DIFFDOCK-PP: RIGID PROTEIN-PROTEIN DOCKING WITH DIFFUSION MODELS

Mohamed Amine Ketata[*1], **Cedrik Laue**[*1], **Ruslan Mammadov**[*1], **Hannes Stärk**[2], **Menghua Wu**[2], **Gabriele Corso**[2], **Céline Marquet**[1], **Regina Barzilay**[2], **Tommi S. Jaakkola**[2]

[1]Technical University of Munich, Germany [2]Massachusetts Institute of Technology, USA
{mohamedamine.ketata,cedrik.laue,ruslan.mammadov}@tum.de

ABSTRACT

Understanding how proteins structurally interact is crucial to modern biology, with applications in drug discovery and protein design. Recent machine learning methods have formulated protein-small molecule docking as a generative problem with significant performance boosts over both traditional and deep learning baselines. In this work, we propose a similar approach for rigid protein-protein docking: DIFFDOCK-PP is a diffusion generative model that learns to translate and rotate unbound protein structures into their bound conformations. We achieve state-of-the-art performance on DIPS with a median C-RMSD of 4.85, outperforming all considered baselines. Additionally, DIFFDOCK-PP is faster than all search-based methods and generates reliable confidence estimates for its predictions.[1]

1 INTRODUCTION

Proteins realize their myriad biological functions through interactions with biomolecules, such as other proteins, nucleic acids, or small molecules. The presence or absence of such interactions is dictated in part by the geometric and chemical complementarity of participating bodies. Thus, learning how individual proteins form complexes is crucial to understanding protein activity. In this work, we focus on rigid protein-protein docking: given two protein structures, the goal is to predict their resultant complex while maintaining internal bonds, angles, and torsion angles fixed.

Traditional approaches for rigid protein-protein docking consist of a search algorithm followed by a scoring function (Chen et al., 2003; De Vries et al., 2010; Yan et al., 2020). After enumerating a vast search space of potential poses, these methods rely on heuristics or empirical methods to select the most plausible poses. Due to the exhaustive search required, these methods are often slow and computationally expensive. More recently, deep learning approaches have tackled protein-protein docking as a regression problem: given two structures, directly predict the final pose (Ganea et al., 2021; Jamasb et al., 2021). While fast, these models have yet to outperform search-based algorithms.

Inspired by recent breakthroughs in protein-small molecule docking (Corso et al., 2022), we instead propose that protein-protein docking be formulated as a generative problem: given two proteins, the goal is to estimate the distribution over all potential poses using a diffusion generative model. To obtain the final docked pose, we sample from this distribution multiple times and select the best one via a learned confidence model, as shown in Figure 1. We call our method DIFFDOCK-PP.

Empirically, DIFFDOCK-PP achieves a top-1 median complex root mean square deviation (C-RMSD) of 4.85 on the Database of Interacting Protein Structures (DIPS), outperforming all considered baselines. Compared to popular search-based docking software, DIFFDOCK-PP is 5 to 60 times faster on GPU.

[*]Equal contribution
[1]Our code is publicly available at https://github.com/ketatam/DiffDock-PP

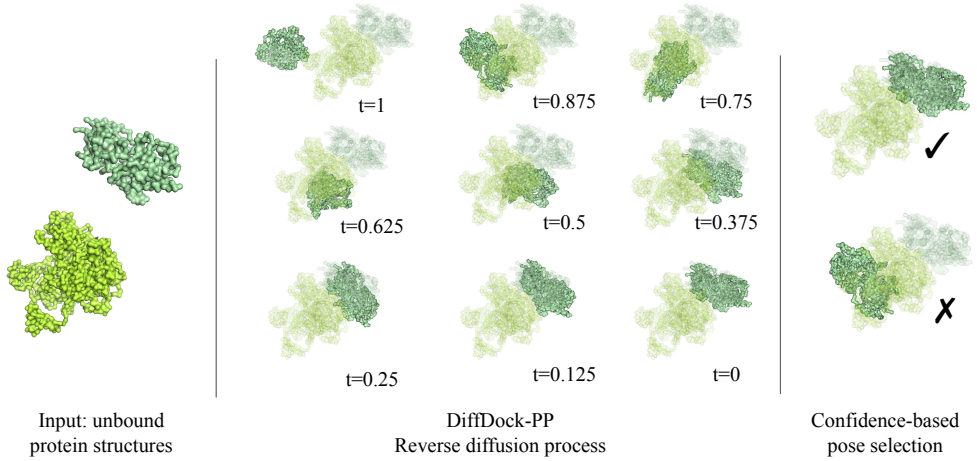

Figure 1: Overview of DIFFDOCK-PP. The model takes two proteins as input, where the ligand has been randomly rotated and translated in 3D space. Then it runs a reverse diffusion process to sample multiple poses. The confidence model ranks these poses, and we output the pose with the highest confidence score. The ground truth pose is shown in light grey and is added to all diffusion steps. Depicted structure is PDB 1KEE over 40 steps of reverse diffusion.

2 BACKGROUND AND RELATED WORK

Protein-Protein Docking. The goal of protein-protein docking is to predict the (bound) structure of a protein-protein complex based on the individual proteins' (unbound) structures. Specifically, we focus on the task of *rigid body* protein-protein docking, which assumes that the proteins do not undergo any deformations during binding, restricting their relative degrees of freedom to a rotation and translation in 3D space. This assumption is often realistic (Vakser, 2014) and even leads to improved results for most interacting proteins (Desta et al., 2020).

To evaluate the quality of the predicted structures, a common approach is to compute the fraction of those lying within some threshold distance to the true bound structure (Vakser, 2014; Basu & Wallner, 2016; Lensink et al., 2007).

Search-based Docking Methods. Traditional methods for protein-protein docking usually rely on the physical properties of the complexes (Chen et al., 2003; De Vries et al., 2010; Yan et al., 2020). These methods typically 1) generate an initial population of plausible complex structures, 2) further pose proposals using optimization algorithms, and 3) refine the complexes with the highest score according to some scoring function. Template-based modeling (TBM), which predicts the structure of a target protein by aligning it to one or multiple template proteins with known structures, is also used as a subroutine by some search-based methods (Vakser, 2014). While some of these methods offer decent predictive performance, they are usually computationally expensive and impractical for large-scale molecular screening campaigns.

Deep Learning-based Docking Methods. Deep learning approaches to protein-protein docking can be broadly partitioned into two categories: single-step and multi-step methods. Single-step methods directly predict the complex structure in a one-shot fashion. Notably, Ganea et al. (2021) proposed EQUIDOCK, a pairwise-independent SE(3)-equivariant graph matching network that directly predicts the relative rigid-body transformation of one of the interacting proteins. Furthermore, Sverrisson et al. (2022) incorporated different physical priors into an energy-based model for predicting protein complex 3D structure. In contrast, multi-step methods produce their final predictions by iteratively refining a set of proposed structures. For instance, ALPHAFOLD-MULTIMER (Evans et al., 2021) was designed to co-fold multiple protein structures, given their primary sequences and multiple sequence alignments (MSAs) to evolutionary-related proteins. In parallel with this work, McPartlon & Xu (2023) proposed DOCKGPT, a generative protein transformer for flexible and site-specific protein docking.

Our method, DIFFDOCK-PP, naturally falls into the category of multi-step methods due to the multiple steps required to sample from the distribution induced by diffusion generative models. Compared to search-based methods, however, we sample orders of magnitude fewer poses during our refinement process.

Diffusion Generative Models (DGMs). DGMs offer a powerful way to represent probability distributions beyond likelihood-based and implicit generative models, circumventing many of their issues. The main idea is to define a diffusion process transforming the data distribution in a tractable prior and learn the *score function*, which is the *gradient* of the log probability density function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$[2], of this evolving distribution. We can then use the learned score function to sample from the underlying probability distribution using well-established algorithms (Song et al., 2020). A plethora of DGMs have been developed for tasks in computational biology, including conformer generation (Jing et al., 2022), molecule generation (Hoogeboom et al., 2022), and protein design (Trippe et al., 2022). The foundation of our method is DIFFDOCK (Corso et al., 2022), a diffusion generative model over the product space of the ligand's degrees of freedom (translational, rotational, and torsional). We extend this approach to the protein docking task.

3 METHOD

3.1 BENEFITS OF GENERATIVE MODELING FOR RIGID PROTEIN DOCKING

Protein-protein docking is often evaluated on the basis of thresholding (Basu & Wallner, 2016; Lensink et al., 2007), e.g., a Ligand-RMSD < 5 Å and an Interface-RMSD < 2 Å are among several conditions to consider a given prediction to be of medium quality in Lensink et al. (2007).

Following the arguments of Corso et al. (2022) and noting that directly optimizing such thresholding-based objectives is not feasible because they are not differentiable, we argue that these objectives are better aligned with training a *generative* model to maximize the likelihood of the observed structures than with fitting a *regression* model as done in previous work. Concretely, since real-world data, as well as complex deep learning models, suffer from inherent multi-modal uncertainty, regression-based methods trained to predict a single pose that, in expectation, minimizes some MSE-type loss (Ganea et al., 2021) would learn to predict a structure as the weighted mean among many viable alternatives, often not a plausible structure itself. In contrast, a generative model would aim to capture the distribution over these alternatives resulting in more plausible, accurate, and diverse structures.

To illustrate this phenomenon, we visualize some of the structures predicted by our model and compare them to those generated by the baselines, especially EQUIDOCK, which was trained using an MSE-type loss and whose one of the main limitations is the existence of steric clashes in its predicted structures (Ganea et al., 2021). Figure 2 illustrates such predictions. We observe that our model predicts structures with no steric clashes, which we hypothesize is in part due to the adopted generative approach to protein-protein docking.

3.2 METHOD OVERVIEW

As defined in Section 2, in rigid protein-protein docking, we aim to predict the complex structure of an interacting protein pair based on the individual structure of each protein.

In this work, we model the proteins on the residue level, representing each protein as a set of amino acid nodes. Each residue is, in turn, represented by its type and the position of its α-carbon atom. We denote $\mathbf{X}_1 \in \mathbb{R}^{3n}$ as the ligand consisting of n residues and $\mathbf{X}_2 \in \mathbb{R}^{3m}$ as the receptor with m residues. The ligand/receptor assignment can, in principle, be arbitrary; however, we define the ligand $\mathbf{X}_1$ as the protein with fewer residues. This means that with $\mathbf{X}_1^* \in \mathbb{R}^{3n}$ and $\mathbf{X}_2^* \in \mathbb{R}^{3m}$ denoting the ground truth complex, the receptor is kept fixed $\mathbf{X}_2 = \mathbf{X}_2^*$ and the task is to predict the structure of the ligand with respect to the receptor.

It is important to note that - since we are considering *rigid-body* protein docking - we only need to consider poses that can be obtained by a *rigid-body* transformation (i.e., a rotation and a translation) of the initial pose $\mathbf{X}_1$. These poses lie on a 6-dimensional submanifold $\mathcal{M} \subset \mathbb{R}^{3n}$ corresponding

[2]Note that this is a very different concept from the *scoring function* of search-based docking methods.

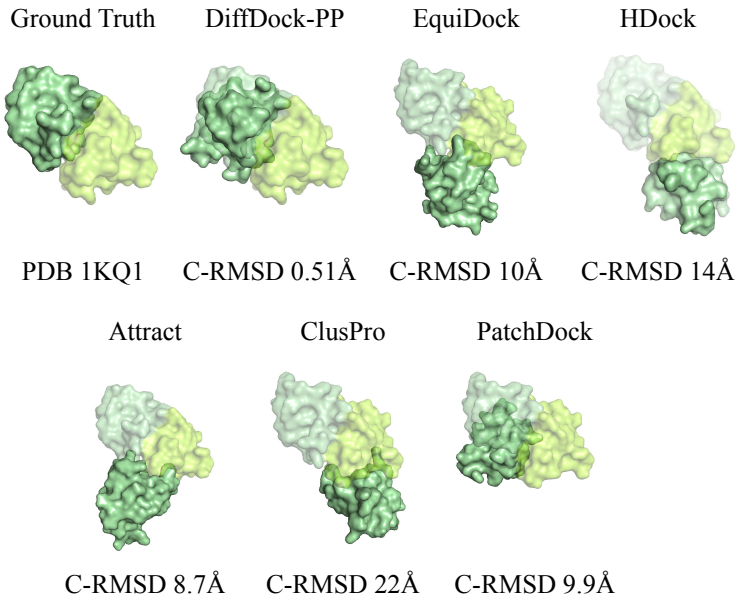

Figure 2: Visualization of the different methods' predictions for PDB 1KQ1 from our test set. The ground truth pose is depicted in light grey.

to the 6 degrees of freedom introduced by the rotation and translation in 3D space. We consider rigid-body protein-protein docking as the task of learning a probability distribution $p(\mathbf{X}_1|\mathbf{X}_2)$ of ligand poses in $\mathcal{M}$, conditioned on the receptor structure $\mathbf{X}_2$.

Now we discuss how we can effectively deploy DGMs to learn this probability distribution. In order to avoid the inefficiencies arising from learning DGMs on arbitrary submanifolds (De Bortoli et al., 2022), we use the framework of intrinsic diffusion models Corso (2023): map the extrinsic submanifold $\mathcal{M}$ to the intrinsic manifold defined by the product space of the rotation and translation group and define our DGM over this manifold.

3.3 Diffusion Process

To formalize the discussion in the previous section, let us introduce the 3D translation group $\mathbb{T}(3)$ and the 3D rotation group $SO(3)$ as well as their product space $\mathbb{P} = \mathbb{T}(3) \times SO(3)$. With this, we can define a rigid-body transformation as the mapping $A : \mathbb{P} \times \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ with

$$A((\boldsymbol{r}, R), \boldsymbol{x})_i = R\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right) + \overline{\boldsymbol{x}} + \boldsymbol{r}, \tag{1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^3$ corresponds to the position of the i-th residue and $\overline{\boldsymbol{x}}$ represents the center of mass of the ligand protein. This equation simply describes a rotation around the center of mass followed by a translation.

The submanifold of ligand poses introduced informally in the previous section can now be described using this transformation as $\mathcal{M} = \{A\left((\boldsymbol{r}, R), \mathbf{X}_1\right) | \, (\boldsymbol{r}, R) \in \mathbb{P}\}$. For similar arguments to Corso et al. (2022), the map $A\left(\cdot, \mathbf{X}_1\right) : \mathbb{P} \to \mathcal{M}$ is a bijection, which guarantees the existence of the inverse map. We can therefore develop a diffusion process over the product space $\mathbb{P}$ to generate a distribution on the manifold.

Given that $\mathbb{P}$ is a product manifold, we can define a forward diffusion process independently on each manifold (Rodolà et al., 2019) with the score as an element of the corresponding tangent space (De Bortoli et al., 2022). A score model can then be trained with denoising score matching (Song & Ermon, 2019). In both groups, we define the forward SDE as $d\mathbf{x} = \sqrt{d\sigma^2\left(t\right)/dt}\; d\mathbf{w}$, where σ^2 is σ^2_{tr} for $\mathbb{T}(3)$ and σ^2_{rot} for $SO(3)$ and $d\mathbf{w}$ denotes the corresponding Brownian motion. The reader is referred to Corso et al. (2022) for a description of how we can sample from and compute the score of the diffusion kernel on each of these groups.

3.4 MODEL ARCHITECTURE

Both the score and confidence models are based on $SE(3)$-equivariant convolutional networks (Thomas et al., 2018; Geiger et al., 2020) adapted from DIFFDOCK's architecture (Corso et al., 2022) to mainly account for: i.) the *symmetry* of protein-protein pairs, and ii.) the *rigidity* assumption of the proteins. Below we summarise the main components of the architecture.

Input representation. Protein structures are represented as heterogeneous geometric graphs with the amino acid residues as nodes. Node features comprise the residue's type, the positions of its α-carbon atoms, as well as language model embeddings trained on protein sequences from ESM2 (Lin et al., 2022). In order to construct the edges, we connect each node to its 20 nearest neighbors from the same protein (intra-edges), and we use a dynamic cutoff distance of (40 + 3 * σ_{tr})Å, where σ_{tr} is the current standard deviation of the diffusion translational noise to connect the nodes from different proteins (cross-edges). The intuition behind using a dynamic cutoff distance is to increase the chances that each node interacts with potentially relevant nodes from the other protein even when the proteins are still far apart (at early diffusion steps) while having a lower computational cost than using a fixed higher cutoff distance (especially at later diffusion steps).

Intermediate layers. After an initial set of embedding layers to process the initial features, the diffusion time, and the edge lengths, we define a different set of convolutional layers for each edge type (intra-edges and cross-edges). However, in contrast to Corso et al. (2022), we use the same intra-edge layers for both proteins to account for symmetry.

Output layers. This is where the main difference between the score and confidence model lies. On the one hand, the score model applies a tensor-product convolution placed at the center of mass of the ligand to produce two $SE(3)$-equivariant 3-dimensional vectors as the translational and rotational scores (lying in the tangent space of the respective manifolds). On the other hand, the confidence model applies a fully connected layer on the mean-pooled scalar representations from the last convolution layer to produce the $SE(3)$-invariant confidence value.

3.5 TRAINING AND INFERENCE

The training and inference regimes follow very closely Corso et al. (2022). We reiterate the most important points here.

Diffusion model. Even though the diffusion kernel and score matching objectives were defined on the product space $\mathbb{P}$, we follow the extrinsic-to-intrinsic framework (Jing et al., 2022) and develop the training and inference procedures directly on ligand-receptor poses in 3D space. This allows the model to reason about physical interactions more easily and should lead to better generalization. Another interesting point to note is that each training example $(\mathbf{X}_1, \mathbf{X}_2)$ is the only available sample from the conditional distribution $p(\cdot|\mathbf{X}_2)$. This is unlike the standard generative modeling setting, where many samples are drawn from the same data distribution. Therefore, during training, we iterate over distinct conditional distributions with only one sample. During inference, in order to avoid the problem of overdispersed distributions typically observed with generative models, we use low-temperature sampling (Ingraham et al., 2022), which allows the model to concentrate on modes with high likelihood.

Confidence model. The confidence model is a simple classification network trained to predict whether the structures sampled from the score model are of "good" quality, which we define as the structure having an L-RMSD below a certain threshold. To this end, we collect the training data for the confidence model by sampling from the (trained) diffusion model multiple times for each training complex and computing the L-RMSD for the sampled complexes. The labels are then generated by simply comparing the L-RMSD values to the threshold. In our experiments, we set the threshold to be 5Å. The confidence model is then trained with cross-entropy loss.

Combined Inference. During inference, we sample a set of candidate poses from the diffusion model. These samples are then ranked by the confidence model according to the predicted confidence value of whether each pose has an L-RSMD below 5Å. The final prediction is the pose with the highest confidence score.

4 EXPERIMENTAL SETUP

We evaluate our model on the Database of Interacting Protein Structures (DIPS) (Townshend et al., 2019). DIPS consists of 42,826 binary protein complexes. We use the same protein-family-based dataset split proposed by Ganea et al. (2021).

DIFFDOCK-PP has 1.62M parameters and was trained on the DIPS training set for 170 epochs. Every 10 epochs, we run reverse diffusion on the DIPS validation set to compute L-RMSD values. The best model obtained with this procedure is finally tested on the DIPS test set. During training and inference, the smaller protein is selected as a ligand, and we randomly rotate and translate the ligand in space before running our model.

We compare our method to search-based docking algorithms CLUSPRO (PIPER) (FFT-based) (Desta et al., 2020; Kozakov et al., 2017), ATTRACT (based on a coarse-grained force field) (Schindler et al., 2017; de Vries et al., 2015), PATCHDOCK (based on shape complementarity principles) (Mashiach et al., 2010; Schneidman-Duhovny et al., 2005), and HDOCK (makes use of template-based modeling and ab initio free docking) (Yan et al., 2020; 2017; Huang & Zou, 2014; 2008). For these baselines, we cannot control their training and testing data. This implies that some of them might have used a part of our test set to train or validate their models. Thus, the reported performance of these methods might be overestimating the true performance.

Furthermore, we compare our method to deep learning baselines ALPHAFOLD-MULTIMER (Evans et al., 2021) and EQUIDOCK (Ganea et al., 2021). Details on how we evaluated these baselines can be found in Appendix A.

To ensure a fair comparison, we follow the evaluation scheme proposed by Ganea et al. (2021). All models were evaluated using complex root mean square deviation (C-RMSD) and interface root mean square deviation (I-RMSD). C-RMSD is determined by superimposing the ground truth and predicted complex structures via the Kabsch algorithm (Kabsch, 1976) and computing the RMSD between all C-α coordinates. I-RMSD is determined by similarly aligning the interface residues of both complexes and computing the RMSD over interface C-α coordinates (within 8Å of the binding partner).

5 RESULTS

Table 1 reports the performances of the different methods on the DIPS test set. DIFFDOCK-PP achieves a C-RMSD median of 4.85, in line with HDOCK and significantly outperforming all other baselines. This performance is further confirmed when looking at the percentage of predictions below a specific threshold with DIFFDOCK-PP achieving respectively 42% and 45% of C-RMSD and I-RMSD below 2 Å.

When limited to generating one sample for each complex, DIFFDOCK-PP still outperforms the majority of the baselines while having a significantly lower runtime than the search-based methods. Ensuring computational efficiency without a significant drop in performance is critical for computational screening applications like drug discovery and antibody design, where one needs to analyze a very large number of complexes.

We note that when we evaluate the model in such a way that we choose the complex with the smallest RMSD from the ones generated by the model, the performance exceeds that of all baselines by a large margin. As such, the performance of DIFFDOCK-PP can be significantly boosted by improving the used confidence model or designing more effective ranking methods. Additionally, this regime can be particularly beneficial for applications where it is desirable to have at least one high-quality recommendation among the predicted suggestions, e.g., if a practitioner is interested in discovering a single good structure based on prior knowledge from a set of proposals where the diversity of the proposals is important.

Figure 3 shows, for all the considered methods, the fraction of predictions having a C-RMSD value below different thresholds ranging from 0 to 10. DIFFDOCK-PP outperforms most baselines for all threshold values, and outperforms HDOCK for thresholds higher than 5.

To evaluate how good our confidence model is, we plot in Figure 4-left the top-1 performance for different numbers of generated samples (this is done by picking the sample with the highest confi-

Table 1: **Results on 100 samples from the DIPS test set.** The last three rows show our method's performance. The number of poses sampled from the diffusion model is in parentheses, and oracle refers to the setting where we can perfectly select the best pose out of the sampled ones. The methods highlighted with * do not use the same training data as our models and might be using parts of our test sets (e.g., to extract templates or as training examples) or have seen proteins more similar to our test set than the training set. Runtimes with † denote that the method can only run on CPU.

	DIPS Test Set								
	Complex RMSD (Å)				Interface RMSD (Å)				Runtime (s)
Methods	%<2	%<5	%<10	Median	%<2	%<5	%<10	Median	Mean
ATTRACT*	20	23	33	17.17	20	22	38	12.41	1285†
HDOCK*	**50**	**50**	50	6.23	**50**	50	58	**3.90**	778†
CLUSPRO*	12	27	35	15.77	21	27	42	12.54	10475†
PATCHDOCK*	31	32	36	15.25	32	32	42	11.45	7378†
ALPHAFOLD-MULTIMER*	39	45	52	8.61	45	47	58	6.67	1560
EQUIDOCK	0	8	29	13.30	0	12	47	10.19	**3.88**
DIFFDOCK-PP(1)	34	41	46	11.95	36	42	53	8.60	4.2
DIFFDOCK-PP(40)	42	**50**	**55**	**4.85**	45	**52**	**63**	4.23	153
DIFFDOCK-PP(40) - oracle	71	79	86	0.67	72	82	91	0.54	153

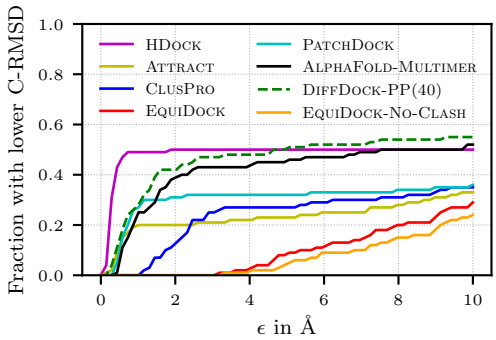

Figure 3: Fraction of complexes with a C-RMSD $< \epsilon$ for different values of ϵ on 100 samples from the DIPS test set.

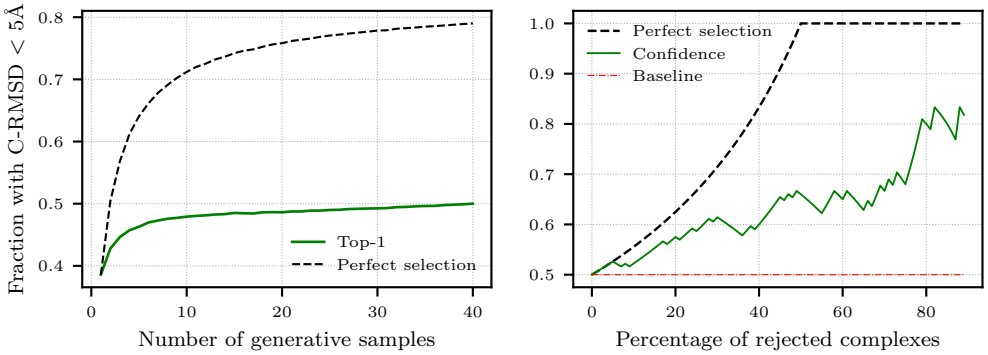

Figure 4: **Left**: Performance of DIFFDOCK-PP for increasing number of generative samples. "Perfect selection" is the theoretically best performance of the diffusion model assuming a perfectly accurate confidence model. **Right**: Fraction of predictions with C-RMSD < 5Å considering only predictions for the part of the dataset where DIFFDOCK-PP(40) is most confident.

dence score according to the confidence model) and compare it to the performance when we ignore

the confidence score and instead pick the sample than minimizes the C-RMSD out of the generated samples. Performance here is defined as the fraction of samples having a C-RMSD below 5Å. While the top-1 performance consistently increases with increasing number of generated samples, there still is a significant gap behind the performance in the perfect selection regime. This points out that DIFFDOCK-PP can achieve significantly better performance by improving the current confidence model or by developing a more involved selection algorithm.

The right plot of Figure 4 illustrates the selective accuracy of our model if we consider only complexes with confidence predictions above a certain threshold. We do this by ordering the final predicted complex structures by their assigned confidence prediction for all 100 complexes and removing the complexes with the lowest confidence score one by one. By removing predictions with lower confidence, we see an increase in the success rate of our prediction model.

We note that due to time constraints, it was not possible to properly evaluate our method on the Docking Becnhmark 5.5 (DB5.5) dataset Vreven et al. (2015) and leave it as an interesting future extension to this work.

6 CONCLUSION

We present DIFFDOCK-PP, a diffusion generative model for rigid protein-protein docking. Our approach is inspired by recent advancements in molecular docking (Corso et al., 2022), which tackles docking via a generative model over ligand poses. DIFFDOCK-PP outperforms existing deep learning models and performs competitively against search-based methods at a fraction of their computational cost. The effectiveness of our simple approach paves the way for further investigation into deep learning for modeling biomolecular interactions.

7 ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1745302. It is also supported by the Machine Learning for Pharmaceutical Discovery and Synthesis (MLPDS) consortium, the Abdul Latif Jameel Clinic for Machine Learning in Health, the DTRA Discovery of Medical Countermeasures Against New and Emerging (DOMANE) threats program, the DARPA Accelerated Molecular Discovery program and the Sanofi Computational Antibody Design grant. The authors thank Dr. Ricardo Acevedo Cabra and Prof. Dr. Massimo Fornasier for providing the opportunity to work on this project as part of the TUM Data Innovation Lab.

REFERENCES

Sankar Basu and Björn Wallner. Dockq: a quality measure for protein-protein docking models. *PloS one*, 11(8):e0161879, 2016.

Rong Chen, Li Li, and Zhiping Weng. Zdock: an initial-stage protein-docking algorithm. *Proteins: Structure, Function, and Bioinformatics*, 52(1):80–87, 2003.

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

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

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modeling. *arXiv preprint arXiv:2202.02763*, 2022.

Sjoerd J De Vries, Marc Van Dijk, and Alexandre MJJ Bonvin. The haddock web server for data-driven biomolecular docking. *Nature protocols*, 5(5):883–897, 2010.

Sjoerd J de Vries, Christina EM Schindler, Isaure Chauvot de Beauchêne, and Martin Zacharias. A web interface for easy flexible protein-protein docking with attract. *Biophysical journal*, 108(3): 462–465, 2015.

Israel T Desta, Kathryn A Porter, Bing Xia, Dima Kozakov, and Sandor Vajda. Performance and its limits in rigid body protein-protein docking. *Structure*, 28(9):1071–1081, 2020.

Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, et al. Protein complex prediction with alphafold-multimer. *BioRxiv*, pp. 2021–10, 2021.

Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Independent se (3)-equivariant models for end-to-end rigid protein docking. *arXiv preprint arXiv:2111.07786*, 2021.

Mario Geiger, Tess Smidt, M Alby, Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice, Kostiantyn Lapchevskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, et al. Euclidean neural networks: e3nn. *Zenodo. https://doi. org/10.5281/zenodo*, 5292912, 2020.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Sheng-You Huang and Xiaoqin Zou. An iterative knowledge-based scoring function for protein–protein recognition. *Proteins: Structure, Function, and Bioinformatics*, 72(2):557–579, 2008.

Sheng-You Huang and Xiaoqin Zou. A knowledge-based scoring function for protein-rna interactions derived from a statistical mechanics-based iterative method. *Nucleic acids research*, 42(7): e55–e55, 2014.

John Ingraham, Max Baranov, Zak Costello, Vincent Frappier, Ahmed Ismail, Shan Tie, Wujie Wang, Vincent Xue, Fritz Obermeyer, Andrew Beam, et al. Illuminating protein space with a programmable generative model. *bioRxiv*, pp. 2022–12, 2022.

Arian R Jamasb, Ben Day, Cătălina Cangea, Pietro Liò, and Tom L Blundell. Deep learning for protein–protein interaction site prediction. *Proteomics Data Analysis*, pp. 263–288, 2021.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. *arXiv preprint arXiv:2206.01729*, 2022.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32 (5):922–923, 1976.

Dima Kozakov, David R Hall, Bing Xia, Kathryn A Porter, Dzmitry Padhorny, Christine Yueh, Dmitri Beglov, and Sandor Vajda. The cluspro web server for protein–protein docking. *Nature protocols*, 12(2):255–278, 2017.

Marc F Lensink, Raúl Méndez, and Shoshana J Wodak. Docking and scoring protein complexes: Capri 3rd edition. *Proteins: Structure, Function, and Bioinformatics*, 69(4):704–718, 2007.

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

Efrat Mashiach, Dina Schneidman-Duhovny, Aviyah Peri, Yoli Shavit, Ruth Nussinov, and Haim J Wolfson. An integrated suite of fast docking algorithms. *Proteins: Structure, Function, and Bioinformatics*, 78(15):3197–3204, 2010.

Matthew McPartlon and Jinbo Xu. Deep learning for flexible and site-specific protein docking and design. *bioRxiv*, pp. 2023–04, 2023.

Emanuele Rodolà, Zorah Lähner, Alexander M Bronstein, Michael M Bronstein, and Justin Solomon. Functional maps representation on product manifolds. In *Computer Graphics Forum*, volume 38, pp. 678–689. Wiley Online Library, 2019.

Christina EM Schindler, Isaure Chauvot de Beauchêne, Sjoerd J de Vries, and Martin Zacharias. Protein-protein and peptide-protein docking and refinement using attract in capri. *Proteins: Structure, Function, and Bioinformatics*, 85(3):391–398, 2017.

Dina Schneidman-Duhovny, Yuval Inbar, Ruth Nussinov, and Haim J Wolfson. Patchdock and symmdock: servers for rigid and symmetric docking. *Nucleic acids research*, 33(suppl_2):W363–W367, 2005.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Freyr Sverrisson, Jean Feydy, Joshua Southern, Michael M. Bronstein, and Bruno Correia. Physics-informed deep neural network for rigid-body protein docking. In *ICLR2022 Machine Learning for Drug Discovery*, 2022. URL `https://openreview.net/forum?id=5yn5shS6wN`.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Raphael Townshend, Rishi Bedi, Patricia Suriana, and Ron Dror. End-to-end learning on 3d protein structure for interface prediction. *Advances in Neural Information Processing Systems*, 32, 2019.

Brian L Trippe, Jason Yim, Doug Tischer, Tamara Broderick, David Baker, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

Ilya A Vakser. Protein-protein docking: From interaction to interactome. *Biophysical journal*, 107 (8):1785–1793, 2014.

Thom Vreven, Iain H Moal, Anna Vangone, Brian G Pierce, Panagiotis L Kastritis, Mieczyslaw Torchala, Raphael Chaleil, Brian Jiménez-García, Paul A Bates, Juan Fernandez-Recio, et al. Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. *Journal of molecular biology*, 427(19):3031–3041, 2015.

Yumeng Yan, Di Zhang, Pei Zhou, Botong Li, and Sheng-You Huang. Hdock: a web server for protein–protein and protein–dna/rna docking based on a hybrid strategy. *Nucleic acids research*, 45(W1):W365–W373, 2017.

Yumeng Yan, Huanyu Tao, Jiahua He, and Sheng-You Huang. The hdock server for integrated protein–protein docking. *Nature protocols*, 15(5):1829–1852, 2020.

A BASELINES: EXPERIMENTAL DETAILS

In this section we give some details on how we evaluated the baselines to which we compared our method.

For HDOCK, we used the HDOCKlite package that can be downloaded from `http://huanglab.phys.hust.edu.cn/software/hdocklite/`.

For EQUIDOCK, we used the official implementation found in `https://github.com/octavian-ganea/equidock_public`.

For ALPHAFOLD-MULTIMER, we first extracted FASTA sequences from the complexes using the residue names given in their respective PDB files. Each complex yielded one FASTA file with two chains. Then, we ran AlphaFold-Multimer v2.3.0 using the official implementation of the inference pipeline provided in `https://github.com/deepmind/alphafold`. This implementation utilizes the following datasets: BFD, MGnify, PDB70, PDB (structures in the mmCIF format), PDB seqres, UniRef30 (formerly UniClust30), UniProt, and UniRef90.

For ATTRACT, CLUSPRO and PATHDOCK, we used the results reported by Ganea et al. (2021).