SynthonBench: Benchmarking Sample-Efficient Optimization in Combinatorial Chemical Spaces

Anonymous Authors[1]

Abstract

Synthesis-on-demand libraries expose vast chemical spaces through reaction templates and compatible synthons. In this setting, the central question is not whether a generator can emit valid molecules, but which candidates to evaluate under a small oracle budget. We introduce SynthonBench, a benchmark for reaction-native, budgeted search in synthon-based libraries. Optimizers propose valid reaction/synthon tuples and receive oracle scores; after each run, stable product identifiers are joined to reference tables to compute top-k recall, enrichment, regret, and two-objective Pareto metrics. SynthonBench includes exact-audit 1M tasks, 10M scale tasks, and single-seed 100M feasibility runs covering docking-surrogate, selectivity, and synthetic diagnostic objectives. Across nine baselines and five seeds on the 1M/10M suites, genetic algorithms and Thompson-style synthon sampling are strongest on docking-surrogate tasks, while factorization machines excel on pairwise synthetic objectives. On 1M docking, the best baseline reaches mean top-100 recall 0.645 versus 0.178 for random search; on 10M, 0.321 versus 0.039. The benchmark exposes where search methods exploit synthon structure and where larger reaction spaces change the leaderboard.

1. Introduction

Ultra-large synthesis-on-demand libraries are now standard inputs to early hit discovery. Their size is largely combinatorial: purchasable products are generated from validated reaction schemes and in-stock building blocks, often accessed as synthon-based spaces rather than as explicitly enumerated molecule lists (Gorgulla et al., 2020; Sadybekov et al., 2022; Kalliokoski et al., 2025). This representation changes the algorithmic problem. Screening is no longer about ranking a fixed file of SMILES. A method must decide which reaction-compatible tuples to score, how to reuse information across building-block positions, and how to spend a limited number of docking, surrogate, or assay evaluations.

Most molecular-generation benchmarks ask whether a model can produce valid and high-scoring molecules. That question remains useful, but it is not the same as search in a synthesis-on-demand library. In synthon-based virtual screening, every candidate carries synthetic provenance by construction, duplicate oracle calls are wasteful, and docking-score sign conventions require care because lower raw docking scores map to higher benchmark utilities. A small gain in best scalar score does not always translate into recovery of elite compounds. We therefore evaluate the search procedure separately from the audit procedure.

SynthonBench studies this setting directly. Each task defines a reaction-specific product space in which a molecule is represented by a reaction identifier and one synthon per reaction component. During search, methods interact only with a reaction-native API and a limited oracle budget; the API returns valid synthon ids per reaction slot and scores proposed tuples on demand. This separation between *search* (reaction-native, budgeted) and *audit* (post-hoc, exact) is the core design choice of SynthonBench. During audit, queried product identifiers are joined to precomputed reference score tables to compute top-100 recall, top-1000 recall, enrichment, regret, and Pareto-front metrics. This lets the benchmark test sample-efficient chemical-space search without requiring every optimizer to enumerate or store the full product space.

Many generative or agentic molecular-design systems ultimately need an acquisition policy over purchasable, reaction-constrained candidates; SynthonBench isolates this decision problem while holding the chemistry and scoring interface fixed.

SynthonBench is intended to complement molecular-generation benchmarks such as GuacaMol (Brown et al., 2019), MOSES (Polykovskiy et al., 2020), PMO (Gao et al., 2022a), Therapeutics Data Commons (Huang et al., 2021),

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anonymous-contact>.

Submitted to the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026). Do not distribute.

and docking-oriented benchmark suites such as DOCKSTRING (Garcia-Ortegon et al., 2022), not replace them. Those resources standardize distribution learning, de novo design, molecular optimization, therapeutic prediction tasks, and docking-score objectives. SynthonBench asks a different question: given a fixed reaction-defined chemical space and a fixed evaluation budget, which search policy recovers the top-ranked products most efficiently?

Related virtual-screening systems motivate this protocol but evaluate a different object. VirtualFlow and V-SYNTHES showed the value of ultra-large and synthon-based docking (Gorgulla et al., 2020; Sadybekov et al., 2022); SpaceLight and FTrees search non-enumerated spaces by ligand similarity (Bellmann et al., 2021; Rarey & Stahl, 2001); SpaceProp and SpaceProp2 compute exact property distributions over fragment spaces without product enumeration (Bellmann et al., 2022; Lübbers et al., 2024); Thompson sampling has been proposed for ultralarge synthesis-on-demand search (Klarich et al., 2024); RosettaAMRLD combines reaction-based proposal generation with Monte Carlo Metropolis and RosettaLigand redocking for structure-based de novo design (Tang et al., 2025); and SpaceHASTEN combines whole-molecule similarity expansion, learned docking-score prediction, and docking iterations over non-enumerated libraries (Kalliokoski et al., 2025). Pool-based active learning for virtual screening studies a related budgeted-acquisition problem over enumerated molecular pools (Graff et al., 2021). SynthonBench turns that family of workflows into an optimizer benchmark: every method receives the same reaction-native API, oracle budget, and exact post-hoc audit.

Synthesis-aware generators address a complementary problem. SynNet, Reaction-GFlowNet, and SynFormer generate or navigate synthesizable molecules by using reaction templates, synthetic trees, or reaction-space policies (Gao et al., 2022b; Horwood et al., 2024; Gao et al., 2025). SynthonBench instead evaluates budgeted search behavior inside a fixed reaction-native library with exact post-hoc audits. The chemistry is fixed, the query budget is fixed, and the primary measurement is sample-efficient recovery of elite products.

Our contributions are:

- A reaction-native benchmark interface for budgeted search in synthon-based libraries, instantiated as a 1M exact-audit tier, a 10M scale tier, and single-seed 100M feasibility runs covering docking-surrogate, selectivity, and synthetic-diagnostic tasks.
- An exact post-hoc audit based on stable product identifiers, reporting top-k recall, enrichment, regret, and two-objective Pareto metrics by joining queried products to reference score tables.
- A five-seed study of nine benchmark-native baselines: random search, Thompson-style synthon sampling, slotwise UCB, factorization machines, factorized random forests, a genetic algorithm, hill climbing, MCTS, and SpaceHASTEN-style iterative search (ISE).
- Reaction-size and query-phase diagnostics showing when product-weighted summaries act as large-reaction audits, and how elite recovery relates to exploration coverage and local exploitation around exact top products.
- A reproducibility package with manifests, command logs, environment captures, exact metric tables, and figure data.

2. Benchmark Design

Formal benchmark statement. Each scalar task is a fixed-budget deterministic black-box optimization problem over a reaction-specific product set

$$\mathcal{P}_r = \{\mathrm{Product}_r(s_1, \ldots, s_m) : s_i \in S_{r,i}\},$$

where r is a reaction template and $S_{r,i}$ is the typed synthon set for slot i. Unless a run explicitly overrides the protocol, the oracle budget is

$$B_r = \max(1000, \min(10000, \lfloor 0.01|\mathcal{P}_r| \rfloor)).$$

An optimizer proposes valid tuples $(r, s_1, \ldots, s_m)$, receives an oracle value, and must stop after B_r oracle calls. Each run records the queried product identifiers, oracle traces, best-so-far curves, and the final selected set. Stable product identifiers are central: post-hoc metrics join queried ids to reference score tables, rather than relying on canonical-SMILES matching. Methods are evaluated as pure-exploration policies by simple regret and elite-set recovery rather than cumulative reward (Bubeck et al., 2009). This pure-exploration framing matches virtual screening: the query cost is paid during search, while the benchmark asks how much of the best reference set was recovered after the budget is exhausted.

Task families. The 1M tier with exact audits uses 990,610 unique products after deduplication of product identifiers. The docking track contains 126 tasks: 42 reactions crossed with three SpaceHASTEN LightGBM docking-surrogate targets, KIF11, PYRD, and TGFR1. Lower raw docking score is better, so utilities are negated raw scores. The surrogates are trained from 0.85–1.00M source docked product rows per target, generated with Phase/LigPrep and Glide from Schrödinger Suite 2023-4 following SpaceHASTEN (Kalliokoski et al., 2025; Friesner et al., 2004), with 0.60–0.70M rows in the 70% LightGBM training split. On held-out products, Spearman rank correlation with the Glide-derived source docking labels is 0.71–0.79, while lowest-0.1% tail recall is 0.153–0.279 (Appendix Table 11). We

therefore interpret the docking tasks as fixed surrogate-landscape search benchmarks, not as substitutes for prospective docking or biological validation. Appendix Table 12 reports the full LightGBM validation metrics; the 70/15/15 split is random over product rows, so product rows are disjoint, but it does not enforce scaffold-, synthon-, or reaction-disjoint partitions. Shared synthons, reaction templates, and related scaffolds can therefore appear on both sides of the split, and the validation measures interpolation within the source score distribution rather than transfer to new chemistry or measured binding. The selectivity track also contains 126 tasks: 42 reactions crossed with three target pairs, KIF11/PYRD, KIF11/TGFR1, and PYRD/TGFR1. The optimizer searches against the first target in the pair as a single scalar utility, but the queried set is audited against the two-target Pareto frontier, for example KIF11 versus PYRD. This measures how well a scalar search recovers the multi-objective trade-off surface.

The synthetic diagnostic track contains 168 1M tasks spanning additive, pairwise, sparse three-way, and constraint objectives. The additive family has

$$u_{\mathrm{add}}(s) = \sum_i a_i(s_i),$$

and the pairwise family adds slot interactions,

$$u_{\mathrm{pair}}(s) = \sum_i a_i(s_i) + \sum_{(i,j)\in E} b_{ij}(s_i, s_j).$$

Sparse three-way objectives add a small number of higher-order interaction bonuses, while constraint objectives combine slot utilities with binary feasibility penalties. These tasks test whether a method can exploit slotwise structure and whether interaction-heavy objectives separate methods that look similar on simple additive functions. The 10M synthetic scale track contains 126 additive, pairwise, and sparse3 tasks on a 10,022,100-product deduplicated space; constraint objectives are exercised only at the 1M tier, where exact post-hoc auditing of feasibility is tractable. The 10M docking scale track uses a table-backed task file generated from the same SpaceHASTEN LGBM prediction table, deduplicated to 10,022,100 product ids. The single-seed 100M docking and synthetic feasibility runs reuse the same 126-task docking and synthetic scale grids over a 99,845,070-product deduplicated tier, but reduce replication to seed 0 for feasibility and runtime accounting. Table 1 summarizes the resulting suites; Appendix Table 9 catalogs the objective-level tasks that are instantiated over the 42 reactions in each reported suite.

Exact metrics. For a score-table task, the exact reference for reaction r and score column c is obtained by sorting all products for r by utility. For a run with queried set Q, top-k recall is

$$\frac{|Q \cap \mathrm{Top}_k(r,c)|}{|\mathrm{Top}_k(r,c)|}.$$

We also report enrichment relative to the top-0.1% hit rate, exact regret $\max(0, u^\star - \max_{p\in Q} u(p))$, and the rank of the best queried product. For selectivity, we compute the exact two-objective Pareto frontier and report frontier recall, hypervolume ratio, and epsilon regret.

For uniform sampling without replacement from $N = |\mathcal{P}_r|$ products with budget B, and an exact top-k set T_k,

$$H = |Q \cap T_k| \sim \mathrm{Hypergeometric}(N, k, B),$$

and hence

$$\mathbb{E}[\text{Top-}k \text{ recall}] = \frac{B}{N}.$$

For top-α enrichment, the expected random enrichment is 1. We therefore report both recall and enrichment: recall measures absolute recovery of elite products, while enrichment normalizes against the random hit-rate scale.

3. Methods

We evaluate nine benchmark-native baselines, each implemented to interact directly with the synthon-space interface; they propose reaction-component tuples, not arbitrary SMILES. They are scalable reference implementations under one API, not faithful reproductions of every named upstream system; the Thompson and ISE rows are adapted analogues. Random search establishes the lower bound. Our Thompson-style sampler is inspired by reagent-space Thompson sampling for synthesis-on-demand databases (Klarich et al., 2024). It maintains empirical slot statistics and samples optimistic slot values within the benchmark budget. SlotUCB is a slotwise empirical-mean plus upper-confidence heuristic. The factorization machine baseline learns low-rank slot interactions (Rendle, 2010); the random-forest factorized baseline uses synthon fingerprints with periodic refits (Breiman, 2001; Rogers & Hahn, 2010). The genetic algorithm maintains a tuple population with crossover, mutation, random immigrants, and unseen-child retries. Hill climbing performs local synthon swaps. MCTS treats partial tuple assignment as a tree search (Browne et al., 2012). SpaceHASTEN-style iterative search (ISE) starts with random seed queries, periodically refits a product-fingerprint regressor, expands local synthon neighborhoods around high-scoring queried or predicted tuples, and greedily queries the best predicted candidates. This preserves the benchmark budget and trace contract while avoiding runtime dependencies on SpaceLight, FTrees, Glide, or chemprop. We do not include a Metropolis-style chain search such as the reaction-driven sampling in RosettaAMRLD (Tang et al., 2025) or related reaction-aware generative work (Swanson et al., 2024) in the main leaderboard; the artifact includes

Table 1. Benchmark-card summary for the reported suites. Budgets follow the B_r rule defined in the formal statement; the budget column reports the minimum and maximum observed budgets across reactions. All docking rows use SpaceHASTEN LGBM surrogate scores rather than measured docking.

Suite	Tasks	Reactions	Median products/task	Max products/task	Budgets
1M docking-surrogate	126	42	5,329	389,017	1,000–3,890
1M selectivity audit	126	42	5,329	389,017	1,000–3,890
1M synthetic diagnostics	168	42	5,329	389,017	1,000–3,890
10M docking-surrogate	126	42	27,225	4,492,125	1,000–10,000
10M synthetic scale	126	42	27,225	4,492,125	1,000–10,000
100M docking scale-up	126	42	152,100	59,319,000	1,000–10,000
100M synthetic scale-up	126	42	152,100	59,319,000	1,000–10,000

a benchmark-native `reaction_metropolis` run for future audit (see Appendix).

All methods are wrapped with a unique-query guard. Duplicate proposals can occur internally, but duplicate oracle calls are rejected before scoring.

4. Experimental Protocol

All five-seed runs use seeds $0, \ldots, 4$. Each method follows a frozen execution protocol with task-count, score-table-uniqueness, command-registration, and smoke-run checks logged to the run manifest. Large scalar sweeps disable on-line diversity tracking (`--no-diversity`); portfolio diversity references are computed separately. The result root contains manifests, command logs, environment captures, exact metric tables, and plot data. The 100M scale-up stage is a single-seed, model-backed feasibility run over the full docking grid; full provenance appears in the Appendix.

5. Results

Table 2 shows that the exact metrics differentiate methods more clearly than the best-utility column alone. On 1M docking-surrogate tasks, the genetic algorithm reaches mean top-100 recall of 0.645 and enrichment 5.62, while random search reaches 0.178 and enrichment 0.99. Reaction-level bootstrap 95% CIs for top-100 recall are [0.620,0.667] for GA and [0.164,0.187] for random. At 10M, absolute top-100 recall drops because the same budget rule covers a smaller fraction of the largest reaction spaces, but separation grows: GA reaches 0.321 versus 0.039 for random, with enrichment above 12 and reaction-level CI [0.301,0.342]. Product-weighted top-100 recall tells a large-reaction-heavy story: the largest six docking tasks are only 4.8% of the task average, but 78.5% of 1M product mass and 89.6% of 10M product mass. GA scores 0.466 product-weighted top-100 recall on 1M and 0.349 on 10M, versus 0.048 and 0.005 for random. Figure 1 makes this heterogeneity explicit. GA remains strongest in both small and large bins, but the middle of the leaderboard changes with scale: Thompson is much stronger on small reactions, whereas ISE is a competitive large-reaction active-screening reference. RF and SlotUCB remain near random on the large reactions that dominate product-weighted mass. The 100M run is a feasibility check for the benchmark interface rather than a statistically powered leaderboard: GA reaches mean best utility 9.182, Thompson reaches 9.091, and random reaches 8.789. Exact top-100 recall at 100M is very sparse because each run queries at most 10,000 products per reaction from spaces as large as 59.3M products; we therefore use it as a supplementary audit rather than as five-seed statistical evidence.

The same API and baselines run unchanged on the 10M and 100M synthon spaces. GA is strongest on 10M docking-surrogate and 10M synthetic diagnostics, has the highest surrogate-scored docking utility in the 100M run, and also leads the 100M synthetic scale-up with mean utility gain 0.427 over random. Thompson remains strong on docking-surrogate tasks; FM remains useful on synthetic interaction tasks but is slower. RF is consistently weak, which is evidence that this RF design is poorly matched to these synthon spaces, not a deficiency of the benchmark.

The 100M docking scale-up in Table 3 and Figure 3 keeps the full docking task grid while reducing the replication axis to seed 0. It confirms that the same benchmark API can run over the 100M synthon tier, with all nine paper methods completing. The ranking by model-backed best utility remains broadly aligned with the 1M/10M story: GA and Thompson lead, ISE is a strong active-screening reference, and random search trails. Peak memory is modest for most methods and highest for ISE because it repeatedly assembles molecules and trains product-fingerprint regressors. Fixed Top-100 and Top-1000 recall are included in the appendix supplements but are too sparse at 100M to serve as the headline scale metric; we therefore pair model-backed best utility with exact-tail enrichment and rank-normalized tail depth.

As an independent robustness check, Appendix A.6 redocks a 2700-molecule stratified sample from the 100M docking-

Table 2. Exact-audit and scale docking-surrogate metrics. Values are means across tasks and seeds for the 1M/10M runs; the 100M columns are the seed-0 full-grid scale-up. Top-100 recall and enrichment are computed post hoc by joining queried product ids to the reference score table. Detailed 100M runtime and memory are reported in Table 3; 100M columns are not included in five-seed statistical comparisons.

	1M docking-surrogate				10M docking-surrogate				100M scale-up	
Method	Best util.	Top-100	Enrich.	Regret	Best util.	Top-100	Enrich.	Regret	Best util.	Top-100
Random	8.253	0.178	0.99	0.221	8.274	0.039	1.01	0.419	8.789	0.012
SlotUCB	8.297	0.204	1.31	0.177	8.334	0.056	1.55	0.358	8.943	0.012
Hill climb	8.322	0.257	1.66	0.152	8.359	0.069	1.93	0.334	8.905	0.011
MCTS	8.379	0.374	2.68	0.095	8.390	0.084	2.49	0.303	8.927	0.010
ISE	8.366	0.349	3.10	0.108	8.418	0.101	4.43	0.275	8.997	**0.012**
Thompson	8.431	0.508	3.93	0.043	8.538	0.223	7.16	0.155	9.091	0.008
GA	**8.445**	**0.645**	**5.62**	**0.029**	**8.594**	**0.321**	**12.15**	**0.099**	**9.182**	0.010
FM	8.305	0.232	1.61	0.169	8.343	0.063	2.10	0.350	8.861	0.010
RF	8.272	0.186	1.11	0.201	8.293	0.043	1.13	0.400	8.905	0.012

Table 3. Full 100M docking-surrogate scale-up, seed 0, over all 126 docking tasks. Best utility and AUC best-so-far are model-backed benchmark metrics. Tail depth is $-\log_{10}(\mathrm{rank}/N)$ for the best exact score-table hit in each task; Top0.1% enrichment is the query hit-rate enrichment against the exact score-table tail. Runtime and memory are per-method process measurements from `/usr/bin/time -v`.

Method	Best util.	AUC best	Tail depth	Top0.1% enrich.	Top0.01% tasks	Wall h	Max RSS GiB
Random	8.789	8.654	3.473	1.022	21.4	3.15	1.58
SlotUCB	8.943	8.799	3.464	1.011	20.6	3.22	1.58
Hill climb	8.905	8.780	**3.502**	1.000	**23.8**	3.15	1.58
MCTS	8.927	8.748	3.375	0.859	15.9	3.25	1.56
ISE	8.997	8.811	3.432	**1.066**	19.8	3.28	5.48
Thompson	9.091	8.909	3.298	0.765	12.7	3.23	1.56
GA	**9.182**	**8.998**	3.289	0.809	15.9	3.16	1.55
FM	8.861	8.671	3.257	0.770	9.5	3.32	1.56
RF	8.905	8.736	3.473	1.022	21.4	3.35	1.71

surrogate scale-up with GNINA/AutoDock Vina (McNutt et al., 2021; Trott & Olson, 2010). This audit is not a replacement oracle for the SpaceHASTEN Glide-derived labels and is not used to rank benchmark submissions. It asks the narrower question of whether molecules selected by the surrogate policies remain enriched under an open docking engine. On that audit, all non-random methods improve over random on median Vina affinity and median GNINA CNN affinity with bootstrap confidence intervals excluding zero; Thompson sampling has the best aggregate redocking rank, while MCTS is strongest for TGFR1.

Figure 4 shows that the scalar curve story is consistent across targets rather than driven by a single protein. Utility-unit gains are modest, but exact top-k recall, the more directly meaningful metric for virtual screening, separates methods by an order of magnitude. Appendix Figure 6 therefore repeats the trajectory analysis directly on exact top-100 recall, stratified by reaction size.

Synthetic tasks behave as intended (Table 4). Additive, constraint, and sparse3 objectives produce small positive gains for many methods. Pairwise objectives produce much stronger separation: factorization machines gain 0.417 utility over random on average, GA gains 0.344, ISE gains 0.240, and Thompson gains 0.141. This separation confirms two things: the synthetic diagnostics are not redundant with docking-surrogate tasks, and they can identify which algorithms genuinely exploit slot–slot interactions.

Table 4. Synthetic diagnostic gains over random on the 1M suite. Values are mean utility deltas; bold marks the best method per family, computed before displayed rounding.

Method	Additive	Constraint	Pairwise	Sparse3
Random	0.000	0.000	0.000	0.000
SlotUCB	0.012	0.008	0.065	0.014
Hill climb	0.011	0.024	0.065	0.009
MCTS	0.015	0.028	0.004	0.013
ISE	0.018	0.029	0.240	0.022
Thompson	0.025	0.042	0.141	0.035
GA	**0.026**	**0.047**	0.344	**0.036**
FM	0.016	0.036	**0.417**	0.025
RF	0.006	0.004	0.022	0.004

The selectivity audit adds information beyond scalar best utility (Table 5). GA and Thompson again lead, but the Pareto metrics also reveal middle-tier differences between MCTS, ISE, hill climbing, FM, and SlotUCB. The optimizer

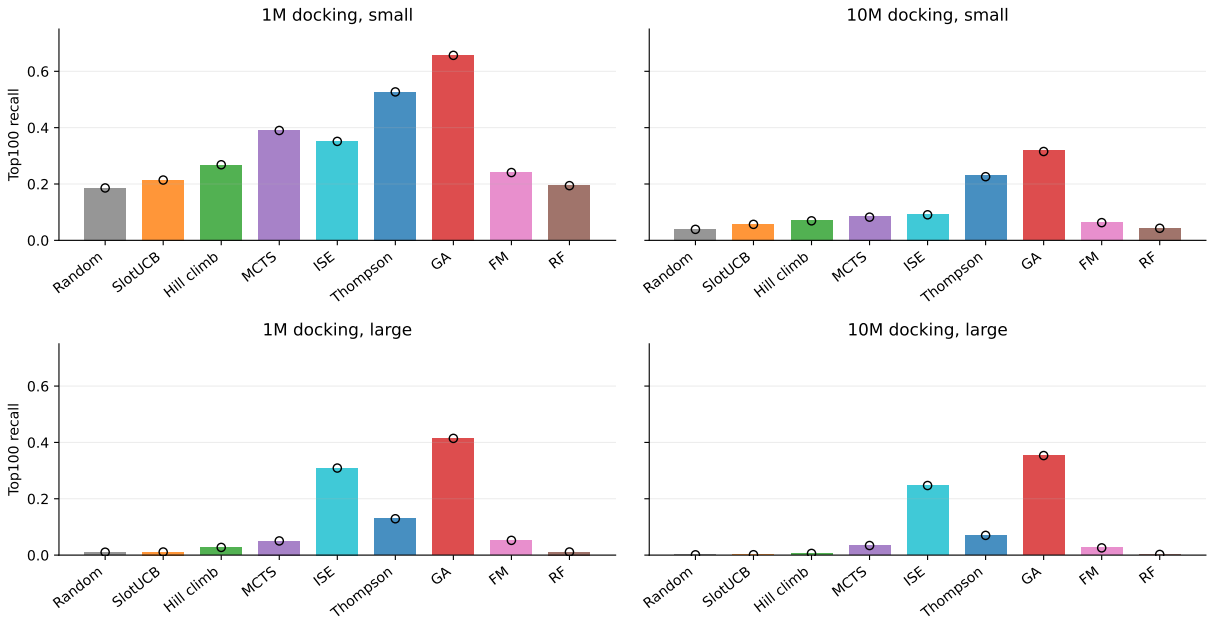

Figure 1. Reaction-size-stratified top-100 recall for the public docking methods. Small reactions are all reactions except the two largest reactions in each suite; large reactions are those two largest reactions crossed with the three docking targets. Hollow markers show the product-weighted value within each panel.

Table 5. Selectivity exact Pareto audit on the 1M exact-audit suite. Metrics are computed post hoc by comparing each method's queried set to the exact two-objective frontier from the reference score table.

Method	Frontier recall	HV ratio	Eps. regret
Random	0.169	0.924	0.476
SlotUCB	0.207	0.932	0.452
Hill climb	0.254	0.939	0.421
ISE	0.320	0.948	0.391
MCTS	0.355	0.952	0.363
Thompson	0.491	0.963	0.312
GA	**0.560**	**0.967**	**0.294**
FM	0.227	0.934	0.444
RF	0.183	0.927	0.467

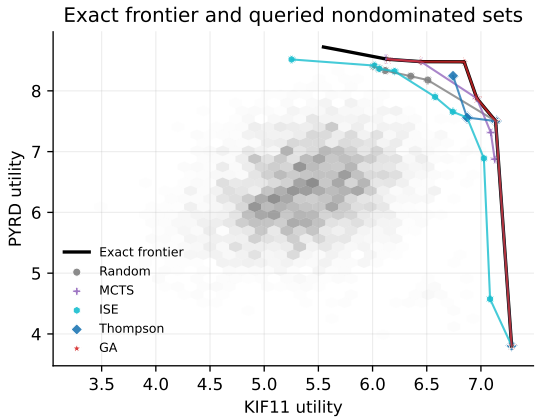

Figure 5. Representative exact selectivity frontier for reaction 11a, KIF11 versus PYRD. Gray shading is exact-space density over 5,329 products; the black line is the exact frontier; colored points and lines are seed-0 nondominated queried subsets, one per method.

still maximizes a scalar utility; the multi-objective metrics audit the queried set rather than the policy, and therefore do not require a true multi-objective optimizer. A true multi-objective baseline, `pareto_ga`, is included as an auxiliary reference in the Appendix. The exact-frontier overlay in Figure 5 illustrates the audit on a single KIF11/PYRD task: search methods discover different parts of the trade-off surface even when scalar best utilities are close.

6. Portfolio References

As a sanity check on the score table itself, we report deterministic score/diversity portfolio references on the 1M tier. A score-only top-1000 selector reaches mean utilities of 7.459 for KIF11, 8.640 for PYRD, and 8.764 for TGFR1, well above random-1000 utilities of 5.398, 6.177, and 6.612. Greedy diversity thresholds trade score for molecular diversity monotonically. Random-budget shadow references, which simulate larger random pools before se-

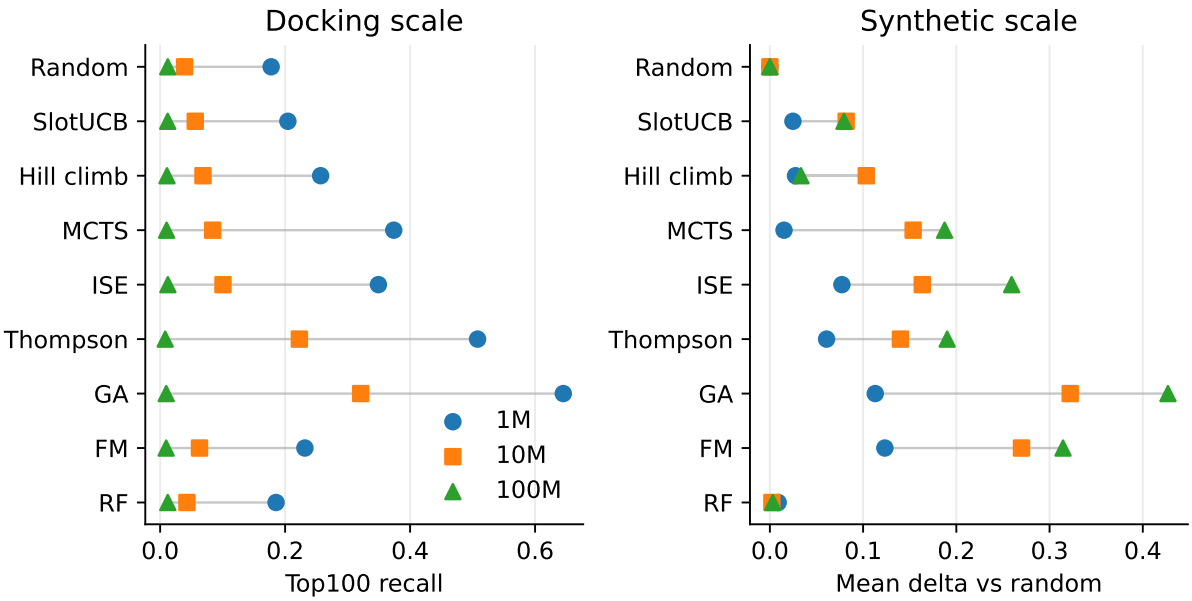

Figure 2. Scaling summary across the 1M, 10M, and 100M docking and synthetic tiers. The 100M points are seed-0 feasibility runs; the 1M/10M points are five-seed means. The 100M markers are not included in statistical comparisons.

lecting top-k products, improve smoothly from random10k to random100k to random1M, which by construction recovers the oracle top-k; a product-RF trained on 10k samples beats random10k but does not match random100k.

7. Discussion

SynthonBench is intentionally simple in method ambition and strict in evaluation. The core result is not that GA or Thompson-style sampling are universally best molecular optimizers. Rather, the benchmark targets four properties a useful molecular-design workflow should exhibit: no duplicate oracle calls, exploitation of synthon-space structure, preservation of synthesizability, and improved elite-product recall under a small budget. These baselines suggest that cheap adaptive methods are hard to beat, that factorized learning helps on synthetic interaction tasks, and that heavier learned baselines need to justify their wall-clock cost.

The reaction-size diagnostics sharpen that summary. Product-weighted docking metrics are best read as audits of the largest reactions, not as replacements for the task-average benchmark. GA remains the most reliable public baseline in both reaction-size regimes, but Thompson sampling, MCTS, and ISE do not degrade in the same way as reaction size grows. Trace-derived diagnostics also caution against the simple claim that broad exploration is sufficient: random and Thompson sampling can cover much of the exact Top-100 slot support, yet recover fewer elite products than GA and ISE on large reactions. The empirical distinction is that GA and ISE query closer to exact Top-100 neighborhoods once useful synthons are discovered.

GA outperforming the benchmark-native Thompson variant should not be read as contradicting prior Thompson-sampling work. Our implementation differs from the released Klarich et al. (Klarich et al., 2024) workflow, and the goal here is to provide a common reaction-native audit rather than to reproduce any one prior system exactly. Thompson-style sampling remains strong and far above random, while exact top-k audit shows that recombination-based search can better exploit reusable high-value synthons on these tasks.

A natural extension is to add SpaceProp-style audits, including exact property distributions and SMARTS-pattern coverage over the same synthon spaces, alongside the current docking-surrogate metrics. We leave that for future work; the present version focuses on score-based audits.

Limitations. Docking scores are SpaceHASTEN LGBM surrogate predictions, not measured docking or wet-lab assays. We treat them as structured benchmark oracles, not biological ground truth; held-out lowest-0.1% recall is moderate, 0.153–0.279 across targets. Method rankings are claims about this fixed surrogate landscape and may change under scaffold-disjoint splits or measured-docking oracles. The 10M docking suite is table-backed rather than model-backed at query time because model-backed scoring was too slow for a full five-seed sweep. The 100M companion artifacts are single-seed scale-ups over the full docking and synthetic scale grids; they establish runtime,

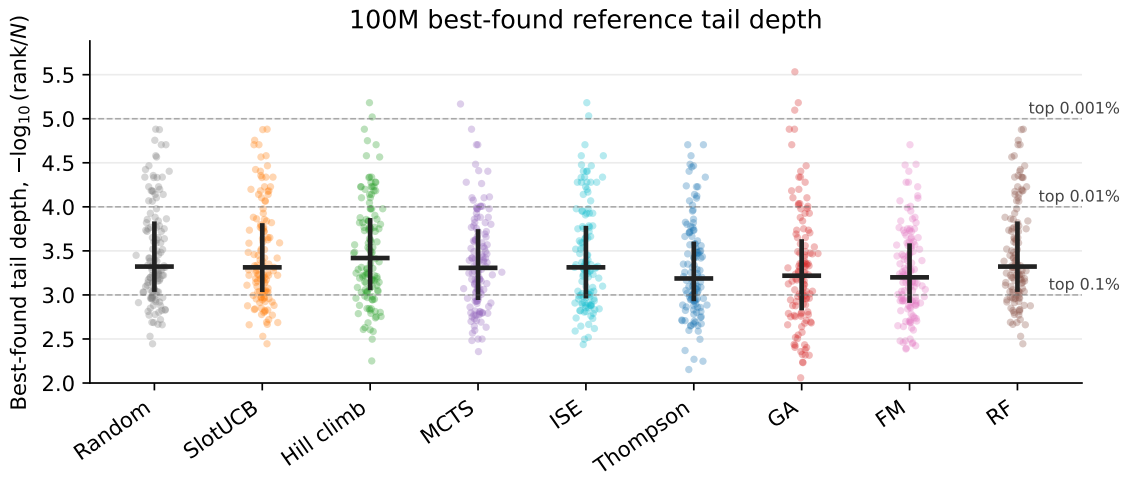

Figure 3. Full 100M docking scale-up exact-tail audit. Each point is one method-task run, plotted by the reference-table tail depth of the best exact hit among queried product ids. Dashed lines mark the top 0.1%, 0.01%, and 0.001% score-table tails. The run uses seed 0 and should be read as feasibility and scaling evidence, not as a replacement for the five-seed 1M/10M exact-audited matrices.

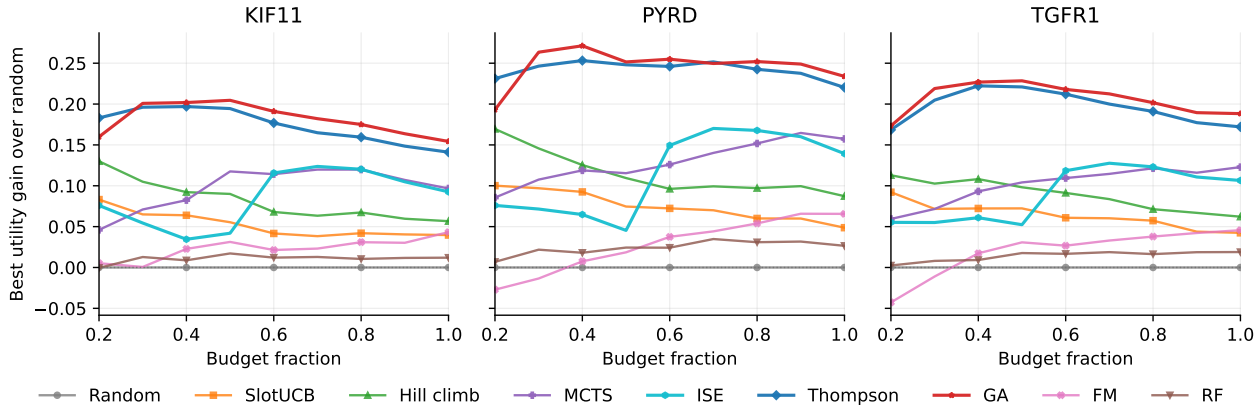

Figure 4. Best-so-far docking-surrogate utility, plotted as gain over random and averaged by target across the 1M exact-audit suite. Curves start after the first 20% of each reaction-specific budget. GA, Thompson sampling, and ISE separate consistently across KIF11, PYRD, and TGFR1.

memory, and ranking feasibility but should not be overread as five-seed statistical evidence. The GNINA/AutoDock Vina redocking audit is an independent enrichment check, not a RosettaAMRLD-faithful pose-aware redocking oracle and not a substitute for the Glide-derived labels; a benchmark-native `reaction_metropolis` companion run is included for future evaluation. Surrogate-based optimization results do not establish biological efficacy, safety, or clinical relevance. The fixed-1000 budget sensitivity check covers random search, Thompson sampling, and GA, but not every public baseline; it therefore checks the central ranking trend rather than replacing the configured-budget benchmark.

8. Conclusion

SynthonBench evaluates the part of molecular design that is often hidden between molecule generation and experimental follow-up: choosing which synthesizable candidates to evaluate when the chemical space is too large to enumerate. By separating reaction-native search from exact product-id audits, the benchmark reports metrics familiar to virtual screening, including top-k recall, enrichment, regret, and Pareto-front coverage. It also retains the oracle-budget framing used in machine learning. The current release uses SpaceHASTEN-derived LightGBM docking surrogates and synthetic diagnostics; it should therefore be read as a benchmark for sample-efficient chemical-space search, not as evidence of binding, biological activity, safety, or clinical relevance. Within that scope, the results show that simple

adaptive policies already recover substantially more elite products than random sampling, that reaction size changes the middle of the leaderboard, and that trace-level diagnostics can separate broad slot exploration from local elite recovery. These results provide reference baselines for future molecular-design algorithms under specified oracle budgets.

References

Bellmann, L., Penner, P., and Rarey, M. Topological similarity search in large combinatorial fragment spaces. *Journal of Chemical Information and Modeling*, 61(1):238–251, 2021.

Bellmann, L., Klein, R., and Rarey, M. Calculating and optimizing physicochemical property distributions of large combinatorial fragment spaces. *Journal of Chemical Information and Modeling*, 62(11):2800–2810, June 2022. ISSN 1549-960X. doi: 10.1021/acs.jcim.2c00334. URL http://dx.doi.org/10.1021/acs.jcim.2c00334.

Breiman, L. Random forests. *Machine Learning*, 45:5–32, 2001.

Brown, N., Fiscato, M., Segler, M. H. S., and Vaucher, A. C. Guacamol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, 2019.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

Bubeck, S., Munos, R., and Stoltz, G. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory*, pp. 23–37, 2009.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, 2002. doi: 10.1109/4235.996017.

Friesner, R. A., Banks, J. L., Murphy, R. B., Halgren, T. A., Klicic, J. J., Mainz, D. T., Repasky, M. P., Knoll, E. H., Shelley, M., Perry, J. K., Shaw, D. E., Francis, P., and Shenkin, P. S. 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. doi: 10.1021/jm0306430. URL https://doi.org/10.1021/jm0306430. PMID: 15027865.

Gao, W., Fu, T., Sun, J., and Coley, C. W. Sample efficiency matters: A benchmark for practical molecular optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21342–21357, 2022a.

Gao, W., Mercado, R., and Coley, C. W. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. In *International Conference on Learning Representations*, 2022b.

Gao, W., Mercado, R., Liu, R., Jeon, W., and Coley, C. W. Generative artificial intelligence for navigating synthesizable chemical space. *Nature Machine Intelligence*, 2025. doi: 10.1038/s42256-025-01085-x.

Garcia-Ortegon, M., Simm, G. N. C., Tripp, A. J., Hernandez-Lobato, J. M., Bender, A., and Bacallado, S. DOCKSTRING: Easy molecular docking yields better benchmarks for ligand design. *Journal of Chemical Information and Modeling*, 62(15):3486–3502, 2022. doi: 10.1021/acs.jcim.1c01334.

Gorgulla, C., Boeszoermenyi, A., Wang, Z.-F., Fischer, P. D., Coote, P. W., Das, K. M., Malets, Y. S., Radchenko, D. S., Moroz, Y. S., Scott, D. A., Fackeldey, K., Hoffmann, M., Iavniuk, I., Wagner, G., and Arthanari, H. An open-source drug discovery platform enables ultra-large virtual screens. *Nature*, 580(7805):663–668, 2020.

Graff, D. E., Shakhnovich, E. I., and Coley, C. W. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical Science*, 12:7866–7881, 2021. doi: 10.1039/D0SC06805E.

Horwood, J., Zimmermann, H., Shen, T., Didi, K., Irwin, R., Barkin, O., Malkin, N., Zhang, D., Tripp, A., Bengio, Y., et al. RGFN: Synthesizable molecular generation using GFlowNets. In *NeurIPS Workshop on AI for New Drug Modalities*, 2024.

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Advances in Neural Information Processing Systems*, volume 34, pp. 3595–3607, 2021.

Kalliokoski, T., Turku, A., and Käsnänen, H. SpaceHASTEN: A structure-based virtual screening tool for nonenumerated virtual chemical libraries. *Journal of Chemical Information and Modeling*, 65(1):125–132, 2025. doi: 10.1021/acs.jcim.4c01790. URL https://doi.org/10.1021/acs.jcim.4c01790. PMID: 39710946.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Klarich, K., Goldman, B., Kramer, T., Riley, P., and Walters, W. P. Thompson sampling: An efficient method for searching ultralarge synthesis on demand databases. *Journal of Chemical Information and Modeling*, 64(4):1158–1171, 2024. doi: 10.1021/acs.jcim.3c01790. URL https://doi.org/10.1021/acs.jcim.3c01790.

Landrum, G. RDKit: Open-source cheminformatics. https://www.rdkit.org/, 2025.

Lübbers, J., Lessel, U., and Rarey, M. Enhanced calculation of property distributions in chemical fragment spaces. *Journal of Chemical Information and Modeling*, 64(6): 2008–2020, Mar 2024. ISSN 1549-960X. doi: 10.1021/acs.jcim.4c00147. URL http://dx.doi.org/10.1021/acs.jcim.4c00147.

McNutt, A. T., Francoeur, P., Aggarwal, R., Masuda, T., Meli, R., Ragoza, M., Sunseri, J., and Koes, D. R. GNINA 1.0: molecular docking with deep learning. *Journal of Cheminformatics*, 13(1):43, 2021. doi: 10.1186/s13321-021-00522-2.

Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., Kadurin, A., Johansson, S., Chen, H., Nikolenko, S., Aspuru-Guzik, A., and Zhavoronkov, A. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11:565644, 2020. doi: 10.3389/fphar.2020.565644.

Rarey, M. and Stahl, M. Similarity searching in large combinatorial chemistry spaces. *Journal of Computer-Aided Molecular Design*, 15:497–520, 2001.

Rendle, S. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pp. 995–1000, 2010.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

Sadybekov, A. V., Sadybekov, A. A., Liu, Y., Iliopoulos-Tsoutsouvas, C., Huang, X.-P., Pickett, J., Houser, B., Patel, N., Tran, N. K., Tong, F., Zvonok, N., Jain, M. K., Sürmeli, N., Roth, B. L., Dror, R. O., Shoichet, B. K., and Katritch, V. Synthon-based ligand discovery in virtual libraries of over 11 billion compounds. *Nature*, 601 (7893):452–459, 2022.

Swanson, K., Liu, G., Catacutan, D. B., Arnold, A., Zou, J., and Stokes, J. M. Generative ai for designing and validating easily synthesizable and structurally novel antibiotics. *Nature Machine Intelligence*, 6(3):338–353, 2024. doi: 10.1038/s42256-024-00809-7. URL https://doi.org/10.1038/s42256-024-00809-7.

Tang, Y., Moretti, R., and Meiler, J. RosettaAMRLD: A reaction-driven approach for structure-based drug design from combinatorial libraries with monte carlo metropolis algorithms. *Journal of Chemical Information and Modeling*, 65(12):5945–5959, 2025. doi: 10.1021/acs.jcim.5c00497. URL https://doi.org/10.1021/acs.jcim.5c00497.

Trott, O. and Olson, A. J. 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, 2010. doi: 10.1002/jcc.21334.

A. Appendix: Benchmark and Artifact Details

A.1. Exact-Audit Semantics

SynthonBench separates the optimizer-facing task from the reader-facing audit. During a run, an optimizer only proposes a reaction id and a valid tuple of synthon ids, and receives one scalar utility per charged oracle call. For score-table suites, exact metrics are attached after the run by joining the queried product ids against the reference score table for the same reaction and objective. Thus "exact" means exact with respect to the reference score table and its stable product ids; it does not mean measured docking, biological activity, synthesis success, toxicity, or clinical relevance.

For docking-surrogate score tables, raw SpaceHASTEN-derived predicted docking scores are minimized. The benchmark converts them to utilities by negating the raw scores, so all optimizers maximize utility and all reported best-utility curves have the same sign convention. For a task with reaction r, objective column c, and queried product-id set Q, top-k recall is computed from the sorted score-table reference $\mathrm{Top}_k(r, c)$ as

$$\text{top-}k \text{ recall} = \frac{|Q \cap \mathrm{Top}_k(r, c)|}{|\mathrm{Top}_k(r, c)|}.$$

Enrichment is the observed top-0.1% hit rate in Q divided by the score-table top-0.1% base rate, and regret is $\max(0, u^\star - \max_{p \in Q} u(p))$, where $u^\star$ is the best table utility for the reaction and objective.

For selectivity tasks, the optimizer still receives a scalar utility for one target. The exact Pareto metrics in the main paper are post-hoc audits of the queried set against two target utilities. The companion implementation `pareto_ga` is the exception: it uses the score table to rank the maintained population by non-dominated sorting and crowding distance, while still paying one benchmark query per proposed product.

A.2. Task Construction and Budget Rule

The reported benchmark uses reaction-specific spaces rather than treating the labels "1M" and "10M" as per-task sizes. The 1M exact-audit tier contains 990,610 unique products after deduplication of product identifiers, spread over 42 reactions. The 10M tier contains 10,022,100 products over the same reaction set. Each scalar suite crosses reactions with objectives: three docking-surrogate targets for the docking track, three pairwise target comparisons for the selectivity track, and synthetic families for diagnostic structure.

Unless explicitly overridden, tasks use the B_r rule from Section 2, applied to each reaction-specific product count $|\mathcal{P}_r|$. Consequently most 1M exact-audit reactions use 1,000 calls, while the largest 1M reactions use 3,890 calls. The 10M suites use 1,000 calls for smaller reactions and the 10,000-call cap for the largest reactions. The fixed-1000 companion ablations in the reproducibility package are included to check that the main GA/Thompson versus random trend is not solely an artifact of this budget rule; they do not replace a full fixed-budget sweep over every method.

A.3. Reaction-Size and Phase Diagnostics

The docking suites are intentionally heterogeneous, so task averages and product-weighted summaries emphasize different parts of the benchmark. In the 1M docking suite, the largest six tasks (the two largest reactions crossed with the three docking targets) are only six of 126 task rows, but they account for 78.5% of product mass. In the 10M docking suite, the same six-task slice accounts for 89.6% of product mass. This is why product-weighted top-100 recall is most useful as a large-reaction audit rather than as a replacement for the task-average benchmark.

Table 6 gives the numeric values behind Figure 1. GA is the best public method in every explicit size bin. The second tier is size-dependent: Thompson is strong on small reactions, while ISE is much more competitive on large reactions. This distinction is hidden by a single task-average row and is amplified by product weighting.

The phase and locality diagnostics support an empirical exploration/exploitation interpretation without claiming that one GA operator is causal. GA reaches large-bin top-100 recall of 0.113 on 1M and 0.115 on 10M by the first 20% of budget, then continues to 0.414 and 0.353 by the end of the budget. ISE is slower initially on large reactions but ends at 0.309 and 0.247. Locality metrics show the same pattern: on the large reactions, GA has the smallest mean distance to exact Top-100 neighborhoods and the highest exact Top-100 hits per run, while random sampling remains far from elite neighborhoods even when it covers many slot ids.

Table 6. Full size-stratified docking leaderboard. Values are task-averaged top-100 recall. “Large” denotes the two largest reactions in each suite crossed with the three docking targets; “small” denotes all remaining reactions.

Method	1M small	1M large	10M small	10M large
Random	0.186	0.010	0.041	0.001
SlotUCB	0.214	0.011	0.059	0.001
Hill climb	0.268	0.027	0.072	0.006
MCTS	0.390	0.050	0.087	0.034
ISE	0.351	0.309	0.093	0.247
Thompson	0.527	0.129	0.230	0.070
GA	**0.657**	**0.414**	**0.319**	**0.353**
FM	0.241	0.052	0.065	0.026
RF	0.194	0.011	0.045	0.002

Table 7. Large-reaction locality summary. “Near Top-100” is mean nearest exact Top-100 Hamming distance for queried products; “Within 2 edits” is the query fraction within two synthon-slot edits of an exact Top-100 product; Hits/run is exact Top-100 products recovered per run.

	1M large reactions			10M large reactions		
Method	Near Top100	Within 2 edits	Hits/run	Near Top100	Within 2 edits	Hits/run
Random	2.275	0.691	1.0	2.558	0.435	0.1
Hill climb	2.251	0.700	2.7	2.548	0.442	0.6
MCTS	2.124	0.788	5.0	2.384	0.588	3.4
ISE	1.849	0.867	30.9	1.985	0.772	24.7
Thompson	2.131	0.759	12.9	2.398	0.558	7.0
GA	**1.490**	**0.949**	**41.4**	**1.603**	**0.911**	**35.3**

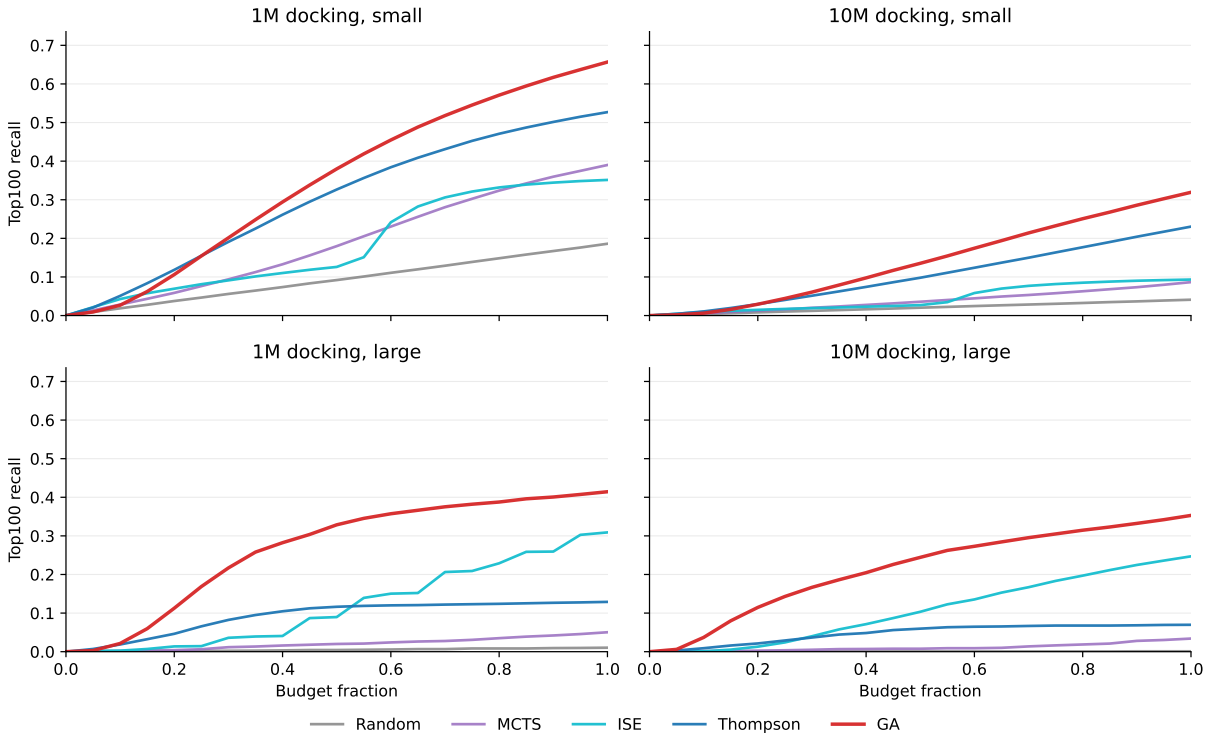

Figure 6. Exact top-100 recall over budget fraction, stratified by suite and reaction size. GA separates early and continues accumulating elite products; ISE starts more slowly but becomes a strong large-reaction reference by the end of the budget.

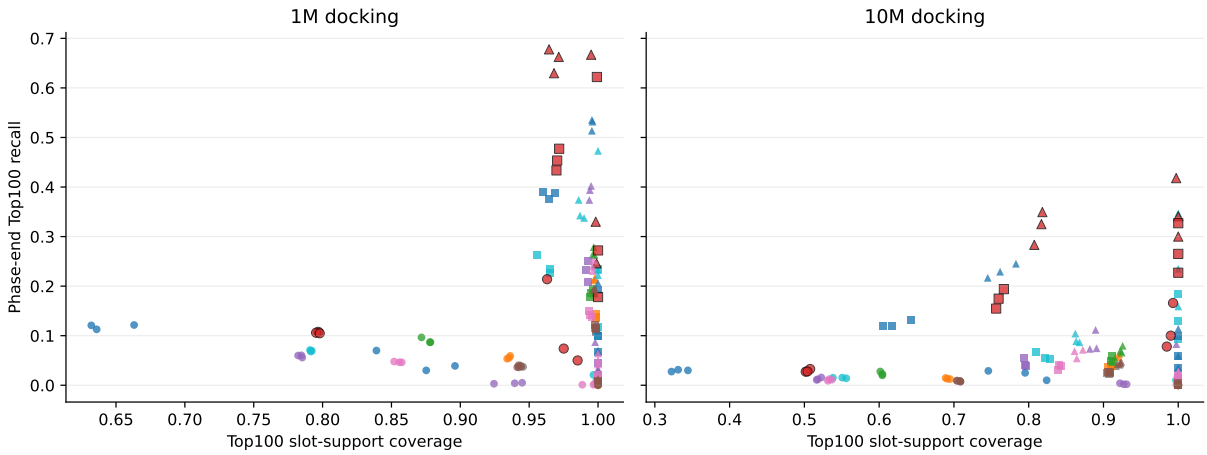

Figure 7. Exploration coverage versus phase-end top-100 recall. Each point aggregates a method, target, size bin, and budget phase. Broad slot-support coverage alone is not sufficient for elite recovery: high-entropy coverage can still miss exact Top-100 products.

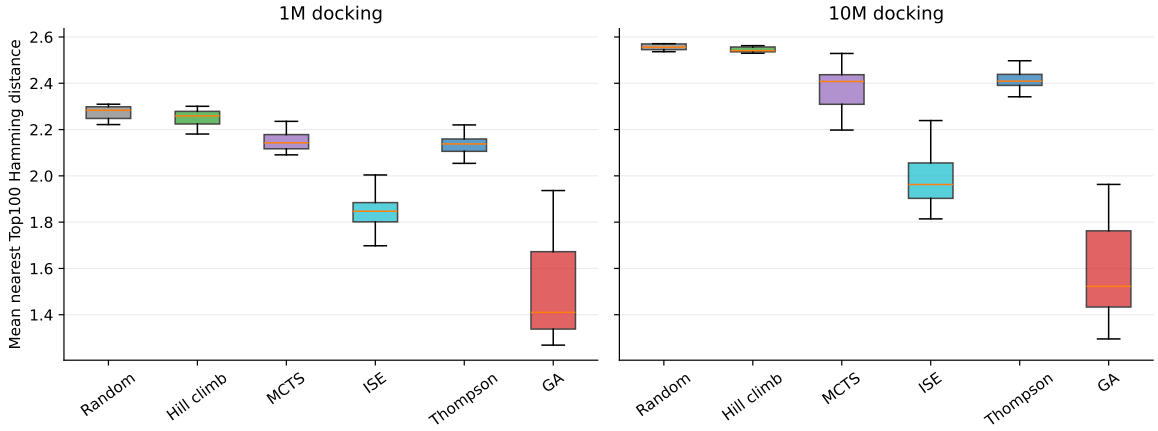

Figure 8. Large-reaction locality diagnostic. The y-axis is the per-run mean Hamming distance from queried tuples to the nearest exact Top-100 tuple. Lower values indicate queries closer to exact elite neighborhoods.

Table 8. Budget-rule sensitivity for the existing fixed-1000 companion ablation. Values are large-reaction task-averaged top-100 recall. The ablation covers random, Thompson, and GA only; smaller reactions already use budget 1,000 under the configured rule.

	1M large reactions		10M large reactions	
Method	Configured	Fixed-1000	Configured	Fixed-1000
Random	0.010	0.002	0.001	0.000
Thompson	0.129	0.068	0.070	0.009
GA	0.414	0.177	0.353	0.037

A.4. Optimizer Details

Table 10 summarizes the optimizer defaults used by the in-repo runner. All methods are instantiated per task and per seed. The benchmark runner wraps optimizers with a unique-query guard: if a proposal has already been evaluated, the wrapper retries the optimizer briefly and then falls back to unseen random sampling. The reported exact tables therefore audit unique charged oracle calls.

Table 9. Objective-level task catalog for the reported benchmark suites. Each row is instantiated over the same 42 reactions unless noted otherwise. Docking rows use SpaceHASTEN-derived LightGBM surrogate predictions, not measured docking.

Suite/task	Explanation	Role
Docking: KIF11	Kinesin-like protein 1 target from the SpaceHASTEN DUD-E validation study: PDB `3CJO`, UniProt `Q02127`, ChEMBL `CHEMBL1966`; redocking cutoff -8.7 (Kalliokoski et al., 2025).	Scalar objective $u = -\hat{d}_{\mathrm{KIF11}}(p)$; exact top-$k$, enrichment, and regret audit.
Docking: PYRD	Dihydroorotate dehydrogenase target from the same study: PDB `1D3G`, UniProt `P52732`, ChEMBL `CHEMBL4581`; redocking cutoff -12.5.	Scalar objective $u = -\hat{d}_{\mathrm{PYRD}}(p)$; exact top-$k$, enrichment, and regret audit.
Docking: TGFR1	TGF-beta receptor type I target from the same study: PDB `3HMM`, UniProt `P36897`, ChEMBL `CHEMBL4439`; redocking cutoff -9.7.	Scalar objective $u = -\hat{d}_{\mathrm{TGFR1}}(p)$; exact top-$k$, enrichment, and regret audit.
Selectivity: KIF11/PYRD	Two-objective reference $(u_{\mathrm{KIF11}}, u_{\mathrm{PYRD}})$, with scalar search run on KIF11.	Post-hoc exact Pareto-front recall, hypervolume ratio, and epsilon-regret audit.
Selectivity: KIF11/TGFR1	Two-objective reference $(u_{\mathrm{KIF11}}, u_{\mathrm{TGFR1}})$, with scalar search run on KIF11.	Post-hoc exact Pareto-front recall, hypervolume ratio, and epsilon-regret audit.
Selectivity: PYRD/TGFR1	Two-objective reference $(u_{\mathrm{PYRD}}, u_{\mathrm{TGFR1}})$, with scalar search run on PYRD.	Post-hoc exact Pareto-front recall, hypervolume ratio, and epsilon-regret audit.
Synthetic: additive	$u_{\mathrm{add}}(s) = \sum_i a_i(s_i)$, with deterministic random-continuous synthon utilities.	Tests whether methods exploit reusable high-value synthons.
Synthetic: pairwise epistasis	$u_{\mathrm{pair}}(s) = \sum_i a_i(s_i) + \alpha \sum_{i<j} e_i(s_i)^\top e_j(s_j)$, with $\alpha = 0.3$ and 32-dimensional deterministic embeddings.	Tests whether methods model slot–slot interactions.
Synthetic: sparse3	$u_{\mathrm{sparse3}}(s) = \sum_i a_i(s_i) + \sum_{\{i,j,k\}} b_{ijk}(s_i, s_j, s_k)$, where bonuses are nonzero only for a hashed sparse subset.	Tests higher-order structure without materializing all interactions.
Synthetic: constraint	$u_{\mathrm{con}}(s) = \sum_i a_i(s_i)$ if all binary feasibility checks pass, and 0 otherwise. This family appears only in the 1M diagnostic suite.	Tests whether search avoids infeasible regions while preserving additive signal.

Table 10. Optimizer implementation details and default hyperparameters. These are scalable baseline implementations rather than heavily tuned chemistry-specific agents.

Method	Policy summary	Default settings
Random	Uniformly samples a valid unseen synthon tuple from the reaction space.	No learned state beyond the random generator and the unique-query guard.
Thompson	Maintains independent empirical mean rewards per slot and synthon. After warmup, samples an optimistic value for every synthon from a normal distribution centered at its empirical mean and chooses the best synthon per slot.	Warmup $\max(4, m)$, where m is the number of slots; posterior sampling scale $1/\sqrt{n+1}$ for synthon count n. This reported row is a simple benchmark-native variant; unlike Klarich et al. (Klarich et al., 2024), it does not warm up every reagent.
Thompson reagent prior	Companion implementation closer to Klarich et al.: targets every synthon during warmup with random partners, estimates a global warmup mean and standard deviation, then replays per-synthon warmup scores through Bayesian normal updates.	Default one scored query per synthon warmup trial is charged to the benchmark budget; per-synthon warmup trials default to 3; duplicate posterior proposals and random fallbacks are instrumented.
SlotUCB	Slotwise empirical-mean plus optimism heuristic.	Warmup $\max(4, m)$; UCB score $\hat{\mu} + 1/\sqrt{n+1}$ with default $\beta = 1.0$.
Hill climb	Samples an initial elite set, then mutates synthon slots of rank-weighted elite parents. If progress stalls, restarts from random unseen tuples.	Warmup $\max(8, 3m)$; elite size 16; mutation rate 1.0 spread across slots; restart patience 64; max resample 10,000.
MCTS	Treats partial synthon assignment as a tree. It uses UCB selection, progressive widening, and occasional random rollout choices; completed paths are valid products.	Exploration 1.4; widening $2.0\,N^{0.5}$; rollout randomness 0.10; max resample 10,000.
Reaction Metropolis	RosettaAMRLD-inspired companion baseline that keeps one accepted current tuple, proposes local slot mutations, ranks proposal pools by product Tanimoto similarity when available, and accepts or rejects with a Metropolis rule over benchmark utility.	Temperature 0.10; proposal pool 64; geometric sampling ratio 0.01–0.20 with 0.002 rejection increments; two-slot mutation probability 0.15; dynamic schedule off after half the budget.
Genetic algorithm	Maintains a population of synthon tuples. Children are produced by slotwise crossover between elite parents, optional one-slot mutation, and random immigrants.	Population 64; mutation probability 0.3; random immigrant probability 0.1; 64 child retries; top half of population used as elites.
Factorization machine	Online low-rank model over synthon ids. It updates a bias, per-synthon linear weights, and latent factors after each observed score, then selects the best candidate from a random pool.	Latent dimension 8; learning rate 0.01; $L_2 = 10^{-4}$; pool size 32; $\epsilon = 0.15$; warmup $\max(8, 2m)$; clipped errors/weights/latents.
RF factorized	Fits one random-forest regressor per slot using synthon Morgan fingerprints and combines per-slot predictions additively through greedy slot selection.	256-bit Morgan fingerprints, radius 2; 8 trees per slot; warmup 12; refit every 200 observations; $\beta = 0.5$ controls the exploration bonus added to predicted slot utility.
SpaceHASTEN-style iterative	SpaceHASTEN-style active-screening baseline: seed with random queries, periodically fit a product-fingerprint regressor, expand local synthon neighborhoods around queried and predicted elites, and greedily query predicted high-utility candidates.	Warmup 32; refit interval 64; candidate pool 512; 16 observed elites and 16 predicted elites; random fraction 0.25; ExtraTrees with 64 trees; 512-bit Morgan product fingerprints plus 128 slot-hash features.
Pareto GA	Auxiliary selectivity optimizer. It uses the GA operators above but maintains the population with NSGA-II-style non-dominated sorting and crowding distance (Deb et al., 2002) over the score-table objective tuple.	Population 64; mutation probability 0.3; random immigrant probability 0.1; 64 child retries; Pareto ranking refreshed every 16 tells.

A.5. Surrogate-Oracle Provenance and Validation

The docking-surrogate tasks use SpaceHASTEN-derived LightGBM (Ke et al., 2017) prediction tables, not measured docking at benchmark time. The source scores used for surrogate training were produced by converting SMILES to 3D structures with Phase/LigPrep and docking with Glide from Schrödinger Suite 2023-4, following the SpaceHASTEN workflow (Kalliokoski et al., 2025; Friesner et al., 2004). Training uses product ECFP4 fingerprints computed from SMILES with radius 2 and 2048 bits, stored as packed bytes and unpacked for LightGBM. The final model package uses LightGBM GPU regression with up to 4000 boosting rounds and early stopping, and applies $3\times$ sample weight to the lowest 10% docking scores in the training split to prioritize tail fitting, since downstream search cares about the most negative scores. The split is random with seed 0: 70% train, 15% validation, and 15% test.

Table 12 reports the held-out metrics used to justify treating these models as structured benchmark oracles. Rank correlations are moderate to good, but elite-tail recall is imperfect. This is why the paper frames the docking-surrogate tracks as surrogate-oracle search tasks and not as evidence of biological activity.

Table 11. Summary LightGBM surrogate validation used by the main text. "Source rows" are retained SpaceHASTEN/Orion Glide-docked product rows after filtering sentinel/non-docking values; "train rows" are the 70% LightGBM training split. Spearman and tail recall are measured on held-out test products; tail recall is recall of the lowest 0.1% docking-score tail.

Target	Source rows	Train rows	Spearman	Tail recall
KIF11	853,505	597,453	0.712	0.279
PYRD	991,545	694,081	0.794	0.154
TGFR1	995,897	697,127	0.713	0.153

A.6. Independent GNINA Redocking Audit

The benchmark oracle for the docking tracks remains the SpaceHASTEN-derived LightGBM surrogate described above. To test whether surrogate-selected products are still enriched under an independent docking stack, we redocked a 2700-molecule audit set from the 100M scale-up: for each of KIF11, PYRD, and TGFR1, the set contains the 100 products with best surrogate utility for each of the eight non-random methods plus 100 uniformly sampled random products. The target sites follow the same target provenance as the SpaceHASTEN labels, but docking is rerun with GNINA/AutoDock Vina (McNutt et al., 2021; Trott & Olson, 2010) rather than reusing the original Glide scores.

Docking used RDKit conformer generation (Landrum, 2025) with no protonation changes, at most three tautomers per molecule, one protomer, five returned poses, Vina exhaustiveness 8, and GNINA CNN rescoring on GPU workers. Lower Vina affinity is better; higher GNINA CNN affinity is better. All 27 target-method shards completed, with 2691 successful molecules out of 2700 and zero no-protonation mismatches in the verification pass. Pose files are retained with the audit artifacts for inspection.

Table 13 and Figure 9 summarize the enrichment result. Every non-random method beats random on median Vina affinity and median CNN affinity, and the bootstrap intervals for the paired method-versus-random differences exclude zero for all non-random methods. Thompson sampling has the best aggregate redocking performance across the two metrics, with the strongest Vina medians on KIF11 and PYRD; MCTS has the strongest TGFR1 Vina median. The audit therefore supports the claim that the surrogate-selected sets are substantially better than random under independent redocking. It should not be read as molecule-level calibration: within-target surrogate/GNINA rank correlations are weak to modest, so the redocking result is reported as set-level enrichment evidence.

Table 12. Held-out LightGBM surrogate validation. Lower raw docking scores are better; tail recall is computed for the most negative true scores.

Target	Input rows	Used rows	Test n	RMSE	MAE	R^2	Pearson	Spearman	Lowest 0.1%	Lowest 1%
KIF11	853,539	853,505	128,026	0.642	0.503	0.537	0.734	0.712	0.279	0.383
PYRD	992,001	991,545	148,732	0.734	0.541	0.619	0.788	0.794	0.154	0.281
TGFR1	996,030	995,897	149,385	0.707	0.542	0.518	0.723	0.713	0.153	0.244

Table 13. Independent GNINA/AutoDock Vina redocking audit on 2700 products sampled from the 100M docking-surrogate scale-up. Vina values are kcal/mol with lower better; CNN affinity values are GNINA rescoring outputs with higher better. The final column reports median Vina difference versus random and median CNN-affinity difference versus random for the same target.

Method	Target	Success	Median Vina	Best Vina	Median CNN	Best CNN	Δ vs random
FM	KIF11	99/100	-8.30	-10.96	6.46	8.23	-0.39 / 0.40
GA	KIF11	97/100	-8.45	-10.11	6.88	8.19	-0.54 / 0.82
Hill climb	KIF11	100/100	-8.78	-10.91	6.42	7.95	-0.87 / 0.36
MCTS	KIF11	99/100	-8.75	-11.06	6.59	8.33	-0.83 / 0.53
RF	KIF11	100/100	-8.75	-10.74	6.43	7.70	-0.84 / 0.37
SlotUCB	KIF11	100/100	-8.95	-10.74	6.38	7.70	-1.04 / 0.32
ISE	KIF11	97/100	-8.64	-10.31	6.76	8.36	-0.73 / 0.70
Thompson	KIF11	100/100	-9.25	-10.37	7.05	8.57	-1.34 / 0.99
Random	KIF11	100/100	-7.91	-10.71	6.06	8.00	0.00 / 0.00
FM	PYRD	100/100	-13.06	-17.13	7.71	8.49	-4.04 / 0.99
GA	PYRD	100/100	-14.37	-18.34	7.83	8.70	-5.35 / 1.12
Hill climb	PYRD	100/100	-12.68	-16.73	7.46	8.38	-3.66 / 0.75
MCTS	PYRD	100/100	-12.76	-16.51	7.66	8.67	-3.74 / 0.94
RF	PYRD	100/100	-12.45	-16.78	7.48	8.59	-3.43 / 0.76
SlotUCB	PYRD	100/100	-12.81	-16.78	7.46	8.59	-3.79 / 0.75
ISE	PYRD	100/100	-14.29	-17.87	7.77	8.70	-5.27 / 1.06
Thompson	PYRD	100/100	-14.56	-18.34	7.82	8.38	-5.54 / 1.10
Random	PYRD	100/100	-9.02	-12.31	6.72	8.39	0.00 / 0.00
FM	TGFR1	100/100	-10.30	-12.80	7.79	8.91	-1.40 / 0.86
GA	TGFR1	100/100	-10.24	-12.60	8.01	9.44	-1.34 / 1.07
Hill climb	TGFR1	99/100	-10.14	-12.14	7.91	8.89	-1.24 / 0.98
MCTS	TGFR1	100/100	-11.02	-13.00	8.16	9.05	-2.12 / 1.22
RF	TGFR1	100/100	-10.56	-13.78	7.74	9.10	-1.66 / 0.80
SlotUCB	TGFR1	100/100	-10.81	-13.78	7.81	9.10	-1.91 / 0.87
ISE	TGFR1	100/100	-10.35	-12.29	7.90	9.44	-1.45 / 0.97
Thompson	TGFR1	100/100	-10.87	-13.19	8.12	9.15	-1.97 / 1.18
Random	TGFR1	100/100	-8.90	-12.45	6.94	9.05	0.00 / 0.00

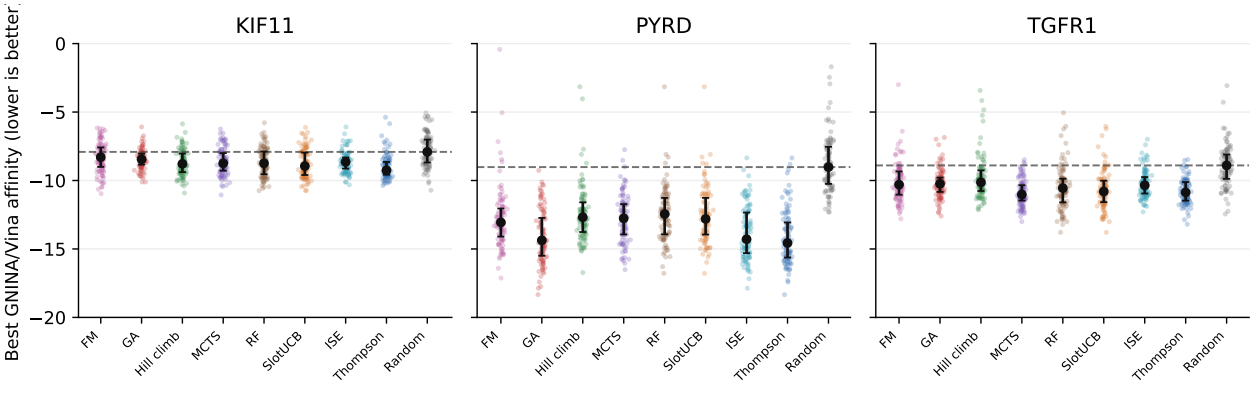

Figure 9. Independent GNINA/AutoDock Vina redocking audit by target and method. Each panel compares the redocked Vina affinity and GNINA CNN affinity distributions for the stratified 100M sample. Non-random methods shift toward better Vina and CNN affinity than the target-matched random baseline.

A.7. Statistical and Companion Artifacts

The reproducibility package includes statistical companion tables for the main exact-audited suites: method summaries, reaction-level bootstrap confidence intervals, product-weighted summaries, size-stratified summaries, and paired differences. The main paper cites only the highest-signal values, but the CSV files are included so readers can inspect reaction-level heterogeneity and product-size weighting directly. The size/phase diagnostic CSVs behind Figures 1 and 6 are included with the same convention, including the run-level traces used for coverage and locality summaries.

Several additional companion artifacts are included for context and future planning (Table 14). The ISE rows report product-weighted docking audit values for the method now shown in the main docking table; the Pareto GA run demonstrates a true multi-objective selectivity optimizer; and the fixed-1000 ablations probe budget sensitivity for random, Thompson, and GA. The former 100M stress subset has been superseded by the 100M docking feasibility run in Table 3. Table 8 gives the corresponding large-reaction size split for the fixed-1000 companion ablation.

Table 14. Compact companion artifacts included in the reproducibility package. Exact-audited values are task-averaged over 126 tasks and five seeds unless otherwise noted.

Suite × method group	Method	Top-100 recall	Product-weighted Top-100	Exact regret
Pareto GA 1M selectivity	`pareto_ga`	0.531	0.305	0.037
Iterative 1M docking	`spacehasten_iterative`	0.349	0.318	0.108
Iterative 10M docking	`spacehasten_iterative`	0.101	0.231	0.275
Fixed-1000 1M docking	`genetic_algorithm`	0.634	0.280	0.031
Fixed-1000 1M docking	`thompson`	0.505	0.166	0.047
Fixed-1000 1M docking	`random`	0.177	0.041	0.230
Fixed-1000 10M docking	`genetic_algorithm`	0.306	0.066	0.112
Fixed-1000 10M docking	`thompson`	0.220	0.031	0.165
Fixed-1000 10M docking	`random`	0.039	0.004	0.434

The Pareto GA companion run also reaches exact selectivity frontier recall 0.745, hypervolume ratio 0.987, and Pareto epsilon regret 0.160.

The 100M scale-up includes two 126-task seed-0 grids: docking, with all 42 reactions crossed with the three docking-surrogate targets, and synthetic scale, with additive, pairwise, and sparse3 objectives for the same 42 reactions. Docking is included as compact aggregate, exact-audit, statistics, and runtime supplements; synthetic is included as compact aggregate, task, and runtime supplements rather than raw trace files. The GNINA redocking audit adds compact molecule-level results, method-versus-random summaries, manuscript figures and tables, sample manifests, run manifests, and pose-file paths.

A.8. Reproducibility Package Contents

The anonymous reproducibility package is designed for inspection and quick execution, not for distributing the full raw run tree. It contains the paper PDF and source, an anonymous code archive, exact and aggregate tables, figure data files, metadata cards, run manifests, command logs, and checksum manifests. The compact package intentionally excludes tens of gigabytes of raw per-task traces and large score tables. Package-relative checksums and archive-internal checksums are included so readers can verify that the exported files are internally consistent. The anonymous review artifact is available at `https://anonymous.4open.science/r/review-artifact-2026-14-1CA4`.

For the workshop submission, the code artifact carries the compact reproducibility surface, including the files covered by the code checksum manifest. A separate result-only archive can mirror these compact files if the live form asks for a dataset URL, but the GenBio submission path does not rely on a second download.

The largest excluded local paths are the paper-v1 raw run tree (38.8GB), the additional raw run tree (7.0GB), the 10M docking score table (491MB, SHA256 prefix `a93fd68c8746`), and the 100M docking score table (5.0GB, SHA256 prefix `4deee1c8caa8`). The reproducibility artifact includes compact summaries for these resources and records their checksums. Final public URLs and DOI will be added after de-anonymization.

Table 15. Reader-facing reproducibility surface.

Item	Reader-facing status
Paper rebuild	LaTeX source and style files included.
Unit/smoke tests	Quickstart command uses plugin isolation and headless plotting; optional-dependency tests skip cleanly.
Toy end-to-end run	Included via `scripts/run_toy_e2e.sh`.
Paper figures	PDFs/PNGs plus CSV/JSON data files included.
Exact tables	Included for 1M docking, 1M selectivity, 10M table-backed docking, and compact 100M docking scale-up audit.
GNINA redocking audit	Compact 2700-molecule results, method-versus-random summaries, figures, tables, run manifests, and pose-file pointers included.
Full raw rerun	Code/scripts included; large score tables are represented by checksummed manifests and regeneration scripts.
100M scale-up	Docking aggregate/exact/statistical/runtime supplements and synthetic aggregate/task/runtime supplements included.