CHEMVERSE: BENCHMARKING ACTIVE LEARNING FOR VIRTUAL SCREENING OF MILLION-SCALE LIBRARIES ACROSS DOCKING ORACLES AND BIOLOGICAL TARGETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Virtual screening is widely used in early-stage drug discovery to prioritize promising molecules before costly experimental testing. Molecular docking enables this process at large scale, but exhaustively screening modern chemical libraries can become computationally prohibitive. Iterative, model-based selection offers an alternative by evaluating a limited subset with a docking oracle, training a surrogate model on the resulting scores, and using rapid inference to prioritize a much larger chemical space. However, it remains unclear which aspects of this setup most strongly determine the performance of adaptive virtual screening. We introduce CHEMVERSE, a benchmark comprising 389 million docking scores across seven protein targets and three chemical libraries ranging from 2×10^6 molecules to 10×10^6 molecules. Across four distinct docking oracles, seven biological targets, and three chemical libraries, complemented by publicly available DOCK3 screens, we systematically compare six molecular representations, three uncertainty-aware surrogate models, and four acquisition strategies. We find that the docking oracle strongly determines the learnability of its score landscape, producing substantially greater variation in optimization performance than molecular representation and uncertainty quantification. Three-dimensional representations perform consistently well across oracles, while uncertainty quantification has comparatively little effect. CHEMVERSE provides a systematic framework for understanding which components of ML-guided virtual screening generalize across oracles and which must be reconsidered as the screening environment changes.

1 INTRODUCTION

Virtual screening (VS) is a central component of early-stage drug discovery, enabling large libraries of candidate molecules to be evaluated computationally and prioritized for experimental testing (Lyu et al., 2019; Gorgulla et al., 2020; Sadybekov et al., 2021; Michino et al., 2023). In structure-based VS, molecular docking predicts candidate binding poses and assigns each molecule a score reflecting its predicted compatibility with a target binding site. These scores provide an inexpensive computational proxy for ranking molecules before substantially more costly experimental evaluation.

Molecular docking has been applied to libraries containing billions of molecules, but the rapid growth of make-on-demand chemical space toward the trillion-molecule scale makes exhaustive screening increasingly impractical (Hoffmann & Gastreich, 2019; Warr et al., 2022; Gorgulla et al., 2022; Cecchini et al., 2026). Larger and more diverse libraries are nevertheless attractive because they increase the opportunity to discover potent, selective, and structurally novel compounds. Thus, the chemical spaces we most want to explore are becoming too large to evaluate exhaustively.

From a machine learning perspective, docking-based VS is a finite-domain black-box optimization problem. Given a fixed molecular library and a limited budget of evaluations from an expensive oracle, the goal is to identify as many of the highest-scoring candidates as possible while querying only a small fraction of the library. Active learning (AL) and Bayesian optimization (BO) offer a natural framework to reduce this cost by iteratively learning from a small set of evaluated molecules

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and selecting promising or informative candidates for subsequent docking. Prior work has shown that such approaches can recover highly ranked compounds while evaluating only a fraction of the full library (Reker & Schneider, 2015; Pyzer-Knapp, 2018; Graff et al., 2021; Cinquin et al., 2025).

However, the performance of AL/BO algorithms depends on several coupled design choices, including surrogate model, uncertainty quantification, molecular representation, and acquisition strategy (Cao et al., 2024; Kristiadi et al., 2024; Cinquin et al., 2025; Wille et al., 2026). Existing studies have explored these choices for virtual screening, but typically within restricted problem settings, limiting the generalizability of their conclusions. However, conclusions about BO methodology may depend strongly on the screening environment. Docking programs differ in search procedures, scoring functions, and approximations, producing distinct objective landscapes even for the same target and library (Boittier et al., 2020; Paggi et al., 2024; Chung et al., 2026). Biological targets and candidate libraries introduce additional variation in interaction landscapes, chemical composition, and screening scale. Consequently, conclusions about representations, uncertainty estimation, or acquisition strategies obtained in one setting may not transfer to another.

To address this gap, we introduce **CHEMVERSE**, a large-scale benchmark for adaptive VS. CHEMVERSE integrates 389 million docking scores across seven protein targets and three chemical libraries ranging from 2×10^6 to 10×10^6 molecules. We generate matched scores using GLIDE (Friesner et al., 2004; Halgren et al., 2004) and three programs from the VINA family,(Trott & Olson, 2009; Eberhardt et al., 2021; Yu et al., 2023; Quiroga & Villarreal, 2016), and complement these with publicly available DOCK3 screens (Coleman et al., 2013; Hall et al., 2025). Matched docking scores enable controlled cross-oracle comparisons on shared target-library pairs while spanning diverse biological targets, chemical libraries, and screening scales. Because all scores are precomputed, VS campaigns can be replayed efficiently, enabling systematic and reproducible comparisons without repeating the underlying docking calculations. Using CHEMVERSE, we systematically evaluate molecular representations, uncertainty-aware surrogate models, and acquisition strategies to determine which methodological choices are robust across screening conditions and which depend on the docking oracle, biological target, or chemical space.

Specifically, our contributions are:

- We introduce **CHEMVERSE**, a public benchmark designed to disentangle how docking oracle, biological target, and chemical space influence machine-learning-guided VS.
- We establish a controlled evaluation framework for testing whether molecular representations, uncertainty-aware surrogate models, and acquisition strategies generalize across screening conditions or require context-specific adaptation.
- We show that optimization performance varies substantially across docking oracles, tracks surrogate learnability and landscape smoothness, and is affected more strongly by molecular representation than by the choice among the uncertainty approximations considered here.

CHEMVERSE provides a standardized foundation for determining which methodological choices generalize across docking-based screening tasks – narrowing the gap between promising retrospective benchmarks and robust deployment across practical docking pipelines.

2 RELATED WORK

2.1 PROBLEM FORMULATION AND TERMINOLOGY

Let $\mathcal{X}$ denote a finite molecular library and $f_{o,t} : \mathcal{X} \to \mathbb{R}$ the docking score assigned by oracle o for biological target t, with scores oriented so that larger values are preferable. Given a budget of $B \ll |\mathcal{X}|$ evaluations, a sequential policy selects molecules using a surrogate model updated with previously observed scores. A common objective is to maximize top-k recovery,

$$R_k(\mathcal{D}_B) = \frac{|\mathcal{D}_B \cap T_k(f_{o,t})|}{k},$$

where $\mathcal{D}_B \subseteq \mathcal{X}$ is the set of evaluated molecules and $\mathcal{T}_k(f_{o,t})$ contains the library's k highest-scoring molecules. This setting has been described as active learning (AL), including pool-based

active learning, and Bayesian optimization (BO) (Graff et al., 2021; Cao et al., 2024; Kristiadi et al., 2024; Cinquin et al., 2025). The term pool-based AL emphasizes adaptive selection from a fixed candidate pool, whereas BO emphasizes identifying high-performing candidates rather than improving predictive accuracy across the full pool. Since the latter resembles the typical practical use case, we use BO throughout the remainder of this paper.

2.2 Methodology Benchmarks for Sample-Efficient Virtual Screening

A growing body of work has benchmarked iterative model-guided docking for large-scale VS. Prominently, DEEP DOCKING used fingerprint-based neural classifiers to filter billion-molecule libraries (Gentile et al., 2020), while MOLPAL systematically compared fingerprint-based and graph-based surrogates, acquisition functions, and batch sizes on libraries spanning approx. 10^4–10^8 molecules (Graff et al., 2021). Subsequent studies examined e.g. regression models, ensemble uncertainty estimates, and score-based or uncertainty-based acquisition in large-scale docking (Yang et al., 2021; Kim et al., 2024), as well as computational efficiency through graph neural network workflows and inexpensive linear-regression ensembles (Sivula et al., 2023; Marin et al., 2023).

Representation-focused studies extended these comparisons to pretrained molecular language models and graph encoders. Across docking and related discovery tasks in drug discovery, materials science, and catalysis, these studies examined molecular representations, surrogate models and their initialization, acquisition functions, and representation fine-tuning (Cao et al., 2024; Kristiadi et al., 2024; Cinquin et al., 2025; Pandey et al., 2026; Wille et al., 2026). Collectively, these studies span a broad methodological design space and identify combinations of representations, surrogates, and acquisition strategies that perform well within their respective evaluation settings.

2.3 Cross-Target and Cross-Oracle Benchmarks

At the same time, the transferability of methodological conclusions across biological targets and docking oracles has received less systematic attention. With respect to biological targets, DEEP DOCKING and HASTEN evaluated screening efficiency across multiple targets while holding the docking program fixed (Gentile et al., 2020; Sivula et al., 2023), whereas Kim et al. (2024) compared acquisition strategies across six Vina targets. DOCKSTRING offers a broader basis for such comparisons, providing Vina scores for a common set of over 260,000 molecules across 58 targets and benchmark tasks for prediction, VS, and molecular design (García-Ortegón et al., 2022).

The original MOLPAL benchmarks included both Vina and DOCK3 datasets, although differences in targets and chemical libraries complicate direct attribution of performance differences to the docking oracle (Graff et al., 2021). Using a shared modeling framework, Yang et al. (2021) investigated screening with DOCK3.7 and GLIDE. More direct comparisons were undertaken by Di Filippo et al. (2025), who applied a fixed fingerprint-based neural-network pipeline to ten targets scored with PLANTS and VINA. More recently, Chung et al. (2026) evaluated BO-based VS with VINA, GLIDE, and SILCS-MC on matched compounds, alongside a DeepAutoQSAR workflow and additional Vina–Glide comparisons across targets and library sizes. These findings indicate a high degree of oracle dependence, and motivate systematic benchmarks into how broadly recommendations about individual representations, surrogate models, and acquisition strategies transfer between screening settings.

CHEMVERSE builds on these foundations by making variation across docking oracles a central dimension of the benchmark. Systematic evaluation across biological targets and chemical libraries allows us to distinguish methodological choices that remain robust across screening environments from those that depend on the docking oracle, target, or library.

3 CHEMVERSE BENCHMARK

3.1 Benchmark Overview

CHEMVERSE is a benchmark for BO in docking-based VS across multiple docking oracles, biological targets, and chemical libraries of different sizes. It contains a total of 389 million docking scores spanning seven protein targets and three chemical libraries, with evaluations of shared target–library

combinations using GLIDE, AUTODOCK VINA, UNI-DOCK VINA, and UNI-DOCK VINARDO. We additionally incorporate publicly available DOCK3 scores where available.

3.2 CHEMICAL LIBRARIES AND PROTEIN TARGETS

We consider three molecular libraries spanning more than an order of magnitude in size: HTS-2M, containing approximately 2×10^6 molecules from the Enamine high-throughput screening collection; LSD-5M, containing approximately 5×10^6 molecules from the ZINC database; and ENAMINE-10M, containing approximately 10×10^6 random molecules from the Enamine REAL database (see Appendix A.2.1). Together, these libraries allow us to study BO across screening scales, with different fractions of the library evaluated at realistic docking budgets.

The dataset spans seven biomedically relevant protein targets: AmpC, D_4, σ_2R, EP4R, SERT, M^{Pro}, and 5-HT_{2A} (Figure A17, Table A2). These targets have been used in previous large-scale structure-based VS studies and span diverse protein families and ligand-binding environments (see A.2.2 for more details). This diversity allows us to evaluate whether trends observed for a particular docking oracle or active-learning method remain stable across distinct biological settings.

3.3 DOCKING ORACLES

For each target–library combination, we generate docking scores using GLIDE, AUTODOCK VINA, UNI-DOCK VINA, and UNI-DOCK VINARDO using identical protein structures with receptor and binding-site preparation adapted to the recommended workflow for each docking program. Full docking parameters, software versions, receptor preparation procedures, and pocket definitions are provided in Appendix A.2.3.

For the comparison to the DOCK3 docking software, we use the LSD-5M scores released by Hall et al. (2025), and include scores for all target proteins shared with CHEMVERSE. Because these data were generated independently, we treat them as complementary cross-oracle evidence rather than as fully matched reproductions of our docking pipeline.

We additionally evaluate BOLTZ-2 co-folding (Passaro et al., 2025) on LSD-10K, a subset of LSD-5M. Because exhaustive co-folding is prohibitively expensive at the million-molecule scale, we restrict this analysis to the 10K subset; results are reported in Appendix A.1.6.

4 EXPERIMENTS AND METHODS

We use CHEMVERSE to evaluate how molecular representations, uncertainty-estimation methods, and acquisition strategies affect BO performance in docking-based VS. All methods follow the common protocol described below. Additional implementation details are provided in Appendix A.3.

4.1 GENERAL WORKFLOW

For each target–library–oracle combination, we initialize a campaign by sampling a set of molecules $\mathcal{D}_0$ uniformly without replacement from the candidate library $\mathcal{X}$ and revealing their docking scores. At each subsequent round t, we fit the surrogate model to all observed molecule–score pairs, compute acquisition function values for the remaining candidates, evaluate a new batch of q molecules, and update the dataset of observations $\mathcal{D}_t$ accordingly. Each campaign comprises an initialization step followed by 10 acquisition rounds. We run five independent replicates per campaign with fixed seed populations across methods and oracles, in order to enable paired comparisons.

For all libraries, the initialization size is equal to the acquisition batch size, $|\mathcal{D}_0| = q$. We use $q = 50$ for LSD-10K, $q = 2000$ for HTS-2M, $q = 5000$ for LSD-5M, and $q = 10000$ for ENAMINE-10M corresponding to nominal library sizes of 10^4, 2×10^6, 5×10^6, and 10^7 respectively.

We measure top-k recovery at three thresholds: the top-100 molecules, the top-1000 molecules, and the top-0.1% of the effective candidate library size. The fixed thresholds assess recovery of a small number of highly ranked candidates, while the fractional threshold enables comparison across library sizes. For each threshold, we summarize performance using the area under the recovery curve (AUC) over the acquisition rounds, with higher AUC indicating greater and earlier recovery.

4.2 MOLECULAR REPRESENTATIONS

We evaluate six molecular representations spanning conventional fingerprints, pretrained graph and sequence encoders, and three-dimensional molecular representations (Table 1)

Table 1: Molecular representations evaluated in CHEMVERSE.

Representation	Type	Molecular Input	Reference
Morgan	Circular fingerprint	2D Graph	Rogers & Hahn (2010)
GROVER	Graph Transformer	2D Graph	Rong et al. (2020)
MolE	Graph Transformer	2D Graph	Méndez-Lucio et al. (2024)
T5Chem	Sequence Transformer	SMILES	Christofidellis et al. (2023)
MolFormer	Sequence Transformer	SMILES	Ross et al. (2022)
UniMol	3D geometric Transformer	3D Graph	Zhou et al. (2023) Ji et al. (2024)

Fingerprints use radius 2 and a length of 1024. All pretrained encoders remain frozen, and embeddings are computed before the BO campaigns. For UniMol, we use the 84-million-parameter Uni-Mol2 model from Ji et al. (2024), denoted *UniMol* throughout. Checkpoint identifiers, embedding extraction procedures, and conformer-generation details are provided in Appendix A.3.3.

4.3 UNCERTAINTY-AWARE SURROGATES

We compare three approaches to approximate Bayesian inference using a common neural-network surrogate architecture. For each molecule x, these methods estimate a predictive mean $\mu(x)$ and uncertainty $\sigma(x)$ from variation in the network parameters, which is used to compute acquisition function values for each molecule (see below).

- **Monte Carlo dropout** estimates predictive uncertainty from repeated forward passes with dropout enabled at inference time (Gal & Ghahramani, 2016).
- **SWAG** constructs a Gaussian approximation to the parameter posterior from the stochastic-gradient-descent trajectory and estimates predictive uncertainty by sampling network parameters (Maddox et al., 2019).
- **Laplace approximation** estimates predictive uncertainty using a local Gaussian approximation to the posterior over network parameters, centered at their maximum a posteriori estimate with covariance determined by the negative-log-posterior curvature (Daxberger et al., 2021).

Architecture, training schedules, prior and likelihood choices, covariance approximations, and sampling settings are specified in Appendix A.3.1.

4.4 ACQUISITION FUNCTIONS

We compare four pointwise acquisition functions, where $\mu(x)$ and $\sigma(x)$ denote the surrogate predictive mean and standard deviation, and $f^\star$ is the best observed objective value.

Greedy selection maximizes $\alpha_{\mathrm{Greedy}}(x) = \mu(x)$, while **upper confidence bound** (UCB) adds an uncertainty bonus, $\alpha_{\mathrm{UCB}}(x) = \mu(x) + \beta\sigma(x)$, with $\beta \geq 0$ controlling exploration. **LogEI** maximizes the logarithm of expected improvement, $\alpha_{\mathrm{LogEI}}(x) = \log \mathbb{E}[(f(x) - f^\star)+]$, whereas **LogPI** maximizes the logarithm of the probability of improvement, $\alpha_{\mathrm{LogPI}}(x) = \log \Pr(f(x) > f^\star)$, with $(z)_+ = \max(z, 0)$ and expectations and probabilities defined under the surrogate predictive distribution. Both use numerically stable logarithmic formulations, with LogEI following Ament et al. (2023).

At each round, we select a batch of q unevaluated molecules with the highest acquisition values. Implementation details are provided in Appendix A.3.2.

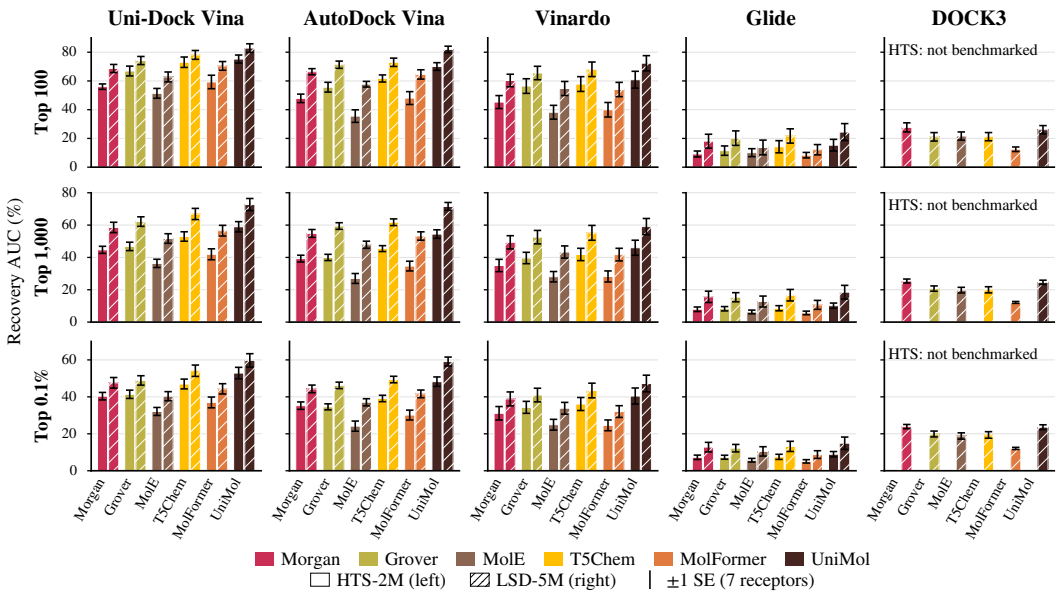


Figure 1: **Molecular representation performance across docking oracles and libraries.** Bars show mean recovery AUC for HTS-2M (solid) and LSD-5M (hatched), using a Laplace BNN surrogate; higher is better. Error bars indicate ± 1 standard error across seven receptors. UniMol leads for most oracles, while Morgan leads for DOCK3. Full evaluation details, as well as results for SWAG and MC dropout surrogates, are provided in Appendix A.1.1. Results for ENAMINE-10M are provided in Appeondix A.1.7. Top-k recovery plots and cumulative optima curves are provided in Appendix A.1.3 and A.1.4, respectively.

4.5 EXPERIMENTAL DESIGN

Representation comparison. We evaluate all six representations with each of the three uncertainty-estimation methods across the available oracle–receptor–lib combinations. For these experiments, the UCB acquisition function is used for its simplicity and good performance (Wang-Henderson et al., 2023). For each docking oracle, we select the representation with the highest mean recovery AUC for top-100, top-1000, and top-0.1% across biological targets.

Acquisition comparison. With the selected representation fixed for each oracle, we compare acquisition functions (Greedy, UCB, LogEI, and LogPI) using all uncertainty-estimation methods. This design evaluates the interaction between acquisition and uncertainty estimation conditional on the selected representation. Detailed hyperparameters, training procedures, software versions, and computational settings are provided in the Appendix A.3.

5 RESULTS

5.1 THE DOCKING ORACLE DOMINATES OPTIMIZATION PERFORMANCE

Figure 1 shows substantial differences in recovery efficiency across docking programs under matched conditions. The median best-to-worst AUC spread is approximately 38.7 percentage points across docking oracles, compared with 15.2 across molecular representations and 0.34 across BNN approximations. While not a formal variance decomposition, these ranges indicate that oracle-level variation substantially exceeds that associated with the modeling choices considered here.

Thus, different docking oracles induce score landscapes of substantially different learnability, making some easier for surrogate-guided search than others. Importantly, greater optimization efficiency does not imply that a docking oracle is more physically accurate or biologically predictive; it only reflects how readily high-ranking compounds under that oracle can be recovered by the BO procedure.

5.2 3D Molecular Representations Perform Best Across Docking Oracles

Across GLIDE and the three Vina-family docking programs, UniMol consistently achieves the highest mean recovery on both HTS-2M and LSD-5M. This ranking is broadly preserved across Laplace (Figure 1), MC Dropout (Figure A1a), and SWAG (Figure A1b). More broadly, these results indicate that representations incorporating three-dimensional structure are particularly well suited to learning docking objectives. While this is not entirely surprising given the additional geometric information they encode, it suggests that patterns learned from free-ligand conformers can remain informative about docked conformations, helping the surrogate identify features relevant to binding-site complementarity.

It should be noted that benchmarking against publicly available DOCK3 data shows a deviation from this trend. On LSD-5M, Morgan fingerprints achieve the highest recovery at the most stringent thresholds, with UniMol remaining competitive. However, since the DOCK3 data were generated independently, this contrast cannot be attributed solely to the docking program.

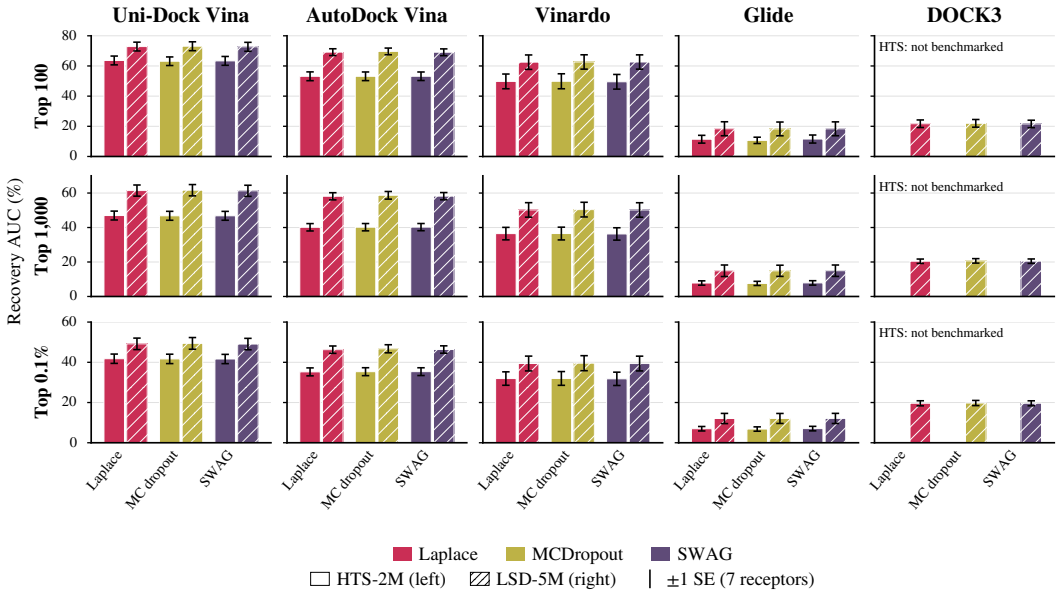


Figure 2: **BNN surrogate performance across docking oracles and screening libraries.** Bars show mean seed-subtracted recovery AUC, averaged across six molecular representations and seven receptors; higher is better. Error bars indicate ± 1 standard error across receptor means. Laplace, MC dropout, and SWAG achieve similar average performance across the evaluated settings.

5.3 Uncertainty Quantification Has Limited Impact in Large-Scale Virtual Screening

Figure 2 compares MC Dropout, SWAG, and Laplace approximation across molecular representations, docking programs, and recovery thresholds. Overall, optimization performance is broadly similar across the three uncertainty-estimation methods. Averaged over all molecular representations, the largest difference between the best- and worst-performing BNN approximation is less than one recovery-AUC percentage point across the evaluated settings.

MC Dropout shows a small numerical advantage overall, particularly on LSD-5M, but no uncertainty approximation consistently dominates across libraries and docking programs. Differences are occasionally larger for individual molecular representations, suggesting some interaction between representation and uncertainty estimation, but these effects are not systematic across the benchmark.

These results indicate that, among the approximate Bayesian neural networks considered here, the uncertainty-estimation method has a comparatively modest effect on BO performance.

5.4 Greedy and UCB Consistently Outperform Improvement-Based Acquisition

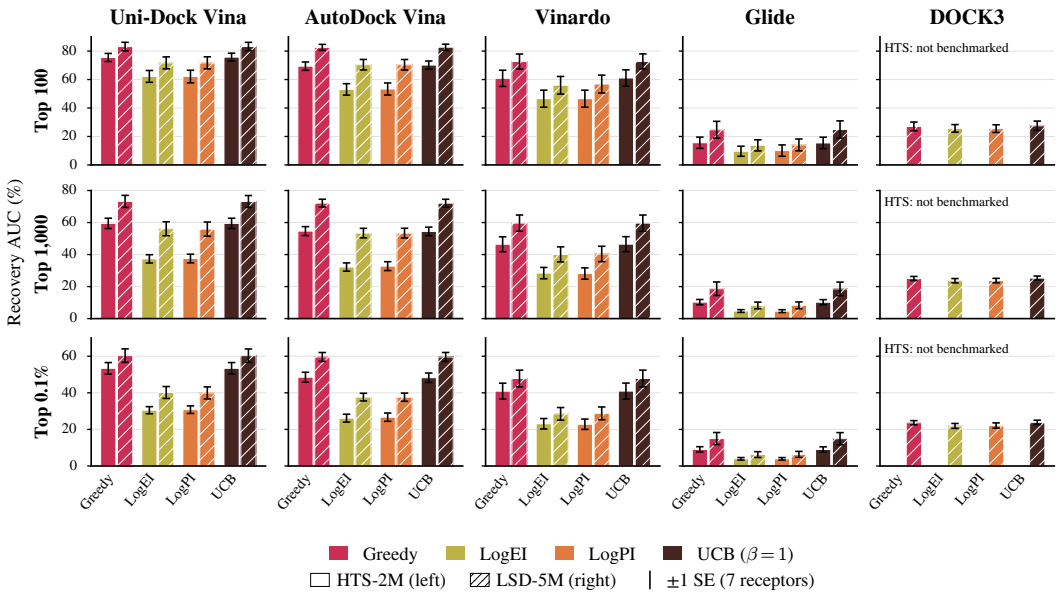


Figure 3: **Acquisition-function performance across docking oracles and screening libraries.** Error bars indicate ± 1 SE across seven receptors. Greedy and UCB achieve similar performance and consistently outperform LogEI and LogPI in mean AUC. UniMol is used except for DOCK3, which uses Morgan. For MC Dropout and SWAG results, refer to Figure A2.

With the selected representation fixed for each oracle, Figure 3 compares Greedy, UCB, LogEI, and LogPI using the Laplace surrogate. Across docking programs, libraries, and recovery thresholds, Greedy and UCB consistently achieve the highest mean AUC recovery while LogEI and LogPI do not outperform either strategy in any of the evaluated comparisons. The same qualitative pattern holds for MC Dropout and SWAG (see Appendix A.1.2). UCB attains the highest mean AUC more frequently than Greedy, but absolute differences remain below one percentage point across the reported settings and methods. In contrast, the gap between Greedy/UCB and LogEI/LogPI is substantially larger, indicating that the dominant distinction is between these two classes of acquisition strategies.

Notably, we find a substantial degree of coupling between acquisition function and uncertainty quantification method. While the UCB optimization behavior remains largely stable across MC Dropout, SWAG, and Laplace, LogEI and LogPI show substantially larger variation. Yet, the trend that LogEI/LogPI lead to slower top-candidate recovery compared to Greedy/UCB, remains stable across the all uncertainty quantification techniques.

Overall, these results establish Greedy acquisition as a strong baseline for large-scale BO-based VS. Moreover, we show that surrogate and acquisition choices interact, supporting their joint evaluation in method benchmarks for VS.

5.5 Oracle Learnability and Landscape Smoothness Track Optimization Performance

The comparable performance of Greedy and UCB, together with the small differences among uncertainty-estimation methods, suggests that optimization efficiency depends primarily on the learnability of the underlying docking landscape.

Across molecular representations and libraries, predictive accuracy is strongly associated with recovery AUC (Figure 4). Test R^2 correlates with campaign performance at $r \approx 0.85$–0.88 (test Spearman ρ: $r \approx 0.91$–0.94), while a local k-nearest-neighbor predictor shows a similar relationship ($r \approx 0.83$–0.87). This relationship also holds within individual docking landscapes, where test R^2 ranks representations similarly to recovery AUC (within-oracle rank correlation ≈ 0.86–0.88).

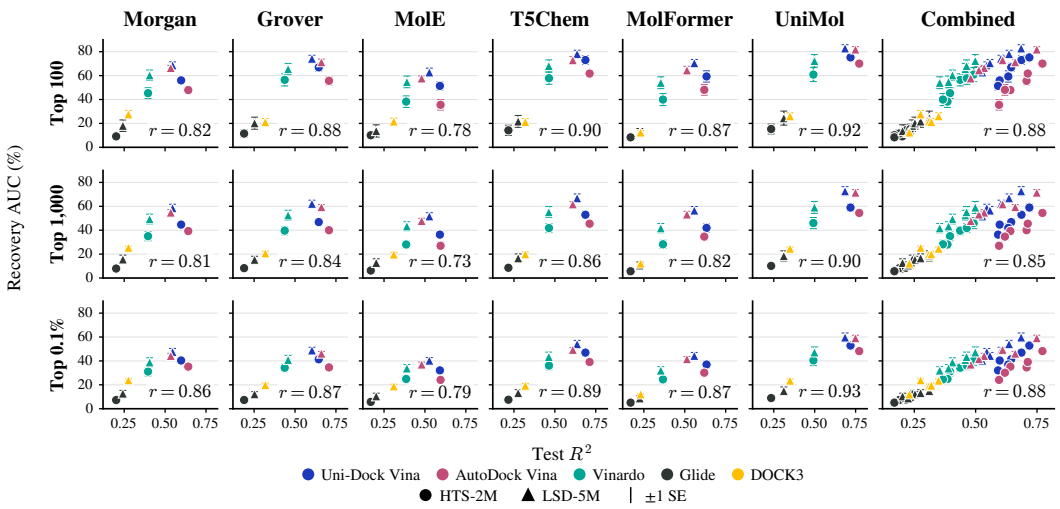


Figure 4: **Predictive accuracy and BO performance across molecular representations.** Vertical error bars indicate ± 1 SE of AUC. Pearson correlations (r) are calculated across both libraries within each panel. DOCK3 results are available only for LSD-5M.

Conversely, rougher landscapes are associated with poorer optimization. ROGI-XD roughness (Graff et al., 2023) correlates negatively with recovery AUC ($r \approx -0.74$ to -0.79), largely reflecting differences between docking oracles, while estimated Lipschitz slopes show a weaker pooled association ($r \approx -0.41$ to -0.47). Within an oracle, however, Lipschitz slopes better capture representation-level differences in recovery (rank correlation ≈ -0.31 to -0.48), whereas ROGI-XD does not. Additional analyses are provided in Appendix A.1.5.

Overall, these results suggest that BO performance in this regime is strongly associated with the predictability and smoothness of the docking landscape, helping explain the large oracle-level differences observed earlier. The learnability gap between oracles persists with up to 100k training molecules (Appendix A.1.5), although these relationships are descriptive rather than causal.

6 DISCUSSION

Our results suggest that active-learning performance in docking-based VS depends more strongly on the learnability of the docking objective than on the treatment of uncertainty. The strong association between predictive accuracy, landscape smoothness, and recovery efficiency points toward improved molecular representations and docking-score predictors as an important direction for accelerated VS. We hypothesize that this effect is amplified by the finite, discrete, and large-batch nature of screening, where success may depend primarily on ranking high-scoring molecules accurately rather than resolving fine-grained uncertainty.

Our acquisition results should be interpreted in the context of pointwise top-q batch selection, which does not model interactions or redundancy within a batch and may diminish the practical value of uncertainty at large batch sizes. Thus, the similarity between Greedy and UCB does not imply that uncertainty is generally unnecessary, but motivates further study of batch acquisition methods for large discrete molecular spaces (Bellamy et al., 2022; Fromer et al., 2024; Bloor et al., 2026).

Scaling also introduces a separate inference bottleneck. Even when active learning greatly reduces the number of docking evaluations, conventional pool-based methods still repeatedly score large candidate libraries with the surrogate. Addressing this challenge through pruning, approximate search, or hierarchical candidate selection is complementary to the present work (Graff et al., 2022; Wang-Henderson et al., 2023; Haddadnia et al., 2026).

Finally, CHEMVERSE provides a reusable benchmark of 389 million docking scores for studying representations, surrogates, uncertainty estimation, acquisition strategies, and transfer across docking oracles.

AI USE STATEMENT

In this work, we have not used generative AI tools for any tasks with required disclosure (such as generating synthetic data sets, developing theoretical models, or implementing methods), and all other tasks requiring disclosure are not applicable to this work. Additionally, we used generative AI tools for tasks with recommended disclosure: specifically, to draft parts of the research paper and to create or edit software code (including the code used to generate our figures). We have reviewed all AI-assisted work. Specifically, the generated text was manually evaluated by the authors, and all LLM-generated code was reviewed, tested, and verified for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024. URL https://doi.org/10.1038/s41586-024-07487-w.

Assaf Alon, Jiankun Lyu, Joao M. Braz, Tia A. Tummino, Veronica Craik, Matthew J. O'Meara, Chase M. Webb, Dmytro S. Radchenko, Yurii S. Moroz, Xi-Ping Huang, Yongfeng Liu, Bryan L. Roth, John J. Irwin, Allan I. Basbaum, Brian K. Shoichet, and Andrew C. Kruse. Structures of the 2 receptor enable docking for bioactive ligand discovery. *Nature*, 600(7890):759–764, 2021. URL https://doi.org/10.1038/s41586-021-04175-x.

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization. In *Advances in Neural Information Processing Systems 36*, NeurIPS 2023, pp. 20577–20612, 2023. URL https://doi.org/10.52202/075280-0904.

Hugo Bellamy, Abbi Abdel Rehim, Oghenejokpeme I. Orhobor, and Ross King. Batched bayesian optimization for drug design in noisy environments. *Journal of Chemical Information and Modeling*, 62(17):3970–3981, August 2022. ISSN 1549-960X. doi: 10.1021/acs.jcim.2c00602. URL http://dx.doi.org/10.1021/acs.jcim.2c00602.

Maximilian Bloor, Liyuan Xu, Hrvoje Stojic, and Victor Picheny. B3o: Scalable boltzmann batch bayesian optimization, 2026. URL https://arxiv.org/abs/2606.30228.

Eric D. Boittier, Yat Yin Tang, McKenna E. Buckley, Zachariah P. Schuurs, Derek J. Richard, and Neha S. Gandhi. Assessing molecular docking tools to guide targeted drug discovery of cd38 inhibitors. *International Journal of Molecular Sciences*, 21(15):5183, 2020. URL https://doi.org/10.3390/ijms21155183.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Zhonglin Cao, Simone Sciabola, and Ye Wang. Large-scale pretraining improves sample efficiency of active learning-based virtual screening. *Journal of Chemical Information and Modeling*, 64(6): 1882–1891, 2024. URL https://doi.org/10.1021/acs.jcim.3c01938.

Domiziana Cecchini, AkshatKumar Nigam, Ming Tang, Joana Reis, Matt Koop, Andrea Gottinger, Callum Robert Nicoll, Yao Wang, Abhilash Jayaraj, Süleyman Selim Çınaroğlu, Ricarda Törner, Yehor Malets, Minko Gehev, Krishna M. Padmanabha Das, Kelly Churion, Jongwan Kim, Nidhin Thomas, Yong Li, Hyuk-Soo Seo, Sirano Dhe-Paganon, Christopher Secker, Mohammad Haddadnia, Alexander Hasson, Minkai Li, Abhishek Kumar, Roni Levin-Konigsberg, Eun-Bee Choi, Geoffrey I. Shapiro, Huel Cox, Luke Sebastian, Chelsea Braithwaite, Puspalata Bashyal, Dmytro S. Radchenko, Aditya Kumar, Lei Yang, Pierre-Yves Aquilanti, Henry Gabb, Amr Alhossary, Eric O'Neill, Gerhard Wagner, Alán Aspuru-Guzik, Yurii S. Moroz, Charalampos G. Kalodimos, Konstantin Fackeldey, John D. Schuetz, Andrea Mattevi, Haribabu Arthanari, and Christoph Gorgulla. Ai-enhanced adaptive virtual screening of large libraries for ligand discovery. *Nature Biotechnology*, 2026. URL https://doi.org/10.1038/s41587-026-03217-x.

Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. Unifying molecular and textual representations via multi-task language modelling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and

Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6140–6157, 2023. URL https://proceedings.mlr.press/v202/christofidellis23a.html.

Joseph Chung, Aashish Bhatt, Jacob Ede Levine, Yi-Chun Lin, Mingtian Zhao, Alexander D. MacKerell, Sunhwan Jo, Sai Chandra Kosaraju, and Yun Lyna Luo. Docking-score landscapes shape active-learning performance across vina, glide, and silcs. *Journal of Computer-Aided Molecular Design*, 40(1), 2026. URL https://doi.org/10.1007/s10822-026-00927-x.

Tristan Cinquin, Stanley Lo, Felix Strieth-Kalthoff, Alan Aspuru-Guzik, Geoff Pleiss, Robert Bamler, Tim G. J. Rudner, Vincent Fortuin, and Agustinus Kristiadi. What actually matters for materials discovery: Pitfalls and recommendations in bayesian optimization. In *AI for Accelerated Materials Design - ICLR 2025*, 2025. URL https://openreview.net/forum?id=GbV2JjfnSY.

Ryan G. Coleman, Michael Carchia, Teague Sterling, John J. Irwin, and Brian K. Shoichet. Ligand pose and orientational sampling in molecular docking. *PLOS ONE*, 8(10):e75992, 2013. URL https://doi.org/10.1371/journal.pone.0075992.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux - effortless bayesian deep learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a7c9585703d275249f30a088cebba0ad-Paper.pdf.

DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Miloš Stanojević, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. The DeepMind JAX Ecosystem, 2020. URL http://github.com/google-deepmind.

Juan I. Di Filippo, Santiago Rómoli, and Claudio N. Cavasotto. Assessing the robustness and scalability of machine learning methods to accelerate ultralarge high-throughput docking campaigns. *ACS Omega*, 10(15):15598–15609, April 2025. ISSN 2470-1343. doi: 10.1021/acsomega.5c00829. URL http://dx.doi.org/10.1021/acsomega.5c00829.

Jerome Eberhardt, Diogo Santos-Martins, Andreas F. Tillack, and Stefano Forli. Autodock vina 1.2.0: New docking methods, expanded force field, and python bindings. *Journal of Chemical Information and Modeling*, 61(8):3891–3898, 2021. URL https://doi.org/10.1021/acs.jcim.1c00203.

Sean R. Eddy. Accelerated profile HMM searches. *PLoS Computational Biology*, 7(10):e1002195, 2011. URL https://doi.org/10.1371/journal.pcbi.1002195.

Elissa A. Fink, Conner Bardine, Stefan Gahbauer, Isha Singh, Tyler C. Detomasi, Kris White, Shuo Gu, Xiaobo Wan, Jun Chen, Beatrice Ary, Isabella Glenn, Joseph O'Connell, Henry O'Donnell, Pavla Fajtová, Jiankun Lyu, Seth Vigneron, Nicholas J. Young, Ivan S. Kondratov, Arghavan Alisoltani, Lacy M. Simons, Ramon Lorenzo-Redondo, Egon A. Ozer, Judd F. Hultquist, Anthony J. O'Donoghue, Yurii S. Moroz, Jack Taunton, Adam R. Renslo, John J. Irwin, Adolfo García-Sastre, Brian K. Shoichet, and Charles S. Craik. Large library docking for novel SARS-CoV-2 main protease non-covalent and covalent inhibitors. *Protein Science*, 32(8), 2023. URL https://doi.org/10.1002/pro.4712.

Richard A. Friesner, Jay L. Banks, Robert B. Murphy, Thomas A. Halgren, Jasna J. Klicic, Daniel T. Mainz, Matthew P. Repasky, Eric H. Knoll, Mee Shelley, Jason K. Perry, David E. Shaw, Perry Francis, and Peter S. Shenkin. Glide: A new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of Medicinal Chemistry*, 47(7):1739–1749, 2004. URL https://doi.org/10.1021/jm0306430.

Jenna C. Fromer, David E. Graff, and Connor W. Coley. Pareto optimization to accelerate multi-objective virtual screening. *Digital Discovery*, 3(3):467–481, 2024. ISSN 2635-098X. doi: 10.1039/d3dd00227f. URL http://dx.doi.org/10.1039/d3dd00227f.

Stefan Gahbauer, Chelsea DeLeon, Joao M. Braz, Veronica Craik, Hye Jin Kang, Xiaobo Wan, Xi-Ping Huang, Christian B. Billesbølle, Yongfeng Liu, Tao Che, Ishan Deshpande, Madison Jewell, Elissa A. Fink, Ivan S. Kondratov, Yurii S. Moroz, John J. Irwin, Allan I. Basbaum, Bryan L. Roth, and Brian K. Shoichet. Docking for EP4R antagonists active against inflammatory pain. *Nature Communications*, 14(1), 2023. URL https://doi.org/10.1038/s41467-023-43506-6.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 2016. URL https://proceedings.mlr.press/v48/gal16.html.

Miguel García-Ortegón, Gregor N. C. Simm, Austin J. Tripp, José Miguel Hernández-Lobato, Andreas Bender, and Sergio Bacallado. Dockstring: Easy molecular docking yields better benchmarks for ligand design. *Journal of Chemical Information and Modeling*, 62(15):3486–3502, 2022. ISSN 1549-960X. doi: 10.1021/acs.jcim.1c01334. URL http://dx.doi.org/10.1021/acs.jcim.1c01334.

Francesco Gentile, Vibudh Agrawal, Michael Hsing, Anh-Tien Ton, Fuqiang Ban, Ulf Norinder, Martin E. Gleave, and Artem Cherkasov. Deep docking: A deep learning platform for augmentation of structure based drug discovery. *ACS Central Science*, 6(6):939–949, May 2020. ISSN 2374-7951. doi: 10.1021/acscentsci.0c00229. URL http://dx.doi.org/10.1021/acscentsci.0c00229.

Christoph Gorgulla, Andras Boeszoermenyi, Zi-Fu Wang, Patrick D. Fischer, Paul W. Coote, Krishna M. Padmanabha Das, Yehor S. Malets, Dmytro S. Radchenko, Yurii S. Moroz, David A. Scott, Konstantin Fackeldey, Moritz Hoffmann, Iryna Iavniuk, Gerhard Wagner, and Haribabu Arthanari. An open-source drug discovery platform enables ultra-large virtual screens. *Nature*, 580(7805):663–668, 2020. URL https://doi.org/10.1038/s41586-020-2117-z.

Christoph Gorgulla, Abhilash Jayaraj, Konstantin Fackeldey, and Haribabu Arthanari. Emerging frontiers in virtual drug discovery: From quantum mechanical methods to deep learning approaches. *Current Opinion in Chemical Biology*, 69:102156, 2022. URL https://doi.org/10.1016/j.cbpa.2022.102156.

David E. Graff, Eugene I. Shakhnovich, and Connor W. Coley. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical Science*, 12(22):7866–7881, 2021. URL https://doi.org/10.1039/d0sc06805e.

David E. Graff, Matteo Aldeghi, Joseph A. Morrone, Kirk E. Jordan, Edward O. Pyzer-Knapp, and Connor W. Coley. Self-focusing virtual screening with active design space pruning. *Journal of Chemical Information and Modeling*, 62(16):3854–3862, August 2022. ISSN 1549-960X. doi: 10.1021/acs.jcim.2c00554. URL http://dx.doi.org/10.1021/acs.jcim.2c00554.

David E. Graff, Edward O. Pyzer-Knapp, Kirk E. Jordan, Eugene I. Shakhnovich, and Connor W. Coley. Evaluating the roughness of structure–property relationships using pretrained molecular representations. *Digital Discovery*, 2(5):1452–1460, 2023. ISSN 2635-098X. doi: 10.1039/d3dd00088e. URL http://dx.doi.org/10.1039/d3dd00088e.

Oleksandr O. Grygorenko, Dmytro S. Radchenko, Igor Dziuba, Alexander Chuprina, Kateryna E. Gubina, and Yurii S. Moroz. Generating multibillion chemical space of readily accessible screening compounds. *iScience*, 23(11):101681, 2020. URL https://doi.org/10.1016/j.isci.2020.101681.

Mohammad Haddadnia, Yuvan Chali, Abhilash Jayaraj, Constance Kraay, Joana Reis, Felix Strieth-Kalthoff, and Haribabu Arthanari. Target-aware bandit allocation for scalable surrogate optimization in chemical space. In *Forty-third International Conference on Machine Learning*,

2026. doi: 10.48550/arXiv.2606.26657. URL https://openreview.net/forum?id=5WwoJ2W0nL.

Thomas A. Halgren, Robert B. Murphy, Richard A. Friesner, Hege S. Beard, Leah L. Frye, W. Thomas Pollard, and Jay L. Banks. Glide: A new approach for rapid, accurate docking and scoring. 2. enrichment factors in database screening. *Journal of Medicinal Chemistry*, 47 (7):1750–1759, 2004. URL https://doi.org/10.1021/jm030644s.

Brendan W. Hall, Tia A. Tummino, Khanh Tang, Olivier Mailhot, Mar Castanon, John J. Irwin, and Brian K. Shoichet. A database for large-scale docking and experimental results. *Journal of Chemical Information and Modeling*, 65(9):4458–4467, 2025. URL https://doi.org/10.1021/acs.jcim.5c00394.

Torsten Hoffmann and Marcus Gastreich. The next level in chemical space navigation: going far beyond enumerable compound libraries. *Drug Discovery Today*, 24(5):1148–1156, 2019. URL https://doi.org/10.1016/j.drudis.2019.02.013.

Thomas A. Hopf, Anna G. Green, Benjamin Schubert, Sophia Mersmann, Charlotta P. I. Schärfe, John B. Ingraham, Agnes Toth-Petroczy, Kelly Brock, Adam J. Riesselman, Perry Palmedo, Chan Kang, Robert Sheridan, Eli J. Draizen, Christian Dallago, Chris Sander, and Debora S. Marks. The EVcouplings Python framework for coevolutionary sequence analysis. *Bioinformatics*, 35 (9):1582–1584, 2019. URL https://doi.org/10.1093/bioinformatics/bty862.

Hyosoon Jang, Hyunjin Seo, Honghui Kim, Seonghyun Park, Taewon Kim, Yunhui Jang, and Sungsoo Ahn. A systematic evaluation of co-folding model representations for small-molecule learning, 2026. URL https://arxiv.org/abs/2602.13249.

Xiaohong Ji, Zhen Wang, Zhifeng Gao, Hang Zheng, Linfeng Zhang, Guolin Ke, and Weinan E. Exploring molecular pretraining model at scale. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/53923bb44655a7defb31c7744c01b62b-Paper-Conference.pdf.

Shali Jiang, Henry Chai, Javier Gonzalez, and Roman Garnett. BINOCULARS for efficient, nonmyopic sequential experimental design. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4794–4803. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/jiang20b.html.

Patrick Kidger and Cristian Garcia. Equinox: neural networks in JAX via callable PyTrees and filtered transformations. *Differentiable Programming workshop at Neural Information Processing Systems 2021*, 2021. URL https://doi.org/10.48550/arXiv.2111.00254.

Jeonghyeon Kim, Juno Nam, and Seongok Ryu. Understanding active learning of molecular docking and its applications, 2024. URL https://arxiv.org/abs/2406.12919.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR 2015)*, 2015. doi: 10.48550/arXiv.1412.6980. URL http://dx.doi.org/10.48550/arXiv.1412.6980.

Agustinus Kristiadi, Felix Strieth-Kalthoff, Marta Skreta, Pascal Poupart, Alán Aspuru-Guzik, and Geoff Pleiss. A sober look at llms for material discovery: Are they actually good for bayesian optimization over molecules? In *ICML*, pp. 25603–25622, 2024. URL https://proceedings.mlr.press/v235/kristiadi24a.html.

Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez-Schmidt, David Cosgrove, Sereina Riniker, Peter Gedeck, Riccardo Vianello, Nadine Schneider, Eisuke Kawashima, Dan N, Gareth Jones, Andrew Dalke, Brian Cole, Matt Swain, Samo Turk, Alexander Savelyev, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Daniel Probst, Kazuya Ujihara, Vincent F. Scalfani, Guillaume Godin, Juuso Lehtivarjo, Axel Pahl, Rachel Walker, Francois Berenger, and Jason D. Biggs. Rdkit (release 2023-03-2), 2023. URL https://doi.org/10.5281/ZENODO.8053810.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations (ICLR 2017)*, 2017. doi: 10.48550/arXiv.1608.03983. URL http://dx.doi.org/10.48550/arXiv.1608.03983.

Jiankun Lyu, Sheng Wang, Trent E. Balius, Isha Singh, Anat Levit, Yurii S. Moroz, Matthew J. O'Meara, Tao Che, Enkhjargal Algaa, Kateryna Tolmachova, Andrey A. Tolmachev, Brian K. Shoichet, Bryan L. Roth, and John J. Irwin. Ultra-large library docking for discovering new chemotypes. *Nature*, 566(7743):224–229, 2019. URL https://doi.org/10.1038/s41586-019-0917-9.

Jiankun Lyu, Nicholas Kapolka, Ryan Gumpper, Assaf Alon, Liang Wang, Manish K. Jain, Ximena Barros-Álvarez, Kensuke Sakamoto, Yoojoong Kim, Jeffrey DiBerto, Kuglae Kim, Isabella S. Glenn, Tia A. Tummino, Sijie Huang, John J. Irwin, Olga O. Tarkhanova, Yurii Moroz, Georgios Skiniotis, Andrew C. Kruse, Brian K. Shoichet, and Bryan L. Roth. Alphafold2 structures guide prospective ligand discovery. *Science*, 384(6702), 2024. URL https://doi.org/10.1126/science.adn6354.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pp. 13153–13164, 2019. URL https://papers.neurips.cc/paper_files/paper/2019/file/118921efba23fc329e6560b27861f0c2-Paper.pdf.

Egor Marin, Margarita Kovaleva, Maria Kadukova, Khalid Mustafin, Polina Khorn, Andrey Rogachev, Alexey Mishin, Albert Guskov, and Valentin Borshchevskiy. Regression-based active learning for accessible acceleration of ultra-large library docking. *Journal of Chemical Information and Modeling*, 64(7):2612–2623, December 2023. ISSN 1549-960X. doi: 10.1021/acs.jcim.3c01661. URL http://dx.doi.org/10.1021/acs.jcim.3c01661.

Oscar Méndez-Lucio, Christos A Nicolaou, and Berton Earnshaw. MolE: a foundation model for molecular graphs using disentangled attention. *Nature Communications*, 15(1):9431, 2024. URL https://doi.org/10.1038/s41467-024-53751-y.

Mayako Michino, Alexandre Beautrait, Nicholas A. Boyles, Aparna Nadupalli, Alexey Dementiev, Shan Sun, John Ginn, Leigh Baxt, Robert Suto, Ruslana Bryk, Steven V. Jerome, David J. Huggins, and Jeremie Vendome. Shape-based virtual screening of a billion-compound library identifies mycobacterial lipoamide dehydrogenase inhibitors. *ACS Bio & Med Chem Au*, 3(6):507–515, 2023. URL https://doi.org/10.1021/acsbiomedchemau.3c00046.

Joseph M. Paggi, Ayush Pandit, and Ron O. Dror. The art and science of molecular docking. *Annual Review of Biochemistry*, 93(1):389–410, 2024. URL https://doi.org/10.1146/annurev-biochem-030222-120000.

Mohit Pandey, Tanvir Sajed, Ivan Semenov, Renfei Zhang, Fuqiang Ban, Ekaterina Manskaia, Martin Ester, and Artem Cherkasov. Deep docking, part 2: an amplified ddu platform for ultra-large virtual screening. *Chemical Science*, 17(27):13677–13690, 2026. ISSN 2041-6539. doi: 10.1039/d5sc09599a. URL http://dx.doi.org/10.1039/d5sc09599a.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, David Kwabi-Addo, Dominique Beaini, Tommi Jaakkola, and Regina Barzilay. Boltz-2: towards accurate and efficient binding affinity prediction. *bioRxiv*, pp. 2025.06.14.659707, 2025. URL https://doi.org/10.1101/2025.06.14.659707.

E. O. Pyzer-Knapp. Bayesian optimization for accelerated drug discovery. *IBM Journal of Research and Development*, 62(6):2:1–2:7, 2018. URL https://doi.org/10.1147/jrd.2018.2881731.

Rodrigo Quiroga and Marcos A. Villarreal. Vinardo: A scoring function based on autodock vina improves scoring, docking, and virtual screening. *PLOS ONE*, 11(5):e0155183, 2016. URL https://doi.org/10.1371/journal.pone.0155183.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://doi.org/10.48550/arXiv.1910.10683.

Daniel Reker and Gisbert Schneider. Active-learning strategies in computer-assisted drug discovery. *Drug Discovery Today*, 20(4):458–465, 2015. URL https://doi.org/10.1016/j.drudis.2014.12.004.

Sereina Riniker and Gregory A. Landrum. Better informed distance geometry: using what we know to improve conformation generation. *Journal of Chemical Information and Modeling*, 55(12): 2562–2574, 2015. URL https://doi.org/10.1021/acs.jcim.5b00654.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. URL https://doi.org/10.1021/ci100050t.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/94aef38441efa3380a3bed3faf1f9d5d-Paper.pdf.

Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022. URL https://doi.org/10.1038/s42256-022-00580-7.

Arman A. Sadybekov, Anastasiia V. Sadybekov, Yongfeng Liu, Christos Iliopoulos-Tsoutsouvas, Xi-Ping Huang, Julie Pickett, Blake Houser, Nilkanth Patel, Ngan K. Tran, Fei Tong, Nikolai Zvonok, Manish K. Jain, Olena Savych, Dmytro S. Radchenko, Spyros P. Nikas, Nicos A. Petasis, Yurii S. Moroz, Bryan L. Roth, Alexandros Makriyannis, and Vsevolod Katritch. Synthon-based ligand discovery in virtual libraries of over 11 billion compounds. *Nature*, 601(7893):452–459, 2021. URL https://doi.org/10.1038/s41586-021-04220-9.

Diogo Santos-Martins, Yuming He, Jerome Eberhardt, Priyansh Sharma, Nicolas Bruciaferri, Matthew Holcomb, Marcos A Llanos, Amber Hansel-Harris, Andrew P Barkdull, Allison F Tillack, et al. Meeko: Molecule parametrization and software interoperability for docking and beyond. *Journal of Chemical Information and Modeling*, 2025. URL https://doi.org/10.1021/acs.jcim.5c02271.

Schrödinger, LLC. The PyMOL molecular graphics system, version 2.5, 2021.

Isha Singh, Anubha Seth, Christian B. Billesbølle, Joao Braz, Ramona M. Rodriguiz, Kasturi Roy, Bethlehem Bekele, Veronica Craik, Xi-Ping Huang, Danila Boytsov, Vladimir M. Pogorelov, Parnian Lak, Henry O'Donnell, Walter Sandtner, John J. Irwin, Bryan L. Roth, Allan I. Basbaum, William C. Wetsel, Aashish Manglik, Brian K. Shoichet, and Gary Rudnick. Structure-based discovery of conformationally selective inhibitors of the serotonin transporter. *Cell*, 186(10): 2160–2175.e17, 2023. URL https://doi.org/10.1016/j.cell.2023.04.010.

Toni Sivula, Laxman Yetukuri, Tuomo Kalliokoski, Heikki Käsnänen, Antti Poso, and Ina Pöhner. Machine learning-boosted docking enables the efficient structure-based virtual screening of giga-scale enumerated chemical libraries. *Journal of Chemical Information and Modeling*, 63(18): 5773–5783, 2023. ISSN 1549-960X. doi: 10.1021/acs.jcim.3c01239. URL http://dx.doi.org/10.1021/acs.jcim.3c01239.

Baris E. Suzek, Yuqi Wang, Hongzhan Huang, Peter B. McGarvey, Cathy H. Wu, and the UniProt Consortium. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015. URL https://doi.org/10.1093/bioinformatics/btu739.

Oleg Trott and Arthur J. Olson. Autodock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2009. URL https://doi.org/10.1002/jcc.21334.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Junmei Wang, Wei Wang, Peter A Kollman, and David A Case. Automatic atom type and bond type perception in molecular mechanical calculations. *Journal of Molecular Graphics and Modelling*, 25(2):247–260, 2006. URL https://doi.org/10.1016/j.jmgm.2005.12.005.

Shuzhe Wang, Jagna Witek, Gregory A. Landrum, and Sereina Riniker. Improving conformer generation for small rings and macrocycles based on distance geometry and experimental torsional-angle preferences. *Journal of Chemical Information and Modeling*, 60(4):2044–2058, 2020. URL https://doi.org/10.1021/acs.jcim.0c00025.

Miles Wang-Henderson, Bartu Soyuer, Parnian Kassraie, Andreas Krause, and Ilija Bogunovic. Graph neural network powered bayesian optimization for large molecular spaces. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. URL https://openreview.net/forum?id=QIrgM7uybw.

Wendy A. Warr, Marc C. Nicklaus, Christos A. Nicolaou, and Matthias Rarey. Exploration of ultralarge compound collections for drug discovery. *Journal of Chemical Information and Modeling*, 62(9):2021–2034, 2022. URL https://doi.org/10.1021/acs.jcim.2c00224.

Tobias Weber, Bálint Mucsányi, Lenard Rommel, Thomas Christie, Lars Kasüschke, Marvin Pförtner, and Philipp Hennig. laplax – laplace approximations with jax, 2025. URL https://doi.org/10.48550/ARXIV.2507.17013.

Henrik Wille, Luis-Finley Schütz, and Felix Strieth-Kalthoff. Domain-adapted molecular language models for efficient search of make-on-demand libraries, 2026. URL https://doi.org/10.48550/ARXIV.2608.17567.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1: democratizing biomolecular interaction modeling. *bioRxiv*, pp. 2024.11.19.624167, 2024. URL https://doi.org/10.1101/2024.11.19.624167.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, 2020. URL https://doi.org/10.18653/v1/2020.emnlp-demos.6.

Ying Yang, Kun Yao, Matthew P. Repasky, Karl Leswing, Robert Abel, Brian K. Shoichet, and Steven V. Jerome. Efficient exploration of chemical space with docking and deep learning. *Journal of Chemical Theory and Computation*, 17(11):7106–7119, 2021. URL https://doi.org/10.1021/acs.jctc.1c00810.

Yuejiang Yu, Chun Cai, Jiayue Wang, Zonghua Bo, Zhengdan Zhu, and Hang Zheng. Uni-dock: Gpu-accelerated docking enables ultralarge virtual screening. *Journal of Chemical Theory and Computation*, 19(11):3336–3345, 2023. URL https://doi.org/10.1021/acs.jctc.2c01145.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6K2RM6wVqKu.

A Appendix

A.1 Additional Results

A.1.1 Molecular Representation Recoveries for MCDropout and SWAG

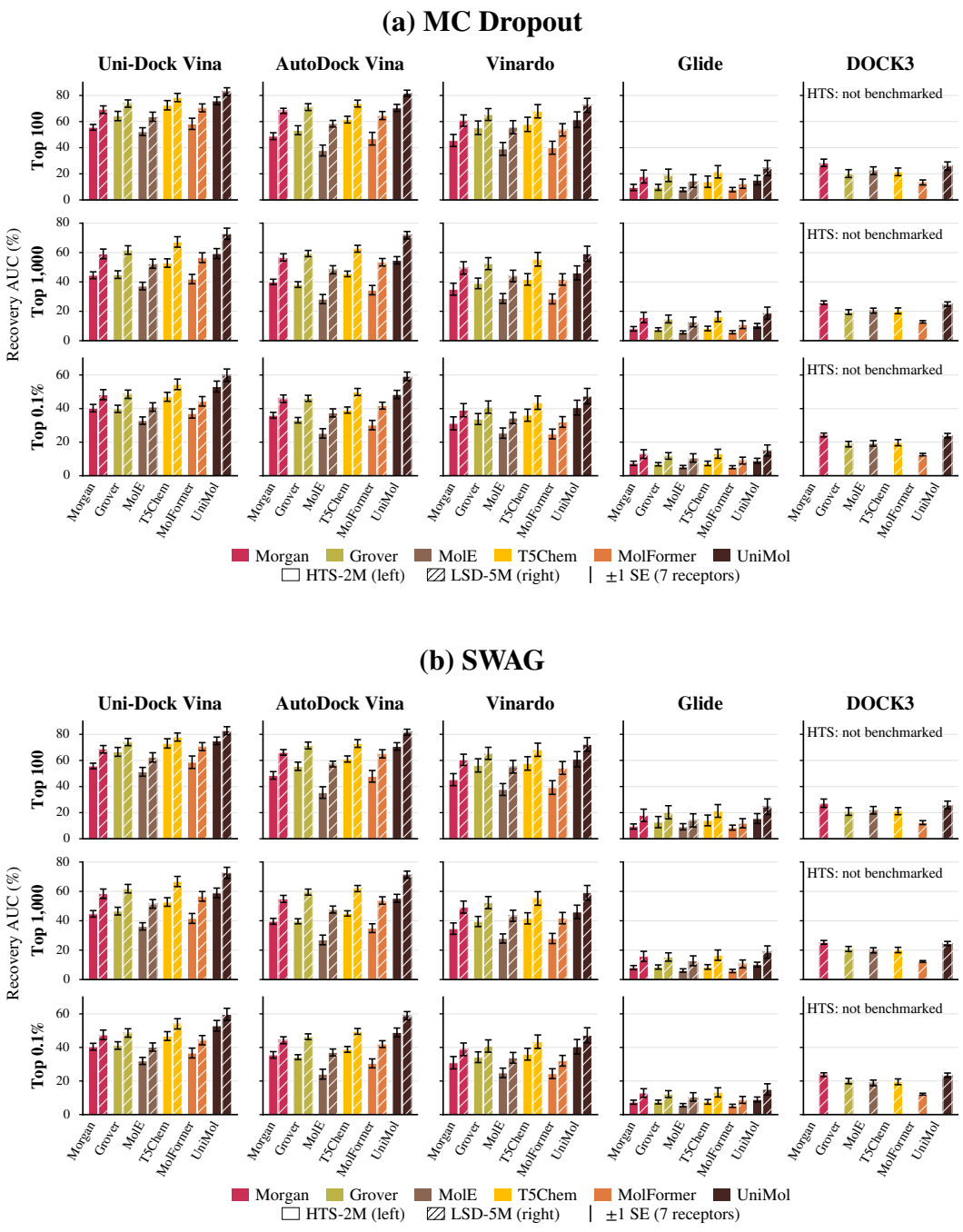


Figure A1: **Molecular representation performance across surrogate approximations.** Mean recovery AUC on HTS-2M (solid) and LSD-5M (hatched); higher is better. Error bars indicate ±1 standard error across seven receptors. The layout matches Figure 1 to facilitate comparison. **(a)** MC dropout. **(b)** SWAG.

A.1.2 Performance by Acquisition Functions

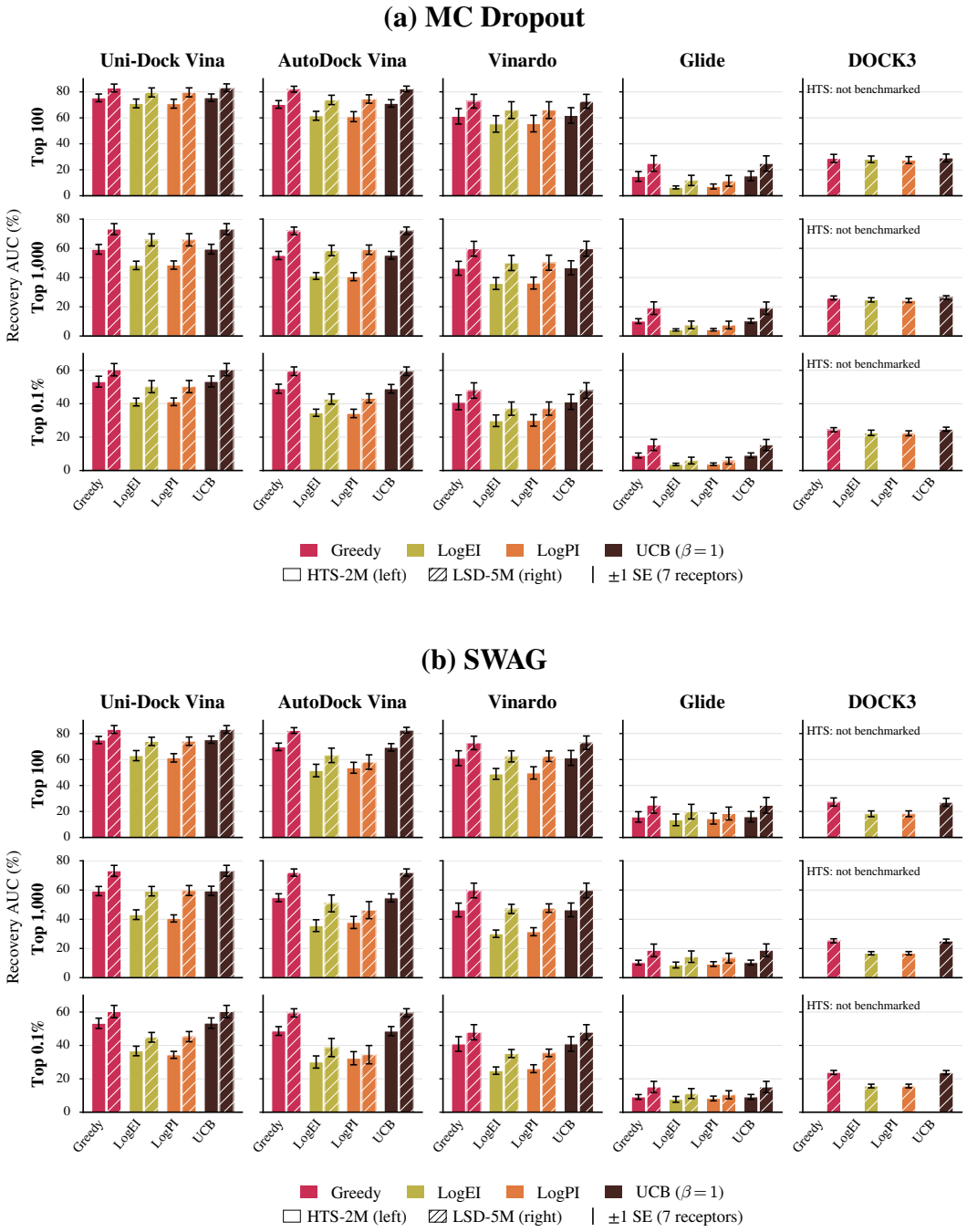


Figure A2: **Acquisition-function performance across surrogate approximations.** Mean seed-subtracted recovery AUC on HTS-2M (solid) and LSD-5M (hatched), with ±1 SE across seven receptors. Evaluation settings and molecular representations match Figure 3. Greedy and UCB remain the strongest acquisition strategies across both surrogate approximations. **(a)** MC dropout. **(b)** SWAG.

A.1.3 Optimization Campaigns Recovery Curves

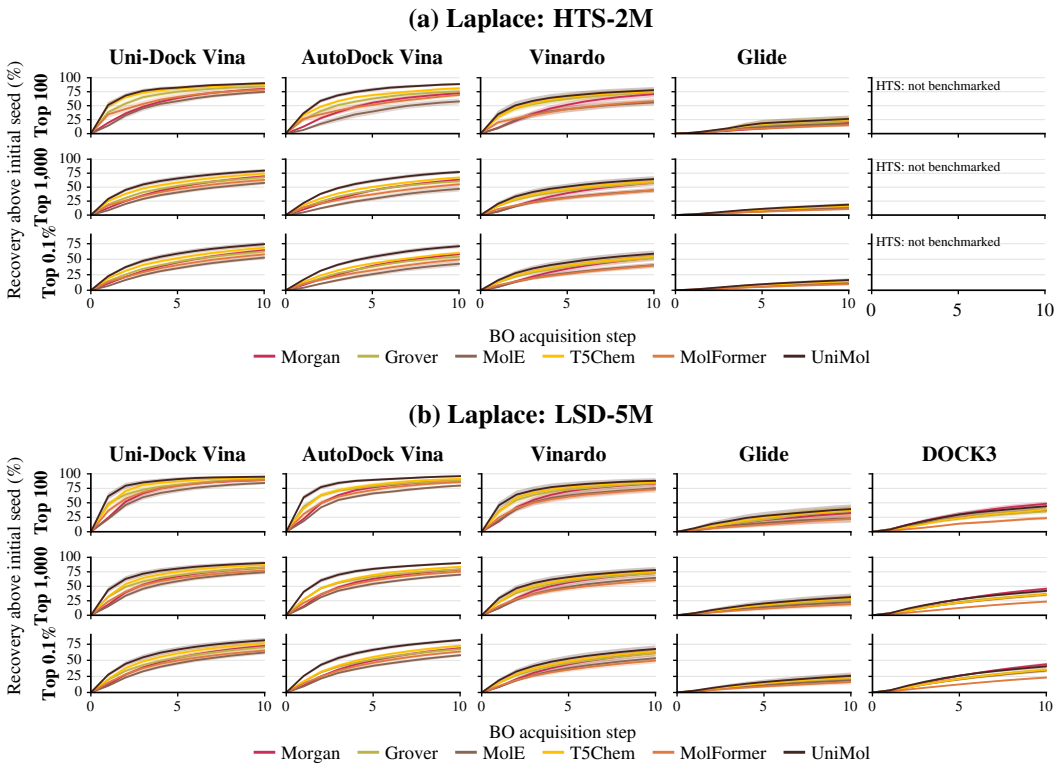


Figure A3: **Top-k recovery trajectories across molecular representations.** Recovery above the initial random seed is shown over ten acquisition rounds for **(a)** HTS-2M and **(b)** LSD-5M. Curves show means across seven receptors, with shaded regions indicating ± 1 standard error across receptor means. DOCK3 was evaluated only on LSD-5M. Results use the Laplace surrogate; corresponding MCDropout and SWAG results are shown in Figures A4.

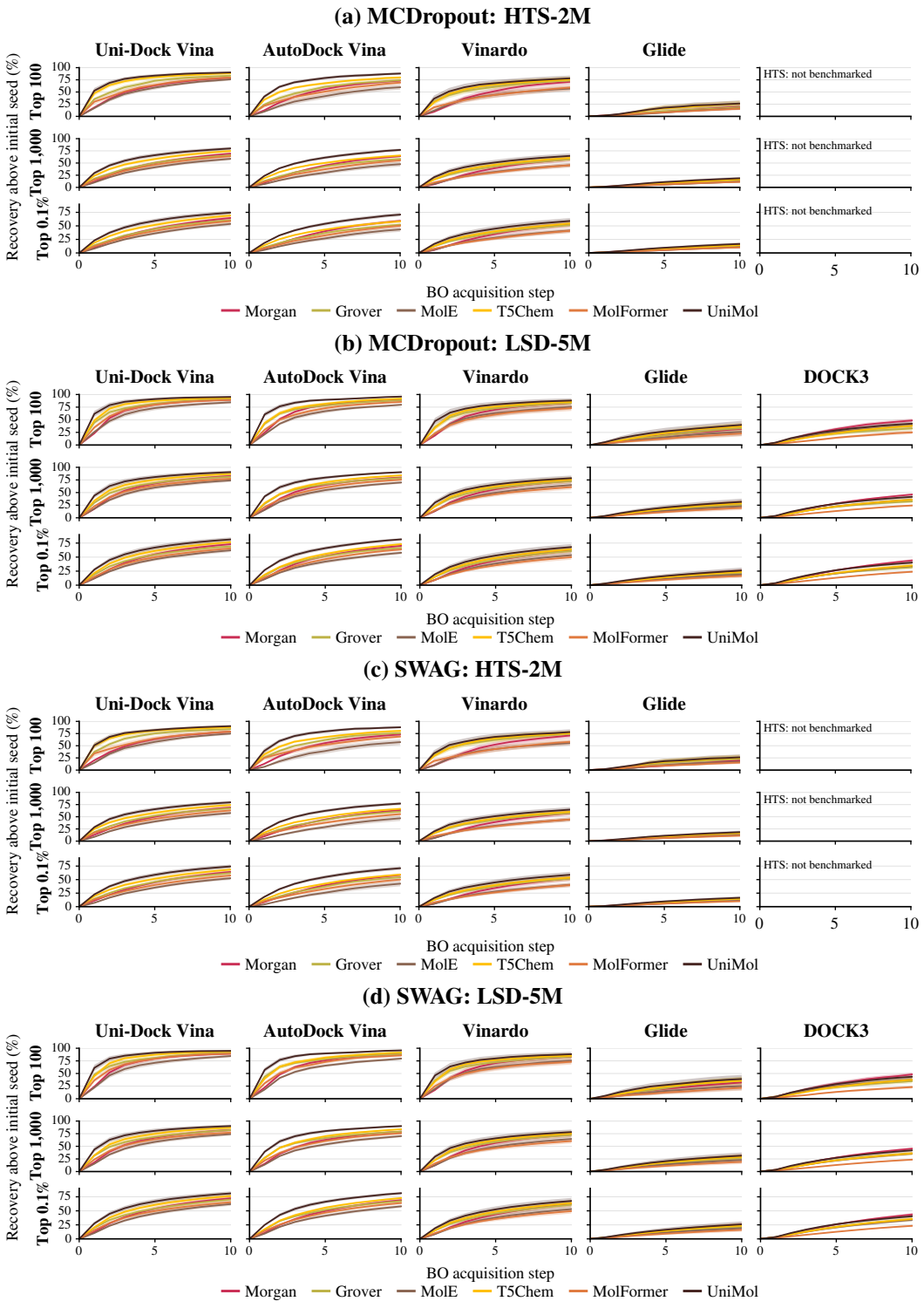


Figure A4: **Top-k recovery trajectories across molecular representations.** Results are shown for MCDropout and SWAG surrogates. See the caption of Figure A3 for further details.

A.1.4 Optimization Campaigns Cumulative Optima

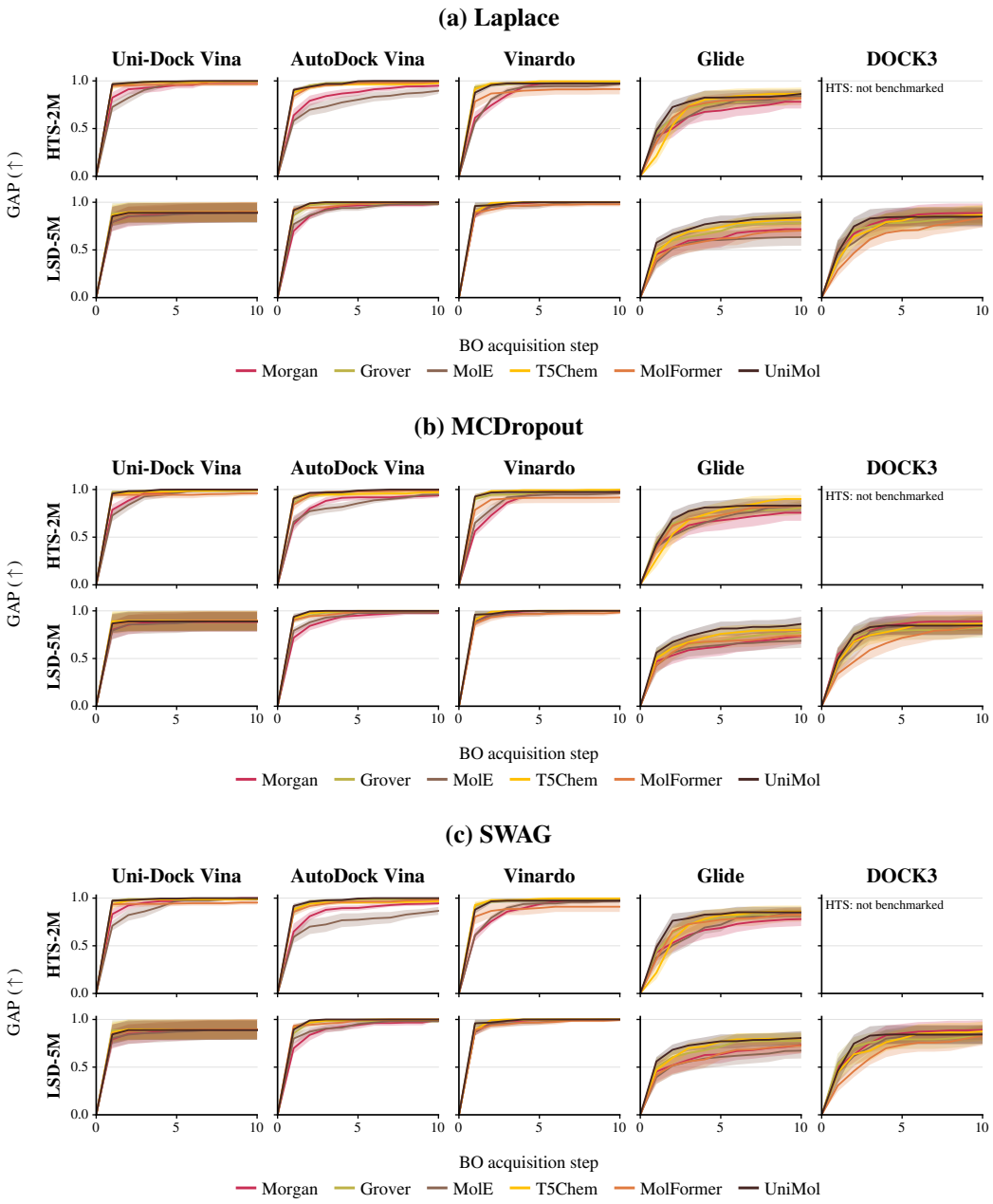


Figure A5: **Cumulative optimum performance across molecular representation, surrogate models, and screening libraries.** GAP (Jiang et al., 2020) measures the fraction of the initial seed-to-library optimum gap closed; higher values indicate better performance, and 1 denotes reaching the best score in the library. Curves average five replicate campaigns within each receptor, then average equally across seven receptors; shading indicates ± 1 standard error across receptor means. Step 0 denotes the initial random seed, and each subsequent BO step acquires 2,000 HTS or 5,000 LSD molecules. DOCK3 was evaluated only on LSD-5M.

A.1.5 Learnability and Smoothness of Docking Landscapes

Section 5.5 relates campaign performance to how learnable and how smooth each docking landscape is. This section defines the landscape descriptors, explains how they were computed, and reports the supporting analyses. All descriptors are measured on *random* samples of a library, independently of the optimization campaigns, so they describe the (library, docking oracle, receptor, representation) landscape itself rather than the trajectory of a particular campaign. We analyse HTS-2M and LSD-5M, the two libraries with campaign results, and the six molecular representations used in the campaigns, with the same embeddings.

Data and splits. For each library we fix one random ordering of all molecules. The first 20,000 molecules form a test set shared by every oracle, receptor, representation and training size. Training sets are drawn from the remaining molecules as nested random prefixes of three seeded orderings, so that for a given seed the 1k, 2k, 5k, 10k and 100k training sets contain one another. A molecule is used for an (oracle, receptor) pair only if its score is valid (Uni-Dock Vina, AutoDock Vina and Vinardo scores in $[-20, 0]$; Glide and DOCK3 scores ≤ 0) and every representation embedded it, so all representations are compared on identical molecules. Train and test molecules are therefore uniform random samples of the library (a random rather than a scaffold split). Continuous representations are standardised feature-wise with training-set statistics; Morgan fingerprints are kept as binary vectors.

Learnability. We train the last-layer Laplace BNN described in Appendix A.3.1 on each training set and evaluate it on the test set. Two settings differ from the campaigns because the training sets here are small: a batch size of 128 instead of 8,192, and a diagonal last-layer curvature approximation. We report the coefficient of determination on the test set $\mathcal{T}$,

$$R^2 = 1 - \frac{\sum_{i\in\mathcal{T}}(y_i - \hat{y}_i)^2}{\sum_{i\in\mathcal{T}}(y_i - \bar{y})^2}, \tag{1}$$

and, as a scale-free check, the Spearman rank correlation ρ between observed and predicted scores. Values for training sets of up to 10k molecules are averaged over the three seeds.

Scaling from 1k to 100k training molecules. Each campaign starts from 2,000 (HTS-2M) or 5,000 (LSD-5M) random molecules and adds ten batches of the same size, so its surrogate is trained on 2k–22k and 5k–55k molecules respectively. We measure learnability at 1k, 2k, 5k and 10k training molecules for every (library, oracle, receptor, representation) cell, and at 100k (one seed) for the two representations with the highest 10k learnability in each (library, oracle, receptor) combination. This range covers every training-set size a campaign surrogate sees. It lets us check two things: whether the ordering of oracles and representations measured at 10k persists at the largest campaign sizes, and whether a landscape that is hard to learn at 10k becomes learnable with more data or remains hard. All comparisons with campaign performance use the 10k values.

Distances. Smoothness descriptors need a distance between molecules. For Morgan fingerprints we use the Tanimoto distance

$$d_{\mathrm{T}}(x_i, x_j) = 1 - \frac{|x_i \wedge x_j|}{|x_i \vee x_j|}, \tag{2}$$

where $|x_i \wedge x_j|$ and $|x_i \vee x_j|$ count the bits set in both and in either fingerprint. For all other representations we use the Euclidean distance between standardised feature vectors. Scores are standardised on the sample, $z_i = (y_i - \bar{y})/s_y$.

Normalised k-NN Lipschitz slope. On the 10,000 training molecules of each cell, distances are divided by the median distance of 2×10^5 random pairs from the same set, so that slopes are comparable across representations with different distance scales:

$$\tilde{d}(i, j) = \frac{d(x_i, x_j)}{\mathrm{median}_{(a,b)\sim\mathrm{random}}\, d(x_a, x_b)}. \tag{3}$$

For every molecule i and each of its $k = 10$ nearest neighbours $j \in \mathcal{N}_k(i)$ we compute the local slope

$$\ell_{ij} = \frac{|z_i - z_j|}{\tilde{d}(i, j)}, \qquad \tilde{d}(i, j) > 0, \tag{4}$$

and report the median $\tilde{L}_{50}$ and the 95th percentile $\tilde{L}_{95}$ over all nk slopes. The Lipschitz constant $\sup_{i \neq j} |z_i - z_j| / \tilde{d}(i,j)$ is dominated by a few near-duplicate pairs with inconsistent scores, whereas its quantiles are robust. $\tilde{L}_{50}$ describes a typical neighbour pair and $\tilde{L}_{95}$ the steepest local changes. Lower values mean a smoother landscape; $\tilde{L}_{50} \approx 1$ means that neighbours differ in score as much per unit of distance as random pairs do. Because Tanimoto and standardised Euclidean distances are normalised separately, slopes are compared across representations only up to this normalisation.

k-NN regression R^2. As a model-free measure of smoothness, each molecule's score is predicted by the mean score of its $k = 10$ nearest neighbours (same distances as above):

$$R^2_{k\text{NN}} = 1 - \frac{\sum_i \big(z_i - \frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} z_j\big)^2}{\sum_i z_i^2}. \tag{5}$$

Higher values mean that nearby molecules are informative about each other's scores. Because $R^2_{k\text{NN}}$ is scale-free, it is directly comparable across representations, including Morgan fingerprints with Tanimoto distance.

ROGI-XD. We follow Graff et al. (2023) and the reference implementation. On the first 5,000 training molecules of each cell, scores are min–max scaled to $[0, 1]$ and molecules are clustered by complete-linkage agglomerative clustering, using Tanimoto distance for Morgan fingerprints and, for the other representations, Euclidean distance on the raw features divided by an estimate of the maximum distance. At each merge threshold t (thresholds closer than 0.01 are skipped) we record the number of clusters $N_c(t)$ and the dispersion of the cluster means,

$$\sigma(t) = 2\sqrt{\frac{\sum_c n_c \, (\bar{y}_c - \bar{y})^2}{\sum_c n_c}}, \qquad \xi(t) = 1 - \frac{\log N_c(t)}{\log N}, \tag{6}$$

and define

$$\text{ROGI-XD} = \sigma(0) - \int_0^1 \sigma \, \mathrm{d}\xi, \tag{7}$$

evaluated with the trapezoid rule. Lower values mean a smoother landscape. Uncertainty is estimated from twenty 80% subsamples drawn without replacement; resampling with replacement introduces zero-distance duplicates that bias ROGI-XD downwards.

Relating descriptors to campaign performance. Descriptors are averaged over the seven receptors, matching the averaging of the recovery AUC, which gives one point per (library, oracle, representation) combination: 4 oracles $\times$ 6 representations on HTS-2M and 5 $\times$ 6 on LSD-5M, 54 points in total. The figures report Pearson r within each representation (across oracles and libraries). Because the docking oracle itself accounts for much of the variation in both descriptors and AUC, we also ask whether a descriptor ranks the representations of the *same* landscape correctly. For each group $g = (\text{library}, \text{oracle})$ we rank the descriptor D and the AUC A within the group, centre the ranks, and correlate them:

$$\rho_{\text{within}} = \text{corr}\big(r_g(D_i) - \bar{r}_g, \; r_g(A_i) - \bar{r}_g\big). \tag{8}$$

Significance uses two-sided permutation tests that shuffle within groups (2,000 permutations); 95% intervals come from 2,000 bootstrap resamples of representations within each group.

Where the landscape is learnable. The docking oracle sets the level of learnability far more than the representation (Figure A6). With 10k training molecules, the median test R^2 over representations and receptors is 0.67 (AutoDock Vina) and 0.66 (Uni-Dock Vina) on HTS-2M, against 0.44 for Vinardo and 0.20 for Glide; on LSD-5M it is 0.58–0.59 for the two Vina oracles, 0.43 for Vinardo, 0.30 for DOCK3 and 0.25 for Glide. Within every oracle column UniMol gives the most learnable landscape, and MolFormer and MolE the least.

Scaling with training-set size. Learnability increases steadily from 1k to 10k training molecules for every oracle and representation (median gain in test R^2 of 0.13; Figure A7), and the gaps between oracles are already fixed at 1k. On the 156 cells trained with 100k molecules, test R^2 increases

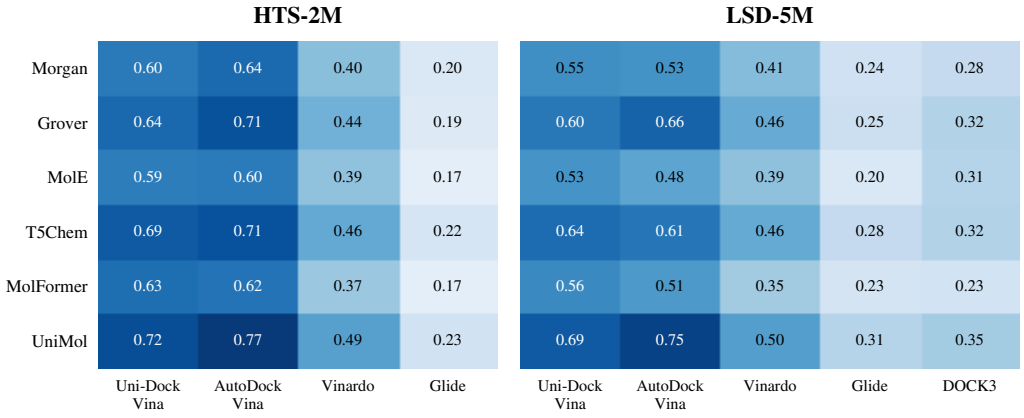


Figure A6: **Learnability of each docking landscape at 10k training molecules.** Test R^2 of the last-layer Laplace BNN on the shared 20k random test set, averaged over three training seeds and the seven receptors, for each representation (rows) and docking oracle (columns). Darker cells are more learnable. DOCK3 scores are available only for LSD-5M.

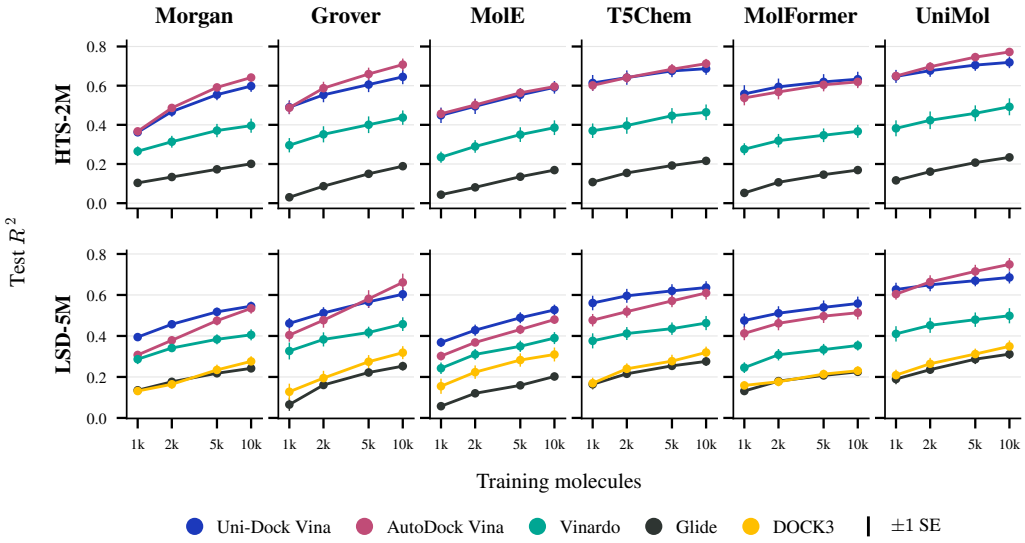


Figure A7: **Learning curves from 1k to 10k training molecules.** Test R^2 of the last-layer Laplace BNN for each library (rows), representation (columns) and docking oracle (colours). Each point is the mean over the seven receptors of the seed-averaged test R^2; vertical bars indicate ± 1 SE over receptors. Training sets are nested, so each curve adds molecules to the same training set.

by a further median of 0.07 over the same cells at 10k, and increases in 98% of them (Figure A8). The ordering of oracles is unchanged across the whole range: at 100k, AutoDock Vina reaches 0.78 (HTS-2M) and 0.81 (LSD-5M), while Glide reaches only 0.29 and 0.37. More data therefore improves every landscape but does not make a hard oracle easy within the training-set sizes a campaign sees, which supports using 10k learnability as a descriptor of each oracle.

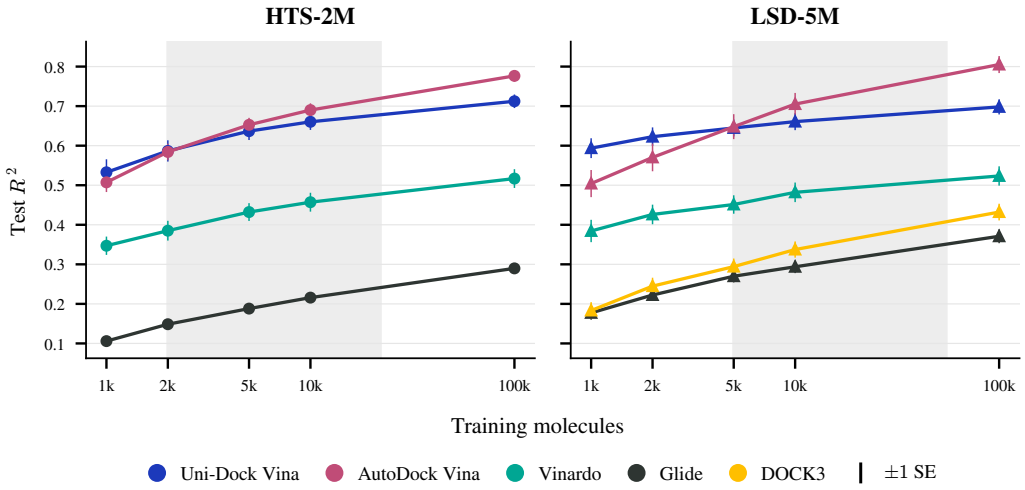


Figure A8: **Learnability from 1k to 100k training molecules.** Test R^2 of the last-layer Laplace BNN on the 156 (library, oracle, receptor, representation) cells that were also trained with 100k molecules (for each library, oracle and receptor, the two representations with the highest test R^2 at 10k). Each point is the mean over these cells, using the same cells at every training-set size; vertical bars indicate ± 1 SE over cells. Values up to 10k are averaged over three seeds; 100k values use one seed. The grey band marks the training-set sizes seen by campaign surrogates (2k–22k on HTS-2M, 5k–55k on LSD-5M).

Smoothness explains learnability. The model-free k-NN R^2 tracks the BNN's test R^2 almost perfectly within each representation (Pearson $r = 0.92$–0.99; $r = 0.94$ over all 378 cells; Figure A9), so the BNN learns a landscape about as well as its local neighbourhoods predict it. The Lipschitz slope $\tilde{L}_{50}$ is also strongly related within each representation ($r = -0.59$ to -0.93) but only weakly across representations ($r = -0.40$), because Tanimoto and Euclidean slopes are normalised on different scales. ROGI-XD is moderately related ($r = -0.51$ to -0.64).

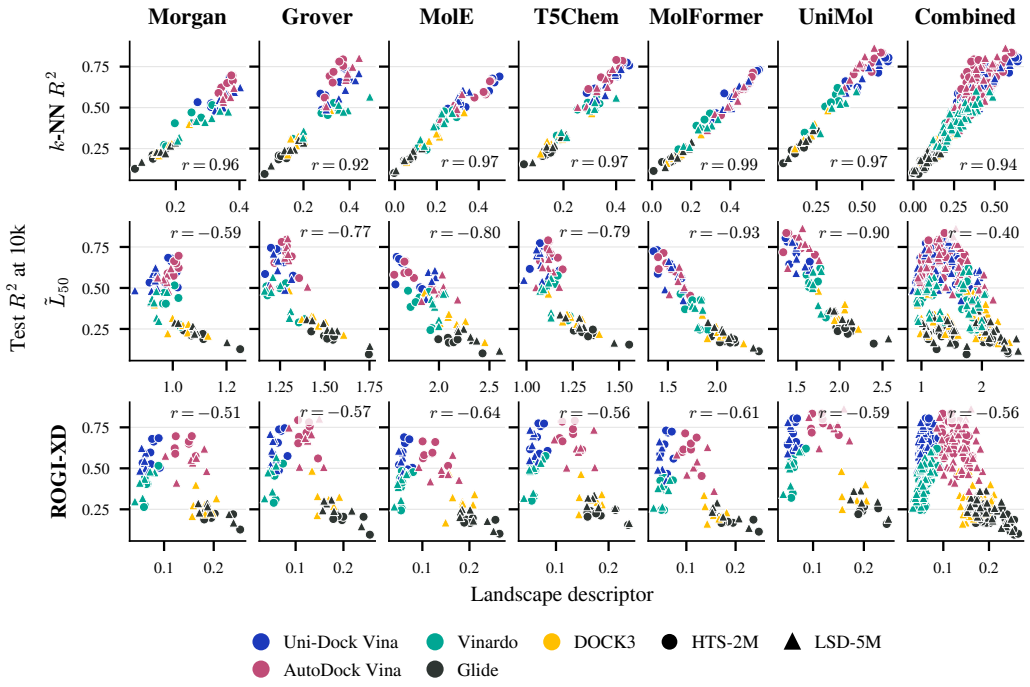


Figure A9: **Landscape smoothness and learnability.** Test R^2 of the last-layer Laplace BNN at 10k training molecules against three smoothness descriptors computed on the same random training molecules: k-NN regression R^2 (top; higher is smoother), the normalised k-NN Lipschitz slope $\tilde{L}_{50}$ (middle; lower is smoother) and ROGI-XD (bottom; lower is smoother). Each point is one (library, oracle, receptor, representation) cell (378 in total). Pearson correlations (r) are calculated within each representation and, in the last column, over all cells.

Descriptors and campaign performance. Figures A10, A11 and A12 plot the recovery AUC of the Laplace campaigns against each descriptor, and Table A1 gives the pooled and within-oracle rank correlations. Learnability is the strongest predictor of campaign performance. Test Spearman ρ at 10k correlates with AUC at $r = 0.91$–0.94 over all 54 points (Figure A10), slightly more than test R^2 ($r = 0.85$–0.88; Figure 4). It is also the best descriptor for ranking the representations of a single landscape ($\rho_{\text{within}} = 0.87$–$0.89$). A campaign selects molecules by their predicted rank, so a rank-based measure of the surrogate may be expected to fit slightly better. The model-free k-NN regression R^2 follows closely ($r = 0.83$–0.87; Figure A11b) and also ranks representations correctly within an oracle, but less reliably ($\rho_{\text{within}} = 0.59$–$0.60$), because fixed neighbourhoods do not reweight the features as a trained surrogate does. The two roughness descriptors behave in opposite ways. ROGI-XD has a strong pooled correlation with AUC ($r = -0.74$ to -0.79; Figure A11a), but this correlation comes entirely from differences between oracles: GLIDE and DOCK3 landscapes are rough and the Vina-family landscapes are smooth for every representation. Within an oracle, ROGI-XD does not rank the representations, and its coefficient has the unfavourable sign ($\rho_{\text{within}} = 0.16$–$0.23$, not significant). The Lipschitz slopes show the reverse pattern. Their pooled correlation is weak ($r = -0.41$ to -0.47; Figure A12) because Tanimoto and Euclidean slopes are normalised on different scales, yet within each representation they are closely related to AUC ($r = -0.46$ to -0.98 for $\tilde{L}_{50}$). Within an oracle, $\tilde{L}_{50}$ ranks the representations significantly at all three top-k levels ($\rho_{\text{within}} = -0.31$ to -0.48). Descriptors that measure how well a molecule's score can be predicted, from a trained model or from its neighbours, therefore carry information about both the oracle and the representation, whereas ROGI-XD carries information only about the oracle. These correlations are descriptive: the 54 points share libraries, oracles and representations, and the within-oracle analysis controls only for the oracle.

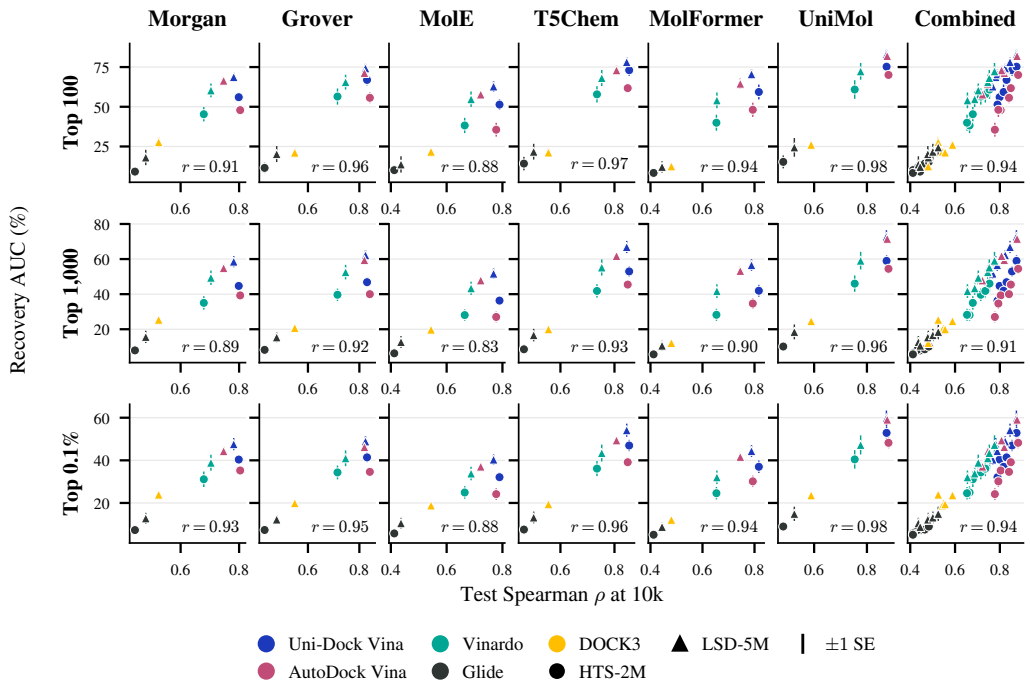


Figure A10: **Ranking accuracy and BO performance across molecular representations.** Recovery AUC of the Laplace campaigns at the top-100, top-1,000 and top-0.1% thresholds (rows) against the test Spearman ρ of the last-layer Laplace BNN trained on 10k random molecules (averaged over seeds and receptors). Vertical error bars indicate ± 1 SE of AUC. Pearson correlations (r) are calculated across both libraries within each panel. DOCK3 results are available only for LSD-5M.

Table A1: **Rank correlation between landscape descriptors and campaign AUC** (Laplace BNN, 54 (library, oracle, representation) combinations). Test R^2 and test ρ are those of the last-layer Laplace BNN at 10k training molecules; k-NN R^2, $\tilde{L}_{50}$ and $\tilde{L}_{95}$ are the k-NN regression R^2 and normalised k-NN Lipschitz slopes. *Within oracle*: ρ_{within} (Eq. 8), which ranks representations against each other inside each (library, oracle) pair, with 95% bootstrap intervals. *Pooled*: Spearman ρ over all 54 points. Column headers give the top-k threshold of the recovery AUC.

		Within-oracle ρ_{within}						Pooled ρ		
Descriptor	Fav.	100		1,000		0.1%		100	1,000	0.1%
Test R^2	↑	0.86^{***}	[0.75, 0.94]	0.88^{***}	[0.77, 0.95]	0.88^{***}	[0.77, 0.95]	0.85^{***}	0.82^{***}	0.86^{***}
Test ρ	↑	0.87^{***}	[0.76, 0.94]	0.89^{***}	[0.79, 0.96]	0.89^{***}	[0.79, 0.96]	0.88^{***}	0.85^{***}	0.89^{***}
k-NN R^2	↑	0.59^{***}	[0.40, 0.74]	0.60^{***}	[0.41, 0.74]	0.59^{***}	[0.41, 0.74]	0.84^{***}	0.82^{***}	0.85^{***}
$\tilde{L}_{50}$	↓	-0.31^{*}	[−0.53, −0.06]	-0.47^{**}	[−0.66, −0.25]	-0.48^{**}	[−0.65, −0.26]	-0.39^{**}	-0.39^{**}	-0.42^{**}
$\tilde{L}_{95}$	↓	−0.30	[−0.55, −0.02]	-0.43^{**}	[−0.64, −0.20]	-0.43^{**}	[−0.64, −0.21]	-0.42^{**}	-0.41^{**}	-0.43^{**}
ROGI-XD	↓	0.16	[−0.06, 0.37]	0.20	[−0.04, 0.43]	0.23	[−0.03, 0.44]	-0.70^{***}	-0.70^{***}	-0.68^{***}

A more learnable or smoother landscape should give a positive coefficient for descriptors marked ↑ and a negative coefficient for those marked ↓. Stars: two-sided permutation tests with 2,000 permutations (within-oracle tests shuffle inside each (library, oracle) pair); $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$. The permutation test and the bootstrap interval can disagree near the threshold, e.g. $\tilde{L}_{95}$ at top-100 ($p = 0.06$, interval excluding zero). Correlations in Figures 4, A10, A11 and A12 are Pearson r and therefore differ slightly from the pooled values here.

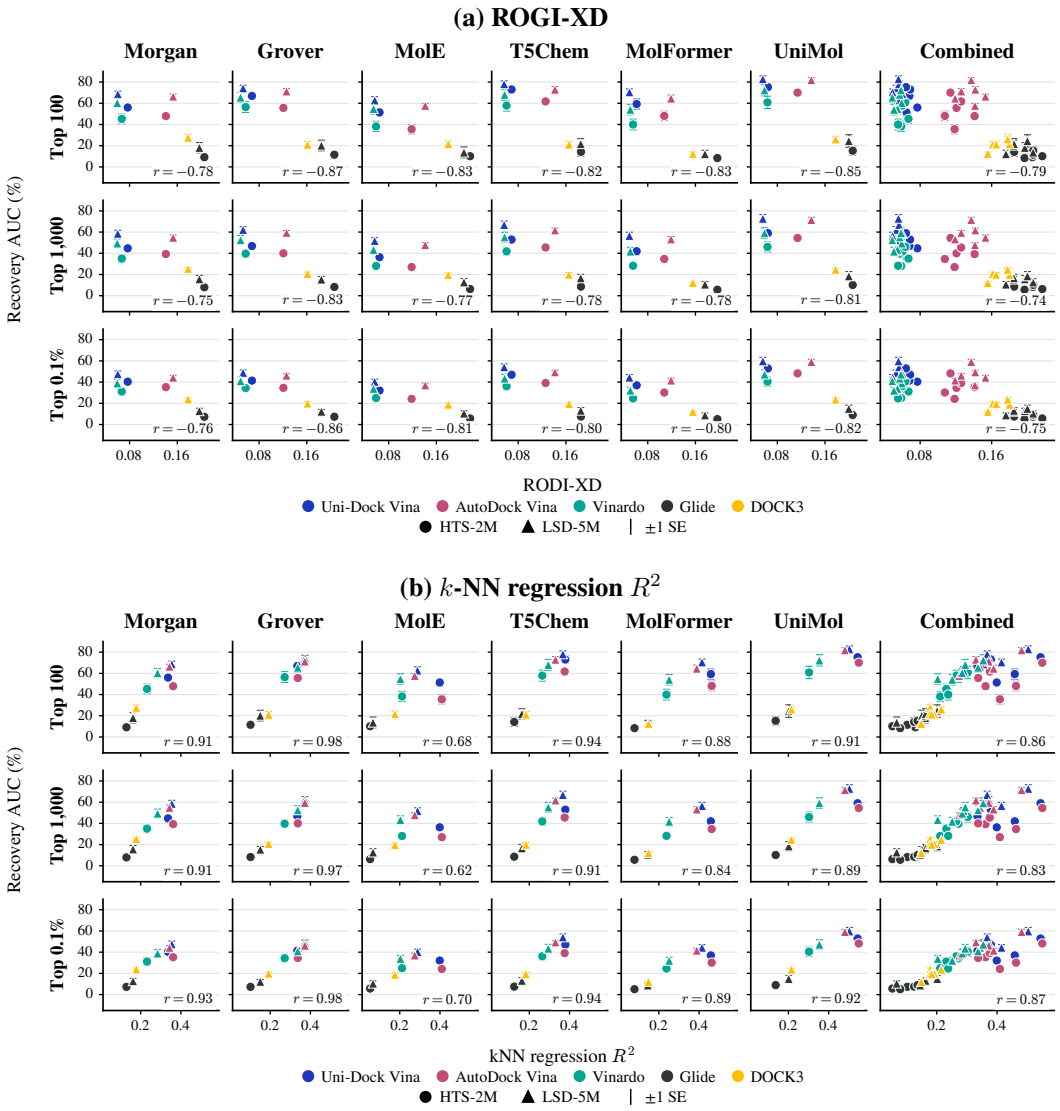


Figure A11: **Landscape roughness and BO performance.** Recovery AUC of the Laplace campaigns against (a) ROGI-XD (lower is smoother) and (b) k-NN regression R^2 (higher is smoother), both computed on random molecules of each library and averaged over the seven receptors. Vertical error bars indicate ± 1 SE of AUC. Pearson correlations (r) are calculated across both libraries within each panel. The pooled ROGI-XD correlation reflects differences between oracles; within an oracle, ROGI-XD does not rank representations (Table A1).

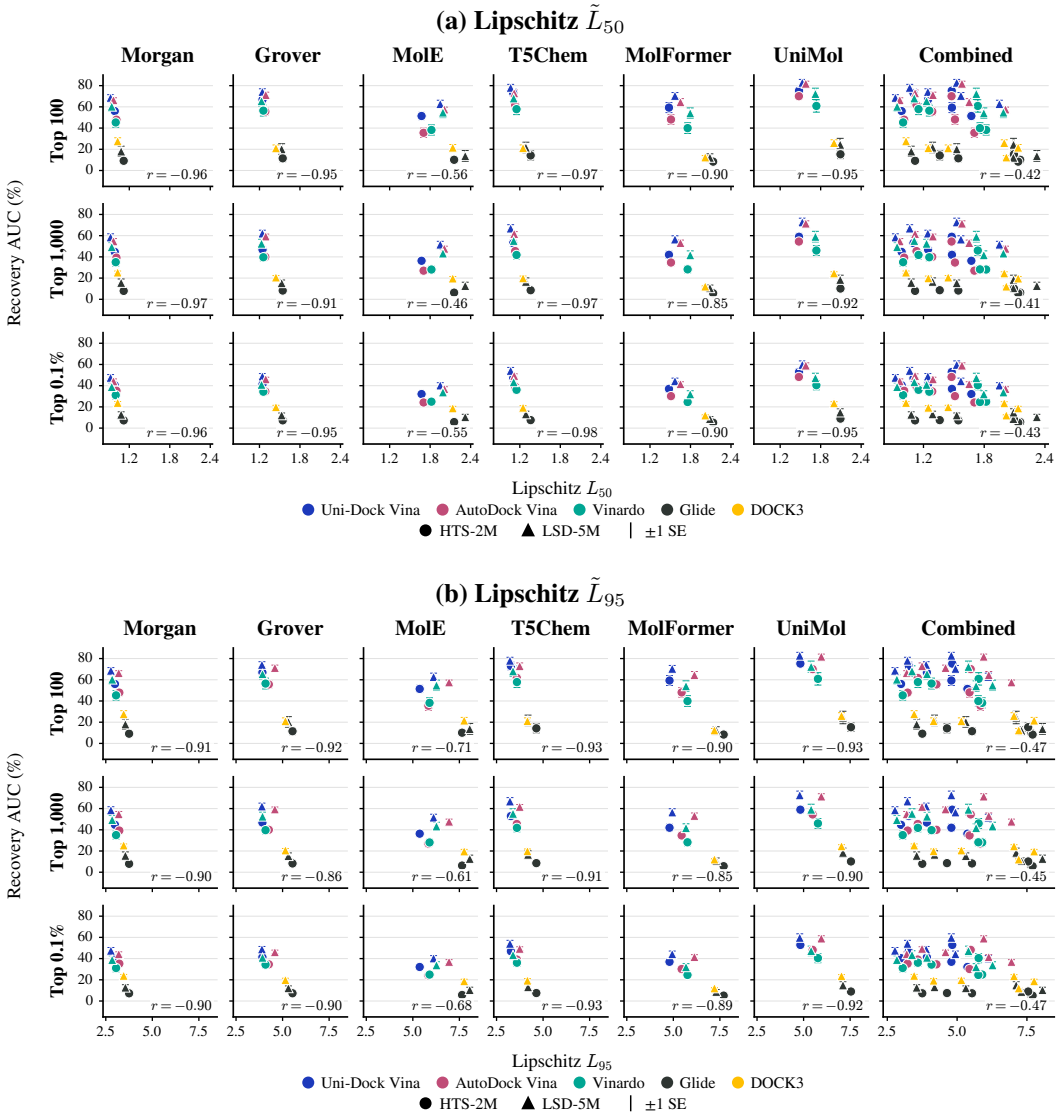


Figure A12: **Local Lipschitz slopes and BO performance.** Recovery AUC of the Laplace campaigns against the median (a) and 95th percentile (b) of the normalised k-NN Lipschitz slope (Eq. 4; lower is smoother), computed on random molecules of each library and averaged over the seven receptors. Vertical error bars indicate ± 1 SE of AUC. Pearson correlations (r) are calculated across both libraries within each panel. Tanimoto (Morgan) and Euclidean slopes are normalised separately, so the Combined column mixes two scales.

A.1.6 Optimization Campaigns with Boltz Co-Folding

We additionally evaluate optimization performance on LSD-10K using docking scores from AutoDock Vina, Uni-Dock Vina, Uni-Dock Vinardo, and Boltz-2 co-folding. In addition to the six molecular representations considered in the main benchmark, we include two representation families derived from Boltz-2.

First, we evaluate BoltzFM, which extracts transferable, receptor-independent small-molecule representations from Boltz-2 for downstream molecular prediction tasks (Jang et al., 2026). Second, we construct a receptor-conditioned representation directly from intermediate Boltz-2 embeddings (see Appendix A.2.3 for more details on co-folding with Boltz-2). For each protein–ligand pair, we extract the z_{pl}, z_{lp}, and z_{ll} representations, mean-pool each tensor across its protein and ligand dimensions to obtain three 128-dimensional vectors, and concatenate them into a single 384-dimensional representation. We refer to this representation as *Receptor-Boltz*.

Figure A13 compares recovery performance at the top-10 and top-100 thresholds. BoltzFM performs worst among the evaluated representations across all four docking oracles. In contrast, Receptor-Boltz is competitive with UniMol for the Vina-family docking scores and substantially outperforms the other representations when optimizing Boltz-2 co-folding scores.

These results suggest that the utility of co-folding representations depends strongly on how interaction context is incorporated. A standalone Boltz-derived molecular embedding does not improve optimization performance in this setting, whereas representations extracted from intermediate protein–ligand interaction states transfer competitively to conventional docking objectives and are particularly effective for predicting the co-folding oracle from which they are derived.

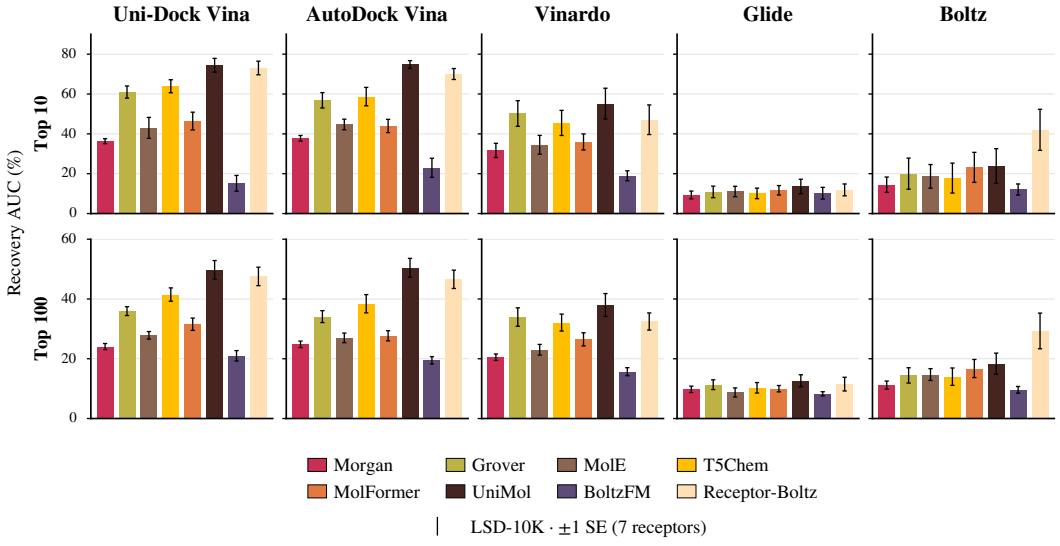


Figure A13: **Molecular representation performance on LSD-10K.** Mean recovery AUC on Uni-Dock Vina, AutoDock Vina, Vinardo, Glide, and Boltz-2 co-folding; higher is better. Error bars indicate ± 1 standard error across seven receptors.

A.1.7 Optimization Campaigns of Enamine-10M

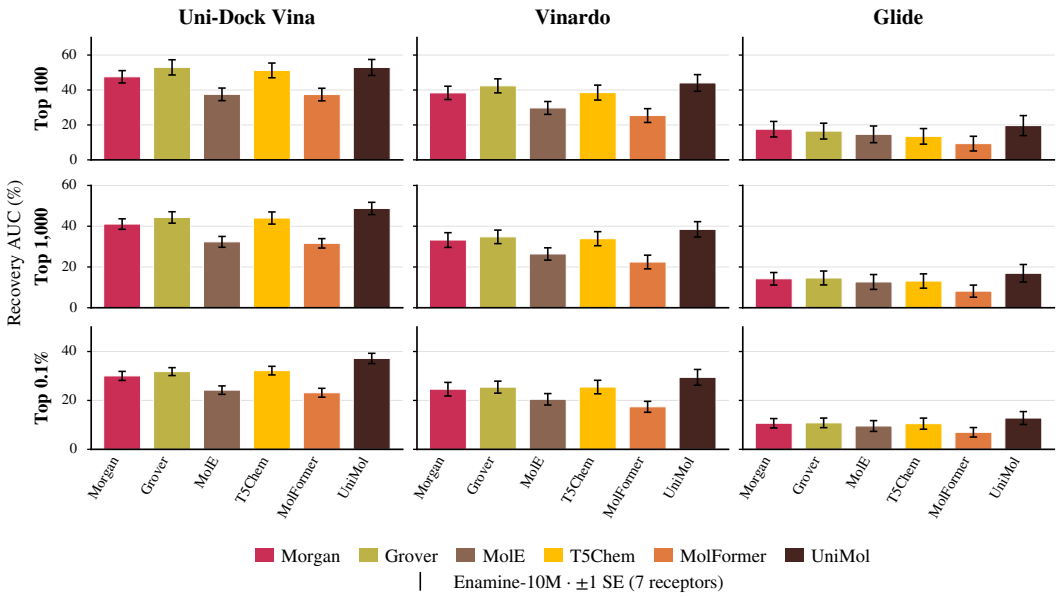


Figure A14: **Molecular representation performance on Enamine-10M.** Mean recovery AUC on Uni-Dock Vina, Vinardo, and Glide; higher is better. Error bars indicate ±1 standard error across seven receptors.

A.2 DATASETS

A.2.1 VIRTUAL LIBRARIES

LARGE SCALE DOCKING (LSD) The LSD-5M dataset, and the LSD-10K subset, is a random sample from the 96M compound library introduced by Lyu et al. (2019). This dataset is a curated set from molecules in what is normally considered biologically relevant chemical space.

HIGH-THROUGHPUT SCREENING (HTS) The HTS-2M list of SMILES is the high-throughput virtual screening benchmarking library created by Graff et al. (2021). This library has a slightly less narrow distribution of physiochemical properties compared to LSD-5M.

ENAMINE-10M The ENAMINE-10M is a random 10M compound dataset drawn from the ultra-large Enamine REAL space (Grygorenko et al., 2020). This dataset maintains the broad distribution of properties of the combinatorial space, which is a realistic setting for sample-efficient virtual screening.

LIBRARY OVERLAP The HTS-2M and LSD-5M share 23,648 identical molecules, and ENAMINE-10M shares at most 196 molecules with HTS-2M or LSD-5M.

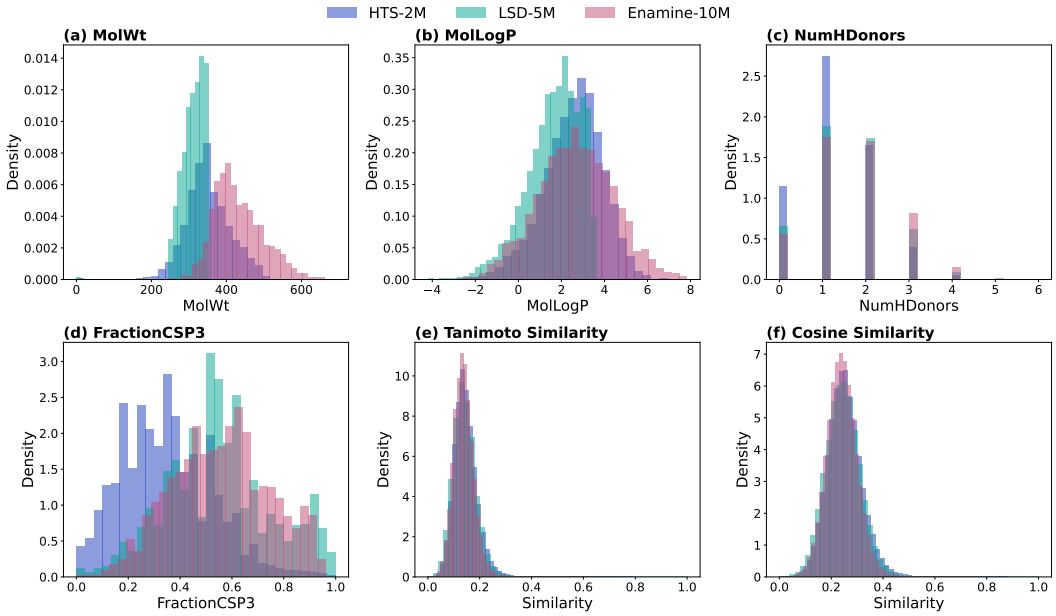


Figure A15: **Physiochemical property distributions and chemical diversity across the benchmark libraries.** (a-d) Histograms of molecular weight, logarithm of the partition coefficient, number of hydrogen bond donors, the sp^3 fraction calculated using RDKit for the HTS-2M, LSD-5M, and ENAMINE-10M datasets. (e and f) Chemical diversity illustrated by Tanimoto similarity and Cosine similarity over Morgan fingerprints for random subsets of 2,000 molecules of the benchmark libraries. The HTS-2M datasets has a more narrow distribution of properties compared to the ENAMINE-10M dataset, as it is a curated set. On the other hand, the ENAMINE-10M dataset is a random selection from the ultra-large Enamine REAL space, which covers a larger portion of chemical space, which is more accurate for representing real-world sample-efficient virtual screening campaigns.

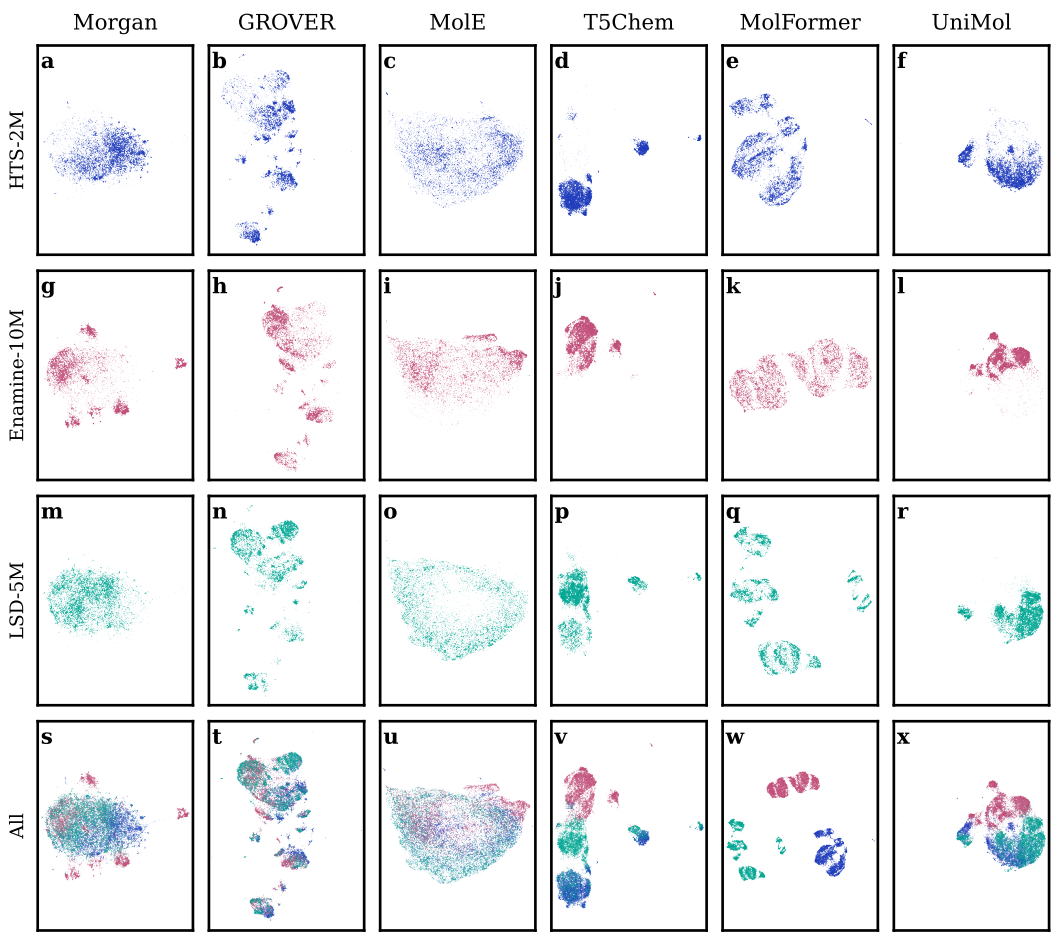
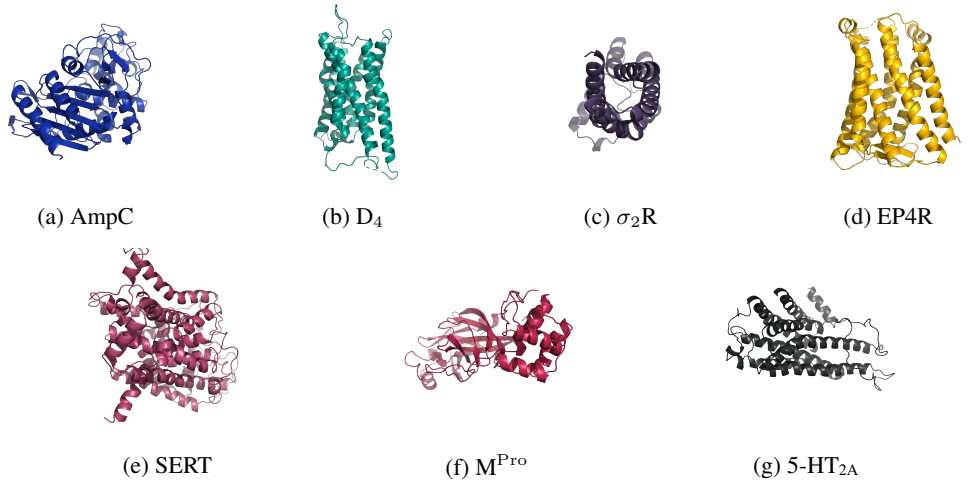


Figure A16: **UMAP visualizations of molecular representations across three screening libraries.** Rows show HTS-2M, ENAMINE-10M, LSD-5M, and an overlay of all three libraries; columns show Morgan fingerprints, GROVER, MolE, T5Chem, MolFormer, and UniMol-v2 (84M). For each library, we uniformly sample 10,000 molecules without replacement, using the same sampled molecules across representations. For each representation, features from the three libraries are jointly embedded in a shared UMAP space. Learned embeddings are standardized per feature using statistics fit jointly across the three library samples and projected using cosine distance; binary Morgan fingerprints remain unscaled and use Tanimoto/Jaccard distance. UMAP is fit once per representation using the combined 30,000-molecule sample, with 15 neighbors and a minimum distance of 0.1. Consequently, the three library-specific panels within each column share the same coordinate system and axis limits, enabling direct comparison of their distributions in representation space; the bottom row overlays all three libraries in this shared embedding.

A.2.2 PROTEIN RECEPTORS

Table A2: Protein targets used in the virtual screening benchmarks.

Receptor	PDB ID	Description	Source
AmpC	1L2S	Class C β-lactamase that hydrolyzes β-lactam antibiotics and confers antibiotic resistance.	Lyu et al. (2019)
D_4	5WIU	Dopamine G protein-coupled receptor involved in neuronal signaling and modulation of cognition, emotion, and behavior.	
σ_2R	7M95	Transmembrane protein involved in cholesterol homeostasis, membrane trafficking, autophagy, and regulation of cell growth.	Alon et al. (2021)
EP4R	5YWY	Prostaglandin E_2 receptor that activates G_s-mediated signaling and regulates inflammatory and physiological responses.	Gahbauer et al. (2023)
SERT	6DZZ	Membrane transporter that clears serotonin from the extracellular space and thereby regulates serotonergic signaling.	Singh et al. (2023)
M^{Pro}	7KEY	SARS-CoV-2 cysteine protease that cleaves viral polyproteins to generate proteins required for viral replication.	Fink et al. (2023)
5-HT_{2A}	8UWL	Serotonin G protein-coupled receptor that mediates serotonergic signaling involved in perception, cognition, and other neural processes.	Lyu et al. (2024)



(a) AmpC (b) D_4 (c) σ_2R (d) EP4R

(e) SERT (f) M^{Pro} (g) 5-HT_{2A}

Figure A17: **Protein targets in the CHEMVERSE benchmark.** Renderings of AmpC, D_4, σ_2R, EP4R, SERT, M^{Pro}, and 5-HT_{2A} generating using PyMOL (Schrödinger, LLC, 2021) to highlight structural diversity across the targets. The targets range from deep, enclosed pockets to more solvent-exposed pockets that the active learning models must navigate across different oracles.

A.2.3 DOCKING WORKFLOWS

Our high-throughput docking workflow was designed to standardize evaluations across multiple docking engines and manage the complexities of generating a million-scale ground-truth dataset. For each protein target, receptor structures were prepared and active sites defined according to the specific requirements of each docking engine. 3D conformers were generated prior to docking where necessary. All of these structures were docked against each receptor to generate a "ground-truth" dataset from which the active learning benchmarks can instantaneously look up scores as needed. Docking failures are excluded separately for each oracle–target–library combination. Molecules without a valid score are removed from the corresponding candidate library, reducing its size relative to the nominal library. All acquisition budgets and recovery metrics are computed relative to this effective pool, so that cross-oracle comparison may involve slightly different candidate sets. Protein structures with associated ligands were retrieved from the LSD database and prepared using the Schrödinger software suite. The structures were inspected for the presence of cofactors, metal ions, modified residues, and missing side chains or residues. All cofactors and metal ions were removed prior to docking. Modified residues, when present, were reverted to their corresponding canonical amino acids. Missing side chains and residues were modeled using the Prime module of Schrödinger, guided by the corresponding UniProt protein sequences. Hydrogen atoms were added at a target pH of 7.2, hydrogen-bonding networks were optimized, and the resulting structures were subjected to restrained energy minimization to resolve unfavorable steric contacts while preserving the overall protein conformation. The ligand associated with each protein structure was used to define the binding-site center for Glide receptor-grid generation. Ligand molecules were docked into the generated receptor grids using the Glide High-Throughput Virtual Screening (HTVS) protocol. The same prepared and energy-minimized protein structures were subsequently converted to PDBQT format and used as receptors for molecular docking with AutoDock Vina. The binding-site coordinates defined from the corresponding experimentally associated ligands were used to define the docking region for both Glide and Vina, enabling docking against comparable binding sites across the two methods.

DISTRIBUTED INFRASTRUCTURE Given the large scale of the benchmark comprising nearly 200M docking evaluations a custom distributed computing infrastructure was developed. We sharded the virtual library structures into localized chunks and deployed a fleet of custom Python orchestration scripts to dynamically dispatch work across multiple computing nodes via SSH and rsync. To ensure fault tolerance and prevent data corruption in this concurrent environment, all shard docking was coordinated centrally through a thread and multiprocess-safe process via file-locking and atomic file operations. Dynamic micro-chunking and an automated bisection protocol isolate and skip poison ligands that lead to CUDA illegal memory accesses.

GLIDE For GLIDE docking, 3D ligand structures were generated directly from SMILES strings using LIGPREP in MAE format. The OPLS_2005 force field was employed and tautomeric and ionization states were generated using a pH of 7.0 ± 0.2. A maximum of one stereoisomer was generated per ligand, and EPIK was not used. All docking used the High-Throughput Virtual Screening (HTVS) precision mode. Jobs were distributed across the cluster by running multiple instances of LIGPREP in parallel, and distributing structure chunks to target nodes, and running GLIDE locally to eliminate network I/O bottlenecks and sync the parsed scores.

AUTODOCK VINA Ligand structures for all Vina-based engines were prepared in PDBQT format. Initially, the ligands were processed to compute AM1-BCC charges and converted to MOL2 format using ANTECHAMBER (Wang et al., 2006). The resulting structures were then parameterized using the `meeko` (Santos-Martins et al., 2025) Python package (via the `MoleculePreparation` class) applying its default protonation and tautomerization rules. CPU-based AUTODOCK VINA was conducted with `exhaustiveness` parameter of 1.

UNI-DOCK VINA/VINARDO For GPU-accelerated docking, UNI-DOCK wasused with PDBQT files as input format. UNI-DOCK was executed using the `fast` search mode and constrained to output a maximum of one binding mode. The Vina and Vinardo scoring functions were evaluated. To maximize GPU usage, the dynamic dispatch system continuously transferred structure files to nodes with available GPU workers.

DOCK3 Unlike the Vina-family and GLIDE evaluations, which we executed directly within our unified benchmarking infrastructure, the DOCK3 scores were obtained from the publicly available datasets released alongside the original virtual-screening studies for these targets (see the references in Table A2.

BOLTZ

TARGET SEQUENCES For each of the seven receptors, the sequence supplied to both the alignment search and the structure prediction was derived from the deposited receptor coordinates rather than from the native UniProt entry, so that the modeled construct matches the protein against which the reference docking scores were computed.

Residues resolved in the structure were read from the receptor file and checked against the corresponding RCSB polymer entity; residues present in the entity but unresolved in the density were restored. Two receptors required additional handling. EP4R (PDB 5YWY, as mentioned in Table A2) is an ICL3-deletion construct: seven residues (RQFFRRI, entity positions 215–221) were present in the crystallised protein but unresolved, and were restored, giving a 287-residue chain; the 42 native residues removed during construct design (UniProt P35408, 218–259) were not reinstated. D_4 (PDB 5WIU) carries a BRIL (cytochrome b_{562}) fusion in place of ICL3; the fusion (entity 232–337, 106 residues) was removed, and the two receptor fragments were joined directly, giving a 278-residue chain with an artificial TM5–TM6 junction after residue 197.

Final chain lengths were 164 (σ_2R), 278 (D_4), 287 (EP4R), 305 (M^{Pro}), 321 (5-HT_{2A}), 358 (AmpC) and 540 (SERT) residues (see Table A3).

MULTIPLE SEQUENCE ALIGNMENTS Alignments were built with EVcouplings 0.3.0 (Hopf et al., 2019), using the `align` stage with the `standard` protocol: five iterations of jackhmmer (HMMER 3.1b2) (Eddy, 2011) against UniRef100 (Suzek et al., 2015), release 2026_01. Search thresholds were expressed as bit scores per residue (`use_bitscores: true`), with both `domain_threshold` and `sequence_threshold` set to 0.8. Post-processing applied a 90 % sequence-identity filter, a 60 % minimum sequence coverage and a 75 % minimum column coverage.

The query for each alignment was the modelled construct sequence defined above rather than the native UniProt sequence, so that the number of alignment columns equals the length of the chain passed to BOLTZ-2. Each alignment was verified before use: the first sequence, with gaps and lower-case insertion columns removed, was required to be identical to the input sequence, and the number of alignment columns was required to equal its length. BOLTZ performs no such check, and a mismatched alignment yields an incorrect result rather than an error.

Alignment statistics are given in Table A3. One alignments are shallow: M^{Pro} reach N_{eff}/L of 0.14. roughly an order of magnitude below the depth at which an alignment carries usable covariation, and predictions for these two targets should be regarded as close to single-sequence.

Table A3: Alignment statistics for the eight targets. L is the length of the modelled chain; N_{eff} is the number of effective sequences after redundancy weighting.

Target	**L**	**Sequences**	$\mathbf{N_{eff}}$	$\mathbf{N_{eff}/L}$	**Column Coverage**
5-HT_{2A}	321	7907	2587.4	8.06	0.894
AmpC	358	1853	909.9	2.54	0.986
D_4	278	5939	2094.6	7.53	0.968
EP4R	287	2029	628.7	2.19	0.958
M^{Pro}	305	93	43.0	0.14	0.977
SERT	540	9314	4272.0	7.91	0.961
σ_2R	164	753	439.8	2.68	0.896

STRUCTURE AND AFFINITY PREDICTION Predictions used BOLTZ-2 version 2.2.1 (Passaro et al., 2025) with the BOLTZ-2 model weights, under PyTorch 2.13.0+cu130 on NVIDIA A40 GPUs (compute capability sm_86). BOLTZ-2 extends the AlphaFold3 architecture (Abramson et al., 2024) with a separately trained affinity module; BOLTZ-1 (Wohlwend et al., 2024) describes the underlying open-source reimplementation.

Each input specified one protein chain with its alignment and one ligand given as SMILES, with the ligand declared as the affinity binder. Sampling parameters were left at their defaults: for the structure stage, 3 recycling steps, 200 sampling steps and 1 diffusion sample; for the affinity stage, 5 recycling steps (fixed internally rather than taking the command-line default of 3), 200 sampling steps and 5 diffusion samples. Alignments were truncated to at most 8192 sequences. Potentia based steering was not enabled, so Feynman–Kac steering and physical guidance updates were both off, and the molecular weight correction to the affinity head was not applied. A fixed random seed of 1 was used throughout. Ligands were processed 100 per invocation so that model weights are loaded once per batch rather than once per ligand.

BOLTZ-2 does not pass the full complex to the affinity module. The input is cropped to the 200 protein tokens nearest the ligand, selected by distance and expanded into contiguous sequence neighborhoods of at least 10 residues, subject to overall limits of 256 tokens and 2048 atoms. For the six targets longer than 200 residues, the affinity prediction therefore sees only the binding site neighborhood, and residues outside it cannot influence the predicted affinity.

REPORTED VALUES The reported docking score is `affinity_pred_value`, defined as $\log_{10}(\mathrm{IC}_{50})$ with IC_{50} in micromolar; the lower values indicate stronger predicted binding. This is in the same direction as the Vina and Glide scores. This allows direct comparison without a sign change.

Two further quantities are retained. `affinity_probability_binary` is the predicted probability that the ligand binds at all; it is trained separately from the regression head and is intended for binder/decoy discrimination rather than potency ranking, and its sign convention is opposite to that of the score columns. The absolute difference between the two ensemble members are reported as a per-ligand uncertainty estimate.

COVERAGE Of the 70 000 predictions (7 targets $\times$ 10 000 ligands), 69 993 completed. One ligand (source row 5123) produced no result on any target: RDKit's ETKDGv3 (Riniker & Landrum, 2015; Wang et al., 2020), which BOLTZ-2 uses to generate the ligand's initial 3D conformer from SMILES, could not generate a conformer for its bridged bicyclic scaffold, and the input was skipped before the model ran. This is a conformer generation failure upstream of the network rather than a prediction failure; BOLTZ-2 itself reported zero failed examples. The corresponding entries are left empty rather than zero-filled.

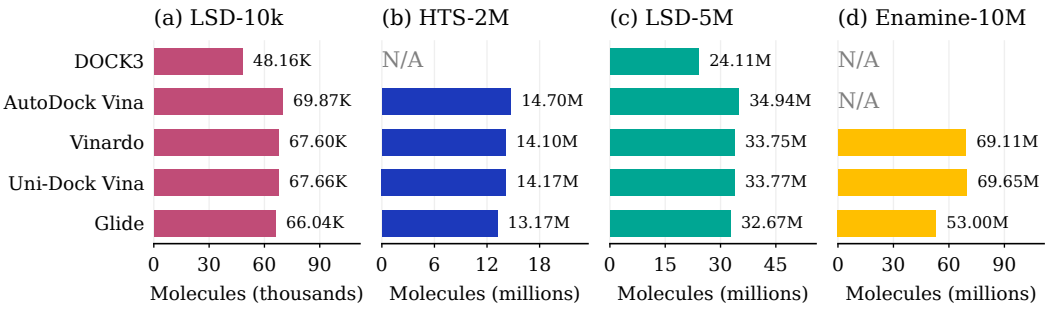


Figure A18: **Total valid docking scores in CHEMVERSE.** The total number of successfully computed valid docking scores generated across each combination of library and docking oracle. For AutoDock Vina, Uni-Dock Vina, and Uni-Dock Vinardo, valid scores were defined as falling between -20 and 0 kcal/mol; for Glide and DOCK3, valid scores were defined as ≤ 0. Missing bars indicate oracles did not run for a specific library (e.g., DOCK3 scores were only sourced for the LSD libraries). There are fewer Glide scores as there were a number of structures that found no pose.

A.3 Implementation Details

A.3.1 Bayesian Neural Networks

All models were implemented in `jax` (v0.10.1) (Bradbury et al., 2018) using `optax` (v0.2.8) (DeepMind et al., 2020) for optimization and `equinox` (v0.13.8)(Kidger & Garcia, 2021) for neural network components.

The neural networks were trained for up to 1500 epochs with a learning rate of 1×10^{-3}, a batch size of 8192, and Adam optimizer (Kingma & Ba, 2015) with a cosine deacy (Loshchilov & Hutter, 2017). The networks comprised two hidden layers of 512 and 128 units with ReLU activations and L_2 weight decay of 1×10^{-4}. We used a 10% validation split and early stopping with a patience of 50 epochs.

For MC Dropout, a dropout probability of 0.1 was applied during training and retained at inference time, with predictive uncertainty estimated from 50 stochastic forward passes. For Laplace, we fit a post-hoc last-layer Laplace approximation using a full curvature approximation. For SWAG, posterior collection began after 75% of training, with parameter snapshots collected every epoch and a low-rank covariance rank of 20; predictive uncertainty was estimated using 30 posterior weight samples.

Monte Carlo Dropout We used a dropout probability of $p = 0.1$ and $T = 50$ Monte Carlo samples. The network was trained normally with dropout enabled; during prediction, we retained dropout and vectorized the 50 stochastic forward passes, taking their empirical mean and variance as the surrogate's predictive mean and epistemic uncertainty.

$$\frac{1}{T}\sum_{t=1}^{T}\left(f_{\mathbf{W}^{(t)}}(\mathbf{x}) - \hat{\mu}(\mathbf{x})\right)^2$$

Our JAX implementation performs the stochastic forward passes in parallel:

```
@eqx.filter_jit  # import equinox as eqx
def mc_dropout_predict(model, x, sample_keys):
    def sample_prediction(key):
        return _predict_model(
            model, x, training=True, key=key
        )

    predictions = jax.vmap(sample_prediction)(sample_keys)
    return (
        jnp.mean(predictions, axis=0),
        jnp.var(predictions, axis=0),
    )

sample_keys = jax.random.split(key, n_mc_samples)
mean, variance = mc_dropout_predict(
    model, x, sample_keys
)
```

SWA-Gaussian (SWAG) We began collecting SWAG snapshots after 75% of training, collected one snapshot per epoch thereafter, retained a maximum low-rank covariance rank of 20, and used $T = 30$ posterior samples for prediction. The empirical mean and variance across sampled networks were used as the surrogate predictive mean and epistemic variance, respectively.

$$\frac{1}{K-1}\mathbf{D}\mathbf{D}^\top$$

with $\mathbf{D}$ containing deviations of the collected parameter vectors from their running mean.

Our JAX implementation samples from the diagonal-plus-low-rank approximation as

```
def sample_weights(mean, sq_mean, D, key):
    diag_key, low_rank_key = jax.random.split(key)

    var = jnp.maximum(sq_mean - mean**2, 1e-30)
    theta = mean + jnp.sqrt(var) * jax.random.normal(
        diag_key, mean.shape
    ) / jnp.sqrt(2.0)

    z = jax.random.normal(low_rank_key, (D.shape[0],))
    theta += D.T @ z / jnp.sqrt(2.0 * (D.shape[0] - 1))
    return theta
```

Laplace Approximation We used a last-layer Laplace approximation with a diagonal curvature approximation, prior precision $\lambda = 1.0$, and observation-noise variance $\sigma^2 = 1.0$. The resulting posterior covariance was propagated analytically through the final linear layer to obtain the predictive mean and epistemic variance.

$$(\mathbf{H} + \lambda \mathbf{I})^{-1},$$

with $\mathbf{H}$ denoting the curvature of the data likelihood and λ the prior precision.

$$\boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma} \boldsymbol{\phi}(\mathbf{x}),$$

with the bias incorporated into the feature vector by appending a constant feature.

We implemented the last-layer Laplace approximation in JAX using laplax. After MAP training, the hidden representations were computed using the frozen network, and `laplax` (Weber et al., 2025) was used to construct the Gaussian posterior over the parameters of the final linear layer:

```
train_features = hidden_features(model, train_x)
last_layer = model.layers[-1]

posterior_fn, _ = laplace(  # from laplax.api import laplace
    model_fn=last_layer_fn,
    params=last_layer_params,
    data=(train_features, train_y),
    loss_fn="mse",
    curv_type="diagonal",
)

posterior = posterior_fn({
    "prior_prec": 1.0,
    "sigma_squared": 1.0,
})
posterior_cov = posterior_covariance(posterior)
```

For prediction, the posterior covariance was propagated analytically through the final linear layer:

```
mean = last_layer(features)
design = design_matrix(last_layer, features)
variance = jnp.einsum(
    "bp,pq,bq->b", design, posterior_cov, design
)
```

A.3.2 Acquisition Functions

At each optimization iteration, acquisition functions were evaluated over the remaining candidate molecules using the predictive mean $\mu(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$ provided by the surrogate model. All acquisition functions were formulated for maximization.

Upper Confidence Bound. Upper Confidence Bound (UCB) trades off predicted performance and epistemic uncertainty according to

$$\alpha_{\mathrm{UCB}}(\mathbf{x}) = \mu(\mathbf{x}) + \beta\, \sigma(\mathbf{x}),$$

where $\beta \geq 0$ controls the strength of exploration. Larger values of β favor candidates with greater predictive uncertainty, while $\beta = 0$ reduces UCB to greedy selection.

Expected Improvement. For incumbent best value $f^\star$, Expected Improvement (EI) is

$$\alpha_{\mathrm{EI}}(\mathbf{x}) = \sigma(\mathbf{x})\left[\phi(z) + z\Phi(z)\right], \qquad z = \frac{\mu(\mathbf{x}) - f^\star - \xi}{\sigma(\mathbf{x})},$$

where ϕ and Φ denote the standard normal density and cumulative distribution functions, respectively, and ξ is an optional exploration parameter. We used $\xi = 0$.

Direct evaluation of EI can suffer severe numerical underflow and cancellation when z is strongly negative. We therefore used the LogEI formulation of Ament et al. (2023), evaluating

$$\log \alpha_{\mathrm{EI}}(\mathbf{x}) = \log \sigma(\mathbf{x}) + \log[\phi(z) + z\Phi(z)]$$

directly in log space. The two terms inside the logarithm were combined using numerically stable log-space operations, with an inverse-Mills-ratio asymptotic expansion used in the extreme negative tail. Since the logarithm is strictly monotonic, LogEI preserves the ordering induced by EI while avoiding numerical collapse for very small improvement values.

Probability of Improvement. Probability of Improvement (PI) selects candidates according to the posterior probability that they improve upon the current best observation,

$$\alpha_{\mathrm{PI}}(\mathbf{x}) = \Phi\left(\frac{\mu(\mathbf{x}) - f^\star}{\sigma(\mathbf{x})}\right).$$

Analogously to EI, we evaluated PI in log space,

$$\log \alpha_{\mathrm{PI}}(\mathbf{x}) = \log \Phi(z), \qquad z = \frac{\mu(\mathbf{x}) - f^\star}{\sigma(\mathbf{x})},$$

using a numerically stable implementation of the log-normal CDF. LogPI preserves the ranking of conventional PI while remaining stable when $\Phi(z)$ is close to zero.

Greedy. As an uncertainty-independent baseline, greedy acquisition ranks candidates solely by their posterior predictive mean,

$$\alpha_{\mathrm{Greedy}}(\mathbf{x}) = \mu(\mathbf{x}).$$

This corresponds to pure exploitation and provides a direct baseline for assessing whether incorporating surrogate uncertainty improves candidate selection.

A.3.3 Molecular Representations

We considered molecular representations spanning fixed topological fingerprints and pretrained neural encoders operating on molecular graphs, SMILES strings, and three-dimensional molecular structures.

Molecular representations were obtained either from the original implementations released by their respective authors or through the Hugging Face `transformers` library (v5.1.0) (Wolf et al., 2020). Morgan fingerprints were computed using `rdkit` (v2025.09.4) (Landrum et al., 2023).

Morgan Fingerprints. Morgan fingerprints are fixed circular molecular fingerprints that encode the local topological environments surrounding each atom (Rogers & Hahn, 2010). Starting from atom identifiers $h_v^{(0)}$, representations are iteratively updated by hashing the identifier of each atom together with those of its neighbors,

$$h_v^{(r)} = \mathrm{Hash}\left(h_v^{(r-1)}, \left\{h_u^{(r-1)} : u \in \mathcal{N}(v)\right\}\right),$$

where r denotes the neighborhood radius. The resulting atom-centered substructures are hashed into a fixed-dimensional binary vector

$$\mathbf{x}_j = \mathbb{I}\left[\exists\, h_v^{(r)} \text{ mapped to bit } j\right].$$

Unlike the learned representations below, Morgan fingerprints contain no pretrained parameters and provide a purely topological baseline.

GROVER. GROVER is a self-supervised molecular graph encoder that combines message-passing networks with a Transformer-style architecture (Rong et al., 2020). A molecule is represented as a graph $G = (V, E)$, with hidden atom and bond representations iteratively updated from their local neighborhoods. Abstractly, a message-passing layer takes the form

$$\mathbf{h}_v^{(\ell+1)} = \mathrm{Update}\left(\mathbf{h}_v^{(\ell)}, \mathrm{AGG}\left\{\mathbf{h}_u^{(\ell)}, \mathbf{e}_{uv} : u \in \mathcal{N}(v)\right\}\right).$$

GROVER augments these local representations with Transformer-style interactions and is pretrained using self-supervised objectives at the node, edge, and graph levels. A molecular representation is obtained by aggregating the learned graph-level features into a fixed-dimensional embedding $\mathbf{z}_{\mathrm{GROVER}}$.

MolE. MolE is a pretrained Transformer for molecular graphs that learns atom-environment representations from large-scale unlabeled molecular data (Méndez-Lucio et al., 2024). MolE modifies disentangled attention to incorporate the relative topological positions of atoms in the molecular graph. In general, attention between atoms i and j can be expressed as

$$a_{ij} = \frac{\exp(s_{ij})}{\sum_k \exp(s_{ik})},$$

where s_{ij} depends jointly on the learned atom representations and the relative graph position between i and j. The resulting contextualized atomic representations

$$\{\mathbf{h}_1, \ldots, \mathbf{h}_{|V|}\}$$

are aggregated to obtain a fixed-dimensional molecular embedding $\mathbf{z}_{\mathrm{MolE}}$. MolE is pretrained by predicting atomic environments and was trained on approximately 842 million molecular graphs.

T5-Chem. T5-Chem is a chemical language model based on the T5 encoder–decoder architecture (Raffel et al., 2020) and operates on molecules represented as SMILES sequences (Christofidellis et al., 2023). Given a tokenized SMILES sequence

$$(s_1, \ldots, s_L),$$

the Transformer encoder produces contextualized token representations

$$(\mathbf{h}_1, \ldots, \mathbf{h}_L) = \mathrm{Encoder}(s_1, \ldots, s_L),$$

where each self-attention layer (Vaswani et al., 2017) computes

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}.$$

A fixed-dimensional molecular representation is then obtained by pooling the contextualized sequence representations,

$$\mathbf{z}_{\mathrm{T5}} = \mathrm{Pool}\left(\mathbf{h}_1, \ldots, \mathbf{h}_L\right).$$

The pretrained model therefore provides a learned continuous representation of molecular structure encoded through chemical language.

MolFormer. MolFormer is a large-scale Transformer encoder pretrained on approximately 1.1 billion SMILES strings from PubChem and ZINC (Ross et al., 2022). Like T5-Chem, it maps a SMILES sequence to contextualized token representations, but employs rotary positional embeddings and a linear-attention mechanism designed to scale efficiently to large pretraining corpora. Denoting its Transformer encoder by f_θ,

$$(\mathbf{h}_1, \ldots, \mathbf{h}_L) = f_\theta(s_1, \ldots, s_L), \qquad \mathbf{z}_{\mathrm{MolFormer}} = \mathrm{Pool}\left(\mathbf{h}_1, \ldots, \mathbf{h}_L\right).$$

Thus, MolFormer provides a learned continuous molecular representation derived entirely from the one-dimensional SMILES representation while capturing contextual relationships between tokens.

UniMol. Uni-Mol is a pretrained three-dimensional molecular representation model based on an SE(3)-aware Transformer architecture (Zhou et al., 2023; Ji et al., 2024). A molecule is represented by atom identities and coordinates

$$\mathcal{M} = \{(a_i, \mathbf{r}_i)\}_{i=1}^{N}, \qquad \mathbf{r}_i \in \mathbb{R}^3.$$

Pairwise geometric information is derived from the coordinates, most directly through interatomic distances

$$d_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|_2,$$

and incorporated into the attention mechanism alongside learned atom representations. The resulting atom-level features are aggregated into a molecular embedding

$$\mathbf{z}_{\mathrm{UniMol}} = \mathrm{Pool}\left(\mathbf{h}_1, \ldots, \mathbf{h}_N\right).$$

Unlike the preceding 1D and 2D representations, Uni-Mol explicitly incorporates molecular geometry and was pretrained on large-scale collections of three-dimensional molecular conformations.