FusionDock: Physics-informed Diffusion Model for Molecular Docking

Matthew R. Masters [1] **Amr H. Mahmoud** [1] **Markus A. Lill** [1]

Abstract

Protein-ligand docking is an important task in drug discovery and structure-based drug design. Generative deep learning models have recently emerged as a new approach to sampling protein-ligand conformations and have shown state-of-the-art results. However, these models are often solely based on patterns observed in a given dataset and are not based on the underlying physical process. We propose FusionDock, a diffusion model for protein-ligand docking which incorporates physical priors, such as classical mechanics principles, known molecular interactions, and constraint satisfaction. The aim of such a physics-informed model is to reduce overfitting, improve generalization, and to generate physically-meaningful structures. We demonstrate the performance of our model across a large test set and show that it outperforms existing deep learning methods.

1. Introduction

Protein-ligand docking is an important task in drug discovery and design that aims to predict the binding mode of a ligand to its protein receptor. Docking is widely-used and can be utilized in many different workflows including rational drug design, virtual screening, and toxicity prediction (Sethi et al., 2019). It can be done with or without knowledge of a specific protein binding site, termed *focused docking* and *blind docking* respectively. While most docking programs are intended for focused docking, they can be adapted for blind docking by pairing them with a pocket prediction tool (Masters et al., 2023a).

This docking process is often divided into two main components: searching and scoring. The search function is tasked with sampling ligand poses within the binding site while the scoring function is tasked with ranking these samples. Existing search functions are largely based on well-studied sampling methods such as Monte Carlo and genetic algorithms paired with a physics-based energy potential (Altuntaş et al., 2016). The same potential is often utilized by the scoring function in order to estimate the energy of each pose and rank low-energy poses towards the top.

A new generation of docking methods have begun to emerge which utilize deep learning (DL) models to replace the search and/or scoring functions. However, these models are often solely based on patterns observed in a given dataset and are not based on the underlying physics. Therefore, they are subject to overfitting and do not obey known physical laws. To address this shortcoming, we propose FusionDock, a DL model for protein-ligand docking which incorporates physical priors. By doing so, our model is able to learn from a smaller set of training samples and generalize better to new systems. We demonstrate the effectiveness of our model on a large test set of protein-ligand complexes and show that it outperforms existing state-of-the-art models.

2. Background and Related Work

2.1. Scoring Functions

The first applications of deep learning models to docking aimed at replacing the scoring function. These methods used an existing docking program to generate the poses but improved the results by reranking them. Several different networks have been applied in this way including convolutional neural networks (Ragoza et al., 2017; Mahmoud et al., 2020) and graph neural networks (Wang et al., 2021; Townshend et al., 2021). Additionally, many DL models have been designed with the aim of predicting binding affinity given a protein-ligand complex, which can also be utilized for the purpose of re-ranking. While these approaches have proven to be powerful in improving the scoring of poses, they do not address the sampling problem.

2.2. Generative Models

More recently, deep learning models targeting the sampling problem have begun to emerge. In McNutt et al. (2021), they extended their DL scoring function to perform Monte Carlo sampling in order to generate new ligand poses. In Ganea and Huang et al. (2021), the authors proposed an SE(3)-equivariant model for rigid protein-protein docking named Equidock. The same group extended this methodology and

[1]Department of Pharmaceutical Sciences, University of Basel, Klingelbergstrasse 50, 4056 Basel, Switzerland. Correspondence to: Markus Lill <markus.lill@unibas.ch>.

The 2023 ICML Workshop on Computational Biology. Baltimore, Maryland, USA, 2023.

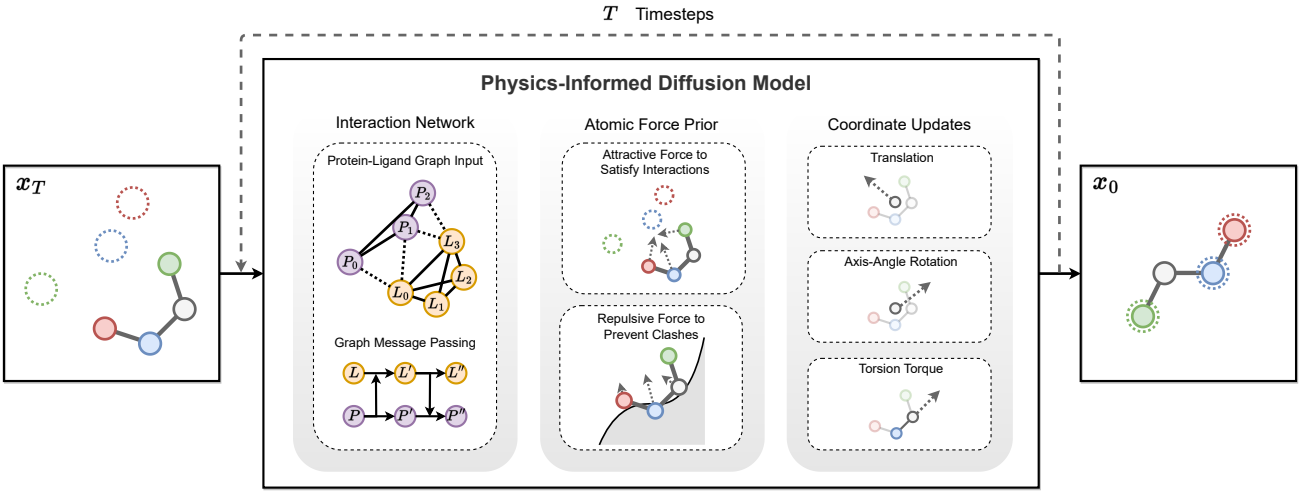


Figure 1. Overview of the physics-informed diffusion model utilized by FusionDock. The interaction network takes input graphs of the ligand and protein *interaction sites* and updates their hidden representation based on the local environment. These embeddings can be used to generate a force on each atom, pushing the ligand to form specific interactions while avoiding clashes with the protein surface. These atomic forces can then be used to calculate translation, rotation, and torsion updates to the ligand. When applied over several time steps, this process can generate docked poses from arbitrary starting configurations.

developed Equibind, a model for blind protein-ligand docking (Stärk et al., 2022). Then Corso et al. (2022) released another blind docking model using a novel diffusion network called DiffDock. DiffDock presented state-of-the-art results for the blind docking task, outperforming previous methods by nearly double. However, since these models aim to solve blind docking directly, they can not be applied in a focused docking context and therefore cannot be compared to most existing methods. Given that most targets have known, well-defined binding sites, it may be preferable to separate the binding site identification and docking tasks. Finally, in Masters et al. (2023b) they proposed a graph neural network which is trained to predict intermolecular protein-ligand distance matrices. While this method shows strong results for focused docking, it is not an end-to-end solution and is only capable of generating a single pose.

2.2.1. Diffusion Models

Diffusion models are a powerful class of generative deep learning model capable of producing high-fidelity molecular structures. Several other molecular diffusion models have been introduced, for example in the applications of protein structure prediction (Anand & Achim, 2022), de novo ligand generation (Peng et al., 2023), and learning molecular force fields (Arts et al., 2023). Diffusion models are built-on the theory of non-equilibrium thermodynamics (Sohl-Dickstein et al., 2015) and generate samples using Langevin dynamics, a popular integrator for simulations of physical systems. Diffusion models define a Markov chain process, termed the *forward diffusion process*, which incrementally adds noise to true data samples $x_0 \sim q(x)$ until they become indistinguishable from a standard normal distribution $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The amount of noise added at each step is controlled by a variance schedule with T time steps, typically defined by a decaying cosine function. The DL network is then tasked with reversing this process by predicting the noise added to an intermediate sample x_t. Therefore, by sampling from a standard normal distribution, we are able to model the *reverse diffusion process* and generate realistic samples which follow the original data distribution $q(x)$. Although developed independently, diffusion models are closely related to score-based models where a network is trained to estimate the Stein score of a sample, defined as $\nabla_x \, log \, q \, (x)$. In order to apply score-based models on high-dimensional data with non-Euclidean manifolds, Song et al. (2019) proposed to add a pre-defined noise to the data and to predict the Stein score of these noised samples, mimicking the diffusion model approach. Thus, while diffusion models can theoretically be trained via maximum likelihood, or by minimizing the evidence lower bound, a better approach seems to be via score-matching, where the loss is simply the mean squared error between the predicted and true Stein scores (Ho et al., 2020).

3. Methodology

3.1. Physical Priors

3.1.1. Interaction Site Protein Representation

Rather than using purely data-driven input features, such as protein sequence embeddings, FusionDock uses a physically meaningful protein representation based on a molecular force field. This was accomplished by running molecular

dynamics simulations of protein structures in explicit water followed by hydration site analysis. This procedure generates *interaction sites* which occupy the solvent-exposed surface and accessible binding pockets, akin to a whole-protein-based pharmacophore. The interaction sites are encoded with water occupancy and thermodynamic data following the protocol of WATsite (Hu & Lill, 2014), pharmacophore features (e.g. hydrogen-bonding, hydrophobicity, charge, etc.) following the protocol of PyRod (Schaller et al., 2019), and geometric features (e.g. curvature) following the protocol of CurPocket (Liu et al., 2020). Collectively these features describe the chemical and physical profile of each interaction site and are highly correlated with binding of specific chemical groups. While these features are expensive to compute, they can be re-used each time a new ligand is docked to an existing protein, as is often the case in virtual screening workflows.

3.1.2. Internal Coordinate Constraints

An issue that often arises in the generation of molecular conformers with deep neural networks is that there is no guarantee that the conformer is physically plausible and doesn't contain artifacts such as exaggerated bond lengths and angles. In addition, we know empirically that bond lengths and angles often have low variability and equilibrium values are easily attainable. Therefore, a ligands conformation can be approximated via it's torsion angles, ensuring that bond lengths and angles remain constrained. This approximation has the added benefit of using a reduced set of degrees of freedom needed to model the conformation ($M << 3N$ for molecules with N atoms and M torsions). This approach has already been introduced to diffusion models (Jing et al., 2022) and applied in the context of molecular docking with state-of-the-art results (Corso et al., 2022). We adopt the score-matching framework established by Corso et al. where the diffusion process is defined on the product space $\mathbb{P} = \mathbb{T}^3 \times SO(3) \times SO(2)^M$ where $\mathbb{T}(3) \cong \mathbb{R}^3$ is the 3D translation group, $SO(3)$ is the group defining 3D rigid rotations, and $SO(2)$ is the group defining 2D rotations about each torsion.

3.1.3. Physics-Guided Diffusion

By understanding the properties of the translation, rotation, and torsion scores, we can make rational design decisions to include physical priors in their prediction. A major insight is that the scores are theoretically equivalent to a set of non-equilibrium forces being applied to the ligand (Wu et al., 2022; Arts et al., 2023). This enables us to 1) calculate the predicted scores in the same way you would calculate the force for the physical system, and 2) to enforce attractive and repulsive constraints for the interaction sites and protein surface respectively.

In the context of the translation of a rigid-body system, it is well-established in classical mechanics that the resultant force is the sum of individual forces acting on each particle in the system. Similarly, the rotational force can be described by a torque about the center-of-mass and the torsion force can be described as the difference between the torques applied on each side of the rotatable bond. Using this approach to calculate the scores establishes an explicit connection to the known physics, and enables more accurate prediction of these variables.

By using the protein interaction sites described earlier, a useful property emerges for modelling the reverse diffusion process. Since ligand atoms should directly overlap with their paired interaction site, the diffusion model can be designed to produce scores which attract ligand atoms towards likely interaction sites. By enforcing this directional constraint, we can prevent repulsive scores between the ligand and interaction sites which could lead to overfitting on training systems. Similarly, we utilize an explicit repulsive force between the ligand and protein surface to ensure complexes do not contain high-energy steric clashes. This is accomplished by detecting ligand atoms which overlap with the protein and applying an additional force, pushing the ligand away from the surface and towards the nearest interaction site. Finally, since the ligand conformation is not only dictated by interactions with the protein, but also the ligand conformation itself, additional ligand-ligand information is used to predict the torsion score.

3.2. Diffusion Model

The diffusion model takes a 3D graph of the interaction sites, a 3D graph of the ligand (at time t), and the current time step variance σ_t as input. The interaction network uses graph message passing between protein and ligand nodes to iteratively update their hidden representation. Since the maximum displacement a ligand can have is dictated by σ_t, the edges connecting the protein and ligand nodes must be less than $3\sigma_t + 4$ Ångstrom. The attractive force can then be calculated by a multi-layer perceptron (MLP) network which takes connected protein and ligand node embeddings and predicts a non-negative magnitude which is multiplied by the corresponding edge vector to obtain the force. These forces are pooled by summation on each ligand node to obtain the *atomic force prior*. This can be formalized by Equation 1 where h_i and h_j are the embeddings for ligand atom i and interaction site j respectively, ϕ is the MLP network, and v_{ij} is the edge vector pointing from i to j.

$$f_i = \sum_j \frac{v_{ij}}{\|v_{ij}\|} \cdot max(0, \phi(h_i, h_j, \sigma_t)) \tag{1}$$

On the other hand, the repulsive force is disentangled from the network and has no learnable weights. Calculating the

repulsive force between the ligand and protein atoms directly can counter-act the action of the attractive force, negatively impacting performance. Therefore, we reformulate the repulsive force as an additional attractive force, pushing overlapping ligand atoms towards their nearest interaction site. This works because these sites approximate the solvent exposed volume and do not clash with the protein. This force can be formalized as $f_i^* = k \cdot v_{ij}$ where v_{ij} is the edge vector pointing from atom i to it's closest interaction site j, and k is the force constant hyperparameter ($k = 0.5$ here). Rather than combining the two atomic force priors together, they are applied independently throughout the reverse diffusion process. While the attractive force is applied once at each time step, the repulsive force can be applied multiple times in order to ensure convergence.

3.3. Auxiliary Models

3.3.1. Pocket Prediction

Recent deep learning models targeting the blind docking task attempt to solve the problem directly without any prior knowledge of the binding pocket location (Stärk et al., 2022; Corso et al., 2022). Assessment of these models has shown that their performance largely arises from their ability to predict the correct pocket, and not the docking task itself (Yu et al., 2023; Masters et al., 2023a). Therefore, FusionDock opts to separate the pocket prediction from the docking task. Since we have already generated interaction sites which occupy the protein pockets and contain features correlated with ligand binding, we can leverage these same features for the pocket prediction task. To this end, we trained a classification model to predict interaction sites within 2Å of any native pose atom. To cluster the output probabilities into pocket predictions, a 10Å sphere is placed on each site and the likelihood of sites within the sphere is summed together. At each clustering step, the site with the highest sum is selected as the next pocket center and all contributing sites are removed from subsequent steps. This straightforward method produces good results on the test set, outperforming widely-used pocket predictions such as P2Rank (Kriválk & Hoksza, 2018) (See Appendix Section A.3).

3.3.2. Scoring Model

The aim of the scoring model is to rank the sampled poses such that the top-ranked poses resemble the most likely binding modes. In this case, the score is a scalar value predicted for each pose and should not be confused with the Stein scores predicted by the diffusion model. Interaction sites and sampled ligand poses are featurized in the same way as the diffusion model and provided as input graphs. Edges connect ligand atoms and interaction sites if they are within 2Å of each other. The scoring model feeds these edge features through a MLP network to generate embeddings which are pooled by summation on each node. This process is reapeated with another MLP to produce a single embedding for each pose, which is provided to a final layer to predict the score. The model is trained as a binary classifier, where poses < 2.5Å RMSD are considered positive, > 5.0Å are considered negative, and intermediates are excluded.

4. Results and Discussion

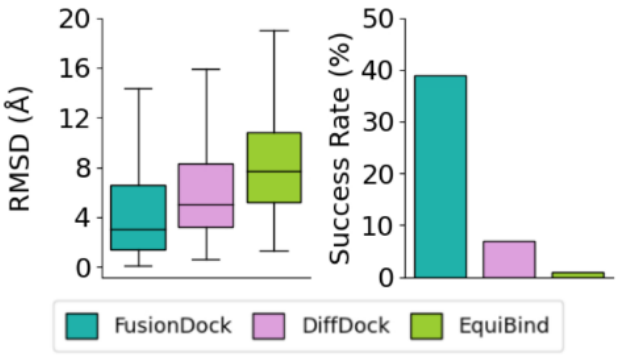


Figure 2. Results for blind docking task on the test set. Left: RMSD (Å) distribution box plots (outliers not shown). Right: Success rate (Fraction of poses $<$ 2Å). All data reported in this figure comes from the top-10 ranked poses. Additional results of the top-1, top-3 and top-5 ranked poses are presented in Appendix Section A.6. Note that EquiBind only produces a single pose, so their results remain the same in each figure.

FusionDock attained a median RMSD of 3.1Å and success rate of 39% among the top-10 ranked poses, outperforming existing deep learning methods by several fold. DiffDock attained a median RMSD of 5.0Å and success rate of 7% while EquiBind faired the worse with a median RMSD of 7.7Å and success rate of just 1%. This ranking of methods is consistent among the top-1, top-3, and top-5 results as well. DiffDock likely underperforms on this test set due to the sequence-similarity-based split. Since DiffDock relies on sequence embeddings as the protein representation, it performs poorly when there is a domain shift with novel target sequences. On the other hand, the interaction sites used by FusionDock is largely independent of a specific protein system and allows the model to generalize under this same domain shift. The dataset also featured a large test:train ratio, which assesses the models' ability to generalize from less training data. The poor performance of deep learning docking methods on this test set further highlights the difficulty of generalization and the need to make physics-informed models which incorporate priors and explainable mechanisms rather than a black-box solution.

5. Conclusion

We have presented a novel deep learning diffusion model for protein-ligand docking that outperforms existing deep-learning-based docking methods, and achieves state-of-the-art results on a rigorous test set. We demonstrated that incorporating physical priors into the model leads to more accurate results, while reducing the need for large training sets and decreasing model complexity. Our method generates realistic protein-ligand complexes that are stable, with no clashes, stretched bonds or angles. Overall, our method represents a significant advance in the field of protein-ligand docking and provides a promising direction for future research in this area.

References

Altuntaş, S., Bozkus, Z., and Fraguela, B. B. Gpu accelerated molecular docking simulation with genetic algorithms. In *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30–April 1, 2016, Proceedings, Part II 19*, pp. 134–146. Springer, 2016.

Anand, N. and Achim, T. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

Arts, M., Satorras, V. G., Huang, C.-W., Zuegner, D., Federici, M., Clementi, C., Noé, F., Pinsler, R., and Berg, R. v. d. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *arXiv preprint arXiv:2302.00600*, 2023.

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

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., and Smola, A. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

Ganea, O.-E., Huang, X., Bunne, C., Bian, Y., Barzilay, R., Jaakkola, T., and Krause, A. Independent se (3)-equivariant models for end-to-end rigid protein docking. *arXiv.org, e-Print Arch.*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hu, B. and Lill, M. A. Watsite: Hydration site prediction program with pymol interface, 2014.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. Torsional diffusion for molecular conformer generation. *arXiv preprint arXiv:2206.01729*, 2022.

Krivák, R. and Hoksza, D. P2rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of cheminformatics*, 10:1–12, 2018.

Liu, Y., Grimm, M., Dai, W.-t., Hou, M.-c., Xiao, Z.-X., and Cao, Y. Cb-dock: A web server for cavity detection-guided protein–ligand blind docking. *Acta Pharmacologica Sinica*, 41(1):138–144, 2020.

Mahmoud, A. H., Masters, M. R., Yang, Y., and Lill, M. A. Elucidating the multiple roles of hydration for accurate protein-ligand binding prediction via deep learning. *Commun. Chem.*, 3(1):1–13, 2020.

Masters, M., Mahmoud, A. H., and Lill, M. A. Pocketnet: Ligand-guided pocket prediction for blind docking. In *ICLR 2023-Machine Learning for Drug Discovery workshop*, 2023a.

Masters, M. R., Mahmoud, A. H., Wei, Y., and Lill, M. A. Deep learning model for efficient protein–ligand docking with implicit side-chain flexibility. *Journal of Chemical Information and Modeling*, 63(6):1695–1707, 2023b.

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. *J. Cheminf.*, 13(1):1–20, 2021.

Peng, X., Guan, J., Peng, J., and Ma, J. Pocket-specific 3d molecule generation by fragment-based autoregressive diffusion models. 2023.

Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J., and Koes, D. R. Protein–ligand scoring with convolutional neural networks. *J. Chem. Inf. Model.*, 57(4):942–957, 2017.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50 (5):742–754, 2010.

Schaller, D., Pach, S., and Wolber, G. Pyrod: Tracing water molecules in molecular dynamics simulations. *Journal of Chemical Information and Modeling*, 59(6):2818–2829, 2019.

Sethi, A., Joshi, K., Sasikala, K., and Alvala, M. Molecular docking in modern drug discovery: Principles and recent applications. *Drug discovery and development-new advances*, 2:1–21, 2019.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

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

Stärk, H., Ganea, O.-E., Pattanaik, L., Barzilay, R., and Jaakkola, T. Equibind: Geometric deep learning for drug binding structure prediction. *arXiv.org, e-Print Arch.*, 2022.

Townshend, R. J., Eismann, S., Watkins, A. M., Rangan, R., Karelina, M., Das, R., and Dror, R. O. Geometric deep learning of rna structure. *Science*, 373(6558):1047–1051, 2021.

Wang, X., Flannery, S. T., and Kihara, D. Protein docking model evaluation by graph neural networks. *Front. Mol. Biosci.*, 8:402, 2021.

Wu, L., Gong, C., Liu, X., Ye, M., and Liu, Q. Diffusion-based molecule generation with informative prior bridges. *arXiv preprint arXiv:2209.00865*, 2022.

Yu, Y., Lu, S., Gao, Z., Zheng, H., and Ke, G. Do deep learning models really outperform traditional approaches in molecular docking? *arXiv preprint arXiv:2302.07134*, 2023.

A. Appendix

A.1. Dataset and Data Preparation

PDBbind (v2020), which contains 19,443 protein-ligand complexes, was used as the dataset for training and evaluation of docking methods. However, many of these complexes contain the same or closely related protein receptors leading to redundancy for many targets. In addition, as mentioned in the main text, the generation of our protein input features is expensive since it involved molecular dynamics simulations. Thus, we opted to select a set of representative protein structures for which to run the simulations. This was done by grouping PDBs by their UniProt ID and randomly selecting a representative structure from the group. In the case where no UniProt ID could be identified, the PDB was treated as its own representative structure. Following this process, the dataset contained a total of 4,240 representative protein structures.

A.1.1. Similarity Split

In order to construct a test set that can evaluate a models ability to generalize to new targets, we opted for a sequence-similarity-based splitting strategy. To this end, MMSeqs2 (Steinegger & Söding, 2017) was used to compute the pairwise sequence similarity of all proteins. The alignment was performed using default parameters, except for coverage which was set at 0.05, meaning at least 5% of query and target sequences are covered by the alignment. MMSeqs2 can produce asymmetric sequence alignments for multiple reasons and therefore the similarities were symmetrized by taking the larger of the two values. The split was created by selecting a PDB at random, recursively finding all related structures with similarity above a given threshold, and adding them to the split. This creates disjoint training and test sets with a maximum sequence similarity between them. In this work, a sequence similarity cutoff of 35% was used, and the splitting process continued until the split reached roughly 75-25 train-test ratio. Setting the similarity cutoff any lower resulted in test sets much larger than intended. The test split used in this work contains 1,087 systems, leaving 3,153 remaining for training and validation.

A.1.2. Molecular Dynamics Simulations

As described in Section 3.1.1, molecular dynamics simulations were run in order to compute the protein input representation. Proteins were prepared by removing co-crystallized small molecules, waters, and ions followed by Schrödinger's Protein Preparation Wizard (Madhavi Sastry et al., 2013). Schrödinger's Desmond was used to perform the simulations following it's standard NPT relaxation protocol (Bowers et al., 2006). The production simulations were run at 300K for 20ns each, with a large positional restraint of 50.0 kcal mol^{-1} Å^{-1} applied to all heavy atoms. Snapshots were taken every 20ps, resulting in 1,000 frames per simulation to be used for analysis.

A.1.3. Interaction Site Generation

The molecular dynamics simulations were aligned and re-centered using VMD (Humphrey et al., 1996). The method used to generate the interaction sites builds upon previous hydration site analysis tools (WATsite and PyRod) which analyze explicit water behavior during the trajectory to predict *hydration sites*. However, the protocols were modified in order to predict a single set of sites which densely covers the whole protein surface. To this end, the position of water molecules were tracked throughout the trajectory and a 3D grid with resolution of 0.25Å was used to accumulate their occupancy. The grid was then clustered using the same hierarchical clustering algorithm implemented in WATsite with a cluster radius of 1.0Å. While hydration site analysis tools typically only cluster the highest occupied sites, we extended the clustering process to include all sites with at least 20% occupancy. This cutoff is slightly higher than the bulk occupancy, leading to the desired dense surface coverage (See Figure 4.I). The clustering process also assigns water molecules throughout the trajectory to the site they overlap with. This information can then be used to calculate thermodynamic quantities, pharmacophore features, and other properties (Summarized in Table 1). Some of these features describe a specific microstate (e.g. presence of a hydrogen bond) while others describe the ensemble of states (e.g. entropy). Since we need our input features to be in the latter form, we apply several different functions in order to describe the distribution. For continuous values, mean μ and variance σ statistics are used. For binary values, mean μ and average state time $\bar{t}$ are used. Average state time is the average time spent occupying the state before exiting the state again and is meant to differentiate between fast and slow transitions. Some additional features were included which encode information about the sites' environment and neighboring features. One feature places a series of spheres around each interaction site and calculates the fraction of the spheres' volume occupied by the sites, referred to here as interaction site density. The other feature takes the same series of spheres and sums the WATsite features (occupancy, enthalpy, entropy, and free energy) of the sites inside each sphere. These engineered features are more correlated with ligand binding than the base features and demonstrate strong predictive power (See Figure 3).

Methodology	Feature Description	Statistic	Size
WATsite (Hu & Lill, 2014)	Water Occupancy	v	1
	Water Enthalpy	μ, σ	2
	Water Configuration Entropy	v	1
	Water Free Energy	v	1
PyRod (Schaller et al., 2019)	Trapped Water (more than 2 H-bonds)	$\mu, \bar{t}$	2
	Hydrogen-Bond Donor (single, double, either)	$\mu, \bar{t}$	6
	Hydrogen-Bond Acceptor (single, double, either)	$\mu, \bar{t}$	6
	Mixed Donor-Acceptor	$\mu, \bar{t}$	2
	Charge (positive, negative)	μ, σ	4
	Uncharged Environment	$\mu, \bar{t}$	2
	Cation-Pi Interaction	μ, σ	2
	Hydrophobicity (w/ and w/o scaling by burriedness)	μ, σ	4
CurPocket (Liu et al., 2020)	Histogram of protein curvature using different sphere radii $r \in [4, 6, 8, 10, 12, 14, 16]$	μ, σ	14
Additional	Histogram of interaction site density using different sphere radii $r \in [2, 4, 6, ..., 20, 22, 24]$	μ, σ	12
	Histogram of cumulative WATsite features using different sphere radii $r \in [2, 4, 6, ..., 20, 22, 24]$	μ, σ	48
Total Features			**107**

Table 1. Overview of interaction site features derived from protocols of WATsite, PyRod, and CurPocket. As described in the methodology, some pre-processed features describe individual states (MD frames) while others describe the collective ensemble of states. Therefore, the statistic column is used to indicate what statistical functions were used to produce the interaction site feature. v indicates the value already describes the ensemble and is used directly, μ indicates the mean, σ indicates the variance, and $\bar{t}$ indicates the average state time.

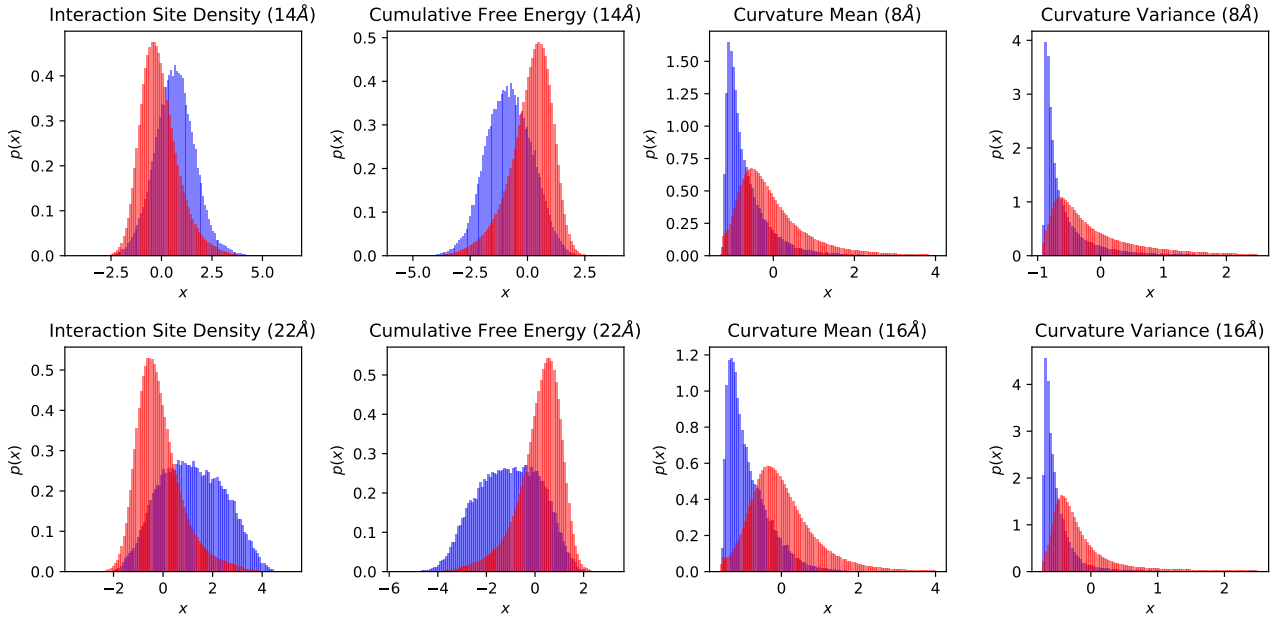
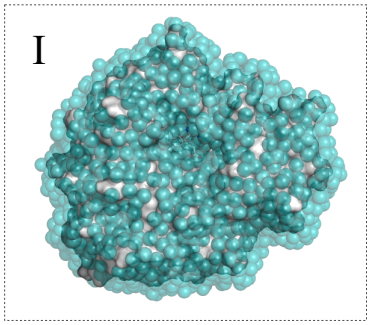
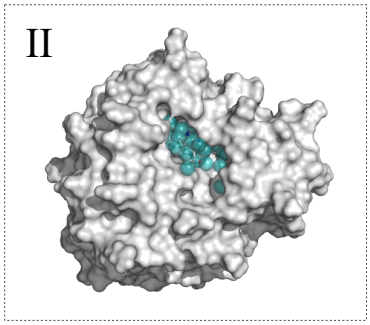
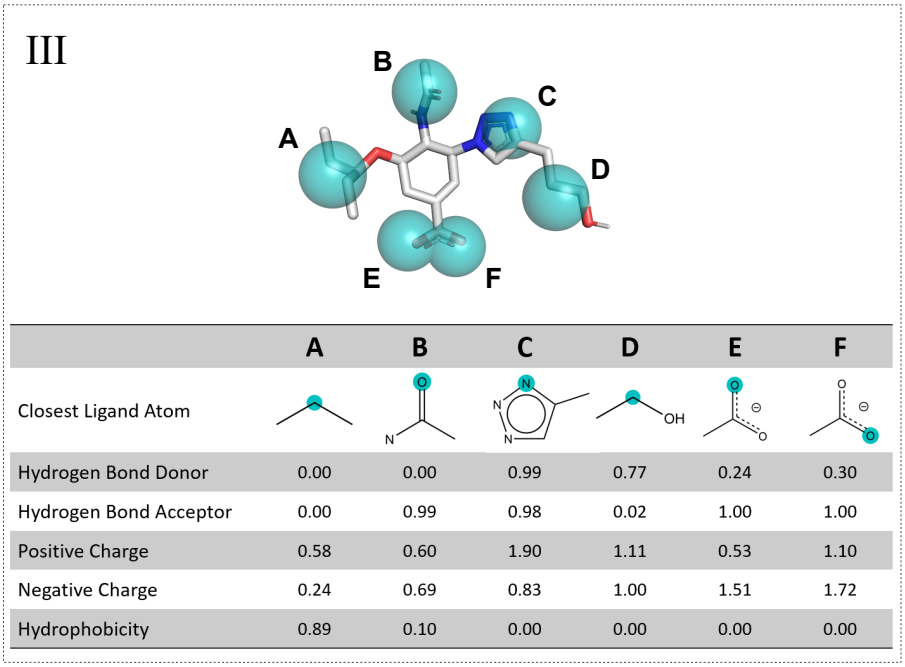


Figure 3. Normalized distributions of select interaction site features which are correlated with drug binding. Sites within 2Å of native pose heavy atoms were used to separate features into positive (blue; n=61,709) and negative (red; n=4,329,334) sets. Features are also normalized (i.e. z-score) by calculating the mean and variance across the full dataset.







	A	B	C	D	E	F
Closest Ligand Atom						
Hydrogen Bond Donor	0.00	0.00	0.99	0.77	0.24	0.30
Hydrogen Bond Acceptor	0.00	0.99	0.98	0.02	1.00	1.00
Positive Charge	0.58	0.60	1.90	1.11	0.53	1.10
Negative Charge	0.24	0.69	0.83	1.00	1.51	1.72
Hydrophobicity	0.89	0.10	0.00	0.00	0.00	0.00

Figure 4. Example of generated interaction sites using PDB 4KS3. I) All interaction sites shown in teal with 40% opacity. II) Only sites with score > 10 from the pocket prediction model shown, highlighting the ability to isolate druggable pockets. III) Selected interaction sites overlapping with ligand atoms, and their associated pharmacophore features. Site A overlaps with an alkyl group and is characterized by low polarity and high hydrophobicity. Site B overlaps with a carbonyl group and often acts as a hydrogen bond acceptor. Site C overlaps with a triazole group and is characterized by high positive charge and a propensity to form hydrogen bonds. Site D overlaps with a hydroxyl group and often acts as a hydrogen bond donor. Sites E and F overlap with the carboxylic acid oxygens and have high negative charge and often acts as a hydrogen bond acceptor.

A.1.4. Ligand Featurization

Feature Description	Encoding	Size
Element†	One-Hot	12
Number of Bonds	One-Hot	5
Number of Rings	One-Hot	6
Ring Size	One-Hot	6
Formal Charge	One-Hot	3
Number of Hydrogens	One-Hot	4
Hybridization Type	One-Hot	5
Chirality Type	One-Hot	4
Atomic Radius	None	1
MMFF Partial Charge	Z-score	1
MMFF Parameters Present	Boolean	1
Hydrogen-Bond Acceptor	Boolean	1
Hydrogen-Bond Donor	Boolean	1
Hydrogen-Bond Donor and 1 Hydrogen	Boolean	1
Hydrogen-Bond Donor and 2 Hydrogens	Boolean	1
Hydrogen-Bond Acceptor and Donor	Boolean	1
Hydrophobic Group	Boolean	1
Positively Charged Group	Boolean	1
Negatively Charged Group	Boolean	1
Total Features		***55***

Table 2. Overview of ligand atom features used for the diffusion and scoring models. Positions are also encoded into the ligand graph and are used to generate edge distance embeddings, but are not included here since they change throughout the diffusion process. Bonds are included as edge features with a one-hot encoding of five types (single, double, triple, aromatic, other). † Limited to the 11 most common elements in dataset, plus a miscellaneous class.

A.1.5. DATA PROCESSING

In order to isolate the diffusion process to a given pocket, we define a sphere centered on the pocket center which is extended to contain the full possible binding site. The radius of the sphere is determined by the largest radius of the ligand plus a constant buffer of 10Å. The largest radius of the ligand is determined by taking the maximum value of the upper-bound distance matrix computed using the ETKDG method (Riniker & Landrum, 2015). This ensures that the value does not depend on a given molecular conformation. Then, protein atoms and interaction sites within the sphere are selected as input to the model. The interaction sites include the features listed in Table 1, as well as the druggability score given by the pocket prediction model. The protein atoms are not encoded with features, except for the position and empirical atomic radius of each atom. These points are only included to detect overlapping atoms and guide the diffusion to prevent steric clashes. Interaction sites are connected if they are within 10Å of each other, forming a locally connected graph. Since the position of the interaction sites remains constant, the distance is pre-computed and used as an edge feature.

A.2. Model Details

A.2.1. DIFFUSION MODEL

Interaction Layers The first layers utilized by the diffusion model are the graph interaction layers. These layers take the input graphs of the ligand and interaction sites and generate new node embeddings which are aware of their context. These layers are message-passing neural networks based on the tensor product convolution layer introduced by Corso et al. (2022). Messages are computed by taking the tensor products of node features with a spherical harmonic representation of the edge vector. In each layer, messages are first passed from ligand nodes to other ligand nodes (determined by covalent bonding plus any additional nodes within 5Å). Then messages are passed between interaction sites based on the locally connected edges of sites within 10ÅḞinally, messages are passed from the ligand to the interaction sites and vice versa. This study uses a stack of three interaction layers, with batch normalization, dropout of 0.5, and spherical harmonics up to a degree of 2. This produces dense embeddings of length 32 which are used by subsequent networks to predict the scores.

Atomic Force Priors In order to generate the scores, we take advantage of the explicit connection between physical forces and the Stein score. We know that the translation and rotation of a molecule does not arise from interactions between a theoretical centroid and the environment directly. Rather, the translation arises from the coupled movement of covalently bonded atoms. Therefore, rather than predict a force acting on the centroid directly, we chose to compute a force for each ligand atom individually and use these to inform the score prediction. Additionally, we can enforce that these forces are attractive and point towards the interaction sites accessible at a given time step. To this end, a network is provided the edges linking the ligand and interaction sites, and predicts a non-negative scalar value. This output corresponds to the magnitude of the attractive force and is multiplied by the edge vector in order to obtain the force vector. Then each of the force vectors originating from a given atom are summed together to produce a single force for each ligand atom. This was formalized in the main text as Equation 1.

In order to provide the model a physical prior which respects high-energy nature of steric clashes, we include another atomic force prior designed to repell atoms overlapping with the protein. As described in the main text, the naive implementation of this repulsion introduces issues with the sampling process. Therefore, we reformulate this repulsive force as another attractive force pointing towards the nearest interaction site. Overlapping atoms are found by computing the nearest protein atom to each ligand atom and determining if they are closer than 1.5-times their combined empirical atomic radii. Then the nearest interaction site to each overlapping atom is identified and used to calculate the force $f_i^* = k \cdot v_{ij}$ where v_{ij} is the edge vector pointing from atom i to it's closest interaction site j, and k is the force constant hyperparameter ($k = 0.5$ here). This force is applied separately from the diffusion score in order to ensure convergence. The number of iterations rises from 0 to 10 throughout the reverse diffusion process according to Equation 2, rounded to the nearest integer.

$$n(t) = 5 \cdot (cos(\pi(t+1)) + 1) \tag{2}$$

Translation Score Head The translation score can be calculated by simply taking the sum of atomic forces generated in the previous section, multiplied by a scaling factor. We can express the unscaled translation score as $\tilde{s}_{tra} = \sum_{i=1}^{N} f_i$ where N is the number of nodes in the ligand graph (number of heavy atoms) and f_i is the force on node i. The scaled translation score can then be expressed by Equation 3 where $\|.\|$ is the l_2 norm, σ_t is the diffusion process variance at time t, ψ is a neural network layer which takes the current translation magnitude and scales it according to an embedding of the

variance. This function maintains the same direction as the original force vector but rescales it according to the current timestep in the diffusion process. The translation score s_{tra} can then be used directly for score matching (See Section A.2.1) and outperforms purely data-based methods (See Figure 6).

$$s_{tra} = \frac{\tilde{s}_{tra} \cdot \psi\left(\left\|\tilde{s}_{tra}\right\|, \sigma_t\right)}{\left\|\tilde{s}_{tra}\right\| \cdot \sigma_t} \tag{3}$$

Rotation Score Head To generate a rotation force, also referred to as the torque, we can utilize the same atomic force prior generated earlier. First, the torque of each atoms force is calculated about the center-of-mass by taking the cross product $\tau_i = r_i \times f_i$ where r_i is the vector drawn from the center-of-mass to atom i and f_i is the force being applied on atom i. These torques can then be summed to obtain the unscaled rotation score $\tilde{s}_{rot} = \sum_{i=1}^{N} \tau_i$. Similar to what was done in the translation score calculation, the rotation score is scaled using σ_t and a final neural network layer. This is formalized in Equation 4 where s_{rot} is the scaled rotation score, ψ is another neural network layer, and $\bar{\sigma_t}$ is the norm of SO3 scores with variance σ_t.

$$s_{rot} = \frac{\tilde{s}_{rot} \cdot \psi\left(\left\|\tilde{s}_{rot}\right\|, \sigma_t\right) \cdot \bar{\sigma_t}}{\left\|\tilde{s}_{rot}\right\|} \tag{4}$$

Torsion Score Head To compute the torsion score we use a combination of the atomic force prior and the pseudotorque method introduced by (Jing et al., 2022). While the atomic force prior is adept at predicting the magnitude of a torsion change, it struggles to predict the correct sign. Therefore, the prior is used to calculate the magnitude and the pseudotorque method is used to calculate the sign, which are combined to produce the unscaled torsion score. In order to calculate the physical torque of a particular torsion from the atomic force prior, we first align each torsion along the z-axis by removing the torsion center and applying a rotation matrix. Then the atoms and forces are projected onto the xy-plane and torques are calculated. This is formalized in Equation 5 where r_i and f_i are the position and force of atom i, R_m and c_m are the rotation matrix and center of torsion m which aligns along the z-axis, and p_{mi} is a binary variable indicating which side of torsion m atom i belongs to. The scaled torsion score can then be calculated using Equation 6 where the absolute value of τ_m is multiplied by the pseudotorque η_m and scaled by the square root of the SO2 score norm with variance σ_t.

$$\tau_m = \sum_i p_{mi} \cdot (R_m r_i \times (R_m(f_i - c_m))) \tag{5}$$

$$s_{tor_m} = |\tau_m| \cdot \eta_m \cdot \sqrt{\bar{\sigma_t}} \tag{6}$$

Training and Inference The diffusion model can be trained via score-matching, which is simply the mean squared error between the predicted scores and the ground truth scores calculated from the pre-defined noise. While other paradigms for training a diffusion model exist, they often underperform or are considerably more expensive to compute (Ho et al., 2020). This is trivial for the translation score, since it can be easily calculated from a standard normal distribution as $s_{tra} = \frac{-x}{\sigma_t^2}$ where x is the noise added from the pre-defined Gaussian with σ_t variance. However, since the rotations and torsions are defined by the $SO(3)$ and $SO(2)$ groups respectively, it is more difficult to predict their score. To address this, Corso et al. 2022 pre-computes the distributions' PDF and scores at different σ_t and uses linear interpolation at runtime in order to by-pass expensive computations.

For validation of changes to the model and optimization of hyperparameters, 5-fold cross-validation was used with random splits. While this does not provide the same robustness and assessment of generalization as splitting by sequence, it was not practical to split the training dataset using the same method used to split the test set. The diffusion model was trained for 500 epochs, with an initial learning rate of 0.001 decaying with a factor of 0.7 and patience of 30. The scoring was trained for 50 epochs with the same learning rate schedule.

During inference, 128 poses were sampled per pocket using 30 time steps. The ligand is initialized with uniformly random orientation and torsions and placed in the center of the pocket. Additional experiments showed no meaningful difference when the ligand is initialized with a RDKit conformation or random translation.

A.3. Pocket Prediction

The pocket prediction model utilizes the same protein interaction site features as the diffusion model. These features are combined with 10 ligand descriptors and a 256-bit Circular molecular fingerprint calculated using RDKit (Rogers & Hahn, 2010). The complete list of features can be seen in Table A.3. The CatBoost model was trained using AutoGluon with the best_quality preset (Erickson et al., 2020). In Figure 5, the results are shown in comparison to another widely-used pocket prediction tool, P2Rank. In this analysis, we define success as having a predicted pocket center within 12Å of the native pose center. We also explored defining the cutoff as a function of molecule size but found no correlation between molecule size and distance.

Feature Description	Size
Interaction Sites (See Appendix Section A.1.3)	107
256-bit Circular Molecular Fingerprint	256
Molecular Weight	1
Radius of Gyration	1
Total Polar Surface Area	1
Computed LogP	1
Molecular Refractivity	1
Number of Rotatable Bonds	1
Number of H-Bond Acceptors	1
Number of H-Bond Donors	1
Number of Rings	1
Number of Amide Bonds	1
Total Features	**373**

Table 3. Overview of input features for the pocket prediction model. Radius of gyration was computed using one randomly generated molecular conformer. The rest of the descriptors are conformation-independent.

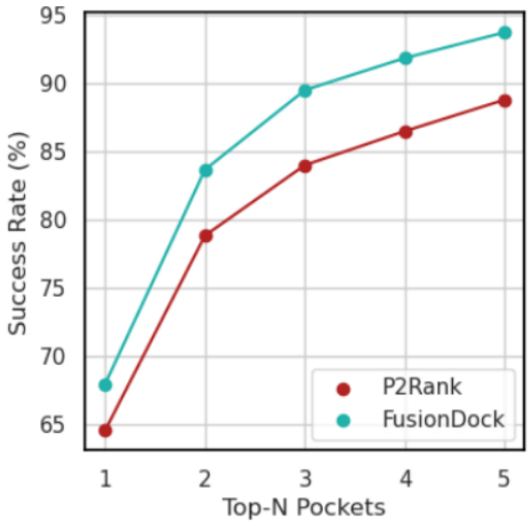


Figure 5. Pocket prediction results of FusionDock (teal) compared against a widely-used tool, P2Rank (red). Success rate is the fraction of systems with a predicted pocket center within 12Å of the native pose center.

A.4. Scoring Model

The scoring model evaluates each of the generated poses and gives it a score in order to be ranked. It aims to rank the near-native poses towards the top of the list while ranking less likely poses towards the bottom. The scoring model is provided graphs of the ligand pose and interaction sites, using the same featurization as the diffusion model. Interaction sites within 2Å of each ligand atom are considered nearby and are linked by edges. The network does not use the same interaction layers as the diffusion model to avoid overfitting and unnecessarily complexity. Instead, small MLP networks embed the node features of the ligand and interaction sites independently. Then edge features are created between ligand atoms and other nearby ligand atoms, accounting for internal molecular strain which may penalize the score, and between ligand atoms and interaction sites, accounting for intermolecular interactions and the contributions of binding. These edge features are fed thru another MLP network to produce an embedding with size 128. These embeddings are pooled by summation so that each node is now described by a single embedding and passed through another MLP network. These node embeddings are pooled by summation once more so that each pose is described by a single embedding. Finally, these molecular embeddings are passed thru one last layer in order to produce a scalar score output. The output is passed through a sigmoid activation function in order to assign a probability to each pose. The model was trained using the binary cross entropy loss where positive samples were those with RMSD < 2.5Å and negative samples are those with RMSD > 5Å. Samples with RMSD between 2.5 and 5Å were excluded since they are ambiguous and could be considered either a good or bad binding mode.

A.5. Evaluation

A.5.1. Benchmark Programs

In order to draw a fair comparison between FusionDock and the two deep-learning-based docking methods, we retrained each of the models with the training dataset constructed in Section A.1.1. The test set employed by these papers follows a temporal split and therefore would be unfair to compare against our method since the representative protein features would appear in both the training and test data. Although, since this dataset is considerably smaller (n=373) and does not take receptor similarity into account, we believe our split offers the same if not better comparison among docking methods. DiffDock was retrained following the same protocol described in the paper, except for batch size which was set to 8 (from 16) to prevent batches which don't fit into memory. First, the small score model was trained for 300 epochs. Then, the confidence model was trained for 100 epochs using samples from the small score model. Finally, the large score model was trained for 850 epochs and used with the confidence model in order to generate samples. DiffDock training was distributed over four GeForce RTX 3090, while EquiBind and FusionDock were trained using a single RTX 3090 GPU. EquiBind was retrained for 574 epochs, due to the early stopping implemented in their training protocol. EquiBind only generates a single pose and therefore all top-k results are the same for this method.

A.5.2. Metrics and Reporting

The models were assessed primarily on the RMSD (Å) of the top-ranked poses from each program, and the fraction of top-ranked poses under 2.0Å, referred to as success rate. All RMSD values are reported in Ångstrom and calculated using the symmetry-aware functionality of sPyRMSD (Meli & Biggin, 2020) which accounts for symmetric chemical groups.

A.6. Additional Results

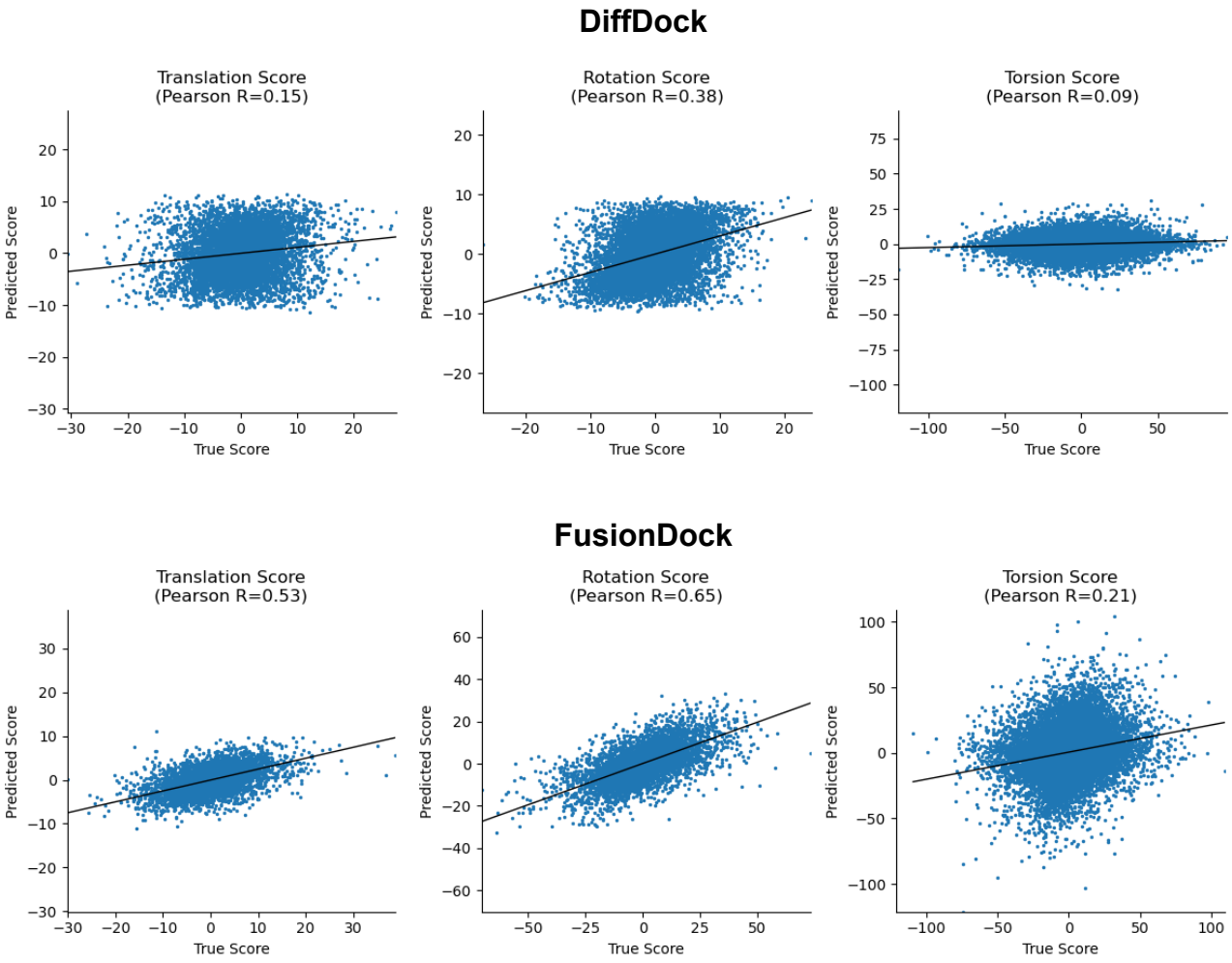


Figure 6. Correlation plots of true and predicted scores from DiffDock and FusionDock on the test set. 10 noised samples were evaluated for each test set system to ensure convergence. It is important to note that the translation diffusion schedule differs between these two methods. Since DiffDock attempts to solve blind docking directly by docking to the full protein structure, their maximum translation sigma is 19Å while FusionDock uses just 6Å. The diffusion schedule for rotations and torsions is the same between both models. It is clear that while translation and rotation scores can reach moderately good correlation (R=[0.53-0.65]), predicting torsion scores remains a larger challenge (R=0.21). We speculate that this is due to the large penalty incurred when the wrong sign (direction) is predicted when the sign is ambiguous (e.g. change in torsion near $\pm\pi$ or when there is molecular symmetry).

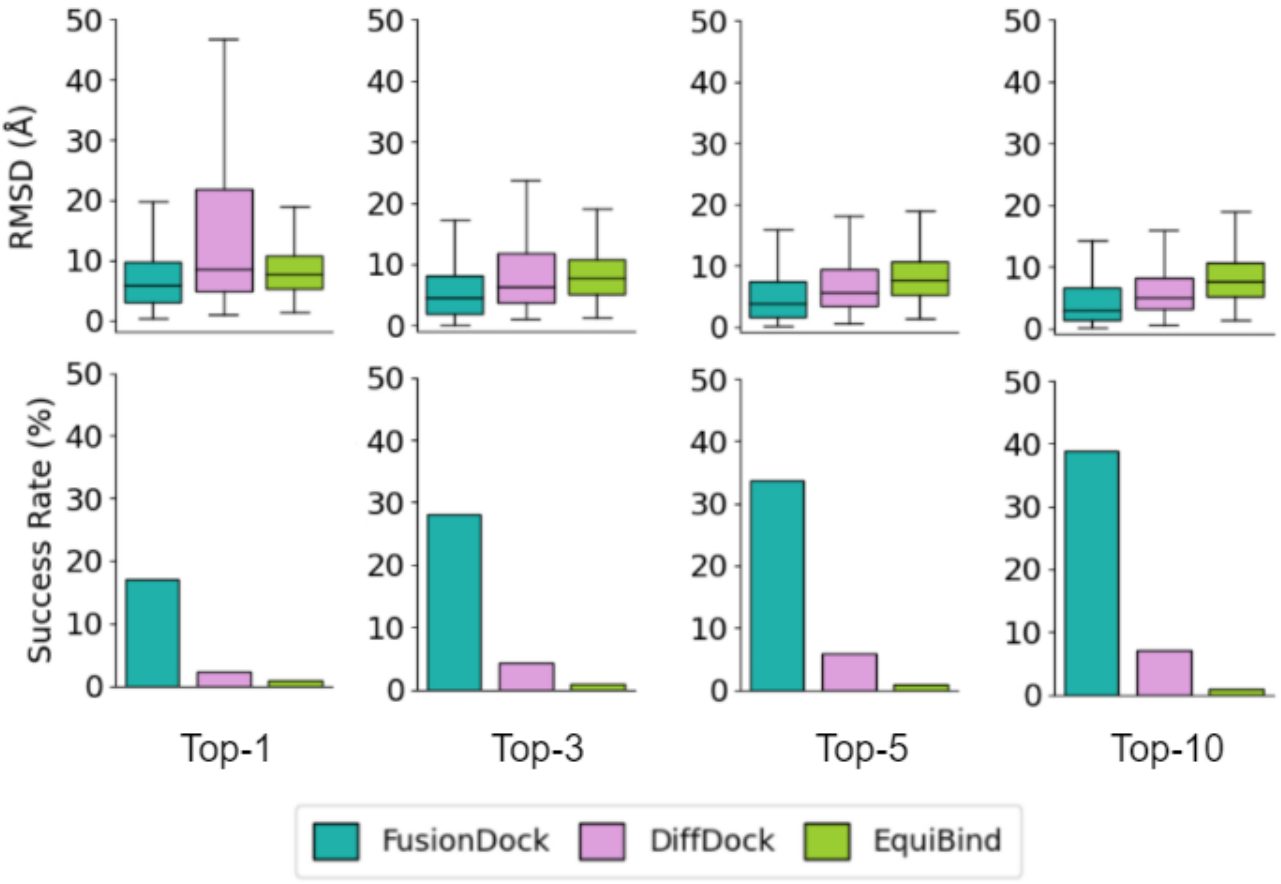


Figure 7. Results for blind docking task on the test set. Top row: RMSD (Å) distribution box plots (outliers not shown). Bottom row: Success rate (Fraction of poses < 2Å). Figure shows results for top-1, top-3, top-5, and top-10 ranked poses from each program. Note that EquiBind only produces a single pose.

Appendix References

Bowers, K. J., Chow, E., Xu, H., Dror, R. O., Eastwood, M. P., Gregersen, B. A., Klepeis, J. L., Kolossvary, I., Moraes, M. A., Sacerdoti, F. D., et al. Scalable algorithms for molecular dynamics simulations on commodity clusters. In *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, pp. 84–es, 2006.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hu, B. and Lill, M. A. Watsite: Hydration site prediction program with pymol interface, 2014.

Humphrey, W., Dalke, A., and Schulten, K. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. Torsional diffusion for molecular conformer generation. *arXiv preprint arXiv:2206.01729*, 2022.

Liu, Y., Grimm, M., Dai, W.-t., Hou, M.-c., Xiao, Z.-X., and Cao, Y. Cb-dock: A web server for cavity detection-guided protein–ligand blind docking. *Acta Pharmacologica Sinica*, 41(1):138–144, 2020.

Madhavi Sastry, G., Adzhigirey, M., Day, T., Annabhimoju, R., and Sherman, W. Protein and ligand preparation: parameters, protocols, and influence on virtual screening enrichments. *Journal of computer-aided molecular design*, 27:221–234, 2013.

Meli, R. and Biggin, P. C. spyrmsd: symmetry-corrected rmsd calculations in python. *Journal of Cheminformatics*, 12(1): 1–7, 2020.

Riniker, S. and Landrum, G. A. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12):2562–2574, 2015.

Schaller, D., Pach, S., and Wolber, G. Pyrod: Tracing water molecules in molecular dynamics simulations. *Journal of Chemical Information and Modeling*, 59(6):2818–2829, 2019.

Steinegger, M. and Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.