DIFFDOCK-POCKET: Diffusion for Pocket-Level Docking with Side Chain Flexibility

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

Abstract

When a small molecule binds to a protein, the 3D structure and function of the protein can significantly change. Understanding this process, called *molecular docking*, is crucial in areas such as drug design. Recent learning-based attempts have shown promising results at this task, yet lack the necessary features that traditional approaches support. In this work, we close this gap by proposing DIFFDOCK-POCKET: a diffusion-based all-atom docking algorithm conditioned on a binding target. Our model supports receptor flexibility by extending the generative diffusion process to the manifold describing the main degrees of freedom of the protein's side chains. Empirically, we improve the state-of-the-art in site-specific-docking on the PDBBind benchmark. In particular, in the realistic scenario that no bound protein structure is available, we double the accuracy of current methods while being 20 times faster than other flexible approaches.

1. Introduction

Molecular docking—the task of predicting the structure in which a small molecule (ligand) binds to a protein (receptor)—encompasses a wide range of problem formulations. Several approaches only model the ligand as flexible while assuming the protein structure to be rigid. This simplification fails to capture the protein's conformational change resulting from the ligand's interactions with the protein. The most significant part of that change is in the protein side chains that directly interact with the ligand. Thus, the most common real-world docking task and the one we tackle is: given the ligand's 2D molecular graph and the protein pocket's *unbound* structure, predict the ligand structure and the *structure of the side chains* interacting with the ligand.

[1]School of Computation, Information and Technology, Technical University of Munich [2]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology. Correspondence to: Michael Plainer <michael.plainer@tum.de>.

No publicly available deep learning method has considered this task formulation yet, despite the amount of work dedicated to docking (Stärk et al., 2022; Lu et al., 2022; Corso et al., 2023) making it challenging to apply deep learning methods in practice (Yu et al., 2023). Existing docking approaches that allow for flexible receptors are constructed using the traditional search-based paradigm which however fails to grapple with the significantly increased dimensionality of the search space that occurs with protein flexibility.

In this work, we introduce DIFFDOCK-POCKET for pocket-level flexible docking with accurate atomic-level structure. DIFFDOCK-POCKET's modeling choices aim to achieve a favorable tradeoff between efficiency and generality. We use the often available prior knowledge about the binding site, which is where the ligand has the most significant effect on the protein structure, and we restrict protein flexibility to the side chains interacting with the ligand while keeping the more rigid backbone atoms fixed. For this reduced number of atoms, we further limit the $\mathbb{R}^{3(m+n)}$ cartesian search space for n ligand and m flexible protein atoms to a submanifold that captures the prior knowledge of the side chains' flexibility lying in their torsion angles.

We frame the problem as a generative modeling task and develop a diffusion model jointly over the protein side chain torsion angles, the ligand torsion angles, and the relative position of the protein and the ligand. Starting from a random ligand structure placed *in the pocket* surrounded by randomly initialized side chains, DIFFDOCK-POCKET iteratively updates the joint structures towards realistic binding conformations. After drawing multiple samples, we score and rank them with an additional confidence model to find the most plausible structures (compare Figure 1). Hence, the motivation and technical ideas behind DIFFDOCK-POCKET can be understood as an extension of DIFFDOCK (Corso et al., 2023) to model the critical aspects of protein flexibility that is simple yet effective. This extension is non-trivial in requiring innovations such as *side chain* conformer matching to align the distribution of side chain conformers seen during training with the distribution during inference.

We demonstrate DIFFDOCK-POCKET's effectiveness along multiple dimensions of problem settings on the PDBBind

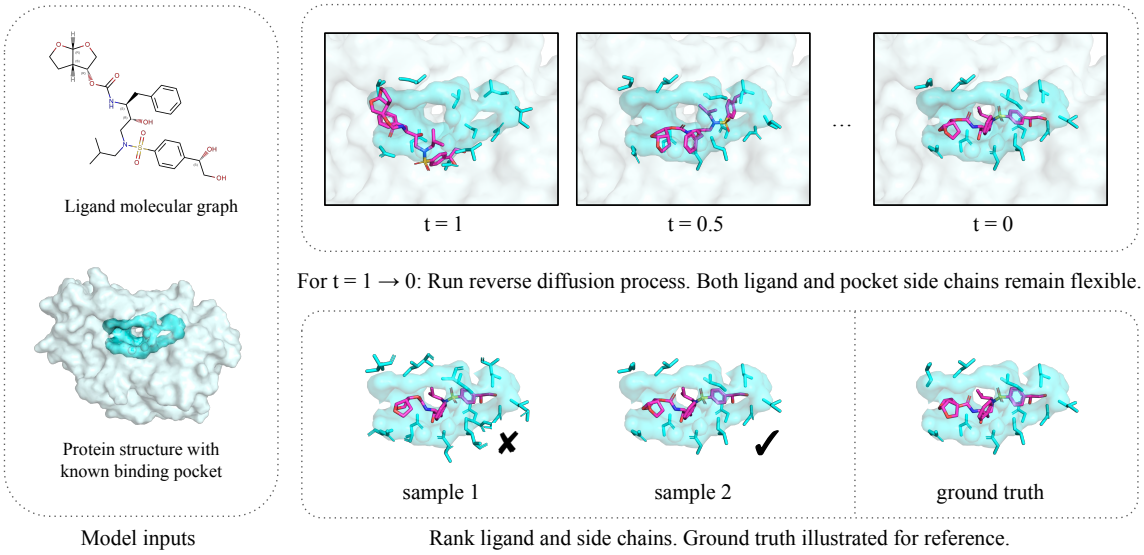

Figure 1. **Overview of our approach.** The model takes a ligand, a protein structure, and the binding pocket as input. The inference starts with random ligand poses (magenta) and side chain conformations (blue), which are gradually improved by a reverse diffusion process until reaching $t = 0$. The generative process modifies the translation, rotation, and torsional angles of the ligand and the torsional angles of the receptor's side chain atoms to predict a final pose for each. A score model then ranks the quality of the samples.

(Liu et al., 2017) and CrossDocked (Francoeur et al., 2020) datasets, starting from bound protein structures, homologous protein structures, and computationally generated structures. We also perform energy minimization of the final ligand poses and assess the physical plausibility of the resulting joint structure. Empirically, we achieve 41.7% of ligand structure predictions with root mean squared deviation (RMSD) under 2Å compared to the 20.3% of the next best method for docking to pockets generated with ESMFold (Lin et al., 2022) for the PDBBind benchmark. For the same task, the side chain structure prediction of DIFFDOCK-POCKET has an RMSD less than 1Å for 33.3% of predictions compared to 0.6% of flexible baselines. Similar improvements hold across various other evaluations we carry out. Hence, DIFFDOCK-POCKET provides useful predictions of the ligand structure and, crucially, the protein atoms interacting with the ligand, making it valuable for analyses such as molecular dynamics simulations, free energy calculations, or cheaper binding affinity estimates.

2. Related Work

Molecular docking. The binding of a ligand to a protein occurs when they can find an accessible energetically favorable bound conformation. Traditional search-based docking methods (Friesner et al., 2004; Thomsen & Christensen, 2006; Trott & Olson, 2010) make use of that and minimize a scoring function that approximates the energy of a given configuration. Approaches such as GNINA (McNutt et al., 2021) use ML to approximate this scoring function, while others such as SMINA (Koes et al., 2013b) use a handcrafted potential.

As the search space to minimize the scoring function can be large, several deep learning methods were developed in recent years to directly sample valid bound conformations bypassing the search process. Initially, these methods (Stärk et al., 2022; Lu et al., 2022) used regression-based targets to supervise the pose prediction but often obtained highly unphysical conformations (Buttenschoen et al., 2023). Corso et al. (2023) argued uncertainty was the major cause of these artifacts and developed a generative model for docked ligand poses. However, all these recent deep-learning models dock the ligand blindly on the complete protein, instead of limiting themselves to a given pocket, making them less useful in practice (Yu et al., 2023).

Docking to unbound structures. Almost all approaches model docking with ligand flexibility, but some do not account for the changes that can occur in the protein (Friesner et al., 2004; Stärk et al., 2022; Lu et al., 2022; Corso et al., 2023). This is especially important in real-world scenarios, where one either has access to the structure of the protein bound to a similar molecule (cross-docking), an unbound (apo) structure, or only to computationally generated structures.

In fact, the quality improvement of computationally generated structures has not led to a direct improvement in binding prediction, mostly due to the incapacity of traditional docking tools to deal with unbound or imprecise structures (Wong et al., 2022; Karelina et al., 2023). Recently, DIFFDOCK showed significant improvements over traditional methods in docking accuracy to *in-silico* generated structures. We argue that part of the success came from modeling proteins at the residue level which makes the model less sen-

sitive to specific atomic placement and allows it to implicitly reason about side chain flexibility. However, modeling only the residues can lead to a loss in prediction quality, as the model cannot learn to avoid steric clashes or other physical constraints.

Flexible docking. Beyond an accurate docking pose (Zhao & Sanner, 2007; Hogues et al., 2018), modeling protein flexibility and predicting the precise bound protein structure at an atomic level is necessary for many downstream applications such as in virtual screening (Teague, 2003) or binding affinity prediction. Search-based approaches such as GNINA or SMINA can include the side chain flexibility in their stochastic energy-optimization procedure. However, this can drastically increase the search space and the computational effort, and thus reduce the accuracy. For ML models, modeling receptor flexibility can be challenging and is typically unsupported (Corso et al., 2023; Stärk et al., 2022; Lu et al., 2022). NEURALPLEXER (Qiao et al., 2023) and DOCKGPT (McPartlon & Xu, 2023) are recent learning-based docking algorithms, respectively for protein-small molecule and protein-protein complexes, that can model protein flexibility. However, as of writing, these programs cannot be used by the public and no code is available.

Diffusion models. Previous works (Corso et al., 2023; Qiao et al., 2023) have shown that generative modeling, and in particular diffusion models, are well-suited for docking due to their ability to capture the stochastic nature of the biological process and its uncertainty. Score-based diffusion models (Song et al., 2021) define a continuous diffusion process $dx = f(x,t)\,dt + g(t)\,dw$ that (approximately) transforms the data distribution p_0 in an easy to sample prior p_T. This 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. Using denoising score matching, one can learn $s(x,t) \approx \nabla_x \log p_t(x)$ and use it to run the reverse SDE to obtain samples from the data distribution. We employ a similar approach, where we add noise to the conformations of the ligand and the protein, and predict their score to iteratively reverse the SDE.

3. Method

3.1. Summary

Given a ligand and a protein unbound structure, flexible docking consists of predicting the bound structure of both the ligand and the protein, i.e. the position of its atoms 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 $\mathbb{R}^{3(m+n)}$. This large space of conformations combined with the small number of data points available makes flexible docking a particularly challenging problem. We therefore focus on finding a suitable way to reduce the dimension of this search space by using domain knowledge and building this knowledge into our diffusion-based generative model. The combination of these simple, yet effective, reformulations allows us to account for the most common protein conformational changes and to improve on existing docking software.

For one, the effects of most ligand-binding interactions are local to the binding pocket region. To account for this, we limit the diffusion process to only predict poses and changes to the protein structure within this pocket. This significantly reduces the number of protein atoms to be considered flexible and aligns with most existing use cases of docking in drug discovery, where the user already knows the binding pocket in advance.

Further, protein structures are composed of a relatively rigid backbone surrounded by more flexible side chains. During docking, these side chain atoms close to the binding site display the most significant structural change (Clark et al., 2019) and their movement is key to enabling docking. Therefore, we are left with the prediction of the bound ligand pose and the binding pocket side chain arrangement.

We combine these observations with the approximation of using torsion angles to capture the main degrees of freedom of a molecular structure (Jing et al., 2022). We apply this not only to the ligand, but also to the receptor side chains. In Section 3.2, we show how these remaining degrees of freedom can be mapped to a low-dimensional product space over which we define our diffusion process. Then, in Section 3.3, we describe how we project the ground truth data to this manifold. Finally, in Section 3.4, we present the architecture and training/inference processes.

3.2. Product space diffusion

We restrict the degrees of freedom of the problem to the relative position and orientation of the ligand w.r.t. the receptor, the torsion angles of the ligand conformation, and the torsion angles of the side chains in the binding pocket. Following the convention from other docking algorithms (McNutt et al., 2021), we define the receptor's amino acids with at least one heavy atom within 3.5Å of any ligand heavy atom as flexible. This can be changed during inference.

These degrees of freedom define a complex $6 + k + \ell$ dimensional submanifold of $\mathbb{R}^{3(m+n)}$ over which we want to learn a diffusion model, where there are 6 rototranslational degrees of freedom, and k and ℓ are the number of rotatable bonds of the ligand and receptor side chains respectively.

One option to learn the model would be to use the extrinsic definition (i.e., over 3D space) of the submanifold and use the Riemannian score-based generative modeling formulation from De Bortoli et al. (2022). However, this would require using slow simulation-based techniques to sample

at both training and inference time. Another way to project the data to internal coordinates (i.e., bond lengths, bond, and torsion angles) is to define the diffusion model exclusively over this intrinsic representation. This formulation would make the ability of the score model to reason about 3D interaction between ligand and protein atoms incredibly challenging. Instead, we bypass these issues by applying the Intrinsic Diffusion Modeling technique (Corso, 2023) introduced by Jing et al. (2022), where we use the intrinsic coordinates to define the diffusion process but the score model still primarily operates in the extrinsic space.

In particular, we note that transformations on the extrinsic $3 + 3 + k + \ell$ dimensional submanifold can be mapped to transformations on an intrinsic product space $\mathbb{P} = \mathbb{R}^3 \times SO(3) \times SO(2)^k \times SO(2)^\ell$. We then define the diffusion process on this product space, while always operating on structures in 3D space. Given the current relative pose and structures of ligand and receptor, the score model predicts an element of the tangent space of $\mathbb{P}$ at every step whose (scaled) exponential map intuitively gives us the different transformations to apply to the current pose: a translation and rotation of the ligand w.r.t. the receptor and updates to the ligand and side chains torsion angles.

3.3. Side Chain Conformer Matching

We have defined the space of poses over which our diffusion model will generate poses as those reachable from a given unbound protein structure and a randomly placed ligand in a low energy conformer through rigid rototranslations of the ligand, changes in the ligand torsion angles and the receptor binding pocket side chain torsion angles. Due to the imperfection of our approximations, however, the ground truth poses that we have for training are not elements of these submanifolds (e.g. because there are small changes in the receptor backbone upon binding). To correctly use these structures for training, we need to define a projection that maps them to a point of our low dimensional manifolds.

For the ligand torsion angles, we use the conformer matching procedure defined by Jing et al. (2022) which consists of finding the torsion angles that minimize the RMSD between the original ligand and conformer matched structures. For the relative position of ligand and receptor, we use individual Kabsch alignments of the ligand (original vs. conformer matched) and the protein backbone coordinates (holo vs. apo), similarly to Corso et al. (2023). For efficiently projecting the side chains, we formulate their conformer matching as individual RMSD minimization problems for each side chain when changing the torsion angles without modifying the backbone structure or alignment. Let $\tilde{\boldsymbol{x}}$ be the conformer matched ligand structure, $\boldsymbol{y}$ the holo side chain, $\boldsymbol{y}'$ the apo side chain, χ the set of possible torsion angle values for the side chain $\boldsymbol{y}$, then the conformer matched structure of side chain $\boldsymbol{y}$ is

$$\tilde{\boldsymbol{y}} = \underset{\hat{\boldsymbol{y}} \in \{\text{apply}(\boldsymbol{y}', \boldsymbol{\chi})\}}{\arg\min} \text{RMSD}(\boldsymbol{y}, \hat{\boldsymbol{y}}) \cdot \text{penalty}(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{y}}). \quad (1)$$

The additional penalty in the optimization goal was introduced to make the conformer-matched complexes 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 B. The minimization is solved with differential evolution, which iteratively combines potential solutions of a population to converge to the global minimum. We can use the computationally generated (apo) structure where the side chains have been conformer-matched with the bound structure during training. This matching still leaves some distance between the structures (as seen in Figure 2) but aligns with our definition of a semi-flexible receptor.

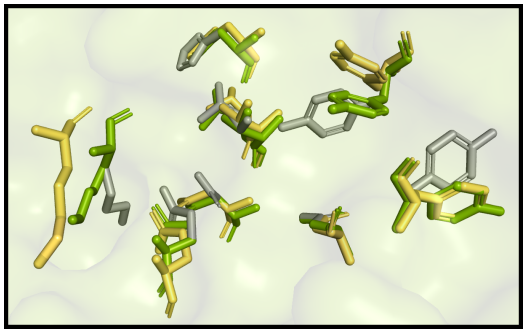

Figure 2. **Side chain structure matching.** Optimize the side chain torsional angles (green) of the computationally generated structure (gray) to minimize the distance to the ground truth positions (yellow). Due to a difference in the backbone and bond lengths between the two structures, they cannot be fully aligned.

3.4. Model Architecture and Training

Data preprocessing. Given the large number of atoms proteins can have and our focus on docking to known binding pockets, as a first step, we restrict our model to only operate on the amino acids in the binding pocket and its surroundings. For this, we discard all amino acids too far from the binding site and reduce our protein to the binding pocket. Figure 3 highlights this restriction whose (see Appendix C.3 for details). This allows us to model even large proteins at the atom-level, without impacting the performance.

Models. Although the underlying graphs are different, the model architecture we are using is inspired by the structure of DIFFDOCK (Corso et al., 2023) and consists of two different models which are executed in sequence during inference: 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: a translation vector, a rotation vector, and a real value for each of

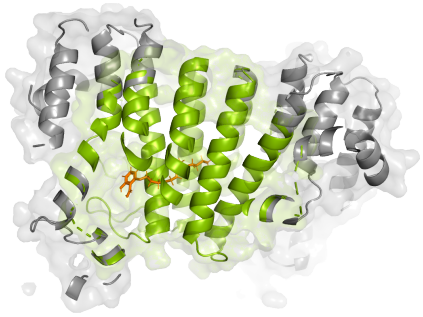

Figure 3. **Pocket reduction.** Only retain amino acids (green) where any heavy atom is within the specified pocket radius determined by the size of the ligand (orange) and a buffer.

the rotatable bonds in the ligand and receptor flexible side chains. With the knowledge of the scores during inference, we can take a protein with pocket and a ligand structure in 3D space and produce $n \in \mathbb{N}$ different complex structures $\left(\tilde{\boldsymbol{x}}^{(1)}, \tilde{\boldsymbol{y}}^{(1)}\right), \ldots, \left(\tilde{\boldsymbol{x}}^{(n)}, \tilde{\boldsymbol{y}}^{(n)}\right)$.

The *confidence model* is then used to rank each protein-ligand prediction $\left(\tilde{\boldsymbol{x}}^{(i)}, \tilde{\boldsymbol{y}}^{(i)}\right)$ such that the best-predicted structures can be selected. Our training routine and objective are defined so that the confidence model learns to predict the accuracy of generated binding structures by considering both the ligand's docking success and the similarity of flexible side chains to the bound structure. The output of the confidence model is a logit and allows practitioners to estimate the accuracy of the predictions without having access to the ground truth.

Architecture. We use the conformer-matched structure of both the ligand and protein and represent them as geometric graphs where each atom is a node and edges are formed between nearby atoms or chemical bonds. There are edges between ligand-ligand atoms, receptor-receptor atoms, and also receptor-ligand atoms. Moreover, we also define additional nodes representing the amino acids of the receptor. This allows us to more efficiently propagate information about the receptor hierarchically over larger distances.

The core architecture between both models is shared and mostly differs in the last few layers. We expect the predicted score vectors on the rotation and translation tangent spaces to be $SE(3)$-equivariant and the torsion angle score values to be $SE(3)$-invariant. We achieve this by using $SE(3)$-equivariant convolutional networks, so-called tensor field networks (Thomas et al., 2018; Geiger et al., 2022) that encode the data into irreducible representations of the $O(3)$ group centered at every node. Six tensor-product-based message-passing layers are executed on the graphs using different weights across different edge types.

After the message-passing layers, the architectures of the score and confidence models differ. We follow the ideas from Corso et al. (2023) for the prediction of the translational score, the rotational score, and ligand torsional scores based on the features and position of the ligand atoms. For the side chain torsional score, we define a pseudotorque layer (Jing et al., 2022) on the receptor atomic features. The confidence model uses the predicted structures represented with the same graph structure as input and aims to determine the probability that the docking is accurate. Its output is a single $SE(3)$-invariant scalar, which is predicted by a feedforward network taking as input the final aggregated flexible atoms and ligand atoms representations.

Training. We use denoising score matching (Song et al., 2021) to train the score model by sampling the transformations from the perturbation kernels (intrinsic space), applying them to the input structures of our model (extrinsic space), and minimizing the denoising score matching loss function.

To train the confidence model, we first sample ligand and side chain configurations with the score model with a higher diversity than we would see during inference. The predictions are then compared with the ground truth training data to assess their quality. For this, we introduce a binary criterion that is 1 if the RMSD of the ligand is within 2Å, and the RMSD of the side chains is within 1Å of the ground truth. This way, the confidence model learns to predict if the sampled configuration is plausible by minimizing a binary cross-entropy loss on those generated structures.

Inference. To predict a docked complex, we apply random transformations in all degrees of freedom to start from an arbitrary ligand and flexible side chain conformation. We then use the score model and its exponential map to the product manifold at each timestep to traverse the reverse SDE. Once the diffusion process is finished, the samples are ranked by the confidence model to predict probable poses.

Due to the maximum likelihood training and model uncertainty, 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 temperature values that maximize the performance on the validation set.

4. Results

Using multiple datasets, we demonstrate how DIFFDOCK-POCKET outperforms baselines for various important pocket-level docking setups that are commonly encountered in real-world applications related to drug discovery. We provide the code for running DIFFDOCK-POCKET in these setups at `https://anonymous.4open.science/r/DiffDock-Pocket-AQ32`.

Table 1. **PDBBind docking performance.** Given a protein structure generated with ESMFold or the bound holo crystal protein structure, the methods are tasked to predict the ligand binding pose and the side chain structure of the protein in the bound state (results in Table 2). Methods that cannot predict side chain conformation changes are marked as *rigid*. All methods, other than DIFFDOCK, operate on the pocket level and have the same pocket definition. The numbers for DIFFDOCK 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

Setup. We use the two datasets: PDBBind 2020 (Liu et al., 2017) and CrossDocked 2020 (Francoeur et al., 2020). For PDBBind, we employ a time-based split and evaluate re-docking to either the bound holo protein structure or docking to computationally generated structures that were aligned to the crystal structure. Protein structures have been generated with ESMFold (Lin et al., 2022) and ColabFold (Mirdita et al., 2022) (compare Appendix E). After training on PDBBind, we also evaluate cross-docking on CrossDocked, which contains protein pocket structures that are bound to various ligands (McNutt et al., 2021). With this, we evaluate the model accuracy when a pocket structure bound to a different ligand is available instead of only having an unbound protein structure available.

Metrics. To evaluate the accuracy of the ligand structure prediction we mainly consider the fraction of predictions of the test set with a root mean square deviation (RMSD) to the ground truth ligand below 2Å. This threshold is commonly used in small molecule docking (Alhossary et al., 2015; Hassan et al., 2017; McNutt et al., 2021). To evaluate side chain structure predictions, we use their RMSD to the side chain structures of the bound (holo) protein structure (SC-RMSD). In the main text, we report the fraction of predictions with SC-RMSD below 1Å and have further thresholds in Appendix F. We further note that computationally generated structures are often considerably different from the ground truth (compare Figure 11), and thus, the best achievable SC-RMSD is typically higher than 0.

When evaluating DIFFDOCK-POCKET, we draw 10 and 40 samples and present metrics for the top-1 prediction, which corresponds to the highest-ranked prediction from the confidence model, as well as for the top-5 predictions, which selects the most accurate pose from the five highest-ranked predictions.

Baselines. We compare our model to the freely available state-of-the-art search-based method GNINA and SMINA which outperform VINA (Koes et al., 2013a) on known binding sites, and the diffusion-based model DIFFDOCK. We omit the blind docking method with receptor flexibility NEURALPLEXER (Qiao et al., 2023) since it is not available as of writing.

Ligand structure prediction results. Table 1 compares the results in terms of ligand structure accuracy. Our approach outperforms both search-based methods and DIFFDOCK in all instances, even when only drawing 10 samples. Only in rigid docking where the bound structure of the protein is presumed to be known, one method, GNINA, performs comparably albeit taking more time. Unlike DIFFDOCK-POCKET, the baselines suffer from a substantial loss in docking accuracy when introducing flexibility.

Furthermore, the baselines become significantly more computationally expensive and slower when predicting side chain changes (unlike DIFFDOCK-POCKET). DIFFDOCK-POCKET's speed advantage can be highly valuable for practitioners performing high-throughput downstream tasks that require accurate ligand structure and pocket structure predictions, which are common in, for instance, drug discovery-related applications.

Side chain prediction results. We then evaluate the accuracy of the predictions of the side chain conformations of the bound protein for protein input structures that are either ESMFold structures or the backbone of the bound protein with randomized side chain torsion angles. The results are visualized in Table 2 and Figure 4. The methods SMINA and GNINA (the only baselines that model side chain flexibility) fail to predict accurate side chains in either case. Meanwhile, DIFFDOCK-POCKET's predictions fall under the 1Å threshold in 33.3% of test cases for ESMFold backbones and 49.2% for bound backbones. Thus DIFFDOCK-POCKET could lead to impactful improve-

Table 2. **PDBBind side chain prediction evaluation.** Given the ESMFold structure of a protein or its bound structure but with randomized side chain torsion angles, the methods are tasked to jointly predict the ligand binding structure and the side chain conformations of the bound protein structure.

	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**

ments for docking applications where the downstream tasks require accurate side chain structure predictions next to ligand structure predictions such as free energy calculations involving molecular dynamics simulations.

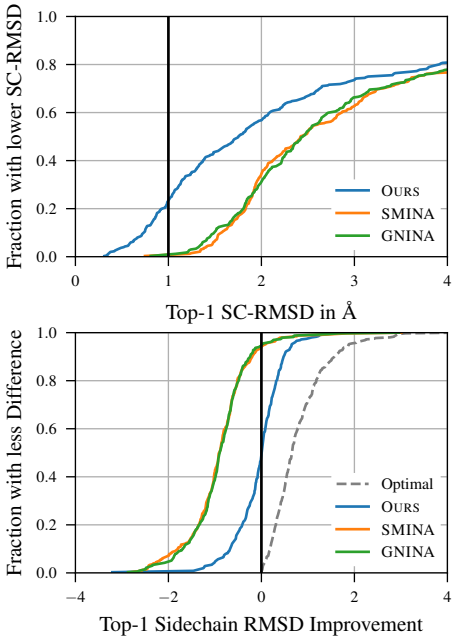

Figure 4. **Quality of predicted side chains for *in-silico* structures.** *Top*: The cumulative distribution function shows how many instances have an SC-RMSD to the holo structure below a certain threshold. *Bottom*: The relative SC-RMSD between the structures before and after the predictions. The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

Energy minimization and physical plausibility. In Table 3, we assess the quality of our predictions on the PDBBind test set by computing the percentage of ligand-protein complexes that pass the PoseBusters (Buttenschoen et al., 2023) checks for ESMFold structures. These checks verify the physical plausibility of complexes and only pass if various properties (no internal steric clashes, no volume overlap, true ligand is within an RMSD of 2Å, ...) are fulfilled. We perform these tests on the predicted complexes, as well as on ligand conformations where we minimized the overall energy inside the pocket. Due to computational restrictions, we limit ourselves to ligands with less than 100 atoms.

In our experiments, we compare the docking success and quality of our method with the baselines of GNINA and DIFFDOCK, the latter of which was reported as the best performing machine learning method by Buttenschoen et al. (2023) regarding the physical plausibility of predictions. We can observe that for deep learning methods, the energy minimization makes structures worse in terms of RMSD to the ground truth. We believe this to be the case because the energy is minimized against computationally generated structures, but the RMSD is measured to the ground truth holo structure. We further see that DIFFDOCK-POCKET with flexibility performed best not only in the success of docking, but also in the quality of predicted structures.

For DIFFDOCK-POCKET, we compare two different relaxation techniques: we start minimizing the energy with the ESMFold protein, and with the predicted structure. When using the predicted structure, the system has a significantly lower median energy (which can most likely be attributed to the penalty in side chain conformer matching), and after pose relaxation a higher percentage of complexes is physically plausible and close to the ground truth. This demonstrates the quality of our predicted protein and the advantage our flexible model has when using apo proteins. This advantage and the successful docking prediction procedure of DIFFDOCK-POCKET is further demonstrated in Figure 5. Especially for downstream tasks, these accurate sidechains can be crucial. Meanwhile, GNINA with flexibility does not show better behavior over rigid docking.

Cross-docking performance. In cross-docking, the input protein structure is obtained from a protein-ligand complex with a different ligand bound to what we aim to dock. We evaluate this task on the CrossDocked 2020 test set without any retraining for cross-docking and removing all proteins in the test set that are in the PDBBind training set (Brocidiacono et al., 2023). Meanwhile, the scoring function of GNINA was fit to the substantially larger CrossDocked 2020 training set.

Table 3. **Pose quality and PoseBusters checks.** We compare the physical plausibility of ligand poses predicted by different methods before and after performing energy minimization of the predicted ligand. As for the structure used for pose relaxation, we have used the predicted flexible proteins, except for methods denoted with *, which use the ESMFold protein. % PoseBusters denotes the percentage of predictions having an RMSD < 2 and passing all of PoseBusters re-docking checks. E denotes the energy of the system.

	Top-1 Pose			Top-1 Pose Relaxed		
Method	% RMSD <2	% PoseBusters	Median E (MJ/mol)	% RMSD <2	% PoseBusters	Median E (MJ/mol)
DIFFDOCK* (blind, rigid)	28.7	3.6	1497	21.0	17.7	117
GNINA* (rigid)	10.5	7.0	**120**	11.8	8.1	**108**
GNINA	7.6	4.2	148	7.2	4.8	140
DIFFDOCK-POCKET* (40)	**51.6**	8.1	1561	36.6	30.7	123
DIFFDOCK-POCKET (40)	**51.6**	**15.1**	305	**41.2**	**36.7**	124

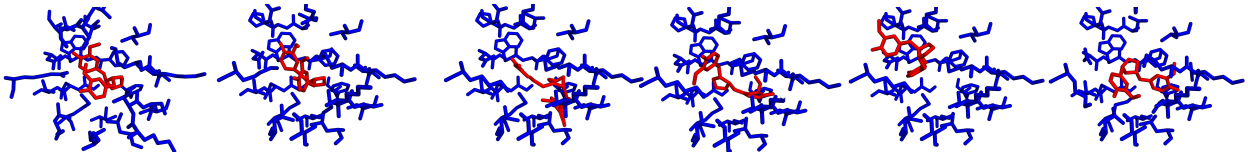

Figure 5. **Predictions and relaxation of 6PYB.** Relaxed ligand poses (red) of different methods are shown. Relaxation is either performed against the predicted protein or the ESMFold protein (marked with *). DIFFDOCK-POCKET visibly outperforms all baselines by a large margin on this complex. Furthermore, the difference caused by relaxing the ligand against the ESMFold-generated protein instead of the predicted structure emphasizes the importance and advantage of including the adjustment of side chain conformations in a docking model.

Since there often are multiple ligands per protein pocket, we follow Brocidiacono et al. (2023) and report metrics averaged first over ligands per pocket and then averaged over pockets in Table 4. Under this metric, DIFFDOCK-POCKET achieves an RMSD of less than 2Å in 28.6% of instances, compared to the second best 24.4% of GNINA.

Table 4. **Cross-docking evaluation on CrossDocked 2020.** Evaluation of the top-1 RMSD between different methods on the CrossDocked 2020 test set. Numbers for the methods marked with a * were taken from Brocidiacono et al. (2023).

	Top-1 RMSD		Average
Method	%<2	%<5	Runtime (s)
VINA*	11.7	40.2	73.7
GNINA*	21.5	51.7	51.6
DIFFDOCK* (blind)	17.3	51.7	98.7
PLANTAIN*	24.4	**73.7**	**4.9**
DIFFDOCK-POCKET (10)	28.3	67.5	22.0
DIFFDOCK-POCKET (40)	**28.6**	67.9	87.2

We note that these results are obtained while a) cross-docked structures were not included in the training data while they were for baselines, and b) the definition of the pocket center was out of distribution for our model. Pockets in CrossDocked are computed by retaining all amino acids that have a heavy atom within 5Å of any crystallized ligand in this pocket. To compute the pocket center required for our approach, we calculate the mean of all C-α atoms in the pocket. If we instead employ our pocket center definition (see Section 3.1) the results of DIFFDOCK-POCKET substantially improve (compare Appendix F.7). Thus, these results could be interpreted as a testament to DIFFDOCK-POCKET's generalization capabilities.

5. Conclusion

We presented DIFFDOCK-POCKET, a diffusion-based generative model to dock small molecules to protein pockets. Compared with prior deep learning approaches, DIFFDOCK-POCKET models the conformational change of side chains which we achieve via a joint diffusion process over side chain torsion angles and the ligand's degrees of freedom. We empirically demonstrate the effectiveness of this approach on multiple datasets and multiple task settings commonly encountered by practitioners. This includes docking to computationally generated protein structures—a task that has emerged in the literature as particularly difficult but important for drug discovery—where DIFFDOCK-POCKET's performance gap to the next best baseline is particularly large, both in accuracy and runtime. Hence, DIFFDOCK-POCKET opens up new capabilities for docking applications with downstream computations that require atomic accurate docking structure prediction, such as binding affinity calculations or molecular dynamics simulations common in drug discovery.

Impact Statement

In this paper, we present a generative molecular docking model that can jointly predict the ligand and protein conformation. The application of molecular docking software had overwhelmingly positive societal impacts in the past—especially in the fields of drug discovery. We believe that our software can aid in the design of novel therapeutic compounds and help to further the field of molecular biology. However, although we believe that this benevolent application will continue in the future, we must acknowledge that such tools could also be used to lower the barriers for entities with malicious intent to develop biological weapons.

References

Alhossary, A., Handoko, S. D., Mu, Y., and Kwoh, C.-K. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics*, 31(13):2214–2216, 02 2015.

Brocidiacono, M., Francoeur, P., Aggarwal, R., Popov, K., Koes, D., and Tropsha, A. BigBind: Learning from nonstructural data for structure-based virtual screening, November 2022.

Brocidiacono, M., Popov, K. I., Koes, D. R., and Tropsha, A. Plantain: Diffusion-inspired pose score minimization for fast and accurate molecular docking. In *Workshop on Computational Biology*, 2023.

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

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

Corso, G. Modeling molecular structures with intrinsic diffusion models. *arXiv preprint arXiv:2302.12255*, 2023.

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. DiffDock: Diffusion steps, twists, and turns for molecular docking. In *International Conference on Learning Representations*, 2023.

De Bortoli, V., Mathieu, E., Hutchinson, M., Thornton, J., Teh, Y. W., and Doucet, A. Riemannian score-based generative modelling, 2022.

Francoeur, P. G., Masuda, T., Sunseri, J., Jia, A., Iovanisci, R. B., Snyder, I., and Koes, D. R. 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.

Friesner, R. A., Banks, J. L., Murphy, R. B., Halgren, T. A., Klicic, J. J., Mainz, D. T., Repasky, M. P., Knoll, E. H., Shelley, M., Perry, J. K., Shaw, D. E., Francis, P., and Shenkin, P. S. 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.

Geiger, M., Smidt, T., M., A., Miller, B. K., Boomsma, W., Dice, B., Lapchevskyi, K., Weiler, M., Tyszkiewicz, M., Batzner, S., Madisetti, D., Uhrin, M., Frellsen, J., Jung, N., Sanborn, S., Wen, M., Rackers, J., Rød, M., and Bailey, M. Euclidean neural networks: e3nn, April 2022.

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

Hogues, H., Gaudreault, F., Corbeil, C. R., Deprez, C., Sulea, T., and Purisima, E. O. ProPOSE: Direct exhaustive protein–protein docking with side chain flexibility. *Journal of Chemical Theory and Computation*, 14(9): 4938–4947, August 2018.

Ingraham, J., Baranov, M., Costello, Z., Frappier, V., Ismail, A., Tie, S., Wang, W., Xue, V., Obermeyer, F., Beam, A., and Grigoryan, G. Illuminating protein space with a programmable generative model, 2022.

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

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, July 2021.

Karelina, M., Noh, J. J., and Dror, R. O. How accurately can one predict drug binding modes using alphafold models? *bioRxiv*, pp. 2023–05, 2023.

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

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

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

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Liu, Z., Su, M., Han, L., Liu, J., Yang, Q., Li, Y., and Wang, R. Forging the basis for developing protein-ligand interaction scoring functions. *Accounts of Chemical Research*, 50(2):302–309, February 2017.

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

Lu, W., Wu, Q., Zhang, J., Rao, J., Li, C., and Zheng, S. TANKBind: Trigonometry-aware neural networks for drug-protein binding structure prediction. In *Advances in Neural Information Processing Systems*, 2022.

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

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

McPartlon, M. and Xu, J. Deep learning for flexible and site-specific protein docking and design. *bioRxiv*, 2023.

Meli, R., Anighoro, A., Bodkin, M. J., Morris, G. M., and Biggin, P. C. Learning protein-ligand binding affinity with atomic environment vectors. *Journal of Cheminformatics*, 13(1):59, August 2021.

Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S., and Steinegger, M. ColabFold: making protein folding accessible to all. *Nature Methods*, 19(6):679–682, May 2022.

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

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

Qiao, Z., Nie, W., Vahdat, A., au2, T. F. M. I., and Anandkumar, A. State-specific protein-ligand complex structure prediction with a multi-scale deep generative model, 2023.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

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

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

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds, 2018.

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

Trott, O. and Olson, A. J. AutoDock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

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

Wong, F., Krishnan, A., Zheng, E. J., Stärk, H., Manson, A. L., Earl, A. M., Jaakkola, T., and Collins, J. J. Benchmarking alphafold enabled molecular docking predictions for antibiotic discovery. *Molecular Systems Biology*, 2022.

Yu, Y., Lu, S., Gao, Z., Zheng, H., and Ke, G. Do deep learning models really outperform traditional approaches in molecular docking? In *Workshop on Machine Learning for Drug Discovery*, 2023.

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

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. We can assess the reduction by counting the degrees of freedom of translations on the ligand and flexible side chains as a function of their number of atoms. Side chains 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. 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).

B.1. Reducing Steric Clashes in Protein Side Chain Alignment

To train our flexible model, we align the side chains 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 side chain 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) is

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

s and l are the positions of atoms of the side chains and ligands respectively.

We calculate the pairwise distances between the ligand and side chain 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 5. 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 5. **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
ESMFold structures	76.7	19.1
Conformer-Matched	68.3	15.4
Conformer-Matched w/ penalty	67.7	13.9

B.2. Model Results

Given this definition of steric clashes, we can evaluate the different models, as done in Table 6. 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 6. **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

C. Model Details

C.1. 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.

C.2. 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 side chains 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 side chain atom positions. The model is then trained with a binary cross-entropy loss. No diffusion is involved in the training of the confidence model.

C.3. Limiting Diffusion to the Binding Pocket

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. Various ways exist to condition the model to a known binding pocket, depending on the underlying method used. 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 maximum translation our model can predict. With this change, all ligand poses are guaranteed to be within the target pocket, but the model still needs to predict a (small) translation to account for the random noise and different poses. Formally, the random ligand translation z_{tr} will be sampled from a normal distribution with a relatively small variance. This will have no effect on the initially random rotation and torsion angles.

However, for large proteins, this would still mean that our approach needs to consider atoms far away, although the atoms close to the binding site influence the actual binding procedure most. By exploiting this fact, we decided to discard all amino acids that are too far away from the target binding site, as depicted in Figure 3. 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). This allows our model to learn more physics-informed predictions.

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 or wrong side chain structure.

C.4. Side Chain 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

$$\begin{aligned} \min(lig_x) - r < x < \max(lig_x) + r \\ \min(lig_y) - r < y < \max(lig_y) + r \\ \min(lig_z) - r < z < \max(lig_z) + r, \end{aligned} \tag{3}$$

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 side chains 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 side chains of the protein, we always rotate the side that does not contain the protein backbone. This simple convention makes the update of the side chain'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 side chain.

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

C.5. Training and Inference of the Score Model

We use ESMFold 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 side chain positions. As such, there is no possibility for data leakage as the model is neither aware of the ground truth side chain positions, nor which side chains 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 $= res = \mathbf{y}_k : \exists$atom $= a \in res, \|a - pc\| <$ r$\}$;
Randomize ligand and side chains 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 $\mathbf{x}_0, \mathbf{y}_0$;

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 side chains 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 7.

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 7. **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. 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 ESMFold 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 8, Table 9, and Table 10. We can see that the results are similar to those

from ESMFold, letting us conclude that the model generalizes to well.

Table 8. **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 9. **Top-1 PDBBind docking with ColabFold structures.** More detailed performance evaluation when docking to *in-silico* structures generated by ColabFold.

	Ligand RMSD					Side Chain 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 10. **PDBBind side chain performance with ColabFold structures.** Evaluating the performance of the side chains 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**

F. Additional Results

F.1. Further Docking Results

We have compiled a list of tables and figures that allow further evaluation of the docking results. In Table 11 and Table 12, we illustrate the different percentiles of our predictions for the ligand and side chain 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 13. Figure 6 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 side chain accuracy. Figure 7 illustrates the fraction of predictions with a lower side chain RMSD for crystal structures and ESMFold structures respectively. Since

the side chains of ESMFold structures cannot be aligned completely to the crystal structures by only changing the torsional angles, Figure 8 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 11. **Top-1 PDBBind crystal docking.** A more detailed performance evaluation of docking with holo crystal structures.

	Ligand RMSD					Side Chain 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 12. **Top-1 PDBBind ESMFold docking.** A more detailed performance evaluation of docking with computationally generated ESMFold structures.

	Ligand RMSD					Side Chain 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 13. **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

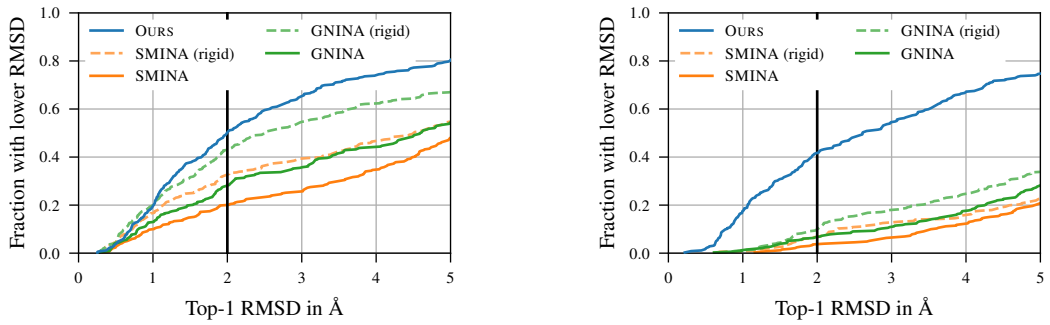

Figure 6. **Cumulative distribution function of RMSD.** *Left*: The CDF when using crystal structures as input. *Right*: The CDF when using ESMFold structures as input.

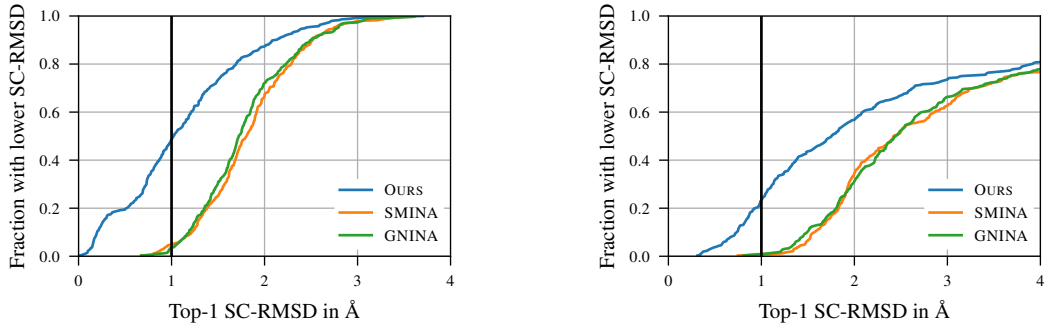

Figure 7. **Cumulative distribution function of SC-RMSD.** *Left*: The CDF when using crystal structures as input. *Right*: The CDF when using ESMFold structures as input.

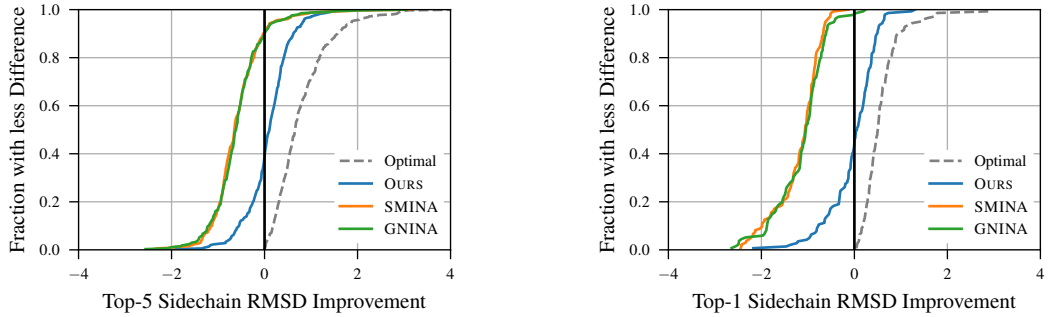

Figure 8. **Relative side chain improvements on ESMFold structures.** *Left*: The relative side chain improvement, when picking the top-5 side chain prediction. *Right*: The relative side chain improvement only for ESMFold complexes that have a pocket RMSD of < 1.5Å.

F.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 14 this comparison is illustrated. We further added a comparison for when we use the flexible model, but do not predict the pose of any side chain positions during training.

From this, we can see that training with flexibility improves the docking accuracy, especially for proteins where the true side chain 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 14. **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

F.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 15. 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 16).

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

Protein	Top-1 Ligand RMSD in Å								
Structure	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 16. **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	Pocket RMSD in Å								
Structure	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**

F.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 9 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 10, 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.

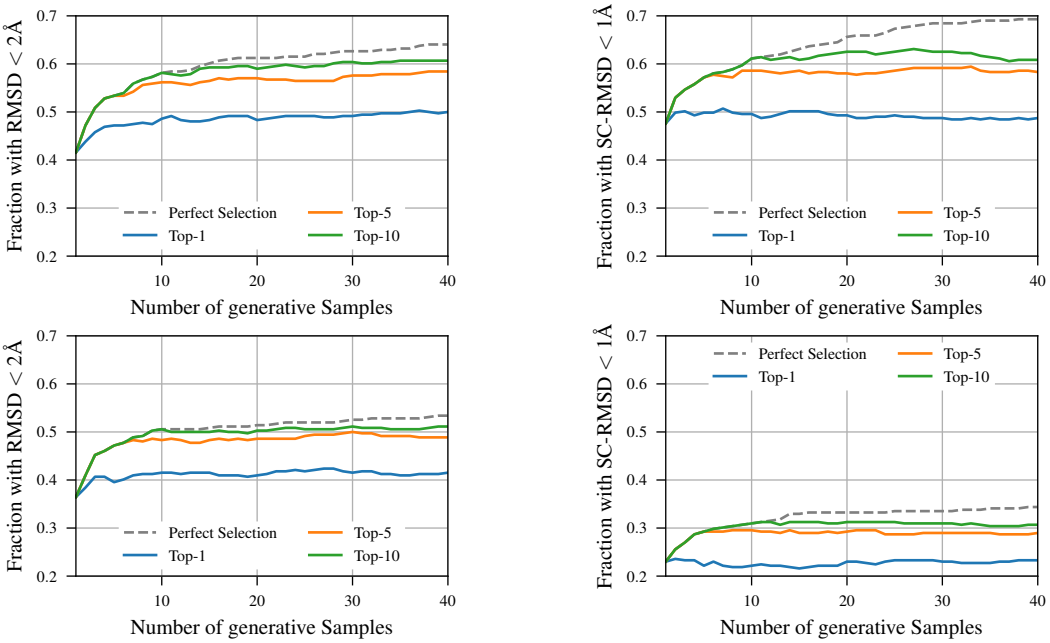

Figure 9. **Performance based on number of generative samples.** Compare the top-1, top-5, and top-10 accuracy based on the number of samples generated by our procedure. In *left*, the RMSD of the ligand can be seen, whereas *right*, the side chain RMSD is illustrated. In the *top* row, the input are crystal structures, while the *bottom* row uses structures generated by ESMFold.

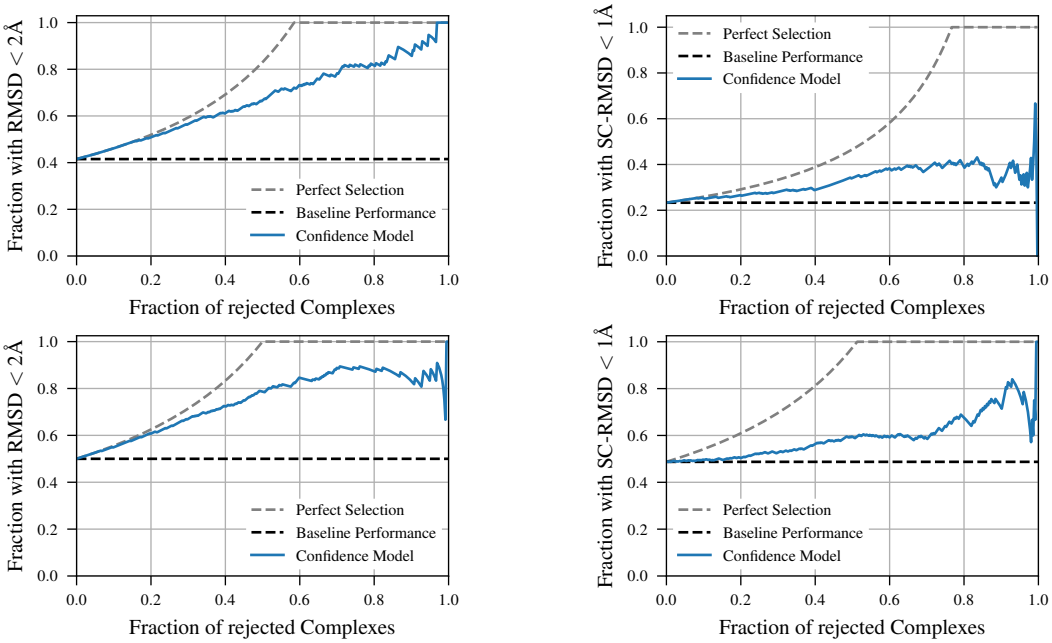

Figure 10. **Selective accuracy of the score-model.** Compare the performance of the model with respect to the confidence model, and a perfect selection. In *left*, the RMSD of the ligand can be seen, whereas *right*, the side chain RMSD is illustrated. In the *top* row, the input are crystal structures, while the *bottom* row uses structures generated by ESMFold.

F.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 11 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 12 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 side chains, the prediction quality of our model strongly depends on the quality of the computationally generated structure.

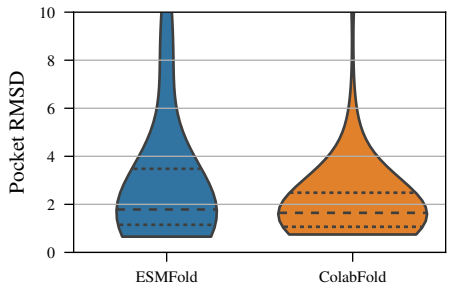

Figure 11. **Pocket RMSD between apo and holo structures.** Apo ESMFold and ColabFold structures have been aligned with the holo crystal structures such that the RMSD in the pocket is the lowest. This figure shows the RMSD of the pocket for proteins in the test set. The dashed lines represent the 25%, 50%, and 75% percentiles respectively. This figure does not show outliers having an RMSD larger than 10Å.

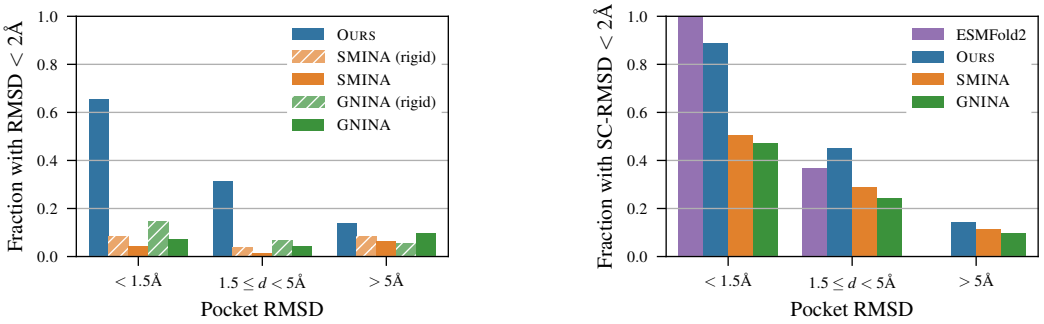

Figure 12. **Model accuracy based on quality of ESMFold predictions.** Comparison of the model accuracy with three different levels of the quality of ESMFold predictions. The predicted ligand (*left*) and side chain quality (*right*) are evaluated respectively.

F.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 13. 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%.

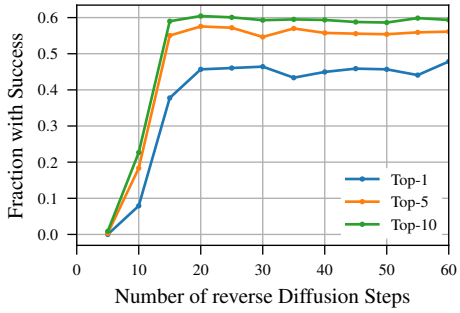

Figure 13. **Comparison of the number of reverse diffusion steps.** Results of the inference with different reverse diffusion steps on the validation set. The values on the y-axis shows the fraction of samples where the RMSD is < 2Å and the SC-RMSD is < 1Å.

F.7. Impact of Pockets for 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. In Table 17, we see how different pockets influence the results of the performance of our model in the cross-docking benchmark. For this testset, we present the numbers for three different choices of pockets.

1. 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 *.
2. 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 †.
3. 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 were presented in the main paper.

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.

Table 17. **Cross-docking performance on CrossDocked 2020 with different pockets.** In this table, we present additional results for the cross-docking benchmarks when using different pockets. The method highlighted with * follows our pocket definition presented with access to the ground truth data to compute the pockets as in training. For the results marked with a †, we use identical pocket centers as presented in Brocidiacono et al. (2023).

Method	Top-1 RMSD %<2	Top-1 RMSD %<5	Average Runtime (s)
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

G. Visualization of Docking Results

We present the visualization for four different dockings in Figure 14. An animation of the docking process for multiple complexes can be found in our repository at `https://anonymous.4open.science/r/DiffDock-Pocket-AQ32`.

H. 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

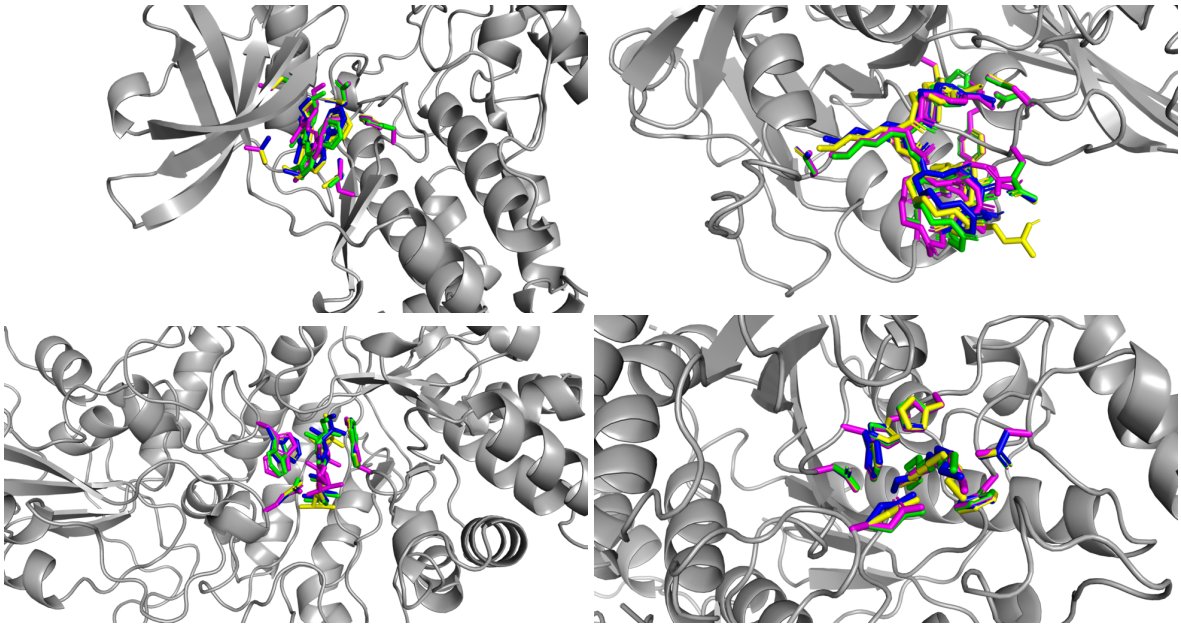

Figure 14. **Flexible docking of unseen complexes.** Visualization of the results of four dockings on arbitrarily selected complexes (*top*: 6a1c, 6hzb, *bottom:* 6md6, 6uii). Four different poses for the side chains and ligand are presented in different colors.

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 re-docking 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 checks 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 18.

Table 18. **Results of PoseBusters quality check without energy minimization.** We report the percentage of predictions that pass all Posebusters 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)	33.4	**15.1**	45.6	33.0

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 both the percentage of generated structures that pass all PoseBusters quality checks and the percentage of structures that are also considered to be a successful re-docking attempt is higher for DIFFDOCK-POCKET, nearly doubling the percentage of good predictions in both cases compared to DIFFDOCK, the best machine learning method reported by Buttenschoen et al. (2023).

In Table 18, 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 side chain positions. However, although more than 90% of generated structures by GNINA

are considered correct, DIFFDOCK-POCKET outperforms all methods when considering successful re-docking (even before energy minimization). This suggests an advantage DIFFDOCK-POCKET could have over classical approaches when docking to apo structures. This becomes more clear in Table 3.

Altogether, we can report that on the PDBBind testset DIFFDOCK-POCKET outperforms DIFFDOCK on all measured metrics on both datasets and outperforms all considered methods on re-docking to ESMFold structures.