A. Methods

In this section, we present the details of our proposed **E**quivariant **T**ransformer **Dock**ing (**ETDock**) model for protein-ligand binding pose prediction. Our model comprises three fundamental components: (1) Feature processing module which merges the atom-level and graph-level features of ligands and learns the interactive characteristics between ligands and proteins. (2) TAMformer module that captures information from ligands and proteins by incorporating a triangle layer, an attention layer, and a message layer. (3) Ligand pose prediction generates a distance matrix for the protein-ligand complex and optimizes the docking pose based on this matrix iteratively.

A.1 Feature processing

A ligand is treated as a molecular graph $\mathscr{G}_l = (\mathscr{V}_l, \mathscr{E}_l)$ and the embedding $\mathbf{M}^l = [\mathbf{m}_1^l, ..., \mathbf{m}_i^l, ..., \mathbf{m}_s^l] \in \mathbf{R}^{d \times s}$ of the ligand is learned by graph isomorphism network (GIN) [1], where d is the dimension of embedding and s is the number of atoms in the ligand. Meanwhile, we use RDKit to generate graph-level features $\mathbf{m}_0^g$. $\mathbf{m}_0^g$ is produced utilizing the RDKit software, and it initially exhibits 1024 distinct characteristics [2, 3, 4]. These 1024 features are generated using RDKit, which computes a set of molecular descriptors (also known as fingerprints) that capture global structural information of the ligand. Then the learned atom-level features $\mathbf{m}_i^l$ and graph-level features $\mathbf{m}_0^g \in \mathbf{R}^{1024 \times 1}$ are fused to enable the model to better capture the complete information of the ligand.

A protein $\mathscr{G}_p = (\mathscr{V}_p, \mathscr{E}_p)$ is denoted as a K-nearest neighbor graph at the level of residues and the embedding of the protein $\mathbf{M}^p = [\mathbf{m}_1^p, ..., \mathbf{m}_j^p, ..., \mathbf{m}_q^p] \in R^{d \times q}$ is learned by geometric vector perceptrons (GVP) [5], where d is the dimension of embedding and q is the number of atoms in the protein. We use a learnable outer product to obtain the features of the protein-ligand interaction $\hat{\mathbf{z}} \in \mathbf{R}^{s \times q \times d}$.

A.1.1 Feature fusion

Previous methods usually focus on learning atom-level features while ignoring the graph-level features of ligands [6, 7], which limits their ability to effectively learn the complete information of ligands. To overcome this limitation, we integrate the atom-level and graph-level features (FCFPs) of ligands, allowing our model to capture more comprehensive information of ligands [4]. Since the feature spaces of atom-level and graph-level features are different, we use the following strategy to map the graph-level features to the atom-level feature space:

$$\mathbf{m}_1^g = MLP(\mathbf{m}_0^g), \tag{1}$$

where $\mathbf{m}_1^g \in R^{d \times 1}$ and $MLP(\cdot)$ is a multilayer perceptron which maps the graph-level features $\mathbf{m}_0^g$ to the atom-level feature space.

Then, similar to [3, 8], we fuse them in the same feature space through the following attention mechanism:

$$\mathbf{w}^g = softmax(\frac{(\mathbf{M}^l)^\top \mathbf{m}_1^g}{\sqrt{d}}), \tag{2}$$

$$\mathbf{m}_i^l = \mathbf{m}_i^l + \mathbf{M}^l \mathbf{w}^g, \tag{3}$$

where $(\mathbf{M}^l)^\top$ is the transpose of $\mathbf{M}^l$, and $\mathbf{w}^g$ is the attention between the individual atoms and the entire ligand.

A.1.2 Interaction feature

To capture the complex interactions between proteins and ligands, outer product is employed to derive the interaction features between ligands and proteins [7]. To enable parameter learning, we introduce a learnable outer product operation as follows:

$$\mathbf{m}_i^l = MLP(\mathbf{m}_i^l), \tag{4}$$

$$\mathbf{m}_j^p = MLP(\mathbf{m}_j^p), \tag{5}$$

$$\hat{\mathbf{z}}_{ij} = \mathbf{m}_i^l \otimes \mathbf{m}_j^p, \tag{6}$$

where $\otimes$ is the outer product. We leverage the multilayer perceptron to introduce learnable parameters into the outer product operation, which enables the outer product to learn parameters during training. This strategy facilitates the effective learning of interaction features, which enhances the prediction accuracy of protein-ligand docking pose.

A.2 TAMformer module

The TAMformer module includes three layers, i.e., a triangle layer, an attention layer and a message layer.

A.2.1 Triangle layer

Incorporating distance constraints within both the protein and the ligand can enhance the accuracy and physical plausibility of predicted binding poses. Therefore, it is important to consider intra-protein and intra-ligand atomic distances during the learning of interaction features [9]. Similar to TankBind [7, 10], we introduce the following triangle layer:

$$\mathbf{m}_{ij}^{tri} = \sum_{k=1}^{q} \mathbf{t}_{ik}^{p} D_{kj}^{p} + \sum_{k'=1}^{s} D_{ik'}^{l} \mathbf{t}_{k'j}^{l}, \tag{7}$$

$$\mathbf{z}_{ij} = \hat{\mathbf{z}}_{ij} + \mathbf{m}_{ij}^{tri} \odot gate(\hat{\mathbf{z}}_{ij}), \tag{8}$$

where $D_{kj}^{p} = \|\mathbf{m}_{k}^{p} - \mathbf{m}_{j}^{p}\|$, $D_{ik'}^{l} = \|\mathbf{m}_{i}^{l} - \mathbf{m}_{k'}^{l}\|$, $\mathbf{t}_{ik}^{p}$ (or $\mathbf{t}_{k'j}^{l}$) is obtained by applying a gated linear transformation on $\hat{\mathbf{z}}_{ik}$ (or $\hat{\mathbf{z}}_{k'j}$), i.e., $\mathbf{t}_{ik}^{p} = MLP(\hat{\mathbf{z}}_{ik}) \odot gate(\hat{\mathbf{z}}_{ik})$ and $gate(\hat{\mathbf{z}}_{ik}) = \sigma(MLP(\hat{\mathbf{z}}_{ik}))$, where $\sigma(\cdot)$ is a sigmoid function. $\hat{\mathbf{z}}_{ij}$ is initialized by Eq. (6), and when there are multiple layers of TAMformer, $\hat{\mathbf{z}}_{ij}$ is replaced by $\mathbf{z}_{ij}$ starting from the second layer. Unlike EquiBind, our model's incorporation of a triangular layer enables more effective capture of physical constraints governing ligand-protein interactions within the molecular framework.

A.2.2 Attention layer

To learn features of atoms in the molecular neighborhood [11, 12], we use self-attention to capture the deep interaction features between proteins and ligands. Since the protein-ligand interaction features are obtained from individual proteins and ligands through learnable outer product, they also contain information about proteins and ligands themselves. However, traditional self-attention applies attention globally across the entire sequence or graph, without the ability to separately focus on distinct components such as proteins and ligands in a protein-ligand interaction setting. To learn the individual information of proteins and ligands from the protein-ligand interaction features simultaneously, we introduce the following modified self-attention [13, 14]:

$$\mathbf{g}_{ij}^{0} = gate(\mathbf{z}_{ij}), \mathbf{g}_{ij}^{1} = MLP_3(\mathbf{g}_{ij}^{0}), \tag{9}$$

$$\mathbf{w}_{ij}^{z}, \mathbf{w}_{ij}^{l}, \mathbf{w}_{ij}^{p} = split(\nu_{ij} \odot \mathbf{g}_{ij}^{1}), \tag{10}$$

$$\mathbf{a}_{ijk'}^{z} = softmax(\mathbf{q}_{ij}\mathbf{k}_{ik'}^{\top}\mathbf{g}_{ij}^{0}), \tag{11}$$

$$\mathbf{h}_{ij}^{z} = \sum_{k'=1}^{q} \mathbf{a}_{ijk'}^{z} \odot \mathbf{w}_{ik'}^{z}, \tag{12}$$

where $\mathbf{q}_{ij}, \mathbf{k}_{ik'}, \nu_{ij}$ are linear projections of the protein-ligand interaction feature $\mathbf{z}_{ij}$, i.e., $\mathbf{q}_{ij} = MLP(\mathbf{z}_{ij})$ and $\mathbf{k}_{ik'} = MLP(\mathbf{z}_{ik'})$ and $\nu_{ij} = MLP_3(\mathbf{z}_{ij})$ (where $MLP(\cdot)$ denotes a multilayer perceptron with linear activation function, the dimension of $MLP_3(\cdot)$ is three times that of input and $split(\cdot)$ is used to divide the data into three parts [12]), $\mathbf{q} \in \mathbf{R}^{s\times q\times d}, \mathbf{k} \in \mathbf{R}^{s\times q\times d}, \nu \in \mathbf{R}^{s\times q\times 3d}$. $\mathbf{w}_{ij}^{z}$, $\mathbf{w}_{ij}^{l}$ and $\mathbf{w}_{ij}^{p}$ denote the values of the features of protein-ligand interactions, ligands and proteins, respectively. As the protein-ligand interaction features are generated from the features of proteins and ligands, so we can extract the original values of proteins and ligands from the values within the interaction features. $\mathbf{a}_{ijk'}^{z}$ is the self-attention weight of interaction feature. $\mathbf{h}_{ij}^{z}$ is the protein-ligand interaction feature. $\mathbf{g}_{ij}^{1}$ captures the information of ligand and protein from the value and $\mathbf{g}_{ij}^{0}$ captures the self-attention weight of the interaction features from the query and key. Finally, the interaction features output is obtained through $\mathbf{a}_{ijk'}^{z}$ and $\mathbf{w}_{ij}^{z}$. In comparison to existing methods such as TankBind, EquiBind, and DiffDock, our proposed attention layer is capable of learning the individual features of the ligand and protein from protein-ligand interaction features, thereby facilitating more effective message passing between the protein and ligand.

A.2.3 Message layer

Previous methods often overlooked the chemical features and 3D spatial information of ligands, proteins, and ligand-protein pairs [7, 6]. To address this, we introduce equivariant vectors to capture the equivariant information of ligands and proteins, updating these vectors using the equivariant graph neural networks (EGNNs) paradigm [15]. We then facilitate the interaction between invariant information and equivariant vectors. The scalar and vector blocks are employed to process the invariant and equivariant vectors before message passing.

The scalar block applies the protein-ligand interaction features to further interact with the ligand and protein. It utilizes the values from the attention layer to further focus on the ligand and protein, enabling the capture of their information from the interaction feature through the following equations:

$$\mathbf{h}_{ij}^{z}, \mathbf{h}_{ij}^{l}, \mathbf{h}_{ij}^{p} = split(\hat{\mathbf{h}}_{ij}^{z}), \tag{13}$$

$$\mathbf{h}_{ij}^{l} = \mathbf{h}_{ij}^{l} \odot \mathbf{h}_{ij}^{z} \odot \mathbf{w}_{ij}^{l}, \tag{14}$$

$$\mathbf{h}_{ij}^{p} = \mathbf{h}_{ij}^{p} \odot \mathbf{h}_{ij}^{z} \odot \mathbf{w}_{ij}^{p}, \tag{15}$$

where $\hat{\mathbf{h}}_{ij}^{z} = MLP_3(\mathbf{h}_{ij}^{z})$, $\mathbf{h}_{ij}^{z}, \mathbf{h}_{ij}^{l}, \mathbf{h}_{ij}^{p}$ denote the embeddings of protein-ligand interactions, ligands and proteins, respectively. $\mathbf{w}_{ij}^{l}$ and $\mathbf{w}_{ij}^{p}$ come from the attention layer, which is used to learn ligand and protein features from protein-ligand interaction features.

The vector block separates the protein-ligand interaction vector into ligand and protein vector features. The vector block learns the relative vector information between ligand and protein, as well as scalar features learned from the ligand and protein vector by the inner product.

$$\overrightarrow{v}_{ij}, \overrightarrow{v}_{ij}^{l}, \overrightarrow{v}_{ij}^{p} = split(\overrightarrow{v}_{ij}^{0}), \tag{16}$$

$$\overrightarrow{v}_{ij}^{-} = \overrightarrow{v}_{ij}^{l} - \overrightarrow{v}_{ij}^{p}, \tag{17}$$

$$v_{ij} = \left\langle \overrightarrow{v}_{ij}^{l}, \overrightarrow{v}_{ij}^{p} \right\rangle, \tag{18}$$

where $\overrightarrow{v}_{ij}, \overrightarrow{v}_{ij}^{l}, \overrightarrow{v}_{ij}^{p}$ are the vectors of interaction feature, ligand feature and protein feature. We initialize $\overrightarrow{v}_{ij}^{0} = MLP_3(\overrightarrow{v}_{ij}^{ini})$, $\overrightarrow{v}_{ij}^{ini} = \overrightarrow{\mathbf{e}}_{i}^{l} \otimes \overrightarrow{\mathbf{e}}_{j}^{p}$, where $\overrightarrow{\mathbf{e}}_{i}^{l} = MLP(\check{c}_{i}^{l})$ is the vector coordinate embedding of ligand and $\check{c}_{i}^{l}$ is the coordinate generated by the RDKit, and $\overrightarrow{\mathbf{e}}_{j}^{p} = MLP(c_{j}^{p})$ is the vector coordinate embedding of protein and c_{j}^{p} is the true coordinates of protein [2]. $\overrightarrow{v}_{ij}^{-}$ is the relative vector information between ligand and protein. $\langle \cdot, \cdot \rangle$ is the inner prodcut. v_{ij} is the scalar feature from ligand and protein vector feature by scalar product.

During message passing, we propagate information among the protein-ligand pair, ligand, and protein. Prior to this step, we use scalar and vector blocks to process invariant and equivariant vector information. This invariant and equivariant vector information is then used to guide the learning of features for the protein-ligand pair, ligand, and protein features through message passing. In this way, the invariant and equivariant vector information serve as a means of communication and coordination between the different features:

$$\begin{aligned} \mathbf{h}_{ij}^{z} &= v_{ij}\mathbf{h}_{ij}^{z}, \\ \mathbf{h}_{ij}^{l} &= v_{ij}\mathbf{h}_{ij}^{l}, \\ \mathbf{h}_{ij}^{p} &= v_{ij}\mathbf{h}_{ij}^{p}, \end{aligned} \tag{19}$$

$$\mathbf{z}_{ij} = \mathbf{h}_{ij}^{z} + \mathbf{h}_{ij}^{l} + \mathbf{h}_{ij}^{p}, \tag{20}$$

where v_{ij} comes from the vector block, and the vector information is integrated into the embedding of ligand and protein. Eq. (19) describes the interaction between scalar information of interaction, ligand, and protein with the vector. Finally, all the embeddings are summed up to update $\mathbf{z}_{ij}$ as described in Eq. (20).

After the message passing using Eqs. (19)-(20), to maintain the equivariance of the learned features and enhance their physical interpretability, we update the vectors using Eq. (21):

$$\overrightarrow{v}_{ij} = \overrightarrow{v}_{ij} + \psi(\mathbf{h}_{ij}^{l} + \mathbf{h}_{ij}^{p})\overrightarrow{v}_{ij}^{-}, \tag{21}$$

where $\psi(\cdot)$ is the mean operation over the embedding dimension. By incorporating invariant information of ligand and protein into the update of the protein-ligand interaction vector with Eq. (21), the invariant information can guide the update of vector information. By ensuring that the message passing layer maintains equivariance, and given that our entire ETDock architecture is designed to be equivariant, our model is better equipped to learn and utilize spatial information than TankBind.

A.3 Optimization objective

In this section, similar to [7], we introduce two loss functions for protein-ligand distance matrix and self-confidence.

A.3.1 Protein-ligand distance matrix

After processing the features of the ligand and protein using feature processing and TAMformer, similar to [7, 10], we use MLP to predict the final distance matrix $\hat{D}_{ij}$ between proteins and ligands, which can then be used to generate the docking pose:

$$\hat{D}_{ij} = MLP(\mathbf{z}_{ij}). \tag{22}$$

and then compute the following loss:

$$\mathscr{L}_a = \delta(r)\frac{1}{sq}\sum_{i=1}^{s}\sum_{j=1}^{q}\left\|\hat{D}_{ij} - D_{ij}\right\|, \tag{23}$$

where D_{ij} is the true distance between ligand and protein, and r is the pocket number by P2Rank [16]. In addition, we utilize root mean square error (RMSE) to minimize the difference between the predicted distance matrix $\hat{D}_{ij}$ and the true distance matrix D_{ij} of the protein and ligand. $\delta(r)$ is set to 1 when the ligand is close to the native pocket, and 0 otherwise [7].

A.3.2 Self-confidence

In protein-ligand docking, there may be multiple potential binding pockets on the protein surface. To address this challenge, we use the P2Rank algorithm to generate a list of the top ten possible binding pockets on the protein surface [16]. However, after the ligand binds to one of the pockets, the likelihood of the ligand binding to other pockets is greatly reduced. To account for this, we employ a self-confidence function $\mathscr{L}_b$, which considers the probability of the ligand binding to each of the ten predicted pockets:

$$\hat{f}_r = \sum_{i=1}^{s}\sum_{j=1}^{q} MLP(\mathbf{z}_{ij}), \tag{24}$$

$$\mathscr{L}_b(\hat{f}_r, f) = \delta(r)(\hat{f}_r - f)^2 + (1-\delta(r))max(0, \hat{f}_r - (f-\varepsilon))^2, \tag{25}$$

where $\hat{f}_r$ is the confident score of pocket r and ε is the margin value. We utilize binding affinity as the true label f. Our final optimization objective $\mathscr{L}$ is the sum of $\mathscr{L}_a$ and $\mathscr{L}_b$:

$$\mathscr{L} = \mathscr{L}_a + \alpha\mathscr{L}_b, \tag{26}$$

where α is a hyperparameter, which is set to 1 in our experiments.

A.4 Ligand binding pose generation

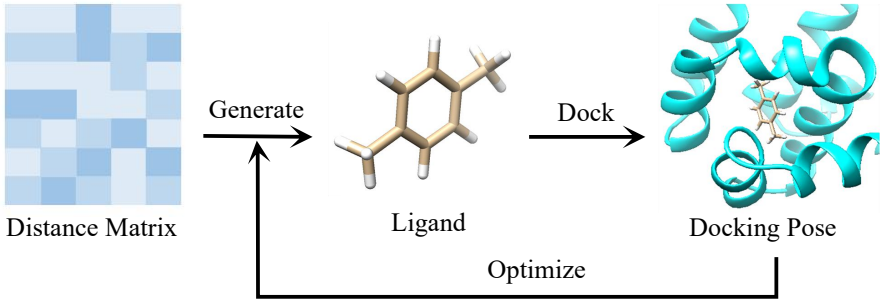

Fig. S1. The workflow for generating the ligand binding pose.

Our model can predict the distance map between the protein and ligand, so we use a two-stage approach to reconstruct the ligand's three-dimensional structure based on the protein-ligand distance map. Following [7, 10], we iteratively generate the final ligand binding structure by incorporating distance constraints within the ligand $\mathscr{L}_{inter}$ and leveraging local atomic structure constraints $\mathscr{L}_{stru}$:

$$\mathscr{L}_{inter} = \sum_{i=1}^{s}\sum_{j=1}^{q}(|\check{D}_{ij} - \hat{D}_{ij}|), \tag{27}$$

$$\mathscr{L}_{stru} = \sum_{j=1}^{s}\sum_{k=1}^{s}(|\check{D}^l_{jk} - D^l_{jk}|), \tag{28}$$

where $\check{D}_{ij} = ||\hat{c}^l_i - c^p_j||$ and $\check{D}^l_{jk} = ||\hat{c}^l_j - \hat{c}^l_k||$. c^p_i are the coordinates of protein nodes. $\hat{c}^l_i$ represents the final predicted coordinates of the ligand atoms. D^l_{jk} denotes the distance matrix between pairs of atomic coordinates within the ligand, which is computed using RDKit. The final generation objective $\mathscr{L}_{generate}$ is formulated as follows:

$$\mathscr{L}_{generate} = (1-\beta)\mathscr{L}_{inter} + \beta\mathscr{L}_{stru}, \tag{29}$$

where β is a hyperparameter. The workflow to generate the ligand binding pose is shown in Fig. S1.

B. Results and discussion

B.1 Experimental settings

B.1.1 Dataset

We assess the performance of our model in protein-ligand docking on the PDBbind dataset [17]. The PDBbind dataset includes structural data collected from the Protein Data Bank (PDB) along with associated experimental measurements [18]. The PDBbind dataset also provides the structural information of protein-ligand complexes, including atomic coordinates of proteins and structural and chemical information of ligands. Additionally, the PDBbind dataset contains experimentally determined binding affinities for protein-ligand complexes. We use the PDBbind v2020 dataset which has 19443 protein-ligand complexes and adopt the time split strategy described in EquiBind [6], i.e., our dataset is split based on the deposition year of protein-ligand complex structures. The training and validation sets include structures deposited before 2019, while the test set consists of structures deposited after 2019. By eliminating a subset of structures that could not be processed using RDKit, our training set consists of 17,787 structures [2]. For the purpose of validation, we allocate 968 structures, while 363 structures are designated for testing.

B.1.2 Overall comparison

Table S1. Experimental results on the PDBbind v2020 dataset.

	LIGAND RMSD						CENTROID DISTANCE					
	Percentiles ↓				%Below Threshold ↑		Percentiles ↓				%Below Threshold ↑	
Methods	25%	50%	75%	Mean	2 Å	5 Å	25%	50%	75%	Mean	2 Å	5 Å
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
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
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
GLIDE	2.6	9.3	28.1	16.2	21.8	33.6	0.8	5.6	26.9	14.4	36.1	48.7
EquiBind	3.9	6.3	10.5	8.3	4.1	40.2	1.2	2.7	6.9	5.5	40.2	69.7
TankBind	2.5	4.4	8.4	7.9	19.0	56.4	0.8	1.7	4.4	5.8	55.3	77.4
DiffDock	2.4	4.9	8.4	8.3	19.3	51.7	0.7	1.8	4.5	5.8	53.8	77.4
ETDock	**2.1**	**3.8**	**7.7**	**7.4**	**23.2**	**61.1**	**0.7**	**1.4**	**3.8**	**5.5**	**59.1**	**79.3**

B.1.3 Evaluation metrics

Following previous studies [7, 6], we evaluate the performance of our model using root mean square deviation (RMSD). In particular, we compute the RMSD (Root Mean Square Deviation) between the predicted and actual positions of ligand atoms, defined as the square root of the average squared Euclidean distance between corresponding atoms, to assess the accuracy of ligand pose prediction. Additionally, we employ RMSD between the centroid distances of the ligand and the true distances to measure the ability of our model in identifying the correct binding region. Furthermore, we utilize quantiles and mean values to assess the predictive performance of our model across different result ranges. Considering that an RMSD below 2 Å is deemed acceptable, we calculate the percentage of predicted ligand poses with an RMSD below 2 Å and below 5 Å. This analysis provides additional evidence of the ability of our model in predicting ligand poses. Note that when calculating these metrics, we exclude hydrogen atoms from the ligand.

B.1.4 Baselines

We compare our ETDock with state-of-the-art baselines from 2 categories as follows:

Search-based docking methods:

- AutoDock **VINA** is a popular molecular docking software that utilizes an efficient search algorithm and scoring function to explore the conformational space of ligands and predict their binding modes to target proteins [19].
- **SMINA** is an enhanced molecular docking software derived from AutoDock VINA, incorporating optimized search algorithms and scoring functions [20].
- **GNINA** enhances SMINA by incorporating a learned 3D Convolutional Neural Network for scoring [21].
- **QVINA-W** is a blind docking software that builds upon the speed-optimized QuickVina 2 by incorporating advanced algorithms for efficient exploration of ligand binding modes [22].
- **GLIDE** docking approach includes initial rough positioning, torsionally flexible energy optimization, and Monte Carlo sampling to obtain precise ligand docking [23].

Deep learning-based docking methods:

- **EquiBind** is an equivariant model that directly predicts the binding pose structure coordinates. It refines the ligand conformation using graph neural networks and aligns the refined ligand by keypoint alignment mechanism into the binding pocket [6].
- **TankBind** employs a deep learning approach to predict the protein-ligand distance matrix, enabling the generation of docking pose. By leveraging optimization algorithms, it reduces the reliance on exhaustive conformational sampling, improving computational efficiency in ligand binding prediction [7].
- **DiffDock** is a diffusion generative model for molecular docking. It maintains higher precision on folded structures and provides fast inference times and confidence estimates with high selective accuracy [24].

B.1.5 Implementation details

We trained our model using the Adam optimizer with a learning rate of 0.0001 for a total of 500 epochs. The model with the highest validation score was selected and subsequently evaluated on the independent test set. The training process was performed on single Tesla V100 GPU 32G and Inter Xeon Gold 5218 16-Core Processor. In our experiments, the results were derived from the average of five trials. Therefore, for baseline comparisons, we execute the EquiBind[1], TankBind[2], and DiffDock[3] inference codes five times with distinct random seeds and subsequently compute their mean values. Other baselines were adopted from TankBind [7]. Since DiffDock is a generative model, and other models are regression models, we only allow Diffdock to generate one ligand. The number of layers in TAMformer was customizable. In our model, we utilized 5 layers. The dataset was split according to the strategy used in EquiBind [6]. To prevent label leakage, we employed RDKit to generate the input coordinates for the ligands and used TorchDrug to extract their atomic features [25]. The dimensions of the node embeddings for both proteins and ligands were set to $d = 128$. For identifying functional regions, we employed P2Rank to identify the top 10 potential binding site regions in the protein. Additionally, for ligand binding pose generation based on the distance matrix, we used the Adam optimizer with a learning rate of 0.2 and performed 8000 iterations.

B.2 Performance evaluation

Table S1 shows the performance of our ETDock method compared to other baselines on the PDBbind v2020 dataset. ETDock outperforms existing search-based docking methods and deep learning-based docking methods. The percentage of results with ligand RMSD below 2 Å is 23.2%, while the percentage of results below 5 Å is 61.1%.

VINA employs a scoring function to search the ligand conformational space for protein–ligand docking. SMINA improves VINA's optimization algorithm, resulting in an increase in the percentage of ligand RMSD below 2 Å from 5.5% to 13.5%. GNINA employs a 3D convolutional neural network (CNN) as its scoring function, enabling the model to capture richer spatial information for improved protein–ligand docking. This approach leads to a significant enhancement in docking performance, with the percentage of ligand RMSD below 2 Å increasing from 13.5% to 21.2%. QVINA-W significantly accelerates the optimization process. GLIDE integrates energy-based optimization strategies, allowing it to achieve superior accuracy among search-based approaches. EquiBind, as the first deep learning-based method for protein–ligand docking, substantially reduces docking time but does not exceed the accuracy of the best-performing search-based methods [6]. TankBind integrates physical constraints and contrastive learning strategies [7], resulting in a significant performance improvement over EquiBind, i.e., the percentage of ligand RMSD below 2 Å increases from 4.1% to 19%. DiffDock employs a diffusion model to generate docking poses. The percentage of ligand RMSD below 2 Å increases from 19.0% to 19.3%. In comparison, our proposed ETDock framework leverages a Transformer-based architecture with built-in equivariant representations, achieving superior performance over all existing methods on the protein-ligand docking task.

B.2.1 Ablation studies

In ETDock, we have designed five modules, including feature fusion, the triangle layer, the attention layer, the message layer and equivariant vector. To assess the effectiveness of these modules, we have developed the following five variants of ETDock:

a. **ETDock-F**: The feature fusion module is removed from ETDock architecture.

b. **ETDock-T**: The triangle layer module is removed from ETDock to validate the effectiveness of the physical constraints.

c. **ETDock-A**: The attention layer module is removed from ETDock to assess the benefit of using the attention layer.

d. **ETDock-M**: The message layer module is removed from ETDock to validate the necessity of the message layer.

e. **ETDock-E**: The equivariant vector features learning component is removed from the message layer in ETDock to assess the impact of vector features on message passing.

In the aforementioned variants, each modification only affects one module while keeping the other modules in ETDock unchanged. The experimental results for the five variants are shown in Table S2. ETDock-M performs the worst, highlighting the importance of the interaction between invariant and equivariant vector information in the message layer. Additionally, the performance of ETDock-E indicates that the equivariant vector information in the message layer significantly enhances the overall results. Unlike EquiBind, TankBind, and DiffDock, our approach involves the interaction of scalar and equivariant vector information across three perspectives: ligands, proteins, and protein-ligand pairs. These results validate the importance of 3D spatial information in protein-ligand docking predictions. The results of the ETDock-T show the necessity of capturing physical constraints between ligands and proteins for effective feature learning. Unlike EquiBind and DiffDock, our method includes the trigonometric physical constraints of the protein and ligand in realistic situations. The performance of ETDock-F demonstrates the benefit of integrating atom-level and graph-level features of the ligand. In contrast to EquiBind, TankBind, and DiffDock, our approach takes into account the graph-level details of ligands and integrates them with the atom-level information. This allows for a more comprehensive understanding of the ligand's information during the learning process. The study conducted by ETDock-A proves that the attention layer effectively improves the extraction of ligand and protein embeddings from interactive embeddings. Moreover, unlike

[1] https://github.com/HannesStark/EquiBind

[2] https://github.com/luwei0917/TankBind

[3] https://github.com/gcorso/DiffDock

EquiBind, TankBind, and DiffDock, we capture specific information about individual ligands and proteins in protein-ligand pairs.

Table S2. The experimental results of ablation studies.

	LIGAND RMSD						CENTROID DISTANCE					
	Percentiles ↓				%Below Threshold ↑		Percentiles ↓				%Below Threshold ↑	
Methods	25%	50%	75%	Mean	2 Å	5 Å	25%	50%	75%	Mean	2 Å	5 Å
ETDock-F	2.3	4.1	8.1	7.7	19.6	58.9	0.8	1.7	4.2	5.7	55.9	77.1
ETDock-T	2.7	4.8	8.3	8.4	15.1	52.8	0.8	1.7	5.2	6.1	53.9	74.1
ETDock-A	2.4	4.2	7.5	8.0	18.4	58.1	0.7	1.7	4.1	6.0	55.6	78.7
ETDock-M	3.3	5.3	9.1	8.7	9.0	46.8	1.2	2.7	6.3	6.7	40.2	68.0
ETDock-E	2.6	4.4	8.4	8.8	18.1	55.9	0.8	1.6	5.0	6.8	56.1	75.0
ETDock	**2.1**	**3.8**	**7.7**	**7.4**	**23.2**	**61.1**	**0.7**	**1.4**	**3.8**	**5.5**	**59.1**	**79.3**

B.2.2 Performance evaluation on the cross-docking tasks

In the current dataset, only known protein–ligand pairs are included. In real-world applications, however, the model needs to predict the binding capability of an unseen protein with various ligands, thereby determining whether a ligand can exert therapeutic effects against the disease. So molecular docking is frequently performed on protein-ligand pairs for which crystal structures are unavailable. Therefore, we evaluated the cross-docking performance of ETDock to assess its effectiveness in these scenarios. Additionally, we tested the model on novel proteins that were not included in the training dataset [7]. As shown in Table S3, ETDock outperforms other docking methods in ligand-RMSD and centroid distance metrics when the values are below 2 Å. During cross-validation, the equivariant architecture of our model and its explicit incorporation of ligand-protein interaction constraints enable the systematic integration of 3D structural features and physicochemical descriptors, resulting in superior generalization capabilities compared to traditional methods.

Table S3. To assess cross-docking performance on novel proteins, all models were evaluated using 142 crystallographically resolved protein–ligand complexes involving proteins that were not part of the training set.

	LIGAND RMSD						CENTROID DISTANCE					
	Percentiles ↓				%Below Threshold ↑		Percentiles ↓				%Below Threshold ↑	
Methods	25%	50%	75%	Mean	2 Å	5 Å	25%	50%	75%	Mean	2 Å	5 Å
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
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
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
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
EquiBind	5.8	9.2	14.0	11.7	0.8	20.0	2.4	6.4	12.7	8.9	16.8	43.6
TankBind	2.9	**4.8**	**8.9**	9.2	4.6	**55.4**	1.2	2.4	4.9	7.3	45.0	**75.8**
DiffDock	**2.8**	5.9	9.1	9.5	11.6	48.9	**0.9**	2.7	**5.0**	6.2	43.2	75.4
ETDock	**2.8**	5.0	**8.9**	**8.8**	**11.7**	50.5	1.0	**2.2**	**5.0**	**6.1**	**47.5**	**75.8**

B.2.3 Performance evaluation on the PoseBusters benchmark dataset

In addition to utilizing the PDBind test sets, we also evaluated our model using the PoseBusters protein–ligand structure benchmark dataset [26], which comprises 428 protein–small molecule interactions. Furthermore, for a docking result to be considered successful on this test set, the ligand's root-mean-square deviation (RMSD) must be less than 2 Å, and specific spatial and energetic criteria must be met: the ligand conformation should be physically reasonable and free from steric clashes with the protein.

Furthermore, we conducted a comparative analysis of docking methods on the PoseBusters benchmark dataset, organized according to sequence identity relative to the PDBbind General Set v2020. In this context, sequence identity refers to the maximum sequence identity between any chains of the PoseBusters test proteins and any chains within the PDBbind General Set v2020. Since deep learning (DL)-based methods were trained on subsets of the PDBbind General Set v2020, this metric provides a quantitative measure of the disparity between the proteins in the test set and those used for training these methods. Based on their maximum percentage sequence identity, we classified the test cases into three categories: low [0–30%], medium medium (30–90%], and high (90–100%]. In the accompanying figure, the blue bars represent the proportion of predictions for each method where the root-mean-square deviation (RMSD) is less than 2 Å.

The red bars indicate predictions that not only meet the RMSD criterion but also pass all PoseBuster validation tests; these are considered PB-valid predictions.

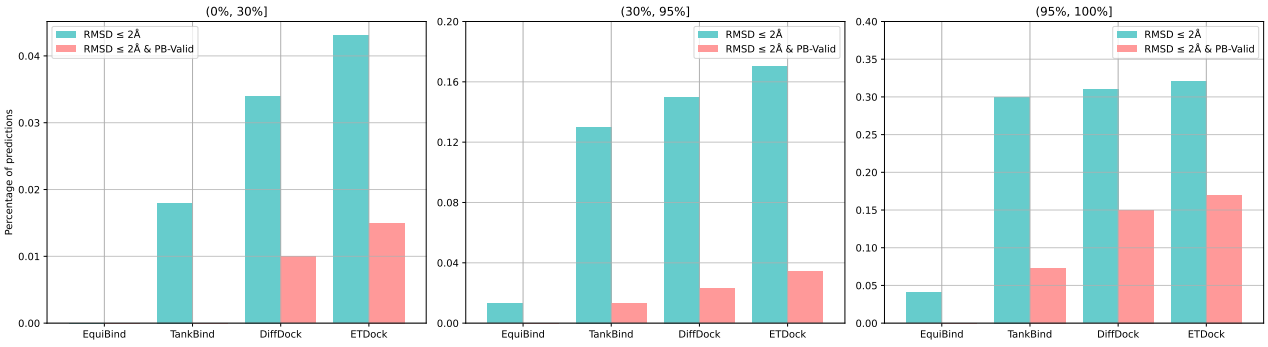

Fig. S2. Based on sequence identity relative to the PDBbind General Set v2020, we compared the performance of docking methods on the PoseBusters benchmark dataset.

From the experimental results in Fig. S2, EquiBind performs poorly on the PoseBusters dataset, with fewer predicted results considered valid. TankBind's PB-valid predictions in the [0%, 30%] range are significantly lower than those of DiffDock and ETDock. In the (30%, 90%] and (90%, 100%] ranges, the performance of TankBind, DiffDock, and ETDock is relatively similar, but ETDock outperforms the other methods.

B.3 Parameter effects

We conduct experiments to assess the effects of β and the number of iterations during ligand pose generation. In generating ligand poses through distance matrix, the parameter β in Eq. (29) has a significant impact. We conducted a parameter analysis on β, as shown in Fig. S3.

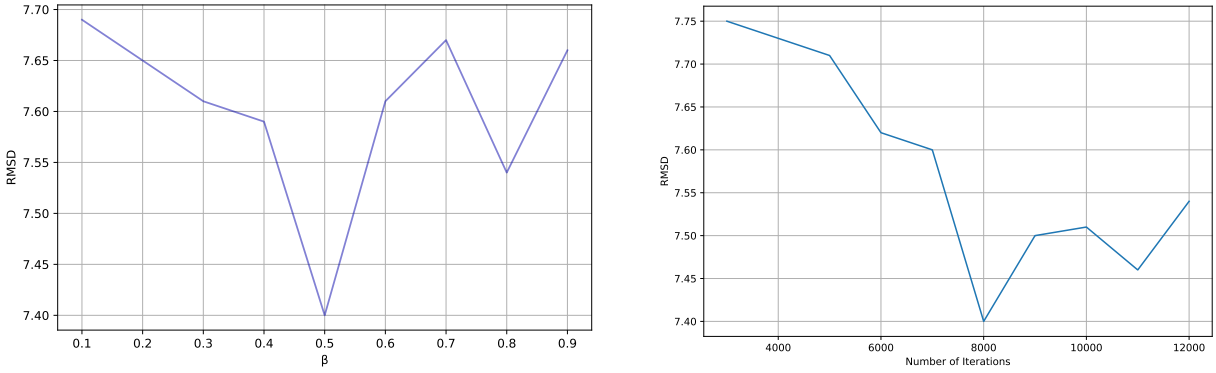

Fig. S3. The experimental results on the hyperparameter β(left) and the number of iterations during ligand pose generation(right). We utilize the average RMSD of the ligand to assess the impact of the number of iterations in the optimization algorithm.

β determines the weighting between two loss functions, $\mathscr{L}_{inter}$ and $\mathscr{L}_{stru}$. From Fig. S3, it can be observed that as the weight of $\mathscr{L}_{stru}$ increases, the RMSD of the ligand pose decreases continuously. We utilize the average RMSD of the ligand to assess. When β is set to 0.5, the ligand pose achieves the minimum RMSD. When β is increased beyond 0.5, with the weight of $\mathscr{L}_{stru}$ surpassing that of $\mathscr{L}_{inter}$, it has been observed that the performance of ligand pose generation tends to decline. This suggests that the incorporation of distance constraints within the ligand becomes increasingly valuable in the generation process. By assigning a higher weight to $\mathscr{L}_{inter}$, which represents the preservation of distance relationships, the algorithm can better capture and maintain the spatial arrangement of the ligand atoms. This emphasis on distance constraints aids in generating ligand poses with improved accuracy and alignment to the target structure.

During the ligand pose generation phase, we employ an iterative optimization algorithm based on the distance matrix. The number of iterations plays a crucial role in determining the precision of the resulting ligand pose and directly influences the computational time required for the process. Consequently, we conducted experimental analyses varying the number of iterations to assess its impact on the quality of the generated ligand poses in Fig. S3.

From Fig. S3, we observe a consistent decrease in the average RMSD of the ligand as the number of iterations increases. This indicates that with an increasing number of iterations, the ligand pose can be optimized to a more precise position. The minimum average RMSD for the ligand is achieved when the number of iterations reaches 8000. However, when we further increase the number of iterations, the average RMSD of the ligand starts to increase again. This suggests that the optimization process may become overly fine-tuned, resulting in diminishing returns.Considering both the diminishing improvement in RMSD and the increasing computational time as the number of iterations grows, we

select 8000 iterations as our final choice. This balance allows us to achieve a sufficiently accurate ligand pose optimization without excessive computational overhead.

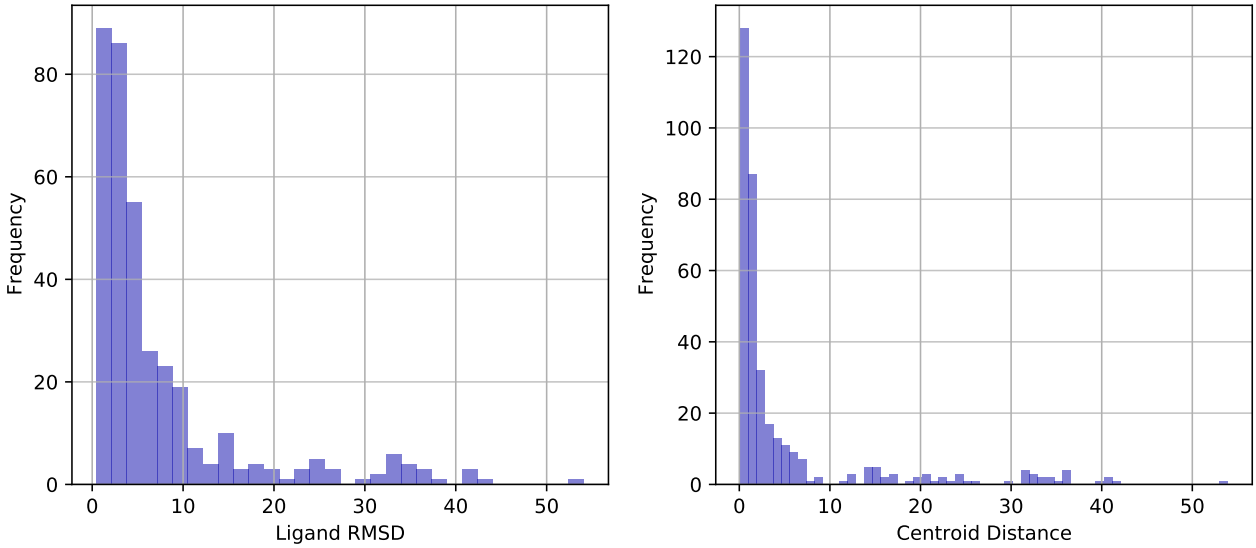

Fig. S4. The frequency histograms of Ligand RMSD (left) and Centeroid Distances (right) predicted by ETDock on the test set. These histograms provide a visual representation of the distribution and occurrence frequency of the RMSD and the centeroid distance for the predicted ligands.

B.4 Statistical analysis

In Table S1, the experimental results are presented using mean and quantiles, which may not provide a clear representation of the frequency distribution of the predicted ligand RMSD and centroid distance. To address this, we computed the RMSD and the centroid distance between the predicted ligand coordinates generated by ETDock on the test set and the corresponding true labels. When the RMSD between the predicted ligand and the true label is smaller, it implies that the predicted binding conformation is closer to the actual one. This not only reflects a more favorable binding interaction but also offers additional avenues for therapeutic development. We then used frequency histograms to visualize the distribution of these RMSD values and centroid distances, as shown in Fig. S4.

In Fig. S4, the ligand RMSD plot illustrates the performance of ETDock in predicting ligand conformational accuracy. It is evident that, out of the 363 ligands evaluated, more than 80 exhibit a predicted RMSD from the true label below the 2 Å threshold. This suggests that ETDock is highly effective in accurately predicting ligand conformations, with a substantial proportion of ligands achieving precision within the 2 Å range. Additionally, the centroid distance plot shows that over 120 ligands have a predicted centroid distance of less than 2 Å. This metric evaluates the accuracy of the predicted ligand centroid position relative to the true centroid position. The fact that a significant number of ligands achieve a centroid distance below 2 Å underscores ETDock's capability in accurately predicting the overall position of the ligand within the binding site. These results demonstrate the effectiveness of ETDock in generating ligand poses with a high degree of accuracy, as evidenced by the low RMSD values and centroid distances.

In ligand-protein docking tasks, we focus on the percentage of predicted ligand coordinates where both the RMSD and centroid distance are less than 2 Å compared to the true label. To better understand these percentages, we visualize them using the estimator of the cumulative distribution function (ECDF), which provides a clear representation of the value distribution, as shown in Fig. S5.

The empirical cumulative distribution function (ECDF) curves for ETDock, with respect to ligand RMSD and centroid distance, are shown in Fig. S5. Notably, the ECDF curve for ETDock consistently surpasses the curves for TankBind, EquiBind, and DiffDock within the 2 Å range. This clearly demonstrates that ETDock outperforms these methods in predicting protein-ligand binding poses, particularly for values below the 2 Å threshold. A closer examination reveals that the ETDock curve consistently lies above the curves of TankBind, EquiBind, and DiffDock, indicating that ETDock produces a higher proportion of ligand binding poses with lower RMSD and centroid distance values. This not only highlights ETDock's exceptional ability to generate ligand binding poses that closely match the true conformation within the 2 Å range but also underscores its strong performance beyond this threshold. In conclusion, these experimental results provide compelling evidence of ETDock's superior performance in accurately predicting ligand binding pose conformations, both within a specific range and across a broader spectrum.

B.5 Runtime analysis

In practical applications, it is crucial to consider not only the accuracy of protein–ligand docking algorithms but also the inference runtime of these models. In Fig. S6, we compute and compare the average time required for each method to generate a binding pose, including EquiBind, TankBind, DiffDock, and ETDock, using the PDBbind v2020 test set.

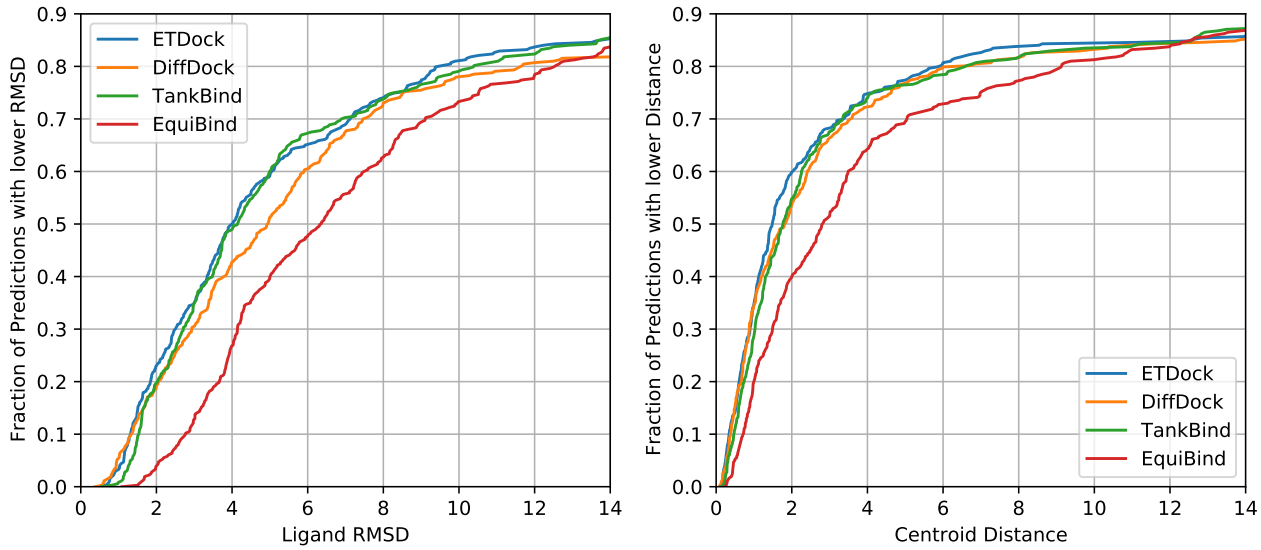

Fig. S5. Estimator of the Cumulative Distribution Function (ECDF) plot for ligand RMSD (left) and Centroid Distance (right) from result evaluated on the test dataset.

Since ETDock has a slightly larger number of parameters compared to TankBind and EquiBind, its computational runtime is correspondingly longer. The testing environments for all methods are the same as the one used to train ETDock, as detailed in Section Implementation details for further context.

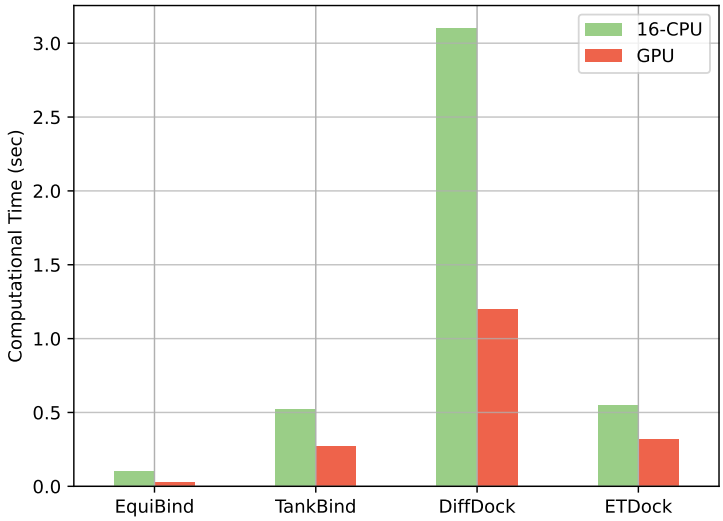

Fig. S6. The average runtime per prediction for different methods, including both GPU and CPU computation times. "16-CPU" refers to a 16-core processor.

Availability of data and materials

The data used in this study are publicly available and can be collected from PDBbind database (`http://www.pdbbind.org.cn/`). The codes are available at GitHub (`https://github.com/gnn4bio/ETDock`).

References

[1] Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826, 2018

[2] Landrum G, others . Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. Greg Landrum, 2013, 8(31.10): 5281

[3] Hua Y, Song X, Feng Z, Wu X J, Kittler J, Yu D J. Cpinformer for efficient and robust compound-protein interaction prediction. IEEE/ACM transactions on Computational Biology and Bioinformatics, 2022, 20(1): 285–296

[4] Deng H, Doonan C J, Furukawa H, Ferreira R B, Towne J, Knobler C B, Wang B, Yaghi O M. Multiple functional groups of varying ratios in metal-organic frameworks. Science, 2010, 327(5967): 846–850

[5] Jing B, Eismann S, Suriana P, Townshend R J, Dror R. Learning from protein structure with geometric vector perceptrons. arXiv preprint arXiv:2009.01411, 2020

[6] Stärk H, Ganea O, Pattanaik L, Barzilay R, Jaakkola T. Equibind: Geometric deep learning for drug binding structure prediction. In: International Conference on Machine Learning. 2022, 20503–20521

[7] Lu W, Wu Q, Zhang J, Rao J, Li C, Zheng S. Tankbind: Trigonometry-aware neural networks for drug-protein binding structure prediction. 2022, 7236–7249

[8] Hua Y, Song X, Feng Z, Wu X. Mfr-dta: a multi-functional and robust model for predicting drug–target binding affinity and region. Bioinformatics, 2023, 39(2): btad056

[9] Abramson J, Adler J, Dunger J, Evans R, Green T, Pritzel A, Ronneberger O, Willmore L, Ballard A J, Bambrick J, others . Accurate structure prediction of biomolecular interactions with alphafold 3. Nature, 2024, 630(8016): 493–500

[10] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, others . Highly accurate protein structure prediction with alphafold. Nature, 2021, 596(7873): 583–589

[11] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in Neural Information Processing Systems, 2017, 30

[12] Thölke P, De Fabritiis G. Torchmd-net: equivariant transformers for neural network based molecular potentials. arXiv preprint arXiv:2202.02541, 2022

[13] Liao Y L, Smidt T. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. arXiv preprint arXiv:2206.11990, 2022

[14] Morehead A, Chen C, Cheng J. Geometric transformers for protein interface contact prediction. arXiv preprint arXiv:2110.02423, 2021

[15] Satorras V G, Hoogeboom E, Welling M. E (n) equivariant graph neural networks. In: International Conference on Machine Learning. 2021, 9323–9332

[16] Krivák R, Hoksza D. P2rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. Journal of Cheminformatics, 2018, 10: 1–12

[17] Liu Z, Li Y, Han L, Li J, Liu J, Zhao Z, Nie W, Liu Y, Wang R. Pdb-wide collection of binding data: current status of the pdbbind database. Bioinformatics, 2015, 31(3): 405–412

[18] Burley S K, Bhikadiya C, Bi C, Bittrich S, Chen L, Crichlow G V, Christie C H, Dalenberg K, Di Costanzo L, Duarte J M, others . 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, 2021, 49(D1): D437–D451

[19] Trott O, Olson A J. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of Computational Chemistry, 2010, 31(2): 455–461

[20] Koes D R, Baumgartner M P, Camacho C J. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. Journal of Chemical Information and Modeling, 2013, 53(8): 1893–1904

[21] McNutt A T, Francoeur P, Aggarwal R, Masuda T, Meli R, Ragoza M, Sunseri J, Koes D R. Gnina 1.0: molecular docking with deep learning. Journal of cheminformatics, 2021, 13(1): 1–20

[22] Hassan N M, Alhossary A A, Mu Y, Kwoh C K. Protein-ligand blind docking using quickvina-w with inter-process spatio-temporal integration. Scientific Reports, 2017, 7(1): 15451

[23] Friesner R A, Banks J L, Murphy R B, Halgren T A, Klicic J J, Mainz D T, Repasky M P, Knoll E H, Shelley M, Perry J K, others . Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. Journal of Medicinal Chemistry, 2004, 47(7): 1739–1749

[24] Corso G, Stärk H, Jing B, Barzilay R, Jaakkola T. Diffdock: Diffusion steps, twists, and turns for molecular docking. arXiv preprint arXiv:2210.01776, 2022

[25] Zhu Z, Shi C, Zhang Z, Liu S, Xu M, Yuan X, Zhang Y, Chen J, Cai H, Lu J, others . Torchdrug: A powerful and flexible machine learning platform for drug discovery. arXiv preprint arXiv:2202.08320, 2022

[26] Buttenschoen M, Morris G M, Deane C M. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. Chemical Science, 2024, 15(9): 3130–3139