JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

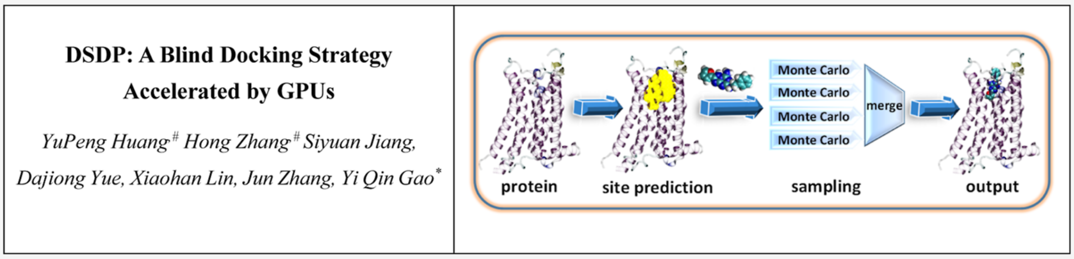
pubs.acs.org/jcim Article

DSDP: A Blind Docking Strategy Accelerated by GPUs

YuPeng Huang,[⊥] Hong Zhang,[⊥] Siyuan Jiang, Dajiong Yue, Xiaohan Lin, Jun Zhang, and Yi Qin Gao[*]

Cite This: J. Chem. Inf. Model. 2023, 63, 4355−4363 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information


DSDP: A Blind Docking Strategy Accelerated by GPUs
YuPeng Huang# Hong Zhang# Siyuan Jiang, Dajiong Yue, Xiaohan Lin, Jun Zhang, Yi Qin Gao*
Monte Carlo
Monte Carlo
Monte Carlo
Monte Carlo
merge
protein
site prediction
sampling
output

ABSTRACT: Virtual screening, including molecular docking, plays an essential role in drug discovery. Many traditional and machine-learning-based methods are available to fulfill the docking task. However, the traditional docking methods are normally extensively time-consuming, and their performance in blind docking remains to be improved. Although the runtime of docking based on machine learning is significantly decreased, their accuracy is still limited. In this study, we take advantage of both traditional and machine-learning-based methods and present a method, deep site and docking pose (DSDP), to improve the performance of blind docking. For traditional blind docking, the entire protein is covered by a cube, and the initial positions of ligands are randomly generated in this cube. In contrast, DSDP can predict the binding site of proteins and provide an accurate searching shape and initial positions for further conformational sampling. The sampling task of DSDP makes use of the score function and a similar but modified searching strategy of AutoDock Vina, accelerated by implementation in GPUs. We systematically compare its performance in redocking, blind docking, and virtual screening tasks with state-of-the-art methods, including AutoDock Vina, GNINA, QuickVina, SMINA, and DiffDock. In the blind docking task, DSDP reaches a 29.8% top-1 success rate (root-mean-squared deviation < 2 Å) on an unbiased and challenging test dataset with 1.2 s wall-clock computational time per system. Its performances on the DUD-E dataset and the time-split PDBBind dataset used in EquiBind, TANKBind, and DiffDock are also evaluated, presenting a 57.2 and 41.8% top-1 success rate with 0.8 and 1.0 s per system, respectively.

■ INTRODUCTION

Molecular docking is a crucial step to generate potential candidates for lead compounds in drug discovery.[1,2] Docking is composed of several steps, for example, binding pocket identification, drug conformations sampling, scoring, and ranking. Generally, the binding pocket is provided by users in redocking, cross-docking, and virtual screening tasks, with the pocket being identified by the cocrystal structure of the target protein and associated ligands in the experiments. However, with the development of protein structure prediction methods, for example, AlphaFold,[3] ColabFold,[4] and RosettaFold,[5] a fast-increasing number of protein structures are generated without information on ligands. Therefore, it is in high demand to perform reliable ligand docking based on protein structures only and without known binding pockets.

Traditionally, blind docking is regarded as a task of docking around the entire protein, and many traditional docking programs are available for such tasks, for example, AutoDock Vina[6] and Glide.[7] It is of great value to improve the docking speed and accuracy, given that, normally, a large space should be sampled in limited searching steps. To deal with such a problem, a number of optimized sampling methods were developed, for instance, QuickVina-W,[8] which was developed based on QuickVina2.[9,10] QuickVina2 optimized the local search frequency by searching only potentially important spatial points. These spatial points are identified by checking the gradients of the scoring function against a thread history before local optimization. QuickVina-W is a program designed for blind docking, and the potentially significant points are identified by examination of the history of the present and other threads. Besides the improvement on the sampling method, another strategy to increase speed and accuracy is to decrease the searching space through an identification of the potential ligand−protein binding pockets. Methods based on both traditional geometrical or machine learning (ML) strategies have been developed to recognize the protein pocket.[11,12] The traditional methods have a relatively long

Received: April 4, 2023
Published: June 30, 2023

https://doi.org/10.1021/acs.jcim.3c00519

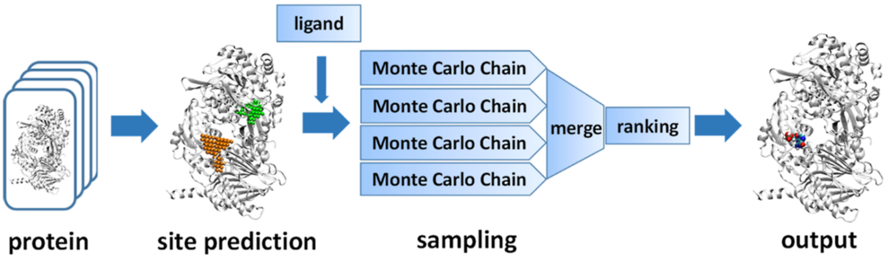


Figure 1. Molecular docking pipeline of DSDP.

history and have observed the development of various strategies. For example, in FunFOLD[13] and COFACTOR,[14] the binding pocket is located by calculations on the similarity between the target and the templates of known pockets. Methods such as Fpocket,[15] on the other hand, are based on an examination of the shape and spatial geometry of the target protein. In another strategy, one performs the binding pocket search using designed probes and identifies the pocket by calculating the interaction energy between the probes and protein.[16] In addition to the traditional methods, the strategies based on ML began to show high performance for binding site prediction over the past few years. Among them, P2Rank[17] is a widely used method based on the random forest algorithm, while COACH[18] is trained by the support vector. In these methods based on deep learning, 3D-convolutional neural networks (CNNs) are often used, as in DeepSite,[19] DeepSurf,[20] and PUResNet.[21]

Besides binding site prediction, many studies focused on combining site recognition, pose sampling, and scoring in one shot to improve the performance of blind docking. EquiBind[22] is a popular method among them, which applies an SE(3)-equivariant geometric deep learning strategy and successfully decreases the runtime of docking to <1 s per system. In addition, TANKBind,[23] another deep learning-based method using trigonometry-aware neural networks, replaces the expensive sampling by evaluation of the protein−ligand interaction energy landscapes of different blocks of protein, which further improves the performance in docking tasks. Recently, another state-of-the-art approach, DiffDock,[24] was reported, which is based on deep learning and treats docking as a generative task. DiffDock uses a diffusion generative model to generate conformations and applies a confidence model to estimate the poses. This method enjoys a significant improvement in docking accuracy, representing a powerful intermediate approach between traditional sampling and one-shot prediction.

The score function, which is commonly used to estimate the confidence of ligand binding poses, is another important factor affecting the accuracy of blind docking. There are four main categories of scoring functions, namely, physics-based, knowledge-based, empirical, and ML-based scoring functions. Many efforts have been invested in improving the performance of score functions, for instance, SMINA,[25] GNINA,[26,27] RF-Score,[28] and IGN.[29] Most of these methods are based on linear regression or ML and present a reasonable performance in estimating the interactions between the proteins and ligands. However, most of the ML-based strategies are not introduced directly into the molecular docking procedure in the form of the scoring function but are used to rescore the poses of ligands generated by the traditional sampling methods. Because a high computational cost is required when the network is used to guide the sampling, implementing a rescoring process after the sampling is a common strategy to improve the accuracy of the latter, as in GNINA.

In the present work, to improve the speed and accuracy of blind docking, we developed a method, deep site and docking pose (DSDP), to combine the advantages of both ML and traditional sampling strategies. It predicts the binding site on the protein and provides the potential location of ligands to decrease the searching space for the following binding pose sampling. A similar strategy was used in EquiBind,[22] DiffDock,[24] and Uni-Dock.[30] In these protocols, the binding site identification and ligand conformation sampling are treated separately, and only the predicted site center is used in the sampling step by ignoring the shape of the binding pocket. In the present work, the geometrical information of predicted pockets is used to guide the sampling, including the initial positions and searching space of ligands. In our method, the sampling is not restricted to a fixed cube but to an adaptive space provided by the binding site predicting step, which efficiently reduces the search space. In addition to the improvement in the accuracy and speed of binding site prediction, GPU acceleration is introduced to further speed up the sampling process. As a mixed method, DSDP provides an integrated docking workflow, which takes advantage of both traditional and ML-based methods and is able to perform blind docking with both high accuracy and speed.

■ METHODS

DSDP is an integrated docking workflow which makes use of ML and traditional sampling strategies, the molecular docking pipeline of which is shown in Figure 1.

Protein−Ligand Binding Site Prediction. The first step of DSDP is the protein−ligand binding site prediction, which is used to provide an accurate search space for the follow-up binding pose sampling. The workflow of this step is a modified realization of the state-of-the-art binding site prediction method, PUResNet.[21] We introduced several modifications to the original method to further improve the performance of binding site prediction and more importantly to facilitate the downstream binding pose sampling. In the following part, we provide a brief description of PUResNet as well as a comparison between DSDP and PUResNet. For more details on PUResNet, the readers are referred to the original publication of Chong and co-workers.[21]

PUResNet is a binding site prediction program making use of a 3D-CNN architecture. The protein is represented by 36 × 36 × 36 voxels at its center with 70 Å on each side, and 18 features are introduced to represent each protein atom. The 18 features include 9 bits encoding atom types (B, C, N, O, P, S, Se, halogen, and metal), 9 atomic features (hybridization, heavy atoms, heteroatoms, hydrophobic, aromatic, partial charge, acceptor, and donor, ring). The 36 × 36 × 36 × 18 voxels are used to represent the protein structure, which is fed into the CNN. The binding site prediction is regarded as a binary segmentation problem, and the binding site can be represented by 36 × 36 × 36 × 1 voxels at the protein center. The value of the fourth dimension (1 or 0) is used to represent whether this point is a binding site or not. PUResNet combines the classical U-Net and ResNet, and it uses the dice loss function. In addition, the training data of PUResNet is a filtered subset of scPDB.[31] It should be noted that the binding site (cavity6.mol2) of the scPDB database is generated by the Volsite tool of the IChem program.[32]

The main differences between DSDP and PUResNet are summarized as follows: first, the learning target in DSDP is taken as the position of the ligand-heavy atoms and not the binding site in the scPDB database generated by Volsite as used in PUResNet. Such a choice is made to meet the requirement that an accurate position of the ligand is needed in the downstream binding pose sampling. This treatment also allows all databases including the protein and ligand positions to be used as the training data, rather than relying on the scPDB database. Second, the 18 chemical features are also modified, with the elements Se and S merged into one channel. Since the Se element is relatively rare in proteins and has a property close to S, to lower the computational cost and at the same time maintain a code structure similar to PUResNet, we merged them into one channel. The newly vacated channel is used to represent the surface of the protein, given that the surface of the protein is known to play an important role in identifying the protein pockets. Along with this feature, we also provided a surface-identifying package (named surface_tool). To speed up the coordinate reading and feature generation, which in PUResNet rely on Openbabel[33] and are time-consuming, DSDP reads the protein coordinates by NumPy[34] and loads the features generated by a newly developed toolkit, protein_feature_tool. In protein_feature_tool, we first generated a permanent list by Openbabel outside the inference of site prediction, which includes features of common protein atoms distinguished through residue and atom types (atom names defined in the default Protein Data Bank format; for example, the atom types of ALA are N, CA, C, O, CB, H, HA, HB1, HB3, and HB2). This tool can directly and quickly generate the protein atom feature by searching the atom type from the preprepared list. Third, we added one extra term to the loss function to the original form of PUResNet, which is the distance between the position of the max score and the center of the ligand. The weight of this term used in this study is 0.1. We introduced this term to bring the output position of the max score closer to the reference center, given that the location of the sampling center is crucial for the accuracy of searching. The final step in binding site prediction is extracting the pockets from the voxels with different scores. In the original PUResNet, a cutoff of the score value of 0.5 and a minimum pocket size of 50 Å^3 were used. To exhaustively search for all potential binding pockets, we extracted and clustered the 36 × 36 × 36 × 1 voxels and selected the top 200 points with high scores. In this way, a series of discrete points describing the binding sites can be provided in the docking stage. After these modifications, we were able to improve the accuracy of binding site prediction (see Figure 6 and further discussion later).

Binding Pose Search by Traditional Sampling Methods. The second component of DSDP is a traditional sampling process accelerated by GPUs, which is similar to AutoDock Vina combined with a number of modifications to the original program. Because the score function of AutoDock Vina was shown to have reached a good balance of accuracy and simplicity during traditional docking, we chose to use the same scoring function in DSDP. The terms of this function are as follows

$$\mathrm{gauss}_1(d) = \mathrm{e}^{-\left(\frac{d}{0.5\,\text{Å}}\right)^2}$$

$$\mathrm{gauss}_2(d) = \mathrm{e}^{-\left(\frac{(d-3\,\text{Å})}{2\,\text{Å}}\right)^2}$$

$$\mathrm{repulsion}(d) = \begin{cases} d^2, & \text{if } d < 0\,\text{Å} \\ 0, & \text{if } d \geq 0\,\text{Å} \end{cases}$$

$$\mathrm{hydrophobic}(d) = \begin{cases} 1, & \text{if } d < 0.5\,\text{Å} \\ -d + 1.5, & \text{if } 0.5\,\text{Å} \leq d \leq 1.5\,\text{Å} \\ 0, & \text{if } d > 1.5\,\text{Å} \end{cases}$$

$$\text{hydrogen bonding}(d) = \begin{cases} 1, & \text{if } d < -0.7\,\text{Å} \\ -\frac{10}{7}d, & \text{if } -0.7\,\text{Å} \leq d \leq 0\,\text{Å} \\ 0, & \text{if } d > 0\,\text{Å} \end{cases}$$

$$\begin{aligned} s = {} & (w_1 \times \mathrm{gauss}_1(d) + w_2 \times \mathrm{gauss}_2(d) \\ & + w_3 \times \mathrm{repulsion}(d) + w_4 \times \mathrm{hydrophobic}(d) \\ & + w_5 \times \text{hydrogen bonding}(d))/(1 + w_6 \times N_{\mathrm{rot}}) \end{aligned}$$

where $d = r_{ij} - R_i - R_j$, r_{ij} is the distance between atoms i and j, and R_i and R_j are their van der Waals radii, respectively. The values of w_1, w_2, w_3, w_4, w_5, and w_6 are −0.0356, −0.00516, 0.840, −0.0351, −0.587, and 0.0585, respectively.

Compared to the original Vina docking process, we made three major modifications, as discussed in the following.

First, Vina uses a grid-based method for energy evaluation to reduce the computational cost of protein−ligand interactions.[35] Every vertex of the grid stores a score as a summation over the ligand interactions with protein atoms within the cutoff distance. In this way, the score of a ligand atom can be calculated by trilinear interpolation and its gradient is directly calculated by the partial derivative of the trilinear interpolation. Considering that the gradient is not continuous on the entire interpolation area, we built protein grids that not only contain scores but also their gradients to increase the computational accuracy. The gradient of a ligand atom is also calculated by trilinear interpolation and therefore is of high accuracy (see Figure S1 in the Supporting Information).

Second, a quasi-Newton method, Broyden−Fletcher−Goldfarb−Shanno (BFGS),[36] was used for the local optimization of sampled conformations in Vina. We replaced BFGS with the Barzilai−Borwein (BB) method.[37] Unlike BFGS, BB only needs a vector inner product to estimate the step length and does not need to perform a line search. This feature makes it well parallelized on GPUs. Our testing shows that 100−150 iteration steps are enough to optimize one structure with the BB method (see Figure 2).

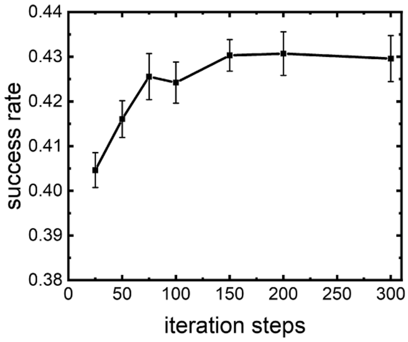


Figure 2. Correlation of the success rate and the iteration steps.

Third, the search space of Vina is chosen to be a cube defined by users, and the initial position of the ligand is randomly generated within this box. In the blind docking task, since the entire protein is included in the cube, it is likely that one has to search in a very large sampling space, which can result in low accuracy. To make full use of information provided by binding site prediction, we used an accurate searching space in the following sampling strategy provided by DSDP. This protocol generates a peripheral 30.0 × 30.0 × 30.0 Å^3 cube surrounding the predicted box center for the trilinear interpolation. A union set of balls with a 7 Å radius, centered at each discrete point of the predicted binding site, is generated (see Figure 3). The intersection of this cube and the union set of balls is taken as the space for binding site search. The initial ligand positions are not fully random but restrained to the searching space. The space of sampling is thus not cubic but has the adaptive shape as described below (see Figure 3). This choice of the searching space makes it closely resemble the shape of the cavity and results in a reduced searching volume than the cube.

As for the parallelization on GPUs, to make full use of the parallel computing advantages of GPUs, we use a large number of copies (128−2048) and short search steps (20−200) in Monte Carlo searching. This strategy was also used in Vina-GPU[38] and Uni-Dock.[39] The original version of Vina suggests 8 to 64 copies, but each copy would undergo a search of 10^4−10^5 steps. To avoid the performance loss caused by data synchronization, we used the ability of asynchronous concurrent execution provided by CUDA. The initialization and the preparation of copies are run on the CPU, which are then packaged and uploaded to the GPU. Each copy undergoes Monte Carlo local searching and optimization independently and asynchronously on GPUs. The resulting binding poses are postprocessed in the CPU. The workflow of the parallelization on GPUs is shown in Figure 4.

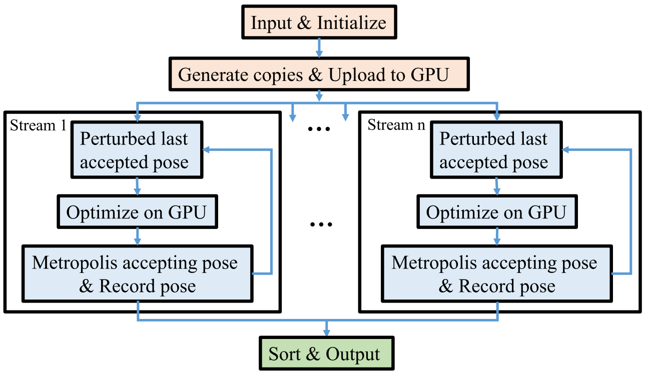

Input & Initialize
Generate copies & Upload to GPU
Stream 1
Perturbed last accepted pose
Optimize on GPU
Metropolis accepting pose & Record pose
...
Stream n
Perturbed last accepted pose
Optimize on GPU
Metropolis accepting pose & Record pose
Sort & Output

Figure 4. Workflow of the parallelization on GPUs. The initialization and the postprocessing of sampled conformations are finished outside the GPUs.

Dataset Preparation. Considering that many ML-based docking methods are trained based on the PDBBind database,[40] it would be convenient to compare with other methods using a unified and high-quality training dataset. Therefore, we trained the binding site prediction module using the PDBBind v2020 preprocessed in EquiBind,[22] which can be downloaded from https://zenodo.org/record/6408497. It should be mentioned that DSDP includes two main parts, binding site prediction by ML and sampling using traditional methods. The binding site prediction based on ML is only included in the blind docking task, and all training set is only used to train this part. For the test dataset for redocking and blind docking tasks, EquiBind, TANKBind,[23] and DiffDock[24] all used a new time-split PDBBind dataset. We found that proteins in this latter dataset overlap significantly with the PDBBind v2020 preprocessed dataset (training dataset), which could potentially introduce a bias to the results. To test and compare our method with other state-of-the-art methods in a more rigorous way, we preprocessed a new and unbiased dataset to estimate the performance of methods. This latest test dataset is a subset of the scPDB database without duplicates, in which proteins and ligands do not overlap with PDBBind v2020 (see Figure 5). We unified the proteins and ligands by the Uniprot ID and SMILES, respectively. A dataset-cleaning process was carried out after we obtained the final complex list. In the preparation of structures, water and metal ions were removed from proteins and ligands. The protein chains were then selected if any atom of them is within a 10 Å radius of any ligand atom. Openbabel,[33] RDkit,[41] Reduce,[42] and ChimeraX[43] were used in the dataset-cleaning process.

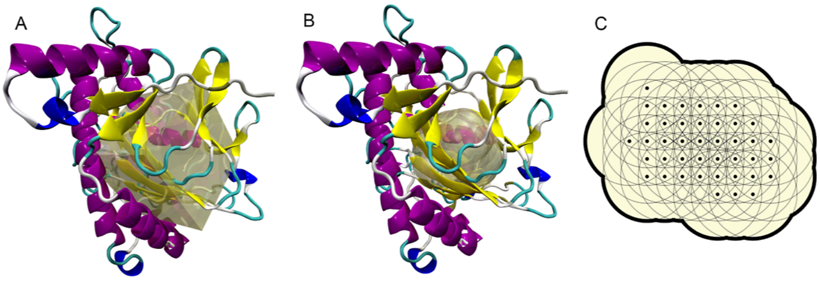


Figure 3. Schematic sampling space of (A) traditional cube and (B) adaptive shape. (C) Schematic diagram of the union set of balls with a fixed radius with each discrete point of predicted binding site as the center.

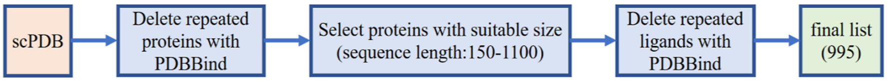

scPDB
Delete repeated proteins with PDBBind
Select proteins with suitable size (sequence length:150-1100)
Delete repeated ligands with PDBBind
final list (995)

Figure 5. Workflow of filtering the final test dataset from the original scPDB database. As a result of this procedure, we constructed a dataset containing 995 ligand−protein complexes.

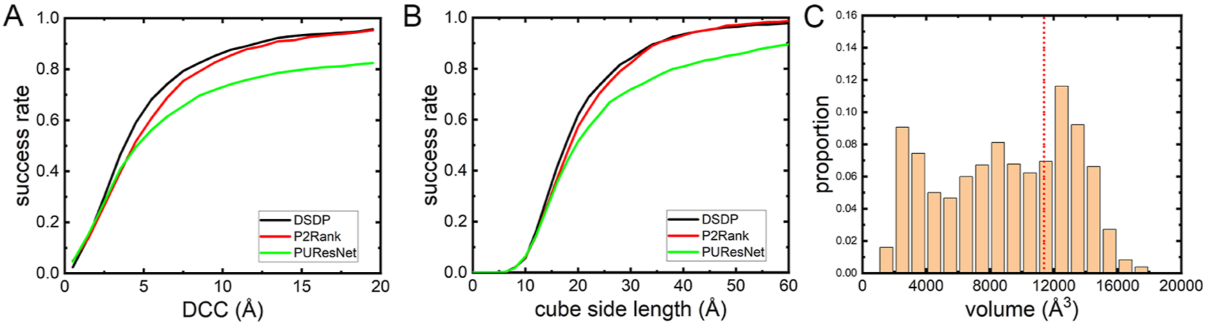


Figure 6. Estimation of (A) DCC and (B) VCR of DSDP, PUResNet, and P2Rank. PUResNet and DSDP use all outputs (the average output number of pockets is no more than 2 per system). P2Rank provides many pockets with different scores, and we only use the top 2 outputs to compare with PUResNet and DSDP. The definition of success in DCC is that the distance between any pocket and the ligand center is within a specific value (the number of the *X* axis). The definition of success in VCR is that the cube with a specific length can cover the entire ligand. (C) Volume proportion of the adaptive shape formed by the union set of balls with a 7 Å radius, and the red dotted line is the volume of 22.5 × 22.5 × 22.5 Å^3 cube.

To directly compare DSDP with other methods, we also estimate the DSDP performance based on public benchmarks. In the binding site prediction part, we chose the COACH420 dataset used in P2Rank[17] and DeepSurf.[20] The DUD-E dataset and the time-split PDBBind dataset used in EquiBind,[22] TANKBind,[23] and DiffDock[24] were used to test the performance of blind docking. The virtual screening tasks were also undertaken using DUD-E database. To choose an unbiased dataset in this part, we used aldr, hxk4, kith, and sahh systems in the DUD-E dataset, the proteins of which are not included in our training database of binding site prediction.

Baseline Setup. In order to estimate the accuracy of the binding site prediction of DSDP, we compared its binding site prediction function with that of P2Rank and PUResNet. P2Rank and PUResNet programs were run using the provided models provided by the authors. The benchmark presented in the P2Rank paper[17] was also obtained to compare with our data directly. To estimate the sampling performance of DSDP, we compared the docking speed and accuracy with AutoDock Vina, QuickVina-W (QuickVina2), GNINA, SMINA, and DiffDock. We followed the protocol suggested by QuickVina, and QuickVina-W and QuickVina2 were used to do blind docking and redocking, respectively. For the traditional docking programs AutoDock Vina, QuickVina-W (QuickVina2), GNINA, and SMINA, we found that the parameter exhaustiveness affects strongly the runtime and performance. Therefore, we used three different values, 8, 32, and 64, in different tasks. Three parallel redocking and blind docking calculations in the unbiased dataset were performed in the present work. DiffDock was performed using the default hyperparameters but with a batch size of 1. The docking boxes were centered on the cocrystallized ligand and the entire protein with 4 Å added in each dimension for redocking and blind docking (including virtual screening) tasks, respectively, in AutoDock Vina, QuickVina-W (QuickVina2), GNINA, and SMINA. The heavy-atom root-mean-squared deviation (RMSD) was calculated to estimate the accuracy of docking. To unify the hardware and exhaust the best parallelism performance of Vina-based methods, we used 64 or 32 CPU cores (Intel Xeon Platinum 8358) for AutoDock Vina, QuickVina, and SMINA in the redocking, blind docking, and virtual screening tasks, respectively. It should be mentioned that 64 or 32 CPU cores are not fully utilized when the exhaustiveness is set to 8. For the GPU-based methods, namely, GNINA, DSDP, and DiffDock, we used NVIDIA GeForce RTX 2080 SUPER GPU (AMD Ryzen 7 2700X Eight-Core Processor).

■ RESULTS AND DISCUSSION

Protein−Ligand Binding Site Prediction. To evaluate the performance of protein−ligand binding site prediction of DSDP, we compared our method with PUResNet and P2Rank. We mainly used two parameters to evaluate the performance of our model, which are highly related to the downstream sampling, namely, the distance between the predicted binding site center to the center of the actual ligand (DCC) and the volume coverage rate (VCR). The former quantity determines the sampling position of ligands, and the latter is explained as follows. In the molecular docking process, a restrained wall is needed to ensure that the docking is only performed in a restricted spatial region. At the same time, the searching space needs to cover the entire ligand, namely, the VCR of the searching space to the actual ligand, should be 1. As shown in Figure 6A, we find that DSDP has a higher success rate of DCC than the other methods. This result indicates that DSDP provides a more accurate location of the sampling center. We then calculated the VCR using cubes of different side lengths, and the cube center is the predicted binding site center. As depicted in Figure 6B, DSDP again shows the highest success rate of VCR among all methods tested. Considering that the searching space of DSDP is an adaptive shape rather than a cube, we compared the size of volume based on the adaptive volume with the 22.5 × 22.5 × 22.5 Å^3 cube, which is widely used in the redocking task.[6,39] We found that the adaptive shape is considerably smaller than the cube (red line in Figure 6C). This comparison indicates that using the adaptive shape

Table 1. Redocking of the Test Dataset[a]

methods	RMSD < 2 Å (%)	RMSD < 5 Å (%)	time (s)	sampling
Vina	43.2 ± 0.4	70.6 ± 0.8	36.0[b]	exhaustiveness = 8
	44.3 ± 0.1	70.9 ± 0.4	70.6[b]	exhaustiveness = 64
QuickVina2	41.3 ± 0.5	68.8 ± 0.3	6.9[b]	exhaustiveness = 8
	43.2 ± 0.5	69.8 ± 0.5	7.7[b]	exhaustiveness = 64
SMINA	42.2 ± 0.1	69.8 ± 0.4	18.7[b]	exhaustiveness = 8
	44.1 ± 0.2	70.7 ± 0.4	22.9[b]	exhaustiveness = 64
GNINA	54.0 ± 0.1	84.1 ± 1.0	23.0[c]	exhaustiveness = 8
	55.1 ± 0.4	85.0 ± 0.4	100.0[c]	exhaustiveness = 64
DSDP	39.1 ± 0.2	74.3 ± 0.6	0.33[c]	128 × 20
	42.5 ± 0.3	74.6 ± 0.4	0.55[c]	384 × 40
	42.5 ± 0.6	73.6 ± 0.6	0.64[c]	512 × 40
	43.5 ± 0.7	72.4 ± 0.2	1.71[c]	512 × 200
	43.7 ± 0.6	71.2 ± 1.0	3.28[c]	1024 × 200
	43.8 ± 0.3	71.1 ± 0.3	6.32[c]	2048 × 200

[a]The runtime of DSDP does not include the time of protein binding pocket prediction in the redocking task. [b]64 CPU cores were used for Vina, QuickVina2, and SMINA. [c]GPU was used in GNINA and DSDP.

Table 2. Blind Docking of the Test Dataset[a]

methods	RMSD < 2 Å (%)	RMSD < 5 Å (%)	time (s)	sampling
Vina	12.9 ± 0.5	26.8 ± 1.7	56.5[b]	exhaustiveness = 8
	24.6 ± 0.3	39.4 ± 0.4	93.9[b]	exhaustiveness = 64
QuickVina-W	10.6 ± 0.9	23.0 ± 1.2	11.5[b]	exhaustiveness = 8
	21.2 ± 0.1	35.8 ± 0.2	13.5[b]	exhaustiveness = 64
SMINA	12.4 ± 0.3	26.1 ± 1.8	22.3[b]	exhaustiveness = 8
	23.9 ± 0.5	39.1 ± 0.4	24.0[b]	exhaustiveness = 64
GNINA	25.6 ± 1.5	39.2 ± 2.0	33.5[c]	exhaustiveness = 8
	37.7 ± 0.7	53.7 ± 0.1	135.1[c]	exhaustiveness = 64
DiffDock	17.6 ± 0.6	43.5 ± 0.2	86[c]	
DSDP (cube)	28.4 ± 0.6	46.2 ± 0.3	1.26[c]	384 × 40
DSDP (shape)	29.8 ± 0.3	47.9 ± 0.1	1.20[c]	384 × 40

[a]The runtime of DSDP includes the time of protein binding pocket prediction in the blind docking task. [b]64 CPU cores were used for Vina, QuickVina2, and SMINA. [c]GPU was used in GNINA, DiffDock, and DSDP.

in the downstream sampling can effectively reduce the search over futile space.

To further estimate the DSDP performance of binding site prediction and make a direct comparison with the benchmark reported in the P2Rank method,[17] including DeepSite, MetaPocket, SiteHound, and Fpocket, we introduced another criterion, distance from the binding site center to the closest ligand atom (DCA). The dataset COACH420 was used in this test. Although DCA can describe the accuracy of pocket prediction, it is not rigorous enough compared with DCC. The latter also matches better than the former to the following sampling task, which requires the pocket to cover the entire ligand. As shown in Table S1 and Figure S2, the DCC success rate of DSDP is 46.8%, which is higher than P2Rank and PUResNet. For the DCA criterion, DSDP still performs better than other tools with the exception of P2Rank. These results suggest that DSDP can accurately predict the binding site pocket center, laying the ground for the following binding configuration sampling.

Redocking in the Known Site. DSDP is not only used for blind docking but also for the redocking task. Before the blind docking test, in the following, we first verify the docking accuracy based on tests on cases with known binding sites. The known binding site is defined by the position of the cocrystallized ligand, and the box size is obtained by adding 4 Å along the negative and positive directions for the minimum and maximum of the x, y, and z coordinates of the ligand, respectively. As listed in Table 1, we performed ligand docking using AutoDock Vina, QuickVina2, GNINA, and SMINA with a different number of copies (the parameter exhaustiveness), as well as DSDP. It should be mentioned that the number of sampling steps of Monte Carlo in the Vina-based methods varies with the number of atoms and degree of freedom of the ligands, which is around 10^4 to 10^5 steps. QuickVina2 optimized the search strategy to reduce the number of required local search steps. In practice, the numbers of copies and searching steps are balanced to jointly optimize the computational speed and accuracy. For Vina, the accuracy changes slightly but the runtime increases significantly when one increases the sampling copies from 8 to 64. Such an observation indicates that the increase in the sampling steps does not bring much benefits to the redocking task because the sampling space is limited. The runtime of QuickVina2 is shorter than the other Vina-based methods. On the other hand, GNINA enjoys a higher accuracy than the methods mentioned above when the same sampling parameters are used, thanks to the rescoring step after the sampling. Rescoring helps extract the lower RMSD conformation of the ligand from the top 50 poses generated by the traditional sampling method SMINA independent of the sampling process. For the implementation of DSDP, we found that satisfactory performance is obtained when the numbers of copies and searching steps are set to 384 and 40, respectively, which are far fewer than the sampling numbers required by the Vina-based methods mentioned above. Although the sampling number is small, the accuracy of DSDP in the redocking task (the RMSD of top-1 pose within 2 Å) matches the Vina-based methods using 8 copies. Notably, the runtime of DSDP (384 × 40) is much less than the other methods, showing a speed-up of ~100 times compared to the original Vina using 64 CPU cores with the exhaustiveness = 64.

Blind Docking. To evaluate the performance of DSDP on blind docking, we also used our preprocessed dataset as the test set. As shown in Table 2, we found that the number of sampling copies affects the accuracy of blind docking in traditional methods. For example, AutoDock Vina presents a 12.9% success rate (RMSD < 2 Å for the top-1 poses) with 8 sampling copies, while this rate increases to 24.6% when the parameter exhaustiveness is increased to 64. This behavior of the model can also be found in other traditional methods. However, further increasing this parameter is not recommended due to the fast increase of the demands on hardware and runtime. AutoDock Vina and SMINA yield similar results when the parameters of configuration used for them are the same as expected since they share the same sampling strategy. Compared to these methods, QuickVina-W uses fewer local searching steps but has a slightly decreased success rate, namely, 10.6 and 21.2% for 8 and 64 sampling copies, respectively. GNINA again shows the best accuracy, benefiting from the rescoring process. The performance of DiffDock on this new dataset is less optimal compared with what is seen in their original paper. To compare the influence of the choice of test sets, we used the same parameters of DiffDock in the present work to study the test set (363 complexes) used in the original paper.[24] We obtained a 36.5% success rate (see Table S2 in the Supporting Information), which reproduces the results of ref 24. The success rate of DSDP on the 363 datasets is 41.8%, which is higher than that of DiffDock. The comparison of the DUD-E dataset between different methods is also provided in Table S3, and the success rate on this dataset is higher than the other datasets. These results indicate that our test set (995 complexes) represents a more challenging task than the 363 complexes and DUD-E datasets. In addition, we examined the effects of the search space shape (cube and adaptive shape) on the speed and accuracy of blind docking using DSDP. The size of the search cube is 22.5 × 22.5 × 22.5 Å^3 using the predicted binding site as the center, and the adaptive shape is obtained by a union set of balls with a 7 Å radius (see more details in Methods). As shown in Table 2, we found that the docking accuracy of adaptive shapes is slightly higher than that of cubes (22.5 × 22.5 × 22.5 Å^3). In addition, the runtime based on the adaptive shape (1.20 s) is shorter than that for the cubic one (1.26 s). Such a difference can be readily rationalized since the former samples have a more relevant space than the latter. We found that although the combination of searching steps and copies for DSDP is far less than the Vina-based method, the success rate of the former (29.8%) is significantly higher than the others except for GNINA using 64 sampling copies. This result indicates that the binding site prediction used in DSDP can not only pinpoint the sampling position but also decrease the sampling space.

Virtual Screening. Virtual screening is the most common application of molecular docking. To estimate the DSDP performance in the virtual screening task, we used the entire DUD-E database to provide a benchmark of our method against QuickVina-W, which is specially designed for blind docking. As shown in Figure 7, we can see that the AUC median of DSDP is notably higher than that of QuickVina-W, although a small number of systems failed to predict the binding site and induce a low AUC (the raw data of this result is shown in Figure S4). In addition, we used four systems in the DUD-E database as a subset to test all methods, namely, aldr, hxk4, kith, and sahh, the proteins of which are not included in our training dataset. As shown in Table S5, the AUC value of DSDP is higher than SMINA, AutoDock Vina, and QuickVina-W in all systems and is the highest in the aldr system. However, the performance of GNINA is the best of the other three systems. This result indicates that rescoring of poses can not only improve the accuracy of redocking and blind docking but is also effective in the virtual screening task. The performances of AutoDock Vina, and QuickVina-W are basically the same, and the AUC value of SMINA improves slightly, benefiting from its score function. In addition, we compared the runtime of these methods; as shown in Table S5, DSDP on one GPU (NVIDIA GeForce RTX 2080 SUPER) is 30 to 50 times faster than the original Vina using 32 CPU cores with the exhaustiveness = 32. These results indicate that the satisfactory performance of DSDP in virtual screening tasks, while a precise score function and a following rescoring section are also important to improve the accuracy, which is the focus of our next study.

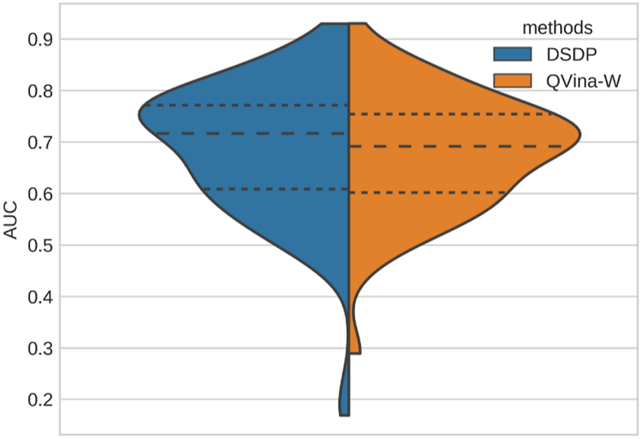


Figure 7. Violin plots of AUC. All 102 targets from the DUD-E dataset were selected and used to compare DSDP and QuickVina-W. The search boxes for QuickVina-W were centered on the entire protein with 4 Å added in every dimension.

CONCLUSIONS

To improve the performance of molecular docking, many ML-based methods have been introduced into the docking tasks, leading to significant successes in the prediction of protein binding sites and estimation of protein–ligand interaction. However, ligand pose sampling is still a challenge for these methods, partly because the available training data of ligand binding poses is highly limited. On the other hand, the traditional sampling methods have observed great development over the past 20 years, although the calculation efficiency is normally low. In the present work, we try to take advantage of both ML-based binding site prediction and traditional sampling methods and compile an accelerated two-task-in-one program on GPUs. The method is named DSDP. DSDP is an integrated docking program developed for blind docking, which can also be used for redocking and virtual screening. To compare DSDP with other state-of-the-art methods, we preprocessed a new and rigorous dataset that has no ligand or protein overlap with the training dataset (PDBBind v2020). DSDP was shown to reach a 29.8% top-1 success rate (RMSD < 2 Å) on this dataset with a computational time of about 1.20 s per system, thus enjoying both high calculation efficiency and high accuracy. DSDP makes full use of the output of binding site prediction, in which the sampling space and the initial positions of ligands are guided to adapt to the pocket. In addition, the binding pose sampling is significantly accelerated by implementation on GPUs, resulting in a speed-up of about 100 times compared to the original Vina using 64 CPU cores with the exhaustiveness = 64 for both redocking and blind docking. The DSDP also performs well in virtual screening, indicating that DSDP is suitable for large-scale applications. Therefore, this integrated program has the benefits of both traditional and ML-based methods, and we hope that it will prove to be a powerful program for future molecular docking studies.

■ ASSOCIATED CONTENT

SI Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.3c00519.

> Introduction of trilinear interpolation, binding site prediction results, blind docking results of time-split PDBBind and DUD-E dataset, virtual screening results, and SMILES of molecules in our well-curated test dataset (ZIP)

■ AUTHOR INFORMATION

Corresponding Author

Yi Qin Gao − *College of Chemistry and Molecular Engineering, Peking University, Beijing 100871, China; Biomedical Pioneering Innovation Center, Peking University, Beijing 100871, China;* orcid.org/0000-0002-4309-9376; Email: gaoyq@pku.edu.cn

Authors

YuPeng Huang − *College of Chemistry and Molecular Engineering, Peking University, Beijing 100871, China;* orcid.org/0000-0002-9978-2393

Hong Zhang − *College of Chemistry and Molecular Engineering, Peking University, Beijing 100871, China;* orcid.org/0000-0002-3303-5109

Siyuan Jiang − *College of Chemistry and Molecular Engineering, Peking University, Beijing 100871, China*

Dajiong Yue − *Huawei Building, Beijing 100085, China*

Xiaohan Lin − *College of Chemistry and Molecular Engineering, Peking University, Beijing 100871, China*

Jun Zhang − *Beijing Changping Lab, Beijing 102200, China;* orcid.org/0000-0002-8760-6747

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.3c00519

Author Contributions

$^{\perp}$Y.H. and H.Z. contributed equally to this work.

Notes

The authors declare no competing financial interest.

The source code of DSDP and our well-curated test dataset are available at https://github.com/PKUGaoGroup/DSDP.git.

■ ACKNOWLEDGMENTS

The authors thank the National Natural Science Foundation of China (22050003 and 92053202).

■ REFERENCES

(1) Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; Zhao, S. Applications of Machine Learning in Drug Discovery and Development. *Nat. Rev. Drug Discovery* **2019**, *18*, 463−477.

(2) Saikia, S.; Bordoloi, M. Molecular Docking: Challenges, Advances and Its Use in Drug Discovery Perspective. *Curr. Drug Targets* **2019**, *20*, 501−521.

(3) 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.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596*, 583−589.

(4) Mirdita, M.; Schütze, K.; Moriwaki, Y.; Heo, L.; Ovchinnikov, S.; Steinegger, M. ColabFold: Making Protein Folding Accessible to All. *Nat. Methods* **2022**, *19*, 679−682.

(5) Baek, M.; DiMaio, F.; Anishchenko, I.; Dauparas, J.; Ovchinnikov, S.; Lee, G. R.; Wang, J.; Cong, Q.; Kinch, L. N.; Schaeffer, R. D.; Millán, C.; Park, H.; Adams, C.; Glassman, C. R.; DeGiovanni, A.; Pereira, J. H.; Rodrigues, A. V.; Van Dijk, A. A.; Ebrecht, A. C.; Opperman, D. J.; Sagmeister, T.; Buhlheller, C.; Pavkov-Keller, T.; Rathinaswamy, M. K.; Dalwadi, U.; Yip, C. K.; Burke, J. E.; Garcia, K. C.; Grishin, N. V.; Adams, P. D.; Read, R. J.; Baker, D. Accurate Prediction of Protein Structures and Interactions Using a Three-Track Neural Network. *Science* **2021**, *373*, 871−876.

(6) Trott, O.; OlsonVina, A. J. AutoDock Improving the Speed and Accuracy of Docking with a New Scoring Function, Efficient Optimization, and Multithreading. *J. Comput. Chem.* **2010**, *31*, 455−461.

(7) 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.; Shenkin, P. S. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *J. Med. Chem.* **2004**, *47*, 1739−1749.

(8) Hassan, N. M.; Alhossary, A. A.; Mu, Y.; Kwoh, C. K. Protein-Ligand Blind Docking Using QuickVina-W with Inter-Process Spatio-Temporal Integration. *Sci. Rep.* **2017**, *7*, 15451.

(9) Alhossary, A.; Handoko, S. D.; Mu, Y.; Kwoh, C. K. Fast, Accurate, and Reliable Molecular Docking with QuickVina 2. *Bioinformatics* **2015**, *31*, 2214−2216.

(10) Handoko, S. D.; Ouyang, X.; Su, C. T. T.; Kwoh, C. K.; Ong, Y. S. QuickVina: Accelerating AutoDock Vina Using Gradient-Based Heuristics for Global Optimization. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2012**, *9*, 1266−1272.

(11) Macari, G.; Toti, D.; Polticelli, F. Computational Methods and Tools for Binding Site Recognition between Proteins and Small Molecules: From Classical Geometrical Approaches to Modern Machine Learning Strategies. *J. Comput.-Aided Mol. Des.* **2019**, *33*, 887−903.

(12) Zhao, J.; Cao, Y.; Zhang, L. Exploring the Computational Methods for Protein-Ligand Binding Site Prediction. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 417−426.

(13) Roche, D. B.; Buenavista, M. T.; McGuffin, L. J. The FunFOLD2 Server for the Prediction of Protein−Ligand Interactions. *Nucleic Acids Res.* **2013**, *41*, W303−W307.

(14) Roy, A.; Yang, J.; Zhang, Y. COFACTOR: An Accurate Comparative Algorithm for Structure-Based Protein Function Annotation. *Nucleic Acids Res.* **2012**, *40*, W471−W477.

(15) Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An Open Source Platform for Ligand Pocket Detection. *BMC Bioinf.* **2009**, *10*, 168.

(16) Tsujikawa, H.; Sato, K.; Wei, C.; Saad, G.; Sumikoshi, K.; Nakamura, S.; Terada, T.; Shimizu, K. Development of a Protein−Ligand-Binding Site Prediction Method Based on Interaction Energy and Sequence Conservation. *J. Struct. Funct. Genomics* **2016**, *17*, 39−49.

(17) Krivák, R.; Hoksza, D. P2Rank: Machine Learning Based Tool for Rapid and Accurate Prediction of Ligand Binding Sites from Protein Structure. *J. Cheminf.* **2018**, *10*, 39.

(18) Yang, J.; Roy, A.; Zhang, Y. Protein−Ligand Binding Site Recognition Using Complementary Binding-Specific Substructure Comparison and Sequence Profile Alignment. *Bioinformatics* **2013**, *29*, 2588−2595.

(19) Jiménez, J.; Doerr, S.; Martínez-Rosell, G.; Rose, A. S.; De Fabritiis, G. DeepSite: Protein-Binding Site Predictor Using 3D-Convolutional Neural Networks. *Bioinformatics* **2017**, *33*, 3036−3042.

(20) Mylonas, S. K.; Axenopoulos, A.; Daras, P. DeepSurf: A Surface-Based Deep Learning Approach for the Prediction of Ligand Binding Sites on Proteins. *Bioinformatics* **2021**, *37*, 1681−1690.

(21) Kandel, J.; Tayara, H.; Chong, K. T. PUResNet: Prediction of Protein-Ligand Binding Sites Using Deep Residual Neural Network. *J. Cheminf.* **2021**, *13*, 65.

(22) Stärk, H.; Ganea, O.-E.; Pattanaik, L.; Barzilay, R.; Jaakkola, T. EquiBind: Geometric Deep Learning for Drug Binding Structure Prediction. **2022**, arXiv:2202.05146.

(23) Lu, W.; Wu, Q.; Zhang, J.; Rao, J.; Li, C.; Zheng, S. TANKBind : Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction. **2022**, bioRxiv:495043.

(24) Corso, G.; Stärk, H.; Jing, B.; Barzilay, R.; Jaakkola, T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. **2022**, arXiv:2210.01776.

(25) Koes, D. R.; Baumgartner, M. P.; Camacho, C. J. Lessons Learned in Empirical Scoring with Smina from the CSAR 2011 Benchmarking Exercise. *J. Chem. Inf. Model.* **2013**, *53*, 1893−1904.

(26) McNutt, A. T.; Francoeur, P.; Aggarwal, R.; Masuda, T.; Meli, R.; Ragoza, M.; Sunseri, J.; Koes, D. R. GNINA 1.0: Molecular Docking with Deep Learning. *J. Cheminf.* **2021**, *13*, 43.

(27) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. Three-Dimensional Convolutional Neural Networks and a Crossdocked Data Set for Structure-Based Drug Design. *J. Chem. Inf. Model.* **2020**, *60*, 4200−4215.

(28) Ballester, P. J.; Mitchell, J. B. O. A Machine Learning Approach to Predicting Protein-Ligand Binding Affinity with Applications to Molecular Docking. *Bioinformatics* **2010**, *26*, 1169−1175.

(29) Jiang, D.; Hsieh, C. Y.; Wu, Z.; Kang, Y.; Wang, J.; Wang, E.; Liao, B.; Shen, C.; Xu, L.; Wu, J.; Cao, D.; Hou, T. InteractionGraphNet: A Novel and Efficient Deep Graph Representation Learning Framework for Accurate Protein-Ligand Interaction Predictions. *J. Med. Chem.* **2021**, *64*, 18209−18232.

(30) Yu, Y.; Lu, S.; Gao, Z.; Zheng, H.; Ke, G. Do Deep Learning Models Really Outperform Traditional Approaches in Molecular Docking? **2023**, arXiv:2302.07134.

(31) Desaphy, J.; Bret, G.; Rognan, D.; Kellenberger, E. sc-PDB: a 3D-database of ligandable binding sites—10 years on. *Nucleic Acids Res.* **2015**, *43*, D399−D404.

(32) Desaphy, J.; Azdimousa, K.; Kellenberger, E.; Rognan, D. Comparison and Druggability Prediction of Protein-Ligand Binding Sites from Pharmacophore-Annotated Cavity Shapes. *J. Chem. Inf. Model.* **2012**, *52*, 2287−2299.

(33) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminf.* **2011**, *3*, 33.

(34) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. E. Array Programming with NumPy. *Nature* **2020**, *585*, 357−362.

(35) Morris, G. M.; Goodsell, D. S.; Halliday, R. S.; Huey, R.; Hart, W. E.; Belew, R. K.; Olson, A. J. Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function. *J. Comput. Chem.* **1998**, *19*, 1639−1662.

(36) Nocedal, J.; Wright, S. J. *Numerical Optimization*; Springer Series in Operations Research; Springer Verlag: Berlin, 1999.

(37) Fletcher, R. On the BarzilaiBorwein Method. *Optimization and Control with Applications*; Springer, 2005.

(38) Tang, S.; Chen, R.; Lin, M.; Lin, Q.; Zhu, Y.; Ding, J.; Hu, H.; Ling, M.; Wu, J. Accelerating AutoDock Vina with GPUs. *Molecules* **2022**, *27*, 3041.

(39) Yu, Y.; Cai, C.; Zhu, Z.; Zheng, H. Uni-Dock: A GPU-Accelerated Docking Program Enables Ultra-Large Virtual Screening. **2022**, ChemRxiv:2022-5t5ts.

(40) Wang, R.; Fang, X.; Lu, Y.; Wang, S. The PDBbind Database: Collection of Binding Affinities for Protein-Ligand Complexes with Known Three-Dimensional Structures. *J. Med. Chem.* **2004**, *47*, 2977−2980.

(41) Landrum, G. *Rdkit: Open-Source Cheminformatics Software*, 2016.

(42) Word, J.; Lovell, S.; Richardson, J.; Richardson, D. Asparagine and Glutamine: Using Hydrogen Atom Contacts in the Choice of Sidechain Amide Orientation. *J. Mol. Biol* **1995**, *285*, 1735−1747.

(43) Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Meng, E. C.; Couch, G. S.; Croll, T. I.; Morris, J. H.; Ferrin, T. E. UCSF ChimeraX: Structure Visualization for Researchers, Educators, and Developers. *Protein Sci.* **2021**, *30*, 70−82.

https://doi.org/10.1021/acs.jcim.3c00519