Title

Retrieval Augmented Docking Using Hierarchical Navigable Small Worlds

Journal

Journal of Chemical Information and Modeling, 64(19)

ISSN

1549-9596

Authors

Hall, Brendan W
Keiser, Michael J

Publication Date

2024-10-03

DOI

10.1021/acs.jcim.4c00683

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

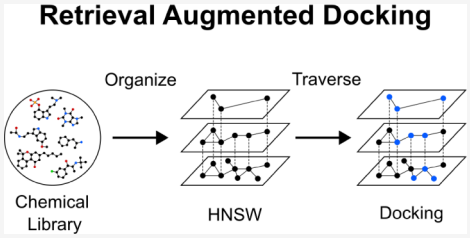
Retrieval Augmented Docking Using Hierarchical Navigable Small Worlds

Brendan W. Hall and Michael J. Keiser*

Cite This: J. Chem. Inf. Model. 2024, 64, 7398−7408 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Make-on-demand chemical libraries have drastically increased the reach of molecular docking, with the enumerated ready-to-dock ZINC-22 library approaching 6.4 billion molecules (July 2024). While ever-growing libraries result in better-scoring molecules, the computational resources required to dock all of ZINC-22 make this endeavor infeasible for most. Here, we organize and traverse chemical space with hierarchical navigable small-world graphs, a method we term retrieval augmented docking (RAD). RAD recovers most virtual actives, despite docking only a fraction of the library. Furthermore, RAD is protein-agnostic, supporting additional docking campaigns without additional computational overhead. In depth, we assess RAD on published large-scale docking campaigns against D4 and AmpC spanning 99.5 million and 138 million molecules, respectively. RAD recovers 95% of DOCK virtual actives for both targets after evaluating only 10% of the libraries. In breadth, RAD shows widespread applicability against 43 DUDE-Z proteins, evaluating 50.3 million associations. On average, RAD recovers 87% of virtual actives while docking 10% of the library without sacrificing chemical diversity.



INTRODUCTION

Virtual screening methods attempt to computationally identify ligands with desired properties from an immense sea of an estimated 10^{60} drug-like molecules.[1] Structure-based molecular docking assesses hundreds of thousands of ligand configurations within a binding site, evaluating each ligand using a physics-based scoring function. Molecular docking requires enumerated virtual libraries of small molecules specifically prepared for use in docking software.

Virtual library sizes have significantly expanded with "make-on-demand" virtual libraries. These libraries of readily synthesizable molecules combine hundreds of relatively simple reactions and hundreds of thousands of building blocks.[2] For instance, the enumerated REAL database from the chemical supplier Enamine[3] has gone from 1.95 billion molecules (May 2021) to 6.75 billion molecules (July 2024). Concurrently, the unenumerated Enamine REAL Space[4] has increased from 19 billion molecules to 48 billion molecules. Emerging trends suggest that larger libraries result in better fitting and better scoring molecules.[5] Consequently, many efforts focus on docking increasingly extensive billion-scale libraries.[6−10] However, the continuous growth of these virtual libraries poses significant computational challenges. Docking all 6.4 billion 3D ZINC-22[11] molecules (July 2024), a database of "ready-to-dock" molecules purchasable through chemical suppliers such as Enamine and WuXi, at a rate of 1 s per molecule would take approximately 203 CPU years. Hence, screening enumerated libraries on the scale of tens of billions requires new methods.

Recently, many researchers have focused on using machine and active learning (ML/AL) methods to facilitate ultralarge-scale screening.[12−16] In these workflows, they dock a small subset of the chemical library to a target of interest and train an ML model to predict the docking score. They use this ML model to predict the scores for the entire library (Figure 1A, machine learning), which we subsequently refer to as "single-iteration" models. However, previous work obtained more accurate predictions for some targets and scoring functions through active learning. In this process, they used the ML model to predict scores for the entire library and a selection criterion to choose additional molecules to dock and retrain the model.[12,13,16] This iterative process, encompassing successive cycles of docking, model training, and prediction, aims to enhance the ability to identify high-scoring molecules from the virtual library (Figure 1A, active learning). Due to the ability of ML models to make rapid but accurate predictions of docking scores, this workflow can reduce the computation time required to screen ultralarge libraries. However, this approach has its limitations:

(1) Although ML predictions are faster than direct docking calculations, the necessity for repeated cycles of active learning introduces a significant computational overhead. This overhead includes repeated model training and inference across the entire library, which is costly, particularly for ultralarge

Received: April 22, 2024
Revised: September 17, 2024
Accepted: September 18, 2024
Published: October 3, 2024

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398−7408

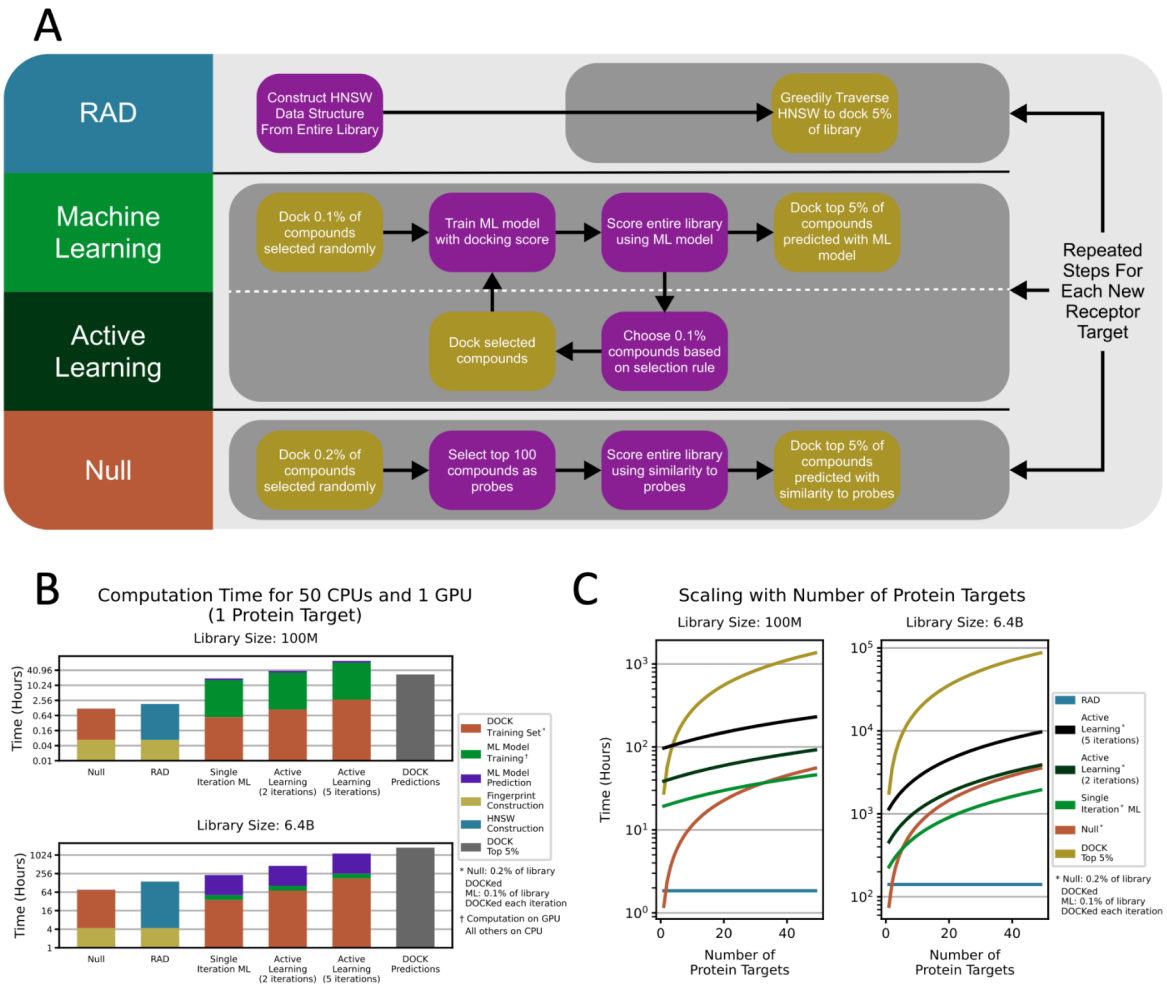


Figure 1. Comparison of workflows to accelerate docking. (A) Virtual screening workflow using RAD compared to the active learning and null workflows described by Yang et al.[12] (B) Time required to prepare each method and perform the DOCK calculations common to all methods. (C) Ability of each method to scale to screen multiple protein targets. Machine learning timing estimates were performed assuming that only a single model was needed to make predictions for all protein targets. The timing estimates assume the model training occurs on a GPU (NVIDIA GTX 1080Ti) and all other calculations occur on the CPU (Intel i5-8400 for DOCK calculations and model prediction and Intel Xeon Gold 6240R for fingerprint construction and HNSW construction).

chemical libraries containing tens of billions of molecules. While strategies like design space pruning[17] aim to minimize the number of ML predictions required in successive active learning iterations, an initial comprehensive library evaluation is unavoidable. Furthermore, each active learning cycle requires model training and validation, contributing to the overall computational overhead. While researchers can limit model training and inference to a single iteration for efficiency, previous work demonstrated that this can perform worse than an active learning workflow.[12,13] Additionally, this strategy still requires a comprehensive evaluation of the entire library using the ML model.

(2) Every virtual screening campaign requires its own ML model. Although there have been attempts to develop general ML models capable of predicting scores across multiple proteins, their accuracy is worse than target-specific models,[18,19] and performance degrades when applied to unseen proteins.[20] Consequently, to achieve accurate results in screening campaigns involving multiple protein targets, researchers must carefully tailor their active learning workflow. In the worst case, this requires tuning, training, and validating models for every protein binding site of interest. In the best case, a single model can make predictions for all proteins of interest at once, and only a single ML model is needed. However, in both cases, the acquisition of docking training data scales linearly with the number of targets in the campaign. Furthermore, the use of ML models requires expertise and computational overhead to validate that they are not biased or overfitted, and each additional screening campaign requires its own model.

Alternatively, de novo generative methods[21−23] and those that operate directly on molecular building blocks and reactions[24,25] sidestep the enumeration problem. While de novo generative methods may explore a larger or complementary chemical space, generated molecules are frequently impossible to synthesize.[26] Likewise, methods that screen unenumerated libraries explore a large chemical space using fragment-based screening, but docking scoring functions may not accurately determine fragment binding modes[27] and rank their typically low affinities.[28] Furthermore, to our knowledge, no benchmark data sets enable a comprehensive comparison between enumerated and unenumerated methods. Given these considerations, we focus on exploring the enumerated chemical space.

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398−7408

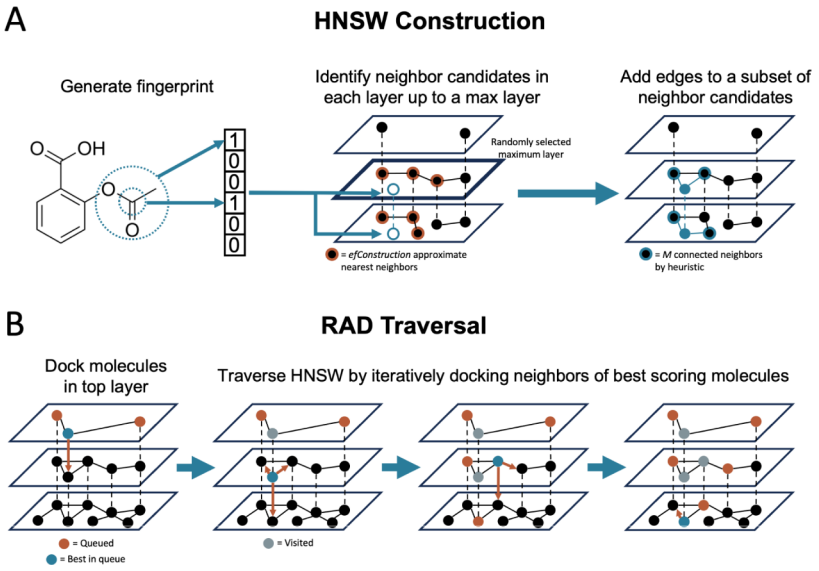


Figure 2. Hierarchical navigable small world (HNSW) graph construction and retrieval augmented docking (RAD). (A) We add molecules to the HNSW structure in every layer below a randomly assigned maximal layer. A heuristic connects M nodes from *ef Construction* approximate nearest neighbors in each layer. (B) RAD begins by scoring the entire top layer and continues by greedily traversing the graph structure prioritized by the dock score.

This study introduces a new way to navigate chemical space using hierarchical navigable small worlds (HNSW), a hierarchical graph-based data structure and algorithm designed for approximate nearest neighbor (ANN) search within high-dimensional spaces.[29] Although HNSW has predominantly served as the foundation for vector databases[30,31] powering retrieval augmented generation (RAG),[32] we leverage HNSW to structure virtual chemical libraries. Instead of following a conventional ANN search, we implement a greedy algorithm to traverse and dock the molecules within the graph's underlying structure in a process we call retrieval augmented docking (RAD).

RAD efficiently recovers a significant proportion of the best-scoring molecules (virtual actives) within the entire chemical library while evaluating only a fraction of the total library. Moreover, this method exhibits comparable performance to the single-iteration models and multi-iteration active learning models presented by Yang et al.[12] and many of the active learning models presented by Graff et al.[13] while addressing some limitations: (1) RAD exploits the intrinsic graph structure of HNSW, necessitating score calculations for only a subset of the library. This is a notable improvement over the machine learning approach, which requires ML predictions for the entire library potentially multiple times for an active learning loop. We used explicit docking calculations as the scoring function during traversal. Nonetheless, the HNSW structure can be traversed with any scoring function, such as an ML-based scoring function, or even be integrated within an active learning framework, eliminating the need for model predictions across the entire library. (2) Because the HNSW data structure involves only the virtual chemical library, its traversal is "just-in-time". Consequently, undertaking a new screening campaign does not require the reconstruction of the HNSW and does not incur additional computational overhead. Our method solely focuses on relating ligands' biological activity, measured by the docking score, to their organization based on similarity, akin to many previous methods.[33−36] This contrasts with the active learning approach, which has the prerequisite of developing and tuning a campaign-specific ML model. Launching a new campaign requires training a new model, unlike the HNSW which can be reused for any number of screening campaigns.

MATERIALS AND METHODS

Data Sets. *AmpC and D4.* We evaluated the retrospective applicability of RAD in large-scale virtual screening using two protein systems. We obtained results for 99 459 562 molecules docked to AmpC β-lactamase (AmpC) and 138 312 677 to the D4 dopamine receptor (D4) using DOCK 3.7 from Lyu et al.[37] We obtained Glide[38,39] docking results for the same molecules and proteins from Yang et al.[12]

DUDE-Z. To assess the reusability of the HNSW structure across many protein targets, we created a data set of 1 169 461 molecules provided by the DUDE-Z "Goldilocks" set docked to the 43 proteins included in the DUDE-Z data set[40] using default DOCK 3.7[41] with no modifications. We used the default parameters, INDOCK files, and grids provided by the DUDE-Z data set for each protein with no modifications. Molecules that failed to dock to a given protein were assigned scores of infinity for that protein. On average, 89% of the molecules per protein were successfully docked and scored, each exploring 5226 orientations and 294 conformations.

Hierarchical Navigable Small World (HNSW) Graph. HNSW is a data structure and algorithm designed for approximate nearest neighbor search that leverages a hierarchical multilayer graph where subsets of elements (molecules) exist in the upper layers and all of the molecules in the bottom layer. These sparse upper layers act as a coarse-graining of the entire data set and allow for the rapid ($O(\log N)$) identification of nearby neighbors in high-dimensional spaces. The HNSW algorithm constructs this hierarchical graph structure by sequentially inserting molecules. Each molecule in the data set is inserted into the graph according to the following procedure.

First, the algorithm assigns a molecule a maximum layer, l, in the hierarchy by sampling an exponentially decaying random

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398−7408

variable. The molecule will be inserted in the graph at this maximum layer and all layers below it (l, $l-1$, ..., 0). The exponentially decaying variable ensures the sparsity of the upper layers of the graph. Only a few molecules will be randomly assigned to the upper levels of the hierarchy. Note that molecules are inserted into all layers below their assigned maximum, so every molecule appears in the bottom layer 0.

Once a molecule is assigned a maximum layer l, it is inserted into layers l, $l-1$, ..., 0. Molecule insertion employs a greedy search to locate *ef Construction* approximate nearest neighbors per layer based on the Tanimoto similarity between molecules. We refer to these as candidates, as they will potentially be connected to the newly inserted molecule. At each layer, M out of the *ef Construction* candidates ($2 \cdot M$ in the bottom layer 0) are chosen to be bidirectionally connected to the inserted molecule in the graph. The following strategy is employed to choose the final M neighbors from the *ef Construction* candidates to connect to the newly inserted molecule.

First, the algorithm connects the inserted molecule to the nearest candidate. It then proceeds to the next nearest candidate. This candidate is connected if it is more similar to the inserted molecule than any of the already connected candidates; otherwise, it is skipped. This process continues until the algorithm connects M neighbors to the inserted molecule or evaluates all of the *ef Construction* candidates (Figure 2A). According to the original paper, this strategy encouraged diverse connections among highly clustered data. Modifying the parameters *ef Construction* and M influenced the HNSW construction speed, memory requirements, and recall rate of our RAD traversal method. These steps are repeated for each molecule in the data set to construct the graph.

We developed a molecular HNSW using the open-source library hnswlib,[29] albeit with two primary adaptations for molecular fingerprint processing: Tanimoto distance calculations for the HNSW construction distance metric[42] and integer data types for memory-efficient fingerprint storage and expedited Tanimoto calculations.[43] Using RDKit,[44] we converted the chemical library into Morgan fingerprints with a radius of 2 and a length of 1024 bits. We constructed the HNSW with an *ef Construction* of 400 and an M of 16. We chose fingerprints and HNSW parameters that demonstrated the best average recall of DUDE-Z virtual actives when applying RAD to screen up to 10% of the chemical library (Figure S1 and Table S3). While these parameters yielded the best recall of virtual actives in this context, we note that many other parameter choices resulted in similar performance while reducing HNSW construction speed and memory usage (Figure S2). Finally, we converted the flat HNSW C++ data structure produced by hnswlib into a series of Python graph-tool graphs[45] for easier traversal.

RAD and Null Traversal. We quantitatively compared two traversal techniques for identifying virtual actives within chemical libraries while minimizing the fraction of the explicitly docked library.

The first technique, RAD, used the docking scoring function to greedily traverse the molecular HNSW graph. All molecules in the top HNSW layer were docked and added to a priority queue, with better-scoring molecules having higher priorities. The traversal then iteratively removed the highest-scoring molecule from the queue and retrieved its unvisited neighboring molecules from the HNSW graph (including the molecule at a lower level in the HNSW hierarchy). These molecules were docked, scored, and added to the priority queue (Figure 2B). This cycle continued until a predefined stop criterion was satisfied. In this work, we stopped this process once it scored 10% of the virtual library.

The alternate technique, known as the null traversal and introduced by Yang et al.,[12] randomly docked 0.2% of the library (2% for the DUDE-Z library) and assigned the 100 best-scoring molecules to a probe library. The remainder of the library molecules was scored based on their maximal Tanimoto similarity to any probe library member. Similarity calculations employed Morgan fingerprints with a radius of 2 and a length of 1024 bits, mirroring the fingerprints used in the HNSW construction.

Qualitatively, we compared the performance of both techniques against the single-iteration and active learning strategies employed by Yang et al.,[12] but we could not compare performance numerically due to the unavailability of that study's codebase. Additionally, we compared the performance against the active learning strategies employed by Graff et al.[13]

Performance Metrics. We evaluated the ability of traversal strategies to identify virtual actives using recall metrics. We defined "virtual actives" as the top ~0.01% scoring molecules in a screening library (top 10k for D4 and AmpC libraries and top 100 for DUDE-Z). Different traversals were compared by their partial area under the curve (pAUC), representing the recall of virtual actives against the screened percentage of the library. Because of the computational costs associated with docking, we were particularly interested in the ability to identify virtual actives while screening only a fraction of the library. To this end, we measured the pAUC associated with screening up to 10% of the library, denoted as $pAUC_{10}$.

Additionally, we assessed the trade-off between top-scoring molecules and structural diversity by measuring the recall of the Bemis–Murcko scaffolds[46] and the Butina clusters[47] of the virtual actives. The Bemis–Murcko scaffolds represent the core structures of molecules by removing the side chain atoms and focusing on the central ring systems and linkers. Butina clustering groups molecules into clusters based on their relative similarity using Tanimoto similarity. These molecular diversity methods complement each other as frameworks for evaluating the core structural components and the relative similarity of the molecules, respectively. We sought to capture different aspects of diversity: scaffolds offer an absolute measure by identifying distinct core structures, while clusters provide a relative measure by grouping molecules based on their similarities within the same library. Maintaining chemical diversity is crucial for virtual screening methods, particularly when applied prospectively, as it avoids redundant testing of similar molecules, which are more likely to have similar properties.[48]

We used RDKit to calculate chemical scaffolds and perform Butina clustering on Morgan fingerprints with a radius of 2 and a length of 1024. The top 10k D4 DOCK virtual actives had 4268 unique Bemis–Murcko scaffolds and 2714 Butina clusters. In contrast, the AmpC DOCK virtual actives encompassed 4519 scaffolds but 3394 clusters (Table S1).

The DUDE-Z molecule library was smaller and less diverse than the D4 and AmpC libraries and did not undergo this diversity analysis. DUDE-Z virtual actives were limited to the top 100 as opposed to the top 10k, and the DUDE-Z Goldilocks library, which averages 3.6 molecules per scaffold,[40] did not accurately represent the diversity typically found in ultralarge screening libraries like ZINC-22, which averages 46.7 molecules per scaffold.[11]

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398–7408

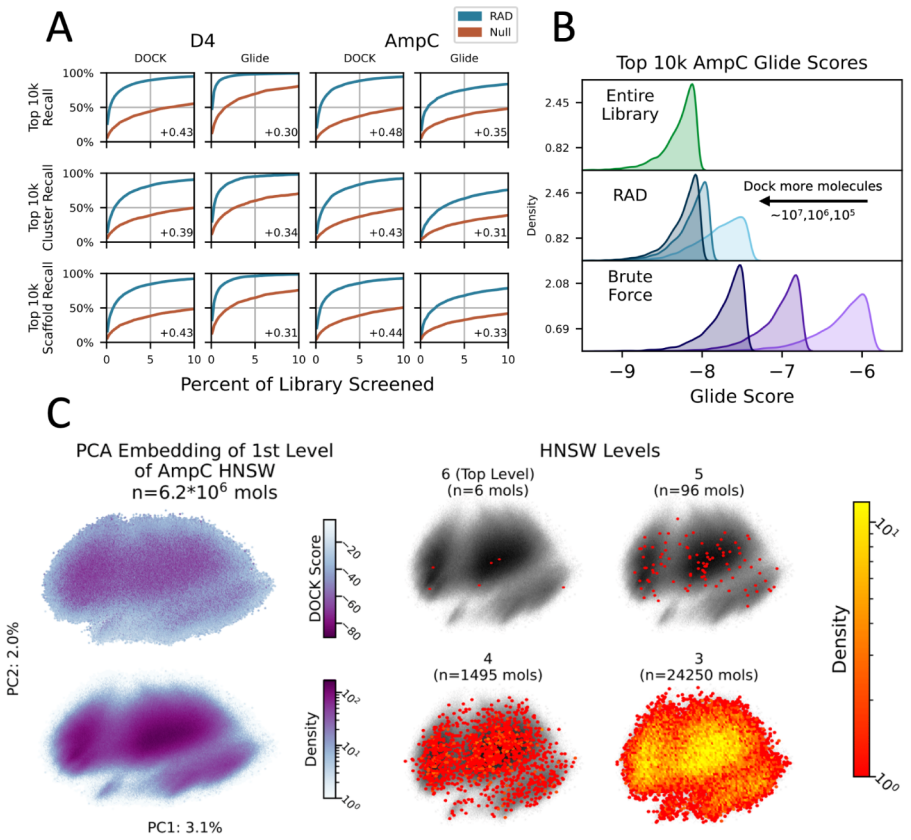


Figure 3. RAD efficiently searches large molecule libraries. (A) RAD versus null recall of the top 10k virtual active molecules, scaffolds, and clusters for the D4 and AmpC libraries using DOCK and Glide. Positive-prefix ("+") numbers indicate the $pAUC_{10}$ increase from null to RAD. (B) Distribution of the top 10k scores in the AmpC Glide screen compared to the top 10k scores from RAD and brute force docking of increasing molecule count. (C) PCA embedding of the 1st level of the AmpC HNSW and the locations of the molecules in the upper layers.

Pocket and Virtual-Active Property Calculations. Because RAD is easily applied to dozens of proteins, we aimed to investigate whether the properties of the protein binding pocket or their virtual actives correlated with the RAD performance. We calculated the protein pocket properties for all 43 DUDE-Z proteins using the program fpocket.[49] We ran fpocket in restricted mode with the crystal ligand provided by DUDE-Z, which allowed fpocket to characterize the ligand binding site explicitly rather than characterize putative binding sites that may not align with the experimentally validated ligand binding site. We leave an exact overview of all descriptors returned by fpocket to the original manuscript,[49] but at a high level, fpocket returned information about each binding site's volume, polarity, hydrophobicity, electrostatics, solvent accessibility, and druggability.

We calculated the properties of the virtual actives for each DUDE-Z protein using the program MOSES.[50] Given a set of molecules, MOSES calculated properties, including the internal diversity defined by Benhenda,[51] the average log *P*, and the average molecular weight. We calculated internal diversity[51] as the average pairwise Tanimoto similarity of a set of molecules, *G*:

$$\mathrm{IntDiv}_p(G) = 1 - \sqrt[p]{\frac{1}{|G|^2}\sum_{m_1,m_2\in G} T(m_1, m_2)^p}$$

where *T* is the Tanimoto similarity. This metric spans from 0 to 1, with higher values indicating greater diversity.

Hyperparameter Optimization. We considered hyperparameters affecting molecular representation and HNSW construction. For the molecular representation, we considered four fingerprint types: Morgan fingerprints with a radius of 2, Morgan fingerprints with a radius of 3,[52] RDKit fingerprints,[44] and MACCS keys.[53] For each fingerprint, we considered five lengths: 128, 256, 512, 1024, and 2048 bits (except for MACCS keys, which always had a length of 166 bits). For HNSW construction, we considered various *ef Construction* and *M* parameters, which controlled the number of neighbor candidates and connected neighbors, respectively, when adding a new vector. We considered four *ef Construction* values: 100, 200, 300, and 400, and six *M* values: 2, 4, 8, 16, 32, and 64. These values were chosen according to reasonable ranges suggested by the authors of the hnswlib library.

We constructed independent DUDE-Z HNSWs for all hyperparameter combinations and evaluated their performance by average $pAUC_{10}$ across all DUDE-Z targets obtained with RAD (Figure S1 and Table S3). We used the hyperparameter combination that obtained the highest average $pAUC_{10}$ across all DUDE-Z targets for the larger D4 and AmpC HNSWs. We compared HNSW construction times, measured on UCSF's Wynton HPC using 50 cores of a 2.00 GHz AMD EPYC 7662, against their $pAUC_{10}$ for each hyperparameter combination (Figure S2).

The storage required for the bottom HNSW layer accounted for most of its memory usage. Each node within the HNSW structure required the following: (1) 4 bytes to specify the number of neighbors, (2) $4\cdot(2\cdot M)$ bytes to store the node IDs

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398−7408

of the (2·M) neighbors on the bottom layer, (3) D/8 bytes to store the fingerprint data where D is the length of the fingerprint, (4) 8 bytes to store the number of set bits, and (5) 8 bytes to store the node ID.

The approximate memory requirement for an HNSW was calculated as

$$\text{Memory} = \text{Library Size}\cdot(20 + 8M + D/8)\text{bytes}$$

The memory requirements for each hyperparameter combination were compared to their $pAUC_{10}$ (Figure S2).

Estimating the Time Required to Screen 6.4 Billion Molecules. To estimate the time necessary to construct an HNSW for the entire 3D ZINC-22 library of 6.4 billion molecules, we extrapolated from the empirical times obtained during the construction of 100 million scale AmpC and D4 HNSWs. The construction time was recorded cumulatively for each 1 million molecules added to the HNSW on 50 cores of a 2.40 GHz Intel Xeon Gold 6240R. The theoretical construction time follows an $N\log(N)$ relationship,[29] and we fit our empirical times to this relationship to estimate the time required to build an HNSW comprising 6.4 billion molecules (Figure S3). We used the slightly slower estimates obtained from the D4 empirical data for this analysis.

We used the average computation times reported by Yang et al.[12] for DOCK, model training, and model inference to approximate the time required to set up an active learning workflow for 6.4 billion molecules. This previous study docked 0.1% of the library to add to the training data at each round of active learning at an average speed of 1 s per molecule per CPU core with DOCK. We assumed that 0.1% of the total library was docked at each round of active learning, regardless of the total size. As such, training set acquisition sizes of 100 000 were used for the 100 million scale library estimates, while training set acquisition sizes of 6.4 million were used for the 6.4 billion 3D ZINC-22 active learning calculations.

Yang et al.[12] reported a 16-h ML training time, which we adopted as a constant regardless of the training set size. We note that this leads to an underestimate of the time required for 6.4-billion scale active learning, as the 16 h training time was based on training set acquisition sizes of ∼100 000. For larger training set acquisition sizes, such as those used for the 3D ZINC-22 estimate, model training would likely take longer. In fact, Sivula et al.[16] investigated billion-scale active learning with training set acquisition sizes of 1.56 million and demonstrated training times closer to 22−35 h.

Lastly, Yang et al.[12] reported a 5 ms per molecule per CPU core ML prediction time. We modeled the DOCK calculations and ML predictions as scaling linearly with the number of CPU cores and predicted the total time required to dock the training set, train the ML model, and perform ML inference for a single-iteration model and two rounds of active learning, as investigated by Yang et al.,[12] as well as five rounds of active learning, as investigated by Graff et al.[13]

RESULTS AND DISCUSSION

RAD Achieves a 100-Fold Speed Up Over Brute-Force Docking. RAD significantly outperformed a naive search, the null traversal (see the Methods section), in identifying virtual actives during the large-scale screening of the 100 million-scale AmpC and D4 libraries (Figure 3A). Despite docking only 10% of the library with DOCK, RAD recovered 95% of D4 and AmpC virtual actives versus the null traversal's 55% and 49%. RAD similarly obtained higher $pAUC_{10}$ values, which measure the area under the curve representing the percentage of virtual actives recovered while docking up to 10% of the library. RAD achieved $pAUC_{10}$ values of 0.84 for D4 and 0.83 for AmpC, more than twice the null traversal's 0.41 and 0.35, respectively. We saw similar trends using Glide (Table S2).

RAD's recall rates were qualitatively similar to the active learning approach presented by Yang et al.[12] (Figure S4; code was unavailable), which required two rounds of deep learning model training per receptor and two rounds of library evaluation. For instance, the Yang et al.[12] active learning workflow recovered ∼87% of the D4 DOCK virtual actives when screening 5% of the library, while RAD recovered ∼90% when screening the same amount. Furthermore, RAD frequently outperformed the single-iteration ML models by Yang et al.,[12] particularly for smaller training batch sizes. For instance, the single-iteration ML model only recovered ∼79%, ∼74%, and ∼85% of the D4 DOCK virtual actives when training batch sizes of 0.1%, 0.2%, and 0.5% were used and screening 5% of the library. For AmpC with Glide, the single-iteration models performed comparably to the active learning models, and these differences were less pronounced. However, when the single-iteration models used training batch sizes smaller than 0.1% or considering only early enrichment, they exhibited worse performance than the active learning models.

RAD's recall rates were worse than the best active learning models presented by Graff et al.,[13] but comparable to or better than many of those investigated (Figure S12). For instance, the best ML architecture investigated in that work (a message-passing neural network) with a training batch size of 0.4% identified 83−95% of the top 50k AmpC DOCK virtual actives and 58−84% of the top 50k D4 DOCK virtual actives while screening 2.4% of the libraries, depending on the active learning acquisition function. In comparison, RAD identified ∼75% of the top 50k AmpC DOCK virtual actives and ∼74% of the top 50k D4 DOCK virtual actives while screening 2.4% of the libraries. While RAD's virtual active recall was lower than the highest-performing active learning models from Graff et al.,[13] these workflows required five rounds of model training and larger training batch sizes of 0.4%, dramatically increasing the active learning computation time (Figure 1B). Furthermore, the performance of RAD was comparable to or better than many of the models with smaller training batch sizes or different ML model architectures. For instance, the feedforward neural network with a 0.2% training batch size by Graff et al.[13] identified 53% of the top 50k AmpC DOCK virtual actives when screening 1.2% of the library. In comparison, RAD identified ∼61% when screening the same number of molecules.

RAD was far more efficient than brute-force docking in finding virtual actives (Figure 3B). For instance, RAD achieved a 100-fold search improvement over brute-force Glide docking for AmpC. When assessing $\sim 10^5$ molecules, RAD found a comparable distribution of top-10k scores to the one that brute-force docking achieved only after $\sim 10^7$ molecules. These trends were consistent across scoring functions and receptors (Figure S5).

We attribute RAD's ability to identify many of the virtual actives while screening a fraction of the library to the multilayered sparse hierarchy of the HNSW structure. The upper levels of the hierarchy span broad regions of chemical space as visualized by PCA (Figures 3C and S13), allowing RAD's early stages to explore more unique scaffolds than the null traversal but fewer than brute-force docking (Figure S6A).

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398−7408

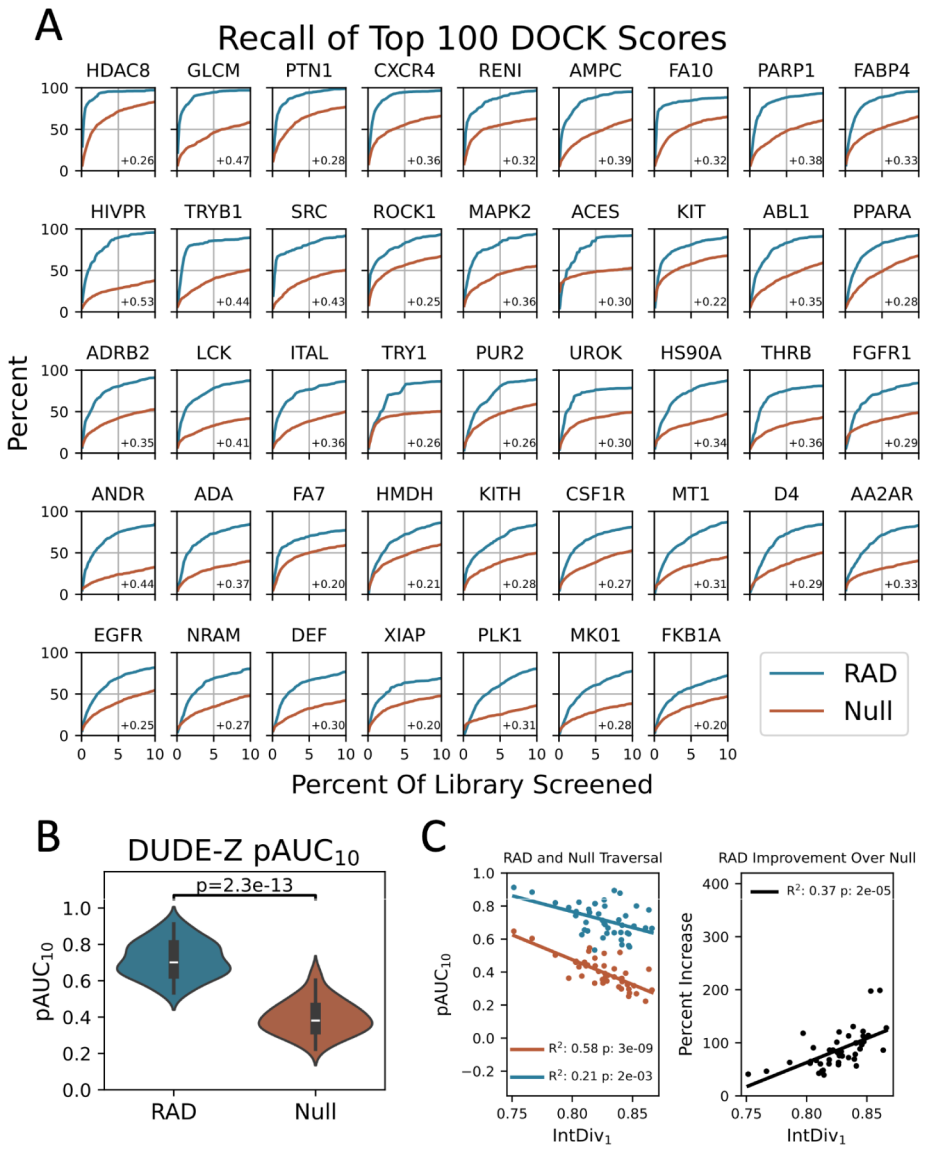


Figure 4. RAD reliably accelerates dock search across 43 DUDE-Z targets. (A) RAD versus null recall of the top 100 virtual active molecules for all 43 DUDE-Z proteins. Positive-prefix ("+") numbers indicate the $pAUC_{10}$ increase from null to RAD. (B) Comparison of RAD and null recall distributions. (C) Correlations between virtual actives' internal diversity ($IntDiv_1$) and the RAD and null recalls.

Despite RAD sampling fewer scaffolds than brute-force docking, the sampled molecules are much more likely to be virtual actives than null traversal or brute-force docking (Figure S6B).

RAD Maintains the Chemical Diversity of Virtual Actives. We found that RAD's efficiency does not necessarily operate at the cost of chemical diversity. To quantify this, we assessed unique scaffolds and clusters within the top virtual actives. Recovery rates of unique Bemis−Murcko scaffolds and Butina clusters for D4 and AmpC virtual actives correlated with simple virtual active score recovery rates (Figure 3A). Within the first 10% of the DOCK D4 library, RAD recovered 91% of the top clusters and 92% of the top scaffolds. For AmpC, RAD recovered 92% of the clusters and 93% of the scaffolds. The null traversal only recovered 50% and 48%, and 49% and 51% of the D4 and AmpC clusters and scaffolds, respectively. These trends were also borne out with Glide (Table S2).

We also found that RAD's recall of top scaffolds and clusters was comparable to the active learning procedure of Yang et al.[12] For instance, when screening 5% of the D4 library, RAD identified 84% and 96% of the top scaffolds compared to active learning's 85% and 98% using DOCK and Glide, respectively. RAD similarly outperformed the single-iteration models' ability to recall the top D4 DOCK clusters. When screening 5% of the D4 DOCK library, RAD identified 82% of the top clusters, qualitatively similar to the active learning workflow of Yang et al.[12] However, the single-iteration models investigated in that work performed worse, recalling fewer than 80% of the top D4 DOCK clusters for all training set sizes. This result was not as pronounced for D4 Glide, where the single-iteration models again performed similarly to the active learning models, except for the smallest training batch sizes.

Interestingly, we found that RAD's recall of top scaffolds and clusters across both proteins and scoring functions lagged behind its recall of virtual actives. For example, when screening 5% of the AmpC library with Glide, RAD identified 75% of

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398−7408

virtual actives, but only 63% of the top scaffolds. One explanation for this discrepancy could be the molecular representations used for the HNSW construction. Because the HNSW uses the Morgan fingerprints and Tanimoto distance, neighbors within the graph have similar fingerprints and are more likely to occupy the same Butina cluster. Despite the sparse upper layers of the HNSW facilitating the exploration of distant regions of chemical space, traversal is inherently constrained to neighbors of good-scoring molecules, potentially impeding the exploration of some clusters. Different representations capture distinct molecular information, influencing the chemotypes considered similar[54] and demonstrating variable performance on downstream biological tasks.[55] We anticipate advancements in these representations, such as those emerging from machine learning,[56,57] will improve the recall rate and diversity of virtual actives identified through RAD.

RAD Preparation is Faster Than Active Learning Preparation. One of RAD's primary advantages over an active learning workflow is its lower computational overhead to prepare a screen. For a library of 100 million molecules, which is the approximate size of the D4 and AmpC libraries, the HNSW preparation time is approximately 20× faster than a two-iteration active learning workflow's preparation and 10× faster than a single-iteration workflow preparation (Figure 1B). ML model training dominates the active learning and single-iteration computation time at this data set size and results in both active learning and single-iteration ML taking longer than RAD. Scaling up, we estimated the approximate time for HNSW, single-iteration ML and active learning preparation for a data set size of 6.4 billion, roughly the number of 3D molecules in ZINC-22. While ML predictions are faster than explicit docking, the sheer volume of predictions for 6.4 billion molecules incurs substantial computational costs and dominates the preparation time for single-iteration ML and active learning. Constructing the HNSW for 6.4 billion molecules is similarly costly but offers an approximately 3.2× speedup over the two-iteration active learning preparation and an approximately 1.6× speedup over the single-iteration ML workflow (Figures 1B and S14). However, we need to build the HNSW data structure only once and can add new molecules to it incrementally thereafter.

Consequently, RAD's efficiency becomes pronounced when screening against multiple protein targets. The protein-agnostic HNSW requires a one-time investment in graph construction, and traversal can begin immediately. In contrast, active learning traversal requires a prerequisite ML model, and acquiring the docking data to train this model scales linearly with the number of targets (Figure 1C). While a single ML model could make predictions for multiple proteins simultaneously, acquiring the docking data to train multitarget models incurs multiplicatively higher costs than RAD, which does not require a trained model for traversal. Intriguingly, the primary time expenditure across all methods, particularly at the 6.4 billion library scale, remains the docking calculations used to traverse the HNSW or rescore the molecules from active learning or null traversals (Figure 1C, right). It has not escaped our notice that a workflow combining ML scoring with HNSW traversals could serve as an exceedingly rapid albeit approximate heuristic (Figure S11), but this is outside the present scope.

RAD Significantly Outperforms the Null Traversal Across 43 Protein Targets. To evaluate the ease of applying RAD to multiple proteins, we constructed a single HNSW for ∼1 million DUDE-Z "Goldilocks" molecules and used it to screen 43 DUDE-Z proteins. In the worst case, performing this task with an active learning workflow would require progressively training and validating 43 individual ML models for each protein target and would take hundreds of hours. Alternatively, a single multitask model may successfully learn to predict scores for all 43 proteins simultaneously, but this would still require multiple hours and rounds of training and inference. Furthermore, ML models predicting docking scores can require more than 50−100k training examples per target,[58] and may not converge effectively at this scale even with active learning. In contrast, constructing an HNSW for 1 million molecules takes only a minute or two and can be reused for each of the 43 proteins.

RAD outperforms the null traversal in the recall of virtual actives for all 43 DUDE-Z proteins (Figure 4A). RAD's average $pAUC_{10}$ of 0.72 across all DUDE-Z targets trounces the null traversal's 0.40 average $pAUC_{10}$. While RAD recalls more virtual actives than the null traversal for all targets, RAD performance spans a large range, from 0.53 $pAUC_{10}$ for FKB1A to 0.91 $pAUC_{10}$ for HDAC8. Furthermore, when using RAD over the null traversal, some targets see much larger improvements than others.

RAD Performance Correlates with the Internal Diversity of the Virtual Actives. The ease with which the same HNSW structure applies across protein targets facilitates a comprehensive analysis of factors that may influence the traversal performance. Accordingly, we investigated the correlation between the properties of the protein binding pockets, the protein's virtual actives, and the performance of RAD.

We find that RAD and the null traversal suffer as the internal diversity of the virtual actives increases, although this correlation is weaker for RAD. This is perhaps unsurprising considering that HNSW construction and the null scoring function exploit Tanimoto similarity and that the same metric is integral to the internal diversity definition. Furthermore, RAD notably outperforms the null traversal when the virtual-active set has high internal diversity (Figure 4C). We find no significant correlation between traversal performance and the virtual actives' average log *P*, molecular weight, or QED score[59] (Figure S7).

Additionally, no protein pocket properties correlate significantly with null traversal (Figure S8) or RAD recall (Figure S9). While the polarity score and charge score weakly correlate with RAD's relative performance vs the null traversal (R^2: 0.10, p: 0.04 and R^2: 0.16, p: 0.01, respectively) (Figure S10), neither is significant under the Bonferroni correction for multiple hypothesis testing (p: 0.72 and p: 0.18, respectively). Future work could explore ligand or pocket property combinations or other properties not calculated here.

Hyperparameter Choices Offer Efficiency Gains with Minimal Performance Loss. RAD's applicability to a wide array of proteins made it possible to programmatically analyze how fingerprint and HNSW hyperparameters influenced virtual active recall with a reduced likelihood of overfitting to a single protein target. We exhaustively constructed HNSWs for all combinations of hyperparameters detailed in the Materials and Methods section and quantitatively compared their performances by the average $pAUC_{10}$ across all DUDE-Z targets. The HNSW using Morgan fingerprints with a radius of 2 and a length of 1024 bits, an *M* of 16, and an *efConstruction* of 400 achieved the highest average $pAUC_{10}$ of 0.72 and required

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398−7408

approximately 0.3 GB of memory and 1.1 min to construct. Because these parameters obtained the highest average $pAUC_{10}$ (Table S3), we used them for all of the HNSWs investigated in this work. These parameters also resulted in the highest average AUC of 0.95 when screening 100% of the library (Table S4) and the fifth highest average log AUC of 0.63 (compared to the maximum value obtained of 0.64), which measures the AUC on a semilog x-axis[60] (Table S5).

Several other hyperparameter combinations achieved slightly lower but similar performance with reduced construction time and memory usage (Figure S2). For instance, the HNSW using Morgan fingerprints with a radius of 2 and a length of 256, an M of 16, and an *ef Construction* of 300 exhibited a high $pAUC_{10}$ of 0.69 (a 4.2% decrease), yet is more efficient: only requiring 42 s to construct (36% decrease) and approximately 0.21 GB of memory (35% decrease). The construction of a 100-million scale HNSW like those used for AmpC and D4 requires approximately 1.8 h and occupies approximately 28 GB of memory when using the highest performing hyperparameters. Opting for the more efficient but slightly less performant hyperparameters mentioned above would reduce the construction time by approximately a quarter and decrease memory requirements to 18 GB. Given these insights, we anticipate that future applications of RAD for ultralarge screening will need to balance performance with computational costs, and we suspect that the easiest way to do this will be by carefully considering the hyperparameter choices.

CONCLUSION

Ultralarge chemical libraries have grown to tens of billions of molecules, requiring new, more efficient screening methods. Previous methods, such as single-iteration machine learning, approached this problem by quickly approximating the docking scores, avoiding the need to prepare molecules and perform expensive explicit docking scores. Further work expanded on this idea by introducing active learning, a technique consisting of iterative cycles of machine learning model training, inference, and retraining. While this technique can improve ML model accuracy for some targets and scoring functions, it does not always outperform the single-iteration approach. Despite the speed of these ML approximations, single-iteration and active learning require inference on the entire chemical library (potentially multiple times), which becomes infeasible at large library sizes.

Here, we propose preorganizing ultralarge chemical libraries using hierarchical navigable small worlds and greedily traversing the underlying graph structure to find top-scoring molecules. This method recovers actionable virtual actives while screening a fraction of the chemical library and addresses several limitations of active learning workflows. Namely, RAD can screen multiple proteins without additional overhead since no prerequisite model is required for traversal, and traversing the graph structure subverts comprehensive library processing.

As ultralarge libraries expand into the hundreds of billions, they will become too large to dock explicitly or comprehensively score by repeated ML prediction. However, any scoring function can guide HNSW structure traversal, so we expect RAD to complement a hundred-billion-scale active learning workflows by vastly reducing the amount of chemical space to explore via rapid ML predictions. Across scales, RAD enables the discovery of high-scoring molecules from much larger libraries than would otherwise be accessible.

ASSOCIATED CONTENT

Data Availability Statement

Open-source code is available at https://github.com/keiser-lab/rad under the MIT License and includes the modified hnswlib library and code for constructing and traversing HNSWs with user-implemented scoring functions. DOCK scores for the DUDE-Z data set are available at https://zenodo.org/records/10989077, and the AmpC and D4 DOCK and Glide scores at Lyu et al.[37] and Yang et al.,[12] respectively.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.4c00683.

Figures and tables presenting: hyperparameter impact on HNSW performance, construction speed, and memory usage; empirical HNSW construction times; recall curves when screening 5% of AmpC and D4 libraries; score distributions of the top 10k molecules found by RAD and brute force docking; number of virtual actives and unique scaffolds sampled by RAD, null traversal, and brute force docking; correlation between DUDE-Z virtual active properties, protein properties, and RAD and null traversal performance; RAD performance when using a machine learning scoring function; recall of the top 50k D4 and AmpC molecules when screening 2.5% of the libraries; explained variance for the first 20 principle components of the fingerprint PCA embedding; schematic diagram of active learning and RAD virtual active enrichment as a function of runtime; number of scaffolds and clusters for the D4 and AmpC virtual actives; performance metrics of the null traversal and RAD when screening 10% of the D4 and AmpC libraries (PDF)

Tables presenting: RAD $pAUC_{10}$, AUC, and log AUC for each DUDE-Z protein for all tested hyperparameter combinations (XLSX)

AUTHOR INFORMATION

Corresponding Author

Michael J. Keiser − *Department of Pharmaceutical Chemistry, University of California, San Francisco, San Francisco, California 94158, United States; Institute for Neurodegenerative Diseases, University of California, San Francisco, San Francisco, California 94158, United States; Bakar Computational Health Sciences Institute, University of California, San Francisco, San Francisco, California 94158, United States; Department of Bioengineering and Therapeutic Sciences, University of California, San Francisco, San Francisco, California 94158, United States;* orcid.org/0000-0002-1240-2192; Email: keiser@keiserlab.org

Author

Brendan W. Hall − *Department of Pharmaceutical Chemistry, University of California, San Francisco, San Francisco, California 94158, United States; Program in Biophysics, University of California, San Francisco, San Francisco, California 94158, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c00683

Author Contributions

B.W.H. contributed to conceptualization, methodology, software, validation, investigation, data curation, writing—

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398−7408

original draft, writing—review and editing, and visualization; M.J.K. contributed to conceptualization, methodology, resources, writing—review and editing, supervision, project administration, and funding acquisition.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by CZI grant DAF2018-191905 (DOI 10.37921/550142lkcjzw) from the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation (funder DOI 10.13039/100014989) (M.J.K.). We used ChatGPT (https://chat.openai.com) to suggest edits for clarity and conciseness after we had written the manuscript draft. We thank Noam Teyssier for running DOCK scripts on the DUDE-Z dataset. We thank Laura Shub, Mahdi Ghorbani, and Zachary Gale-Day for their input throughout the project.

■ REFERENCES

(1) Bohacek, R. S.; McMartin, C.; Guida, W. C. The Art and Practice of Structure-Based Drug Design: A Molecular Modeling Perspective. *Med. Res. Rev.* **1996**, *16* (1), 3−50.

(2) Warr, W. A.; Nicklaus, M. C.; Nicolaou, C. A.; Rarey, M. Exploration of Ultralarge Compound Collections for Drug Discovery. *J. Chem. Inf. Model.* **2022**, *62* (9), 2021−2034.

(3) *REAL Database - Enamine*. https://enamine.net/compound-collections/real-compounds/real-database. (Accessed 17 June 2024).

(4) *REAL Space*. https://enamine.net/compound-collections/real-compounds/real-space-navigator. (Accessed 17 June 2024).

(5) Lyu, J.; Irwin, J. J.; Shoichet, B. K. Modeling the Expansion of Virtual Screening Libraries. *Nat. Chem. Biol.* **2023**, *19* (6), 712−718.

(6) 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.; Fackeldey, K.; Hoffmann, M.; Iavniuk, I.; Wagner, G.; Arthanari, H. An Open-Source Drug Discovery Platform Enables Ultra-Large Virtual Screens. *Nature* **2020**, *580* (7805), 663−668.

(7) Rogers, D. M.; Agarwal, R.; Vermaas, J. V.; Smith, M. D.; Rajeshwar, R. T.; Cooper, C.; Sedova, A.; Boehm, S.; Baker, M.; Glaser, J.; Smith, J. C. SARS-CoV2 Billion-Compound Docking. *Sci. Data* **2023**, *10* (1), 173.

(8) Fink, E. A.; Bardine, C.; Gahbauer, S.; Singh, I.; Detomasi, T. C.; White, K.; Gu, S.; Wan, X.; Chen, J.; Ary, B.; Glenn, I.; O'Connell, J.; O'Donnell, H.; Fajtová, P.; Lyu, J.; Vigneron, S.; Young, N. J.; Kondratov, I. S.; Alisoltani, A.; Simons, L. M.; Lorenzo-Redondo, R.; Ozer, E. A.; Hultquist, J. F.; O'Donoghue, A. J.; Moroz, Y. S.; Taunton, J.; Renslo, A. R.; Irwin, J. J.; García-Sastre, A.; Shoichet, B. K.; Craik, C. S. Large Library Docking for Novel SARS-CoV-2 Main Protease Non-Covalent and Covalent Inhibitors. *Protein Sci.* **2023**, *32* (8), No. e4712.

(9) Singh, I.; Li, F.; Fink, E. A.; Chau, I.; Li, A.; Rodriguez-Hernández, A.; Glenn, I.; Zapatero-Belinchón, F. J.; Rodriguez, M. L.; Devkota, K.; Deng, Z.; White, K.; Wan, X.; Tolmachova, N. A.; Moroz, Y. S.; Kaniskan, H. Ü.; Ott, M.; García-Sastre, A.; Jin, J.; Fujimori, D. G.; Irwin, J. J.; Vedadi, M.; Shoichet, B. K. Structure-Based Discovery of Inhibitors of the SARS-CoV-2 Nsp14 N7-Methyltransferase. *J. Med. Chem.* **2023**, *66* (12), 7785−7803.

(10) Ton, A.-T.; Gentile, F.; Hsing, M.; Ban, F.; Cherkasov, A. Rapid Identification of Potential Inhibitors of SARS-CoV-2 Main Protease by Deep Docking of 1.3 Billion Compounds. *Mol. Inform.* **2020**, *39* (8), No. e2000028.

(11) Tingle, B. I.; Tang, K. G.; Castanon, M.; Gutierrez, J. J.; Khurelbaatar, M.; Dandarchuluun, C.; Moroz, Y. S.; Irwin, J. J. ZINC-22−A Free Multi-Billion-Scale Database of Tangible Compounds for Ligand Discovery. *J. Chem. Inf. Model.* **2023**, *63* (4), 1166−1176.

(12) 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* (11), 7106−7119.

(13) Graff, D. E.; Shakhnovich, E. I.; Coley, C. W. Accelerating High-Throughput Virtual Screening through Molecular Pool-Based Active Learning. *Chem. Sci.* **2021**, *12* (22), 7866−7881.

(14) 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 Cent. Sci.* **2020**, *6* (6), 939−949.

(15) 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* (9), 1169−1175.

(16) Sivula, T.; Yetukuri, L.; Kalliokoski, T.; Käsnänen, H.; Poso, A.; Pöhner, I. Machine Learning-Boosted Docking Enables the Efficient Structure-Based Virtual Screening of Giga-Scale Enumerated Chemical Libraries. *J. Chem. Inf. Model.* **2023**, *63* (18), 5773−5783.

(17) Graff, D. E.; Aldeghi, M.; Morrone, J. A.; Jordan, K. E.; Pyzer-Knapp, E. O.; Coley, C. W. Self-Focusing Virtual Screening with Active Design Space Pruning. *J. Chem. Inf. Model.* **2022**, *62* (16), 3854−3862.

(18) Wang, D.; Cui, C.; Ding, X.; Xiong, Z.; Zheng, M.; Luo, X.; Jiang, H.; Chen, K. Improving the Virtual Screening Ability of Target-Specific Scoring Functions Using Deep Learning Methods. *Front. Pharmacol.* **2019**, *10*, 924.

(19) Imrie, F.; Bradley, A. R.; van der Schaar, M.; Deane, C. M. Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. *J. Chem. Inf. Model.* **2018**, *58* (11), 2319−2330.

(20) Wójcikowski, M.; Ballester, P. J.; Siedlecki, P. Performance of Machine-Learning Scoring Functions in Structure-Based Virtual Screening. *Sci. Rep.* **2017**, *7*, 46710.

(21) Ghorbani, M.; Gendelev, L.; Beroza, P.; Keiser, M. J. *Autoregressive Fragment-Based Diffusion for Pocket-Aware Ligand Design*. arXiv, 2023.

(22) Guan, J.; Qian, W. W.; Peng, X.; Su, Y.; Peng, J.; Ma, J. *3D Equivariant Diffusion for Target-Aware Molecule Generation and Affinity Prediction*. arXiv, 2023.

(23) Peng, X.; Luo, S.; Guan, J.; Xie, Q.; Peng, J.; Ma, J. Pocket2Mol: Efficient Molecular Sampling Based on 3D Protein Pockets. In *Proceedings of the 39th International Conference on Machine Learning*, PMLR, 2022, pp. 1764417655.

(24) Sadybekov, A. A.; Sadybekov, A. V.; Liu, Y.; Iliopoulos-Tsoutsouvas, C.; Huang, X.-P.; Pickett, J.; Houser, B.; Patel, N.; Tran, N. K.; Tong, F.; Zvonok, N.; Jain, M. K.; Savych, O.; Radchenko, D. S.; Nikas, S. P.; Petasis, N. A.; Moroz, Y. S.; Roth, B. L.; Makriyannis, A.; Katritch, V. Synthon-Based Ligand Discovery in Virtual Libraries of over 11 Billion Compounds. *Nature* **2022**, *601* (7893), 452−459.

(25) BioSolveIT. *Chemical Space Docking*. https://www.biosolveit.de/application-academy/chemical-space-docking/. (Accessed 1 April 2024).

(26) Gao, W.; Coley, C. W. The Synthesizability of Molecules Proposed by Generative Models. *J. Chem. Inf. Model.* **2020**, *60* (12), 5714−5723.

(27) Verdonk, M. L.; Giangreco, I.; Hall, R. J.; Korb, O.; Mortenson, P. N.; Murray, C. W. Docking Performance of Fragments and Druglike Compounds. *J. Med. Chem.* **2011**, *54* (15), 5422−5431.

(28) Sándor, M.; Kiss, R.; Keserű, G. M. Virtual Fragment Docking by Glide: A Validation Study on 190 Protein−Fragment Complexes. *J. Chem. Inf. Model.* **2010**, *50* (6), 1165−1172.

(29) Malkov, Y. A.; Yashunin, D. A. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42* (4), 824−836.

(30) Chroma: The AI-Native Open-Source Embedding Database; Github.

(31) Weaviate: Weaviate Is an Open-Source Vector Database That Stores Both Objects and Vectors, Allowing for the Combination of Vector Search with Structured Filtering with the Fault Tolerance and Scalability of a Cloud-Native Database; Github.

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398−7408

(32) Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Kuttler, H.; Lewis, M.; Yih, W.-T.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459−9474.
(33) Scheiber, J.; Jenkins, J. L.; Sukuru, S. C. K.; Bender, A.; Mikhailov, D.; Milik, M.; Azzaoui, K.; Whitebread, S.; Hamon, J.; Urban, L.; Glick, M.; Davies, J. W. Mapping Adverse Drug Reactions in Chemical Space. *J. Med. Chem.* **2009**, *52* (9), 3103−3107.
(34) Bender, A.; Scheiber, J.; Glick, M.; Davies, J. W.; Azzaoui, K.; Hamon, J.; Urban, L.; Whitebread, S.; Jenkins, J. L. Analysis of Pharmacology Data and the Prediction of Adverse Drug Reactions and off-Target Effects from Chemical Structure. *ChemMedChem* **2007**, *2* (6), 861−873.
(35) Lounkine, E.; Keiser, M. J.; Whitebread, S.; Mikhailov, D.; Hamon, J.; Jenkins, J. L.; Lavan, P.; Weber, E.; Doak, A. K.; Côté, S.; Shoichet, B. K.; Urban, L. Large-Scale Prediction and Testing of Drug Activity on Side-Effect Targets. *Nature* **2012**, *486* (7403), 361−367.
(36) Scheiber, J.; Chen, B.; Milik, M.; Sukuru, S. C. K.; Bender, A.; Mikhailov, D.; Whitebread, S.; Hamon, J.; Azzaoui, K.; Urban, L.; Glick, M.; Davies, J. W.; Jenkins, J. L. Gaining Insight into off-Target Mediated Effects of Drug Candidates with a Comprehensive Systems Chemical Biology Analysis. *J. Chem. Inf. Model.* **2009**, *49* (2), 308−317.
(37) Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K.; Tolmachev, A. A.; Shoichet, B. K.; Roth, B. L.; Irwin, J. J. Ultra-Large Library Docking for Discovering New Chemotypes. *Nature* **2019**, *566* (7743), 224−229.
(38) 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* (7), 1739−1749.
(39) 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* (7), 1750−1759.
(40) Stein, R. M.; Yang, Y.; Balius, T. E.; O'Meara, M. J.; Lyu, J.; Young, J.; Tang, K.; Shoichet, B. K.; Irwin, J. J. Property-Unmatched Decoys in Docking Benchmarks. *J. Chem. Inf. Model.* **2021**, *61* (2), 699−714.
(41) Coleman, R. G.; Carchia, M.; Sterling, T.; Irwin, J. J.; Shoichet, B. K. Ligand Pose and Orientational Sampling in Molecular Docking. *PLoS One* **2013**, *8* (10), No. e75992.
(42) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38* (6), 983−996.
(43) Hnswlib; Github.
(44) Rdkit: The Official Sources for the RDKit Library; Github.
(45) Peixoto, T. P. *The Graph-Tool Python Library*, Figshare, 2014.
(46) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39* (15), 2887−2893.
(47) Butina, D. Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39* (4), 747−750.
(48) Johnson, M. A.; Maggiora, G. M. *Concepts and Applications of Molecular Similarity*; Wiley, 1990.
(49) Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An Open Source Platform for Ligand Pocket Detection. *BMC Bioinf.* **2009**, *10*, 168.
(50) Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; Kadurin, A.; Johansson, S.; Chen, H.; Nikolenko, S.; Aspuru-Guzik, A.; Zhavoronkov, A. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Front. Pharmacol.* **2020**, *11*, 565644.
(51) Benhenda, M. *ChemGAN Challenge for Drug Discovery: Can AI Reproduce Natural Chemical Diversity?* arXiv, 2017.
(52) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742−754.
(53) Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL Keys for Use in Drug Discovery. *J. Chem. Inf. Comput. Sci.* **2002**, *42* (6), 1273−1280.
(54) Bender, A.; Jenkins, J. L.; Scheiber, J.; Sukuru, S. C. K.; Glick, M.; Davies, J. W. How Similar Are Similarity Searching Methods? A Principal Component Analysis of Molecular Descriptor Space. *J. Chem. Inf. Model.* **2009**, *49* (1), 108−119.
(55) Nettles, J. H.; Jenkins, J. L.; Bender, A.; Deng, Z.; Davies, J. W.; Glick, M. Bridging Chemical and Biological Space: "Target Fishing" Using 2D and 3D Molecular Descriptors. *J. Med. Chem.* **2006**, *49* (23), 6802−6810.
(56) Sun, M.; Xing, J.; Wang, H.; Chen, B.; Zhou, J. MoCL: Data-Driven Molecular Fingerprint via Knowledge-Aware Contrastive Learning from Molecular Graph. *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery and data mining*; ACM, 2021, 35853594.
(57) Fare, C.; Turcani, L.; Pyzer-Knapp, E. O. Powerful, Transferable Representations for Molecules through Intelligent Task Selection in Deep Multitask Networks. *Phys. Chem. Chem. Phys.* **2020**, *22* (23), 13041−13048.
(58) Gale-Day, Z. J.; Shub, L.; Chuang, K. V.; Keiser, M. J. *Proximity Graph Networks: Predicting Ligand Affinity with Message Passing Neural Networks*. ChemRxiv, 2024.
(59) Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the Chemical Beauty of Drugs. *Nat. Chem.* **2012**, *4* (2), 90−98.
(60) Mysinger, M. M.; Shoichet, B. K. Rapid Context-Dependent Ligand Desolvation in Molecular Docking. *J. Chem. Inf. Model.* **2010**, *50* (9), 1561−1573.

https://doi.org/10.1021/acs.jcim.4c00683
J. Chem. Inf. Model. 2024, 64, 7398−7408