DiffDock-Pocket: Diffusion for Pocket-Level Docking with Sidechain Flexibility

Michael Plainer[1], **Marcella Toth**[1], **Simon Dobers**[1], **Hannes Stärk**[2], **Gabriele Corso**[2], **Céline Marquet**[1], **Regina Barzilay**[2]

Correspondence to {michael.plainer,marcella.toth}@tum.de
[1]School of Computation, Information and Technology, Technical University of Munich
[2]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology

Abstract

When a small molecule binds to a protein, the 3D structure of the protein and its ability and efficiency to perform its function can significantly change. Understanding this process, called *molecular docking*, can be crucial in areas such as drug design. Recent learning-based attempts have shown promising results at this task, yet lack features that traditional approaches support. In this work, we close this gap by proposing DIFFDOCK-POCKET, a diffusion-based docking algorithm that is conditioned on a binding target to predict ligand poses only in a specific binding pocket. On top of this, our model supports receptor flexibility and predicts the position of sidechains close to the binding site. Empirically, we improve the state-of-the-art in site-specific-docking on the PDBBind benchmark. Especially when using *in-silico* generated structures, we achieve more than twice the performance of current methods while being more than 20 times faster than other flexible approaches. Although the model was not trained for cross-docking to different structures, it yields competitive results in this task.

1 Introduction

Proteins are the building blocks of life and are ubiquitous in biochemical processes of all organisms. When a molecule (ligand) binds to a protein (receptor), they form a complex with a new 3D structure and function [Stank et al., 2016]. Accurately predicting these molecular interactions and structures can give insight into biological processes and is thus a highly important task in computational biology and drug discovery [Kubinyi, 2006; Meng et al., 2011; Pinzi & Rastelli, 2019].

In drug discovery campaigns, the processes underlying diseases are usually well-researched and specific targets are identified, which, if modified or inhibited, can potentially treat a disease [Weisel et al., 2009]. This means a specific part of the protein is known to be the target for docking [Zheng et al., 2012]. Site-specific docking incorporates this prior knowledge of a binding site and limits possible docking poses of ligands to a specific receptor region. Many machine-learning (ML) based approaches do not support this [Stärk et al., 2022; Lu et al., 2022; Corso et al., 2023], which is seen as one of the most significant limitations of current ML approaches [Yu et al., 2023].

Therefore, we consider the task of pocket-level docking and additionally model receptor flexibility near the binding site. When a ligand docks to a receptor, they both undergo conformational changes [Huang, 2017]. Modeling this flexibility is critical, as it can directly influence prediction accuracy [Teague, 2003; Zhao & Sanner, 2007; Hogues et al., 2018]. Still, many current methods model rigid receptors [Stärk et al., 2022; Lu et al., 2022; Corso et al., 2023], and for others adding flexibility significantly impacts the accuracy and runtime [Koes et al., 2013; McNutt et al., 2021], making them unsuitable for large-scale tasks.

Machine Learning for Structural Biology Workshop, NeurIPS 2023.

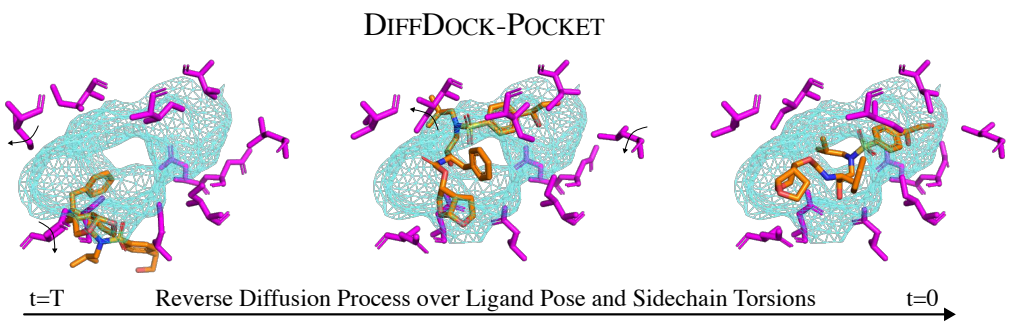

Figure 1: **Overview of our approach.** The model takes as an input a ligand, an (*in-silico* generated) protein structure, and the binding target (blue). The process starts with random ligand poses (orange) and sidechain conformations (magenta), which are improved by a reverse diffusion process (left to right) to produce conformations. The generative process modifies the translation, rotation, and torsional angles of the ligand and the torsional angles of the receptor's sidechain atoms.

This paper proposes DIFFDOCK-POCKET: a diffusion-based model for pocket-level molecular docking with receptor sidechain flexibility inspired by the ideas of DIFFDOCK [Corso et al., 2023]. It uses diffusion over a reduced product space to predict sidechain and ligand confirmations, as illustrated in Figure 1. Our model demonstrates state-of-the-art performance on the PDBBind [Liu et al., 2017] docking benchmark, where we achieve a root mean squared deviation (RMSD) of less than 2Å in 49.8% of cases compared to 27.8% achieved by the best method using receptor flexibility while being 25–90 times faster than other flexible approaches. When relying on *in-silico* generated protein structures, the model retains most of its capabilities for docking and sidechain predictions. We achieve a score of 41.7% for *in-silico* structures generated from ESMFold2 [Lin et al., 2022].

2 Related Work

Molecular docking. Docking a small molecule to a protein is governed by the energy of the interacting atoms. Using this knowledge, traditional search-based models such as GLIDE, [Friesner et al., 2004; Halgren et al., 2004], MOLDOCK [Thomsen & Christensen, 2006], and AUTODOCK [Trott & Olson, 2010] minimize a scoring function that calculates the energy of configurations. Approaches such as GNINA [McNutt et al., 2021] and DEEPDOCK [Méndez-Lucio et al., 2021] use ML to approximate this score function, while SMINA [Koes et al., 2013] takes a more classical approach. Furthermore, the search space can be limited since key binding regions are often already known. Most approaches, especially classical ones, use this information, but many ML-based approaches such as DIFFDOCK [Corso et al., 2023], EQUIBIND [Stärk et al., 2022], and TANKBIND [Lu et al., 2022] usually fail to account for binding pockets completely. Similarly, these ML-based approaches also fail to model the protein as flexible, even though already slightly different protein conformations can change the energetically optimal structure [Zhao & Sanner, 2007; Hogues et al., 2018].

Diffusion. Previous work [Corso et al., 2023] has shown that generative modeling is well-suited for docking. Score-based diffusion models [Song et al., 2021] apply the continuous diffusion process $dx = f(x,t)\,dt + g(t)\,dw$ to datapoints (in this paper, $f(x,t)$ will be 0) which has a corresponding reverse SDE $dx = [f(x,t) - g(t)^2\nabla_x \log p_t(x)]dt + g(t)\,dw$ where only the score $\nabla_x \log p_t(x)$ is unknown. Given an initial data distribution p_0, if the evolving score is learned, the reverse equation can be numerically solved to produce new points of the underlying data distribution. For molecular docking, this means that a random starting conformation of the ligand can be transformed to a good estimate of the binding complex.

3 Method

Given a ligand and a protein, flexible docking models predict the geometrical structure of the ligand and the protein. Assuming a fixed scaffold, the structure of this binding complex is uniquely described by its atom positions in the three-dimensional space. For a ligand with n atoms, and a protein with m flexible atoms, the space of possible predictions is in $\mathbb{R}^{3(m+n)}$. The large space w.r.t. the number of data points available makes docking a challenging problem.

Instead of modeling the protein and ligand atom coordinates, the conformations can also be described by the changes the ligand and the sidechains undergo during binding. The main biochemically possible changes are the rigid 3D translation or rotation of the complete ligand w.r.t. the receptor and the rotation of the torsion angles of the ligand's chemical bonds and similarly, the torsion angles of the receptor sidechains. While this does not cover all possible conformations of the protein and ligand, it accounts for the most prominent changes and keeps properties such as the rather stable bond lengths fixed. By searching in the product space, we reduce the dimensionality of the search space (see Appendix A).

3.1 Site-Specific Docking

Since docking sites are often known or chosen in advance, we can further reduce the space and speed up the search for an optimal conformation by including this prior information. With this, we can expect more accurate results while requiring less computational effort. Diffusion models build on the idea that they iteratively refine a random initial configuration. To condition the ligand pose on a binding pocket, we propose to center the ligand's initial random configuration around the pocket's center while also limiting the translations our model can predict. With this change, all ligand poses are guaranteed to be within the target pocket. Formally, the random ligand translation z_{tr} will be sampled from a normal distribution with a relatively small variance.

However, for large proteins, our approach would still need to consider atoms far away, although they do not substantially influence the binding procedure. By exploiting this fact, we decided to discard all amino acids that are too far away from the target binding site (compare Section B.1), as depicted in Figure 2. This focuses the model's attention on the binding site and reduces all proteins to a similar size. Additionally, this reduced view of the protein allows us to represent even large proteins using only a comparatively small subset of amino acids. With this, all atom positions can be used as input to the model instead of simply using the coordinates of the backbone (C-α atoms), as was done in previous work [Corso et al., 2023] which allows our model to learn more physics-informed predictions, potentially improving the accuracy.

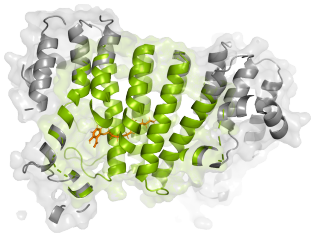

Figure 2: **Pocket reduction.** Only retain amino acids close to the ligand (green) and discard all others (gray).

3.2 Flexible Sidechains

In principle, all of the remaining amino acids can be modeled flexibly. However, flexibility of all residues would increase computational complexity without providing much benefit as it has been shown that flexibility is mostly restricted to the residues close to the binding site [Clark et al., 2019]. Therefore, we follow the convention from other docking algorithms [McNutt et al., 2021], and model only amino acids which have at least one atom within 3.5Å of any ligand atom as flexible.

Once the flexible sidechains have been selected, the rotatable bonds are determined as described in Section B.2. This way, the conformation of the sidechains can be described by the torsion angles of each rotatable bond, and the model can learn to predict the score of these angles. During inference the model produces ℓ^a ordered torsion angles $\chi_1^a, \ldots, \chi_\ell^a$ for each amino acid a. Rotating each sidechain bond of the protein by the predicted angles yields the new atom positions.

3.3 Model Architecture and Training

Models. The model architecture we are using is inspired by the structure of DIFFDOCK [Corso et al., 2023] and consists of two different models: the score model and the confidence model. The aim of the *score model* is to learn the (diffusion) scores of the tangent spaces of the transformation manifolds. With the knowledge of the scores, during inference we can take a protein with pocket and a ligand structure in 3D space and produce $i \in \mathbb{N}$ different complex structures.

The *confidence model* is then used to rank each protein-ligand prediction such that the best-predicted structures can be selected. Our confidence model learns to predict the accuracy of generated binding structures by considering both the ligand's docking success and the flexible sidechain structure accuracy. The output of the confidence model is a logit and important for real-world application since it allows practitioners to judge the accuracy of the predictions without access to the ground truth.

Architecture. Both the ligand and protein are represented as geometric graphs where nodes represent atoms and edges are between close neighbors or chemical bonds. There are edges between ligand-ligand nodes, receptor-receptor nodes, and receptor-ligand nodes. We also define a graph for each amino acid in the receptor. This representation follows multiple $SE(3)$-equivariant convolutional networks, so-called tensor field networks [Thomas et al., 2018; Geiger et al., 2022], where we make use of message passing between the nodes based on the node and edge features.

After the convolutional layers, the architecture between the score and confidence model differs. The score model needs to output a translational score, a rotational score, and one torsional score for each of the k rotatable bonds of the ligand. To allow for a flexible receptor, it also needs to predict ℓ^a torsional scores, one for each rotatable bond in each flexible amino acid a. For this, we use a pseudotorque layer as introduced by [Jing et al., 2022] similar to the architecture predicting the torsion scores of the ligand. The output of the confidence model is a single $SE(3)$-invariant scalar, which is predicted by an MLP that uses the flexible atom and ligand representations. It uses the predicted structures as input and aims to determine the probability that the docking is accurate.

Training. We use diffusion score-matching [Song et al., 2021] to train our score model by sampling the transformations from the perturbation kernels, applying them to the input structures of our model, and minimizing the theoretical denoising score matching loss function for each transformation T

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{\text{trf}\in T} \mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)} \left[\left\| \mathbf{s}^{\text{trf}}_{\boldsymbol{\theta}}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p^{\text{trf}}_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) \right\|_2^2 \right] \right\}, \quad (1)$$

as described in Song et al. [2021], with $\lambda(t)$ a positive weighting function. The minimization is done while iterating through the conditional distributions corresponding to each ligand-protein pair.

To train the confidence model, we use structures generated by our score model, which are then compared with the ground truth training data to assess their quality. The confidence model learns to predict wether this quality is above a certain threshold using binary cross-entropy loss.

4 Results

In this section, we will demonstrate that our model generalizes well beyond the data seen during training and exhibits high performance over different tasks, including docking to computationally generated structures. We will also show that our model can be used to improve the sidechain configuration of *in-silico* generated protein structures to better account for the ligand-bound structure. The code of our model and the trained weights are available at `https://github.com/plainerman/DiffDock-Pocket`.

Setup. As a training set, we relied on PDBBind [Liu et al., 2017], a subset of PDB [Berman et al., 2003], with a time-based split and a mixture of crystal and ESMFold2 generated structures. In this section, we evaluate it on the unseen testset. Similar studies for evaluating structures generated by ColabFold [Mirdita et al., 2022], a faster version of AlphaFold2 [Jumper et al., 2021], can be found in Appendix F. However, although the model has never seen ColabFold structures during training, the performance is similar to ESMFold structures. Evaluation of our model on the CrossDocked 2020 dataset [Francoeur et al., 2020] is found in Appendix H.

Metrics. To assess docking predictions, we employ root mean squared deviation (RMSD) to measure the difference between predicted ligand poses and ground truth atom positions. A pose prediction with an *RMSD below* 2Å is considered to be approximately correct [Alhossary et al., 2015; Hassan et al., 2017; McNutt et al., 2021], so we calculate the percentage of predictions under this threshold. To evaluate the predictions of the sidechain atoms, we rely on a similar metric, the RMSD of the sidechain atoms to the ground truth holo crystal structure (or SC-RMSD). As the position of the sidechains shows less variation, we decided to use an SC-RMSD threshold of 1 for the main comparisons (see Appendix G). All values reported are measured against the holo crystal structure.

Docking performance. We are comparing our model to the freely available state-of-the-art search-based methods GNINA and SMINA, as well as the diffusion-based model DIFFDOCK (which performs blind docking). Results are shown in Table 1. Our model is evaluated for drawing 10 and 40 samples, where we present metrics for the top-1 prediction, as well as for the top-5 predictions, which involves selecting the most accurate pose from the one or five highest-ranked predictions.

Table 1: **PDBBind docking performance.** This table compares the performance of different docking methods on computationally generated structures and crystal structures. Methods that do not model the receptor as flexible, have been marked with the keyword rigid. All methods other than DIFFDOCK use site-specific docking and use the same pocket definition (i.e., the mean of C-α atoms within 5Å of any ligand atom). For a more detailed explanation, see Appendix D. The numbers for the methods highlighted with a * were taken from Corso et al. [2023].

	Apo ESMFold Proteins				Holo Crystal Proteins				
	Top-1 RMSD		Top-5 RMSD		Top-1 RMSD		Top-5 RMSD		Average
Method	%<2	Med.	%<2	Med.	%<2	Med.	%<2	Med.	Runtime (s)
DIFFDOCK (blind, rigid)*	20.3	5.1	31.3	3.3	38.2	3.3	44.7	2.4	40
SMINA (rigid)	6.6	7.7	15.7	5.6	32.5	4.5	46.4	2.2	258
SMINA	3.6	7.3	13.0	4.8	19.8	5.4	34.0	3.1	1914
GNINA (rigid)	9.7	7.5	19.1	5.2	42.7	2.5	55.3	1.8	260
GNINA	6.6	7.2	12.1	5.0	27.8	4.6	41.7	2.7	1575
DIFFDOCK-POCKET (10)	41.0	**2.6**	47.6	2.2	47.7	2.1	56.3	1.8	**17**
DIFFDOCK-POCKET (40)	**41.7**	**2.6**	**47.8**	**2.1**	**49.8**	**2.0**	**59.3**	**1.7**	61

Our approach outperforms both search-based methods and DIFFDOCK in all instances, even when only drawing 10 samples. For bound protein docking with predicting 40 samples, we achieve an approximately correct docking pose in 49.8% of instances. In rigid docking, GNINA achieves a close 42.7%, but no other compared method with flexibility is competitive (27.8%). We can see that current methods suffer from a substantial loss in docking accuracy when introducing flexibility while also requiring significantly more time to predict poses (and sidechains). Furthermore, when docking to computationally generated structures, we achieve four times higher results as the best search-based method GNINA and nearly double the previous state-of-the-art DIFFDOCK.

Sidechain prediction quality. All flexible methods investigated predict the sidechain positions jointly with the ligand pose. Both SMINA and GNINA fail to predict accurate sidechains, as shown in Table 2. Our approach achieves good sidechain reconstruction in 33.3% and 49.2% of cases for computationally generated structures and crystal structures respectively.

Table 2: **PDBBind sidechain performance.** Comparing the predicted sidechains of the different models with different inputs to the ground truth crystal structures.

	Apo ESMFold Proteins				Holo Crystal Proteins			
	Top-1 SC-RMSD		Top-5 SC-RMSD		Top-1 SC-RMSD		Top-5 SC-RMSD	
Method	%<1	Med.	%<1	Med.	%<1	Med.	%<1	Med.
SMINA	0.6	2.4	1.8	2.0	4.7	1.8	8.3	1.4
GNINA	0.6	2.5	1.8	2.0	3.3	1.7	7.7	1.4
DIFFDOCK-POCKET (10)	**33.3**	**1.2**	**44.6**	**1.1**	**49.2**	**1.0**	58.6	**0.9**
DIFFDOCK-POCKET (40)	32.6	**1.2**	44.4	**1.1**	48.7	**1.0**	**59.2**	**0.9**

5 Conclusion

In this paper, we presented DIFFDOCK-POCKET, a fast diffusion-based generative model to dock small molecules. In contrast to many other ML-based approaches, we can incorporate prior knowledge of the binding pocket and model the protein as semi-flexible. Our approach improves the state-of-the-art in almost all tested instances while also being significantly faster. Traditional approaches exhibit a drastic decline in runtime and accuracy when modeling receptor flexibility, which is not the case for our approach. A similar trend can be observed when using computationally generated structures, with which our approach works exceptionally well and loses almost no accuracy. Especially in combination with *in-silico* generated structures, which can be generated quickly, we believe that our model opens new capabilities in high-throughput tasks, such as drug screening.

References

Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics*, 31(13):2214–2216, 02 2015.

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

Michael Brocidiacono, Paul Francoeur, Rishal Aggarwal, Konstantin Popov, David Koes, and Alexander Tropsha. BigBind: Learning from nonstructural data for structure-based virtual screening, November 2022.

Michael Brocidiacono, Konstantin I. Popov, David Ryan Koes, and Alexander Tropsha. Plantain: Diffusion-inspired pose score minimization for fast and accurate molecular docking. In *Workshop on Computational Biology*, 2023.

Martin Buttenschoen, Garrett M. Morris, and Charlotte M. Deane. PoseBusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences, 2023.

Jordan J. Clark, Mark L. Benson, Richard D. Smith, and Heather A. Carlson. Inherent versus induced protein flexibility: Comparisons within and between apo and holo structures. *PLOS Computational Biology*, 15(1):e1006705, January 2019.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. DiffDock: Diffusion steps, twists, and turns for molecular docking. In *International Conference on Learning Representations*, 2023.

Paul G. Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B. Iovanisci, Ian Snyder, and David R. Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of Chemical Information and Modeling*, 60(9): 4200–4215, August 2020.

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, David E. Shaw, Perry Francis, and Peter S. Shenkin. 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, February 2004.

Mario Geiger, Tess Smidt, Alby M., Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice, Kostiantyn Lapchevskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, Dylan Madisetti, Martin Uhrin, Jes Frellsen, Nuri Jung, Sophia Sanborn, Mingjian Wen, Josh Rackers, Marcel Rød, and Michael Bailey. Euclidean neural networks: e3nn, April 2022.

Thomas A Halgren, Robert B Murphy, Richard A Friesner, Hege S Beard, Leah L Frye, W Thomas Pollard, and Jay L Banks. Glide: a new approach for rapid, accurate docking and scoring. 2. enrichment factors in database screening. *Journal of medicinal chemistry*, 2004.

Nafisa M. Hassan, Amr A. Alhossary, Yuguang Mu, and Chee-Keong Kwoh. Protein-ligand blind docking using quickvina-w with inter-process spatio-temporal integration. *Scientific Reports*, 7 (1):15451, Nov 2017.

Hervé Hogues, Francis Gaudreault, Christopher R. Corbeil, Christophe Deprez, Traian Sulea, and Enrico O. Purisima. ProPOSE: Direct exhaustive protein–protein docking with side chain flexibility. *Journal of Chemical Theory and Computation*, 14(9):4938–4947, August 2018.

Sheng-You Huang. Comprehensive assessment of flexible-ligand docking algorithms: current effectiveness and challenges. *Briefings in Bioinformatics*, 19(5):982–994, March 2017.

John Ingraham, Max Baranov, Zak Costello, Vincent Frappier, Ahmed Ismail, Shan Tie, Wujie Wang, Vincent Xue, Fritz Obermeyer, Andrew Beam, and Gevorg Grigoryan. Illuminating protein space with a programmable generative model, 2022.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24240–24253. Curran Associates, Inc., 2022.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, July 2021.

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.

Dániel Kozma, István Simon, and Gábor E. Tusnády. Pdbtm: Protein data bank of transmembrane proteins after 8 years. *Nucleic Acids Research*, 41(D1):D524–D529, November 2012.

Hugo Kubinyi. *Computer Applications in Pharmaceutical Research and Development*. Wiley, June 2006.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, 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.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

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, February 2017.

Mikhail A. Lomize, Irina D. Pogozheva, Hyeon Joo, Henry I. Mosberg, and Andrei L. Lomize. Opm database and ppm web server: resources for positioning of proteins in membranes. *Nucleic Acids Research*, 40(D1):D370–D376, September 2011.

Wei Lu, Qifeng Wu, Jixian Zhang, Jiahua Rao, Chengtao Li, and Shuangjia Zheng. TANKBind: Trigonometry-aware neural networks for drug-protein binding structure prediction. In *Advances in Neural Information Processing Systems*, 2022.

Manjeera Mantina, Adam C. Chamberlin, Rosendo Valero, Christopher J. Cramer, and Donald G. Truhlar. Consistent van der waals radii for the whole main group. *The Journal of Physical Chemistry A*, 113(19):5806–5812, April 2009.

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.

Rocco Meli, Andrew Anighoro, Mike J Bodkin, Garrett M Morris, and Philip C Biggin. Learning protein-ligand binding affinity with atomic environment vectors. *Journal of Cheminformatics*, 13 (1):59, August 2021.

Oscar Méndez-Lucio, Mazen Ahmad, Ehecatl Antonio del Rio-Chanona, and Jörg Kurt Wegner. A geometric deep learning approach to predict binding conformations of bioactive molecules. *Nature Machine Intelligence*, 3(12):1033–1039, 2021.

Xuan-Yu Meng, Hong-Xing Zhang, Mihaly Mezei, and Meng Cui. Molecular docking: A powerful approach for structure-based drug discovery. *Current Computer Aided-Drug Design*, 7(2):146–157, June 2011.

Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. ColabFold: making protein folding accessible to all. *Nature Methods*, 19(6):679–682, May 2022.

Thomas D Newport, Mark S P Sansom, and Phillip J Stansfeld. The memprotmd database: a resource for membrane-embedded protein structures and their lipid interactions. *Nucleic Acids Research*, 47(D1):D390–D397, November 2018.

John P. Overington, Bissan Al-Lazikani, and Andrew L. Hopkins. How many drug targets are there? *Nature Reviews Drug Discovery*, 5(12):993–996, December 2006.

Luca Pinzi and Giulio Rastelli. Molecular docking: Shifting paradigms in drug discovery. *International Journal of Molecular Sciences*, 20(18):4331, September 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Antonia Stank, Daria B. Kokh, Jonathan C. Fuller, and Rebecca C. Wade. Protein binding pocket dynamics. *Accounts of Chemical Research*, 49(5):809–815, April 2016.

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.

Simon J. Teague. Implications of protein flexibility for drug discovery. *Nature Reviews Drug Discovery*, 2(7):527–541, July 2003.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds, 2018.

René Thomsen and Mikael H Christensen. MolDock: a new technique for high-accuracy molecular docking. *Journal of medicinal chemistry*, 49(11):3315–3321, 2006.

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.

Martin Weisel, Ewgenij Proschak, Jan M. Kriegl, and Gisbert Schneider. Form follows function: Shape analysis of protein cavities for receptor-based drug design. *PROTEOMICS*, 9(2):451–459, January 2009.

Stephen H. White. Biophysical dissection of membrane proteins. *Nature*, 459(7245):344–346, May 2009.

Yuejiang Yu, Shuqi Lu, Zhifeng Gao, Hang Zheng, and Guolin Ke. Do deep learning models really outperform traditional approaches in molecular docking? In *Workshop on Machine Learning for Drug Discovery*, 2023.

Yong Zhao and Michel F. Sanner. Protein-ligand docking with multiple flexible side chains. *Journal of Computer-Aided Molecular Design*, 22(9):673–679, November 2007.

Xiliang Zheng, LinFeng Gan, Erkang Wang, and Jin Wang. Pocket-based drug design: Exploring pocket space. *The AAPS Journal*, 15(1):228–241, November 2012.

A Bound on Reduced Prediction Space

As mentioned in the main text, our model makes predictions in a reduced, lower-dimensional space instead of predicting all atom positions. These transformations form an algebraic group structure and span a $3 + 3 + k + \ell$ dimensional manifold, which we refer to as the *product space*. k, ℓ are the number of torsion angles in the ligand and protein, respectively. We can assess the reduction by counting the degrees of freedom of translations on the ligand and flexible sidechains as a function of their number of atoms. Sidechains have $m - r$ degrees of freedom for m atoms on r residues, since each residue has $m_r - 1$ torsion angles (where m_r is the number of atoms in one residue). Since the maximum number of torsional angles in an amino acid (counted by our algorithm) is five, we can further bound $m - r$ with $0.8m$. Similarly, we can bound the ligand degrees of freedom by $n - 2 + 6$, 6 for the freedom of rotations and translations, and $n - 2$ the degrees of freedom from the torsion angles. This is because we can use an upper bound by assuming a tree-like bond structure between the ligand atoms, which means $n - 1$ bonds for n atoms and, therefore $n - 2$ degrees of freedom (in case there is a cycle the ligand graph would have one more bond but it would also lose a degree of freedom from the restriction of the cycle structure). We can then compare the dimensions of $0.8m + n + 4$ to $3(m + n)$ and conclude that the three-dimensional coordinate space clearly has magnitudes larger (about three times as many) degrees of freedom, already for molecules with a small number of atoms.

B Model Details

B.1 Pocket Definition

We require knowledge of the pocket center position in $\mathbb{R}^3$ and a radius indicating the pocket's size to center the translational noise and reduce the protein. As for the pocket size, we use the radius of the smallest sphere centered at the mean of the ligand that can fit all atoms. We then also add an additional buffer of 10Å to the radius to retain the surrounding context of the pocket for the model to make predictions. If any atom of an amino acid falls within this distance from the pocket center, the whole amino acid is kept, whereas all other amino acids are discarded. Defining the pocket center can be a bit more challenging because, in practice, one might be able to infer the general area where a ligand might dock but cannot pinpoint the exact center of the ligand. To avoid bias in the training data, we calculate the pocket center by taking the average positions of the C-α atoms within 5Å of any ground truth ligand atom. This technique aligns with a setting where one would visually analyze the protein and suspect the pocket location. By only using the rigid backbone to calculate the center, this definition of a pocket works well, even when the protein has a different sidechain structure.

B.2 Sidechain Flexibility

The flexible residues can be automatically determined based on the distance to the ground truth ligand pose or, at inference, manually specified when there is no access to a ground truth ligand. We then select residues with atoms inside a rectangular prism around the ligand as also used in previous works [McNutt et al., 2021]. This means that with a "radius" of r every residue is selected where for the coordinates x, y, z any atom of this amino acids it holds that

$$\begin{aligned} \min(lig_x) - r < x < \max(lig_x) + r \\ \min(lig_y) - r < y < \max(lig_y) + r \\ \min(lig_z) - r < z < \max(lig_z) + r, \end{aligned} \tag{2}$$

where lig_x, lig_y and lig_z mean the collection of x, y and z coordinates of the ligand atoms. This defines a prism around the ligand with an additional radius r. For a flexible radius, we chose 3.5Å as modeling flexibility for sidechains within this radius to the ligand was found to be a reasonable representation for structural changes happening upon ligand binding in Meli et al. [2021]. During inference, we cannot assume to have any information regarding the ligand position therefore instead of calculating a prism around the ligand, the user needs to set them manually.

To determine the concrete bonds at which torsional angles need to be applied, we build a graph for each amino acid according to the chemical structure. Each found rotatable bond is stored as the corresponding edge and subgraph that starts at the second vertex/end of the edge, onto which a rotation would be applied. See Algorithm 1 for the implementation.

Corso et al. [2023] had to rely on the definition of preservation of angular velocity and Kabsch alignments to disentangle the effect of the updates in torsion angles of the ligands from the roto-translation of the ligand w.r.t. the protein. In our case, we keep this convention for the disentanglement of the degrees of freedom of the ligand. When it comes to defining the direction of update of the torsion angles of the sidechains of the protein, we always rotate the side that does not contain the protein backbone. This simple convention makes the update of the sidechain's conformation disentangled from the roto-translation of the ligand w.r.t. the protein without requiring any additional Kabsch alignment. We note that in practice this is very similar to the induction of no linear or angular velocity in the protein due to the significantly larger size of the rest of the protein compared to the individual sidechain.

Algorithm 1: Graph Traversal to Compute Rotatable Bonds

Input: Atom positions x, atom names $\mathcal{N}$
Output: Rotable bonds $\mathcal{B}$, rotation mask $\mathcal{M}$
$(x, \mathcal{N}) \leftarrow \texttt{removeHydrogens}(x, \mathcal{N})$;
$G \leftarrow \texttt{constructDirectedGraph}(x, \mathcal{N})$;
for $e \in edges(\texttt{BFS}(G))$ **do**
  $G_U \leftarrow \texttt{toUndirected}(G)$;
  $G_U \leftarrow \texttt{removeEdge}(G_U, e)$;
  if *not* $\texttt{isConnected}(G_U)$ **then**
    $c \leftarrow \texttt{connectedComponents}(G_U)$;
    if $\texttt{size}(sorted(c)[0]) > 1$ **then**
      $\mathcal{M}.append(c[1])$;
      $\mathcal{B}.append(e)$;
    end
  end
end

B.3 Sidechain Conformer Matching

When learning the torsional angles with our proposed diffusion approach, we need access to a protein where the atoms of the flexible sidechains are bound to the ligand. Holo crystal structures already contain this correct information. A problem arises when we want to rely on different data for our method, such as either computationally generated structures or proteins that are bound to different ligands. In these cases, the torsional bond angles and bond lengths will be different from the ground truth data. This shift can be attributed to other (non-prominent) conformational changes the protein undergoes (e.g., the lengthening or shortening of bonds) or to inaccuracies of predictive models when using synthetic data (i.e. ESMFold is not perfect). Hence, such protein structures cannot be used directly for training.

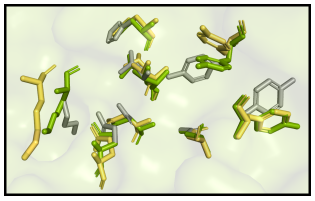

Figure 3: **Sidechain conformer matching.** Optimize the sidechain torsional angles (green) of the computationally generated structure (gray) to minimize the distance to the ground truth positions (yellow).

To still be able to expose the model to different structures, we prepared computationally generated structures with a procedure referred to as *sidechain conformer matching*. The idea is to align the torsional angles of the computationally generated structures to the ground truth ligand-bound crystal structures while keeping the rigidity of the bonds, as can be seen in Figure 3. This allows us to use computationally generated structures for the training procedure while having sidechain positions that are close to the ground truth. At inference time no sidechain conformer matching will be performed, as this could leak test data.

Similarly to Jing et al. [2022], we define the search for these structures as a minimization problem of the RMSD between the ground truth structure $\boldsymbol{y}$ and *in-silico* structure $\boldsymbol{y}'$ over the torsional angles of the flexible amino acids. When referring to the ligand as $\boldsymbol{x}$ and assuming we have a sidechain for amino acid a with ℓ^a rotatable bonds $\chi_1^a, \ldots, \chi_\ell^a$ the goal can be phrased as ℓ minimization problems

for each amino acid

$$\text{match}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{y}') = \underset{\tilde{\boldsymbol{y}} \in \{\text{apply}(\boldsymbol{y}', \boldsymbol{\chi})\}}{\arg\min} \text{RMSD}(\boldsymbol{y}, \tilde{\boldsymbol{y}}) \cdot \text{penalty}(\boldsymbol{x}, \tilde{\boldsymbol{y}}). \tag{3}$$

The additional penalty in the optimization goal was introduced to make the matched proteins more realistic. It aims to reduce the number of steric clashes (i.e., atoms that would be too close together), and is described in more detail in Appendix C. The minimization is solved with differential evolution, which iteratively combines potential solutions of a population to converge to the global minimum. We can then use the computationally generated structure where the sidechains have been conformer-matched with the bound structure in training. This matching still leaves some distance between the structures (as seen in Figure 3) but aligns with our definition of a semi-flexible receptor.

B.4 Architecture

The protein and the ligand structures can be represented as geometric graphs. Our architecture uses three different graphs: a graph containing the ligand atoms, one that contains the protein atoms, and a third where each node corresponds to a residue (i.e., an amino acid). The atom nodes of the ligands and proteins are featurized with their chemical properties, the residue nodes with embeddings of the ESM2 language model [Lin et al., 2023].

The nodes in each graph are connected to nodes in the same graph with inter-graph edges. We construct receptor-receptor and residue-residue edges between an atom and its k nearest neighbors (for residues we use the C-α positions). The ligand-ligand edges correspond to bonds between the ligand atoms that are featurized by their bonding type, and additionally, we form edges between atoms under a cutoff distance of 5Å.

Nodes can also be connected to nodes in the other graphs by (dynamic) cross edges. For the ligand-receptor and ligand-atom edges, we form edges between atoms based on a distance threshold that is calculated with the diffusion noise. As for the atom-residue graphs, we connect each residue to the atoms it consists of. As the positions of the ligand and receptor atoms are dynamic in the diffusion process, these graphs need to be reconstructed at each time step.

Several convolutional layers are concatenated in which the nodes pass messages using tensor products based on the node features and irreducible representations of the edges. The number of convolutional layers differs between the score and confidence model. MLPs are then used on the node embeddings to make the final predictions.

B.5 Training the Confidence Model

To train the confidence model, we trained a smaller score model (in the same way as the main/large model) that predicts more diverse but less accurate ligand poses and protein structures. The predictions are then evaluated against the ground truth to create a label that indicates whether the RMSD is < 2Å and the RMSD of the flexible atoms in the sidechains is < 1Å. The confidence model then learns to predict a label of 1 iff the prediction of the score model is good in terms of docking and sidechain atom positions. The model is then trained with a binary cross-entropy loss. No diffusion is involved in the training of the confidence model.

B.6 Training and Inference of the Score Model

We use ESMFold2 predicted structures conformer-matched to the PDBBind crystal structures to train the score model. If the RMSD in the pocket between the ground truth and *in-silico* structure is larger than 2Å, we assume that ESMFold was unable to predict a good structure and use the ground truth holo structure instead. The training and inference procedures were inspired by DIFFDOCK and can be seen in Algorithm 2 and Algorithm 3 respectively.

At inference (i.e., Algorithm 3), it is important to note that the model is not aware of any of the ground-truth ligand or sidechain positions. As such, there is no possibility for data leakage as the model is neither aware of the ground-truth sidechain positions, nor which sidechains are flexible.

Algorithm 2: Training Epoch

Input: Training pairs: $\{(\mathbf{x}^\star, \mathbf{y}^\star), \}$, flexibility radius: r, pocket radius: p with buffer
foreach $\mathbf{x}^\star, \mathbf{y}^\star$ **do**
  Let $\mathbf{x}_0 \leftarrow \arg\min_{\mathbf{x}^\dagger \in \mathcal{M}_{tr,rot,tor,\mathbf{x}^\star}} \mathrm{RMSD}(\mathbf{x}^\star, \mathbf{x}^\dagger)$;
  Let $\tilde{\mathbf{y}}^\star \leftarrow \{res \in \mathbf{y}^\star : \exists \text{ atom} = (a_x, a_y, a_z) \in res, a_x \in [\min_x(\mathbf{x}^\star) - \mathbf{r}, \max_x(\mathbf{x}^\star) + \mathbf{r}], a_y \in [\min_y(\mathbf{x}^\star) - \mathbf{r}, \max_y(\mathbf{x}^\star) + \mathbf{r}], a_z \in [\min_z(\mathbf{x}^\star) - \mathbf{r}, \max_z(\mathbf{x}^\star) + \mathbf{r}]\}$;
  Let $\mathbf{y}_0^\star \leftarrow \arg\min_{\mathbf{y}^\dagger \in \mathcal{M}_{sc-tor,\mathbf{y}^\star}} \mathrm{RMSD}(\tilde{\mathbf{y}}^\star, \tilde{\mathbf{y}}^\dagger) \cdot \text{penalty}$;
  Let pocket center = $pc \leftarrow$ average of positions of $C_\alpha \in \{$residue $\in \mathbf{y}^\star$ $\exists$atom $= a \in$ residue for which $\exists$ ligand atom $l \in \mathbf{x}_0 \| a - l \| < p\}$ if the set is empty, then closest C_α;
  Let $\mathbf{y}_0 \leftarrow \{$res $\in \mathbf{y}_0^\star : \exists a \in$ res for which $\exists l \in \mathbf{x}_0 : \|a - l\| <$ circumradius$(\mathbf{y}_0^\star) +$ buffer$\}$;
  Sample $t \sim \mathcal{U}([0, 1])$;
  Sample $\Delta\mathbf{r}, \Delta R, \Delta\boldsymbol{\theta}_l, \Delta\boldsymbol{\theta}_{sc}$, from diffusion kernels $p_t^{\text{tr}}(\cdot \mid 0), p_t^{\text{rot}}(\cdot \mid 0), p_t^{\text{tor}_l}(\cdot \mid 0), p_t^{\text{tor}_{sc}}(\cdot \mid 0)$;
  Compute $\mathbf{x}_t$ by applying $\Delta\mathbf{r}, \Delta R, \Delta\boldsymbol{\theta}_{\boldsymbol{l}}$ to $\mathbf{x}_0$;
  Compute $\mathbf{y}_t$ by applying $\boldsymbol{\theta}_{\boldsymbol{sc}}$ to $\tilde{\mathbf{y}}_0$;
  Predict scores $\alpha \in \mathbb{R}^3, \beta \in \mathbb{R}^3, \gamma \in \mathbb{R}^n, \delta \in \mathbb{R}^m = \mathbf{s}(\mathbf{x}_t, \mathbf{y}_t, t)$;
  Take optimization step on loss $\mathcal{L} = ||\alpha - \nabla \log p_t^{\text{tr}}(\Delta\mathbf{r} \mid 0)||^2 + ||\beta - \nabla \log p_t^{\text{rot}}(\Delta R \mid 0)||^2 + \left|\left|\gamma - \nabla \log p_t^{\text{tor}_l}(\Delta\boldsymbol{\theta}_{\boldsymbol{l}} \mid 0)\right|\right|^2 + \left|\left|\delta - \nabla \log p_t^{\text{tor}_{sc}}(\Delta\boldsymbol{\theta}_{\boldsymbol{sc}} \mid 0)\right|\right|^2$
end

Algorithm 3: Inference Algorithm

Input: RDKit prediction $\mathbf{c}$, generated protein structure $\mathbf{d}$, flxibility radius r, pocket radius p with buffer (both centered at origin)
Output: Sampled ligand pose $\mathbf{x}_0$, sampled protein pose $\mathbf{y}_0$ with applied pocket knowledge
Let pocket center = $pc \leftarrow$ average of positions of $C_\alpha \in \{$residue $\in \mathbf{d}$ $\exists$atom $= a \in$ residue for which $\exists$ ligand atom $l \in \mathbf{c} \| a - l \| < p\}$;
Let $\mathbf{d}^\star \leftarrow \{res \in \mathbf{d} : \exists a \in res, \|a - pc\| <$ circumradius$(\mathbf{c}) +$ buffer$\}$;
Sample $\boldsymbol{\theta}_{l;N} \sim \mathcal{U}(SO(2)^k)$, $R_N \sim \mathcal{U}(SO(3))$, $\mathbf{r}_N \sim \mathcal{N}(0, \sigma_{\text{tr}}^2(T))$ $\boldsymbol{\theta}_{sc,N} \sim \mathcal{U}(SO(2)^m)$;
Define $\tilde{\mathbf{y}_k}$ from $\mathbf{y}_k$ as $\{$residue $= res \in \mathbf{y}_k : \exists$atom $= a \in res, \|a - pc\| <$ r$\}$;
Randomize ligand and sidechains by applying $\mathbf{r}_N, R_N, \boldsymbol{\theta}_{l;N},$ to $\mathbf{c}$ and $\boldsymbol{\theta}_{sc;N}$ to $\tilde{\mathbf{d}}^\star$;
for $n \leftarrow N$ **to** 1 **do**
  Let $t = n/N$ and $\Delta\sigma_{\text{tr}}^2 = \sigma_{\text{tr}}^2(n/N) - \sigma_{\text{tr}}^2((n-1)/N)$ and similarly for $\Delta\sigma_{\text{rot}}^2, \Delta\sigma_{\text{tor}_l}^2, \Delta\sigma_{\text{tor}_{sc}}^2$;
  Predict scores $\alpha \in \mathbb{R}^3, \beta \in \mathbb{R}^3, \gamma \in \mathbb{R}^k, \delta \in \mathbb{R}^m, \leftarrow \mathbf{s}(\mathbf{x}_n, \mathbf{y}_n, t)$;
  Sample $\mathbf{z}_{\text{tr}}, \mathbf{z}_{\text{rot}}, \mathbf{z}_{\text{tor}_l}, \mathbf{z}_{\text{tor}_{sc}}$ from $\mathcal{N}(0, \Delta\sigma_{\text{tr}}^2), \mathcal{N}(0, \Delta\sigma_{\text{rot}}^2), \mathcal{N}(0, \Delta\sigma_{\text{tor}_l}^2), \mathcal{N}(0, \Delta\sigma_{\text{tor}_{sc}}^2)$ respectively;
  Set $\Delta\mathbf{r} \leftarrow \Delta\sigma_{\text{tr}}^2\alpha + \mathbf{z}_{\text{tr}}$ and similarly for $\Delta R, \Delta\boldsymbol{\theta}_{\boldsymbol{l}}, \Delta\boldsymbol{\theta}_{\boldsymbol{sc}}$;
  Compute $\mathbf{x}_{n-1}$ by applying $\Delta\mathbf{r}, \Delta R, \Delta\boldsymbol{\theta}_{\boldsymbol{l}},$ to $\mathbf{x}_n$;
  Compute $\mathbf{y}_{n-1}$ by applying $\Delta\boldsymbol{\theta}_{\boldsymbol{sc}},$ to $\tilde{\mathbf{y}}_n$;
end
Return $\mathbf{x}_0, \mathbf{y}_0$;

B.7 Low-Temperature Sampling

Due to the maximum likelihood training, the predictions of the score model can be dispersed over multiple modes of the target distribution. We perform low-temperature sampling to prevent this problem of overdispersion at inference due to model uncertainty and thereby emphasize the modes of the distribution. This is done via the approach proposed by Ingraham et al. [2022, Apx. B]. For this, we have determined the temperature values for our score model that maximize its performance on the validation set.

C Steric Clashes

Steric clashes play a fundamental role in molecular interactions and structural biology. These clashes occur when atoms, or groups of atoms, come too close to each other, resulting in repulsive forces that hinder their ability to adopt certain conformations. In the context of generative modeling of complex structures, these clashes occur when atoms or groups of atoms in a three-dimensional structure are placed too closely together, violating the principles of molecular geometry and leading to unfavorable interactions. In essence, steric clashes represent a clash of physical space, as atoms cannot occupy the same space simultaneously due to their electron clouds. Understanding and mitigating steric clashes are important to check in generative modeling because they can lead to the generation of incorrect or physically unrealistic structures.

To quantify and evaluate steric clashes, several computational methods have been developed. One common approach involves computing the overlap between van der Waals radii of atoms. The van der Waals radii represent the approximate size of atoms and are typically defined as the distance at which the attractive van der Waals forces balance the repulsive forces between two atoms. To detect steric clashes, we assessed whether the van der Waals radii of atoms or groups of atoms in a molecular structure overlap by at least 0.4 Angstroms (Å). If the overlap exceeds this threshold, it indicates a steric clash, suggesting that the molecular conformation is unfavorable due to repulsive forces. For the concrete values, we followed the tables from Mantina et al. [2009].

C.1 Reducing Steric Clashes in Protein Sidechain Alignment

To train our flexible model, we align the sidechains of the unbound (apo) ESMFold protein with the bound (holo) crystal structure with conformer matching. Especially in cases where the predicted atomic structure differs from the actual true structure, simply reducing the RMSD between those two structures might lead to unrealistic proteins. For example, there could be a lot of steric clashes or the sidechain atoms completely turned away from the pocket. We introduced an additional penalty term when aligning the two protein structures to overcome these issues. The term that produced the most reasonable outputs (with regard to the number of steric clashes) was

$$\mathrm{RMSD}(\text{Crystal Sc}, \tilde{\text{Sc}}) \cdot \frac{\sqrt{\sum_{l \in Lig, s \in Sc} e^{-(s-l)^2}}}{\sqrt{\sum_{l \in Lig, s \in Sc} e^{-(s-l)^2}(s-l)^2}}. \tag{4}$$

s and l are the positions of atoms of the sidechains and ligands respectively.

We calculate the pairwise distances between the ligand and sidechain atoms, with an exponential weighting scheme applied to emphasize closer atoms of the protein. The weights are calculated based on the exponential of the negative distances, indicating a stronger penalty for closer atomic interactions. The resulting weighted distances are then summed and normalized, contributing to an overall penalty term incorporated into the calculation of the root-mean-square deviation (RMSD) of the modified atoms. This RMSD, adjusted by the weighted penalty term, measures the structural deviation while accounting for steric clashes. The method reduces clashes by penalizing close atomic contacts and promoting greater separation between the ligand and protein, as seen in Table 3. While conformer matching already reduces the number of steric clashes, this penalty can further reduce the number. All RMSDs that are shown in this paper are calculated by removing the hydrogens and computing the distance between all atoms, not just the C-α backbone.

Table 3: **Steric clashes for *in-silico* structures.** This table analyzes the number of steric clashes between the receptor and the ligand.

Method	Percentage with Steric Clashes	Average Number of Steric Clashes
Crystal structures	14.3	0.2
ESMFold2 structures	76.7	19.1
Conformer-Matched	68.3	15.4
Conformer-Matched w/ penalty	67.7	13.9

C.2 Model Results

Given this definition of steric clashes, we can evaluate the different models, as done in Table 4. It can be seen that flexible models produce substantially more steric clashes, especially when executed on computationally generated structures. This aligns well with the fact that the ESMFold structure itself already exhibits many steric clashes. Our model produces more steric clashes than search-based methods on *in-silico* structures and drastically more on the crystal structure. For the ESMFold predictions, this may be because our model achieves more than four times the docking performance on this data, and the other methods typically predict wrong ligand poses, which are possibly far away (see high median RMSD). For example, SMINA predicts the least number of steric clashes, but also has the lowest docking performance. However, this table definitely highlights a shortcoming of our approach for at least crystal structures. Those shortcomings of ML docking methods have been discussed by Buttenschoen et al. [2023] and can be reduced by performing small optimizations of the predicted docking poses.

Table 4: **Steric clashes for top-1 predictions.** Comparison of the number of steric clashes between the receptor and ligand atoms using the predictions of different models and structures.

	Apo ESMFold Proteins		Holo Crystal Proteins	
Method	Percentage with Steric Clashes	Average Number of Steric Clashes	Percentage with Steric Clashes	Average Number of Steric Clashes
SMINA (rigid)	0.9	0.1	0.0	0.0
SMINA	60.4	12.8	1.1	0.0
GNINA (rigid)	5.4	0.4	1.7	0.1
GNINA	52.7	12.7	0.3	0.0
DIFFDOCK-POCKET (10)	69.3	9.8	57.7	4.4
DIFFDOCK-POCKET (40)	69.0	9.2	55.3	4.1

D Benchmarking Details

In our experimentation, we used NVIDIA RTX 6000 GPUs to conduct the assessment of our model's performance. To ensure robustness and reliability, we executed the model three times, each run initiated with seeds 0, 1, and 2. It is crucial to note that while seeds were employed to initialize the runs, achieving 100 percent reproducibility proved challenging due to the inherent non-deterministic nature of certain operations when executed on a GPU. To enhance the reliability of our reported values, we computed the mean across the three runs, providing a more stable and indicative measure of the model's performance rather than relying on individual figures from a single run. This approach ensures that our reported results reflect the averaged behavior of the model under different seed initializations, acknowledging and addressing the inherent stochasticity introduced by GPU computations.

D.1 Parameters for GNINA and SMINA

We opted to use the default/suggested parameters as much as possible when running GNINA and SMINA. We set the exhaustiveness (number of Monte Carlo chains for searching) to 8. When applying the flexible features we chose the flexible radius to be 3.5Å as in our model, where GNINA also specifies the flexible sidechains as we do during training with a rectengular prism. We generated 10 modes for each run on which we were able to evaluate top-N metrics and provide a fair assessment accounting for the variance of the results of the algorithm.

For site-specific docking, GNINA has two distinct approaches. The first method involves establishing a rectangular prism around the ground truth atom, utilizing the minimum and maximum values for the x, y, and z coordinates. This prism can be further customized with the addition of a buffer (and in case the box defined by the prism is too small, it is appended in such a way that the ligand can rotate inside of it). Alternatively, the second method permits the construction of a Cartesian box by directly specifying the coordinates. In our comparative analysis with our results, we opted for the Cartesian box approach, as it aligns more closely with our definition of the ligand-binding pocket. This choice was also motivated by the perception that the prism method, relying on knowledge of the original ligand position, may introduce strong bias. However, even when using the autobox method to level the playing field, our results demonstrate that the performance of our model remains

competitive. In this case, we compared the different approaches using the rigid model on crystal structures of the testset of PDBBind depicted in Table 5.

Even with no additional buffer when autoboxing the ligand, we can see that the results of GNINA are below 50% on the pre-processed files. We can also see that even doubling the exhaustiveness does not significantly affect the docking results. This plateau effect may indicate that the algorithm has adequately explored the conformational space, and additional computational resources do not lead to a proportional enhancement in the quality of predictions. When looking at the results of the preprocessed and original protein files, we can also observe that minor changes in the protein structure inputs result in significant differences in docking performance, suggesting a concerning sensitivity to variations in molecular configurations. This sensitivity is undesirable, especially when handling generated protein structures is a goal.

Clearly, the case of only autoboxing the ligand with no additional buffer does not reflect reality as the user would have to know the exact bounding box of the ligand with a 0Å margin of error. We can then observe that with an increase in the search space, the docking performance of GNINA deteriorates. The Cartesian pocket we selected exhibits very similar performance to the default setting, which incorporates a 4Å buffer through autoboxing, with only a marginal 1-2% difference. This justifies our comparison to the Cartesian box instead of the default GNINA settings while also being fair in having a similar pocket definition.

Table 5: **GNINA results with different attributes.** In this table, we present additional results for benchmarking GNINA: the differences in results with differently defined or sized pockets, exhaustiveness and input protein files.

		preprocessed PDB files				on original PDB files			
		Top-1 RMSD		Top-5 RMSD		Top-1 RMSD		Top-5 RMSD	
Pocket Type	Exhaustiveness	<2%	Median	<2%	Median	<2%	Median	<2%	Median
Our pocket center + 10Å	8	42.7	2.5	55.3	1.8	48.2	2.2	63.0	1.5
Autobox Ligand + 0Å	8	**48.0**	**2.2**	63.9	**1.5**	**53.0**	**1.9**	**69.8**	**1.3**
	16	45.7	**2.2**	**85.6**	**1.5**	-	-	-	-
Autobox Ligand + 4Å	8	43.6	2.3	58.1	1.7	51.0	**1.9**	67.2	**1.3**
	16	46.4	**2.2**	60.4	1.6	-	-	-	-
Autobox Ligand + 10Å	8	39.6	3.0	49.9	2.0	47.0	2.3	61.5	1.5
	16	42.2	2.7	54.7	1.8	-	-	-	-

E Further Analysis of Sidechain Reconstruction

The *in-silico* generated structures already have a median SC-RMSD of 1.5Å and 20.5% of structures have an SC-RMSD of less than 1Å. This means that the sidechain predictions of SMINA and GNINA are worse than those of structure generation algorithms without access to information about the ligand. This becomes more apparent when investigating these numbers visually in Figure 4. Both score-based methods improve the sidechains only in less than 10% of cases. Overall, DIFFDOCK-POCKET predicts sidechains that are substantially closer to the ground truth.

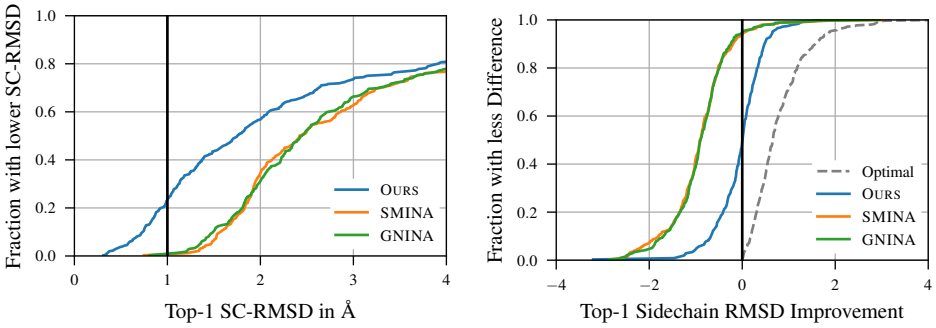

Figure 4: **Quality of predicted sidechains for *in-silico* structures.** *Left*: The cumulative distribution function shows how many instances have an SC-RMSD below a certain threshold to the holo structure. *Right*: The relative SC-RMSD between the structures before and after the predictions. The optimal line is computed by conformer matching the *in-silico* structures to the crystal structure.

F Performance on ColabFold

ColabFold [Mirdita et al., 2022] is a faster version of AlphaFold2 [Jumper et al., 2021] and is often used to generate a 3D structure based on a given sequence. In this part, we show how the model behaves on these structures instead of using ESMFold2 structures. This study is crucial since the model uses ESMFold embeddings during training for all proteins, and some of the training set also consists of high-quality structures predicted by ESMFold. This could mean that the model only works well with those specific structures while producing inferior results otherwise. To answer this, we have presented similar studies for ColabFold structures in Table 6, Table 7, and Table 8. We can see that the results are similar to those from ESMFold, letting us conclude that the model generalizes to well.

Table 6: **PDBBind docking performance with ColabFold structures.** Comparing the top-1 and top-5 results of multiple docking approaches when using structures generated by ColabFold.

	Apo ColabFold Proteins			
	Top-1 RMSD		Top-5 RMSD	
Method	%<2	Med.	%<2	Med.
SMINA (rigid)	5.7	7.5	13.1	5.5
SMINA	5.3	7.0	11.5	5.4
GNINA (rigid)	10.5	7.3	18.0	5.0
GNINA	7.7	6.8	15.6	4.9
DIFFDOCK-POCKET (10)	37.5	2.8	45.0	2.3
DIFFDOCK-POCKET (40)	**39.5**	**2.7**	**46.0**	**2.2**

Table 7: **Top-1 PDBBind docking with ColabFold structures.** More detailed performance evaluation when docking to *in-silico* structures generated by ColabFold.

	Ligand RMSD					Sidechain RMSD				
	Percentiles ↓			% below threshold ↑		Percentiles ↓			% below threshold ↑	
Methods	25th	50th	75th	2 Å	5 Å	25th	50th	75th	1 Å	2 Å
SMINA (rigid)	5.1	7.5	11.4	5.7	23.9	-	-	-	-	-
SMINA	5.0	7.0	9.7	5.3	25.6	1.9	2.3	3.2	0.6	32.1
GNINA (rigid)	3.7	7.3	11.6	10.5	34.8	-	-	-	-	-
GNINA	4.1	6.8	10.3	7.7	33.5	1.9	2.3	3.1	0.3	32.9
DIFFDOCK-POCKET (10)	**1.5**	2.8	**5.0**	37.5	**75.2**	**1.0**	**1.4**	**1.9**	**28.2**	**79.0**
DIFFDOCK-POCKET (40)	**1.5**	**2.7**	**5.0**	**39.5**	74.6	**1.0**	**1.4**	**1.9**	27.6	**79.0**

Table 8: **PDBBind sidechain performance with ColabFold structures.** Evaluating the performance of the sidechains when relying on *in-silico* structures generated by ColabFold.

	Apo ColabFold Proteins			
	Top-1 SC-RMSD		Top-5 SC-RMSD	
Method	%<1	Med.	%<1	Med.
SMINA	0.6	2.3	0.6	2.0
GNINA	0.3	2.3	1.2	1.9
DIFFDOCK-POCKET (10)	**28.2**	**1.4**	**35.1**	**1.2**
DIFFDOCK-POCKET (40)	27.6	**1.4**	34.9	**1.2**

G Additional Results

G.1 Further Docking Results

We have compiled a list of tables and figures that allow further evaluation of the docking results. In Table 9 and Table 10, we illustrate the different percentiles of our predictions for the ligand and sidechain predictions for both crystal structures and ESMFold. We also evaluate the models on a subset of the testset where UnitProt IDs that are present in the training or validation set have been

removed. The results are shown in Table 11. Figure 5 shows the cumulative distribution functions of the top-1 docking RMSD.

Similarly as for the ligand docking accuracy, we also provide further studies for the sidechain accuracy. Figure 6 illustrates the fraction of predictions with a lower sidechain RMSD for crystal structures and ESMFold structures respectively. Since the sidechains of ESMFold structures cannot be aligned completely to the crystal structures by only changing the torsional angles, Figure 7 shows further studies on the relative SC-RMSD. The relative SC-RMSD is computed by subtracting the SC-RMSD of the ESMFold structure from the SC-RMSD of the predicted protein.

Table 9: **Top-1 PDBBind crystal docking.** A more detailed performance evaluation of docking with holo crystal structures.

	Ligand RMSD					Sidechain RMSD				
	Percentiles ↓			% below Threshold ↑		Percentiles ↓			% below Threshold ↑	
Methods	25th	50th	75th	2 Å	5 Å	25th	50th	75th	1 Å	2 Å
SMINA (rigid)	1.6	4.5	8.0	32.5	54.7	-	-	-	-	-
SMINA	2.8	5.4	7.8	19.8	47.9	1.6	1.8	2.2	2.0	63.8
GNINA (rigid)	1.2	2.5	6.8	42.7	67.0	-	-	-	-	-
GNINA	1.8	4.6	7.9	27.8	54.4	1.4	1.7	2.1	3.3	71.9
DIFFDOCK-POCKET (10)	**1.1**	2.1	4.5	47.7	78.7	**0.6**	**1.0**	1.6	**49.2**	85.7
DIFFDOCK-POCKET (40)	**1.1**	**2.0**	**4.3**	**49.8**	**79.8**	**0.6**	**1.0**	**1.5**	48.7	**87.0**

Table 10: **Top-1 PDBBind ESMFold docking.** A more detailed performance evaluation of docking with computationally generated ESMFold structures.

	Ligand RMSD					Sidechain RMSD				
	Percentiles ↓			% below threshold ↑		Percentiles ↓			% below threshold ↑	
Methods	25th	50th	75th	2 Å	5 Å	25th	50th	75th	1 Å	2 Å
SMINA (rigid)	5.4	7.7	11.9	6.6	22.5	-	-	-	-	-
SMINA	5.5	7.3	9.9	3.6	20.5	1.9	2.4	3.7	0.6	34.4
GNINA (rigid)	4.1	7.5	12.0	9.7	33.6	-	-	-	-	-
GNINA	4.7	7.2	10.5	6.6	28.0	1.9	2.5	3.7	0.6	31.0
DIFFDOCK-POCKET (10)	1.3	**2.6**	5.1	41.0	74.6	**0.9**	**1.2**	**1.8**	**33.3**	79.6
DIFFDOCK-POCKET (40)	**1.2**	**2.6**	**5.0**	**41.7**	**74.9**	**0.9**	**1.2**	**1.8**	32.6	**80.3**

Table 11: **Filtered PDBBind docking performance.** This table mirrors the resutls from Table 1, but has filtered out all the complexes of the testset where the UniProt ID appears in the training or validation set.

	Apo ESMFold Proteins				Holo Crystal Proteins				
	Top-1 RMSD		Top-5 RMSD		Top-1 RMSD		Top-5 RMSD		Average
Method	%<2	Med.	%<2	Med.	%<2	Med.	%<2	Med.	Runtime (s)
DIFFDOCK (blind, rigid)*	-	-	-	-	20.8	6.2	28.7	3.9	40
SMINA (rigid)	6.5	7.7	15.9	6.2	29.0	5.1	45.7	2.2	258
SMINA	4.8	7.6	12.7	5.3	18.3	6.2	38.7	3.0	1914
GNINA (rigid)	10.1	7.2	20.3	5.3	**39.9**	2.6	**54.5**	**1.9**	260
GNINA	8.7	6.6	15.9	4.9	24.8	4.5	38.7	2.9	1575
DIFFDOCK-POCKET (10)	**27.7**	**3.3**	**34.6**	2.8	36.5	2.5	49.4	2.0	**17**
DIFFDOCK-POCKET (40)	26.3	**3.3**	33.6	**2.7**	39.2	**2.4**	52.4	**1.9**	61

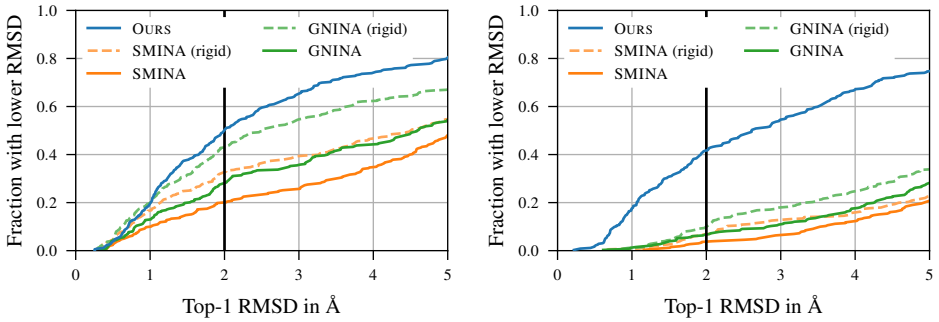

Figure 5: **Cumulative distribution function of RMSD.** *Left*: The CDF when using crystal structures as input. *Right*: The CDF when using ESMFold structures as input.

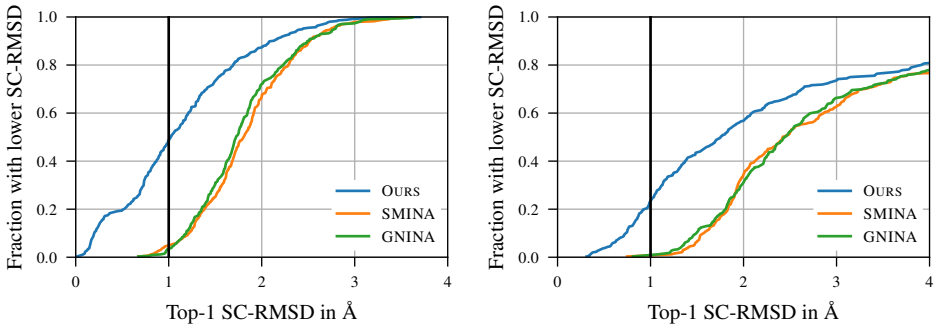

Figure 6: **Cumulative distribution function of SC-RMSD.** *Left*: The CDF when using crystal structures as input. *Right*: The CDF when using ESMFold structures as input.

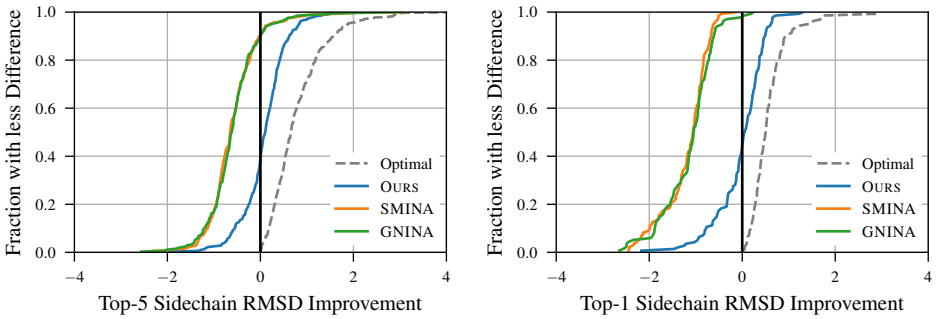

Figure 7: **Relative sidechain improvements on ESMFold structures.** *Left*: The relative sidechain improvement, when picking the top-5 sidechain prediction. *Right*: The relative sidechain improvement only for ESMFold complexes that have a pocket RMSD of < 1.5Å.

G.2 Rigid Model Comparison

In this section, we will investigate the impact of training with flexibility on the model's performance. For this, we trained a rigid model on the holo crystal structure of proteins with pocket reduction, and compared it to a flexible model. In all cases, we used models without low-temperature sampling, 20 inference steps and 10 samples per complex. In Table 12 this comparison is illustrated. We further added a comparison for when we use the flexible model, but do not predict the pose of any sidechain positions during training.

From this, we can see that training with flexibility improves the docking accuracy, especially for proteins where the true sidechain conformations are unknown (i.e., apo). We can also see that the performance decreases when using a flexible model in a rigid fashion. However, in our experiments, these effects were less prominent when relying on low-temperature sampling.

Table 12: **PDBBind docking performance rigid and flexible.** We compare the docking performance of a rigid model, a model that was trained with flexibility (marked with *), and the same model but without flexibility at inference†. None of the models use low-temperature sampling.

	Apo ESMFold Proteins				Holo Crystal Proteins			
	Top-1 RMSD		Top-5 RMSD		Top-1 RMSD		Top-5 RMSD	
Method	%<2	Med.	%<2	%<5	%<2	Med.	%<2	%<5
DIFFDOCK-POCKET (rigid)	29.8	3.6	40.7	76.7	44.7	2.4	55.0	86.5
DIFFDOCK-POCKET*	**37.7**	**3.0**	**45.9**	**82.2**	**45.4**	**2.2**	**57.2**	**87.6**
DIFFDOCK-POCKET†	24.9	4.0	41.0	76.8	27.7	3.5	45.9	81.5

G.3 Performance on Membrane Proteins

Membrane proteins make up more than 60% of the drug targets in humans and hence play a crucial role in drug discovery [Overington et al., 2006]. In the testset of PDBBind, there are nine proteins that are membrane proteins that have been classified as such by either White [2009]; Lomize et al. [2011]; Kozma et al. [2012]; Newport et al. [2018]. The corresponding PDB ids are: 6e4v, 6h7d, 6iql, 6kqi, 6n4b, 6qxa, 6qzh, 6r7d, 6rz6. The docking performance of our model on these nine proteins is illustrated in Table 13. We can see that for experimentally generated crystal structure and ColabFold membrane proteins our model archives only in 33.3% of cases a ligand RMSD of < 2. For ESMFold, there is no successful docking for these proteins. We believe this is the case because the quality of the structure of ESMFold is worse on these proteins as ColabFold (compare Table 14).

Since the available number of membrane proteins in our testset is small, this study does not allow us to give definitive answers on the performance of our model on these types of proteins.

Table 13: **Docking performance on PDBBind membrane proteins.** This table denotes the Top-1 ligand RMSD on the listed proteins for different protein structures.

Protein Structure	Top-1 Ligand RMSD in Å								
	6e4v	6h7d	6iql	6kqi	6n4b	6qxa	6qzh	6r7d	6rz6
Crystal	8.8	1.6	**2.2**	3.5	**1.2**	13.7	**2.3**	**4.8**	2.0
ESMFold	**5.6**	3.4	3.1	**2.3**	5.3	10.5	5.1	9.0	2.7
ColabFold	6.3	**1.5**	2.9	3.6	1.7	**10.0**	5.5	5.4	**1.9**

Table 14: **Pocket RMSDs of PDBBind membrane proteins.** The RMSDs between the atoms of the receptor and the computationally generated protein are shown in this table.

Protein Structure	Pocket RMSD in Å								
	6e4v	6h7d	6iql	6kqi	6n4b	6qxa	6qzh	6r7d	6rz6
ESMFold	**2.3**	1.7	**4.0**	2.8	4.5	5.9	**3.7**	8.6	2.7
ColabFold	3.0	**1.3**	4.2	**1.7**	**2.8**	**3.8**	5.8	**1.2**	**2.1**

G.4 Confidence Model Evaluation

To determine the effectiveness of the confidence model, we have compared how the impact of the number of generated samples on the quality. When having a strong confidence model, the performance with more samples will be monotonically increasing. This analysis is illustrated in Figure 8 for RMSD, SC-RMSD, and for crystal and ESMFold structures respectively. However, if the model only produced very similar poses, then the number of generative samples would not be indicative of the quality of the confidence model. To further investigate the performance of the confidence model, we compare the selective accuracy. For this, we rank the confidence of all top-1 predictions and discard the lowest-ranking ones (according to the confidence model). How this selection compares to an oracle with perfect selection gives insight into the quality of the confidence model. This is

shown in Figure 9, where we see that the confidence model works especially well for the RMSD, and is less accurate for the SC-RSMD. In all cases, a higher confidence correlates with a better pose.

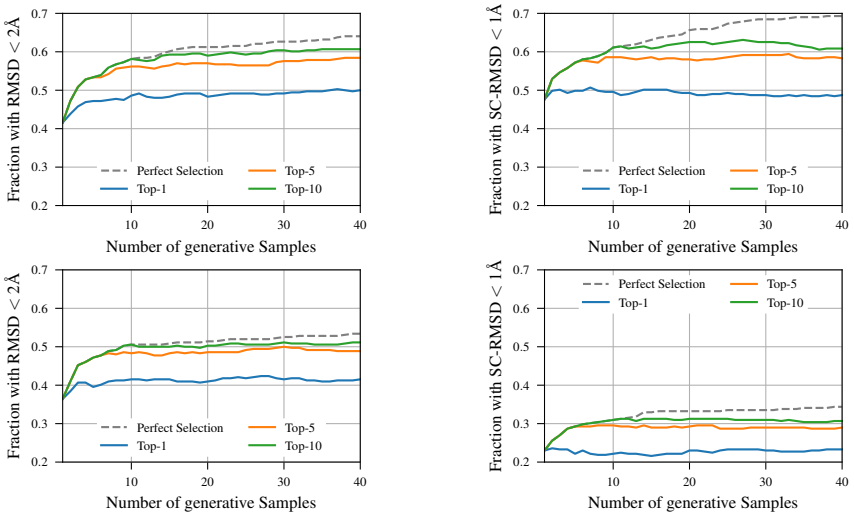

Figure 8: **Performance based on number of generative samples.** Compare the top-1, top-5, and top-10 accuracy based on the number of samples generated by our procedure. In *left*, the RMSD of the ligand can be seen, whereas *right*, the sidechain RMSD is illustrated. In the *top* row, the input are crystal structures, while the *bottom* row uses structures generated by ESMFold.

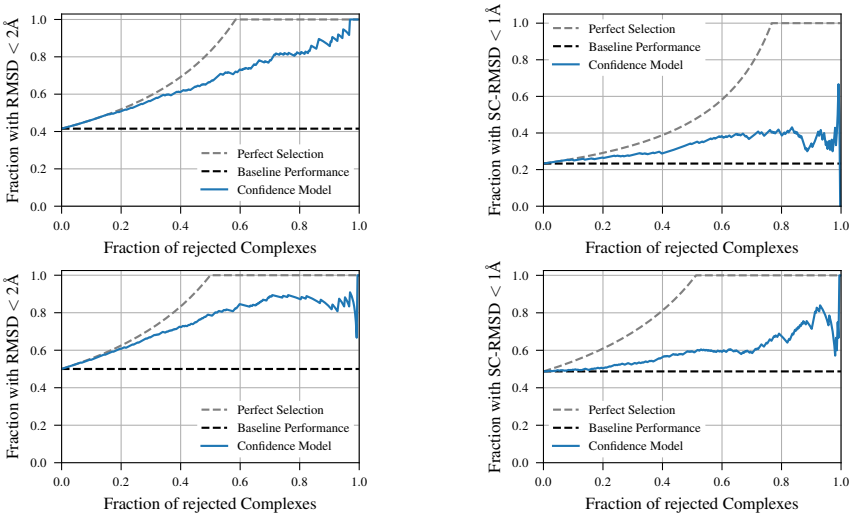

Figure 9: **Selective accuracy of the score-model.** Compare the performance of the model with respect to the confidence model, and a perfect selection. In *left*, the RMSD of the ligand can be seen, whereas *right*, the sidechain RMSD is illustrated. In the *top* row, the input are crystal structures, while the *bottom* row uses structures generated by ESMFold.

G.5 Performance based on Quality of Computational Structures

While we saw that the docking results between ESMFold and ColabFold structures did not change much, we will investigate whether the quality of the computationally generated structures impacts the performance. Figure 10 shows the overall quality of the predictions by illustrating the RMSD to the ground truth protein structure in the pocket. We see that more than half of the predictions have an RMSD of < 2Å to the ground truth structure. Figure 11 shows the percentage of complexes with a good RMSD and SC-RMSD respectively. For this, we have split the test set into roughly three equally sized parts based on the RMSD of all atoms in the pocket between ESMFold structures and the ground truth crystal structures. We can clearly see that the performance degrades with

worse predictions. For structures that are not accurate, our method is not notably better than others. Especially for the sidechains, the prediction quality of our model strongly depends on the quality of the computationally generated structure.

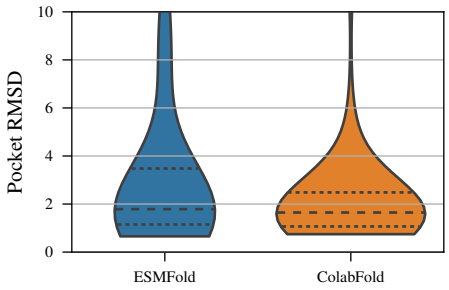

Figure 10: **Pocket RMSD between apo and holo structures.** Apo ESMFold and ColabFold structures have been aligned with the holo crystal structures such that the RMSD in the pocket is the lowest. This figure shows the RMSD of the pocket for proteins in the test set. The dashed lines represent the 25%, 50%, and 75% percentiles respectively. This figure does not show outliers having an RMSD larger than 10Å.

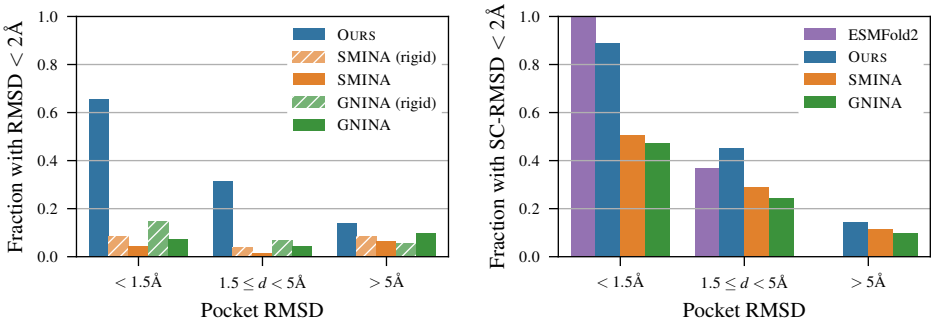

Figure 11: **Model accuracy based on quality of ESMFold predictions.** Comparison of the model accuracy with three different levels of the quality of ESMFold predictions. The predicted ligand (*left*) and sidechain quality (*right*) are evaluated respectively.

G.6 Number of Reverse Diffusion Steps

We evaluated multiple values for the concrete number of reverse diffusion steps on the validation set to determine the best number at inference time. The results are visualized in Figure 12. 30 reverse diffusion steps yielded the best results while not impacting the performance too much. We can see that we could reduce the number of reverse diffusion steps to 20 without losing too much performance. This reduction in reverse diffusion steps could reduce the runtime by up to 33%.

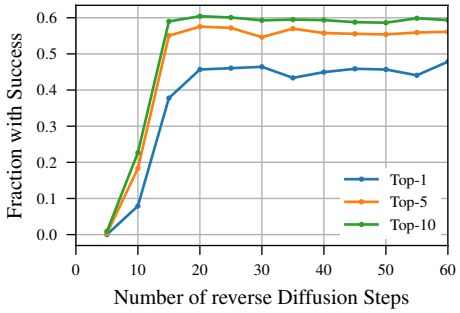

Figure 12: **Comparison of the number of reverse diffusion steps.** Results of the inference with different reverse diffusion steps on the validation set. The values on the y-axis shows the fraction of samples where the RMSD is < 2Å and the SC-RMSD is < 1Å.

H Cross-docking to CrossDocked 2020

To demonstrate that the model can generalize to different scenarios, we evaluated it on the task of pocket-level cross-docking. When comparing works that use site-specific docking, it is important to compare which pockets they used and if the definitions are similar enough not to skew the results. More accurate pockets typically result in better predictions. For this testset, we present the numbers for three different choices of pockets.

1. Pre-process the pockets from Brocidiacono et al. [2023] by computing the mean of the α-carbon atoms in the pocket. This does not use any additional data and follows a more similar definition to our pocket. These numbers represent the fairest comparison.
2. Use the pocket center definition as we did in training which is defined as the mean α-carbon atoms that are within 5Å of any ligand atom. This requires the ground truth ligand and would thus be an unfair comparison. Marked with a *.
3. Use the pocket center definition as Brocidiacono et al. [2023] where they rely on information from multiple ligands [Brocidiacono et al., 2022]. This can be very different from our definitions. Marked with a †.

The results can be seen in Table 15. Our model achieves a pocket-normalized RMSD of less than 2Å in 28.6% of instances, compared to the best other method of 24.4%. As for the overall accuracy, GNINA yields the best results. Brocidiacono et al. [2023] argue that the pocket-normalized score is more important since the size of the dockings per pocket is unevenly distributed. These results for our model are especially impressive considering that a) cross-docked structures were never seen during training, but some of the other approaches trained with this data, and b) the definition of the pocket center was out of distribution for our model. If the pockets were constructed the same way as in training (i.e., no distribution shift but different data than competitors), we would achieve results improving on the state-of-the-art in all < 2Å accuracy metrics. Even giving better predictions than GNINA. When using the exact pockets specified by Brocidiacono et al. [2023], the results are slightly worse than those presented in the paper's main text but still show the same trend.

I Visualization of Docking Results

We present the visualization for four different dockings in Figure 13. An animation of the docking process for multiple complexes can be found in our repository at `https://github.com/plainerman/DiffDock-Pocket`.

Table 15: **Cross-docking performance on CrossDocked 2020.** Evaluation of the top-1 RMSD between different methods on the CrossDocked 2020 testset with complexes removed that were seen during training. The pocket-normalized percentage is presented for each value, and the overall score is listed in brackets. For the pocket-normalized score, the average performance on each pocket is reported instead of the performance across all complexes. Numbers for the methods marked with a $^+$ were taken from Brocidiacono et al. [2023].

Method	Top-1 RMSD %<2	Top-1 RMSD %<5	Average Runtime (s)
VINA$^+$	11.7 (15.6)	40.2 (37.9)	73.7
GNINA$^+$	21.5 (23.5)	51.7 (47.3)	51.6
DIFFDOCK$^+$ (blind)	17.3 (11.6)	51.7 (47.3)	98.7
PLANTAIN$^+$	24.4 (15.2)	**73.7** (**71.9**)	**4.9**
DIFFDOCK-POCKET (10)	28.3 (17.7)	67.5 (50.2)	22.0
DIFFDOCK-POCKET (40)	28.6 (18.5)	67.9 (49.4)	87.2
DIFFDOCK-POCKET* (10)	**32.7** (**31.8**)	68.2 (71.5)	20.6
DIFFDOCK-POCKET† (10)	26.8 (17.0)	67.2 (50.5)	21.4
DIFFDOCK-POCKET† (40)	28.3 (18.2)	68.2 (49.6)	71.6

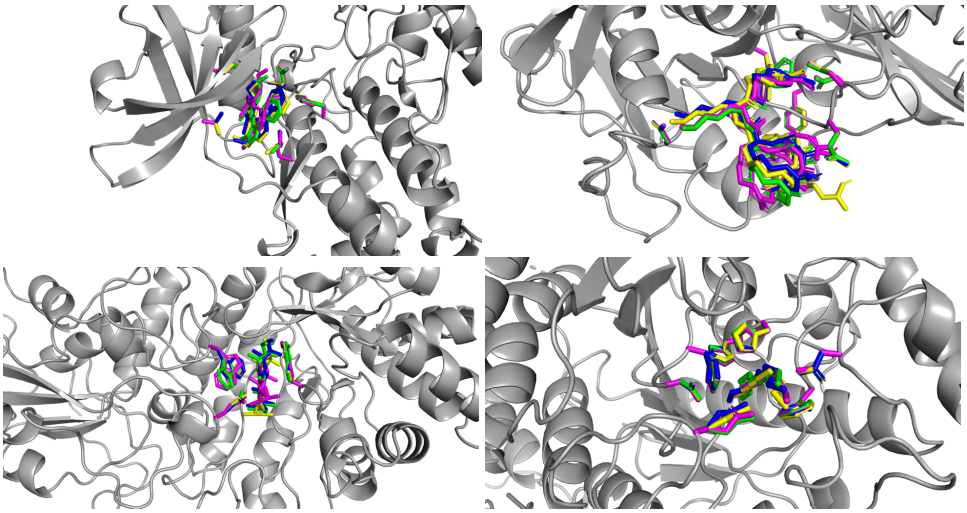

Figure 13: **Flexible docking of unseen complexes.** Visualization of the results of four dockings on arbitrarily selected complexes (*top*: 6a1c, 6hzb, *bottom:* 6md6, 6uii). Four different poses for the sidechains and ligand are presented in different colors.

J Evaluation with PoseBusters

We have evaluated our results with the PoseBusters [Buttenschoen et al., 2023] method to determine the percentage of our results which are physically plausible. For this, we used two separate tests implemented by Buttenschoen et al. [2023], one that measures the quality of the predicted complex structures (including intramolecular and intermolecular validity, such as the bond lengths and internal steric clashes in the ligand or its volume overlap with the protein) and the redocking success, which also takes into account the accuracy of the prediction of the ligand, including that the RMSD between the predicted and true ligand is below 2Å but also checking that the molecules have the same chirality and double bond stereochemical properties. We report these results on our model and baselines for both holo-crystal and ESMFold generated apo structures of the PDBBind benchmark in Table 16.

Table 16: **Results of PoseBusters quality check.** We report the percentage of predictions that pass all Posebuster quality checks required for the docked complex structure and the redocked structure compared to the ground truth ligand.

	ESMFold Structures		Holo-crystal Structures	
Method	Docking Structure	Re-docking	Docking Structure	Re-docking
GNINA (rigid)	90.4	7.0	**95.8**	**36.3**
GNINA	**93.2**	4.2	95.4	14.8
DIFFDOCK	11.4	3.6	23.4	18.4
ESMFOLD	16.0	-	-	-
DIFFDOCK-POCKET (40)	21.6	**10.9**	29.4	17.4

The results on holo-crystal structures align with the findings presented by Buttenschoen et al. [2023]: The classical model GNINA outperforms both deep learning models in both the physical plausibility of predicted structures as well as the physical plausibility of *good* predicted structures. Comparing DIFFDOCK-POCKET with DIFFDOCK, we can observe that while the percentage of generated structures that pass all PoseBuster quality checks is higher for DIFFDOCK-POCKET, this advantage disappears when looking at the percentage of structures that are also considered to be a successful redocking attempt and DIFFDOCK even slightly outperforms DIFFDOCK-POCKET.

What we find very promising and something we believe would require further examination is the results on ESMFold structures. In Table 16, we also report that only 16% of ESMFold generated protein structures pass all quality checks when comparing it with the ground-truth ligand. GNINA and DIFFDOCK-POCKET both improve on this number in their generated structures which can be attributed to better sidechain positions. However, although more than 90% of generated structures by GNINA are considered correct, DIFFDOCK-POCKET outperforms all methods when considering successful redocking. This suggests a possible advantage DIFFDOCK-POCKET could have over classical approaches when docking to apo structures, however further examination is needed.

Altogether we can report that on the PDBBind testset DIFFDOCK-POCKET outperforms DIFFDOCK on the percentage of generated structures that pass all PoseBusters checks (with DIFFDOCK being the best deep-learning method reported by Buttenschoen et al. [2023]) and outperforms all considered methods on redocking to ESMFold structures. Further evaluation on the PoseBusters benchmark for both holo-crystals and generated structures as well as comparison to other docking methods is needed to reassure these claims and is an objective of future work.