TANKBind: Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Illuminating interactions between proteins and small drug molecules is a long-standing challenge in the field of drug discovery. Despite the importance of understanding these interactions, most previous works are limited by hand-designed scoring functions and insufficient conformation sampling. The recently-proposed graph neural network-based methods provides alternatives to predict protein-ligand complex conformation in a one-shot manner. However, these methods neglect the geometric constraints of the complex structure and weaken the role of local functional regions. As a result, they might produce unreasonable conformations for challenging targets and generalize poorly to novel proteins. In this paper, we propose Trigonometry-Aware Neural networKs for binding structure prediction, TANKBind, that builds trigonometry constraint as a vigorous inductive bias into the model and explicitly attends to all possible binding sites for each protein by segmenting the whole protein into functional blocks. We construct novel contrastive losses with local region negative sampling to jointly optimize the binding interaction and affinity. Extensive experiments show substantial performance gains in comparison to state-of-the-art physics-based and deep learning-based methods on commonly-used benchmark datasets for both binding structure and affinity predictions with variant settings.

1 Introduction

Proteins are the workhorses of human bodies. They have a wide range of interaction partners, small molecules, other proteins, and DNA/RNA, for example. In this paper, we focus on drug-like small molecules as the interaction partners for proteins. The words *ligands*, *drugs*, *small molecules* and *compounds* are used interchangeably throughout the paper. Small molecules activate or inhibit activities of target proteins through mostly non-covalent interactions. In 2021, FDA approved 60 new drugs, among which 36 were small molecules [1]. Understanding the mechanism-of-actions and off-target effects of drug molecules typically requires analyzing the structures of the related protein-ligand complexes [2, 3], but solving the complex structure experimentally is a an extremely challenging task. Despite tremendous effort spent on this topic over the last 50 years, only about 19,000 protein-ligand complex structures have been solved experimentally using X-ray, Cryo-EM or NMR [4]. On the other hand, the estimated chemical space of drug is 10^{60} and estimated number of unique proteins in human body is at least $20,000$, making the number of possible protein-ligand complex far exceeding the number of experimentally solved structures [5, 6].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

On the computational side, molecular docking is a commonly-used method for predicting the protein-ligand complex structures the corresponding binding affinities [7–10]. Generally, the docking process involves three main stages: (1) locating favorable binding sites given a protein target; (2) sampling the ligand conformation as well as its position and orientation within these sites; (3) scoring and ranking the conformations of the complex using physics-inspired empirical energy functions to refine the structures and assess protein-ligand binding affinity. Due to its good interpretability and usability, docking has been integrated in drug development process for a long time and a number of successful cases have been reported [11]. However, most open-source docking packages use atom-level pairwise scoring functions, limiting the capacity to model the many-body effects. Moreover, they need to sample a large range of possible ligand poses and protein side-chain conformations, which leads to relatively high computational cost [7, 12].

To overcome these challenges, we propose a two-stage deep learning framework to neuralize the molecular docking process and predict the binding structures with better accuracy and lower computational cost. In the first stage, we segment the whole protein into functional blocks and predict their interactions with the ligand, creating an protein-ligand interaction energy landscape using a novel trigonometry-aware architecture. The trigonometry module has enough model capacity to capture many-body effects. In the second stage, we prioritize the crystallized binding structures by constrastively ensuring a weaker binding affinity for non-native interactions. In particular, our model improves the drug-protein binding structure predictions with a combination of (i) a novel trigonometry-aware architecture that jointly infuses trigonometry constraints and excluded-volume effects as inductive biases, (ii) a new divide-and-conquer strategy that constructs the protein-ligand local functional binding pairs in a contrastive manner. By doing so, we create a funnel-shape energy landscape for the inter-molecular interaction, removing the need of extensive sampling [12–14].

Our novel method is well-motivated by leveraging prior knowledge from physics and biology. Physically, the inter-molecular trigonometry module, inspired by the intra-molecular Evoformer module used in AlphaFold2 [13], ensures that our energy landscape disfavors configurations of protein-ligand complexes that are prohibited by laws of nature, for instance, no two atoms could overlap and the distances between atoms have to satisfy triangle inequality theorem in euclidean geometry. More details on these constraints is shown in section 3.3. Biologically, the functional regions of proteins tend to be more conserved and closely associated with binding [15, 16], allowing the model to learn critical information and generalize better to unseen proteins.

We evaluate our algorithm against several state-of-the-art deep learning and physics-based docking methods on task of binding structure prediction under multiple settings. Compared with baselines, our model increase the fraction of predictions with ligand root-mean-square deviation (RMSD) less than 5Å by 16% in re-docking setting, 22% in self-docking setting, and 42% in the more difficult new-protein setting. Our model is also capable of predicting binding affinities, achieving better correlations with experimentally-measured values than sequence-based, structure-based and even complex-based methods. We also show that TankBind has the potential to discover novel mechanism-of-actions of drug molecules by identifying unseen protein binding sites.

2 Related Work

Geometric Deep Learning for drug discovery. There has been a surge of interest in integrating geometric priors for representation learning in the domain of drug discovery [13, 17–23]. Recent researches have incorporated geometric information and symmetry properties of the input signals to improve the spatial perception of the learned representations. These works have been shown great potential in various applications like protein structure modeling [13, 17–19], molecular low-energy generation prediction [24–26], property/function prediction [22, 23] and molecule design [20, 21]. Among which, AlphaFold 2 achieved outstanding performance in protein structure prediction [13], representing the state-of-the-art geometry-aware method. Our work is inspired from this ground-breaking work, adapting it from the intra-molecular structure prediction to the field of predicting the inter-molecular binding structure and binding affinity.

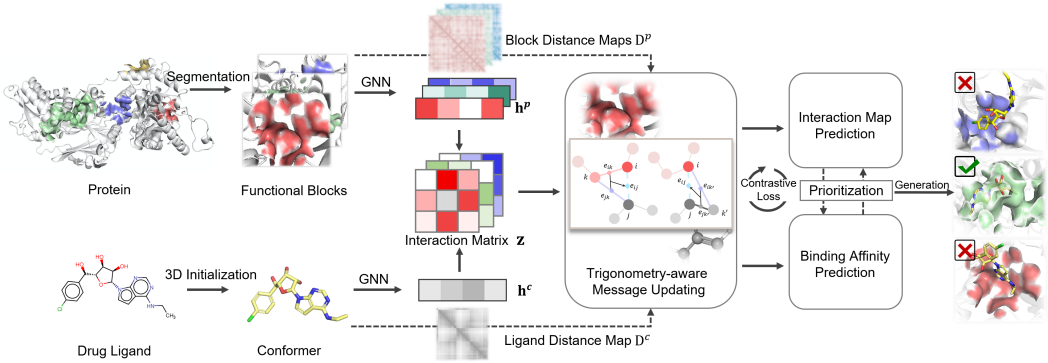

Figure 1: Overview of TankBind Model. The whole protein is divided into blocks of radius 20Å, each block is going through the TankBind model along with the drug compound. Both protein blocks and drug compound are modeled as graphs. The block-compound interaction matrix evolved multiple times with additional input based on the distance maps of the protein block and the compound through trigonometry module. Based on the updated interaction embedding, the model predicts the binding affinity of the compound to the blocks and the block-ligand distance maps. A constrastive loss function is used to ensure the native block binds stronger to the compound than decoys.

Drug-protein Interaction (DPI) prediction. The goal of DPI prediction is to illustrate the binding structure and binding affinity between protein and ligand. Apart from docking-based approaches [7, 8], prior machine learning-based works either use complex-free models to predict the binding affinity directly from protein-ligand pairs [27–32] or make predictions through complex structure that has been previously obtained by experimental or docking approaches [33–35]. The former ones are less interpretable while the latter requires data involved in vast experimental costs and labour. More recently, EquiBind [36] takes a new approach by directly predicting the key points on both the protein and the compound, and aligning their key points through the ingeniously designed optimal transport loss. However, this method may generate compound structures clashing with the protein structures and currently lacks the capability to predict the binding affinity, limiting its use in drug discovery. In contrast, our approach has a trigonometry module imposing geometry constraints and a state-of-the-art binding affinity prediction capability.

3 TankBind Model

3.1 Overview of TankBind model

The general protocol of our model is shown in figure 1. The encoding of protein and compound is described in section 3.2. The rationale and implementation of trigonometry module is detailed in section 3.3. The design of loss functions for training is described in section 3.4. The generation of atom coordinates from predicted inter-molecular distance map is introduced in section 3.5.

3.2 Structural encoders of protein and drug

Our model input is the separate structures of a protein and a drug compound, both encoded as graphs. Indices i, k always operate on the residue dimension, j, k' always on the compound dimension. n is the number of protein nodes and m is the number of compound nodes.

Protein. The protein is represented as a proximity 3D graph following Jing et al. [37]. We denote the protein graph as $\mathcal{G}^p = (\mathcal{V}^p, \mathcal{E}^p)$, where each node $\mathfrak{v}_i^p \in \mathcal{V}^p$ corresponds to an amino acid, and has feature $\mathbf{h}_{\mathfrak{v}^p}^{(i)}$ with both scalar and vector features. Each node also has a position $\mathbf{x}_i^p \in \mathbb{R}^3$ equal to the Cartesian coordinate of C_{α_i}. An edge $\mathfrak{e}_{ik}^p$ exists if $\mathfrak{v}_k^p$ is among the 30 nearest neighbors of $\mathfrak{v}_i^p$. Each edge $\mathfrak{e}_{ik}^p \in \mathcal{E}^p$ also encodes both the scalar and the vector features. We then apply the geometric

vector perceptrons (GVP) [18, 37] to embed the protein and arrive at feature $\mathbf{h}^p \in \mathbb{R}^{n\times s}$ after graph propagation, where n is the number of nodes and s is the embedding size.

To implicitly model side-chain flexibility, we choose a residue-level representation ignoring the finer details of protein structure, separating our method from other methods that use all-atoms or surface vertexes representation [38, 39]. Also, as shown by Jumper et al. [13, 40], residue-level embedding is enough to infer the side-chain conformation.

Motivated by protein co-evolution [15] and divide-and-conquer theory, the protein graph, $\mathcal{G}^p$, is further divided into subgraphs $\mathcal{G}^{p'}$. Each subgraph $\mathcal{G}^{p'}$ includes all the $\mathfrak{v}_i^p$ and $\mathfrak{e}_{ij}^p$ inside the functional block. The subgraph is denoted as $\mathcal{G}^{p'} = (\{\mathfrak{v}_i^p, \mathfrak{e}_{ik}^p\} \mid \|\mathbf{x}_i^p - \mathbf{x}_o\| \leq 20\text{Å}, \|\mathbf{x}_k^p - \mathbf{x}_o\| \leq 20\text{Å})$, where $\mathbf{x}_o$ is the center of the functional block predicted by a widely-used ligand-agnostic method, P2rank (published in 2018)[41]. Justification for the size of radius and use of P2rank is described in appendix G.

Drug compound. The drug compound is represented as a graph using TorchDrug toolkit [42]. The compound graph is denoted as $\mathcal{G}^c = (\mathcal{V}^c, \mathcal{E}^c)$ where each node $\mathfrak{v}_j^c \in \mathcal{V}^c$ corresponds to a heavy atom (non-hydrogen atom), and has feature $\mathbf{h}_{\mathfrak{v}^c}^{(j)}$ and each edge $\mathfrak{e}_{jk'}^c$ has feature $\mathbf{h}_{\mathfrak{e}^c}^{(jk')}$. We use Graph Isomorphism Network (GIN) [43] to embed the compound and arrive at feature $\mathbf{h}^c \in \mathbb{R}^{m\times s}$ after graph propagation, where m is the number of heavy atoms and s is the embedding size.

3.3 Details of trigonometry module

The compound feature, $\mathbf{h}^c$, and the protein block feature, $\mathbf{h}^p$, are used to form the initial interaction embedding $\mathbf{z}^{(0)} \in \mathbb{R}^{n\times m\times s}$, $\mathbf{z}_{ij}^{(0)} = \mathbf{h}_i^p \odot \mathbf{h}_j^c$. The interaction embedding will be further updated with pair distance map of protein nodes, $D_{ik}^p = \|\mathbf{x}_i^p - \mathbf{x}_k^p\|$ and pair distance map of compound nodes, $D_{jk'}^c = \left\|\mathbf{x}_j^c - \mathbf{x}_{k'}^c\right\|$.

The rationale for including both the pair distance map of the protein nodes and the pair distance map of the compound nodes in updating the protein-compound interaction embedding is explained with two simplified examples. As shown in the upper part of figure 2, if a protein node A is in close proximity with compound node B, then compound node C will not be in contact with node A due to the large distance constraint between node B and C. Distance constraint between compound nodes B and D could also force a node D to be in close contact with protein node A.

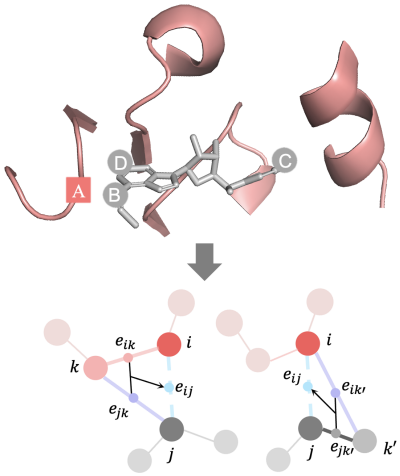

Figure 2: Rationale for including trigonometry module. Upper: Protein node in square, compound nodes in circles. Lower: Trigonometry module ensures that the interaction between protein node i and compound node j depends on all protein and compound nodes k, k'.

To build this observation of trigonometry constraints into our model, we design the following module to update the interaction embedding, in layer ℓ, $\forall(i,j)$:

$$\tilde{\mathbf{z}}_{ij}^{(\ell)} = \mathbf{z}_{ij}^{(\ell)} + \Phi\Big(\sum_{k=1}^{n} \mathbf{p}_{ik}\mathbf{t}_{kj}^{(\ell)} + \sum_{k'=1}^{m} \mathbf{t}'^{(\ell)}_{ik'}\mathbf{c}_{k'j}\Big) \odot \mathbf{g}(\mathbf{z}_{ij}^{(\ell)}) \tag{1}$$

where $\mathbf{p}_{ik} = \phi(D_{ik}^p)$ is the linear embedding of encoded pair distance between protein nodes. $\mathbf{p} \in \mathbb{R}^{n\times n\times s}$, n is the number of nodes in protein block, s is the embedding size. $\mathbf{c}_{jk'} = \phi(D_{jk'}^c)$ is the linear embedding of encoded pair distance between compound nodes. $\mathbf{c} \in \mathbb{R}^{m\times m\times s}$, m is the number of compound nodes. $\mathbf{t}_{ij}^{(\ell)}$ and $\mathbf{t}'^{(\ell)}_{ij}$ are the same gated linear transformations of $\mathbf{z}_{ij}^{(\ell)}$ but with non-shared parameters, $\mathbf{t}_{ij}^{(\ell)} = \text{Linear}(\mathbf{z}_{ij}^{(\ell)}) \odot \mathbf{g}(\mathbf{z}_{ij}^{(\ell)})$, $\mathbf{t}^{(\ell)} \in \mathbb{R}^{n\times m\times s}$, $\mathbf{g}(\mathbf{z}_{ij}^{(\ell)}) = \text{sigmoid}(\text{Linear}(\mathbf{z}_{ij}^{(\ell)}))$, Φ is a layernorm function followed by a linear transformation.

Another type of physical constraint need to be take into consideration is the excluded-volume (Van Der Waals) and saturation effect. As shown in the upper figure 2, if protein node A forms a strong interaction, hydrogen bonding for example, with compound node B, then node D is unlikely to form the same type of interaction with node A because node A has limited number of hydrogen donors or acceptors. To account for these effects, we designed a self-attention module to modulate the interaction between a protein node and all compound nodes by taking the whole interaction between this protein node and all compound nodes into consideration.

$$\dot{\mathbf{z}}_{ij}^{(\ell)} = \tilde{\mathbf{z}}_{ij}^{(\ell)} + \Phi(\text{concat}_h(\sum_{k'=1}^{m}(w_{ijk'}^{(\ell)h}\mathbf{v}_{ik'}^{(\ell)h}) \odot \mathbf{g}^h(\tilde{\mathbf{z}}_{ij}^{(\ell)}))) \tag{2}$$

$$w_{ijk'}^{(\ell)h} = \text{softmax}_{k'}({\mathbf{q}_{ij}^{(\ell)h}}^\top \mathbf{k}_{ik'}^{(\ell)h}) \tag{3}$$

, where $\mathbf{q}_{ij}^{(\ell)h}, \mathbf{k}_{ij}^{(\ell)h}, \mathbf{v}_{ij}^{(\ell)h}$ are linear transformation of $\tilde{\mathbf{z}}_{ij}^{(\ell)}$, h is number of attention heads. Function $\mathbf{g}^h$ is the standard $\mathbf{g}$ with reshaping the embedding into heads at the end, Φ is a linear transformation.

Lastly, a non-linear transition module is added to transit the interaction embedding to the next layer through multilayer perceptron, $\mathbf{z}_{ij}^{(\ell+1)} = \text{MLP}(\dot{\mathbf{z}}_{ij}^{(\ell)})$. The whole trigonometry module is composed of three consecutive parts, the trigonometry update, the self-attention modulation, and the non-linear transition module. Layernorm is applied on every input $\mathbf{z}_{ij}^{(\ell)}$ and a 25% dropout is applied to the trigonometry update and self-attention modulation during training. The final outputs, drug-protein binding affinity, $\hat{a} = \sum_{i=1}^{n}\sum_{j=1}^{m}\text{Linear}(\mathbf{z}_{ij}^{(L)})$, and inter-molecular distance map, $D_{ij}^{pred} = \mathbf{g}(\mathbf{z}_{ij}^{(L)})\text{Linear}(\mathbf{z}_{ij}^{(L)})$, are predicted directly based on the last layer embedding $\mathbf{z}_{ij}^{(L)}$, where L is the number of module stacks.

3.4 Design of binding interaction and affinity loss functions

Many previous works model the interaction between compound and protein by only preserving the interaction region, residues that far away are ignored [26, 44]. On the positive side, the computation and memory demand for characterize the interaction between protein and the drug compound is greatly reduced by focusing on regional interaction. But the fact of not binding to alternative binding sites is also a valuable information. By the nature of crystallization, if a protein-compound complex could be successfully crystallized, other possible binding sites on this protein definitely bind less strongly than the native binding site to the compound, therefore, those other binding sites could be used as high-valued decoys. Based on this observation, we designed a max-margin constrastive affinity loss, equation 4, following the idea of [45]. Such that the compound's predicted affinity, $\hat{a}$, to the decoys is less than the experimentally measured affinity, a, by a margin value, ϵ.

$$\mathcal{L}_{\text{affinity}}(\hat{a}_\zeta, a) = \mathbb{1}(\zeta)(\hat{a}_\zeta - a)^2 + (1 - \mathbb{1}(\zeta))\max(0, \hat{a}_\zeta - (a - \epsilon))^2 \tag{4}$$

where $\hat{a}_\zeta$ is the predicted affinity to block ζ, and indicator function $\mathbb{1}(\zeta) = 1$ when block ζ encloses the native ligand, and $\mathbb{1}(\zeta) = 0$ otherwise. We, therefore, take full use of information stored in the whole protein instead of only the native binding region. We also include a mean squared erorr (MSE) loss for native interaction distance map, $\mathcal{L}_{\text{distance}} = \mathbb{1}(\zeta)\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}(D_{ij}^{pred} - D_{ij})^2$. The overall training objective of TankBind is: $\mathcal{L} = \mathcal{L}_{\text{affinity}} + \mathcal{L}_{\text{distance}}$.

3.5 Generation of drug coordinates based on predicted inter-molecular distance map.

The Cartesian coordinates, $\{\hat{\mathbf{x}}_j^c\}$, of the heavy atoms of a drug compound could be deduced analytically based on the predicted inter-molecular distance matrix, D_{ij}^{pred}, the coordinates of protein nodes, $\{\mathbf{x}_i^p\}$, and the pair distance matrix of compound nodes, $D_{jk'}^c$ [46, 47]. But since predicted distance matrix contains noise, we take a numerical approach [46, 48]. By minimizing the total loss, $\mathcal{L}_{\text{generation}}$, which consists of two parts, the interaction loss and the compound configuration loss, we could derive the coordinates of the docked drug coordinates, $\{\hat{\mathbf{x}}_j^c\}$.

$$\mathcal{L}_{\text{generation}} = \mathcal{L}_{\text{interaction}} + \mathcal{L}_{\text{configuration}} = \sum_{i}^{n}\sum_{j}^{m}(|\hat{D}_{ij} - D_{ij}^{pred}|) + \sum_{j}^{m}\sum_{k'}^{m}(|\hat{D}_{jk'}^c - D_{jk'}^c|) \tag{5}$$

Table 1: Blind self-docking. All models take a pair of ligand structure (generated by RDKit) and protein structure as input, trying to predict the atom coordinates of the ligand after binding. In blind docking, the protein binding site is assumed unknown. Test set is composed of 363 protein-ligand structure crystallized after 2019 curated by PDBbind database. Details about model runtime and the number of model parameters are in appendix C

	Ligand RMSD						**Centroid Distance**					
	Percentiles ↓				% Below Threshold ↑		Percentiles ↓				% Below Threshold ↑	
Methods	25%	50%	75%	Mean	2Å	5Å	25%	50%	75%	Mean	2Å	5Å
QVINA-W	2.5	7.7	23.7	13.6	20.9	40.2	0.9	3.7	22.9	11.9	41.0	54.6
GNINA	2.8	8.7	22.1	13.3	21.2	37.1	1.0	4.5	21.2	11.5	36.0	52.0
SMINA	3.8	8.1	17.9	12.1	13.5	33.9	1.3	3.7	16.2	9.8	38.0	55.9
GLIDE(c.)	2.6	9.3	28.1	16.2	**21.8**	33.6	**0.8**	5.6	26.9	14.4	36.1	48.7
VINA	5.7	10.7	21.4	14.7	5.5	21.2	1.9	6.2	20.1	12.1	26.5	47.1
EQUIBIND-U	3.3	5.7	9.7	7.8	7.2	42.4	1.3	2.6	7.4	5.6	40.0	67.5
EQUIBIND	3.8	6.2	10.3	8.2	5.5	39.1	1.3	2.6	7.4	5.6	40.0	67.5
TANKBind-R	2.8	5.2	11.2	9.4	16.0	47.9	1.0	2.3	7.7	7.3	44.9	69.4
TANKBind-C	2.4	4.5	8.4	8.2	19.6	54.8	0.9	1.9	5.4	6.3	53.2	73.3
TANKBind-P	2.6	4.5	8.1	8.5	16.3	54.0	0.9	1.9	5.2	6.4	53.2	74.4
TANKBind	**2.4**	**4.0**	**7.7**	**7.4**	19.3	**61.7**	0.9	**1.7**	**4.2**	**5.5**	**56.5**	**77.4**

$$\hat{D}_{ij} = \left\|\mathbf{x}_i^p - \hat{\mathbf{x}}_j^c\right\|, \hat{D}_{jk'}^c = \left\|\hat{\mathbf{x}}_j^c - \hat{\mathbf{x}}_{k'}^c\right\| \tag{6}$$

where n is the number of protein nodes, and m is number of compound nodes, and $\{\mathbf{x}_j^p\}$ are the Cartesian coordinates of protein nodes. All inter-molecular distances are clamped to have an upper bound of 10Å to focus on the direct interaction. In self-docking setting, when the compound configuration is unknown, we add a local atomic structures (LAS) mask to the configuration loss to allow for compound flexibility while enforcing basic geometric constraint, $\mathcal{L}_{\text{configuration}} = \sum_j^m \sum_{k'}^m \mathbb{1}(j,k')(|\hat{D}_{jk'}^c - D_{jk'}^c|)$ where $\mathbb{1}(j,k') = 1$ when compound atom j and k' are connected by connected by a bond, or 2-hop away, or in the same ring structure, and $\mathbb{1}(j,k') = 0$ otherwise [7, 36]. For every test protein-ligand pair, TankBind predicts the binding affinity of the ligand to all segmented functional blocks and chooses the one with strongest affinity to generate the binding structures.

4 Evaluation

4.1 Protein-ligand binding structure prediction

Dataset. We used publicly available PDBbind v2020 dataset [4] which has the structures of 19443 protein-ligand complexes along with their experimentally measured binding affinity. PDBbind is a database curated based on the Protein Data Bank (PDB) [49]. We followed the same time split as defined in EquiBind paper [36] in which the training and validation data are the protein-ligand complex structures deposited before 2019 and the test set is the structures deposited after 2019. After removing a few structures that unable to process using RDKit from the training set, we had 17787 structures for training, 968 for validation and 363 for testing[50]. We also reduced the possibility of encountering equally valid binding sites by removing chains that have no atom within 10Å from any atom of the ligand following the protocol described in [36].

Baselines. We compared TankBind with the most widely-used docking method AutoDock Vina[7] and the recent proposed geometry-based DL method EquiBind [36]. We also included four popular docking methods QVina-W, GINA[51], SMINA[52] and GLIDE[8] as listed in Stärk et al. [36].

Evaluation metrics. We follow prior work Stärk et al. [36] and use ligand root-mean-square deviation (RMSD) of atomic positions and centroid distance to compare predicted binding structures with ground-truths. The Ligand RMSD calculates the normalized Frobenius norm of the two corresponding matrices of ligand coordinates. The centroid distance is defined as the the distance

Table 2: Blind self-docking for unseen receptors. All models evaluated on 142 crystallized protein-compound structures where the proteins have not been observed in training set.

	Ligand RMSD						**Centroid Distance**					
	Percentiles ↓				% Below Threshold ↑		Percentiles ↓				% Below Threshold ↑	
Methods	25%	50%	75%	Mean	2Å	5Å	25%	50%	75%	Mean	2Å	5Å
QVINA-W	3.4	10.3	28.1	16.9	15.3	31.9	1.3	6.5	26.8	15.2	35.4	47.9
GNINA	4.5	13.4	27.8	16.7	13.9	27.8	2.0	10.1	27.0	15.1	25.7	39.5
SMINA	4.8	10.9	26.0	15.7	9.0	25.7	1.6	6.5	25.7	13.6	29.9	41.7
GLIDE	3.4	18.0	31.4	19.6	**19.6**	28.7	**1.1**	17.6	29.1	18.1	29.4	40.6
VINA	7.9	16.6	27.1	18.7	1.4	12.0	2.4	15.7	26.2	16.1	20.4	37.3
EQUIBIND-U	5.7	8.8	14.1	11.0	1.4	21.5	2.6	6.3	12.9	8.9	16.7	43.8
EQUIBIND	5.9	9.1	14.3	11.3	0.7	18.8	2.6	6.3	12.9	8.9	16.7	43.8
TANKBind-R	3.6	6.9	17.0	12.6	5.6	35.2	1.3	3.6	15.7	10.3	35.2	58.5
TANKBind-C	3.4	5.5	9.8	9.9	9.2	43.0	**1.1**	2.6	8.1	7.9	46.5	65.5
TANKBind-P	3.3	5.5	10.9	11.2	5.6	45.1	1.3	2.3	7.9	9.1	**47.9**	66.9
TANKBind	**2.9**	**4.7**	**8.8**	**9.1**	4.9	**55.6**	1.3	**2.3**	**4.8**	**7.0**	45.1	**75.4**

between the averaged 3D coordinates of the predicted and ground-truth bound ligand atoms, indicating the model capability of identifying correct binding region. Hydrogens are not involved in the calculation.

Performance in blind flexible self-docking We start with a real-world blind self-docking experiment, in which the ligand conformation is not fixed, and the result of re-docking experiment, in which the native ligand conformation is given, is reported in Appendix A. As shown in the table 1, TankBind achieves state-of-the-art performance, outperforming geometry DL-based model EquiBind. This advantage is particularly evident in the top 25% and top 50% ligand RMSD, which allows our method to predict 22% more qualified (below Threshold 5Å) binding poses than EquiBind. This results are also consistent in the metrics of centroid distance, demonstrating that our method also has a clear advantage in the identification of binding region. Even though GLIDE (commercial) and Autodock Vina are established docking software with more than a decade of continuous development, our model remarkably frequently outperforms them. At the same time, we are orders of magnitude faster than them, and on the same level as EquiBind (Appendix C). In addition, we explore the possible of TankBind-R, where we randomly segment the protein, TankBind-P, where we only doing the summation over protein nodes in equation 1, and TankBind-C, where we only sum over compound nodes. The performance reduction on the these variants supports our view that trigonometry message passing between proteins and ligand and segmentation choice are critical to the prediction of binding structures.

Performance in self-docking unseen protein We next focus on the new protein setting, in which the tested proteins have not been observed in the training set. Table 2 shows that Tankbind leads to larger improvements over EquiBind and docking methods with regard to ligand-RMSD and centriod distance.This is in line with our expectation that TankBind has better generalization ability due to the physical-inspired trigonometry module and explicit consideration of conservative functional blocks. In this setting, as shown in Figure 3 and table 2, for fractions smaller than 2Å, 5Å and 15Å, the performance between EquiBind and other docking method are comparable, while TankBind is always better by a large margin, further confirming the effectiveness of our method and indicating that the proposed strategy has practical values for the virtual screening of new proteins.

4.2 Protein-ligand binding affinity prediction

TankBind is also capable of predicting protein-ligand binding affinity because of the constrastive affinity loss function. Since we segmented the whole protein into protein blocks, the predicted binding affinity of ligand to the whole protein is equal to the binding affinity to the one protein block that

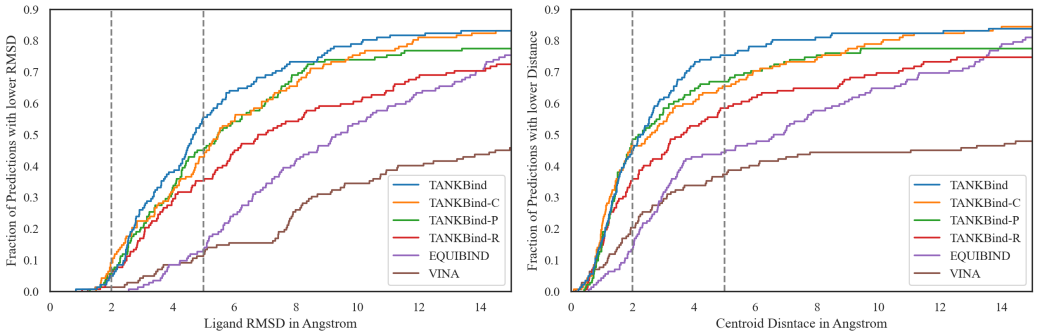

Figure 3: Estimator of the Cumulative Distribution Function (ECDF) plot for ligand RMSD (left) and Centroid Distance (right) from result evaluated on new receptors subset. The x axis of the figure stops at 15Å because comparison for larger RMSD is less meaningful when the predicted location of the ligand is away from the true binding site, a RMSD of 15Å is not better than RMSD of 50Å.

Methods	RMSE↓	Pearson↑	Spearman↑	MAE↓
TransCPI	1.741	0.576	0.540	1.404
MONN	1.438	0.624	0.589	1.143
PIGNet※	2.640	0.511	0.489	2.110
IGN	1.433	0.698	0.641	1.169
HOLOPROT	1.546	0.602	0.571	1.208
STAMPDPI	1.658	0.545	0.411	1.325
TANKBind	**1.346**	**0.726**	**0.703**	**1.070**

Table 3: Binding affinity prediction. TankBind achieves SOTA on all four metrics.

Methods	Ligand↓	Centroid↓	Below2A↑	Below5A↑
w/o P2Rank	9.37	7.30	44.90	69.42
w/o Trig	8.73	6.44	44.08	74.93
TAPE	8.81	6.89	50.69	73.00
GAT	8.27	6.23	56.47	**78.51**
TankBind-P	8.47	6.44	53.17	74.38
TankBind-C	8.20	6.27	53.17	73.28
Origin	**7.43**	**5.51**	**56.47**	77.41

Table 4: Ablation results. We listed four main metrics here, a complete table is in appendix E

predicted to bind strongest with the ligand. To demonstrate the ability, we compared TankBind with the state-of-the-art binding affinity prediction models.

Dataset. We split the dataset into training, test and validation splits based on the same time split described earlier. The experimentally measured affinity data in PDBbind dataset has three different names, depending on the exact experiment setups, 50% inhibiting concentration (IC50), inhibition constant (K_i), and dissociation constant (K_D), all converted to the unit of molar concentration. Similar to previous methods [23, 44], we predict negative log-transformed binding affinity.

Baselines and evaluation metrics. We compare TankBind against two state-of-the-art sequence-based methods, TransformerCPI [53] and MONN [28], two complex-based methods, IGN [38] and PIGNet [54] both requiring prior knowledge of the inter-molecular structure to predict affinity, and two structure-based methods, HOLOPTOT [23] and STAMPDPI [55]. For evaluating various methods, we use four metrics – root mean squared error (RMSE), Pearson correlation coefficient, Spearman correlation coefficient and mean absolute error (MAE). We also include the mean and standard deviation across 3 experimental runs in appendix D.

Result As shown in Table 3, our model obtains the best performance in PDBbind test set, consistently outperforms SOTA binding affinity prediction methods. Note that even without the prior interaction information, TankBind also achieves better result than complex-based methods (PIGNET and IGN), proving that the predicted binding structural information provided considerable gain to the affinity prediction task.

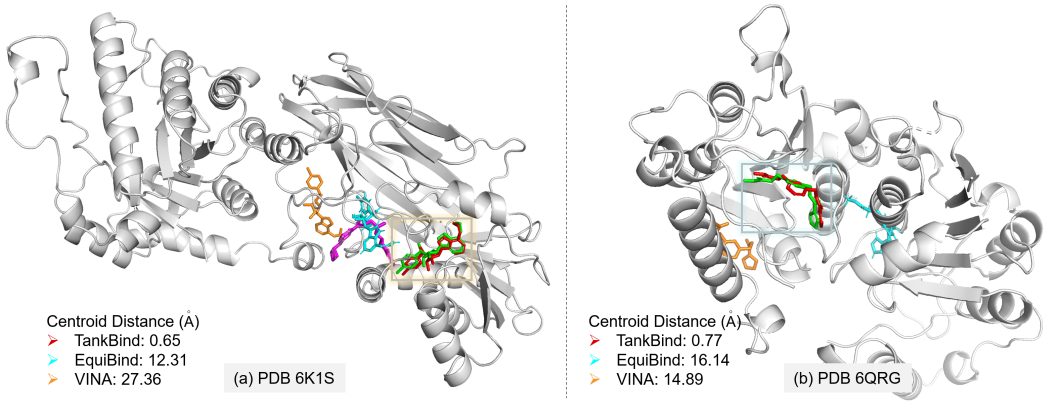

Figure 4: (a) An example of TankBind identifying an unseen binding site. The protein is shown in white, co-crystallized compounds of three PDBs in the training set is shown in purple. The ligand of 6K1S is shown in green. TankBind is able to find this correct pose for the compound, shown in red, while the other two, Vina in orange, and Equibind in cyan, place the compound away from the true binding site. (b) For PDB 6QRG, both protein and compound have not been seen in the training set. But TankBind still find the correct pose. Crystallized ligand colored in green, TankBind prediction in red, EquiBind in cyan and Vina result in organ.

4.3 Ablation study

We conducted ablation studies to investigate factors that influence the performance of proposed TankBind framework. As shown in Table 4, the original version of TankBind with the trigonometry message passing between protein and ligand shows the best performance among all architectures. Replacing the P2rank with a randomly split of blocks performed the worst, which verifies our hypothesis that functional block segmentation can improve generalization. Simple architecture substitutions for protein (TAPE) [56] and molecular representation (GAT) [57] decrease slightly the model performance. Replacing the intra-trigonometry module with the uni-modal variants (TankBind-P and TankBind-C) both caused noticeable decreases in performances.

4.4 Case studies

TankBind correctly identifies an unseen binding site for a new drug compound. As a representative case, in PDB 6K1S, a seen protein binds to a new drug compound at a site that has not been observed before. This protein has three co-crystallized complex structures in the training set, PDB 4X60, 4X61, 4X63. As shown in the left of figure 4, our method, shown in red, aligns well with the true ligand, shown in green, despite our method has never seen any compound locates at this site before. While other two methods, EquiBind in cyan, Vina in orange identify an incorrect site for this compound. Packages Kalign, Biopython, and Smith-Waterman library are used to systematically analyze the results [28, 58–60] (see Appendix H).

TankBind finds the correct pose when both compound and protein are unseen. We picked two representative examples with both compound and protein are unseen, one, PDB 6QRG, in the right of figure 4 and another, PDB 6KQI, in appendix B. Both PDB 6QRG and PDB 6KQI have max protein similarity below 0.8 (6QRG 0.78, 6KQI 0.57), and max compound similarity below 0.4 (6QRG 0.36, 6KQI 0.27).

5 Conclusion

In this work, we propose a novel binding structure and affinity prediction model, TankBind, that builds trigonometry constraints into the model and explicitly attends to all possible binding sites by

segmenting the whole protein into functional blocks. We observe significant improvements on task of binding structure prediction over existing deep learning methods: a 22% increase in the fraction of prediction below 5Å in ligand RMSD, and a 42% increase when the proteins have not been observed in the training set. Moreover, we demonstrate that the model is able to predict affinity and outperform SOTA methods on PDBbind. This work opens a new direction for modelling the inter-molecular interaction between protein and drug molecule. Numerous directions for further exploration include incorporating a ligand conformer generation module, enhancing the dataset with AlphaFold-predicted structure and public available SAR data, integrating the segmentation of functional block in an end-to-end manner, and combining the model with protein backbone dynamics modeling to handle larger scale conformation changes induced by drug-protein interactions.

References

[1] Michael S Kinch, Zachary Kraft, and Tyler Schwartz. 2021 in review: Fda approvals of new medicines. *Drug discovery today*, 2022.

[2] Subramanian Boopathi, Adolfo B Poma, and Ponmalai Kolandaivel. Novel 2019 coronavirus structure, mechanism of action, antiviral drug promises and rule out against its treatment. *Journal of Biomolecular Structure and Dynamics*, 39(9):3409–3418, 2021.

[3] Lei Xie, Li Xie, and Philip E Bourne. Structure-based systems biology for analyzing off-target binding. *Current opinion in structural biology*, 21(2):189–199, 2011.

[4] 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.

[5] Jean-Louis Reymond, Ruud Van Deursen, Lorenz C Blum, and Lars Ruddigkeit. Chemical space as a source for new drugs. *MedChemComm*, 1(1):30–38, 2010.

[6] Elena A Ponomarenko, Ekaterina V Poverennaya, Ekaterina V Ilgisonis, Mikhail A Pyatnitskiy, Arthur T Kopylov, Victor G Zgoda, Andrey V Lisitsa, and Alexander I Archakov. The size of the human proteome: the width and depth. *International journal of analytical chemistry*, 2016, 2016.

[7] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[8] 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.

[9] Suzanne Ackloo, Rima Al-awar, Rommie E Amaro, Cheryl H Arrowsmith, Hatylas Azevedo, Robert A Batey, Yoshua Bengio, Ulrich AK Betz, Cristian G Bologa, John D Chodera, et al. Cache (critical assessment of computational hit-finding experiments): A public–private partnership benchmarking initiative to enable the development of computational methods for hit-finding. *Nature Reviews Chemistry*, pages 1–9, 2022.

[10] Francesco Gentile, Jean Charle Yaacoub, James Gleave, Michael Fernandez, Anh-Tien Ton, Fuqiang Ban, Abraham Stern, and Artem Cherkasov. Artificial intelligence–enabled virtual screening of ultra-large chemical libraries with deep docking. *Nature Protocols*, pages 1–26, 2022.

[11] Amy C Anderson. The process of structure-based drug design. *Chemistry & biology*, 10(9): 787–797, 2003.

[12] Ajay N Jain. Scoring functions for protein-ligand docking. *Current Protein and Peptide Science*, 7(5):407–420, 2006.

[13] 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.

[14] José Nelson Onuchic, Zaida Luthey-Schulten, and Peter G Wolynes. Theory of protein folding: the energy landscape perspective. *Annual review of physical chemistry*, 48(1):545–600, 1997.

[15] David De Juan, Florencio Pazos, and Alfonso Valencia. Emerging methods in protein co-evolution. *Nature Reviews Genetics*, 14(4):249–261, 2013.

[16] Fabian Glaser, Tal Pupko, Inbal Paz, Rachel E Bell, Dalit Bechor-Shental, Eric Martz, and Nir Ben-Tal. Consurf: identification of functional regions in proteins by surface-mapping of phylogenetic information. *Bioinformatics*, 19(1):163–164, 2003.

[17] 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.

[18] Bowen Jing, Stephan Eismann, Pratham N Soni, and Ron O Dror. Equivariant graph neural networks for 3d macromolecular structure. *arXiv preprint arXiv:2106.03843*, 2021.

[19] Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Independent se (3)-equivariant models for end-to-end rigid protein docking. *arXiv preprint arXiv:2111.07786*, 2021.

[20] Wengong Jin, Jeremy Wohlwend, Regina Barzilay, and Tommi Jaakkola. Iterative refinement graph neural network for antibody sequence-structure co-design. *arXiv preprint arXiv:2110.04624*, 2021.

[21] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in Neural Information Processing Systems*, 32, 2019.

[22] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

[23] Vignesh Ram Somnath, Charlotte Bunne, and Andreas Krause. Multi-scale representation learning on proteins. *Advances in Neural Information Processing Systems*, 34, 2021.

[24] Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *International Conference on Machine Learning*, pages 9558–9568. PMLR, 2021.

[25] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

[26] 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*, 3(12):1033–1039, 2021.

[27] Jian Wang and Nikolay V Dokholyan. Yuel: Compound-protein interaction prediction with high generalizability. *bioRxiv*, 2021.

[28] Shuya Li, Fangping Wan, Hantao Shu, Tao Jiang, Dan Zhao, and Jianyang Zeng. Monn: a multi-objective neural network for predicting compound-protein interactions and affinities. *Cell Systems*, 10(4):308–322, 2020.

[29] Masashi Tsubaki, Kentaro Tomii, and Jun Sese. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, 35(2): 309–318, 2019.

[30] Kyle Yingkai Gao, Achille Fokoue, Heng Luo, Arun Iyengar, Sanjoy Dey, and Ping Zhang. Interpretable drug target prediction using deep neural representation. In *IJCAI*, volume 2018, pages 3371–3377, 2018.

[31] Mostafa Karimi, Di Wu, Zhangyang Wang, and Yang Shen. Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, 35(18):3329–3338, 2019.

[32] Shuangjia Zheng, Yongjian Li, Sheng Chen, Jun Xu, and Yuedong Yang. Predicting drug–protein interaction using quasi-visual question answering system. *Nature Machine Intelligence*, 2(2):134–140, 2020.

[33] José Jiménez, Miha Skalic, Gerard Martinez-Rosell, and Gianni De Fabritiis. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling*, 58(2):287–296, 2018.

[34] 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.

[35] 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.

[36] Hannes Stärk, Octavian-Eugen Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. *arXiv preprint arXiv:2202.05146*, 2022.

[37] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

[38] 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.

[39] 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.

[40] John M Jumper, Nabil F Faruk, Karl F Freed, and Tobin R Sosnick. Accurate calculation of side chain packing and free energy with applications to protein molecular dynamics. *PLoS computational biology*, 14(12):e1006342, 2018.

[41] Radoslav Krivák and David Hoksza. P2rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of cheminformatics*, 10(1): 1–12, 2018.

[42] Zhaocheng Zhu, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, et al. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.

[43] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[44] Raphael JL Townshend, Martin Vögele, Patricia Suriana, Alexander Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon Anderson, Stephan Eismann, et al. Atom3d: Tasks on molecules in three dimensions. *arXiv preprint arXiv:2012.04035*, 2020.

[45] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[46] Matthew Masters, Amr H Mahmoud, Yao Wei, and Markus Lill. Deep learning model for flexible and efficient protein-ligand docking. In *ICLR2022 Machine Learning for Drug Discovery*, 2022.

[47] Moritz Hoffmann and Frank Noé. Generating valid euclidean distance matrices. *arXiv preprint arXiv:1910.03131*, 2019.

[48] Zsolt Zsoldos, Darryl Reid, Aniko Simon, Sayyed Bashir Sadjad, and A Peter Johnson. ehits: a new fast, exhaustive flexible ligand docking system. *Journal of Molecular Graphics and Modelling*, 26(1):198–212, 2007.

[49] Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V Crichlow, Cole H Christie, Kenneth Dalenberg, Luigi Di Costanzo, Jose M Duarte, et al. Rcsb protein data bank: powerful new tools for exploring 3d structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic acids research*, 49(D1):D437–D451, 2021.

[50] Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.

[51] Andrew T McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Ryan Koes. Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics*, 13(1):1–20, 2021.

[52] 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.

[53] Lifan Chen, Xiaoqin Tan, Dingyan Wang, Feisheng Zhong, Xiaohong Liu, Tianbiao Yang, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. Transformercpi: improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. *Bioinformatics*, 36(16):4406–4414, 2020.

[54] 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*, 13(13):3661–3673, 2022.

[55] Penglei Wang, Shuangjia Zheng, Yize Jiang, Chengtao Li, Junhong Liu, Chang Wen, Atanas Patronov, Dahong Qian, Hongming Chen, and Yuedong Yang. Structure-aware multimodal deep learning for drug–protein interaction prediction. *Journal of Chemical Information and Modeling*, 62(5):1308–1317, 2022.

[56] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

[57] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[58] Timo Lassmann. Kalign 3: multiple sequence alignment of large datasets, 2020.

[59] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

[60] Mengyao Zhao, Wan-Ping Lee, Erik P Garrison, and Gabor T Marth. Ssw library: an simd smith-waterman c/c++ library for use in genomic applications. *PloS one*, 8(12):e82138, 2013.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See appendix M
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] see section I and J
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] see section D
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] see section J
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] PDBbind [4]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] our code is in supplemental material
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Additional binding structure prediction results and pseudo code for trigonometry module

A.1 Blind re-docking.

In re-docking, the ligand conformations are given as input. As shown in table 5, In 64.5 % of the test set, TankBind has ligand RMSD less than 5Å, compared to 48.2 % for EquiBind and 26.7 % for Vina.

Table 5: Blind re-docking. In re-docking setting, the input compound conformations are the co-crystallized conformations. All three models evaluated on the test set of 363 protein-compound structures crystallized after 2019 curated by PDBbind database.

	Ligand RMSD						**Centroid Distance**					
	Percentiles ↓				% Below Threshold ↑		Percentiles ↓				% Below Threshold ↑	
Methods	25%	50%	75%	Mean	2Å	5Å	25%	50%	75%	Mean	2Å	5Å
QVINA-W	1.6	7.9	24.1	13.4	27.7	39.0	0.9	3.8	23.2	11.8	40.4	55.4
GININA	1.3	6.1	22.9	12.2	32.2	46.8	0.7	2.8	22.1	10.9	43.8	58.4
SMINA	1.4	6.2	15.2	10.3	30.1	46.7	0.8	2.6	12.7	8.5	45.3	63.5
GLIDE	**0.5**	8.3	29.5	15.7	**43.4**	45.7	**0.3**	4.9	28.5	14.8	45.4	50.4
VINA	4.5	9.7	19.9	13.4	13.2	26.7	1.7	5.5	18.7	11.2	29.8	47.9
EQUIBIND-R	2.0	5.1	9.8	7.4	25.1	49.0	1.4	2.6	7.3	5.8	40.8	66.9
TankBind	1.4	**3.3**	**6.8**	**6.9**	37.5	**64.5**	0.8	**1.8**	**4.2**	**5.4**	**55.9**	**78.5**

We also show the cumulative distribution plot for ligand RMSD and centroid distance in figure 5.

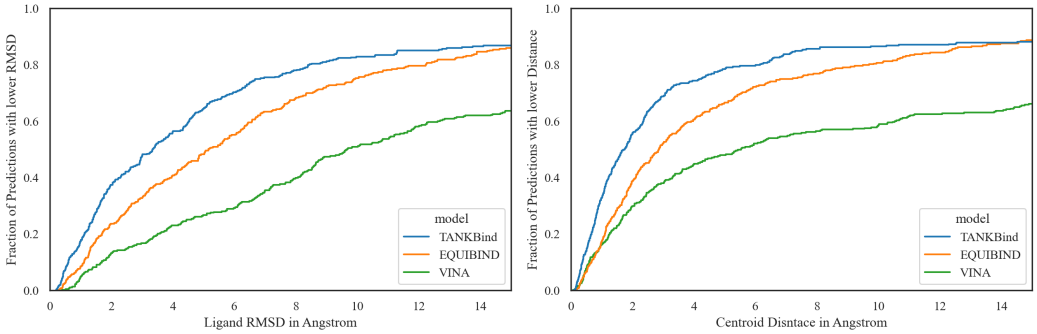

Figure 5: Blind re-docking. Estimator of the Cumulative Distribution Function (ECDF) plot for ligand RMSD (left) and Centroid Distance (right) based on the test set result. TankBind has a higher fraction of prediction with lower RMSD. The x axis of the figure stops at 15Å because comparison for larger RMSD is less meaningful, a RMSD of 15Å is not necessarily better than RMSD of 50Å.

A.2 Blind re-docking new receptors

We also benchmark on the new receptors test set in which the proteins have not been seen in the training set, shown in table 6 and figure 6.

Table 6: Blind re-docking new receptors. All three models evaluated on 142 PDBs which is a subset of the original 363 PDBs. The proteins in this subset have not been seen in the training set.

	Ligand RMSD						**Centroid Distance**					
	Percentiles ↓				% Below Threshold ↑		Percentiles ↓				% Below Threshold ↑	
Methods	25%	50%	75%	Mean	2Å	5Å	25%	50%	75%	Mean	2Å	5Å
VINA	6.6	12.3	25.9	16.1	8.5	19.7	2.4	7.3	25.2	14.0	23.2	39.4
EQUIBIND	4.9	9.6	15.8	11.3	8.5	25.4	2.9	6.6	14.3	9.3	16.2	41.6
TankBind	**2.6**	**4.6**	**9.2**	**8.8**	**20.4**	**53.5**	**1.4**	**2.5**	**6.0**	**6.9**	**37.3**	**73.2**

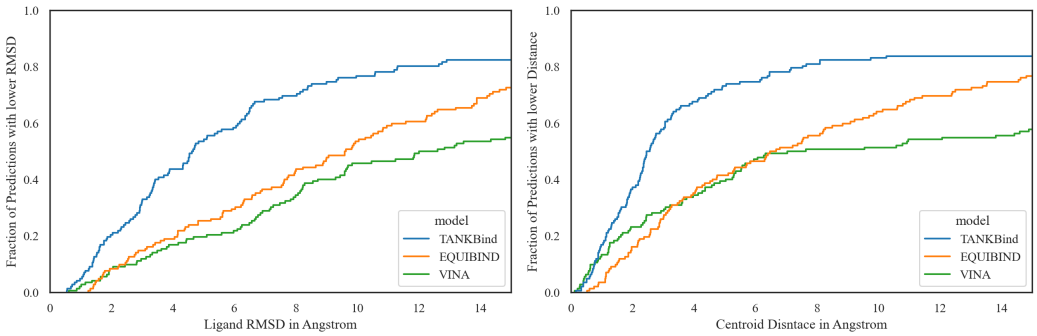

Figure 6: Blind re-docking new receptors. Estimator of the Cumulative Distribution Function (ECDF) plot for ligand RMSD (left) and Centroid Distance (right) from result evaluated on new receptors subset and with known ligand conformations. TankBind achieves a significantly better performance than the other two methods.

A.3 Pseudo code for the trigonometry module and an illustrative figure for trigonometry updating.

The pseudo code for the whole trigonometry module is listed below. The whole module contains three sub-module: *TrigonometryUpdating*, *TriangleSelfAttention*, and *Transition*.

Algorithm 1: The `TrigonometryModule` function for updating $\mathbf{z}_{ij}$.

Input: $\{\mathbf{z}_{ij}\}, \{\mathbf{p}_{ik}\}, \{\mathbf{c}_{k'j}\}$

// $\mathbf{z}_{ij} \in \mathrm{R}^{n\times m\times s}, \mathbf{p}_{ik} \in \mathrm{R}^{n\times n\times s}, \mathbf{c}_{k'j} \in \mathrm{R}^{m\times m\times s}$

Output: updated $\{\mathbf{z}_{ij}\}$

// $\mathbf{z}_{ij} \in \mathrm{R}^{n\times m\times s}$

/* Indices i, k always operate on the residue dimension, j, k' always on the compound dimension. n is the number of protein nodes and m is the number of compound nodes. */

for $\ell \leftarrow 0$ ***to*** L **do**

  $\{\mathbf{z}_{ij}\} \leftarrow \{\mathbf{z}_{ij}\} + \mathrm{Dropout}(\texttt{TrigonometryUpdating}(\{\mathbf{z}_{ij}\}, \{\mathbf{p}_{ik}\}, \{\mathbf{c}_{k'j}\}))$

  $\{\mathbf{z}_{ij}\} \leftarrow \{\mathbf{z}_{ij}\} + \mathrm{Dropout}(\texttt{TriangleSelfAttention}(\{\mathbf{z}_{ij}\}))$

  $\{\mathbf{z}_{ij}\} \leftarrow \texttt{Tranistion}(\{\mathbf{z}_{ij}\})$

end

Algorithm 2: The `TrigonometryUpdating` function.

Input: $\{\mathbf{z}_{ij}\}, \{\mathbf{p}_{ik}\}, \{\mathbf{c}_{k'j}\}$

// $\mathbf{z}_{ij} \in \mathrm{R}^{n\times m\times s}, \mathbf{p}_{ik} \in \mathrm{R}^{n\times n\times s}, \mathbf{c}_{k'j} \in \mathrm{R}^{m\times m\times s}$

Output: updated $\{\mathbf{z}_{ij}\}$

// $\mathbf{z}_{ij} \in \mathrm{R}^{n\times m\times s}$

/* Indices i, k always operate on the residue dimension, j, k' always on the compound dimension. n is the number of protein nodes and m is the number of compound nodes. */

$\mathbf{z}_{ij} \leftarrow \mathrm{LayerNorm}(\mathbf{z}_{ij})$

$\mathbf{p}_{ik} \leftarrow \mathrm{LayerNorm}(\mathbf{p}_{ik})$

$\mathbf{c}_{k'j} \leftarrow \mathrm{LayerNorm}(\mathbf{c}_{k'j})$

$\mathbf{p}_{ik} \leftarrow \mathrm{sigmoid}(\mathrm{Linear}(\mathbf{p}_{ik})) \odot \mathrm{Linear}(\mathbf{p}_{ik})$

$\mathbf{c}_{k'j} \leftarrow \mathrm{sigmoid}(\mathrm{Linear}(\mathbf{c}_{k'j})) \odot \mathrm{Linear}(\mathbf{c}_{k'j})$

$\mathbf{t}_{ij}, \mathbf{t}'_{ij} = \mathrm{sigmoid}(\mathrm{Linear}(\mathbf{z}_{ij})) \odot \mathrm{Linear}(\mathbf{z}_{ij})$

$\mathbf{o}_{ij} = \sum_{k=1}^{n} \mathbf{p}_{ik}\mathbf{t}_{kj} + \sum_{k'=1}^{m} \mathbf{t}'_{ik'}\mathbf{c}_{k'j}$

$\mathbf{g}_{ij} = \mathrm{sigmoid}(\mathrm{Linear}(\mathbf{z}_{ij}))$

$\mathbf{z}_{ij} \leftarrow \mathrm{Linear}(\mathrm{LayerNorm}(\mathbf{o}_{ij})) \odot \mathbf{g}_{ij}$

Algorithm 3: The `TriangleSelfAttention` function.

Input: $\{\mathbf{z}_{ij}\}$ // $\mathbf{z}_{ij} \in \mathbb{R}^{n \times m \times s}$

Output: updated $\{\mathbf{z}_{ij}\}$ // $\mathbf{z}_{ij} \in \mathbb{R}^{n \times m \times s}$

/* Indices i, k always operate on the residue dimension, j, k' always on the compound dimension. n is the number of protein nodes and m is the number of compound nodes. */

$\mathbf{z}_{ij} \leftarrow \text{LayerNorm}(\mathbf{z}_{ij})$

$\mathbf{q}_{ij}^h, \mathbf{k}_{ij}^h, \mathbf{v}_{ij}^h = \text{LinearNoBias}(\mathbf{z}_{ij})$ // $h \in \{1, ..., N_{head}\}$

$\mathbf{g}_{ij}^h = \text{sigmoid}(\text{Linear}(\mathbf{z}_{ij}))$

$w_{ijk'}^h = \text{softmax}_{k'}({\mathbf{q}_{ij}^h}^\top \mathbf{k}_{ik'}^h)$

$\mathbf{o}_{ij}^h = \mathbf{g}_{ij}^h \odot \sum_{k'=1}^m (w_{ijk'}^h \mathbf{v}_{ik'}^h)$

$\mathbf{z}_{ij} \leftarrow \text{Linear}(\text{concat}_h(\mathbf{o}_{ij}^h))$

Algorithm 4: The `Tranistion` function.

Input: $\{\mathbf{z}_{ij}\}$ // $\mathbf{z}_{ij} \in \mathbb{R}^{n \times m \times s}$

Output: updated $\{\mathbf{z}_{ij}\}$ // $\mathbf{z}_{ij} \in \mathbb{R}^{n \times m \times s}$

$\mathbf{z}_{ij} \leftarrow \text{LayerNorm}(\mathbf{z}_{ij})$

$\mathbf{z}_{ij} \leftarrow \text{Linear}(\text{ReLU}(\text{Linear}(\mathbf{z}_{ij})))$

Figure 7 shows the embeddings used to update a single interaction embedding, z_{ij} between a protein node i and a ligand node j.

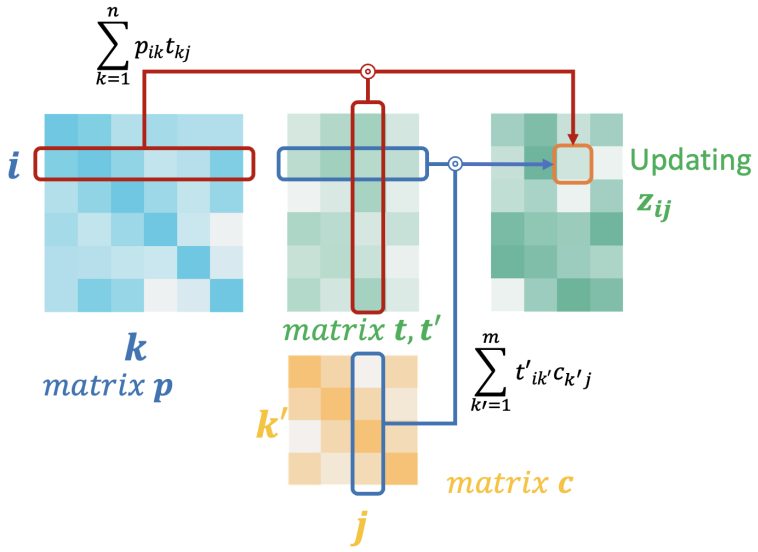

Figure 7: Illustrative figure for equation 1, $\tilde{\mathbf{z}}_{ij}^{(\ell)} = \mathbf{z}_{ij}^{(\ell)} + \Phi(\sum_{k=1}^n \mathbf{p}_{ik} \mathbf{t}_{kj}^{(\ell)} + \sum_{k'=1}^m \mathbf{t'}_{ik'}^{(\ell)} \mathbf{c}_{k'j}) \odot \mathbf{g}(\mathbf{z}_{ij}^{(\ell)})$

A.4 Visualization of the gate functions and the self-attention map

To show the motivation behind the employment of the gate functions and self-attention, we have visualization them using an example, PDB 6HD6. Figure8 shows the output of the gate function at the last stacked layer for the native protein block with the compound FYH. Its resemblance to the inter-molecular distance map indicates the gate function is able to modulate the interaction based on the predicted distance.

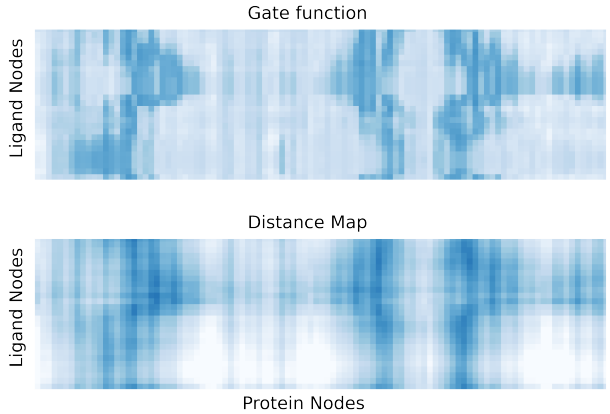

Figure 8: In upper figure, darker color means the gated value is closer to 1. lighter color means the gated value is closer to 0. In lower figure, darker color means the distance between the protein node and the ligand node is smaller, and lighter color means the distance is larger.

Figure9 shows the output of the self-attention map (3rd head) at the last stacked layer averaged over all protein nodes for the native protein block with compound FYH. The output resembles the compound intra-molecular distance map. This indicates that the self-attention module is aware of the compound conformation and is able to update the interaction embedding based on this.

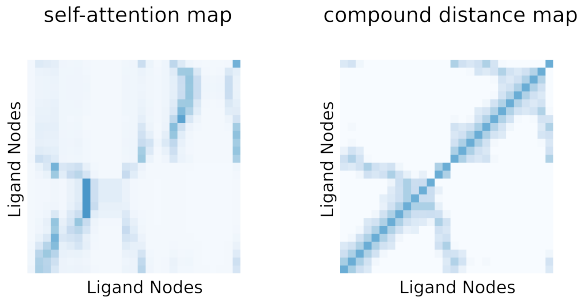

Figure 9: In upper figure, darker color means the self-attention value is larger. lighter color means the self-attention value is smaller. In lower figure, darker color means the distance between the ligand nodes is smaller, and lighter color means the distance is larger.

B Another example of TankBind finding the correct binding site when both the protein and the ligand are unseen.

A protein is unseen when the max protein sequence similarity (normalized Smith-Waterman alignment score) to the training set is less than 0.8. A compound is unseen when max compound similarity (Tanimoto Similarity of Morgan fingerprints) to the training set is less than 0.5. Figure 10 shows that, for PDB 6KQI, we correctly locate the true binding site on the protein, while the other two methods fail to do so under the same re-docking setting.

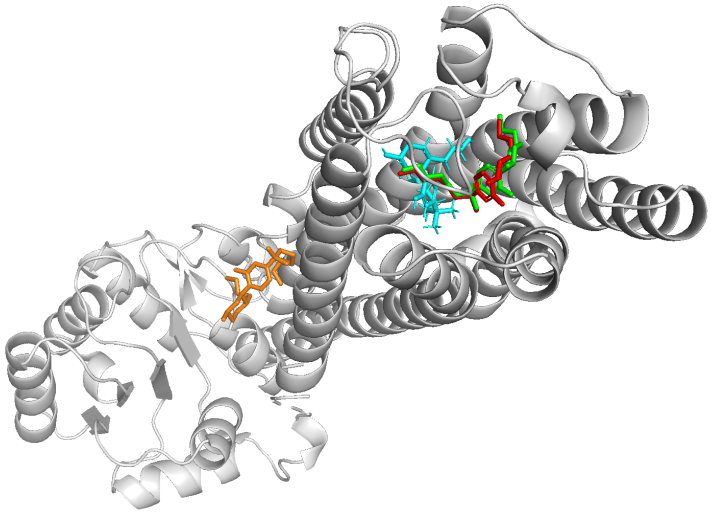

Figure 10: Visual inspection of PDB 6KQI. Another example of finding the native binding site when both the protein and the ligand are unseen. Crystallized ligand colored in green, TankBind prediction in red, EquiBind in cyan and Vina result in organ.

C Runtime details of different methods

The averaged runtime is shown in table 7. Baseline numbers are taken from EquiBind paper. The TankBind model has 1.8M parameters, comparable to EquiBind and GNINA, having 1.4M and 0.4M parameters respectively.

Table 7: averaged runtime per prediction for different methods.

Methods	AVG. SEC. 16-CPU	AVG. SEC. GPU
QVINA-W	49	-
GNINA	247	146
SMINA	146	-
GLIDE(c.)	1405*	-
VINA	205	-
EQUIBIND	0.16	0.04
TankBind	0.54	0.28

D Repeated runs of protein-ligand binding affinity prediction.

Table 8 provides more details than the figure in main text, including the mean and standard deviation of various methods across 3 experimental runs. Our model outperforms other models. For PIGNet and STAMP-DPI, we were unable to re-train the model, so we directly used the save-model provided by official repository for prediction.

Table 8: Comparison of predictive performance of ligand binding affinity using the PDBbind2020 dataset under time split.

Methods	RMSE ↓	Pearson ↑	Spearman ↑	MAE ↓
Sequence-based Methods				
TransformerCPI	1.741 ± 0.058	0.576 ± 0.022	0.540 ± 0.016	1.404 ± 0.040
MONN	1.438 ± 0.027	0.624 ± 0.037	0.589 ± 0.011	1.143 ± 0.052
Complex-based Methods				
PIGNet※	2.640*	0.511*	0.489*	2.110*
IGN	1.433 ± 0.028	0.698 ± 0.007	0.641 ± 0.014	1.169 ± 0.036
Structure-based Methods				
HOLOPROT	1.546 ± 0.065	0.602 ± 0.006	0.571 ± 0.018	1.208 ± 0.038
STAMPDPI※	1.658*	0.545*	0.411*	1.325*
TANKBind	1.346 ± 0.007	0.726 ± 0.007	0.703 ± 0.017	1.070 ± 0.019

E Additional ablation studies

Our ablation studies compose of mainly two categories. The first one is mainly associated with the framework of model, and the second one is associated with the training protocol. On the model side, TankBind-P is only doing the first summation over protein nodes inside the bracket of the equation 1, TankBind-C is only doing the second summation. We also included the results when the whole trigonometry module is only applied once, (single stack), and is completely removed, (no trig). For protein embedding, we tested using the pre-trained model TAPE to embed the protein instead of the GVP, and, for compound embedding, using GAT in place of GIN . On the side of training protocol, we tried replacing P2Rank binding sites with randomly selected binding sites, "TankBind-R", removing the random shift added to the center of protein block, "no random", and only training on the protein block contains the native ligand, "native only".

Table 9: Complete ablation results.

	Ligand RMSD						**Centroid Distance**					
	Percentiles ↓				% Below Threshold ↑		Percentiles ↓				% Below Threshold ↑	
Methods	25%	50%	75%	Mean	2Å	5Å	25%	50%	75%	Mean	2Å	5Å
baseline	2.45	3.96	7.67	7.43	19.28	61.71	0.87	1.74	4.22	5.51	56.47	77.41
TankBind-R	2.84	5.24	11.17	9.37	15.98	47.93	0.98	2.31	7.71	7.30	44.90	69.42
native only	3.01	7.14	21.49	12.92	17.08	41.05	1.05	4.68	20.23	11.34	37.47	51.52
no shift	2.65	3.94	7.73	7.57	19.56	58.68	0.79	1.75	4.53	5.60	55.10	76.58
GIN to GAT	2.48	4.05	7.72	8.27	19.01	57.02	0.82	1.66	4.19	6.23	56.47	78.51
TAPE	2.48	4.55	9.20	8.81	19.01	53.44	0.90	1.97	5.86	6.89	50.69	73.00
TankBind-C	2.38	4.47	8.36	8.20	19.56	54.82	0.93	1.87	5.41	6.27	53.17	73.28
TankBind-P	2.58	4.53	8.14	8.47	16.25	53.99	0.93	1.87	5.15	6.44	53.17	74.38
single stack	2.73	4.59	8.23	8.04	13.22	55.10	0.95	1.95	4.82	5.97	50.69	75.48
no Trig	3.34	5.26	8.56	8.73	4.13	47.93	1.25	2.22	5.01	6.44	44.08	74.93

F Example of the existence of equally valid binding sites that confuses the result

PDBbind curated the raw PDB file by only preserving a single ligand. In a few cases, when two or more identical chains are crystallized together, there are more than one valid binding site for the ligand. Here, we show an example with PDB 6MO9 in figure 11.

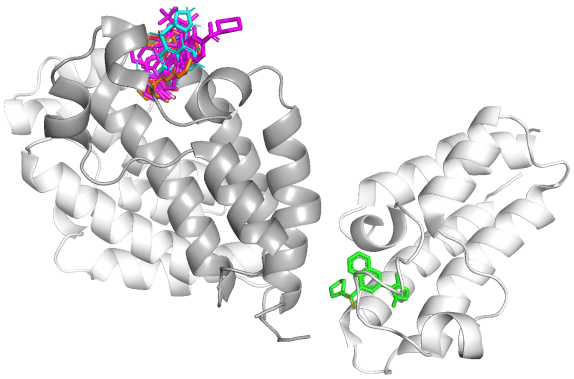

Figure 11: An example of equally valid binding sites that confuses the result. Green is the ligand preserved by PDBbind, but there is an equally valid binding site for each chain. PDB 6MO9.

G Details about protein segmentation

Protein graph is segmented for two main reasons: computational memory efficiency and biological functional generalization. One the computational side, since the size of proteins has a large variation, ranging from a few dozen amino acids to more than 3000 amino acids, the memory consumption to represent the protein and the interaction between protein and ligand could easily exceed the capacity of a typical GPU. On the biological side, large protein typically have multiple domains. Each protein domain, typically of size 200 amino acids, folds and interacts with ligand independently from the rest. Also, protein domains, as the building blocks of proteins, are more evolutionary conserved which means that a model explicitly learning on a domain level could generalize better to new domains. Each block is a sphere of radius 20Å typically includes about 200 amino acids, in line with the size of a protein domain. Block of radius 20Å is large enough to enclose the drug molecules, which is usually less than 15Å long, and small enough to be memory efficient.

We tried two ways of segmenting the whole protein. First approach is random segmentation; We randomly select an protein node, and use this node as the block center. But this approach is not efficient, since the binding site on protein has certain characteristic, more hydrophobic for instance. In second approach, we used a ligand-agnostic method, P2rank v2.3 (trained model was released on 2018) [41] to identify possible ligand binding sites, and use the centers of those potential binding sites as the block centers. For some small proteins, no binding site is identified, we therefore add an extra protein block located at the center of the whole protein. During training, we also add an extra protein block centering at the centroid of the co-crystallized ligand.

Despite P2rank assigns a score to each predicted binding site on the protein, the scores are fixed regardless of the interacting ligand because of the ligand-agnostic nature of P2rank. If we simply use the center of most probable binding site predicted by P2rank as the center of interaction block, this interaction block encloses the ligand for 73% of the test set. Our method, being a ligand-dependent method, improve the rate to 90%.

H Systematically analysis of the PDBbind dataset

In order to examine whether our model has the capability to place the compound to a unseen binding site, we aligned all the training protein-ligand complex structure with the same protein, defined by having the same *UniProt ID*, to the test set protein. *Kalign* is used to align the protein sequences first, and *Superimposer* function within package *BioPython* is used to align the structures. After the alignment, we computed centroid distance between aligned compounds and centroid of the test set compound. We found three cases in total having the centroid of all aligned training set compounds at least 10Å away from the centroid of test set ligand. They are PDB 6HMY, 6MO9 and 6K1S. With a visual examination of these PDBs along with aligned training set PDBs, we found that the, for PDB 6HMY and 6MO9, the apparently unseen binding site are caused by the process of crystallization and multimeric nature of certain proteins. For PDB 6MO9, the single chain protein having two identical binding sites due to packing during crystallization. For PDB 6HMY, the protein is a pentamer which means there are actually 5 identical binding sites for the complete protein complex. But in PDB 6K1S, we found a genuine unseen site. We found three training set PDBs having the same protein: 4X60, 4X61, 4X63. In order to remove the possibility that a close homolog exists, we computed the normalized Smith-Waterman alignment score, and found no homolog with score above 0.8 for PDB 6K1S other than the three PDBs with identical protein mentioned beforehand. Compound similarity is computed based on the Tanimoto Similarity of their Morgan fingerprint using RDKit.

I Hyper-parameters

The embedding sizes of the embedding of protein blocks and compound embedding are 128. The channel sizes of distance embedding are also 128. The trigonometry module is stacked 5 times. Transition module is a multilayer perceptron, where the hidden channel size is four times the input channel size. Dropout rate is set to 25%. layernorm is applied after each transition.

J Training details

During training, data is augmented in two ways. First the protein blocks that do not bind to the specific compound are used as decoys. The constrastive loss function is designed to ensure the compound binds weaker to those decoys than the native protein block. The margin, ϵ in constrastive loss is set to 1, corresponding to 1 order of magnitude in binding concentration. A protein block encloses the ligand when it covers more than 90% of the native interaction. Second, the model will see a slightly different protein block for every training data because the center of block will have a random shift of -5Å to 5Å, drawn from the uniform distribution, in all three axes. In bind re-docking, the native conformation is given as input, while in bind self-docking setting, the local atomic structures (LAS) mask, as defined in section 3.5 is applied to the compound node pair-distance map. The compound node pair distance map is based on the native conformation during training and based on the conformation generated by RDKit during testing. Our model include the models in ablation studies are trained for 200 epochs, after which no performance gain was observed. The model with the lowest validation loss was chosen as the best model. Each epoch has 20,000 randomly sampled block-ligand pairs. The total training process takes about 50 hours on a single NVIDIA RTX 3090 GPU. We use Adam optimizer with a constant learning rate of 0.0001.

K Implementation details of baselines for drug-protein binding structure prediction

Vina AutoDock Vina v1.2.3 is downloaded from `https://github.com/ccsb-scripps/AutoDock-Vina`. We follow the tutorial listed in `https://autodock-vina.readthedocs.io/en/latest/docking_basic.html` and use the center of ligand as the box center. The box size is set to 100Å and exhuastiveness is set to 32.

EquiBind EquiBind is downloaded from `https://github.com/HannesStark/EquiBind`. We follow the instruction and use saved model as listed in the GitHub. Our result slightly differs from the reported value. It could be due to a version change made by the developer, since its still an active repository. In the setting of "new receptors", following the same procedure, We got 142 "new receptors" PDBs while EquiBind got 144 (exact list not provided). We estimated that the 142 version and 144 version will affect the reported value by less than 2%. The result for other baselines are copied directly from the EquiBind paper for ease of comparison.

L Implementation details of baselines for binding affinity prediction

TransformerCPI We downloaded the code from the official repository `https://github.com/lifanchen-simm/transformerCPI`. we changed from the default classification task to regression task and switched to the PDBbind2020 dataset with time split, word2vec model is also retrained to extract sequence features based on the new dataset.

MONN We downloaded the code from the official repository `https://github.com/lishuya17/MONN`. The authors did not use a separate validation set, but instead used a clustering-based cross-validation strategy. We switched the data split mode to time split and repeated the original authors' data preprocessing steps on PDBbind2020 dataset.

PIGNet We downloaded the code and best save-model (best performance on CASF2016 benchmark) from their official repository `https://github.com/ACE-KAIST/PIGNet`. Due to the lacking of pre-processing scripts for data augmentation, we were unable to re-train the model using PDBbind2020. Instead, we used the best save-model presented by the authors. The result could be improved with additional data augmentation on the whole dataset instead of the PDBbind2019 refined set currently used for training.

IGN We downloaded the code from their official repository `https://github.com/zjujdj/InteractionGraphNet/tree/master`. The authors used PDBbind V2016 as an experimental dataset. We switched the data split mode to time split and repeated the data pre-processing protocol used by the authors on PDBBind2020.

HOLOPROT We downloaded the code from their official repository `https://github.com/vsomnath/holoprot`. The authors used PDBbind2019 refined set as an experimental dataset split by ligand scaffold and protein sequence. We followed the original authors' data pre-processing on PDBbind2020 and calculated the multi-scale representation of proteins. The model was retrained on this new dataset with the default setting.

STAMP-DPI We downloaded the code from their official repository `https://github.com/biomed-AI/STAMP-DPI`. The authors used PDBbind2016 general set as training set. We followed the original data pre-processing and performed on the split PDBbind 2020 test set. We were unable to extract all features required for training due to time constraints. The released model was employed to evaluate on the test set without re-training.

M Potential negative societal impacts

The greatest possible risk comes from virtual screening, such as practitioners blindly trusting the affinity predictions of learned models. Inspection of the predicted protein-ligand structure could help alleviate this issue, and fundamentally, further model Improvements and data augmentation to binding affinity prediction are essential to address this.