Supplementary Materials for *"FABind: Fast and Accurate Protein-Ligand Binding Prediction"*

Anonymous Author(s)
Affiliation
Address
`email`

1 More Detailed Descriptions

1.1 Dataset Preprocessing

As we stated paper Section 4.1, PDBBind v2020 dataset [8] contains 19,443 ligand-protein complex structures, and we pre-process the structures as follows. First, we only keep complex structures whose ligand structure file (in `sdf` or `mol2` format) can be processed by RDKit [6] or TorchDrug [15], leaving 19,126 complexes. Then, to address the multiple equally valid binding ligand pose issue for symmetric receptor structures, we only keep the protein chains that have an atom within 10 Å radius of any atom of the ligand. We further filter out complexes in which the contact (distance is within 10Å) number between ligand atom and protein amino acid C_α are less than or equal to 5, or the number of ligand atom is more than or equal to 100. After applying these filters, $18,624$ complexes are left. Finally, we proceed remained complexes with the time split as described in EquiBind [12].

1.2 Experiment Settings

Baseline. We compare our model with traditional score-based docking methods and recent geometry-based deep learning methods. For traditional docking methods, QVina-W, GNINA [11], SMINA [4], GLIDE [2] and AutoDock Vina [13] are included. For deep learning methods, EquiBind [12], TankBind [10], E3Bind [14] and DiffDock [1] are included.

We report corrected results for the deep learning baselines including EquiBind, TankBind, and E3Bind. The corrected results adopt post-optimization methods on model outputs, including fast point cloud fitting (used in EquiBind) and gradient descent (used in TankBind and E3Bind), which can further enforce geometry constraints within ligand. For TankBind, the post-optimization method is used to get final ligand coordinates through the predicted distance matrix, which is essential for distance to coordinate transformation.

Training and Evaluation. The training process consists of two stages. In the initial warm-up stage, only the native pockets are used for docking. Once the pocket prediction performance reaches a certain threshold (specifically, when the center coordinate distance between the predicted center and ground truth is less than 4Å), the training progresses to the second stage. During the second stage, the predicted pockets are integrated into the docking training process. The sampling protocol involves a 25% probability of selecting the ground truth pocket and a 75% probability of selecting the predicted pocket. Note that the task of pocket prediction is consistently incorporated into the entire training process. Following E3Bind [14], we also apply normalization (divided by 5) and unnormalization (multiplied by 5) techniques on the coordinate and distance. Additionally, to improve the generalization ability of the model, the pocket is randomly shifted from -5Å to 5Å in all three axes during training. FABind models are trained for approximately 500 epochs using the AdamW [9] optimizer on 8 NVIDIA V100 16GB GPU with batch size set to 3 on each GPU. The learning rate is $5e-5$, which is scheduled to warm up in the first 15 epochs and then decay linearly. To further

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

enforce geometric constraints, we also incorporate local atomic structures (LAS) constraints in the training process by ensuring the distance between ligand atoms i and k in the transformed conformer ($\mathbf{X}$) by model are consistent with those in the initial low-energy conformer ($\mathbf{Z}$) for atoms that are either 2-hop neighbors or in the same ring structure, as proposed in EquiBind [12].

1.3 Ligand and Protein Feature Encoding

As we stated in paper Section 3.1, we construct ligand feature by TorchDrug [15] toolkit and protein feature with pre-trained ESM-2 model. Here we give a detailed description of the encoding. For ligand compound, the node embedding $\mathbf{h}_i$ is a 56-dimensional vector containing the following features: atom number; degree; the number of connected hydrogens; total valence; formal charge; whether or not it is in an aromatic ring. For protein target, we directly use the pre-trained 33-layer ESM-2 [7] model[1], which contains 650M parameters and is trained on UniRef 50M dataset. The node feature in the protein graph is derived from the amino acid feature and the hidden size is 1280.

1.4 Model Architecture Details

Edge Construction. We now introduce how to construct the edges in our FABind layers. For clarity, we use FABind in pocket prediction module for a demonstration. The cut-offs are set to the same in both pocket prediction and docking. As defined in the paper, we have three types of edges, $\mathcal{E} := \{\mathcal{E}^l, \mathcal{E}^p, \mathcal{E}^{lp}\}$, for ligand, protein, and ligand-protein interface, respectively. $\mathcal{E}^l$ and $\mathcal{E}^p$ are constructed from independent context, while $\mathcal{E}^{lp}$ is constructed from external interface. For independent context of a ligand, we directly refer to chemical bonds as constructed edges $\mathcal{E}^l$ with the biological insight that a molecule keeps its chemical bonds during the process. For independent context of a protein, $\mathcal{E}^p$ is defined as the edges connecting to nodes when the spatial distance is below a cutoff distance c_{in}. We set $c_{\text{in}} = 8.0$ in our work following Kong et al. [5]. Note that independent edges for ligands and proteins differ. For external edges $\mathcal{E}^{lp}$, we also add edges with a spatial radius threshold $c_{\text{ex}} = 10.0$ following TankBind [10].

Global Nodes. MEAN [5] demonstrates that the global nodes intensify information exchange during message passing. Therefore, we insert a global node into each component (ligand or protein) of the complex as well. A global node connects to all nodes in the same component and the other global node. The coordinates are initialized as zero tensors and can be updated during feed forward.

Iterative Refinement. Iterative structure refinement has been proved as a critical design in structure prediction task [3]. It allows the network to go deeper without adding much computational overhead. Specifically, we update coordinates during all iterations, and update hidden representations only in the last iteration. To stabilize the training process and save memories, we stop the gradients except for the last iteration. In our implementation, we also accelerate training speed by randomly sampling an iteration number less than or equal to the configuration N for each batch, while always refining N iterations during inference.

2 Apo-Structure Docking

As stated in DiffDock [1], the PDBBind benchmark primarily focuses on evaluating the ability of docking methods to dock ligand to its corresponding receptor holo-structure, which is a simplified and less realistic scenario. However, in real-world applications, docking is often performed on apo or holo-structures that are bound to different ligands. To address this limitation, DiffDock proposed a new benchmark that combines the crystal complex structures of PDBBind with protein structures predicted by ESMFold [7]. In order to validate the efficacy of our FABind in the apo-structure docking scenario, we also evaluated its performance under the same settings with DiffDock. In order to validate the efficacy of our FABind in the apo-structure docking scenario, we implement the same experimental setup as DiffDock. For 363 test samples, we firstly extract their sequences from PDBBind and use `esmfold_v1` to predict their structures, from which process we assume that the apo protein structures are obtained. One sample (PDB: 6OTT) is filtered out due to out-of-memory error. Then we align the rest of samples with PDBBind, where 12 samples are further excluded due to memory limitations, the same as DiffDock. The results in Table 1 demonstrate that FABind

---

[1] The pre-trained ESM-2 checkpoint can be found at `https://dl.fbaipublicfiles.com/fair-esm/models/esm2_t33_650M_UR50D.pt`

outperforms DiffDock, achieving an RMSD of less than 2Å on 24.9% of the complexes generated by ESMFold. This demonstrates the strong predictive capacity of FABind for apo-structure predictions.

Table 1: **PDBBind blind docking on apo proteins.** The top half contains results from traditional docking software; the bottom half contains results from recent deep learning based docking methods. The last line shows the results of our FABind. No method received further training on ESMFold generated structures.

	Apo ESMFold proteins	
	Top-1 RMSD	
Method	%<2	Med.
GNINA	2.0	22.3
SMINA	3.4	15.4
EQUIBIND	1.7	7.1
TANKBIND	10.4	5.4
P2RANK+SMINA	4.6	10.0
P2RANK+GNINA	8.6	11.2
EQUIBIND+SMINA	4.3	8.3
EQUIBIND+GNINA	10.2	8.8
DIFFDOCK (10)	21.7	5.0
DIFFDOCK (40)	20.3	5.1
FABIND	**24.9**	**4.2**

3 Study

3.1 Full Ablation

The comprehensive ablations are listed in Table 2. We can observe that each of the components contribute to the good performance. Firstly, the scheduled training strategy, when removed, leads to a slight decrease in performance for challenging cases (e.g., RMSD 75%). This indicates that the scheduled training strategy contributes positively to handling difficult scenarios. Regarding loss construction, the inclusion of the distance map loss is crucial, and solely utilizing the first term of distance map loss (i.e., the distance loss between D_{ij} and $\widetilde{D}_{ij}$ in $\mathcal{L}_{dist}$, which we call "coord loss + dist. loss (1)" in Table 2) does not yield optimal results. The architecture design also has a substantial impact. The removal of cross-attention update severely impairs the model's ability to handle favorable cases (e.g., RMSD 25%). This suggests that cross-attention update is vital for capturing important structural dependencies. Furthermore, iterative refinement is found to be indispensable for most structure prediction models, including ours. In terms of feature representation, the utilization of ESM-2 features for residues proves to be most beneficial for challenging cases. It enhances the model's capability to handle difficult scenarios effectively. Lastly, post optimization does not significantly affect the overall performance.

3.2 Iterative Refinement

In this section, we investigate the impact of iterative refinement on our model. We utilize our best-performing model and evaluate its performance using different iterations during inference. The results are reported in Table 3. From the table, we can find that (denote the number of iterations as i, $1 \leq i \leq 12$): (1) the performance improves as i increases. (2) The results tend to be stable when i increases to some extent. (3) When $i = 8$, the results are generally optimal. However, the results are similar when $4 \leq i \leq 12$. Thus, a smaller value of i can be used for better efficiency.

3.3 More Cases

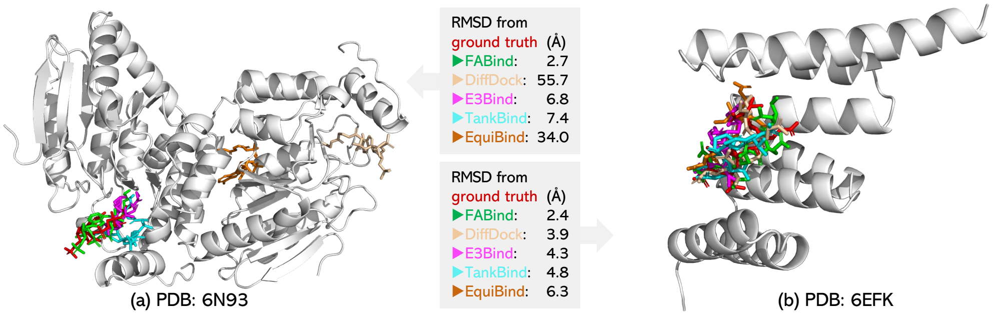
Here we show more cases on test sets to further verify the ability of FABind in finding the correct pocket for unseen protein and docking at atom level. From Fig. 1(a), in PDB 6N93, the protein is unseen in the training set, and only the predictions of FABind, E3Bind and TankBind are in the right pocket, among which FABind predict the most accurate binding pose (RMSD 2.7Å). From Fig. 1(b),

	Ligand RMSD						Centroid Distance					
	Percentiles ↓				% Below ↑		Percentiles ↓				% Below ↑	
Methods	25%	50%	75%	Mean	2Å	5Å	25%	50%	75%	Mean	2Å	5Å
FABind	1.7	3.1	6.7	6.4	33.1	64.2	0.7	1.3	3.6	4.7	60.3	80.2
No scheduled sampling	1.7	3.5	7.0	6.4	28.7	63.4	0.7	1.5	3.5	4.5	60.3	82.6
Coord loss only	2.6	4.1	7.3	6.9	16.3	60.9	1.0	1.7	4.5	4.7	53.9	77.4
Coord loss + dist. loss (1)	1.9	3.5	7.4	6.5	26.2	63.4	0.7	1.5	4.1	4.5	56.7	80.0
No cross-attention	2.2	4.2	7.0	6.4	21.4	59.9	0.9	1.9	4.1	4.5	52.3	80.4
No iterative	2.2	4.1	7.2	6.6	22.5	58.5	0.9	1.9	4.7	4.7	52.3	80.4
No ESM-2	1.9	3.6	8.0	6.3	27.5	61.2	0.8	1.6	3.8	4.4	56.2	79.1
With post optimization	1.8	3.5	7.2	6.6	30.9	61.4	0.7	1.4	3.5	4.7	59.8	80.7

Table 2: Results of full ablation study.

	Ligand RMSD						Centroid Distance						
	Percentiles ↓				% Below ↑		Percentiles ↓				% Below ↑		**Average**
Iteration	25%	50%	75%	Mean	2Å	5Å	25%	50%	75%	Mean	2Å	5Å	**Runtime (s)**
Iteration=1	3.1	4.8	7.9	7.3	5.2	51.8	1.2	2.0	5.3	5.3	49.9	74.1	0.03
Iteration=2	2.4	3.8	7.7	6.8	20.1	60.1	0.9	1.5	4.6	4.9	56.5	77.1	0.05
Iteration=4	1.8	3.3	7.2	6.5	28.4	63.6	0.7	1.4	3.8	4.8	59.5	78.8	0.07
Iteration=6	1.8	3.2	7.0	6.4	30.9	64.5	0.7	1.3	3.7	4.7	60.6	79.6	0.10
Iteration=8	1.7	3.1	6.7	6.4	33.1	64.2	0.7	1.3	3.6	4.7	60.3	80.2	0.12
Iteration=10	1.7	3.1	6.6	6.4	32.5	65.3	0.7	1.3	3.5	4.7	60.6	80.4	0.16
Iteration=12	1.7	3.1	6.6	6.4	32.5	65.6	0.7	1.4	3.6	4.7	61.7	80.4	0.19

Table 3: Inference on different iterations.



(a) PDB: 6N93

(b) PDB: 6EFK

Figure 1: Additional case studies. Pose prediction by FABind (green), DiffDock (wheat), E3Bind (magenta), TankBind (cyan), and EquiBind (orange) are placed together with protein target structure, and RMSD to ground truth (red) are reported. (a) For unseen protein (PDB 6N93), FABind, E3Bind, and TankBind successfully identifies the pocket, among which FABind predicts the most precise binding pose with lowest RMSD 2.7Å, while the other methods are all off-site. (b) For PDB 6EFK, all deep learning models find the right pocket, among which FABind predicts the most precise binding pose with lowest RMSD 2.4Å.

in PDB 6EFK, though every method correctly finds the native pocket, FABind predicts the most accurate ligand conformation (RMSD 2.4Å).

4 Broader Impacts and Limitations

Broader Impacts. Developing and maintaining the computational resources necessary to conduct AI-based molecular docking requires considerable resources, which may lead to a waste of resources.

Limitations. In FABind, we represent the protein structure at the residue level, assuming rigidity of the protein. While many existing molecular docking methods adopt a similar protein modeling strategy, we believe that employing atom-level protein modeling and incorporating protein flexibility into the modeling process could yield improved results. However, due to the scope and limitations of our current work, we have decided to defer these aspects to future research.

References

[1] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

[2] 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.

[3] 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.

[4] 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.

[5] Xiangzhe Kong, Wenbing Huang, and Yang Liu. Conditional antibody design as 3d equivariant graph translation. *arXiv preprint arXiv:2208.06073*, 2022.

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

[7] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022.

[8] Zhihai Liu, Minyi Su, Li Han, Jie Liu, Qifan Yang, Yan Li, and Renxiao Wang. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of chemical research*, 50 (2):302–309, 2017.

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[10] Wei Lu, Qifeng Wu, Jixian Zhang, Jiahua Rao, Chengtao Li, and Shuangjia Zheng. Tankbind: Trigonometry-aware neural networks for drug-protein binding structure prediction. *bioRxiv*, 2022.

[11] 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.

[12] Hannes Stärk, Octavian Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. In *International Conference on Machine Learning*, pp. 20503–20521. PMLR, 2022.

[13] 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.

[14] Yangtian Zhang, Huiyu Cai, Chence Shi, Bozitao Zhong, and Jian Tang. E3bind: An end-to-end equivariant network for protein-ligand docking. *arXiv preprint arXiv:2210.06069*, 2022.

[15] 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.