Chemical Science

Accepted Manuscript

This article can be cited before page numbers have been issued, to do this please use: M. Pandey, T. Sajed, I. Semenov, R. Zhang, F. Ban, E. Manskaia, M. Ester and A. Cherkasov, *Chem. Sci.*, 2026, DOI: 10.1039/D5SC09599A.

ROYAL SOCIETY OF CHEMISTRY

rsc.li/chemical-science

View Article Online
DOI: 10.1039/D5SC09599A

Deep Docking, Part 2: An Amplified DDU Platform for Ultra-Large Virtual Screening

Mohit Pandey[1,2], Tanvir Sajed[1,2], Ivan Semenov[1,2], Renfei Zhang[3], Fuqiang Ban[1,2], Ekaterina Manskaia[1,2], Martin Ester[3], Artem Cherkasov[1,2]*

[1]*Vancouver Prostate Centre, University of British Columbia, Vancouver, British Columbia, Canada*
[2]*Faculty of Medicine, University of British Columbia, Vancouver, BC, Canada*
[3]*School of Computing Science, Simon Fraser University, Burnaby, BC, Canada*

*Corresponding author: acherkasov@prostatecentre.com

Abstract

The exponential growth of accessible chemical space represents a significant computational challenge for structure-based virtual screening. Hence, active-learning and machine-learning approaches, such as Deep Docking, have been introduced to significantly speed up this process; yet even such methods became computationally prohibitive as docking libraries expanded into and beyond billion-entries levels. To address this challenge, we herein introduce the *Deep Docking Ultra (DDU)* approach, which integrates advanced acquisition functions with a pre-trained molecular large language model (MLLM). We demonstrate that such a combination improves accuracy of docking score emulations, while significantly reducing their computational costs. Through 384 virtual screening experiments involving 12 proteins from all major target classes, we systematically benchmarked DDU performance to identify optimal configurations that reduce required computations by up to 45-fold compared to the original Deep Docking method, and by up to 28,500-fold, compared to brute-force docking, without compromising predictive accuracy. We further demonstrate that DDU is able to screen 10.1 billion ligands against the phosphoglycerate kinase 2 target in just 10 days using 50 Tesla V100 GPUs, and yields an overall docking enrichment factor of 12,000.

View Article Online
DOI: 10.1039/D5SC09599A

The DDU code is available at https://github.com/diamondspark/DDU.

Introduction

Recent expansion of accessible chemical space, along with advancements in protein structure determination and prediction, have transformed early-stage drug discovery and returned structure-based virtual screening (SBVS) methods into the spotlight. Among these approaches, molecular docking remains a cornerstone technique, providing predictions of protein-ligand binding poses and interaction energetics. While docking has led to the discovery of numerous of relevant ligands, its application to modern ultra-large libraries, nowadays routinely exceeding billions of molecules, has become computationally restrictive. This challenge has driven the development of more efficient screening strategies, including fragment-based docking[1,2], machine learning (ML)-assisted docking and various deep learning (DL)-enhanced VS approaches[3–8] among others.

Broadly, strategies addressing navigation of billion-scale chemical space fall into two categories. The first, exemplified by ultra-high-throughput docking (UHTD) and AL-accelerated VS approaches[9–11] operates over pre-enumerated and synthetically accessible vendor libraries. . The second- category methods such as fragment-based generative approaches[12] or structure-based diffusion models[13,14], construct molecules *de novo*, guided by learned representations of bioactivity or target geometry. While such generative methods can theoretically extend beyond the existing libraries, they face the distinct challenge of synthetic accessibility [15]. These two paradigms are therefore complementary. UHTD and AL-based approaches offer immediate access to synthetically tractable hits, while generative approaches extend the boundaries of accessible chemical novelty. As chemical libraries themselves continue to grow toward and beyond the

View Article Online
DOI: 10.1039/D5SC09599A

trillion-molecule scale, efficient navigation of pre-enumerated space remains an essential and practical challenge.

Thus, in 2020 we introduced Deep Docking (DD)[16,17] - an AL-driven approach that trains deep neural networks to predict docking scores from an iteratively sampled docking database. DD demonstrated significant acceleration over exhaustive brute-force docking and led to numerous successful hit discovery campaigns involving billions of molecules, including the identification of inhibitors for SARS-CoV-2 papain-like protease (PLpro)[18], Lin28 protein[19], WDR domain of LRRK2 protein[20], Macrodomain 1 (Mac1) of non-structural protein 3 (NSP3)[21], human Androgen Receptor[22], and A2A adenosine receptor[23] among many others. The use of DD facilitated our winning strategies in two worldwide hit discovery competitions CACHE-1[24] and CACHE-3[25].

Nonetheless, DD also remains relatively resource-intensive and does not fully allow optimal training set selection as chemical libraries continue to expand exponentially. To address these limitations, we introduce Deep Docking Ultra (DDU) - an enhanced AL framework that integrates more informative acquisition strategies with a pre-trained molecular large language model (MLLM) tailored for small molecules. The DDU uses the MLLM as a feature extractor within a deep architecture incorporating multi-head attention and residual layers (Figure 1), enabling accurate identification of high-scoring compounds through binary classification while requiring substantially fewer explicit docking calculations.

We conducted an extensive benchmarking of DDU method across 384 virtual screening (VS) experiments under diverse conditions and identified configurations that achieved up to a 45-fold reduction in docking computations compared to the original DD protocol, using a library of one million compounds (approximating the smallest size on which DD can be accurately trained). We

View Article Online
DOI: 10.1039/D5SC09599A

further validated the scalability of DDU on an ultra-large chemical library, screening over 10.1 billion molecules from the Enamine REAL database within 10 days, while maintaining strong enrichment within the top one percent and achieving approximately a 28,500-fold reduction in docking computations compared to exhaustive docking[26]. These results position DDU as an efficient and practical tool for large-scale SBVS, demonstrating that the integration of language models and optimized sampling strategies can significantly enhance computational drug discovery at reduced costs.

Results

We developed the DDU workflow (outlined in Figure 1), as an active learning framework for large-scale docking that integrates a pretrained molecular large language model, MoLFormer-DR[27], with uncertainty-guided acquisition (BALD)[28] and a fixed binarization threshold of docking scores.

View Article Online
DOI: 10.1039/D5SC09599A

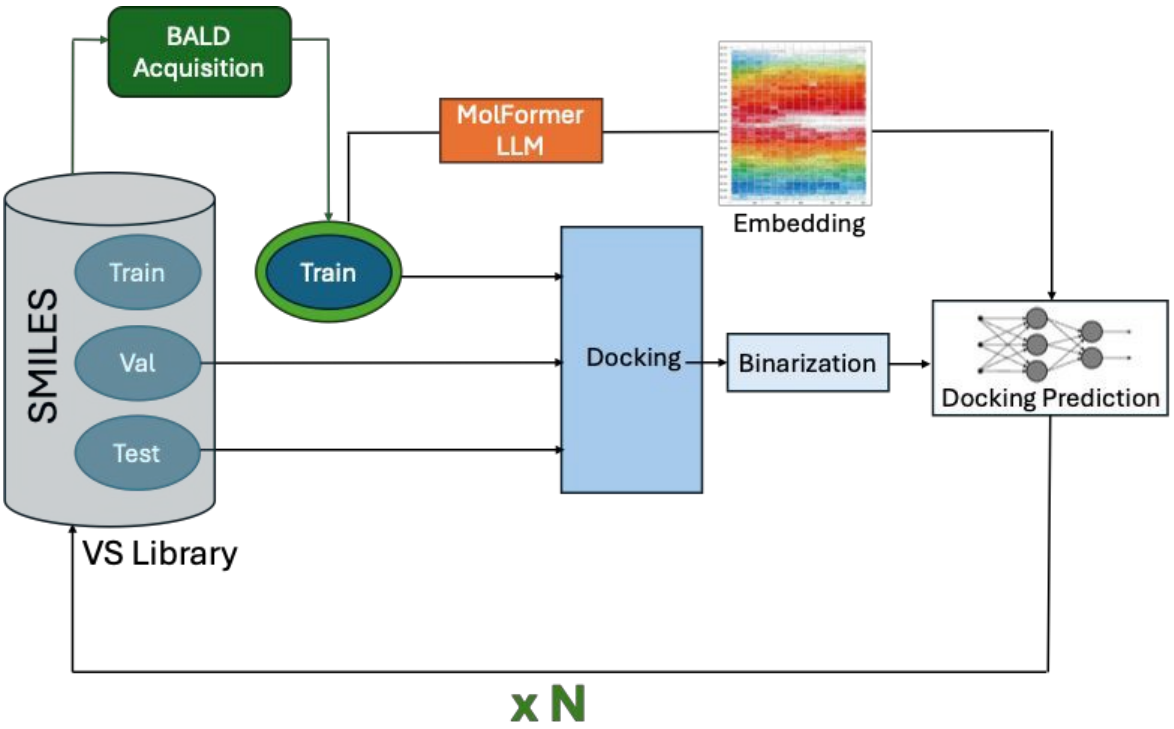


Figure 1. Deep Docking Ultra architecture. *DDU integrates a molecular language model (MoLFormer-DR) with BALD-based active learning to iteratively predict, dock, and retrain on the most informative compounds. In each cycle, compounds are ranked by their predicted probability of belonging to the active class, with the acquisition function (BALD) further prioritizing those with highest model uncertainty; the top-ranked compounds by this acquisition score are selected for docking, binarized into virtual actives and inactives using the fixed score threshold and used to refine the classifier for the next iteration."*

Our original DD method relied on a feedforward neural network trained on Morgan molecular fingerprints and was augmented iteratively with newly docked molecules [16]. While very effective, this approach still required docking of millions of molecules during the training phase and relied on precomputing computationally expensive 1024-bit molecular fingerprints, which incurred significant time and storage costs. In contrast to the traditional DD model, DDU operates directly on SMILES representations and achieves comparable or superior performance while reducing docking computations by up to 45-fold. Following the standard 11-iteration AL training cycle described in our original DD publication[16], DDU requires docking only a total of 21,000 molecules: 20,000 in the first iteration and 100 additional molecules in each of the subsequent 10

View Article Online
DOI: 10.1039/D5SC09599A

iterations. The original DD approach, by comparison, requires docking approximately 11 million molecules per target for efficient training and active learning. This translates to a speedup of more than three orders of magnitude compared to DD[16] (Figure 2).

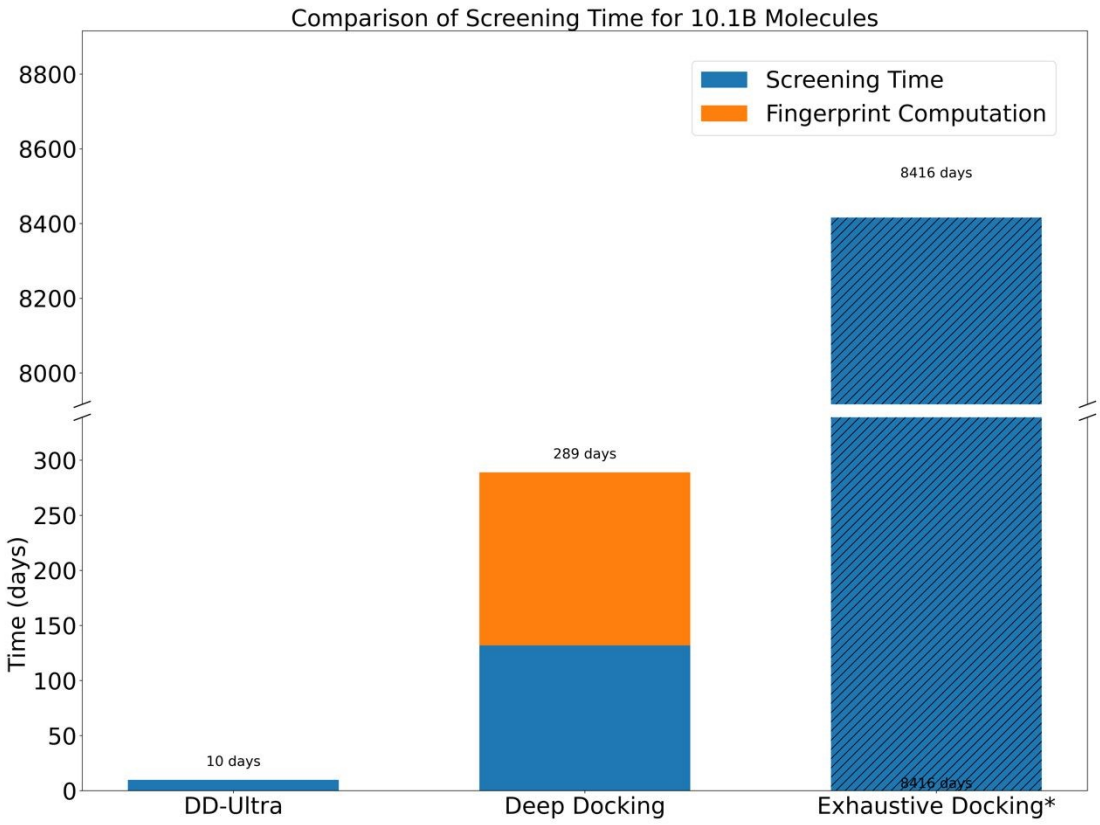


Figure 2 | Time comparison for Deep Docking (DD), DDU, and exhaustive docking on 10.1 billion compound Enamine REAL library*. DD includes a one-time fingerprint computation cost, which DDU bypasses. All three methods assume 50 NVIDIA V100 GPUs.* Reported times for DD and DDU include training, inference, docking, and active learning iterations, with 11 iterations for DD and 5 iterations for DDU. *Time for exhaustive docking and DD are estimated based on projected throughput, not actual runs.*

Using the developed DDU protocol, we further performed comparative benchmarking across 12 proteins representing four major drug-target classes that were used in the original DD publication. This panel included nuclear receptors, represented by androgen receptor (AR), estrogen receptor-alpha (ERα), and peroxisome proliferator-activated receptor gamma (PPARγ); kinases, including

View Article Online
DOI: 10.1039/D5SC09599A

calcium/calmodulin-dependent protein kinase kinase 2 (CAMKK2), cyclin-dependent kinase 6 (CDK6), and vascular endothelial growth factor receptor 2 (VEGFR2); G protein-coupled receptors (GPCRs), such as adenosine A2A receptor (ADORA2A), thromboxane A2 receptor (TBXA2R), angiotensin II receptor type 1 (AT1R); and ion channels, represented by Nav1.7 sodium channel (Nav1.7), Gloeobacter ligand-gated ion channel (GLIC), and gamma-aminobutyric acid type A receptor (GABAA).

The results on those 12 targets demonstrated that while both DD and DDU successfully enriched for high-scoring compounds significantly above the 1st percentile threshold of randomly selected training compounds, DDU exhibited superior performance across multiple metrics. Thus, the docking score distributions (Figure 3) reveal that DDU consistently achieved taller peaks with reduced variance, indicating more effective enrichment for high-affinity compounds, while the t-SNE projections and corresponding entropy calculations (H) demonstrated that DDU identified substantially more diverse chemical space compared to DD across all targets. These results indicate that DDU overcomes the traditional enrichment-diversity trade-off in VS by simultaneously maintaining superior scoring performance and exploring broader chemical diversity. This represents a significant methodological advancement for drug discovery, where scaffold diversity is critical for identifying novel bioactive compounds.

View Article Online
DOI: 10.1039/D5SC09599A

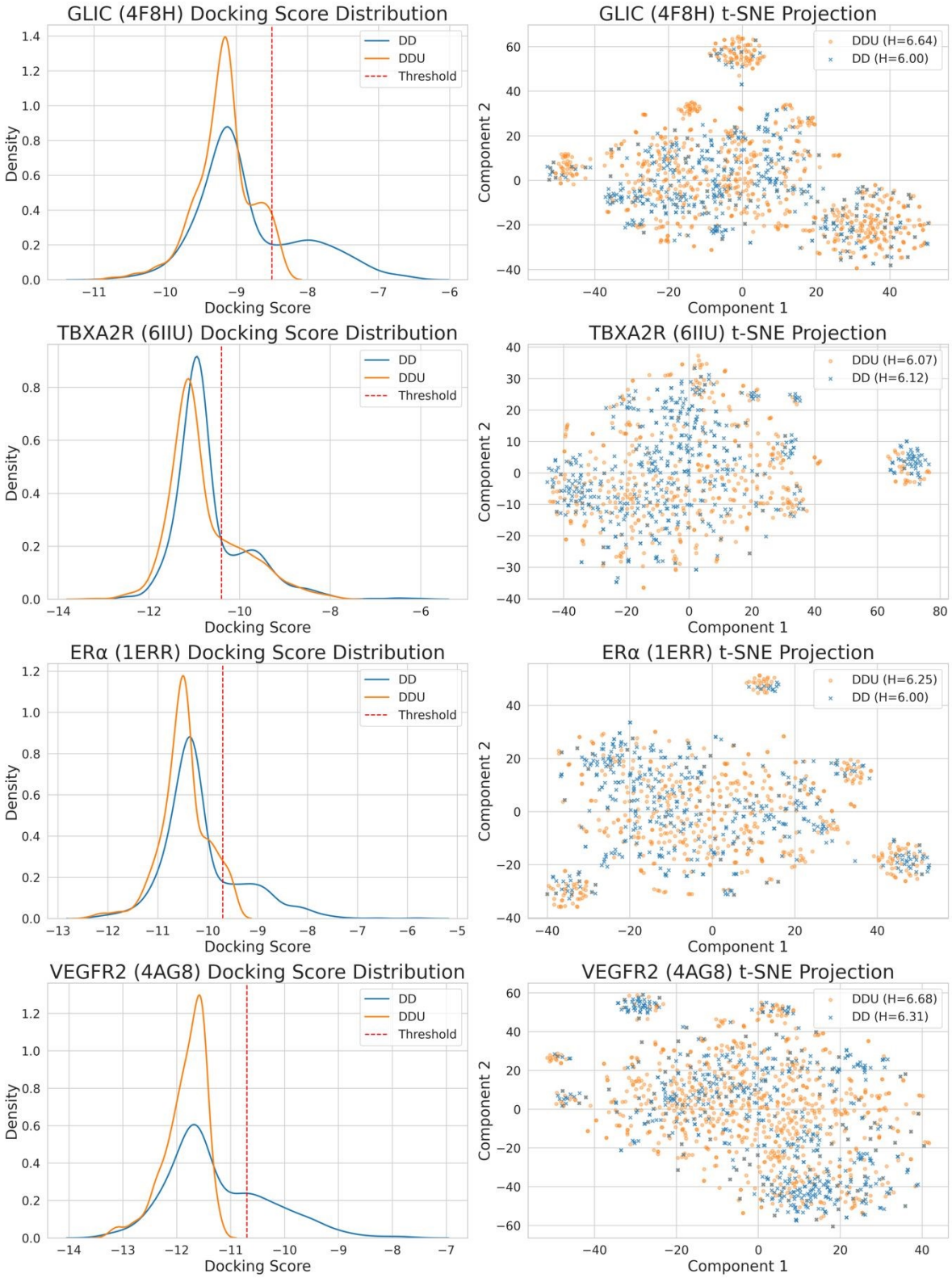


Figure 3.* | *Comparative performance of Deep Docking and DDU: *Docking score distributions (left panels) and t-SNE projections (right panels) are shown for the top 2000 virtual hits identified by Deep Docking (blue) and DDU (orange) for four randomly selected targets from the benchmark*

View Article Online
DOI: 10.1039/D5SC09599A

target set. Red dashed lines indicate the first percentile threshold derived from randomly sampled training compounds. DDU consistently produces more concentrated docking score distributions with higher peak density while maintaining greater chemical diversity, as reflected by higher entropy values (H) in the t-SNE projections. Entropy quantifies the dispersion of compounds in chemical space, with higher values indicating greater diversity. Results for the remaining eight targets are provided in Supplementary Figures S10 and S11.

In addition, DDU eliminated the need to compute and store molecular fingerprints, significantly lowering the memory usage. For instance, generating and storing Morgan fingerprints for a billion-compound library such as ZINC20 requires over 200 GB of disk space and 11 days of processing on a single CPU core. By using a SMILES-based model, DDU bypassed this bottleneck altogether. In practical terms, the peak RAM footprint of the DDU orchestrator is just around 4.5 GB during model training on a 1-million-molecule screen (Supplementary Table S1), with no dependency on pre-stored fingerprint data.

Optimization of DDU settings

The performance improvements of DDU were driven by a carefully selected combination of model architecture and AL strategy, identified through extensive benchmarking on the 12 diverse protein targets. All benchmarking experiments on these targets used a screening library of one million molecules randomly sampled from the ZINC22 database. For each target, the AL pipeline was initialized with 20,000 training, 50,000 validation, and 100,000 test molecules drawn from this pool; the validation and test sets were held fixed throughout all subsequent iterations.

Among the 10 benchmarked DL architectures, which varied in model capacity from simple feedforward neural networks (FFNN) to modified transformer architectures known to perform well in molecular modeling tasks[27,29,30], MoLFormer-DR was finally selected because of its stable performance, minimal preprocessing requirements and effectiveness in large-scale screening

View Article Online
DOI: 10.1039/D5SC09599A

(Figure 4A). Although the residual FFNN and attention-based FFNN alternatives such as mlp6K and mlpTx offer slightly higher F1 scores, they require extensive molecular fingerprint computation, which limits scalability. MoLFormer-DR and MoLFormer-AH[27], while somewhat sensitive to overfitting at larger training sizes, achieve optimal performance with only around 50,000 training examples. It is also noteworthy that beyond 20,000 samples, the performance gains are marginal, indicating that such a modest training set is sufficient and very suitable for data-efficient VS pipelines (Figure 4B).

View Article Online
DOI: 10.1039/D5SC09599A

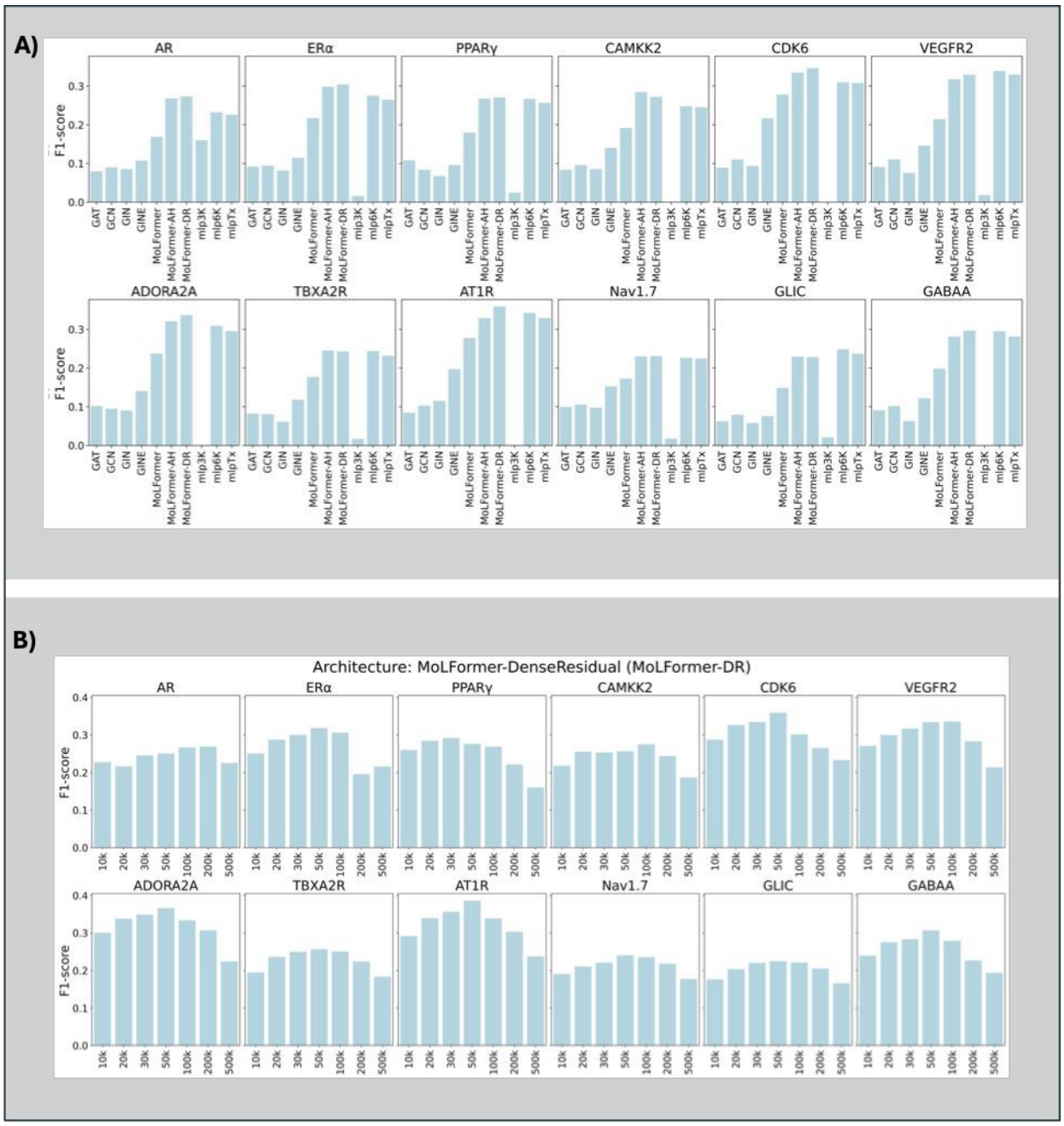
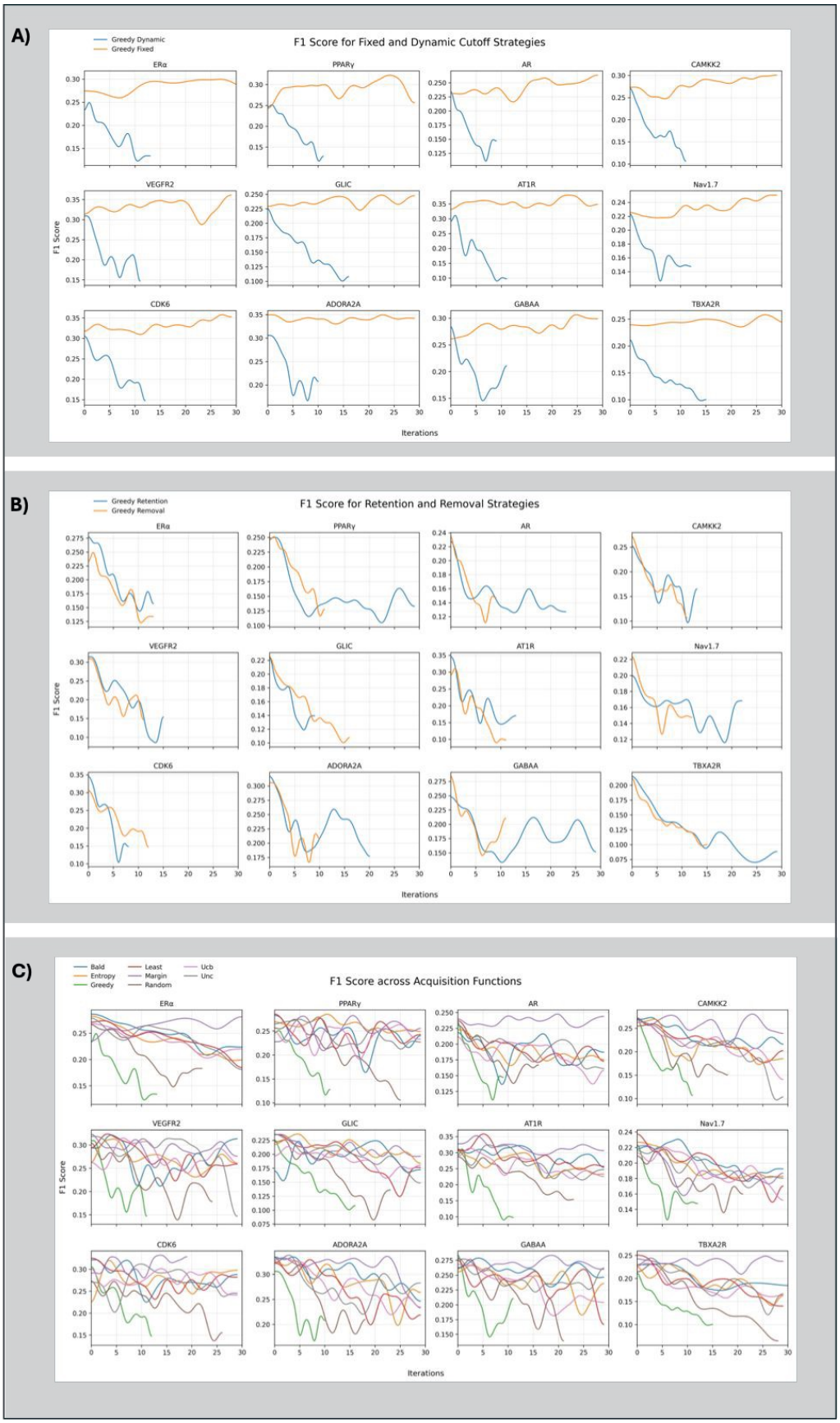


Figure 4. (A) *Comparison of deep learning architectures across twelve targets based on peak F1-scores.* (B) *Effect of training dataset size on F1-scores using the MoLFormer-DR architecture. Peak F1 scores represent the maximum F1-score achieved across all training epochs on the frozen held-out test set of 100,000 molecules, as derived from the per-epoch training curves shown in Supplementary Figures S1–S4.*

View Article Online
DOI: 10.1039/D5SC09599A

We further established that the DDU performance was highly sensitive to the applied docking score thresholds across the studied 12 targets. A fixed threshold of the 1% quantile of the docking scores yielded significantly more stable and accurate classification compared to the dynamic AL threshold (Figure 5A). The dynamic threshold recalculates the docking score cutoff at each AL iteration as the top 1% quantile of the current (growing) training set. While the active-to-inactive ratio within the training set remains approximately constant, the absolute cutoff value becomes progressively more negative. Thus, the performance decline observed under the dynamic threshold conditions is attributable to the concept drift. i.e as virtual screening progresses, fewer molecules qualify as actives, reducing the positive class fraction and making the classification task progressively more difficult. Simultaneously, previously labeled active molecules may no longer satisfy the updated threshold criterion, causing the training objective to shift from one iteration to the next. This progressive misalignment between iteratively redefined labels and the fixed held-out test set is the primary driver of the F1 score deterioration visible in Figure 5A.

View Article Online
DOI: 10.1039/D5SC09599A



View Article Online
DOI: 10.1039/D5SC09599A

Figure 5 | Analysis of active learning strategies and acquisition functions across 12 protein targets. *(A) F1 scores on held-out test sets across 12 targets over AL iterations, comparing fixed vs. dynamic docking-score thresholds. The fixed 1% quantile threshold stabilizes class balance, yielding consistently higher F1 scores, whereas the dynamic threshold tightens progressively over iterations, introducing concept drift: as the score cutoff becomes more stringent, molecules labeled as virtual actives in earlier iterations may be reclassified as virtual inactives under the updated criterion, causing a growing divergence between training labels and the fixed test set threshold that reduces F1 performance. (B) F1 scores across 12 targets comparing database retention vs. database removal strategies. Retention slows the decline in F1 and often maintains slightly higher values than removal, mitigating premature pool depletion and early termination. (C) Comparison of eight acquisition functions across 12 targets. Uncertainty-based methods such as BALD and margin sampling consistently outperform greedy or random approaches, aligning with information-gain and uncertainty-sampling principles in AL. Variation in the number of AL iterations across targets reflects DDU's early-stopping criterion: the AL loop terminates when the overlap between the top-K predicted candidates in consecutive iterations exceeds 90%, avoiding unnecessary docking compute once the surrogate model has converged. Differences in convergence speed reflect target-specific hit rates and binding site complexity.*

Moreover, retaining predicted inactive molecules in the sampling pool of DDU (rather than removing them as was done in the original DD) resulted in further improved model performance and allows avoiding premature pool depletion and early termination of AL runs (Figure 5B and Figure S6). It should be noted, however, that such a benefit was not consistent across all 12 targets and depends on the overall false-negative rate, pool size and cut-off strategies implemented. Therefore, we recommend adopting a database retention approach for libraries of manageable size, whereas for ultra-large-scale VS campaigns, a database removal strategy is more practical.

Finally, we evaluated eight most widely adopted acquisition functions in AL-accelerated VS literature[7,8] to guide AL sampling, indicating that uncertainty-based methods such as BALD[28] and margin sampling consistently outperform greedy or random selection employed in the original DD method (Figure 5C). The BALD sampling was most consistently among the top performers across nearly all targets when used with dynamic thresholding and database removal. Under the recommended configuration (fixed 1% quantile threshold with database retention), BALD acquisition consistently attained the highest or near-highest F1 scores across the majority of the 12

View Article Online
DOI: 10.1039/D5SC09599A

benchmark targets (Supplementary Figures S7–S8) and is the recommended default acquisition function for DDU. For standard screens of up to ~1M compounds, 30 AL iterations is recommended as a practical default (~10 hours on a single GPU, docking ~21,000 molecules in total). For ultra-large campaigns where per-iteration docking cost is substantially higher, 5 iterations is a pragmatic alternative, as demonstrated in the PGK2 10.1B prospective campaign.

A detailed analysis of all benchmarking efforts is presented in Supplementary Sections S2-S3 and Supplementary Figures S5-S9. These findings characterize the *Deep Docking Ultra* approach as an effective, fast, scalable, and accurate alternative to the previously reported *Deep Docking*, with strong potential for real-world application in ultra-large drug screening campaigns. Detailed computational benchmarks substantiating the scalability claims are provided in Supplementary Table S1.

It is important to note that DDU is released as a fully configurable framework rather than a fixed protocol. All methodological choices evaluated in this work, including model architecture, acquisition function, thresholding strategy, database management approach, active learning budget parameters, and compute environment are all exposed as user-defined parameters in a single configuration file and do not require modification of the source code. The recommended default configuration is MoLFormer-DR with BALD acquisition, a fixed one percent quantile threshold, and database retention, based on the benchmarking results presented in this study. Users may adopt this default or adjust individual parameters to match the scale, computational resources, and scientific goals of their screening campaign.

Decoy Benchmarking on Deep Docking Ultra performance

View Article Online
DOI: 10.1039/D5SC09599A

We further evaluated generalization capacity of the DDU and DD surrogate models trained on QuickVina2-derived binary docking score labels. This evaluation was performed using experimentally validated active compounds and property-matched decoys from the Database of Useful Decoys: Enhanced (DUD-E)[31]. Among the 12 targets considered in this study, five had corresponding experimental actives available in DUD-E and were therefore included in this analysis.

Notably, the DUD-E compounds were not explicitly docked. Instead, the trained DDU and DD models were used to rank these molecules based solely on their molecular structures. This exercise aimed to test whether the learned representations transfer to the discrimination of experimentally confirmed bioactive molecules. It therefore provides a more stringent assessment of generalization than the in-distribution docking score prediction evaluated in the benchmarking experiments.

For each of these five targets, the DDU model achieved a higher area under the receiver operating characteristic curve (ROC-AUC) compared to the original DD model (Figure 6A). This result demonstrates that, across the entire ranked list of screened compounds, DDU provides a more accurate and consistent separation between active and inactive molecules. Further comparison of enrichment factors at different cutoffs (EF@K for K = 10, 100, and 1000) revealed complementary strengths between the two models. The DD exhibited higher enrichment at lower K values, suggesting that it is particularly sensitive to a few top-ranking compounds. In contrast, DDU performed better at larger K values, such as K = 1000, which represents more realistic VS conditions (Figure 6B). The training strategy and BALD acquisition policy used in DDU promote chemical diversity and information gain by intentionally sampling molecules that enhance model generalization rather than focusing exclusively on the highest docking scorers. This approach

View Article Online
DOI: 10.1039/D5SC09599A

improves overall ranking performance, as reflected in higher ROC-AUC values, but may slightly reduce early-stage enrichment by trading immediate precision for broader exploration of chemically promising regions of the search space.

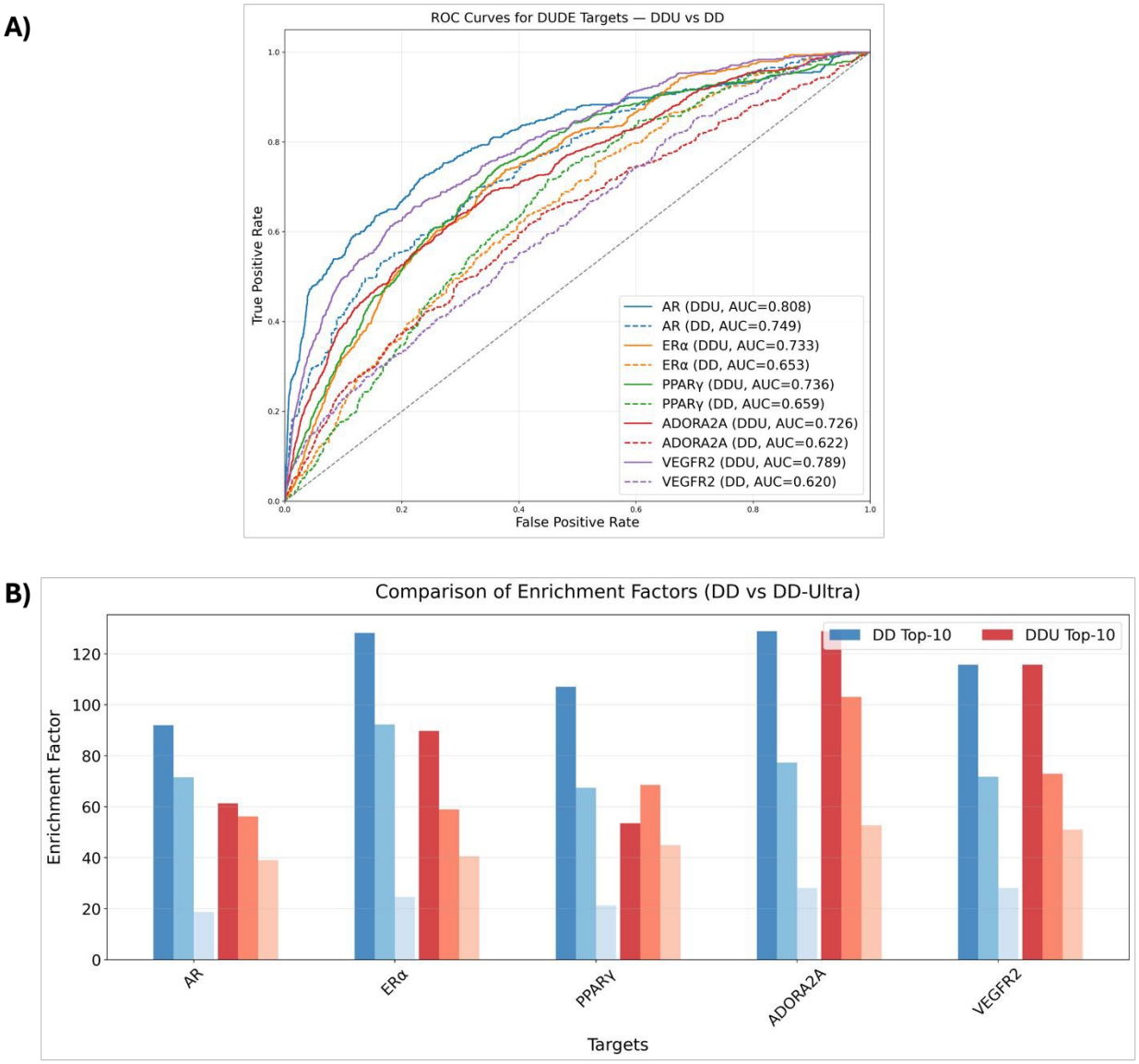


Figure 6 | Performance comparison of Deep Docking (DD) and Deep Docking Ultra (DDU) on experimental actives from the DUD-E dataset. *(A) Receiver operating characteristic (ROC) curves for five pharmaceutically relevant targets show consistently higher AUC values for DDU compared to DD, indicating improved global discrimination between actives and inactives. (B) Enrichment factor (EF@K) comparison for top-10, top-100, and top-1000 ranked compounds demonstrates that while DD achieves higher early enrichment (top-10), DDU excels at broader ranking depths. Lighter shades correspond to higher K values.*

View Article Online
DOI: 10.1039/D5SC09599A

Ultra Large Virtual Screening with Deep Docking Ultra

To evaluate the scalability and practical applicability of DDU for ultra-large screening campaigns, we extended our benchmarking to phosphoglycerate kinase 2 (PGK2) - a sperm-specific isozyme essential for male fertility. This target aligns with CACHE Challenge #7 (which our group is currently participating in) that aims to identify selective PGK2 inhibitors as potential non-hormonal male contraceptives. The Enamine REAL library (accessed October, 2025), comprising approximately 10.1 billion synthetically accessible, drug-like molecules, all compliant with the Lipinski's Rule of Five and Veber criteria (MW ≤ 500, SlogP ≤ 5, HBA ≤ 10, HBD ≤ 5, rotatable bonds ≤ 10, TPSA ≤ 140), was used as the chemical space for screening.

Within a fixed 10-day compute budget on 50 Nvidia Tesla V100 GPUs, DDU was executed for five AL iterations, retraining the model at each cycle using BALD-based uncertainty acquisition to select informative candidates among the top-scoring molecules. On a held-out test set constructed by randomly sampling the Enamine REAL 10.1 billion dataset, the best model achieved AUC-ROC = 0.71 and F1 = 0.31, reflecting strong early recognition performance despite the extreme class imbalance inherent to billion-scale VS. Notably, this AUC-ROC is comparable to the best-performing models in the original DD study (e.g., AR = 0.77, PPARγ = 0.79), while screening a dataset nearly seven times larger (10.1B vs. 1.3B molecules) within the same computational budget.

To further quantify the enrichment by the DDU, we computed virtual Enrichment Factor (EF) of *virtual actives* defined as compounds with docking scores within the top 1 % (lowest energies) for top-K = 10 000, 100 000, and 1 000 000 ranked predictions (Figure 7A). Hence, DDU exhibited

View Article Online
DOI: 10.1039/D5SC09599A

pronounced early enrichment, with EF@10 000 ≈ 1.2×10^4, representing more than a 10 000-fold concentration of virtual actives relative to random expectation. As expected, EF values decreased with increasing K, consistent with progressive inclusion of less informative compounds in the top-ranked pool. The corresponding docking score distributions (Figure 7B) further illustrate the selectivity of the DDU method, compared to a randomly sampled subset of one million molecules. Figure 7B clearly demonstrates that the DDU docking score distribution is markedly left-shifted toward more favorable docking scores and exhibits a sharper, higher-density peak beyond the virtual-active threshold (–8.5 kcal mol^{-1}). Future work will focus on assessing experimental translatability by incorporating the official CACHE Challenge #7 (PGK2) results upon their release in 2026, followed by the corresponding publications.

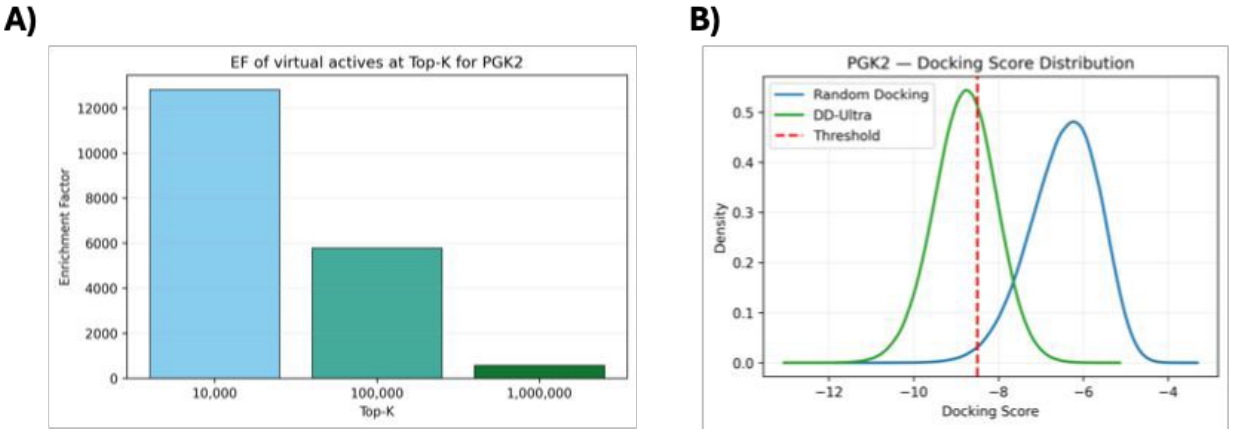


Figure 7 | Performance of DDU in ultra-large virtual screening for PGK2. *(A) Enrichment factors (EF@K) for the top 10 000, 100 000, and 1 000 000 predicted compounds demonstrate DDU's strong early enrichment, with EF@10 000 ≈ 1.2×10^4, corresponding to >10 000-fold enrichment relative to random expectation. (B) Docking score distributions for one million random molecules and the DDU-selected subset show that DDU preferentially concentrates molecules with more favorable docking scores. The vertical red dashed line indicates the virtual-active threshold (–8.5 kcal mol^{-1}).*

Computational Efficiency of Deep Docking Ultra

View Article Online
DOI: 10.1039/D5SC09599A

To substantiate the method's computational efficiency, we conducted a dedicated resource benchmarking on a 1 million molecules library (target 6IIU, 50 Tesla V100-32GB GPUs). A single AL iteration completed in approximately 34 minutes: molecular docking of the 170,000-molecule seed set across 50 GPU sub-jobs (22.8 min; 6.6 mol/s per GPU), MoLFormer-DR model retraining on one GPU (10.0 min; 1,331 samples/s), and pool inference scoring of 830,000 unlabeled molecules across 50 GPUs (1.3 min; 1,759 mol/s per GPU). Peak GPU memory demand was 14.8 GB for docking, 11.8 GB for training, and 1.2 GB for inference, all within the capacity of a single V100-32GB or equivalent ≥16 GB consumer GPU. The orchestrator process required at most 4.5 GB of system RAM, and per-iteration disk output was approximately 1 GB. Complete hardware specifications and throughput figures are provided in Supplementary Table S1. As such, we demonstrate that DDU, as an AL-based accelerator for ultra-large docking screens, performs as intended, and we report its computational benchmarking results here.

Discussion

The benchmarking of DDU performance against the original DD using the same set of 12 targets demonstrated that the active learning (AL) approach optimized in the new DDU implementation is superior to the original DD in terms of required computational resources and efficiency of model refinement. The fundamental upgrade from the original DD to the current DDU is the adoption of chemical structural features in SMILES that can be directly extracted and transformed by AL. It is far more efficient than using the Morgan fingerprint that has to be pre-calculated and stored for the entire screen compound database in the original DD. In addition, the application of high-capacity, generalizable molecular embeddings derived from the MLLM leads to excellent predictive performance of the model trained on 21,000 docked molecules, compared to the approximately 13 million dockings required for optimal performance of the original DD.

View Article Online
DOI: 10.1039/D5SC09599A

On the other hand, by augmenting both high-scoring molecules and uncertain regions of chemical space through BALD-based acquisition, DDU enhances the docking surrogate model's exploration–exploitation balance and improves predictive robustness. In addition, it is worth noting that DDU allows retention of the predicted inactives in the entire unlabelled molecular pool across iterations, which are completely discarded in the original DD model refinement from one iteration to the other. This strategy ensures further improvement of robustness and recall, since the molecules erroneously predicted as inactives by the model of the previous iterations of DDU will be automatically corrected by the refined prediction models in subsequent active learning rounds. Despite these advances, DDU shares limitations common to surrogate model-based VS paradigms, particularly regarding class imbalance, as typically fewer than 1 % of molecules qualify as actives. This imbalance can reduce generalizability and affect model F1 scores. Since DDU prioritizes compounds based on model predictions, improving its classification robustness remains critical. Future research could explore integrating protein structural features into the surrogate model to better capture target-specific interactions and extending the pretraining of MLLM models to broader, multi-source chemical datasets to enhance their generalization. It should also be noted that DDU, as implemented here, is designed specifically as a surrogate accelerator for standard small-molecule docking campaigns against defined binding pockets. Extension to more challenging binding site geometries such as protein-protein interfaces, which typically present flat contact surfaces that challenge conventional docking scoring functions, would require re-evaluation of the docking oracle's suitability in that context and represents an important direction for future work.

Conclusions

View Article Online
DOI: 10.1039/D5SC09599A

To summarize this work, by integrating powerful MLLMs with principled acquisition strategies, the developed *Deep Docking Ultra* (DDU) platform enables efficient exploration of ultra-large chemical libraries that were previously inaccessible or prohibitively expensive to screen. Its robust performance across hundreds of experiments and diverse protein targets underscores both its scalability and practical relevance to real-world drug discovery workflows. As chemical space continues to expand, DDU will increasingly be a scalable and resource-efficient paradigm for accelerating hit identification with unprecedented speed and precision.

Methods

Large Language Models (LLMs)

MoLFormer: The Molecular Language Transformer (MoLFormer) is a transformer encoder model pre-trained on over a billion SMILES strings from the PubChem and ZINC datasets[27]. It aims to learn general-purpose molecular representation for downstream tasks such as docking score classification. To do this, the model leverages linear attention mechanisms and rotary positional embeddings to enhance the representation of molecular structures. During pre-training, a fixed proportion of SMILES tokens is randomly masked, and MoLFormer predicts their identities to learn contextualized embeddings. Attention analyses show that MoLFormer obtains valuable spatial and molecular structural information, such as covalent bonds and interatomic distances, purely from SMILES sequences, despite their lack of explicit topological information. Notably, MoLFormer-XL, the largest variant, outperforms or competes with state-of-the-art graph neural networks (GNNs) and other baselines across classification and regression tasks from the MoleculeNet benchmark, including quantum-chemical property predictions.

View Article Online
DOI: 10.1039/D5SC09599A

We implemented three MoLFormer variants in PyTorch, each with progressively increasing architectural complexity, aimed at improving the performance of the docking score classifier by building upon the pretrained model.

Docking MoLFormer: The Docking MoLFormer architecture modifies the original pre-trained model by adding a shallow classification head. It first aggregates the token-wise hidden states from the last transformer layer using mean pooling. The resulting pooled representation is then passed through a fully connected layer, reducing its dimensionality to 64, followed by a ReLU activation function. A dropout layer (p=0.1) is applied before the final fully connected layer, which maps the input to two output nodes for binary classification.

MoLFormer-AH: The MoLFormer-AH architecture extends the Docking MoLFormer by replacing mean pooling with attention-based pooling, adding a highway network for adaptive feature transformation, and adding layer normalization for stable training. First, the model produces a pooled representation by computing a softmax-normalized attention score across token embeddings and applying a weighted sum over the hidden states. Then the pooled representation undergoes an additional fully connected layer that projects it into a higher-dimensional space (three times the hidden size), followed by a GeLU activation, and layer normalization. This is followed by a highway network that applies a gated sigmoid mechanism to selectively combine the input with its transformed representation through a GeLU activation function. Finally, the refined feature representation passes through a second layer that maps it back to the hidden dimension, before being converted to 2 output nodes through a third layer for binary classification.

MoLFormer-DR: The MoLFormer-DR architecture extends the MoLFormer-AH by adding dense connections, multi-head attention, and residual blocks to refine feature extraction and transformation. Like MoLFormer-AH, it starts by aggregating the input features using attention

View Article Online
DOI: 10.1039/D5SC09599A

pooling. However, instead of directly passing the pooled representation through a fully connected layer, MoLFormer-DR introduces DenseNet-like dense connections. The pooled representation is iteratively combined with the output of a sequence of three fully connected layers (each mapping to the hidden dimension). The concatenated representation is then projected back to the original hidden dimension using another fully connected layer. Subsequently, the output goes through a multi-headed attention mechanism with eight attention heads to generate a contextualized representation. The output of the attention module is then processed through two residual blocks. Each residual block consists of a fully connected layer that projects the feature representation to four times the hidden size, layer normalization, a Parametric Rectified Linear Unit (PReLU) activation function, dropout layer (p=0.1), second fully connected layer that projects the representation back to the original dimension, and another layer normalization followed by PReLU activation. After passing through the residual blocks, the input goes through a fully connected layer that expands it to four times the hidden size, followed by ReLU activation and layer normalization, and a second fully connected layer that projects it back to the hidden dimension, followed by ReLU activation and layer normalization.

Hyperparameter Tuning

The hyperparameters and training configurations for FFNNs are detailed in the table below (details pertaining to architecture are included in the description).

Number of training epochs	40
Optimizer	Adam
Learning rate	0.001
Scheduler	Scheduler with step size 4 and gamma 0.1
Weight decay	0.01

View Article Online
DOI: 10.1039/D5SC09599A

Patience	5
Loss Function	Cross Entropy
Batch Size	2048

Table 1. **|** *Hyperparameters used in training the FFNN-based (mlp3K, mlp6K, mlpTx) and MoLFormer-based (MoLFormer, MoLFormer-AH, MoLFormer-DR) models.*

Number of training epochs	10
Optimizer	Adam
Learning rate	0.001
Scheduler	Cosine Scheduler
Step Size	40
Gamma	0.1
Weight decay	0.01
Loss Function	Focal Loss
Batch Size	2048

Table 2. **|** *Hyperparameters used in training the GNN-based (GCN, GIN, GINE, GAT) models.*

Docking Protocol

For all 12 targets, docking scores used as the oracle were generated using QuickVina2-GPU[32]. QuickVina2 is an established, GPU-accelerated docking engine that has been independently validated across diverse target classes and shown competitive enrichment performance at substantially higher throughput than conventional tools. In the AL-based VS context, the docking

View Article Online
DOI: 10.1039/D5SC09599A

oracle defines the training label quality and therefore the performance ceiling of the surrogate model. The benchmarking results in Figures 3–5 measure how faithfully DDU recovers the top-scoring compounds as ranked by QuickVina2; experimental generalization is assessed separately via the DUD-E analysis (Figure 6).

In order to dock the 12 targets, protein structures were prepared in PDBQT format, and ligands were docked into a predefined binding site defined by a rectangular search box. For each target, the docking search space was specified by the box center coordinates (x,y,z) and box dimensions ($size_x$, $size_y$, $size_z$), and the corresponding receptor PDBQT file (Table 3). Ligands were prepared from SMILES by generating 3D conformers using the RDKit ETKDGv3[33] procedure, followed by explicit hydrogen addition and conversion to PDBQT format prior to docking. The QuickVina2-GPU configuration used a fixed OpenCL binary[34] and a fixed thread setting (8000) across targets; target specificity is entirely captured by the receptor PDBQT and the box center/size parameters. Docking scores were parsed from the Vina output and used as the ground-truth values for downstream thresholding, labeling, and benchmarking.

Target	PDB ID	center_x	center_y	center_z	size_x	size_y	size_z
AR	1t7r	-0.252	30.301	38.95	22.456	25.328	18.771
ERα	1err	67.276	33.964	76.012	21.081	26.879	28.435
PPARγ	1nyx	16.562	63.569	15.1	25.14	21.336	28.261
CAMKK2	2zv2	1.076	-7.19	-25.943	20.517	26.708	18.8
CDK6	5l2s	22.198	38.486	-8.657	24.205	29.091	29.61
VEGFR2	4ag8	19.892	25.629	38.443	29.78	23.803	23.32
ADORA2A	5mzj	-38.176	5.319	22.425	20.713	33.058	21.556
TBXA2R	6iiu	25.175	164.914	148.502	26.289	26.333	24.027
AT1R	4yay	-16.726	9.992	41.794	24.057	27.611	24.281
Nav1.7	5ek0	-85.815	-12.83	-14.228	27.526	28.223	25.905
GLIC	4f8h	16.815	-15.092	25.306	20.866	21.622	22.089
GABAA	6d6t	119.613	169.091	154.264	24.698	22.391	21.573

Table 3. | *Bounding boxes for the 12 targets benchmarked in the DDU study.*

View Article Online
DOI: 10.1039/D5SC09599A

DDU workflow

At AL iteration 1, an initial training dataset of size 20,000, a validation set of size 50,000, and a test set of size 100,000 molecules are randomly sampled from a molecule pool derived from the Enamine REAL database[26]. The validation and test datasets remain fixed throughout the AL pipeline. Each molecule in these datasets is docked against the target of interest using QuickVina2 (the docking oracle), which provides ground-truth docking scores used to generate binary activity labels. The docking scores across all three datasets are converted into binary classification labels using the lowest 1% quantile score from the initial training dataset as the fixed cutoff. As a result, 1% of the training dataset is initially classified as hits (virtual active compounds), while the remaining 99% are considered non-hits (virtual inactive compounds). The MoLFormer-DR model is then trained on molecular embeddings of SMILES sequences extracted from a pre-trained MoLFormer model, alongside their corresponding binary labels. The model learns to predict binary activity labels based on the SMILES representations of input molecules within the training dataset.

From the second iteration onward, the training dataset is iteratively expanded by adding the top 100 molecules according to the BALD acquisition strategy. Additionally, an uncertainty metric is computed across the entire unlabeled pool; inactive molecules with low acquisition scores are discarded to accelerate the process, while the 100 molecules with top scores are sent to the oracle for docking. Finally, after a pre-decided user-defined number of iterations, the molecules predicted as virtual actives by the model are returned to the user. Although DDU operates as a binary classifier, compounds in the unlabeled pool can be meaningfully ranked by their predicted probability of belonging to the active class, as output by the softmax layer. The acquisition function (BALD in the recommended configuration) uses this probability alongside a model uncertainty estimate derived from Monte Carlo dropout to assign an acquisition score to each unlabeled

View Article Online
DOI: 10.1039/D5SC09599A

compound. The 100 compounds with the highest acquisition scores are selected for docking at each iteration, providing a principled internal ranking within the binary classification framework. It is noteworthy that the 100,000-molecule held-out test set used throughout the benchmarking experiments in this work is an optional component of the DDU pipeline, controlled by the `test_rand_samples` parameter in the configuration file. For prospective screening campaigns where maximizing library coverage is the priority, this parameter can be set to `false`, allowing all sampled molecules to contribute to the AL pool. The test set is recommended when quantitative model performance estimates are required, such as in benchmarking studies or when reporting screening results for publication.

Computational Implementation

All DDU experiments were executed on a high-performance computing cluster with NVIDIA Tesla V100 GPU nodes. Docking and pool inference are parallelised across up to 50 GPU sub-jobs (one GPU per job), with each sub-job allocated a single GPU and 50 GB of host RAM. Model training uses a single GPU. The orchestrating process runs on a CPU node and requires ≤8 CPU cores; the peak measured RAM footprint was 4.5 GB for a 1-million-molecule screen. The minimum viable GPU specification for all three phases (docking, training, inference) is ≥16 GB of VRAM. Docking and inference sub-jobs scale linearly with GPU count: reducing from 50 to 1 GPU increases only the wall-clock time of those phases, with no effect on accuracy or AL convergence. The source code, SLURM submission scripts, and benchmark configuration files are available at https://github.com/diamondspark/DDU.

Evaluation Metrics

View Article Online
DOI: 10.1039/D5SC09599A

The evaluation metrics, including precision, recall, F1 metric, AUC score, and top-N enrichment are computed on a fixed test set of 100,000 randomly sampled molecules from the screening library, as described in the Evaluation Metrics section of the Methods. This test set is held out from the beginning of training in iteration 1 and is never used for model selection or hyperparameter tuning across any of the active learning iterations. The validation set, consisting of 50,000 molecules randomly sampled from the same screening library at the start of iteration 1, is used exclusively for early stopping.

Precision is computed as

$$Precision = \frac{TP}{TP + FP}$$

where TP (True Positives) represents the number of molecules the model correctly classified as hits, and FP (False Positives) represent the number of non-hit molecules that the model misclassified as hits.

Recall is computed as $Recall = \frac{TP}{TP + FN}$

where FN (False Negatives) denotes the number of molecules that are actual hits but are misclassified as non-hits.

F1 metric, the harmonic mean of precision and recall, is computed as

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The top N enrichment score is computed as

$$Enrichment_{Top\ N} = \frac{TP_{\ Top\ N}}{TP_{\ Random\ N}}$$

View Article Online
DOI: 10.1039/D5SC09599A

where $TP_{Top\,N}$ is the number of TP within the top N molecules ranked by the model, and $TP_{Random\,N}$ is the number of TP within a subset of randomly chosen N molecules.

Acquisition Functions

The following acquisition functions were tested in this study:

Random Acquisition

$$Random(x) \sim U(0,1)$$

The Random strategy selects the next point $\boldsymbol{x}$ for evaluation by sampling uniformly at random from the interval [0,1], without incorporating any information from the model's predicted labels or uncertainties.

Greedy Acquisition

$$Greedy(x) = \hat{u}(x)$$

The Greedy strategy emphasizes exploitation by selecting the point with the highest predicted value û(x) from the surrogate model, where û(x) is the predicted probability of the positive class. If the number of selected points from the positive class is fewer than the required acquisition size, additional samples are drawn with replacement until the desired size is reached.

Upper confidence bound (UCB)

$$UCB(x) = \hat{u}(x) + \beta\hat{\sigma}(x)$$

The UCB strategy balances exploration and exploitation by accounting for both the predicted class label $\hat{u}(\boldsymbol{x})$ and the predicted uncertainty $\hat{\boldsymbol{\sigma}}(\boldsymbol{x})$ scaled by a hyperparameter β (here β = 2).

Uncertainty (UNC)

$$UNC(x) = \hat{\sigma}(x)$$

View Article Online
DOI: 10.1039/D5SC09599A

The UNC acquisition metric favors exploration by selecting the point with the highest predicted uncertainty, ensuring prediction accuracy of data points from both classes.

Bayesian Active Learning by Disagreement (BALD)

BALD selects points where the predictive uncertainty is high, but this uncertainty comes from disagreement among different plausible models (parameterized by $\boldsymbol{\omega}$). In other words, data points with a high BALD score are those where the individual models, sampled from the posterior distribution, remain confident in their respective predictions, but their overall model prediction shows large uncertainty. However, it often uses Monte Carlo dropout for sampling models from the Bayesian posterior distribution, which could be computationally expensive.

$$BALD(x) = H(y|x, D_{train}) - E_{p(\omega|D_{train})}[H(y|x, \omega, D_{train})]$$

where $H(y|x, D_{train})$ represents the entropy of the average model prediction given input x and training data D_{train}, and the expectation term $E_{p(\omega|D_{train})}[H(y|x, \omega, D_{train})]$ accounts for the average of the entropies of predictions made by individual sampled models drawn from posterior distribution of model parameters $p(\omega|D_{train})$.

Margin Sampling

$$Margin(x) = p(y_1|x) - p(y_2|x)$$

where $p(y_1|x)$ is the predicted probability of the most likely class for input x, and $p(y_2|x)$ is the predicted probability of the second most likely class for the same input. Margin sampling chooses the input $\boldsymbol{x}$ where the difference (margin) between the top two predicted class probabilities is smallest, as a measure of uncertainty.

Entropy Sampling

View Article Online
DOI: 10.1039/D5SC09599A

$$Entropy(x) = H[p(y|x)] = -\sum_{c=1}^{n} p(y = c|x) log(p(y = c|x))$$

Data points with large entropy occur when the model's prediction probabilities are spread across multiple classes, signaling high uncertainty about the correct label.

Least Confidence Sampling

$$Least\ Confidence(x) = 1 - argmax_c\, p(y = c|x)$$

Least Confidence Sampling picks the input for which the model is most uncertain, meaning the confidence in its prediction is low.

Overall, BALD, margin, entropy and least confidence sampling strategies focus on individual point metrics without ensuring diversity of selected points across the feature space.

Code and Data Availability

The code and the one million Zinc22 subset with corresponding Vina dock scores for the five considered targets for AL benchmarking are provided in project's repository, available at https://github.com/diamondspark/DDU. All user-configurable parameters, including model architecture, acquisition function, thresholding strategy, database management, and compute settings, are documented in config/params.yml in the repository.

Author contributions

M.P. designed and implemented the deep neural network architectures, conducted benchmarking to identify optimal configurations. M.P., T.S., and I.S. carried out data analysis and interpretation. T.S. contributed to the implementation of the computational pipeline and assisted with large scale virtual screening experiments. I.S. and R.Z. analyzed the results and provided technical support. F.B. contributed to manuscript writing. E.M. assisted with data preprocessing and library

View Article Online
DOI: 10.1039/D5SC09599A

management. M.E. provided expertise in deep learning methodologies and active learning strategies and critically revised the manuscript. A.C. supervised the project, provided strategic direction, secured funding, and critically revised the manuscript. All authors contributed to manuscript preparation and review.

Conflict of Interest

A.C. is an officer and shareholder of Artemis Therapeutics. All other authors declare no competing interests.

Acknowledgements

This research was enabled by funds provided by a Canadian Institutes of Health Research (CIHR) doctoral award (FRN: FBD-187593) to MP. AC (RGPIN-2024-04153) and ME would like to acknowledge the support by their respective NSERC Discovery grants. AC would additionally like to acknowledge Natural Sciences and Engineering Research Council of Canada [NSERC, RGPIN-2024-04153]) and the Canada Foundation for Innovation (CFI) in collaboration with the British Columbia Knowledge Development Fund (BCKDF, 36194), and the Canada Research Chairs Program (CRC-2020-00007).

References

1. Liao, J. M., Wang, Y. T. & Lin, C. L. S. A fragment-based docking simulation for investigating peptide–protein bindings. *Physical Chemistry Chemical Physics* **19**, 10436–10442 (2017).
2. Sadybekov, A. A. *et al.* Synthon-based ligand discovery in virtual libraries of over 11 billion compounds. *Nature* **601**, 452–459 (2022).
3. Sivula, T. *et al.* Machine Learning-Boosted Docking Enables the Efficient Structure-Based Virtual Screening of Giga-Scale Enumerated Chemical Libraries. *J. Chem. Inf. Model.* **63**, 5773–5783 (2023).
4. Marin, E. *et al.* Regression-Based Active Learning for Accessible Acceleration of Ultra-Large Library Docking. *J. Chem. Inf. Model.* **64**, 2612–2623 (2024).
5. Yang, Y. *et al.* Efficient Exploration of Chemical Space with Docking and Deep Learning. *J. Chem. Theory Comput.* **17**, 7106–7119 (2021).
6. Mehta, S. *et al.* MEMES: Machine learning framework for Enhanced MolEcular Screening. *Chem. Sci.* **12**, 11710 (2021).

View Article Online
DOI: 10.1039/D5SC09599A

7. Kim, J., Nam, J. & Ryu, S. Understanding active learning of molecular docking and its applications. https://arxiv.org/abs/2406.12919v1 (2024).
8. Graff, D. E., Shakhnovich, E. I. & Coley, C. W. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chem. Sci.* **12**, 7866 (2021).
9. Cherkasov, A., Ban, F., Li, Y., Fallahi, M. & Hammond, G. L. Progressive Docking: A Hybrid QSAR/Docking Approach for Accelerating In Silico High Throughput Screening. https://doi.org/10.1021/JM060961 (2006) doi:10.1021/JM060961.
10. Ahmed, L. *et al.* Efficient iterative virtual screening with Apache Spark and conformal prediction. *J. Cheminform.* **10**, 1–8 (2018).
11. Svensson, F., Norinder, U. & Bender, A. Improving Screening Efficiency through Iterative Screening Using Docking and Conformal Prediction. *J. Chem. Inf. Model.* **57**, 439–444 (2017).
12. Imrie, F., Hadfield, T. E., Bradley, A. R. & Deane, C. M. Deep generative design with 3D pharmacophoric constraints. *Chem. Sci.* **12**, 14577 (2021).
13. Schneuing, A. *et al.* Structure-based drug design with equivariant diffusion models. *Nature Computational Science 2024 4:12* **4**, 899–909 (2024).
14. Guan, J. *et al.* 3D Equivariant Diffusion for Target-Aware Molecule Generation and Affinity Prediction. *11th International Conference on Learning Representations, ICLR 2023* https://arxiv.org/pdf/2303.03543 (2023).
15. Gao, W. & Coley, C. W. The Synthesizability of Molecules Proposed by Generative Models. *J. Chem. Inf. Model.* **60**, 5714–5723 (2020).
16. Gentile, F. *et al.* Deep Docking: A Deep Learning Platform for Augmentation of Structure Based Drug Discovery. *ACS Cent. Sci.* **6**, 939–949 (2020).
17. Gentile, F. *et al.* Artificial intelligence–enabled virtual screening of ultra-large chemical libraries with deep docking. *Nature Protocols 2022 17:3* **17**, 672–697 (2022).
18. Garland, O. *et al.* Large-Scale Virtual Screening for the Discovery of SARS-CoV-2 Papain-like Protease (PLpro) Non-covalent Inhibitors. *J. Chem. Inf. Model.* **63**, 2158–2169 (2023).
19. Radaeva, M. *et al.* Discovery of Novel Lin28 Inhibitors to Suppress Cancer Cell Stemness. *Cancers (Basel).* **14**, (2022).
20. Gusev, F. *et al.* Active Learning-Guided Hit Optimization for the Leucine-Rich Repeat Kinase 2 WDR Domain Based on In Silico Ligand-Binding Affinities. *J. Chem. Inf. Model.* **65**, 5706–5717 (2025).
21. Herasymenko, O. *et al.* CACHE Challenge #3: Targeting the Nsp3 Macrodomain of SARS-CoV-2. *Mykola V. Protopopov* **17**, 40 (2025).
22. Radaeva, M. *et al.* Novel Inhibitors of androgen receptor's DNA binding domain identified using an ultra-large virtual screening. *Mol. Inform.* **42**, 2300026 (2023).
23. Tang, M., Wen, C., Lin, J., Chen, H. & Ran, T. Discovery of novel A2AR antagonists through deep learning-based virtual screening. *Artificial Intelligence in the Life Sciences* **3**, 100058 (2023).
24. Li, F. *et al.* CACHE Challenge #1: Targeting the WDR Domain of LRRK2, A Parkinson's Disease Associated Protein. *J. Chem. Inf. Model.* **64**, 8521–8536 (2024).
25. Ban, F. *et al.* Structure-based discovery of inhibitors of Mac1 domain of nonstructural protein-3 of SARS-CoV-2 by machine learning-augmented screening of chemical space. *bioRxiv* 2025.09.05.674529 (2025) doi:10.1101/2025.09.05.674529.

View Article Online
DOI: 10.1039/D5SC09599A

26. Shivanyuk, A. N. *et al.* Enamine real database: Making chemical diversity real. *Chemistry today* **25**, 58–59 (2007).
27. Ross, J. *et al.* Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence 2022 4:12* **4**, 1256–1264 (2022).
28. Houlsby, N., Huszár, F., Ghahramani, Z. & Lengyel, M. Bayesian Active Learning for Classification and Preference Learning. https://arxiv.org/pdf/1112.5745 (2011).
29. Noutahi, E., Gabellini, C., Craig, M., Lim, J. S. C. & Tossou, P. Gotta be SAFE: A New Framework for Molecular Design. *Digital Discovery* **3**, 796–804 (2023).
30. Honda, S., Shi, S. & Ueda, H. R. SMILES Transformer: Pre-trained Molecular Fingerprint for Low Data Drug Discovery. Preprint at https://arxiv.org/abs/1911.04738 (2019).
31. Mysinger, M. M., Carchia, M., Irwin, John. J. & Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem* **55**, 16 (2012).
32. Alhossary, A., Handoko, S. D., Mu, Y. & Kwoh, C. K. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics* **31**, 2214–2216 (2015).
33. Riniker, S. & Landrum, G. A. Better Informed Distance Geometry: Using What We Know To Improve Conformation Generation. *J. Chem. Inf. Model.* **55**, 2562–2574 (2015).
34. E., StoneJ., GoharaD. & ShiGuochun. OpenCL. *Comput. Sci. Eng.* https://doi.org/10.5555/2220077.2220227 (2010) doi:10.5555/2220077.2220227.

View Article Online
DOI: 10.1039/D5SC09599A

All the data produced in this work will be made available on the public repository https://github.com/diamondspark/DDU.