FIGRDock: Fast Interaction-Guided Regression for Flexible Docking

Shikun Feng [*1] **Bicheng Lin** [*2] **Yuanhuan Mo** [*3] **Yuyan Ni** [1 4] **Wenyu Zhu** [1] **Bowen Gao** [1] **Wei-Ying Ma** [1] **Haitao Li** [5] **Yanyan Lan** [1 6]

Abstract

Flexible docking, which predicts the binding conformations of both proteins and small molecules by modeling their structural flexibility, plays a vital role in structure-based drug design. Although recent generative approaches, particularly diffusion-based models, have shown promising results, they require iterative sampling to generate candidate structures and depend on separate scoring functions for pose selection. This leads to an inefficient pipeline that is difficult to scale in real-world drug discovery workflows. To overcome these challenges, we introduce FIGRDock, a fast and accurate flexible docking framework that understands complicated interactions between molecules and proteins with a regression-based approach. FIGRDock leverages initial docking poses from conventional tools to distill interaction-aware distance patterns, which serve as explicit structural conditions to directly guide the prediction of the final protein-ligand complex via a regression model. This one-shot inference paradigm enables rapid and precise pose prediction without reliance on multi-step sampling or external scoring stages. Experimental results show that FIGRDock achieves up to 100× faster inference than diffusion-based docking methods, while consistently surpassing them in accuracy across standard benchmarks. These results suggest that FIGRDock has the potential to offer a scalable and efficient solution for flexible docking, advancing the pace of structure-based drug discovery.

[*]Equal contribution [1]Institute for AI Industry Research (AIR), Tsinghua University [2]School of Life Sciences, Tsinghua University [3]School of Software Engineering, South China University of Technology [4]Academy of Mathematics and Systems Science, Chinese Academy of Sciences [5] School of Basic Medical Sciences, Tsinghua University [6]Beijing Academy of Artificial Intelligence. Correspondence to: Yanyan Lan <lanyanyan@air.tsinghua.edu.cn>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

1. Introduction

Molecular docking refers to predicting the three-dimensional structure of a protein–ligand complex given the individual structures of the protein and the small molecule. This task is fundamental to structure-based drug discovery, as it enables large-scale screening and mechanistic understanding of molecular interactions that underlie pharmacological effects. While conventional docking methods typically assume a rigid protein conformation, flexible docking models the conformational adjustments of both the ligand and the protein, especially those arising from induced-fit effects. By capturing this dynamic binding process, flexible docking provides a more biologically realistic framework, though it also introduces significant computational and modeling complexity.

Deep learning has recently brought significant advances to the flexible docking problem, offering data-driven alternatives to traditional physics-based methods. Among these, two major paradigms have emerged: co-folding approaches that predict complex structures directly from protein sequences, and generative approaches that operate on given unbound(apo) protein and ligand structures. Co-folding models, exemplified by AlphaFold 3 (Abramson et al., 2024), achieve impressive accuracy but remain computationally intensive due to the inherent complexity of protein structure prediction. In contrast, generative methods (Corso et al., 2022; Plainer et al., 2023; Huang et al., 2024b; Corso et al., 2025) leverage diffusion models to sample ligand poses, including global translation, rotation, and torsion angles of ligand rotatable bonds and protein side chains, conditioned on apo structures. By restricting the generative process to this product space, these methods significantly reduce the dimensionality of the prediction task, and offer substantial improvements in efficiency over co-folding approaches.

Despite these advances, current generative models still face notable limitations that hinder their

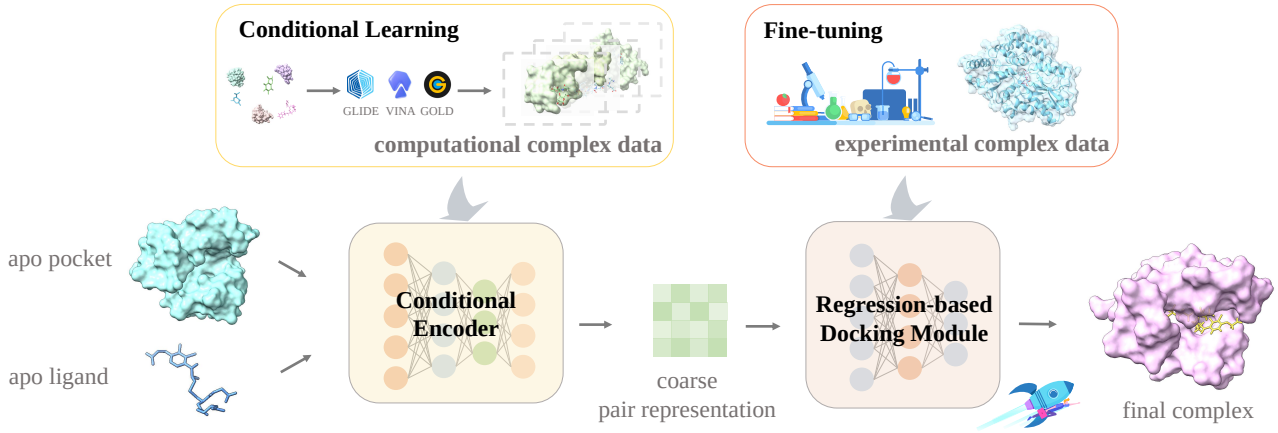

Figure 1. Overview of FIGRDock. The model mainly comprises two modules: a conditional encoder and a regression-based docking module. The conditional encoder, which is pre-trained using computational complex data, aims at providing a coarse pair representation. The regression-based docking module, fine-tuned on more accurate experimental complex data, is tailored to conduct flexible docking with efficiency and accuracy under the guidance of conditional pair representation.

practical deployment. They typically rely on multi-step sampling to produce accurate protein-ligand complexes and require repeated generation–scoring cycles, where performance improves with more iterations (Corso et al., 2022; Plainer et al., 2023; Huang et al., 2024b). Moreover, they depend heavily on pre-trained protein language models, such as ESM2 (Lin et al., 2023), to provide amino acid embeddings that serve as initial node features (Pei et al., 2023; Corso et al., 2022).

In contrast to diffusion-based models, regression-based approaches have been widely adopted in rigid docking frameworks such as EquiBind (Stärk et al., 2022) and TANKBind (Lu et al., 2022), offering superior efficiency via one-shot pose prediction. However, their performance in flexible docking scenarios is often suboptimal (Corso et al., 2022), as one-shot inference struggles to capture induced-fit effects and conformational changes in the ligand or protein. This gap raises a compelling question: **Can we design a regression-based docking framework that retains the efficiency of one-shot prediction while achieving high accuracy under flexible docking scenarios?**

To the best of our knowledge, accurate interaction modeling is essential to realize the full potential of regression-based docking. Unlike generative approaches, which refine predictions through multiple iterations, regression models infer binding conformations in a single pass. This one-shot approach demands precise interaction modeling, as there is no iterative correction process like in generative methods. In flexible docking scenarios, where even subtle conformational adjustments are critical, any misrepresentation of interactions can lead to significant deviations in predicted binding poses.

Building on this insight, we propose Fast Interaction-Guided Regression for Docking (FIGRDock). This method directly regresses to an accurate docking complex structure through a single network inference, guided by interaction representations, enabling both higher precision and greater efficiency in the docking process. FIGRDock's training is organized into two stages, as illustrated in Figure 1. The first stage involves conditional pair representation learning. We leverage the SIU dataset (Huang et al., 2024a), which contains a substantial amount of synthetic computational complex data generated by docking software, as pre-training data to learn interaction-informed paired representations between the protein and ligand. Despite the lower precision compared to crystallographic data, computational structures compensate for the limited availability of experimental structures. In the second stage, this learned pair representation is used as input for the regression-based docking module, followed by fine-tuning on more accurate crystal complex structures. The regression approach requires only a single network inference to predict the structure. Guided by the interaction pair representation, it produces more accurate predicted structures than iterative generative methods.

Experimental results show that when compared to

generative methods, FIGRDock reduces inference time from the order of tens of seconds to hundreds of milliseconds—a nearly 100× speedup. Furthermore, by leveraging pair representations as conditions, FIGRDock achieves superior performance across both holo and apo input test scenarios. To the best of our knowledge, this is the first regression-based method to achieve comparable or even better performance than diffusion-based methods. In the context of the dominance of generative models in flexible docking, our work offers a promising alternative approach that could provide valuable insights and solutions for future research in the field.[1]

2. Related work

In this section, we briefly review related work on flexible docking, focusing on two main approaches: co-folding methods and diffusion-based generative models.

2.1. Co-folding Methods

Co-folding methods aim to predict the three-dimensional structure of protein–ligand complexes in an end-to-end fashion. These approaches take as input a protein sequence and a molecular representation of the ligand, typically in the form of a molecular graph or SMILES string, and directly output the bound complex structure. Recent advances such as NeuralPLexer (Qiao et al., 2024), Umol (Bryant et al., 2024), AlphaFold3 (Abramson et al., 2024), and HelixFold3 (Liu et al., 2024) have demonstrated the effectiveness of this paradigm, achieving impressive accuracy in modeling protein–ligand interactions from minimal input information. However, the high computational demands of training and inference in these models pose significant challenges, limiting their scalability and practicality for large-scale virtual screening applications.

2.2. Diffusion-based Generative Models

Diffusion-based generative models have emerged as a leading paradigm for flexible docking. These methods take as input the unbound (apo) structures of both the protein and ligand and generate the bound complex structure by modeling the joint conformational changes that occur upon binding. Instead of searching over large configuration spaces or simulating the full folding process from the sequence, these models leverage generative diffusion processes to sample binding poses in a data-driven manner. Representative methods such as DiffDock-Pocket (Plainer et al., 2023), DiffBindFR (Zhu et al., 2024), and Re-Dock (Huang et al., 2024b) use diffusion or diffusion-bridge frameworks to capture pocket side-chain flexibility. FlexDock (Corso et al., 2025) and DynamicBind (Lu et al., 2024) further incorporate backbone flexibility using techniques such as unbalanced flow matching and geometric diffusion. While these approaches have shown strong accuracy in modeling flexible binding, they often suffer from inefficiencies due to iterative sampling and dependence on external scoring functions for pose selection.

Recently, there have been several initial attempts to alleviate the inefficiency problem. A representative example is FABFlex (Zhang et al., 2025), which directly predicts protein-ligand conformation with a regression model. Unlike diffusion-based methods that rely on iterative sampling, regression models aim to directly predict the final bound structure in a single forward pass, offering significantly improved inference efficiency. FABFlex, which takes the apo ligand and protein backbone as input and regresses the ligand pose along with the C_α coordinates of binding site residues. While this approach greatly reduces computational cost, its accuracy still lags behind state-of-the-art diffusion-based models. Moreover, because it does not explicitly model side-chain flexibility, where much of the binding-induced conformational change occurs, its ability to capture fine-grained interactions remains limited. These limitations motivate the development of more accurate and interaction-aware regression-based approaches, such as our proposed FIGRDock.

3. FIGRDock

In this section, we present our proposed method, FIGRDock. We begin by introducing the notations and formalizing the flexible docking task. We then describe the two key components of FIGRDock: (1) the conditional pair representation learning module, which captures interaction-aware features between the protein and ligand, and (2) the regression-based docking module, which directly predicts the bound complex structure in a single forward pass.

3.1. Preliminaries and Problem Formalization

A protein-ligand complex can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{X})$, where $\mathcal{V}$ represents the set of atom types v_i for the vertices i, and $\mathcal{X}$ represents the set of coordinates x_i for each vertex. The complex can be divided into two parts: a ligand and a protein. The ligand part is represented as $\mathcal{G}^l = (\mathcal{V}^l, \mathcal{X}^l)$, and the protein part is represented as $\mathcal{G}^p = (\mathcal{V}^p, \mathcal{X}^p)$. The atom types for the small molecule (ligand) are consistent with those defined by the periodic table (e.g., C, N, O...). However, for the pocket atom types, to model information about side-chain variations, we treat the same element at different positions on the side chain and backbone as different types. For instance, for the element Carbon (C), we distin-

[1]The code is publicly available at https://github.com/fengshikun/FIGRDock

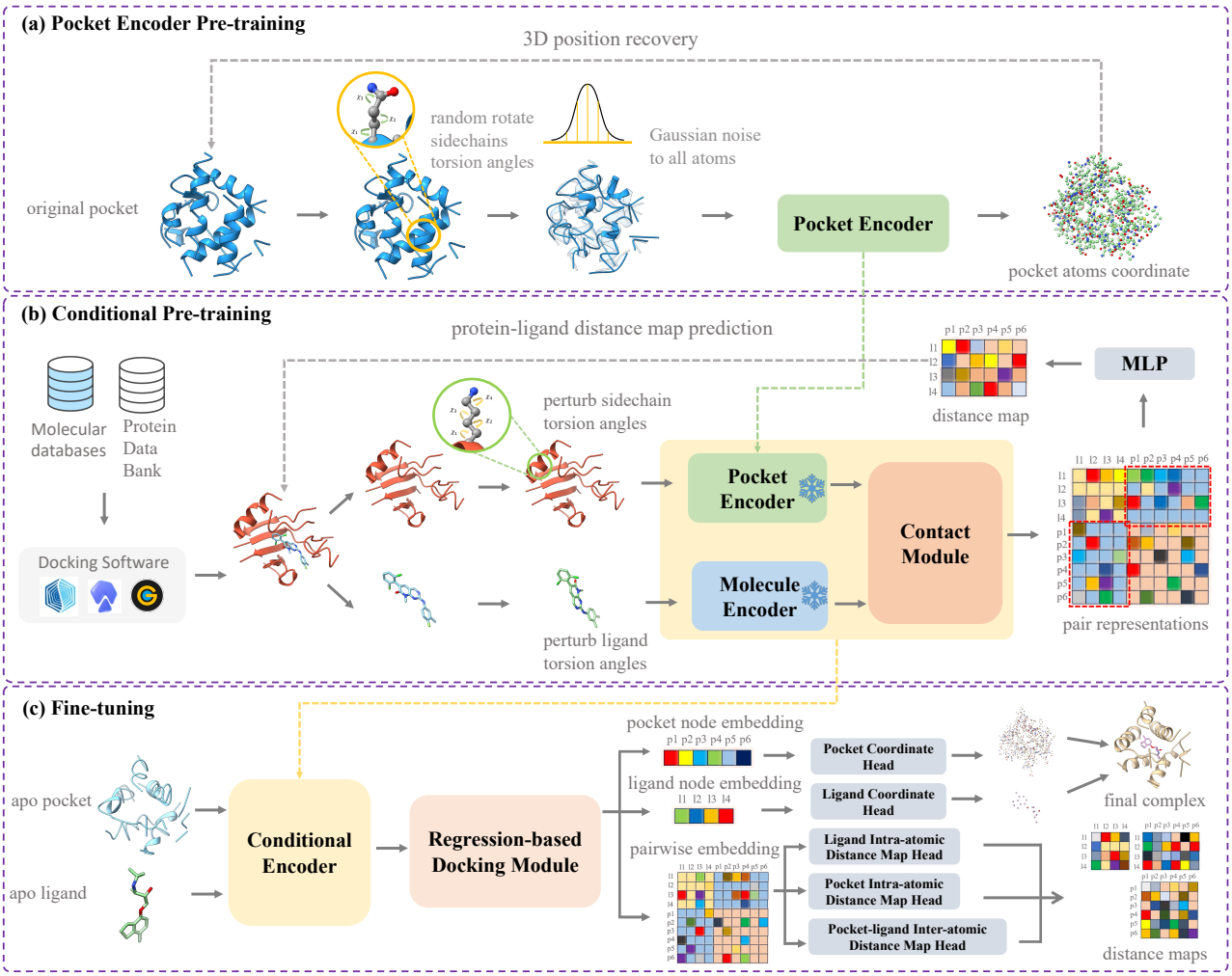

Figure 2. Illustrations of FIGRDock consist of three components. **a**: Apply a combination of noise to the pocket, including perturbed dihedral angles and coordinates, then denoise it to train a pocket encoder that is aware of side chains. **b**: Use coarse structures generated by docking software to learn a conditional pair representation. **c**: Fine-tune on accurate crystal complex structures, using the coarse conditional pair representation to guide the regression-based docking module during the fine-tuning process.

guish between C (backbone carbonyl), CA (alpha carbon), CB (beta carbon), etc. (Refer to the Appendix 7.1 for the complete list of side-chain atom types).

In the flexible docking setting, the goal is to predict the structural changes of both the protein and the small molecule that occur during the binding process. Given the unbound apo structures of the ligand and the protein, represented as $\mathcal{G}^{l^*} = (\mathcal{V}^l, \mathcal{X}^{l^*})$ and $\mathcal{G}^{p^*} = (\mathcal{V}^p, \mathcal{X}^{p^*})$, respectively, the task is to predict their bound conformation: $\mathcal{G}^l = (\mathcal{V}^l, \mathcal{X}^l)$ for the ligand and $\mathcal{G}^p = (\mathcal{V}^p, \mathcal{X}^p)$ for the protein. This requires modeling the mutual conformational adjustments that occur upon binding, making the task significantly more complex than rigid docking.

3.2. Conditional Pair Representation Learning

Figure 2a illustrates the process for learning conditional pair representations. Initially, the protein pocket and the ligand are processed by two separate pre-trained encoders to obtain initial node representations: $h_p = \theta_p(\mathcal{G}^{p^*})$ and $h_l = \theta_l(\mathcal{G}^{l^*})$. Here, θ_p and θ_l denote the encoders for the pocket and the molecule, respectively. For the ligand encoder θ_l, we adopt the pre-trained molecular encoder from Uni-Mol (Zhou et al., 2023).

To pre-train the pocket encoder θ_p, we design a side-chain denoising task using pocket data provided by ProFSA (Gao et al., 2023). Specifically, we apply a combined noise scheme to the original pocket $\mathcal{G}^p$: first, we perturb its rotatable dihedral angles, and second, we add Gaussian noise to the coordinates of all its atoms. This process yields the

noised pocket $\mathcal{G}^{\tilde{p}}$. The learning objective for the pocket encoder pre-training can be represented as:

$$\mathcal{L}_d = \mathbb{E}_{\mathcal{G}^p, \mathcal{G}^{\tilde{p}}} ||f_p(\theta^p(\mathcal{G}^{\tilde{p}})) - (\mathcal{X}^{\tilde{p}} - \mathcal{X}^p)||_2^2, \quad (1)$$

Among them, $\mathcal{X}^{\tilde{p}}$ and $\mathcal{X}^p$ represent the coordinates of $\mathcal{G}^{\tilde{p}}$ and $\mathcal{G}^p$ respectively, and f_p represents an MLP (Multi-Layer Perceptron) structure, which is used to predict the coordinate noise of the pocket from the pocket representation.

Subsequently, using the initial node representations h_l and h_p obtained from the separate encoders, we learn the conditional pair representation utilizing complex data from SIU (Huang et al., 2024a) (generated by docking software calculations). Specifically, for this stage, we perturb the rotatable dihedral angles within the ligand and the side chains of the pocket in the complex data. This generates noised, approximately apo-like conformations denoted as $\mathcal{G}^{\hat{l}}$ (ligand) and $\mathcal{G}^{\hat{p}}$ (pocket). Let $\mathcal{D}_{pl}$ represent the distance matrix of the holo pocket structure and the small molecule structure within the complex. As shown in Figure 2b, the purpose of the interaction network module θ_i is to take the noisy conformations $\mathcal{G}^{\hat{p}}$ and $\mathcal{G}^{\hat{l}}$ as inputs, and learn the conditional pair representation by predicting $\mathcal{D}_{pl}$. Specifically, the loss function can be defined as:

$$\mathcal{L}_c = \mathbb{E}_{\mathcal{G}^{\hat{l}}, \mathcal{G}^{\hat{p}}} ||f_i(h_{\hat{pl}}) - \mathcal{D}_{pl}||_2^2, \quad (2)$$

Where f_i represents the MLP utilized for predicting the distance matrix, and $h_{\hat{pl}} = \theta_i(\theta_l(\mathcal{G}^{\hat{l}}), \theta_p(\mathcal{G}^{\hat{p}}))$ represents the conditional pair representation learned by network θ_i. During the training of θ_i, the parameters of θ_p and θ_l are kept frozen.

3.3. Regression-based Docking Module

As shown in Figure 2c, the regression-based docking module θ_r takes the unbound (apo) structures $\mathcal{G}^{l^*}$ and $\mathcal{G}^{p^*}$, along with the learned pair representation h_{pl}, as inputs to predict the bound (holo) complex structures $\mathcal{G}^l$ and $\mathcal{G}^p$.

To enable direct coordinate regression while capturing both intra- and inter-molecular structural constraints, we construct three distance matrices:

- $\mathcal{D}_l$: the intra-ligand atomic distance matrix,
- $\mathcal{D}_p$: the intra-protein atomic distance matrix,
- $\mathcal{D}_{pl}$: the inter-molecular atomic distance matrix between ligand and protein atoms.

These matrices are computed from the predicted coordinates and serve as targets in our training objective. Specifically, the coordinate prediction loss for the ligand is defined by comparing the predicted intra-ligand atomic distances with the ground truth, encouraging the model to preserve realistic molecular geometry.

$$\mathcal{L}_{ligand} = \mathbb{E}_{\mathcal{G}^{p^*}, \mathcal{G}^{l^*}} (||f_{lc}(\hat{h}_l) - (\mathcal{X}^l - \mathcal{X}^{l^*})||_2^2 + ||f_{ld}(\hat{h}_{pl}) - \mathcal{D}_l||_2^2). \quad (3)$$

Similarly, the coordinate prediction loss for the pocket can be expressed as follows to enforce physically realistic geometry within the binding pocket:

$$\mathcal{L}_{pocket} = \mathbb{E}_{\mathcal{G}^{p^*}, \mathcal{G}^{l^*}} (||f_{pc}(\hat{h}_p) - (\mathcal{X}^p - \mathcal{X}^{p^*})||_2^2 + ||f_{pd}(\hat{h}_{pl}) - \mathcal{D}_p||_2^2). \quad (4)$$

Lastly, the following loss is defined to penalize deviations between the predicted and ground-truth inter-molecular atomic distance matrix across the protein-ligand interface:

$$\mathcal{L}_{interface} = \mathbb{E}_{\mathcal{G}^{p^*}, \mathcal{G}^{l^*}} (||f_{pld}(\hat{h}_{pl}) - \mathcal{D}_{pl}||_2^2). \quad (5)$$

In the above losses, $\hat{h}_l, \hat{h}_p, \hat{h}_{pl} = \theta_r(\mathcal{G}^{l^*}, \mathcal{G}^{p^*}, h_{pl})$ denotes the resulting node embedding of ligand, node embedding of pocket and pairwise embedding encoded by θ_i, respectively. The f_{lc}, f_{ld}, f_{pc}, f_{pd}, and f_{pld} represent the head network for the prediction of the coordinate matrix and the distance.

The total regression docking loss is the sum of these three components:

$$\mathcal{L}_r = \mathcal{L}_{ligand} + \mathcal{L}_{pocket} + \mathcal{L}_{interface}. \quad (6)$$

4. Experiments

4.1. Main Experiments

Experimental Setup For pocket encoder pre-training, we use pocket data provided by ProFSA (Gao et al., 2023) to perform sidechain-aware pre-training. Since the noise-adding process involves perturbing sidechain torsions, we filtered the dataset to remove samples with incomplete sidechains, reducing the total number of samples from 5 million to 4.8 million. The pre-training was conducted on 4 GPUs for 10 epochs with a batch size of 64, taking approximately 6 days to complete.

For conditional pre-training, we pre-train on the SIU (Huang et al., 2024a) dataset, which consists of 5.34 million complex conformations generated by docking software. The training was conducted using 4 A100 GPUs with a batch size of 16, and the pre-training took approximately 20 days to complete.

In the fine-tuning stage, we fine-tune FIGRDock on the commonly adopted PDBbind v2020 dataset(Wang et al., 2005), which contains 19K crystal complex structures. We employ the time-split of PDBbind with 17k complexes from 2018 or earlier for training and validation, and 363 test structures from 2019, ensuring consistency with previous

works(Stärk et al., 2022; Corso et al., 2025). The input apo ligand conformation is generated using RDKit with a random seed, while the input apo protein structure is predicted by ESMFold (Lin et al., 2023). The fine-tuning is performed on 4 A100 GPUs for 100 epochs with a batch size of 16, taking approximately 3 days to complete. Detailed hyperparameters can be found in the Appendix 7.1.

Evaluation Metric We evaluate FIGRDock on the PDBbind test set and the PoseBusters V2 (Buttenschoen et al., 2024) test set. The PoseBusters V2 Benchmark is a curated collection of 308 high-quality, drug-like protein–ligand crystal complexes released after 2021, specifically designed to assess docking methods not only in terms of RMSD but also based on chemical and geometric plausibility through RDKit-based quality checks. The primary evaluation metric is the RMSD of Cartesian coordinates. We report the percentage of samples with RMSD below different thresholds, specifically < 2Å and < 5Å for ligands, along with the median RMSD value across all samples. We also report the average runtime to evaluate the model's efficiency. Finally, for the PoseBusters benchmark, we report the PBValid score, which reflects the model's ability to generate chemically and structurally reasonable conformations.

Baselines For the PDBbind benchmark, we compare FIGRDock with search-based models SMINA (Koes et al., 2013) and GNINA (McNutt et al., 2021), which are traditional methods employing scoring functions and search algorithms to effectively explore ligand poses at a considerable computational cost. We also compare FIGRDock with generation model-based pocket-level docking methods, DiffDock-Pocket (Plainer et al., 2023), ReDock (Huang et al., 2024b), and FlexDock (Corso et al., 2025). For the PoseBusters V2 benchmark, we measure FIGRDock against search-based models GOLD (Verdonk et al., 2003) and VINA (Trott & Olson, 2010), generation model-based FlexDock (Corso et al., 2025), and co-folding models UMol (Bryant et al., 2024) and AlphaFold3 (Abramson et al., 2024).

4.1.1. PDBBIND

As shown in Table 1, we compare FIGRDock's performance and runtime with search-based models SMINA (Koes et al., 2013) and GNINA (McNutt et al., 2021), sidechain flexible models DiffDock-Pocket (Plainer et al., 2023) and ReDock (Huang et al., 2024b), and all-atom flexible model FlexDock (Corso et al., 2025). We report results for both rigid and flexible versions of SMINA and GNINA. Overall, FIGRDock outperforms existing methods in both accuracy and inference efficiency. In terms of RMSD performance, metric %RMSD<2Å is a crucial metric as the predicted structure is considered successful when it meets this criterion. FlexDock (Corso et al., 2025) is the only generative model that has modeled all atoms, making it the most equitable model for comparison. FIGRDock significantly outperformed FlexDock (Corso et al., 2025) in metric %RMSD<2Å by nearly 7% (46.6% vs. 39.7%) with apo input. Furthermore, FIGRDock improves upon the previous best-performing method, ReDock (Huang et al., 2024b) in metric %RMSD<2Å, by nearly 4% with both holo and apo input (57.2% vs. 53.9% and 44.6% vs. 42.9%). At the same time, in terms of model efficiency, FIGRDock significantly accelerates inference compared to ReDock (Huang et al., 2024b), achieving over a 100-fold speedup (0.4s vs. 58s). This demonstrates that under the guidance of conditional pair embeddings, the regression-based module, which avoids repetitive sampling and iterative reasoning, not only substantially enhances efficiency but also maintains state-of-the-art accuracy.

4.1.2. POSEBUSTERS

On PoseBusters, as shown in Figure 3, rigid docking methods including DeepDock (Méndez-Lucio et al., 2021), Uni-Mol (Zhou et al., 2023), GOLD (Verdonk et al., 2003) and VINA (Trott & Olson, 2010) receive holo pockets as input. While flexible docking methods, including FlexDock and FIGRDock use apo input generated by ESMFold. We also report the result that FIGRDock uses holo as input. Finally, co-folding methods like Umol (Bryant et al., 2024) and AlphaFold3 (Abramson et al., 2024) take sequences as input. FIGRDock performs significantly better than FlexDock (Corso et al., 2025) as well as other deep learning based rigid docking models like DeepDock (Méndez-Lucio et al., 2021) and Uni-Mol (Zhou et al., 2023). Compared to search-based methods like GOLD (Verdonk et al., 2003) and VINA (Trott & Olson, 2010), although FIGRDock with apo input falls slightly behind, it is much faster and takes on a prominently harder task. FIGRDock achieves better performance than GOLD and Vina with holo input. AlphaFold3 (Abramson et al., 2024) significantly outperformed all methods, as it is trained using a larger volume of data. FIGRDock can generate more physically plausible conformations, achieving 99.5% and 96.7% PBValid for apo and holo input. Details of validity checks for the PoseBusters V2 benchmark are deferred to Appendix 7.2.

4.2. Ablation study

4.2.1. COMPARISON WITH ESM EMBEDDING

In this study, we compare our approach with the traditional method that uses protein language model–generated embeddings as conditions, as reported in Table 2. 'Without condition' refers to the model trained without any conditional pre-training, while 'With ESM' denotes the use of amino acid–level node embeddings extracted from ESM2 (Lin et al., 2023). Our method, FIGRDock, employs the proposed conditional pair representation. All settings use the

Table 1. RMSD performance and runtime comparison of different methods on the PDBbind dataset. The best results are highlighted in **bold**. FIGRDock demonstrates a significant advantage in both accuracy and efficiency.

Models	Holo Crystal Proteins			Apo ESMFold Proteins			Average Runtime (s)
	%<2 ↑	%<5 ↑	Med.↓	%<2 ↑	%<5 ↑	Med.↓	
SMINA(rigid)	32.5	54.7	4.5	6.6	22.5	7.7	258
SMINA	19.8	47.9	5.4	3.6	20.5	7.3	1914
GNINA(rigid)	42.7	67.0	2.5	9.7	33.6	7.5	260
GNINA	27.8	54.4	4.6	6.6	28.0	7.2	1575
DiffDock-Pocket(40)	49.8	79.8	2.0	41.7	74.9	2.6	61
ReDock(40)	53.9	80.3	1.8	42.9	76.4	2.4	58
FlexDock	-	-	-	39.7	-	2.5	11
FIGRDock	**57.2**	**82.3**	**1.6**	**46.6**	**76.8**	**2.3**	**0.4**

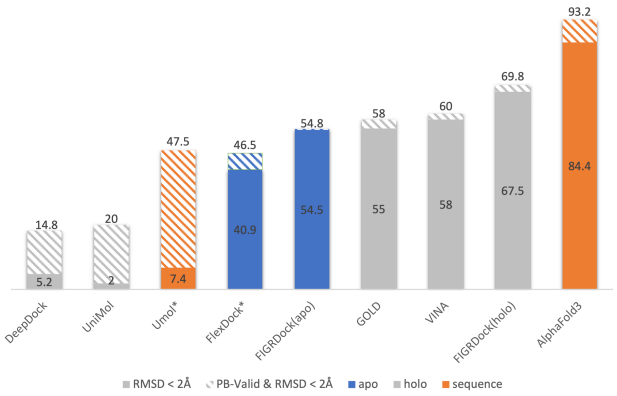

Figure 3. Results of the PoseBusters V2 benchmark with known pockets. FIGRDock outperforms the flexible docking method FlexDock with apo input. Meanwhile, FIGRDock outperforms search-based methods (Gold and Vina) with holo input. For methods marked with *, we demonstrate results reported by the FlexDock paper (Corso et al., 2025).

same network architecture and fine-tuning strategy to ensure a fair comparison.

Results shown in Table 2 demonstrate that our interaction-guided approach provides substantial benefits for molecular docking. First, the significant performance improvement of our method over the unguided baseline validates the effectiveness of our conditional learning strategy. Moreover, FIGRDock consistently outperforms the 'With ESM' setting across both holo and apo inputs—especially for apo ESMFold proteins, where the success rate (%RMSD<2Å) increases by nearly 9%. This performance gain is particularly notable considering that our approach incurs significantly lower training costs than ESM2. These findings suggest that representations capturing interaction-specific knowledge offer more relevant and efficient guidance for docking tasks compared to general-purpose protein representations.

Table 2. RMSD performance comparison of different protein representations for docking. 'Without condition' uses no conditional pre-training; 'With ESM' uses ESM2-based residue level node embeddings; FIGRDock employs interaction-aware conditional representations. The best results are in **bold**, showing the advantage of our approach over general protein features.

Models	Holo Crystal Proteins		Apo ESMFold Proteins	
	%<2 ↑	Med.↓	%<2 ↑	Med.↓
Without condition	46.2	2.2	37.8	2.8
With ESM	49.0	2.0	37.8	2.7
FIGRDock	**57.2**	**1.6**	**46.6**	**2.3**

4.2.2. Impact of Apo Structure Prediction Methods on Docking Performance

Table 3. RMSD performance of FIGRDock models trained on apo structures predicted by different folding methods (AlphaFold2 vs. ESMFold), evaluated on both AlphaFold2- and ESMFold-predicted apo test sets.

Models	Apo AlphaFold2 Proteins		Apo ESMFold Proteins	
	%<2 ↑	Med.↓	%<2 ↑	Med.↓
FIGRDock(Training by AlphaFold2)	47.5	**2.1**	36.7	2.6
FIGRDock(Training by ESMFold)	**48.1**	**2.1**	**46.6**	**2.3**

Apo protein conformations can be predicted using either ESMFold or AlphaFold2, both of which generate 3D protein structures from amino acid sequences. However, prediction accuracy varies between methods, and few studies have explored how different predicted apo structures influence downstream docking performance. Here, we evaluate the robustness of our model when provided with apo structures predicted by different folding algorithms. This experiment is critical to determine whether our model's performance depends on specific conformational inputs.

To this end, we constructed two dataset variants for training and evaluation. In addition to the main experimental setup using ESMFold, we created a variant based on AlphaFold2-

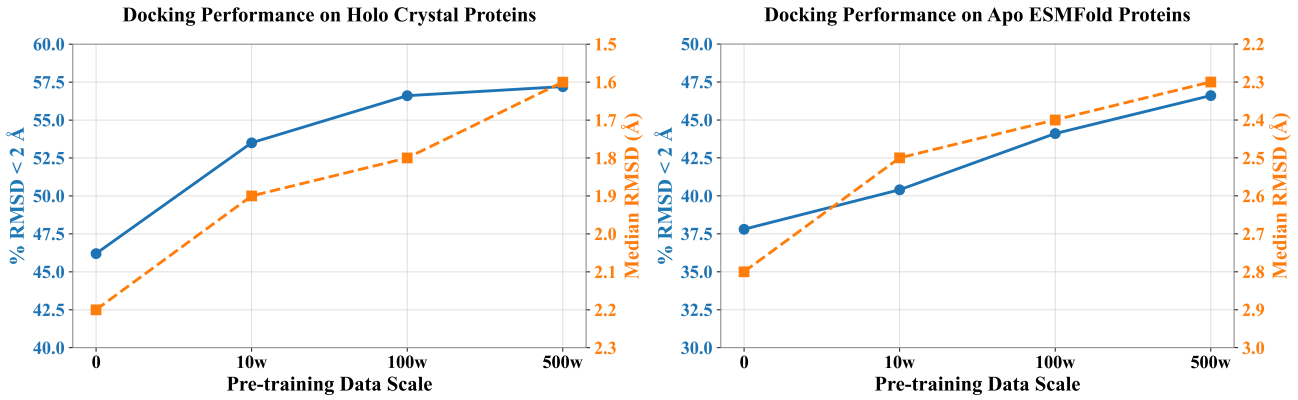

Figure 4. Comparison of docking performance with different scales of pre-training data. Larger pre-training datasets lead to better performance across both holo (left) and apo (right) settings.

predicted apo structures. The data processing and split strategy follows the same protocol as FABFlex (Zhang et al., 2025). We trained two models separately using apo structures predicted by ESMFold and AlphaFold2, and evaluated each model on both ESMFold- and AlphaFold2-predicted test sets. Results are shown in Table 3.

We observe that when the training and testing apo structures come from the same folding method, FIGRDock performs well in both cases. Interestingly, the model trained on ESMFold data generalizes well to AlphaFold2-predicted test structures. In contrast, the model trained on AlphaFold2 data performs poorly on the ESMFold-predicted test set. We hypothesize that this is due to AlphaFold2's higher prediction accuracy, which may result in less noisy apo conformations in the training set, thereby limiting the model's ability to generalize to noisier samples in the ESMFold test set.

4.2.3. Scaling Study of Pre-training Data

In this section, we investigate how the scale of pre-training data influences model performance by varying the dataset size used for conditional pre-training, ranging from 0 (as noted earlier, this corresponds to the 'Without condition' setting in Table 2, i.e., no conditional pre-training) to 5 million samples (the full SIU (Huang et al., 2024a) dataset). Figure 4 demonstrates a positive correlation between the scale of pre-training data and docking performance across all evaluation metrics, under both apo and holo test scenarios. This consistent improvement with increasing data scale validates the effectiveness and flexibility of our framework, as well as its potential to benefit from even larger datasets. Due to computational constraints, we currently report full experiments only on the 5-million-sample setting. Future work can explore larger-scale datasets to further enhance performance.

5. Conclusion

In this work, we present FIGRDock, a fast and accurate regression-based framework for flexible molecular docking. Unlike mainstream generative methods that rely on repetitive sampling, scoring, and pre-trained protein embeddings, FIGRDock adopts an interaction-aware conditional representation to guide direct regression of protein-ligand complex structures. By decoupling the learning of interaction patterns from the final docking prediction, FIGRDock achieves high docking accuracy with a single forward pass, significantly improving inference efficiency. Extensive experiments on both holo and apo settings demonstrate that FIGRDock not only outperforms previous diffusion-based models in accuracy but also achieves nearly 100× faster inference. These results highlight the promise of regression-based docking under interaction-guided supervision and open new directions for efficient and scalable structure-based drug design.

6. Acknowledgements

This work is supported by Beijing Academy of Artificial Intelligence (BAAI).

References

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

Bryant, P., Kelkar, A., Guljas, A., Clementi, C., and Noé, F. Structure prediction of protein-ligand complexes from sequence information with umol. *Nature Communications*,

15(1):4536, 2024.

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

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

Corso, G., Somnath, V. R., Getz, N., Barzilay, R., Jaakkola, T., and Krause, A. Composing unbalanced flows for flexible docking and relaxation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Gao, B., Jia, Y., Mo, Y., Ni, Y., Ma, W., Ma, Z., and Lan, Y. Profsa: Self-supervised pocket pretraining via protein fragment-surroundings alignment. *arXiv preprint arXiv:2310.07229*, 2023.

Huang, Y., Gao, B., Jia, Y., Ma, H., Ma, W.-Y., Zhang, Y.-Q., and Lan, Y. Siu: A million-scale structural small molecule-protein interaction dataset for unbiased bioactivity prediction. *arXiv preprint arXiv:2406.08961*, 2024a.

Huang, Y., Zhang, O., Wu, L., Tan, C., Lin, H., Gao, Z., Li, S., Li, S., et al. Re-dock: towards flexible and realistic molecular docking with diffusion bridge. *arXiv preprint arXiv:2402.11459*, 2024b.

Koes, D. R., Baumgartner, M. P., and Camacho, C. J. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of chemical information and modeling*, 53(8):1893–1904, 2013.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

Liu, L., Zhang, S., Xue, Y., Ye, X., Zhu, K., Li, Y., Liu, Y., Gao, J., Zhao, W., Yu, H., et al. Technical report of helixfold3 for biomolecular structure prediction. *arXiv preprint arXiv:2408.16975*, 2024.

Lu, W., Wu, Q., Zhang, J., Rao, J., Li, C., and Zheng, S. Tankbind: Trigonometry-aware neural networks for drug-protein binding structure prediction. *Advances in neural information processing systems*, 35:7236–7249, 2022.

Lu, W., Zhang, J., Huang, W., Zhang, Z., Jia, X., Wang, Z., Shi, L., Li, C., Wolynes, P. G., and Zheng, S. Dynamicbind: predicting ligand-specific protein-ligand complex structure with a deep equivariant generative model. *Nature Communications*, 15(1):1071, 2024.

McNutt, A. T., Francoeur, P., Aggarwal, R., Masuda, T., Meli, R., Ragoza, M., Sunseri, J., and Koes, D. R. Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics*, 13(1):43, 2021.

Méndez-Lucio, O., Ahmad, M., del Rio-Chanona, E. A., and Wegner, J. K. A geometric deep learning approach to predict binding conformations of bioactive molecules. *Nature Machine Intelligence*, 3(12):1033–1039, 2021.

Pei, Q., Gao, K., Wu, L., Zhu, J., Xia, Y., Xie, S., Qin, T., He, K., Liu, T.-Y., and Yan, R. Fabind: Fast and accurate protein-ligand binding. *Advances in Neural Information Processing Systems*, 36:55963–55980, 2023.

Plainer, M., Toth, M., Dobers, S., Stark, H., Corso, G., Marquet, C., and Barzilay, R. Diffdock-pocket: Diffusion for pocket-level docking with sidechain flexibility. 2023.

Qiao, Z., Nie, W., Vahdat, A., Miller III, T. F., and Anandkumar, A. State-specific protein–ligand complex structure prediction with a multiscale deep generative model. *Nature Machine Intelligence*, 6(2):195–208, 2024.

Stärk, H., Ganea, O., Pattanaik, L., Barzilay, R., and Jaakkola, T. Equibind: Geometric deep learning for drug binding structure prediction. In *International conference on machine learning*, pp. 20503–20521. PMLR, 2022.

Trott, O. and 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*, 31(2):455–461, 2010.

Verdonk, M. L., Cole, J. C., Hartshorn, M. J., Murray, C. W., and Taylor, R. D. Improved protein–ligand docking using gold. *Proteins: Structure, Function, and Bioinformatics*, 52(4):609–623, 2003.

Wang, R., Fang, X., Lu, Y., Yang, C.-Y., and Wang, S. The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119, 2005.

Zhang, Z., Wu, L., Gao, K., Yao, J., Qin, T., and Han, B. Fast and accurate blind flexible docking. *arXiv preprint arXiv:2502.14934*, 2025.

Zhou, G., Gao, Z., Ding, Q., Zheng, H., Xu, H., Wei, Z., Zhang, L., and Ke, G. Uni-mol: A universal 3d molecular representation learning framework. 2023.

Zhu, J., Gu, Z., Pei, J., and Lai, L. Diffbindfr: an se (3) equivariant network for flexible protein–ligand docking. *Chemical Science*, 15(21):7926–7942, 2024.

Table 4. Hyperparameter settings for Pocket Pretraining, Conditional Pretraining, and Fine-tuning stages.

	Pocket Pretraining	**Conditional Pretraining**	**Fine-tuning**
Batch Size	64	16	16
Training Epochs	10	12	100
Learning Rate	1×10^{-4}	3×10^{-4}	3×10^{-4}
LR Scheduler	`polynomial_decay`	`polynomial_decay`	`polynomial_decay`
Warmup Ratio	0.01	0.06	0.06
Optimizer	Adam	Adam	Adam
Weight Decay	1×10^{-4}	0	0
GPU Number	4	4	4

7. Technical Appendices and Supplementary Material

7.1. Implementation Details

The training of FIGRDock consists of three stages: Pocket encoder pre-training, conditional pre-training, and fine-tuning. The hyperparameter settings used in all stages are listed in Table 4.

For pocket encoding, to enhance the model's sensitivity to side-chain variations during the docking process, we treat atoms with the same elemental type but different structural roles, such as backbone versus side-chain atoms, as distinct atom types. In particular, for side-chain atoms, we define a comprehensive set of atom types to capture their structural specificity, including:

`C, CA, CB, CD, CD1, CD2, CE, CE1, CE2, CE3, CG, CG1, CG2, CH2, CZ, CZ2, CZ3, N, ND1, ND2, NE, NE1, NE2, NH1, NH2, NZ, O, OD1, OD2, OE1, OE2, OG, OG1, OH, SD, SE`

We employ a Transformer-based architecture to encode molecular and protein structures. Specifically, the encoding schemes for atom types and 3D positions, along with the design of the Transformer layers, are adopted from Uni-Mol(Zhou et al., 2023).

7.2. Additional Experiment Results

We conducted a comprehensive evaluation of FIGRDock on the PoseBusters V2 benchmark. As shown in Figure 5 findings revealed that FIGRDock consistently passed various validity tests in the vast majority of instances. Nevertheless, instances of clashes were detected in a limited number of cases.

7.3. Visualized Examples

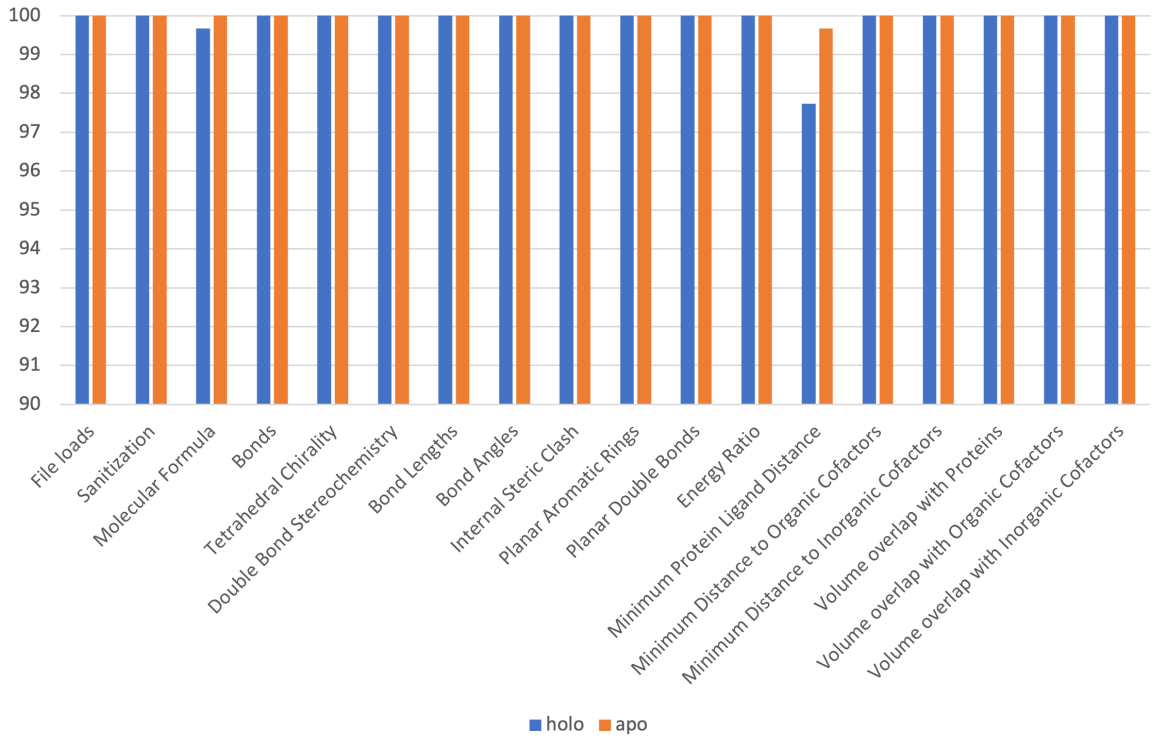

Figure 5. Detailed plausibility checks for predictions by FIGRDock on PoseBusters V2 benchmark with holo and apo input. FIGRDock achieves 99.5% and 96.7% PBValid for apo and holo input, generating physically reasonable conformations.

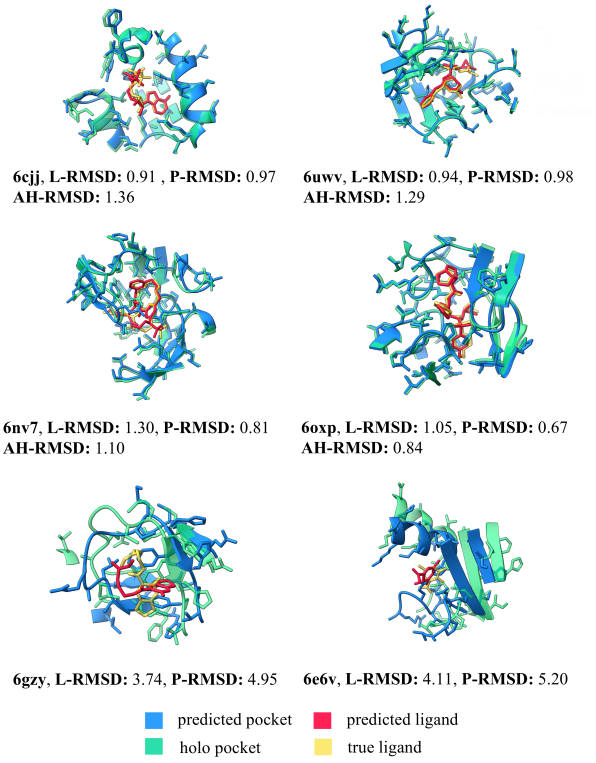

Figure 6. Visualized examples of complexes 6cjj, 6uwv, 6nv7, 6oxp, 6gzy and 6e6v in PDBbind test dataset. L-RMSD measures the RMSD between the predicted ligand and the ground-truth ligand. P-RMSD denotes the RMSD between the predicted pocket and the ground-truth holo pocket. AH-RMSD represents the RMSD between the input apo pocket and the ground-truth holo pocket.