPose architecture

LigPose directly predicts the structure of the ligand-binding conformations with their target protein pockets in the 3-D space in an end-to-end manner. As shown in Fig. 1**d**, LigPose has a graph transformer architecture, with three major components, (1) a sampling and recycling strategy, (2) a feature update block followed by a coordinate update block to forward the features and coordinates through the graph, and the two blocks are stacked 8 times with unshared weights, and (3) a randomly initialization method for ligand nodes and a novel symmetric-aware loss. We will then illustrate them below.

4.3.1 Sampling and recycling

A protein pocket contains about 400 atoms on average. Using all of them as the input is inefficient and memory unaffordable for common GPU devices. Therefore, we adopt a sampling strategy to generate a sub-graph consisting of all *core atoms* and some randomly selected *context atoms*, with their respective edges, to feed to the network. Specifically, the core atoms involve all ligand atoms, and the Cα, Cβ atoms of the protein, as they are enough to determine the position and orientation of the amino acids and protein backbone [72, 73]. The other protein atoms are regarded as the context atoms to describe the structural details of the amino acids. Since the nodes are not fully used, we further introduce a recycling strategy, to enhance the representability of the method, as inspired by [17]. In each cycle, a new sub-graph is sampled and forwarded, then, the updated graph is reused in the next cycle. In the next cycle, the newly sampled graph directly inherits the coordinates of corresponding nodes from the last cycle, and its features of the core atoms are combined with the last updated features by using an element-wise gate on the new features, implementing a partial update. (See details in Suppl. Sec. A.2.2)

4.3.2 Feature Update Block

The feature update block maintains a graph Transformer-based architecture to forward the node and edge features. It iteratively performs two operations, *i.e.*, message aggregation and feedforward. In message aggregation, information are aggregated from neighbouring nodes to a central node using the multi-head attention (MHA) mechanism (Fig. 1**e**) [74]. In addition, the edge features are also incorporated to enhance the representation.

We use conventional cross-attention to calculate the MHA in the network, according to Query (denote as **q**), Key (denote as **k**) and Value (denote as **v**) [74] (See details in Suppl. Sec. A.2.2). Briefly, **q** and **v** are produced by the features of the central nodes and the neighbouring nodes, respectively (Fig. 1**f**). **k** is extracted from the edge feature and a linear transformed neighbouring node feature (Fig. 1**f**). The edge feature here is enhanced by the spatial information using the distance between the central node and its neighbouring node (Fig. 1**e**). A softmax layer is then applied to the product of **q** and **k** to get attention masks for each **v**. The central node aggregates the masked **v** to get an updated

feature, and is followed by a linear layer to merge multi-head information (Fig. 1**g**). An element-wise gate is implemented on these merged results. Finally, a two-layer multi-layer perceptron (MLP) is applied in the feedforward step, and is also enhanced by an element-wise gate. The edge feature is then updated with the element-wise production of **q** and **k** (Fig. 1**h**).

4.3.3 Coordinate Update Block

The coordinate update block updates the coordinates of the nodes (atoms) in 3-D space based on the attentions derived from the feature update block, as also introduced in [75] (Fig. 1**f**). To be specific, the attention is transformed to a one-dimensional distance gradient, indicating the change of distance between the central node and each of its neighbouring nodes. Then, the coordinate of a certain central node is updated by the sum of these distance gradients calculated with all neighbouring nodes, as written by,

$$\mathbf{\Delta}_h = \sum_{j \in Neighbor(i)} \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} \Phi(\mathbf{q}_{ij} \odot \mathbf{k}_{ij}), \tag{1}$$

$$\mathbf{x}_i^{'} = \mathbf{x}_i + \sum_{h=1,\ldots,N_h} \lambda_h \mathbf{\Delta}_h, \tag{2}$$

where $\mathbf{x}_i$ denote the coordinate of a central node i, $\mathbf{x}_j$ denote the coordinate of a neighbouring node j, $\mathbf{q}_{ij}$ and $\mathbf{k}_{ij}$ are the Query and Key for i and j, respectively, as obtained in the feature update block. The element-wise multiplication ($\odot$) of $\mathbf{q}_{ij}$ and $\mathbf{k}_{ij}$ are then transformed to a single distance variable by a linear layer Φ, and multiplied by the direction of a central node's coordinate ($\mathbf{x}_i$) to one of its neighbouring nodes' coordinate ($\mathbf{x}_j$). The coordinate update $\mathbf{\Delta}_h$ for node i is then obtained by aggregating $\mathbf{\Delta}$ for all i's neighbouring nodes ($Neighbor(i)$), where i is not equal to j. $\mathbf{x}_i$ is then updated by the weighted sum of all $\mathbf{\Delta}_h$ to $\mathbf{x}_i^{'}$. Finally, the coordinates were updated using MHA with λ_h weighted N_h heads.

4.4 Training

In each training iteration, we use the Monte Carlo method to choose one cycle to optimize the network parameters, as also introduced in [17] (Suppl. Sec. A.3). We used the Adam optimizer with a learning rate of 1×10^{-4}, and exponentially decayed by a rate of 0.99 for each epoch.

Given a ligand containing N_{lig} atoms (nodes). In each iteration, an index is created for mapping the nodes from the predicted pose to the native pose, ensuring the loss is calculated on atoms with the same chemical identity. Then, the loss for structure prediction is calculated by averaging the differential between the predicted and native ligand coordinates using the atom mapping (denote as $\mathcal{L}_{coor}$),

$$\mathcal{L}_{coor} = \frac{1}{N_{lig}} \sum_{i \in Nodes(lig)} \|\mathbf{x}_i^{pred} - \mathbf{x}_i^{true}\|_2. \tag{3}$$

where $Nodes(lig)$ denote the set of all ligand nodes. However, many ligands have symmetric structures. In such cases, multiple equivalent indexes can be created for one ligand (see an example in Suppl. Fig. A1). To solve the problem, we design a specific loss function, *i.e.*, the symmetric-aware loss (denote as $\mathcal{L}_{sym}$), to update the coordinates, which can be formally written as,

$$\mathcal{L}_{sym} = \min_{sym}(\mathcal{L}_{coor,s}), s \in \{1, ...N_s\}. \quad (4)$$

where $\min_{sym}(\cdot)$ calculates the losses for N_s equivalent indexes of the ligand, and chooses the minimum one. The final loss for one complex is calculated as a weighted sum of the following three terms: (1) an average of $\mathcal{L}_{sym}$ calculated with the outputs of all coordinate update blocks; (2) a $\mathcal{L}_{sym}$ calculated with the final coordinate prediction in the chosen cycle; and (3) an affinity loss (see Suppl. Methods for details).

4.5 Metrics

The Root Mean Square Deviation (RMSD) value between the predicted and native structures is used to evaluate the predictions. The RMSD value of 2Å is widely adopted in related works [10, 33] as a standard to determine the success or failure of the prediction, as also adopted in this work.

4.6 Configurations of docking tools

For docking tools, the ligands were firstly rotated 180° through the Z-axis with their original conformation [10], followed by energy minimization with Open Bable toolbox [76]. Then all proteins and ligands were converted to *pdbqt* files as described in [77]. The docking sites of the ligands were set by an automatic box creation tool [24] using their native poses. And the amount of generated poses for a certain ligand was set to 20, as also adopted in other works [13, 10, 15]. The number of CPU cores was set to 16 to calculate the inference time. The rest of the parameters were set by default. The outputs were analyzed using RDKit [68]. 379 (7.1%) protein-ligand complexes in the refined set were not successfully processed by the methods mentioned in this section, and were not included in the evaluation for docking tools. The remaining data showed a similar distribution of rotatable bonds to the data used by LigPose, therefore not affecting the fairness of the comparison (Suppl. Fig. A6).

4.7 Investigation of the protein-ligand interaction

Non-covalent interactions between proteins and ligands were recognized by PLIP [78], including water bridge, hydrophobic interaction, salt bridge, π-cation, hydrogen bond, π-stack and halogen bond. To better assess the non-covalent interactions between the predicted and native poses, we defined the

interaction reproducibility similar to [42], as written by,

$$\text{Interaction reproducibility} = \frac{1 + \text{number of shared interactions}}{2 + \text{total number of unique interactions}}. \quad (5)$$

The predicted ligands were placed to the original *pdb* file, and then processed by PLIP to calculate a score of interaction reproducibility. Specifically, the interactions were calculated in atom-level for ligands and in residue-level for proteins.

The visualization of attention weights for a single head was analyzed with the non-covalent interactions derived from true ligand pose, the weight of each protein residue was calculated by the average attention weight of Cα and Cβ nodes.

4.8 Data availability

All datasets used in this paper are freely available at PDBbind (http://www.pdbbind.org.cn/, version 2020), including the general set and the CASF-2016.

4.9 Code availability

The source code of LigPose will be available under an open-source license at https://github.com/KeleiHe/LigPose after acceptance.

Supplementary information. This paper accompanies supplementary information in the "Supplementary Material".

Acknowledgments. This work was supported in part by the National Nature Science Foundation of China under grant No. 62106101. This work was also supported in part by the Natural Science Foundation of Jiangsu Province under grant No. BK20210180.

Author contributions. K.H. and J.Z. led the research. K.H. and T.D. contributed the idea, developed the method. T.D. developed the data processing, analytics. K.H., T.D., J.W. and J.Z. wrote the paper.

Correspondence and requests for materials. should be addressed to Jinhui Wu and Junfeng Zhang.

Competing interests. The authors declare no competing interests.

Appendix A Supplementary Materials

A.1 Notations

We summarize the notations present in this paper in Table A1.

A.2 Methods

LigPose consists of three key components (main text Fig. 1**d**), 1) a sampling and recycling strategy, 2) a network backbone consisting of stacked feature and coordinate update blocks, and 3) a stochastic coordinate initialization combined with a symmetric-aware loss function. The implementation details of these components will be introduced below.

A.2.1 Sampling and recycling

To reduce the computational burden of handling a large number of protein atoms, we proposed a sampling strategy for protein atoms. Specifically, for a given graph representing a protein-ligand pair, a sub-graph including all ligand atoms, and the Cα and Cβ atoms derived from the protein pocket, is used to feed the model. These atoms are denoted as *core atoms*. Then, a set of atoms that are randomly sampled from the rest of the protein pocket atoms (denote as *context atoms*), are also included in the graph. The total number of core and context atoms is 500 in this work. All edges are assigned according to the real inter-atom relationship. Since the atoms of the complex are not fully used, which may weaken the representative ability of the method, we introduce a recycling strategy to compensate for it. In this case, the method is processed cycles for one complex, and within each cycle, the features and the coordinates of the complex are updated $N_l = 8$ times using the stacked feature and coordinate update blocks. Besides, in each cycle except the first cycle, the last updated features of the core atoms will be added to a newly sampled sub-graph with a same Ψ as defined in the feature update block, while the last updated features of the context atoms are ignored, to conduct a "partial update". The edge features are updated with the same principle. And the coordinates of the ligand nodes in the newly sampled sub-graph are directly assigned with their last updated values. Then, the sub-graph will be fed to the model again. The recycling will be performed N_c times ($N_c = 3$ in this work).

A.2.2 Feature update block

The feature update block extracts features of all nodes with a architecture based on Graph Transformer [79, 80]. Given an input graph $G = (V, E)$, the feature of the ith node $\mathbf{f}_i \in V$, and the feature of the edge between the ith node and its jth neighbouring node (denote as $\mathbf{e}_{ij} \in E$) are initialized as $\mathbf{f}_i^{init}$ and $\mathbf{e}_{ij}^{init}$, respectively, according to Table A2. Then, each feature is fed to a linear layer with respect to its feature type (*i.e.*, the protein node feature, the ligand node feature and the edge feature). The output features of the nodes (denote

as $\mathbf{f}_i^0$) are in a same size d_f. The output features of the edges are denoted as $\mathbf{e}_{ij}^0$ with size of d_e.

The features are then processed by number of $N_l = 8$ feature and coordinate update blocks. The feature update block conducts two operations, *i.e.*, message aggregation and feedforward. For a message aggregation step in the $l \in \{1, ..., N_l\}$th block, the central nodes aggregate the information from the neighboring nodes using both their features and coordinates. In particular, we calculate the Query (denote as $\mathbf{q}$), Key (denote as $\mathbf{k}$) and Value (denote as $\mathbf{v}$) in the block, as written by,

$$\mathbf{m}_{ij}^{h,l} = \mathbf{W}_{e,2}^{l}\text{Concat}(\mathbf{W}_{e,1}^{h,l}\mathbf{e}_{ij}^{l-1}, \mu\|\mathbf{x}_i^{l-1} - \mathbf{x}_j^{l-1}\|_2), \tag{A1}$$

$$\mathbf{k}_{ij}^{h,l} = \mathbf{W}_{k}^{h,l} f_j^{l-1} \odot \mathbf{W}_{e,3}^{l}\text{LeakyReLU}(\mathbf{m}_{ij}^{h,l}), \tag{A2}$$

$$\mathbf{q}_{ij}^{h,l} = \mathbf{W}_{q}^{h,l}\mathbf{f}_i^{l-1}, \tag{A3}$$

$$\mathbf{v}_{ij}^{h,l} = \mathbf{W}_{v}^{h,l}\mathbf{f}_j^{l-1}, \tag{A4}$$

where $\mathbf{x}_i^{l-1}$ and $\mathbf{x}_j^{l-1}$ denote the output coordinates of the ith central node and its jth neighboring node of the $l-1$th block, respectively. In the rest of Supplementary Materials, $\mathbf{W}$ are referred as learnable parameters of Linear layers, and their bias are omitted for simplicity, where $\mathbf{W}_{e,1}^{h,l} \in \mathbb{R}^{d_h \times d_e}$, $\mathbf{W}_{e,2}^{l} \in \mathbb{R}^{2d_h \times (d_h+1)}$, $\mathbf{W}_{e,3}^{h,l} \in \mathbb{R}^{d_h \times 2d_h}$, $\mathbf{W}_{q}^{h,l}, \mathbf{W}_{k}^{h,l}, \mathbf{W}_{v}^{h,l} \in \mathbb{R}^{d_h \times d_f}$. $\text{Concat}(\cdot)$ denotes the concatenation operation. μ is a learnable weight. d_h denote the dimension of the hth ($h \in \{1, ..., N_h\}$) attention head. In this work, $d_h = d_f/N_h$. $\odot$ stands for the element-wise multiplication.

The output of the message aggregation is then calculated by the following steps,

$$\mathbf{a}_{ij}^{h,l} = \frac{1}{\sqrt{d_h}}(\mathbf{q}_{ij}^{h,l} \odot \mathbf{k}_{ij}^{h,l}), \tag{A5}$$

$$\omega_{ij}^{h,l} = \text{softmax}_j(\|\mathbf{a}_{ij}^{h,l}\|_1), \tag{A6}$$

$$\hat{\mathbf{f}}_i^{l} = \mathbf{W}_{f,o}^{l}\underset{h\in 1,...,N_h}{\text{Concat}}\Big(\sum_{j\in Neighbor(i)} \omega_{ij}^{h,l}\mathbf{v}_{ij}^{h,l}\Big), \tag{A7}$$

$$\hat{\mathbf{e}}_{ij}^{l} = \mathbf{W}_{e,o}^{l}\underset{h\in 1,...,N_h}{\text{Concat}}(\mathbf{a}_{ij}^{h,l}), \tag{A8}$$

where $\mathbf{W}_{f,o}^{h,l} \in \mathbb{R}^{d_f \times d_f}$, $\mathbf{W}_{e,o}^{h,l} \in \mathbb{R}^{d_e \times d_f}$. $Neighbor(i)$ denote the set of neighboring nodes of the ith central node.

Before feeding the node and edge features to the feedforward step in lth block, we apply a novel proposed element-wise gate $\Psi(\cdot)$ on them, as inspired by [80]. The gated output of the node feature $\hat{\hat{\mathbf{f}}}_i^{l}$ can be written as,

$$\mathbf{g}^{l} = \text{sigmoid}(\mathbf{W}_{g}^{l}\text{Concat}(\hat{\mathbf{f}}_i^{l}, \mathbf{f}_i^{l-1}, \hat{\mathbf{f}}_i^{l} - \mathbf{f}_i^{l-1})), \tag{A9}$$

$$\hat{\hat{\mathbf{f}}}_i^{l} = \Psi_{f,1}^{l}(\hat{\mathbf{f}}_i^{l}, \mathbf{f}_i^{l-1}) = \text{Norm}(\mathbf{g}^{l} \odot \hat{\mathbf{f}}_i^{l} + \mathbf{f}_i^{l-1}), \tag{A10}$$

where $\mathbf{W}_g \in \mathbb{R}^{d_f \times 3d_f}$, Norm$(\cdot)$ denote the Layer Normalization [81], and $\mathbf{f}_i^l$ is the final updated features in the lth block. $\hat{\hat{\mathbf{f}}}_i^l$ is then fed to the feedforward step, as calculated by,

$$\mathbf{f}_i^l = \Psi_{f,2}^l(\mathbf{W}_{f,2}^l \text{LeakyReLU}(\mathbf{W}_{f,1}^l \hat{\hat{\mathbf{f}}}_i^l), \hat{\hat{\mathbf{f}}}_i^l) \tag{A11}$$

where $\mathbf{W}_{f,1}^l \in \mathbb{R}^{2d_f \times d_f}$, $\mathbf{W}_{f,2}^l \in \mathbb{R}^{d_f \times 2d_f}$. The edge feature $\mathbf{e}_{ij}^l$ is calculated similarly.

A.2.3 Coordinate update block

The coordinate update block updates the coordinates of the nodes (atoms) in 3-D space along with the feature update block, as inspired by [75]. Specifically, for the lth block, the coordinates are updated as follows,

$$\mathbf{\Delta}_h^l = \sum_{j \in Neighbor(i)} \frac{\mathbf{x}_i^{l-1} - \mathbf{x}_j^{l-1}}{\|\mathbf{x}_i^{l-1} - \mathbf{x}_j^{l-1}\|_2} \mathbf{W}_x^l \mathbf{a}_{ij}^{h,l}, \tag{A12}$$

$$\mathbf{x}_i^l = \mathbf{x}_i^{l-1} + \sum_{h \in 1,\ldots,N_h} \lambda_h^l \mathbf{\Delta}_h^l, \tag{A13}$$

where $\mathbf{a}_{ij}^{h,l}$ is calculated in the feature update block (refer to Eq. A5), $\mathbf{x}$ denote the coordinate, $\mathbf{W}_x^l \in \mathbb{R}^{1 \times d_h}$. Note that only the coordinates of the ligand nodes are updated.

A.2.4 Affinity and coordinate prediction

LigPose estimates the affinity of the protein-ligand complex as the auxiliary task, to be complementary to the coordinate prediction. In this case, the node features $\mathbf{f}_i^{N_l}$ are fed to a two-layer multi-layer perceptron, as calculated by,

$$\mathbf{y}_{aff} = \mathbf{W}_{a,2} \text{LeakyReLU}(\mathbf{W}_{a,1} \frac{1}{N_f} \sum_{i \in 1,\ldots,N_f} \mathbf{f}_i^{N_l}), \tag{A14}$$

where $\mathbf{y}_{aff}$ denote the prediction of the affinity, $\mathbf{W}_{a,1} \in \mathbb{R}^{d_f \times d_f}$, $\mathbf{W}_{a,2} \in \mathbb{R}^{1 \times d_f}$, and N_f denotes the number of nodes.

A.3 Training

The loss for affinity prediction is defined as,

$$\mathcal{L}_{aff} = (\mathbf{y}_{aff}^{pred} - \mathbf{y}_{aff}^{true})^2. \tag{A15}$$

Then, the final loss of the network is defined as,

$$\mathcal{L} = \gamma_1 (\frac{1}{N_l} \sum_{l \in 1,\ldots,N_l} \mathcal{L}_{sym}^l + \mathcal{L}_{sym}^{N_l}), + \gamma_2 \mathcal{L}_{aff} \tag{A16}$$

where $\mathcal{L}_{sym}$ is the proposed symmetric-aware loss as defined in the main text (refer to Main text Eq. 4) and the equivalent indexes used for $\mathcal{L}_{sym}$ are obtained from RDKit [68]. γ_1 and γ_2 are weight parameters to balance the losses.

We train our model with the Monte Carlo method, as also adopted by [17]. For all N_c cycles, we only train a data batch with one random cycle at a time while there is no gradient calculation and parameter update for the rest cycles. We train our model with a total of 150 epochs. The learning rate is initialize according to Table A3, and set to 1×10^{-6} for last 5 epochs.

Affinities are used with a negative log scale. Besides, all coordinates and affinities are rescaled to 1/10 of their original values for training stability.

A.4 Evaluation

In evaluation, we use an ensemble strategy for LigPose, *i.e.*, predicting $N_{ens} = 10$ times for a given complex.

A.5 Supplementary tables

Table A1 Notations.

Notations	Descriptions
N_l	Number of blocks
N_{ens}	Number of ensemble
N_c	Number of cycles
N_h	Number of attention heads
N_f	Number of nodes
N_{lig}	Number of ligand nodes
N_s	Number of ligand equivalent indexes
l	Index of blocks
h	Index of heads
d_f	Dimension of node feature
d_e	Dimension of edge feature
d_h	Dimension of attention head
γ_1	Weight for symmetric-aware loss
γ_2	Weight for affinity loss
$\mathbf{f}$	Node feature
$\mathbf{e}$	Edge feature
$\mathbf{x}$	Node coordinate
$\mathbf{q}$	Query for attention
$\mathbf{k}$	Key for attention
$\mathbf{v}$	Value for attention
$\mathbf{\Delta}$	Distance gradient
λ	Weight parameter for multi-head in coordinate update
$\mathcal{L}$	Total loss
$\mathcal{L}_{coor}$	Coordinate loss
$\mathcal{L}_{sym}$	Symmetric-aware loss
$\mathcal{L}_{aff}$	Affinity loss
$\mathbf{W}$	Weight parameter of Linear layer

Table A2 Feature list of nodes and edges. All features are processed with one-hot encoding except the distances. The initialization of distances are described in Main text Methods. The feature sizes are denoted in brackets.

Node features		Edge features
Protein node	Ligand node	
element type (4)	element type (10)	covalent connection (1)
atom degree (5)	atom degree (6)	distance (1)
implicit valence (5)	implicit valence (5)	covalent bond type (5)
neighboring H atoms (5)	neighboring H atoms (5)	
atom hybridization (3)	atom hybridization (6)	
amino acid type (20)	formal charge (6)	
atom name (37)	ring of size (6)	
	aromatic type (1)	

Table A3 Hyperparameters for LigPose.

Hyperparameters	Settings
Number of feature update blocks	8
Number of coordinate update blocks	8
Dimension of node feature (d_f)	128
Dimension of edge feature (d_e)	64
Number of attention heads (N_h)	4
Number of ensemble (N_{ens})	10
Number of cycles (N_c)	3
Weight for symmetric-aware loss (γ_1)	1
Weight for affinity loss (γ_2)	1
Batch size	8
Learning rate	1×10^{-4}

A.6 Supplementary figures

Fig. A1 A symmetric molecule (phenol). Obviously, two equivalent indexes, *i.e.*, C1-C2-C3-C4-C5-C6-O7 and C1-C6-C5-C4-C3-C2-O7, can be created for this molecule.

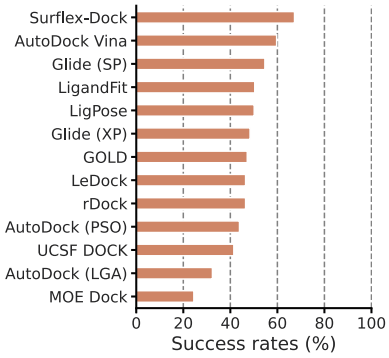


Fig. A2 Quantitative comparison of success rate between LigPose and the best pose generated by docking tools on peptide/peptide-like ligands. Surflex Dock achieved the highest success rate of 67.3%. LigPose (success rate of 50.2%) ranked the 5th among all methods. As described in main text Sec. 2.2, these results can be regarded as the upper bound of docking tools, where LigPose predicts greatly higher success rate than 8 docking tools.

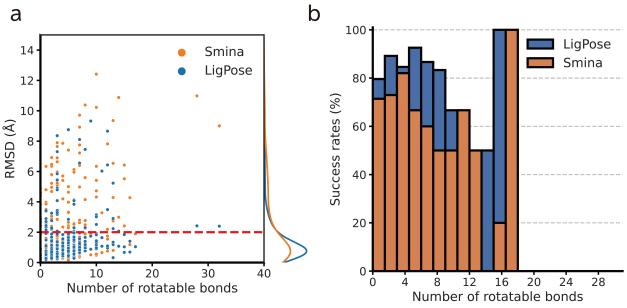


Fig. A3 a, RMSD values and b, success rate for prediction of LigPose and Smina with respect to the number of rotatable bonds in the core set of PDBbind. The red dashed line denotes the RMSD threshold of 2Å. LigPose showed better performance on most cases.

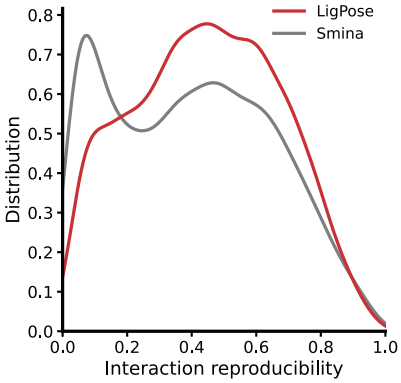


Fig. A4 Distribution of the interaction reproducibility of all non-covalent interactions of LigPose (red line) and Smina (grey line). The higher accuracy of LigPose is positively related to a better performance on reproducing the native-like non-covalent interactions, which is important for drug screening and lead optimization.

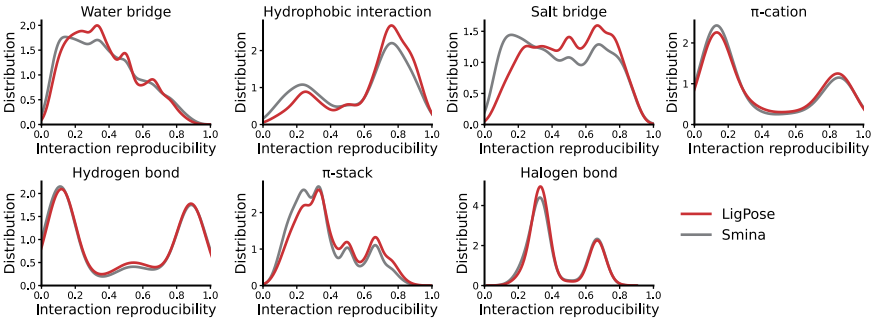


Fig. A5 Distribution of the interaction reproducibility of seven non-covalent interactions of LigPose (red line) and Smina (grey line), including water bridge, hydrophobic interaction, salt bridge, π-cation, hydrogen bond, π-stack and halogen bond. These results indicated that LigPose consistently outperformed Smina on all seven types of non-covalent interactions.

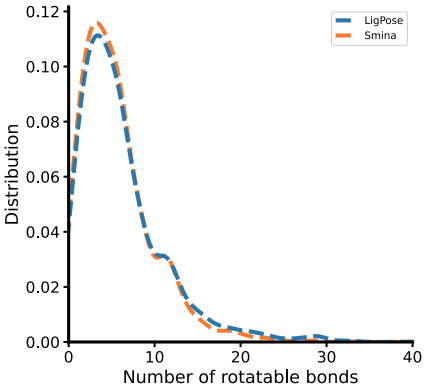


Fig. A6 Distribution of the number of rotatable bonds of successfully prepared ligands by LigPose and Smina. A small part (7.1%) of data in the refined set cannot be successfully processed by Smina (as mentioned in main text Methods Sec. 4.6). Since the number of rotatable bonds largely influences the prediction difficulty, we showed the distribution of ligand samples which were successfully processed by Smina or LigPosein experiments, respectively. This figure indicates that the number of rotatable bonds for samples used in Smina were slightly less than samples used in LigPose, where the comparison is still in a fair condition.

References

[1] Priyadeep Bhutani, Gaurav Joshi, Nivethitha Raja, Namrata Bachhav, Prabhakar K Rajanna, Hemant Bhutani, Atish T Paul, and Raj Kumar. US FDA approved drugs from 2015–june 2020: a perspective. *Journal of Medicinal Chemistry*, 64(5):2339–2381, 2021.

[2] Jiankun Lyu, Sheng Wang, Trent E Balius, Isha Singh, Anat Levit, Yurii S Moroz, Matthew J O'Meara, Tao Che, Enkhjargal Algaa, Kateryna Tolmachova, et al. Ultra-large library docking for discovering new chemotypes. *Nature*, 566(7743):224–229, 2019.

[3] Mingyue Zheng, Jihui Zhao, Chen Cui, Zunyun Fu, Xutong Li, Xiaohong Liu, Xiaoyu Ding, Xiaoqin Tan, Fei Li, Xiaomin Luo, et al. Computational chemical biology and drug design: Facilitating protein structure, function, and modulation studies. *Medicinal research reviews*, 38(3):914–950, 2018.

[4] Douglas B Kitchen, Hélène Decornez, John R Furr, and Jürgen Bajorath. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature reviews Drug discovery*, 3(11):935–949, 2004.

[5] Shengzheng Wang, Guoqiang Dong, and Chunquan Sheng. Structural simplification: an efficient strategy in lead optimization. *Acta Pharmaceutica Sinica B*, 9(5):880–901, 2019.

[6] Heping Zheng, Katarzyna B Handing, Matthew D Zimmerman, Ivan G Shabalin, Steven C Almo, and Wladek Minor. X-ray crystallography over the past decade for novel drug discovery–where are we heading next? *Expert opinion on drug discovery*, 10(9):975–989, 2015.

[7] Jean-Paul Renaud, Ashwin Chari, Claudio Ciferri, Wen-ti Liu, Hervé-William Rémigy, Holger Stark, and Christian Wiesmann. Cryo-EM in drug discovery: achievements, limitations and prospects. *Nature reviews Drug discovery*, 17(7):471–492, 2018.

[8] Luca Pinzi and Giulio Rastelli. Molecular docking: shifting paradigms in drug discovery. *International journal of molecular sciences*, 20(18):4331, 2019.

[9] Xuan-Yu Meng, Hong-Xing Zhang, Mihaly Mezei, and Meng Cui. Molecular docking: a powerful approach for structure-based drug discovery. *Current computer-aided drug design*, 7(2):146–157, 2011.

[10] Zhe Wang, Huiyong Sun, Xiaojun Yao, Dan Li, Lei Xu, Youyong Li, Sheng Tian, and Tingjun Hou. Comprehensive evaluation of ten docking programs on a diverse set of protein–ligand complexes: the prediction accuracy of sampling power and scoring power. *Physical Chemistry Chemical Physics*, 18(18):12964–12975, 2016.

[11] Dejun Jiang, Chang-Yu Hsieh, Zhenxing Wu, Yu Kang, Jike Wang, Ercheng Wang, Ben Liao, Chao Shen, Lei Xu, Jian Wu, et al. InteractionGraphNet: A novel and efficient deep graph representation learning framework for accurate protein–ligand interaction predictions. *Journal of medicinal chemistry*, 64(24):18209–18232, 2021.

[12] Oscar Méndez-Lucio, Mazen Ahmad, Ehecatl Antonio del Rio-Chanona,

and Jörg Kurt Wegner. A geometric deep learning approach to predict binding conformations of bioactive molecules. *Nature Machine Intelligence*, pages 1–7, 2021.

[13] Jaechang Lim, Seongok Ryu, Kyubyong Park, Yo Joong Choe, Jiyeon Ham, and Woo Youn Kim. Predicting drug–target interaction using a novel graph neural network with 3D structure-embedded graph representation. *Journal of chemical information and modeling*, 59(9):3981–3988, 2019.

[14] Jingxiao Bao, Xiao He, and John ZH Zhang. DeepBSP—a machine learning method for accurate prediction of protein–ligand docking structures. *Journal of Chemical Information and Modeling*, 2021.

[15] Joseph A Morrone, Jeffrey K Weber, Tien Huynh, Heng Luo, and Wendy D Cornell. Combining docking pose rank and structure with deep learning improves protein–ligand binding mode prediction over a baseline docking approach. *Journal of chemical information and modeling*, 60(9):4170–4179, 2020.

[16] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. Protein–ligand scoring with convolutional neural networks. *Journal of chemical information and modeling*, 57(4):942–957, 2017.

[17] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[18] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[19] Minkyung Baek and David Baker. Deep learning and protein structure modeling. *Nature methods*, 19(1):13–14, 2022.

[20] Ian R Humphreys, Jimin Pei, Minkyung Baek, Aditya Krishnakumar, Ivan Anishchenko, Sergey Ovchinnikov, Jing Zhang, Travis J Ness, Sudeep Banjade, Saket R Bagde, et al. Computed structures of core eukaryotic protein complexes. *Science*, 374(6573):eabm4805, 2021.

[21] Tomer Tsaban, Julia K Varga, Orly Avraham, Ziv Ben-Aharon, Alisa Khramushin, and Ora Schueler-Furman. Harnessing protein folding neural networks for peptide–protein docking. *Nature communications*, 13(1):1–12, 2022.

[22] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew W Senior, Timothy Green, Augustin Žídek, Russell Bates, Sam Blackwell, Jason Yim, et al. Protein complex prediction with AlphaFold-Multimer. *BioRxiv*, 2021.

[23] Patrick Bryant, Gabriele Pozzati, and Arne Elofsson. Improved prediction of protein-protein interactions using alphafold2. *Nature Communications*,

13(1):1–11, 2022.

[24] David Ryan Koes, Matthew P Baumgartner, and Carlos J Camacho. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. *Journal of chemical information and modeling*, 53(8):1893–1904, 2013.

[25] Sheng-You Huang. Comprehensive assessment of flexible-ligand docking algorithms: current effectiveness and challenges. *Briefings in bioinformatics*, 19(5):982–994, 2018.

[26] Zhihai Liu, Yan Li, Li Han, Jie Li, Jie Liu, Zhixiong Zhao, Wei Nie, Yuchen Liu, and Renxiao Wang. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics*, 31(3):405–412, 2015.

[27] Yan Li, Zhihai Liu, Jie Li, Li Han, Jie Liu, Zhixiong Zhao, and Renxiao Wang. Comparative assessment of scoring functions on an updated benchmark: 1. compilation of the test set. *Journal of chemical information and modeling*, 54(6):1700–1716, 2014.

[28] Gareth Jones, Peter Willett, Robert C Glen, Andrew R Leach, and Robin Taylor. Development and validation of a genetic algorithm for flexible docking. *Journal of molecular biology*, 267(3):727–748, 1997.

[29] Garrett M Morris, Ruth Huey, William Lindstrom, Michel F Sanner, Richard K Belew, David S Goodsell, and Arthur J Olson. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*, 30(16):2785–2791, 2009.

[30] Richard A Friesner, Jay L Banks, Robert B Murphy, Thomas A Halgren, Jasna J Klicic, Daniel T Mainz, Matthew P Repasky, Eric H Knoll, Mee Shelley, Jason K Perry, et al. Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of medicinal chemistry*, 47(7):1739–1749, 2004.

[31] Hongtao Zhao and Amedeo Caflisch. Discovery of ZAP70 inhibitors by high-throughput docking into a conformation of its kinase domain generated by molecular dynamics. *Bioorganic & medicinal chemistry letters*, 23(20):5721–5726, 2013.

[32] Lorenz M Mayr and Dejan Bojanic. Novel trends in high-throughput screening. *Current opinion in pharmacology*, 9(5):580–588, 2009.

[33] Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative assessment of scoring functions: the CASF-2016 update. *Journal of chemical information and modeling*, 59(2):895–913, 2018.

[34] Seokhyun Moon, Wonho Zhung, Soojung Yang, Jaechang Lim, and Woo Youn Kim. PIGNet: A physics-informed deep learning model toward generalized drug-target interaction predictions. *Chemical Science*, 2022.

[35] Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 2018.

[36] Zixuan Cang and Guo-Wei Wei. TopologyNet: Topology based deep

convolutional and multi-task neural networks for biomolecular property predictions. *PLoS computational biology*, 13(7):e1005690, 2017.

[37] Derek Jones, Hyojin Kim, Xiaohua Zhang, Adam Zemla, Garrett Stevenson, WF Drew Bennett, Daniel Kirshner, Sergio E Wong, Felice C Lightstone, and Jonathan E Allen. Improved protein–ligand binding affinity prediction with structure-based deep fusion inference. *Journal of chemical information and modeling*, 61(4):1583–1592, 2021.

[38] Liangzhen Zheng, Jingrong Fan, and Yuguang Mu. Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction. *ACS omega*, 4(14):15956–15965, 2019.

[39] Dejun Jiang, Chang-Yu Hsieh, Zhenxing Wu, Yu Kang, Jike Wang, Ercheng Wang, Ben Liao, Chao Shen, Lei Xu, Jian Wu, et al. InteractionGraphNet: A novel and efficient deep graph representation learning framework for accurate protein–ligand interaction predictions. *Journal of medicinal chemistry*, 64(24):18209–18232, 2021.

[40] Yeji Wang, Shuo Wu, Yanwen Duan, and Yong Huang. A point cloud-based deep learning strategy for protein–ligand binding affinity prediction. *Briefings in Bioinformatics*, 23(1):bbab474, 2022.

[41] Huimin Shen, Youzhi Zhang, Chunhou Zheng, Bing Wang, and Peng Chen. A cascade graph convolutional network for predicting protein–ligand binding affinity. *International journal of molecular sciences*, 22(8):4023, 2021.

[42] Joseph M Paggi, Julia A Belk, Scott A Hollingsworth, Nicolas Villanueva, Alexander S Powers, Mary J Clark, Augustine G Chemparathy, Jonathan E Tynan, Thomas K Lau, Roger K Sunahara, et al. Leveraging nonstructural data to predict structures and affinities of protein–ligand complexes. *Proceedings of the National Academy of Sciences*, 118(51), 2021.

[43] Lijuan Yang, Guanghui Yang, Xiaolong Chen, Qiong Yang, Xiaojun Yao, Zhitong Bing, Yuzhen Niu, Liang Huang, and Lei Yang. Deep scoring neural network replacing the scoring function components to improve the performance of structure-based molecular docking. *ACS Chemical Neuroscience*, 12(12):2133–2142, 2021.

[44] Regine S Bohacek, Colin McMartin, and Wayne C Guida. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal research reviews*, 16(1):3–50, 1996.

[45] Arman A Sadybekov, Anastasiia V Sadybekov, Yongfeng Liu, Christos Iliopoulos-Tsoutsouvas, Xi-Ping Huang, Julie Pickett, Blake Houser, Nilkanth Patel, Ngan K Tran, Fei Tong, et al. Synthon-based ligand discovery in virtual libraries of over 11 billion compounds. *Nature*, pages 1–8, 2021.

[46] Mengran Fan, Jian Wang, Huaipan Jiang, Yilin Feng, Mehrdad Mahdavi, Kamesh Madduri, Mahmut T Kandemir, and Nikolay V Dokholyan. Gpu-accelerated flexible molecular docking. *The Journal of Physical Chemistry*

B, 125(4):1049–1060, 2021.

[47] Xinqiang Ding, Yujin Wu, Yanming Wang, Jonah Z Vilseck, and Charles L Brooks III. Accelerated CDOCKER with GPUs, parallel simulated annealing, and fast fourier transforms. *Journal of chemical theory and computation*, 16(6):3910–3919, 2020.

[48] Leonardo Solis-Vasquez, Andreas F Tillack, Diogo Santos-Martins, Andreas Koch, Scott LeGrand, and Stefano Forli. Benchmarking the performance of irregular computations in Autodock-GPU molecular docking. *Parallel Computing*, 109:102861, 2022.

[49] Francesco Gentile, Vibudh Agrawal, Michael Hsing, Anh-Tien Ton, Fuqiang Ban, Ulf Norinder, Martin E Gleave, and Artem Cherkasov. Deep docking: a deep learning platform for augmentation of structure based drug discovery. *ACS central science*, 6(6):939–949, 2020.

[50] Stanisław Jastrzębski, Maciej Szymczak, Agnieszka Pocha, Stefan Mordalski, Jacek Tabor, Andrzej J Bojarski, and Sabina Podlewska. Emulating docking results using a deep neural network: a new perspective for virtual screening. *Journal of Chemical Information and Modeling*, 60(9):4246–4262, 2020.

[51] Sean Ekins, Ana C Puhl, Kimberley M Zorn, Thomas R Lane, Daniel P Russo, Jennifer J Klein, Anthony J Hickey, and Alex M Clark. Exploiting machine learning for end-to-end drug discovery and development. *Nature materials*, 18(5):435–441, 2019.

[52] Nasim Abdollahi, Ali Madani, Bo Wang, and Stephen MacKinnon. Residue characterization on AlphaFold2 protein structures using graph neural networks. In *NeurIPS*, 2021.

[53] Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

[54] Limeng Pu, Rajiv Gandhi Govindaraj, Jeffrey Mitchell Lemoine, Hsiao-Chun Wu, and Michal Brylinski. Deepdrug3d: classification of ligand-binding pockets in proteins with a convolutional neural network. *PLoS computational biology*, 15(2):e1006718, 2019.

[55] Rishal Aggarwal, Akash Gupta, Vineeth Chelur, CV Jawahar, and U Deva Priyakumar. Deeppocket: ligand binding site detection and segmentation using 3d convolutional neural networks. *Journal of Chemical Information and Modeling*, 2021.

[56] Bin Huang, Yang Xu, Xiuhong Hu, Yongrui Liu, Shanhui Liao, Jiahai Zhang, Chengdong Huang, Jingjun Hong, Quan Chen, and Haiyan Liu. A backbone-centred energy function of neural networks for protein design. *Nature*, pages 1–6, 2022.

[57] Ivan Anishchenko, Samuel J Pellock, Tamuka M Chidyausiku, Theresa A Ramelot, Sergey Ovchinnikov, Jingzhou Hao, Khushboo Bafna, Christoffer Norn, Alex Kang, Asim K Bera, et al. De novo protein design by deep network hallucination. *Nature*, 600(7889):547–552, 2021.

[58] Shuang Wang, Tao Song, Shugang Zhang, Mingjian Jiang, Zhiqiang Wei, and Zhen Li. Molecular substructure tree generative model for de novo drug design. *Briefings in Bioinformatics*, 2022.

[59] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018.

[60] Mingyang Wang, Huiyong Sun, Jike Wang, Jinping Pang, Xin Chai, Lei Xu, Honglin Li, Dongsheng Cao, and Tingjun Hou. Comprehensive assessment of deep generative architectures for de novo drug design. *Briefings in Bioinformatics*, 23(1):bbab544, 2022.

[61] Pengyong Li, Jun Wang, Yixuan Qiao, Hao Chen, Yihuan Yu, Xiaojun Yao, Peng Gao, Guotong Xie, and Sen Song. An effective self-supervised framework for learning expressive molecular global representations to drug discovery. *Briefings in Bioinformatics*, 22(6):bbab109, 2021.

[62] W Patrick Walters and Regina Barzilay. Applications of deep learning in molecule generation and molecular property prediction. *Accounts of chemical research*, 54(2):263–270, 2020.

[63] Qifeng Bai, Shuoyan Tan, Tingyang Xu, Huanxiang Liu, Junzhou Huang, and Xiaojun Yao. MolAICal: a soft tool for 3D drug design of protein targets by artificial intelligence and classical algorithm. *Briefings in bioinformatics*, 22(3):bbaa161, 2021.

[64] Xiaoqi Wang, Bin Xin, Weihong Tan, Zhijian Xu, Kenli Li, Fei Li, Wu Zhong, and Shaoliang Peng. DeepR2cov: deep representation learning on heterogeneous drug networks to discover anti-inflammatory agents for COVID-19. *Briefings in bioinformatics*, 22(6):bbab226, 2021.

[65] Sowmya Ramaswamy Krishnan, Navneet Bung, Gopalakrishnan Bulusu, and Arijit Roy. Accelerating de novo drug design against novel proteins using deep learning. *Journal of Chemical Information and Modeling*, 61(2):621–630, 2021.

[66] Michael Moret, Moritz Helmstädter, Francesca Grisoni, Gisbert Schneider, and Daniel Merk. Beam search for automated design and scoring of novel ROR ligands with machine intelligence. *Angewandte Chemie International Edition*, 60(35):19477–19482, 2021.

[67] Helen Berman, Kim Henrick, and Haruki Nakamura. Announcing the worldwide protein data bank. *Nature Structural & Molecular Biology*, 10(12):980–980, 2003.

[68] Greg Landrum et al. Rdkit: Open-source cheminformatics. 2006.

[69] Brian J Bender, Stefan Gahbauer, Andreas Luttens, Jiankun Lyu, Chase M Webb, Reed M Stein, Elissa A Fink, Trent E Balius, Jens Carlsson, John J Irwin, et al. A practical guide to large-scale docking. *Nature protocols*, 16(10):4799–4832, 2021.

[70] Thomas A Halgren. Identifying and characterizing binding sites and assessing druggability. *Journal of chemical information and modeling*, 49(2):377–389, 2009.

[71] Alasdair TR Laurie and Richard M Jackson. Q-SiteFinder: an energy-based method for the prediction of protein–ligand binding sites. *Bioinformatics*, 21(9):1908–1916, 2005.

[72] Dominik Gront, Sebastian Kmiecik, and Andrzej Kolinski. Backbone building from quadrilaterals: a fast and accurate algorithm for protein backbone reconstruction from alpha carbon coordinates. *Journal of computational chemistry*, 28(9):1593–1597, 2007.

[73] Sebastian Kmiecik, Dominik Gront, Michal Kolinski, Lukasz Wieteska, Aleksandra Elzbieta Dawid, and Andrzej Kolinski. Coarse-grained protein models and their applications. *Chemical reviews*, 116(14):7898–7936, 2016.

[74] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *stat*, 1050:20, 2017.

[75] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021.

[76] Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open Babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1):1–14, 2011.

[77] Thomas Gaillard. Evaluation of Autodock and Autodock Vina on the CASF-2013 benchmark. *Journal of chemical information and modeling*, 58(8):1697–1706, 2018.

[78] Sebastian Salentin, Sven Schreiber, V Joachim Haupt, Melissa F Adasme, and Michael Schroeder. PLIP: fully automated protein–ligand interaction profiler. *Nucleic acids research*, 43(W1):W443–W447, 2015.

[79] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

[80] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

[81] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.