JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

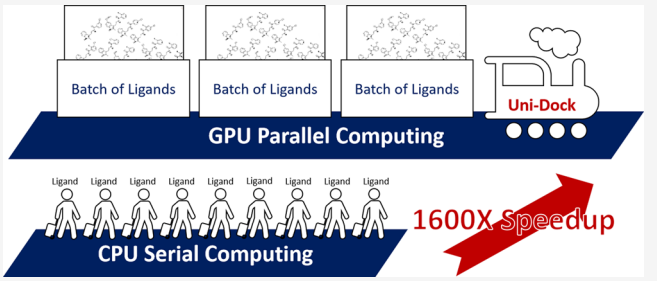
Uni-Dock: GPU-Accelerated Docking Enables Ultralarge Virtual Screening

Yuejiang Yu, Chun Cai, Jiayue Wang, Zonghua Bo, Zhengdan Zhu,* and Hang Zheng*

Cite This: J. Chem. Theory Comput. 2023, 19, 3336−3345 Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: Molecular docking, a structure-based virtual screening method, is a reliable tool to enrich potential bioactive molecules from molecular databases. With the rapid expansion of compound library sizes, the speed of existing molecular docking programs becomes less than adequate to meet the demand for screening ultralarge libraries containing tens of millions or billions of molecules. Here, we propose Uni-Dock, a GPU-accelerated molecular docking program that supports various scoring functions including vina, vinardo, and ad4. Uni-Dock achieves more than 1000-fold speedup with high accuracy compared with the AutoDock Vina running in single CPU core, outperforming reported GPU-accelerated docking programs including AutoDock-GPU and Vina-GPU based on head-to-head experiments. Uni-Dock docks molecules in batches simultaneously using concurrent threads of each molecule. The data flow between GPU and CPU is optimized to eliminate CPU hotspots and maximize GPU utility. Additionally, Uni-Dock also supports hydrogen bond biased docking for all scoring functions and can be migrated to multiple GPUs of different architectures and manufacturers. We analyzed the improved performance of Uni-Dock on the CASF-2016 and DUD-E datasets and recommend three combinations of hyperparameters corresponding to different docking scenarios. To demonstrate Uni-Dock's capability on routinely screening ultralarge libraries, we performed hierarchical virtual screening experiments with Uni-Dock on the Enamine Diverse REAL druglike set containing 38.2 million molecules to a popular target KRAS G12D in 12 h using 100 NVIDIA V100 GPUs. To the best of our knowledge, Uni-Dock should be the fastest GPU-accelerated docking program to date.



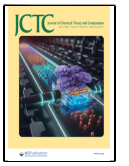
INTRODUCTION

Virtual screening (VS), which aims to identify hit compounds as the starting point for drug discovery, plays a crucial role in modern drug design, due to its high efficiency, low cost, and freedom from chemical entity limitations, compared to high-throughput screening (HTS).[1,2] Among various VS approaches, molecular docking has become one of the most popular tools.[3] Molecular docking searches for the binding pose of a ligand within a receptor and calculates the binding affinity of the resulting receptor−ligand complex. Various benchmark studies have proved that molecular docking has a good ability to predict the binding pose and to enrich bioactive molecules.[4−7]

Previous works indicate that the screening of larger molecular libraries could lead to candidate compounds with higher potency.[8] Considering that the "low-hanging fruits" are rare in modern drug discovery, the pursuit of chemical diversity to empower drug discovery of targets with high value, as well as difficulty, is expected. Possibly in response to this trend, a rapid increase in the size of commercial libraries was observed in recent years. A representative example is Enamine REAL Space,[9] which consists of 22.7 billion make-on-demand molecules and challenges the virtual screening efficiency.[10] On the other hand, urgent public health events such as the COVID-19 pandemic and monkeypox virus infection place high demands on speedy screening of databases against fast emerging receptors.[11] Consequently, widely used molecular docking methods, like AutoDock4,[12] AutoDock Vina,[13] and Glide,[14,15] may not meet the up-to-date in-field requirements for fast and robust screening capability, due to the bottleneck of computational resources.[16]

To speed up screening computation, efforts have been made to improve the performance of the searching algorithm, thus facilitating running acceleration primarily based on CPU. AutoDock Vina 1.2[17] is compatible with the ad4 scoring function so that the speed of AutoDock4 can take advantage of the acceleration of MC/BFGS[18] searching method in AutoDock Vina. QuickVina2[19] and QuickVina-W[20] reduce redundancy in searching by allowing searching threads of AutoDock Vina to communicate with a record of the explored

Received: November 15, 2022
Published: April 26, 2023



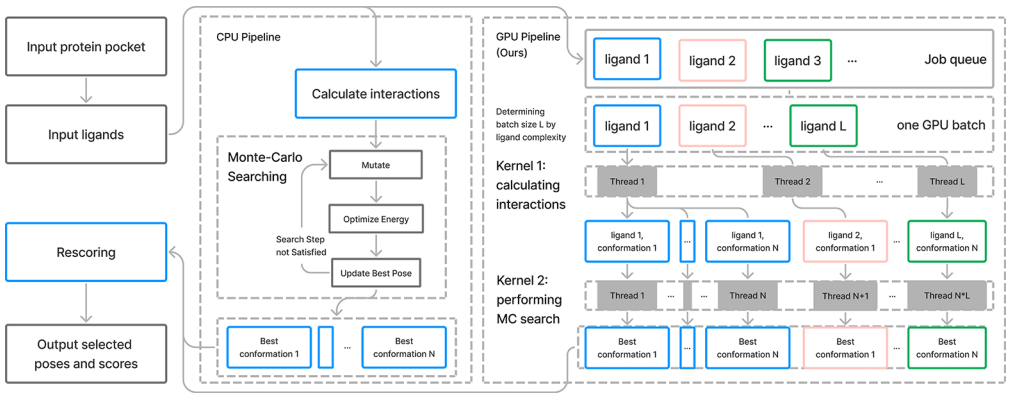


Figure 1. Diagram of algorithms for Uni-Dock in comparison with AutoDock Vina 1.2. In general, Uni-Dock functions in a thread-level parallelization form to largely benefit from GPU computational power. Ligands input are processed in parallel. MC threads for each ligand are executed simultaneously on GPU to cover the whole computation space rapidly. Specific operations inside each MC thread are identical to Vina. MC threads in Vina are put in a queue called a task container. Each MC search is iterated within a given step range. In each iteration step, random mutation of rotatable bonds, evaluating energy, force, and gradient descent is performed sequentially. The best pose with corresponding evaluation score within searching scope is proposed as output for each thread.

searching space. However, the resulting improvement in docking speed is generally within 1 order of magnitude, which seems to be inadequate when screening ultralarge libraries.

In addition, efforts have also been made to benefit from hardware acceleration by high-performance cluster computing based on CPU. For example, VirtualFlow[16] takes advantage of the linear scalability of docking and calculates 1 billion compounds in approximately 2 weeks when leveraging 10 000 CPU cores simultaneously. Despite the reduced wall time for all compounds, it does not change the average CPU running time for each molecule, thus the cost of computational resources remains the bottleneck.

With the development of artificial intelligence (AI), some efforts such as DeepDocking,[10] active learning Glide,[21] and GNINA[22] attempt to train deep learning (DL) models on a subset of the large molecular database and then predict docking scores for the rest. However, since deep learning models rely on a large amount of labeled data to improve generalization, training a reliable model requires large-scale molecular docking first,[10] which might still be a limitation for its wide application. Machine learning (ML) models focusing on the accuracy of either binding poses or affinities[23−25] also need a large amount of high-quality data generated by docking for training.

Hence, the urgent need for computationally feasible virtual screening on ultralarge molecular databases calls for the improvement of the docking efficiency of each molecule. The application of GPU is a promising strategy. As a widely used acceleration hardware, GPU has shown its powerful capability of high-performance computing, especially for highly parallelized docking workloads because the scalable array of multithreaded streaming multiprocessors (SM) of GPU makes docking calculations in parallel naturally. Therefore, it is unsurprising that attempts of migrating molecular docking software to GPU have been made, including AutoDock-GPU[26−28] and Vina-GPU.[29] However, the speed of the original AutoDock-GPU is only comparable to AutoDock Vina since it is developed on AutoDock4, which uses a low-efficiency global searching algorithm. After migrating and tuning on the Summit supercomputer, the efficiency of pipelined AutoDock GPU is boosted by a factor of ∼4. However, there is still a lot more room for improvement due to the advent of the Monte Carlo (MC) search of Vina. Vina-GPU makes use of the efficient MC searching method proposed in AutoDock Vina, but the computational power of GPU has not been discovered sufficiently. When tested on the AutoDock-GPU dataset, Vina-GPU lead to an average speedup of ∼20 times more than AutoDock Vina.[26] Additionally, the performance of Vina-GPU was conducted on 9125 ligands from DrugBank,[30] which is far less than common scenarios of high-throughput virtual screening. Thus, the enrichment factor on the widely accepted benchmark datasets remains to be determined to further validate the screening power of Vina-GPU. Moreover, new docking features, such as vinardo and ad4 scoring functions, batch mode, and grid map support, developed in version 1.2 of AutoDock Vina, have not yet been included in Vina-GPU.

In this work, we propose Uni-Dock, a GPU-accelerated docking method built upon AutoDock Vina 1.2, which reduces the average time cost of docking for each molecule to a minimum of 0.1 s, thus enabling ultralarge virtual screening with varied scoring functions including vina, vinardo, and ad4. In the following parts, we demonstrate how we change the searching pattern of Monte Carlo (MC) algorithm to exploit the marvelous parallelism performances and memory bandwidth of GPU with various methods. Moreover, bias docking is supported with all mentioned scoring functions and Uni-Dock is available both with NVIDIA and AMD GPUs. The embarrassingly parallel (a term used in parallel computing, also called delightfully parallel) performance of Uni-Dock and the comparison with previous GPU-accelerated docking programs are also shown.

Validations on the widely accepted benchmark databases DUD-E[7] and CASF-2016[5] are performed to demonstrate the screening power and docking capability of the Uni-Dock with various parameter configurations and scoring functions. Moreover, to demonstrate the powerful screening ability of Uni-Dock on large compound libraries, we performed a hierarchical virtual screening approach[2] on the Enamine Diverse REAL druglike set[9] and docked all 38.2 million ligands therein to a popular target KRAS G12D (PDB ID: 7RPZ)[31] using a GPU cluster containing 100 NVIDIA V100 GPUs. The whole process finished within 12 h.

https://doi.org/10.1021/acs.jctc.2c01145
J. Chem. Theory Comput. 2023, 19, 3336−3345

METHODOLOGY

Architecture of Uni-Dock. Our improvements consist of three stages, identified as Stages I, II, and III. In Stage I, we dock one ligand with hundreds of threads from GPUs. In Stage II, docking is performed on multiple ligands in parallel with multiple threads running simultaneously. In Stage III, we further reduce GPU memory footprint and boost computational speed by using single-precision floating point numbers in GPU calculation. With reduced memory usage of GPU and multithreaded CPU IO, ligand batch size is determined on the fly. Uni-Dock is developed based on the framework of AutoDock Vina 1.2.[17] The architecture of Uni-Dock and its comparison with Vina is depicted in Figure 1.

The details regarding the three critical stages are as follows.

Stage I: Single Ligand, Multiple Threads. For the first stage of migrating CPU code to GPU, we implement the calculation of MC search in CUDA without any modification in the searching strategy, since each MC thread calculates the derivatives of atoms in the force field needed in BFGS. Uncoalesced random memory access of the force map between atoms undoubtedly becomes the most critical memory bottleneck. The scheduling of searching tasks is implemented with a CPU task queue. MC jobs with different initial conformations are put into the task queue and asynchronously distributed to CPU threads. Identical to the CPU thread, each GPU thread handles the searching of one initial conformation. It is more efficient than the SIMT (single instruction, multiple threads) architecture of GPU and processes the conformations of one ligand simultaneously. As the GPU computing unit is not able to directly access the main memory, for each GPU batch of ligands, we manage the data flow in the following steps: calculate the interaction maps of the pocket, allocate GPU memory for data and result, transfer the prepossessed data to GPU, call a function that requires GPU data, and gather the result back to CPU memory. To adapt GPU usage, we migrate the data structure from array-of-structure to structure-of-array for transferring memory between CPU and GPU in stride.

After the migration, the main workload MC searching is processed on GPU with the same number of threads (*nthreads*, namely *exhaustiveness* in Vina) of the CPU pipeline. A typical value of exhaustiveness is 8. However, the number of GPU threads is on the order of 10 000, which means the degree of parallelism is not properly utilized yet. Within our expectation, the running time temporarily becomes longer compared with AutoDock Vina running on CPU. The main cause is the lower single-thread capacity of GPU and the additional overhead of transferring data.

For conformations, the search space is determined by search width (the number of the searching threads with different initial poses) and search depth (the number of samples in MC searches). Since the searching space is high-dimensional, enough threads with randomly initialized poses can exploit possible outcomes and find poses with good scores. This strategy is used and validated in Vina-GPU[29] with consistent accuracy. Therefore, we change our searching strategy from depth-first search to breadth-first search, utilizing the parallel capacity of GPU. By setting the width of over 1000 threads, we limit the search depth to 10−40 steps without missing favorable poses in the search space. With these threads running simultaneously on GPU, the searching speed is drastically improved. We note that the combination of the number of threads per ligand (*nthreads*) and search depth (namely, *nsteps*, *max_evals* in Vina) can significantly affect speed and accuracy. Vina also provides user-defined parameters to set *exhaustiveness* and *max_evals* to balance the speed and accuracy of the docking process. The search depth of Vina is a heuristic value based on ligand properties while *exhaustiveness* is given by users. Likewise, we recommend three setting patterns of accuracy and speed after comprehensively tests on various combinations of these two parameters as *fast*, *balanced*, and *detailed*. We will discuss these settings in detail with more experimental results in the Results section.

Stage II: Multiple Ligands, Multiple Threads. In this stage, we perform searching on multiple ligands in one GPU batch with one kernel launch. Since the threads used by one ligand are still far below the capacity of GPU, the processed data of ligands are transformed in array form to load in GPU memory thus aligned with calculation threads. Most ligands have 20−40 atoms, while some might have more than 100 atoms, and the unit capacity of a fixed-length array is square to the maximum atom number of ligands in a dataset. To maximize memory usage, we allocate memory by the required space of interacting forces between atom pairs of each ligand.

In addition to computational improvement, Uni-Dock takes fault tolerance into consideration as well. Since there are various reasons to raise errors in large-scale virtual screening and one failed compound derived from these errors should not interfere with the calculation of other ligands within the same batch. Therefore, we add error detection and recovery mechanisms in ligand parsing and searching procedures. To ensure enough GPU memory, a memory viability check is performed before the actual MC searches.

The computation speed and user experience should benefit from the above modifications, since we calculate the most time-consuming part in parallel with a low error rate. Considering that GPU has a higher memory bandwidth compared with CPU, the running should be the more ligands we put in one batch, the higher the efficiency.

The efficiency of the whole system allows for further improvement. Input ligands require memory to store their property traits and interaction maps. Since the memory size of GPU limits the maximum number of ligands in one batch, we fix the batch size to ∼200 to avoid out-of-memory failure. In addition, I/O expense and preprocessing calculations on CPU become new hotspots that require more than 60% of the overall running time.

Stage III: Exploit GPU. We further exploit GPU memory and calculating ability in this stage. First, we move computationally intensive preprocessing to GPU by calculating interaction forces between atoms of each ligand simultaneously on GPU with higher throughput. This improvement also eliminates the large transfer of preprocessed data from CPU to GPU.

Besides, we replace double-precision floating point numbers with single-precision ones. As a result, single-precision data helps to save memory and speeds up arithmetic operations. The Docking score is a rough prediction to evaluate binding affinity,[4] compared to other methods,[32,33] and the score is coarse-grained (for the *n*-outliers, [−15.00, 0] with an accuracy of 0.1). In single precision, we save more than 30% of kernel running time while preserving almost identical energy scores and we can increase the batch size by up to two-fold. The difference between single and double precision does not affect accuracy, as shown in the following results.

https://doi.org/10.1021/acs.jctc.2c01145
J. Chem. Theory Comput. 2023, 19, 3336−3345

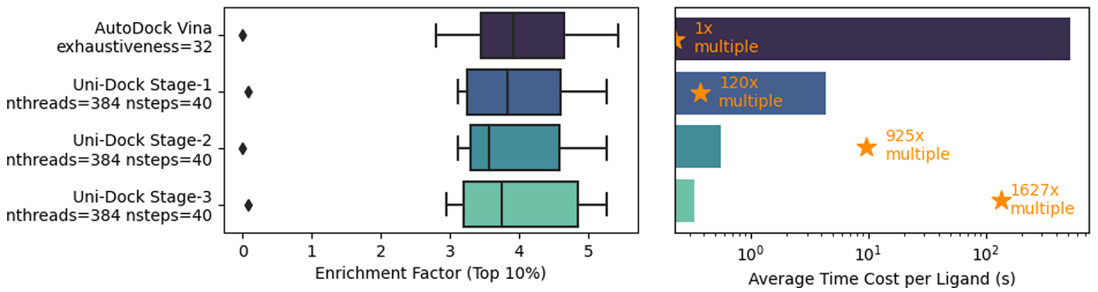


Figure 2. Performance improvements of three critical stages: 1627-fold speedup with consistent accuracy.

To fully use the GPU memory to perform as many docking processes simultaneously as possible, we develop a linear regression model for predicting peak memory usage based on ligand properties and docking settings. Our regression function of vina/vinardo scoring function based on NVIDIA V100 32G GPU is

$$memory = 0.011978 \times \sum_{i=1}^{batch\ size} atom\ number_i^2 + 1.214869 \times batch\_size + 0.003852 \times thread + 20017 \quad (1)$$

Here, *memory* (in MB) is the predicted peak global memory usage on GPU, $atom\_number_i$ is the number of atoms in the ith ligand in the current batch, *batch_size* is the number of ligands put in one batch to be searched in parallel, *thread* is the overall number of MC threads in this batch, which is equal to *nthreads* × *batch_size*, and the constant stands for temporary memory consumption during MC searches.

Due to the complex energy terms of ad4, the memory requirement of the ad4 scoring function is slightly different, which is

$$memory = 0.079216 \times \sum_{i=1}^{batch\ size} atom\ number_i^2 + 1.911645 \times batch\_size + 0.003910 \times thread + 20052 \quad (2)$$

The correlation coefficient (R^2) of our regression reaches 0.999 on real data; thus, we can put ligands into one batch until the predicted memory hits hardware limitation. With this method, we successfully call more than 95% of the GPU memory with an appropriate batch size setting. What's more, Uni-Dock can automatically choose the correct memory function according to GPU architecture and parameters.

Eight targets from the DUD-E dataset are applied to evaluate the screening power and speed of our implementation with AutoDock Vina operated on CPU with *exhaustiveness* = 32 as the baseline. In each stage, we analyze the drawbacks of the current implementation through detailed profiling results and use a hardware-adapted method to accelerate without a loss of accuracy.

Overall, Uni-Dock achieves 1627× acceleration, compared with a single CPU core with the same accuracy. The speed-up ratio and screening performance indicator, enrichment factor (EF) of top 10%, is illustrated in Figure 2. The acceleration ratio is calculated by the execution time of Vina on one 2.5 GHz Intel Xeon Platinum 8269CY (Cascade Lake) CPU and Uni-Dock on one NVIDIA V100 32G GPU.

Uni-Dock on Different GPU Types. We successfully migrated Uni-Dock to multiple GPUs of different architectures and manufacturers. First, we are able to run Uni-Dock with NVIDIA T4, NVIDIA 2080, NVIDIA 3090, and NVIDIA A100 easily, thanks to a uniform CUDA driver, with minor updates of memory fitting based on available global memory and the architecture of GPU. Then, we successfully execute Uni-Dock on ROCm-compatible GPUs such as AMD MI210. The CPU codes are almost identical, while CUDA APIs in GPU codes are translated to ROCm APIs using hipify—perl.

Biased Docking for All Scoring Functions. Biased docking, biasing the results toward the formation of relevant protein−ligand interactions, is a promising method to improve the accuracy of predicting the binding poses for docking programs.[34] However, biased docking previously was only available for the ad4 scoring function.

Herein, we implement bias for hydrogen-bond receptor/acceptor or any given kind of atoms for all three scoring functions, by modifying the grid map before searching and computing energy modifications directly in rescoring poses after searching. Specifically, a standard bias parameter file (BPF) is first loaded, which points out the 3D coordinates of the center of bias, its energy and range, and the type of bias. After computing or reading the grid map, Uni-Dock modifies the grid map based on bias information and automatically chooses corresponding atom types according to the assigned bias. The grid map enables MC search using the difference between grids as gradients. The impact of bias from refining and rescoring is calculated after searching. Moreover, the time consumed by adding bias can be neglected. Only a scan of every grid point is performed in less than 0.1 s for processing the bias on one receptor.

Our results show that Uni-Dock bias can significantly improve docking accuracy when hydrogen-bond constraints are given or atom position is restricted. The effect of energy and range parameters is non-negligible, which should be assigned according to the kind of bias and receptor features.

RESULTS

Screening Power and Docking Power of Uni-Dock. To demonstrate the performance of Uni-Dock, we tested the docking power (i.e., the ability to predict the native ligand binding pose) and the screening power (i.e., the ability to find bioactive ligands among a pool of bioactive ligands and decoys) with the vina scoring function on the CASF-2016 dataset and the DUD-E dataset, respectively. Docking power is evaluated by calculating the proportion of the ligands of which RMSD between the binding pose given by Uni-Dock and crystal structure is less than the given threshold. Screening power is evaluated by the enrichment factor (1% and 20%) calculated according to the following formula:[15]

https://doi.org/10.1021/acs.jctc.2c01145
J. Chem. Theory Comput. 2023, 19, 3336−3345

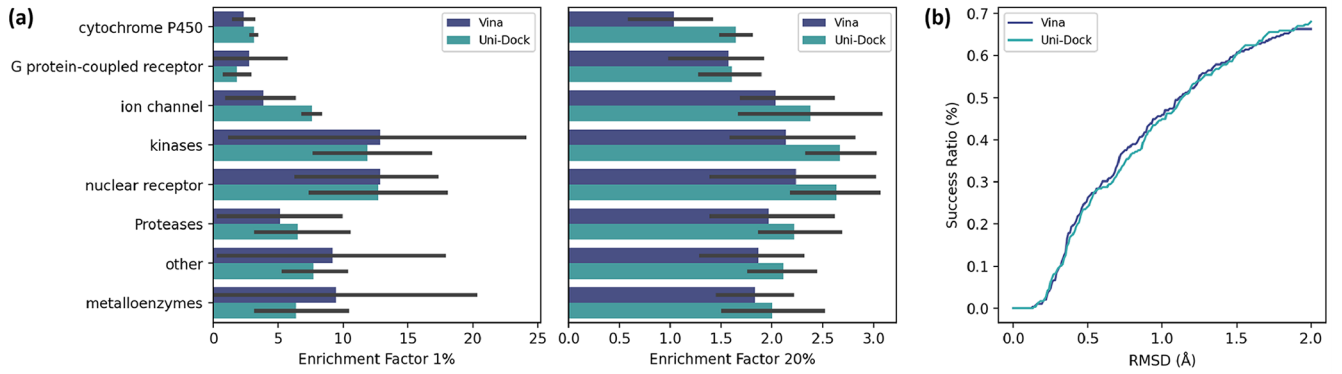


Figure 3. Screening power and docking power of Uni-Dock and AutoDock Vina with vina scoring function. (a) Screening power evaluated by an enrichment factor (EF) of (left) 1% and (right) 20%) on the DUD-E dataset. (b) Docking power evaluated by the root-mean-square deviation (RMSD) between the binding pose in crystal structures and predicted by docking software on the CASF-2016 dataset.

$$\mathrm{EF} = \frac{\mathrm{Hits}_{\mathrm{sampled}}/N_{\mathrm{sampled}}}{\mathrm{Hits}_{\mathrm{total}}/N_{\mathrm{total}}} \quad (3)$$

where $N_{\mathrm{sampled}}/N_{\mathrm{total}}$ = 1% or 20%.

For 285 targets in the CASF-2016 dataset, the cocrystallized ligand from every complexes is docked. For 102 targets in the DUD-E dataset, every ligands from both active and decoy sets are docked. The receptor structures and ligands are prepared on the web service Hermite (https://hermite.dp.tech). The ligands are set flexible during docking, and the receptor is rigid.

The screening power and the docking power between Uni-Dock and AutoDock Vina are shown in Figure 3. The reference data for the screening power is based on previous work.[35] The reference data for docking power is obtained by using AutoDock Vina with an exhaustiveness equal to 64. The maximum iteration number of MC search in Uni-Dock equals 40, and the number of MC threads is set as 512, corresponding to the *detailed* mode in our recommendation setting pattern. We use the default value for the other settings, including the number of ligand poses generated, which is equal to 9.

The screening power and the docking power between Uni-Dock and AutoDock Vina are comparable. We observed that on the index of an enrichment factor of 20%, Uni-Dock performs generally better than AutoDock Vina, which may be caused by the larger search space of Uni-Dock, which has been more fully searched for binding poses.

Performance of Uni-Dock with Typical Parameter Combinations. Preference for accuracy and speed when performing molecular docking would vary in different cases. For example, efficiency should be the top priority when screening large-scale databases, while high precision is more desired in binding mode determination.

To meet the requirements of different scenarios, we recommend three combinations of hyperparameters that control calculation speed and precision, namely *fast* mode (*nthreads* = 128 and *nsteps* = 20), *balanced* mode (*nthreads* = 384 and *nsteps* = 40), and *detailed* mode (*nthreads* = 512 and *nsteps* = 40).

As shown in Figure 4, Uni-Dock *detailed* mode can achieve the best performance, with regard to both docking power and screening power. In the docking power task, the RMSD of binding pose predicted by Uni-Dock *detailed* mode from the crystal structure is within 2 Å in 68.07% (194/285) of targets in the CASF-2016 dataset, and within 1 Å in 46.67% (133/285) of the same targets in the CASF-2016 dataset. In the

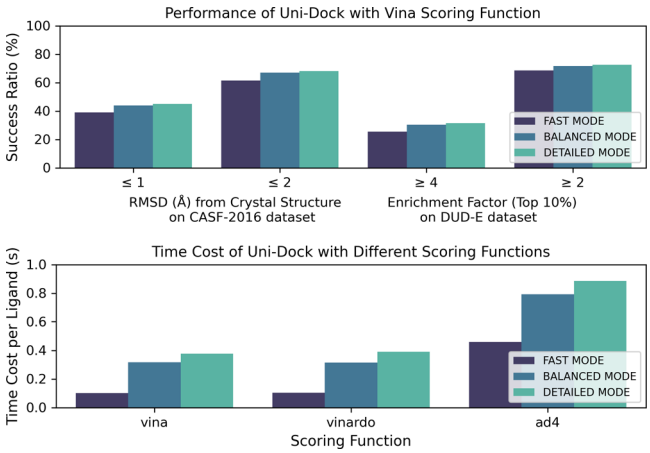


Figure 4. Performance of Uni-Dock with typical parameter combinations: the upper graph shows the success ratio on docking power and screening power of different Uni-Dock modes; the lower graph shows the time cost for docking each ligand with different scoring functions of different Uni-Dock modes. All experiments operated on one node using 16 2.5 GHz Intel Xeon Platinum 8269CY (Cascade Lake) CPU cores and one NVIDIA V100 32G GPU.

screening power task, the enrichment factor carried out by Uni-Dock *detailed* mode is larger than 2 in 72.55% (74/102) of the targets in the DUD-E dataset, and larger than 4 in 31.37% (32/102) of the targets in the DUD-E dataset. The performance of Uni-Dock *balanced* mode is slightly worse than Uni-Dock *detailed* mode, and the *fast* mode is slightly worse than Uni-Dock *balanced* mode. On the other hand, the success ratio of $\mathrm{EF}_{\mathrm{top\ 10\%}} \geq 2$ for Uni-Dock *fast* mode reaches 68.63%, only 3.92% smaller than the score of Uni-Dock *detailed* mode, which indicates Uni-Dock *fast* mode can perform as well as Uni-Dock *detailed* mode on many systems upon which Uni-Dock *detailed* mode has good performance.

Parameter combination strategies differ in average calculation speed for each docked molecule. As shown in Figure 4, with vina scoring function, the average time cost for docking each ligand is ∼0.10 s under Uni-Dock *fast* mode, ∼0.32 s for Uni-Dock *balanced* mode, and ∼0.38 s for Uni-Dock *detailed* mode. Different scoring functions also impact the running speed. Since vina and vinardo scoring functions share similar energy terms, the performance of calculation speed is very similar. In contrast, the ad4 scoring function has more complicated energy terms, so the average time cost for each

https://doi.org/10.1021/acs.jctc.2c01145
J. Chem. Theory Comput. 2023, 19, 3336−3345

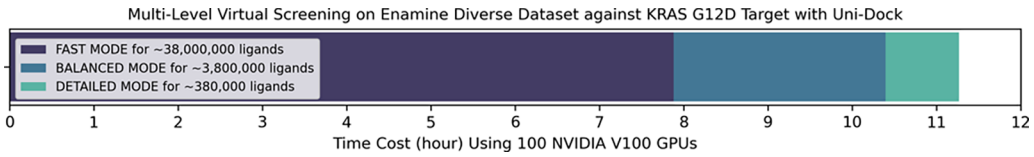


Figure 5. Time cost of hierarchical virtual screening approach on an Enamine Diverse REAL druglike set against KRAS G12D with Uni-Dock performed on 100 NVIDIA V100 GPUs.

ligand is slightly larger, ~0.46 s for Uni-Dock *fast* mode, ~0.79 s for Uni-Dock *balanced* mode, and 0.89 s for Uni-Dock *detailed* mode. While using AutoDock Vina 1.2 with exhaustiveness equal to 8, the average time cost for each ligand is 124.56 s using the vina scoring function, 149.99 s using the vinardo scoring function, and 335.89 s for the ad4 scoring function.

Above all, for high-throughput virtual screening tasks, we recommend Uni-Dock *fast* mode, since it showed a speed of over 2-fold faster compared with *detailed* mode but resemble the corresponding accuracy performance. For the binding pose prediction task, we recommend Uni-Dock *detailed* mode, since the Uni-Dock *detailed* performs best. The Uni-Dock *balanced* mode has moderate performance in both precision and speed, which is suitable for virtual filtering on small datasets.

Virtual Screening on an Ultralarge Dataset Using Uni-Dock. To demonstrate the advantages of Uni-Dock in high-throughput screening, we conducted a high-throughput virtual screening of the recently popular drug target, KRAS G12D (PDB ID: 7RPZ), on the Enamine Diverse REAL druglike set[9] containing 38.2 million ligands.

To take into account both speed and precision, we adopt a hierarchical virtual screening approach. We first applied Uni-Dock *fast* mode to dock the 38.2 million ligands of the Enamine Diverse REAL druglike set. Then, out of the top 10% of the ligands (3.82 million ligands) obtained in the previous step, we apply the Uni-Dock *balanced* mode for docking. Finally, out of the molecules with a score of 10% (~382 000 ligands) obtained in the previous step, we apply Uni-Dock *detailed* mode to obtain the final results.

We use a GPU cluster with 100 NVIDIA V100 GPUs for the hierarchical virtual screening approach described above. As shown in Figure 5, the docking step using Uni-Dock *fast* mode on 38 million molecules took 7.88 h, accounting for 69.92% of the whole running time, the docking step using Uni-Dock *balanced* mode on 3.8 million molecules took 2.52 h, accounting for 22.36% of running time, and the step using Uni-Dock *detailed* mode on 0.38 million molecules took 0.87 h, accounting for 7.72%.

Above all, a total of 11.27 h was required to complete the hierarchical virtual screening approach on 38 million molecules with Uni-Dock using a cluster with 100 NVIDIA V100 GPUs, which means that rapid high-throughput virtual screening of extremely large molecular libraries is feasible and affordable.

Influence of Hyperparameters and Ligand Properties on Speed. Based on the experiments above, a regression equation of running time with respect to the settings of hyperparameters and properties of input ligands is obtained. The equation is first proposed on the basis of our algorithm's time complexity analysis and then refined with a huge amount of real data. We finally achieve an accurate prediction of running time (R^2 = 0.946).

We first realized that the running time of MC search on GPU roughly goes linear with the product of *nthreads* and *nsteps* if other settings remain the same. Moreover, complex and large ligands take more time to mutate, move, and calculate energy in MC searches. We also need storage on GPU in $O(n^2)$ to save precalculated interactions between each atom, in which n is the number of atoms of ligand. In each iteration of MC search, operation complexity roughly goes linear with the square of the number of atoms of the ligand, since we calculate the interaction between each pair of atoms in the ligand. Furthermore, there are operations outside MC search such as initializing GPU kernels, collecting results, and transferring memory between CPU and GPU. Last but not least, file interactions and constant time in the workflow take an approximately fixed time.

Based on the analysis above, we collect running time data from screening experiments using the vina scoring function. The raw data are aggregated for approximately every 10 000 ligand docking samples, since we put them in one Uni-Dock job. We then perform OLS regression and successfully fit running time with a correlation coefficient of R^2 = 0.946:

$$\begin{aligned} time = {} & 7.23 \times 10^{-8} \times nthreads \times natoms^2 \\ & + 1.86 \times 10^{-8} \times nthreads \times nsteps \times natoms^2 \\ & - 0.007984 \end{aligned} \tag{4}$$

where *time* is the average end-to-end running time of docking one ligand in a batch, *nthreads* the number of threads, *natoms* the number of atoms of ligand, and *nsteps* the number of steps. Note that we have to put enough ligands in one batch to fully utilize the power of GPU and we recommend putting no less than 10 000 ligands in the command line and let Uni-Dock divide them into appropriate batches. Since our dataset mainly consists of ligands with less than 50 atoms, our function might have to be modified in the case of much heavier ligands.

Scaling Performance of Uni-Dock on GPU Cluster. The scaling performance of Uni-Dock is measured on a GPU cluster with eight NVIDIA V100 32G GPUs and 96 CPUs. We choose a test target from the DUD-E dataset with 36 341 ligands for the test. We use one, two, four, and eight GPUs, respectively, to dock all ligands to this target and report the overall running time. As shown in Figure 6, Uni-Dock achieves the embarrassingly parallel performance, because of the fact that Uni-Dock does not require any communication between GPUs and the docking tasks of different ligands are completely independent. Meanwhile, although the scaling performance of Uni-Dock is guaranteed theoretically, it is also noteworthy that other factors such as disk IO might become new bottlenecks if too many GPUs are used for parallel computing at the same time. Nevertheless, according to our experience from the previous test, the scaling performance of Uni-Dock should generally be applicable for at least 100 V100 GPUs and little impact from factors like disk IO was observed.

Runtime Performance of Uni-Dock on Different GPUs. The runtime performance of Uni-Dock on different GPUs is measured on eight ligand−receptor systems from the

https://doi.org/10.1021/acs.jctc.2c01145
J. Chem. Theory Comput. 2023, 19, 3336−3345

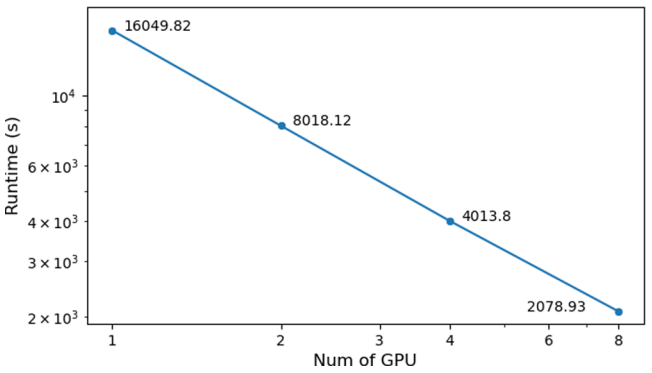


Figure 6. Scaling performance of Uni-Dock on eight GPUs.

DUD-E data set with a total of 29 664 ligands. We run Uni-Dock under all three search modes with other parameters unchanged. Uni-Dock maintains the speed on various GPUs with the same accuracy. The runtime of Uni-Dock for each ligand on various GPUs is shown in Figure 7. In general, the

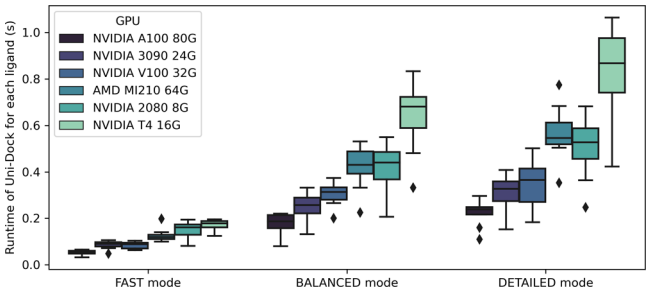


Figure 7. Runtime performance of Uni-Dock on different GPUs in three modes.

higher-end GPUs (such as NVIDIA A100 and NVIDIA 3090) achieve higher speeds, and the lower-end GPUs (such as NVIDIA T4 and NVIDIA 2080) achieve lower speeds. We find that the speed of Uni-Dock on different GPUs is simultaneously affected by computing power and memory size, and is not linearly related to any of them. Uni-Dock works well on AMD architecture GPUs such as MI210, and its runtime performance is close to that of NVIDIA 2080.

Performance of Biased Docking Using Uni-Dock. A case is shown in Figure 8 to demonstrate how the application of biased docking implemented in Uni-Dock helps preserve the native protein−ligand interactions. As shown in Figure 8 a, the cocrystallized ligand of the tankyrase 2 protein (PDB ID: 4L33) forms a hydrogen bond with the hydrogen from residue ILE-1051. When we tried to dock the ligand directly with the vina scoring function by Uni-Dock, the obtained binding pose failed to reproduce this hydrogen bond, as shown in Figure 8b. However, by adding a hydrogen bond donor bias near the ILE-1051, as shown in Figure 8c (depicted as red spheres), a biased docking using Uni-Dock with the vina scoring function was conducted. As expected, the binding pose dock succeeded in reproducing the hydrogen bond in the predicted binding pose, as shown in Figure 8c, indicating that the biased docking implemented in Uni-Dock should be helpful especially when the biased interaction in the system is known.

Comparison between AutoDock-GPU and Vina-GPU of Speed and Accuracy. We tested the virtual screening runtime of the same DUD-E ligand−receptor systems with 29 664 ligands on AutoDock-GPU, Vina-GPU, and Uni-Dock, and find that, under comparable settings, Uni-Dock is the fastest docking engine.

We set up the experiment of Vina-GPU using exactly the same parameters as Uni-Dock, which guarantees almost the same accuracy, since they both use the Monte Carlo searching method. Specifically, we run Vina-GPU in *fast* (*nthread* = 128, *nstep* = 20), *balanced* (*nthread* = 384, *nstep* = 40), and *detailed* (*nthread* = 512, *nstep* = 40) modes and report the average runtime of Vina-GPU for each ligand. Since Vina-GPU has not supported batch docking, only one ligand is docked in each execution of Vina-GPU.

We set up the experiment of AutoDock-GPU using the settings that are most comparable to Uni-Dock in searching complexity and accuracy performance. The *nrun* parameter describes the number of independent Lamarckian Genetic Algorithm (LGA) runs in AutoDock-GPU, which defines the searching complexity. As discussed in the work of Vieira and Sousa,[35] which compares the docking accuracy of AutoDock4 and Vina, *nrun* = 10 is sufficient to reproduce successful poses (RMSD < 2 Å) in redocking, comparable to *exhaustiveness* = 8 of Vina in screening and docking power. In Santos-Martins et al.,[26] *nrun* is set to 100 to have better statistics, while the searching performance basically converges with *nrun* = 20. Therefore, we choose *nrun* = 4, 10, and 20, respectively, to compare its performance to *fast*, *balanced*, and *detailed* modes of Uni-Dock. The AutoDock-GPU version we used is the nonpipelined version provided by a NGC AutoDock-GPU container. [See https://catalog.ngc.nvidia.com/orgs/hpc/containers/autodock.] For the pipelined version, please refer to another paper.[28]

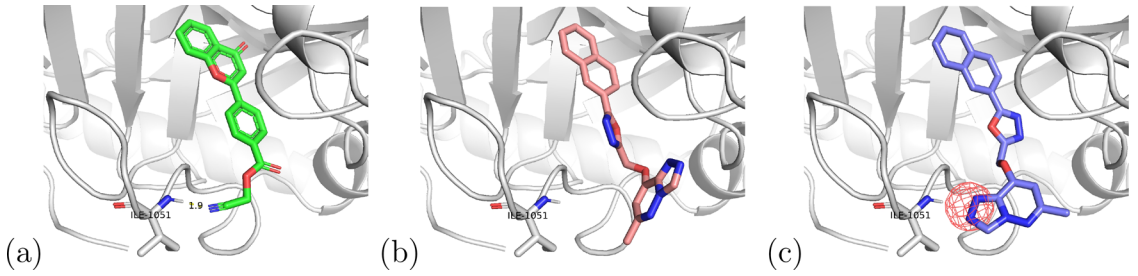


Figure 8. Showcase of biased docking using Uni-Dock with the vina scoring function. (a) The crystal structure (PDB ID: 4L33) in which there is a hydrogen bond between residue ILE-1051 and the cocrystallized ligand. (b) The binding pose of the example compound docked using Uni-Dock with the vina scoring function. (c) The binding pose of the example compound docked by biased docking using Uni-Dock with vina scoring function. The red sphere is the hydrogen bond acceptor bias that we added.

https://doi.org/10.1021/acs.jctc.2c01145
J. Chem. Theory Comput. 2023, 19, 3336−3345

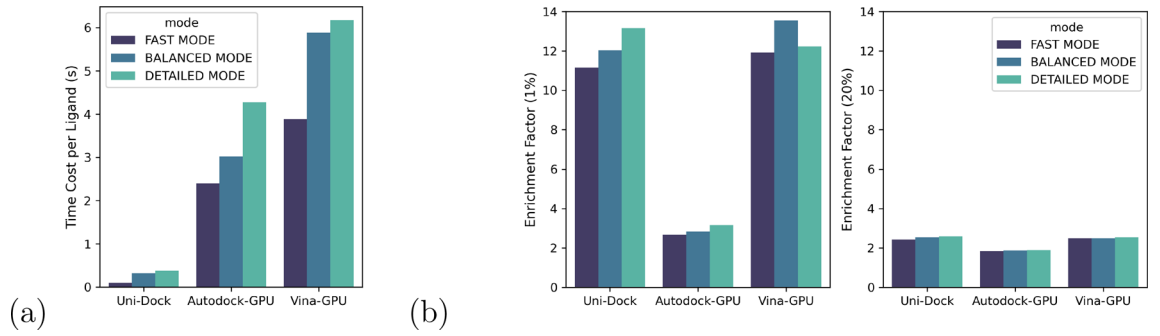


Figure 9. Running time and accuracy of Uni-Dock, Autodock-GPU, and Vina-GPU in three modes using receptor−ligand systems from the DUD-E dataset. Note that the *fast*, *balanced*, and *detailed* modes of Autodock-GPU stand for *nrun* = 4, 10, and 20, respectively: (a) docking time per ligand and (b) enrichment factor ((left) 1% and (right) 20%).

As demonstrated in Figure 9, Uni-Dock is over 10 times faster, compared to the two currently reported GPU-accelerated docking programs under comparable docking parameters in all three modes, while maintaining comparable or even better screening accuracy reflected by enrichment factors tested on the DUD-E dataset. The performance in the accuracy of AutoDock-GPU is worse than our expectation, which might be caused by the selection of receptor systems. Five of our eight systems had $EF_{top\ 1\%}$ values of zero, whereas two systems achieved normal $EF_{top\ 1\%}$ values. In terms of docking power, we tested AutoDock-GPU under the same conditions as Uni-Dock with *nrun* = 20. The RMSD of binding poses predicted by AutoDock-GPU from the crystal structure is within 2 Å in 52.67% (68.07% for that of Uni-Dock) of targets in the CASF-2016 dataset. As for Vina-GPU, the identical strategy of applying Monte Carlo/BFGS searching method with the vina scoring function actually guarantees its similar accuracy with Uni-Dock intrinsically. As a result, we can conclude that Uni-Dock achieves significantly faster speed than AutoDock-GPU and Vina-GPU, while it has comparable or even better performance in both terms of screening power and docking power.

CONCLUSION

In this work, we proposed Uni-Dock, a GPU-accelerated molecular docking program supporting the vina, vinardo, and ad4 scoring functions. The significant high efficiency greatly reduced the time and computation resource cost of the high-throughput virtual screening on ultralarge molecular databases.

We implemented parallel MC search threads in CUDA to cover the search space, and took the strategy of increasing the number of MC search threads and reducing the number of search steps of each MC search thread to shorten the time cost of exhaustiveness searching. At the same time, we implemented the parallel docking of multiple ligands with dynamic batch processing based on accurate memory space prediction, thereby making full use of the graphical memory space and computing power. In addition, we reduced the data exchange between CPU and GPU by migrating the whole calculation process. We also reduced the precision level of GPU calculations to allocate more ligands in single threads with negligible precision loss.

Studies on large benchmarks demonstrate that Uni-Dock keeps comparable docking accuracy to AutoDock Vina 1.2, and the average time cost to dock a single molecule is reduced to 0.1 s for the vina and vinardo scoring functions and to 0.5 s (equivalent to more than 1000 times speedup than CPU), for the ad4 scoring function on one NVIDIA V100 GPU. We conducted a hierarchical virtual screening with Uni-Dock against a recently popular drug target, KRAS G12D, on an Enamine Diverse REAL druglike set containing 38.2 million ligands using a GPU cluster of 100 NVIDIA V100 GPUs. Less than 12 h of calculation time showed Uni-Dock helps to make the ultralarge scale virtual screening more feasible.

Moreover, biased docking is supported in Uni-Dock, not only for hydrogen constraints, but also for any other given atom type and is available for all embedded scoring functions. Uni-Dock is suitable for multiple GPU architectures; one can easily use it on either the NVIDIA or AMD platform. We also proved the linear speedup of Uni-Dock with strong scaling on eight V100 GPUs. Head-to-head experiments were also set up, which showed that Uni-Dock is more than 10 times faster than Autodock-GPU and Vina-GPU with comparable or better accuracy.

Overall, Uni-Dock shows high efficiency in accelerating large-scale virtual screening tasks, providing a feasible solution for virtual screening and binding affinity evaluation on ultralarge molecular databases. To the best of our knowledge, Uni-Dock should now be the fastest GPU-accelerated docking program. We believe that Uni-Dock should be easy to integrate with active learning methods such as DeepDocking,[10] thus making it possible to finish billion-level virtual screening quickly and inexpensively to provide sufficient training data. Uni-Dock's great computing capability makes it possible to generate a large amount of docking data, which will help the machine learning methods to gain more virtual data for pretraining models to improve model performance on the tasks related to the protein−ligand complex structure and binding affinity, or to accelerate the model training and interpreting.

ASSOCIATED CONTENT

Data Availability Statement

Uni-Dock is available on the online drug design platform Hermite (https://hermite.dp.tech), which can call hundreds of GPUs at the same time, and use Uni-Dock to quickly complete large-scale virtual screening. A version with frequent updates of Uni-Dock is also available for free on https://labs.dp.tech/projects/uni-dock-serving/, where users can test the latest features of Uni-Dock.

https://doi.org/10.1021/acs.jctc.2c01145
J. Chem. Theory Comput. 2023, 19, 3336−3345

AUTHOR INFORMATION

Corresponding Authors

Zhengdan Zhu − *Beijing DP Technology Co., Ltd., Beijing 100080, China*; orcid.org/0000-0001-9260-6226; Email: zhuzd@dp.tech

Hang Zheng − *Beijing DP Technology Co., Ltd., Beijing 100080, China*; Email: zhengh@dp.tech

Authors

Yuejiang Yu − *Beijing DP Technology Co., Ltd., Beijing 100080, China; School of EECS, Peking University, Beijing 100871, China*; orcid.org/0000-0002-5051-5948

Chun Cai − *Beijing DP Technology Co., Ltd., Beijing 100080, China*

Jiayue Wang − *Beijing DP Technology Co., Ltd., Beijing 100080, China*

Zonghua Bo − *Beijing DP Technology Co., Ltd., Beijing 100080, China*; orcid.org/0000-0003-4728-2968

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.2c01145

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors thank Linfeng Zhang, Yuqing Deng, Denghui Lu, Zhaohan Ding, Yannan Yuan, Xinyu Li, Dongdong Wang, Kexin Chu, Tao Xu, Junhan Chang, Xiaokuang Bai, and many colleagues in DP Technology for their great help in this project. Most experiments are carried out on the cloud platform Bohrium (https://bohrium.dp.tech).

REFERENCES

(1) Walters, W. P.; Stahl, M. T.; Murcko, M. A. Virtual screening—an overview. *Drug Discovery Today* **1998**, *3*, 160−178.

(2) Kumar, A.; Zhang, K. Y. Hierarchical virtual screening approaches in small molecule drug discovery. *Methods* **2015**, *71*, 26−37.

(3) Gimeno, A.; Ojeda-Montes, M. J.; Tomás-Hernández, S.; Cereto-Massagué, A.; Beltrán-Debón, R.; Mulero, M.; Pujadas, G.; Garcia-Vallvé, S. The light and dark sides of virtual screening: what is there to know? *Int. J. Mol. Sci.* **2019**, *20*, 1375.

(4) Cheng, T.; Li, X.; Li, Y.; Liu, Z.; Wang, R. Comparative assessment of scoring functions on a diverse test set. *J. Chem. Inf. Model.* **2009**, *49*, 1079−1093.

(5) Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Comparative assessment of scoring functions: the CASF-2016 update. *J. Chem. Inf. Model.* **2019**, *59*, 895−913.

(6) 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.

(7) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *J. Med. Chem.* **2012**, *55*, 6582−6594.

(8) Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K.; et al. Ultra-large library docking for discovering new chemotypes. *Nature* **2019**, *566*, 224−229.

(9) Shivanyuk, A.; Ryabukhin, S.; Tolmachev, A.; Bogolyubsky, A.; Mykytenko, D.; Chupryna, A.; Heilman, W.; Kostyuk, A. Enamine real database: Making chemical diversity real. *Chem. Today* **2007**, *25*, 58−59.

(10) Gentile, F.; Agrawal, V.; Hsing, M.; Ton, A.-T.; Ban, F.; Norinder, U.; Gleave, M. E.; Cherkasov, A. Deep docking: a deep learning platform for augmentation of structure based drug discovery. *ACS Central Sci.* **2020**, *6*, 939−949.

(11) Luttens, A.; Gullberg, H.; Abdurakhmanov, E.; Vo, D. D.; Akaberi, D.; Talibov, V. O.; Nekhotiaeva, N.; Vangeel, L.; De Jonghe, S.; Jochmans, D.; et al. Ultralarge virtual screening identifies SARS-CoV-2 main protease inhibitors with broad-spectrum activity against coronaviruses. *J. Am. Chem. Soc.* **2022**, *144*, 2905−2920.

(12) Morris, G. M.; Huey, R.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S.; Olson, A. J. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *J. Comput. Chem.* **2009**, *30*, 2785−2791.

(13) Trott, O.; Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **2009**, *31*, 455−461.

(14) 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.; et al. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J. Med. Chem.* **2004**, *47*, 1739−1749.

(15) Halgren, T. A.; Murphy, R. B.; Friesner, R. A.; Beard, H. S.; Frye, L. L.; Pollard, W. T.; Banks, J. L. Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. *J. Med. Chem.* **2004**, *47*, 1750−1759.

(16) Gorgulla, C.; Boeszoermenyi, A.; Wang, Z.-F.; Fischer, P. D.; Coote, P. W.; Padmanabha Das, K. M.; Malets, Y. S.; Radchenko, D. S.; Moroz, Y. S.; Scott, D. A.; et al. An open-source drug discovery platform enables ultra-large virtual screens. *Nature* **2020**, *580*, 663−668.

(17) Eberhardt, J.; Santos-Martins, D.; Tillack, A. F.; Forli, S. AutoDock Vina 1.2. 0: New docking methods, expanded force field, and python bindings. *J. Chem. Inf. Model.* **2021**, *61*, 3891−3898.

(18) Nocedal, J.; Wright, S. J. *Numerical Optimization*; Springer, 1999.

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

(20) 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*, 1−13.

(21) Yang, Y.; Yao, K.; Repasky, M. P.; Leswing, K.; Abel, R.; Shoichet, B. K.; Jerome, S. V. Efficient exploration of chemical space with docking and deep learning. *J. Chem. Theory Comput.* **2021**, *17*, 7106−7119.

(22) 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. Cheminform.* **2021**, *13*, 1−20.

(23) Stärk, H.; Ganea, O.; Pattanaik, L.; Barzilay, R.; Jaakkola, T. Equibind: Geometric deep learning for drug binding structure prediction. *Int. Conf. Mach. Learn.* **2022**, 20503−20521.

(24) Wong, F.; Krishnan, A.; Zheng, E. J.; Stärk, H.; Manson, A. L.; Earl, A. M.; Jaakkola, T.; Collins, J. J. Benchmarking AlphaFold-enabled molecular docking predictions for antibiotic discovery. *Mol. Syst. Biol.* **2022**, *18*, e11081.

(25) Wójcikowski, M.; Ballester, P. J.; Siedlecki, P. Performance of machine-learning scoring functions in structure-based virtual screening. *Sci. Rep.* **2017**, *7*, 1−10.

(26) Santos-Martins, D.; Solis-Vasquez, L.; Tillack, A. F.; Sanner, M. F.; Koch, A.; Forli, S. Accelerating AutoDock4 with GPUs and gradient-based local search. *J. Chem. Theory Comput.* **2021**, *17*, 1060−1073.

(27) LeGrand, S.; Scheinberg, A.; Tillack, A. F.; Thavappiragasam, M.; Vermaas, J. V.; Agarwal, R.; Larkin, J.; Poole, D.; Santos-Martins, D.; Solis-Vasquez, L. GPU-accelerated drug discovery with docking on the summit supercomputer: Porting, optimization, and application to COVID-19 research. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2020; pp 1−10.

(28) Glaser, J.; Vermaas, J. V.; Rogers, D. M.; Larkin, J.; LeGrand, S.; Boehm, S.; Baker, M. B.; Scheinberg, A.; Tillack, A. F.; Thavappiragasam, M.; et al. High-throughput virtual laboratory for drug discovery using massive datasets. *Int. J. High Perform. Comput. Appl.* **2021**, *35*, 452−468.
(29) 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.
(30) Wishart, D. S.; Feunang, Y. D.; Guo, A. C.; Lo, E. J.; Marcu, A.; Grant, J. R.; Sajed, T.; Johnson, D.; Li, C.; Sayeeda, Z.; et al. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* **2018**, *46*, D1074−D1082.
(31) Wang, X.; Allen, S.; Blake, J. F.; Bowcut, V.; Briere, D. M.; Calinisan, A.; Dahlke, J. R.; Fell, J. B.; Fischer, J. P.; Gunn, R. J.; et al. Identification of MRTX1133, a noncovalent, potent, and selective KRASG12D inhibitor. *J. Med. Chem.* **2022**, *65*, 3123−3133.
(32) Genheden, S.; Ryde, U. The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities. *Expert Opinion on Drug Discovery* **2015**, *10*, 449−461.
(33) Lu, N.; Kofke, D. A. Accuracy of free-energy perturbation calculations in molecular simulation. I. Modeling. *J. Chem. Phys.* **2001**, *114*, 7303−7311.
(34) Arcon, J. P.; Modenutti, C. P.; Avendaño, D.; Lopez, E. D.; Defelipe, L. A.; Ambrosio, F. A.; Turjanski, A. G.; Forli, S.; Marti, M. A. AutoDock Bias: improving binding mode prediction and virtual screening using known protein−ligand interactions. *Bioinformatics* **2019**, *35*, 3836−3838.
(35) Vieira, T. F.; Sousa, S. F. Comparing AutoDock and Vina in Ligand/Decoy Discrimination for Virtual Screening. *Appl. Sci.* **2019**, *9*, 4538.

https://doi.org/10.1021/acs.jctc.2c01145
J. Chem. Theory Comput. 2023, 19, 3336−3345