Generation vs. Retrieval in Molecular Docking: Zero-Parameter Baselines Match Learned Generators Across 67 Targets

Elliot Tower

Independent researcher
elliot@elliottower.ai

Abstract

Background: Molecular optimization benchmarks evaluate learned generators—reinforcement learning, genetic algorithms, diffusion models—under fixed oracle budgets. Leaderboard entries introduce learned parameters whose cost must be justified against simpler alternatives, yet no standard nonparametric floor exists for docking objectives.

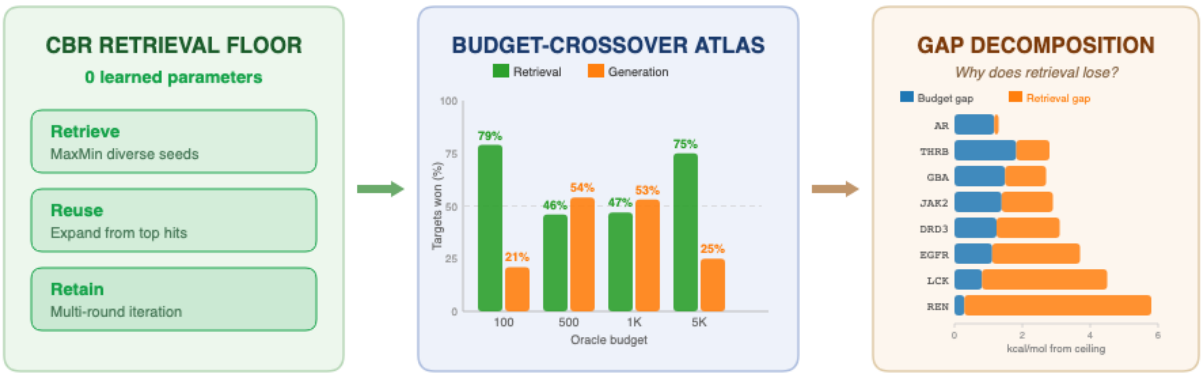
Methods: We constructed three zero-parameter retrieval methods instantiating successive steps of the case-based reasoning (CBR) cycle [19]: retrieval-only (Retrieve), expansion (Retrieve + Reuse), and iterative refinement (Retrieve + Reuse + Retain), using only Morgan fingerprints and Tanimoto similarity. We evaluated these across 58 DOCKSTRING targets and 9 TDC docking targets (10 seeds per condition, bootstrap 95% CIs) and constructed a budget-crossover atlas mapping where retrieval stops winning. An oracle-ranker gap decomposition separates the gap between retrieval and the library ceiling into a retrieval gap and a budget gap. All parameters were pre-registered under commit SHAs `94989e0` and `bacc0ff` before any docking calls.

Results: On TDC's DRD3, expansion places first at 100 oracle calls (-8.00 kcal/mol vs SMILES-LSTM -7.59). Across 57 standard DOCKSTRING targets, iterative refinement wins 79% at budget 100 (CI: 68–90%) and 75% at budget 5000 (CI: 63–86%), while expansion wins at intermediate budgets (54–67%). Uniform random sampling from the library nearly matches targeted retrieval (AUC -11.66 vs -11.85), indicating that library composition dominates low-budget performance. The gap decomposition reveals that 52% of the ceiling gap is budget-limited (a surrogate ranker would help) and 48% is retrieval-limited (generation needed).

Conclusions: Zero-parameter fingerprint retrieval establishes a nonparametric floor that learned generators surpass only beyond a few hundred oracle calls. The budget-crossover atlas provides a per-target map of where generation begins to earn its computational cost. We propose that future docking benchmarks report a retrieval floor alongside all learned methods, as information retrieval benchmarks report BM25.

Scientific contribution: We introduce the first systematic nonparametric floor for molecular docking optimization, evaluated across 67 targets with pre-registered protocols. The budget-crossover atlas and oracle-ranker gap decomposition provide new diagnostic tools for separating retrieval-solvable from generation-requiring targets.

Keywords: molecular docking, fingerprint retrieval, case-based reasoning, zero-parameter baseline, budget-crossover, oracle efficiency, DOCKSTRING, Therapeutics Data Commons, molecular optimization, benchmarking



Graphical abstract. Zero-parameter fingerprint retrieval (CBR cycle: Retrieve, Reuse, Retain) wins 79% of 67 docking targets at budget 100. The budget-crossover atlas maps where generation overtakes retrieval; the gap decomposition separates budget-limited from retrieval-limited targets.

1 Background

Molecular optimization benchmarks measure a method's ability to propose high-affinity ligands under a fixed oracle budget [1–3]. Leaderboard entries typically employ reinforcement learning [5], genetic algorithms [6], diffusion models [7], or LLM agents [8]. Each introduces learned components whose parameter cost must be justified against simpler alternatives.

Gao et al. [3] established this principle for property-based objectives: their Practical Molecular Optimization (PMO) benchmark showed that older methods consistently outperform newer ones, and introduced AUC over oracle calls as a standardized metric. Thomas et al. [4] extended the critique to docking objectives specifically. Both findings point toward a sharper question: *what is the nonparametric floor*—the best performance achievable with zero learned parameters—and at what budget does each optimizer surpass it?

We construct this floor using the classical case-based reasoning (CBR) cycle [19]. CBR decomposes problem-solving into four steps: *Retrieve* relevant cases, *Reuse* their solutions, *Revise* solutions for the new problem, and *Retain* successful solutions for future use. We instantiate three of these steps as molecular retrieval methods of increasing sophistication—retrieval-only, expansion, and iterative—and map their performance against 58 DOCKSTRING targets [2] and 9 TDC docking targets [1].

The result is a *budget-crossover atlas*: a per-target map showing where retrieval stops winning and generation begins to earn its computational cost. Every generator's result becomes an answer to one question: how many oracle calls did it take to beat the free retrieval floor?

Our contributions are: (1) three deterministic retrieval methods that instantiate the CBR cycle with zero learned parameters; (2) a budget-crossover atlas across 67 targets mapping where each method wins; (3) an oracle-ranker gap decomposition identifying whether each target needs generation or better ranking; and (4) evidence that random library sampling is itself a strong baseline, contextualizing the value of targeted retrieval.

2 Related Work

2.1 Molecular optimizers as CBR

We adopt the following definitions for each CBR step in the molecular optimization context: *Retrieve*—select candidate molecules from an existing library (by diversity, similarity, or surrogate score); *Reuse*—apply information from evaluated candidates to guide subsequent selection (e.g., expanding from top-scoring hits); *Revise*—modify molecular structures to improve properties (via mutation, crossover, or learned generation); *Retain*—store evaluated molecules and their scores for use in later rounds.

Table 1 maps molecular optimization methods onto these four steps. Every optimizer performs some subset; what varies is whether each step uses learned components or deterministic heuristics. Our retrieval methods use only deterministic heuristics, establishing the nonparametric floor against which learned components can be evaluated.

Table 1: Molecular optimization methods mapped to the CBR cycle. **R**=Retrieve, **U**=Reuse, **V**=Revise, **T**=Retain. Checkmarks indicate which CBR steps each method performs. Our methods (bottom three rows) use no learned components.

Method	R	U	V	T	Mechanism
REINVENT [5]			✓	✓	RL policy generates + retains high-scoring SMILES
Graph-GA [6]			✓	✓	Crossover/mutation revises; fitness retains
GP-BO [13]	✓	✓			GP surrogate retrieves + reuses latent candidates
MolPAL [12]	✓	✓			Surrogate-guided pool retrieval + reuse
Saturn [14]		✓	✓	✓	Memory-augmented RL reuses + revises + retains
SEISMO [8]		✓	✓		LLM reuses trajectory context to revise proposals
MolMem [28]	✓		✓	✓	Exemplar retrieval + RL revision + skill retention
f-RAG [27]	✓		✓		Fragment retrieval augments generative model
Retrieval (ours)	✓				MaxMin diverse seed selection only
Expansion (ours)	✓	✓			Seeds + neighbor retrieval from top hits
Iterative (ours)	✓	✓		✓	Multi-round expansion; scores carry across rounds

2.2 Oracle-budget benchmarks

TDC evaluates methods at budgets of 100–5000 calls against 9 receptor targets [1]. DOCKSTRING provides precomputed AutoDock Vina scores for ∼260K molecules across 58 targets, enabling benchmarking without live docking [2]. PMO standardized evaluation of 25 methods under fixed budgets using AUC over oracle calls on property-based objectives [3].

2.3 Concurrent work

PMO-Dock [9] brings PMO-style evaluation to docking objectives with 23 tasks split into validation and test sets. Its central finding—that hyperparameters optimized on validation targets fail to

transfer to test targets—exposes fragility in current learned methods. PMO-Dock benchmarks four generators (Saturn, GenMol, Genetic-GFN [24], Chemlactica [25]) but includes no retrieval baseline. Our work fills this gap: the retrieval floor is the control their benchmark lacks.

SEISMO [8] uses an LLM agent for online molecular optimization, achieving 2–3× higher AUC than baselines and reaching near-maximal scores within 50 oracle calls. SEISMO operates in the same low-budget regime we study, and its authors flag LLM memorization of known actives as a potential confound. Our retrieval methods provide the memorization-free, fully reproducible floor that SEISMO's gains should be measured against: any improvement over deterministic retrieval is attributable to the LLM's chemical priors, not its memory of training data. The three contributions are complementary: PMO-Dock establishes that learned generators are fragile; we establish the floor they must beat; SEISMO demonstrates LLM-based optimization that our retrieval floor can benchmark.

MolMem [28] introduces a dual-memory agentic RL framework for sample-efficient molecular optimization: a static exemplar memory retrieves relevant molecules for cold-start grounding, and an evolving skill memory distills successful trajectories into reusable strategies. The exemplar retrieval component maps onto CBR Retrieve, while the skill memory maps onto Retain. MolMem targets the same low-budget regime we study, and our retrieval floor provides the nonparametric baseline against which its learned memory components should be evaluated.

2.4 Pool-based active learning

MolPAL learns a surrogate model to prioritize which library molecules to dock, operating at budgets of 10^4–10^6 calls [12]. Graff et al. extended this with design space pruning, which irreversibly removes low-scoring candidates to reduce inference cost [26]. Both approaches require enough labeled data to train a surrogate. Our methods occupy the regime below 10^3 calls, where too few labeled examples exist to train any surrogate. The oracle-ranker gap analysis (Section 5.4) identifies which targets would benefit from surrogate-guided ranking versus which require generation of molecules outside the library.

2.5 Case-based reasoning

Aamodt and Plaza [19] formalized the four-step CBR cycle (Retrieve, Reuse, Revise, Retain). CBR has been applied to knowledge graph question answering [10, 11] and drug repositioning, but not to molecular docking optimization. Das et al. [11] showed that a no-training CBR baseline achieves strong HITS@1 on knowledge base tasks when the case library covers the query distribution—the same mechanism that drives our retrieval floor on docking benchmarks.

3 Methods

3.1 Three retrieval variants

We instantiate three levels of the CBR cycle, each adding one step to the previous:

Retrieval. Select n_{seed} maximally diverse molecules from the library via MaxMin diversity picking [18] on Morgan fingerprints (radius 2, 2048 bits, ECFP4 [15]). Filter through a med-chem gate (PAINS + Lipinski + SA ≤ 6.0 [16]); at budget 100, 60% of seeds pass (9 of 15), rising to 79% at budget 5000 (118 of 150). Dock all passing seeds. This is pure CBR Retrieve: no reuse of scores, no expansion, no iteration.

Expansion. After seeding, identify the top-5 seeds by docking score (Reuse). Retrieve their nearest neighbors by Tanimoto similarity. Filter and dock until the budget is exhausted. Scores from the seed phase carry into the final ranking, so expansion reuses seed-phase information.

Iterative. Extend expansion with multi-round refinement (Retain): after seeding with 10 diverse molecules, perform 3 rounds of expand-from-best. Each round selects the top-3 molecules and retrieves their nearest filtered neighbors, splitting the remaining budget equally across rounds. Scores accumulate across all rounds, so good molecules found early persist into later rankings.

All parameters are drawn from pre-registered allocation tables (Table 2).

Table 2: Pre-registered seed allocation by oracle budget. Expansion uses the left column; iterative uses the right.

Budget B	Expansion seeds	Iterative seeds
100	15	10
500	40	25
1000	60	40
5000	150	100

3.2 AUC over oracle budget

Following PMO [3] and SEISMO [8], our primary metric is the area under the optimization curve. Let $s_B = \max_{i \le B} F(m_i)$ be the best docking score found within budget B. We evaluate at $B \in \{100, 500, 1000, 5000\}$ and compute AUC via trapezoidal integration:

$$\text{AUC} = \frac{1}{B_{\max}} \int_0^{B_{\max}} s_B \, dB \approx \frac{1}{5000} \sum_{k=1}^{3} \frac{(B_{k+1} - B_k)(s_{B_k} + s_{B_{k+1}})}{2}$$

Each experiment is replicated with 10 RNG seeds; we report mean ± std and bootstrap 95% confidence intervals on crossover percentages (10,000 resamples).

3.3 Pre-registration

All parameters—fingerprint type, seed allocations, filter thresholds, RNG seeds, and the retrieval algorithm—were committed under SHA `94989e0` (expansion) and `bacc0ff` (iterative) before any real docking calls. We derived one method from each applicable step of the CBR cycle; Revise was excluded because it requires learned parameters. No post-hoc method selection or hyperparameter tuning was performed. The pre-registration document, deviation log, and frozen code are available in the repository.

4 Experimental Setup

DOCKSTRING: 58-target precomputed atlas. We run all three retrieval variants against 58 DOCKSTRING targets [2] using precomputed AutoDock Vina scores over ∼260K molecules from ExCAPE-DB. Budgets of 100, 500, 1000, and 5000 calls are evaluated with 10 replicate seeds

(20260701–20260710). The oracle is a lookup table, so the entire sweep requires zero docking compute.

TDC: 9-target live docking validation. We validate against 9 TDC docking targets using live AutoDock Vina [17] on ZINC 250K (249,455 molecules). DRD3 (PDB: 3PBL) serves as the primary comparison against TDC leaderboard methods.

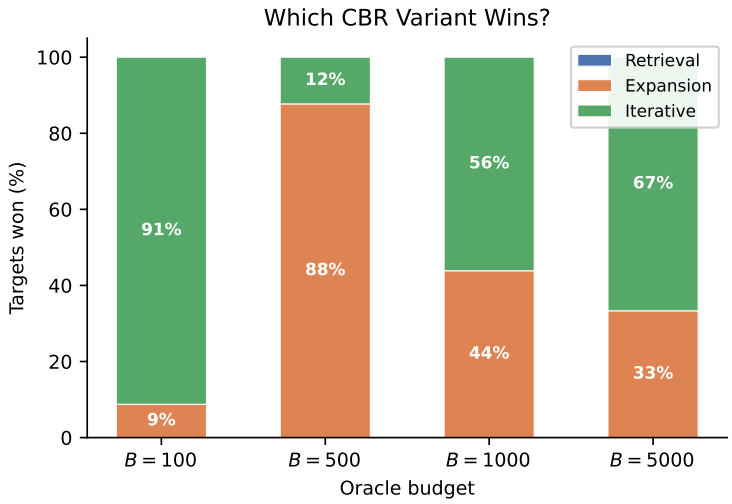


Figure 1: Which CBR variant wins at each oracle budget across 57 standard DOCKSTRING targets (10 seeds). Iterative dominates at $B = 100$ (91%) and $B = 5000$ (67%); expansion wins at intermediate budgets. Retrieval-only never wins.

5 Results

5.1 AUC over oracle budget

Table 3 reports the primary metric: AUC over oracle budget for each method, aggregated across all 57 standard DOCKSTRING targets. More negative AUC indicates better overall performance across the budget range.

Table 3: AUC over oracle budget (top-1 docking score, kcal/mol, trapezoidal integration over $B \in \{100, 500, 1000, 5000\}$). Mean ± std across 57 DOCKSTRING targets, each averaged over 10 seeds.

Method	Mean AUC
Random	-11.66 ± 0.84
Retrieval	-10.35 ± 0.73
Expansion	-11.85 ± 0.86
Iterative	-11.91 ± 0.89

The iterative variant achieves a 15.1% improvement over retrieval-only (AUC difference of −1.56 kcal/mol), while expansion performs within 0.5% of iterative. Neighbor expansion accounts for nearly all of the AUC improvement; multi-round iteration adds a marginal further gain.

The most striking result in Table 3 is the random baseline. Uniform random sampling from the 260K-molecule library achieves AUC −11.66, within 1.6% of expansion (−11.85). This indicates that library composition—the fact that ExCAPE-DB was curated from bioactivity databases—drives most of the performance at low budgets. Targeted retrieval adds only a marginal improvement over drawing molecules at random from a well-curated library.

The random–expansion crossover is itself budget-dependent: at budget 100, random beats expansion on 51 of 57 targets because 100 random draws from 260K molecules sample more chemical diversity than 15 MaxMin seeds plus their neighbors. At budget 5000, expansion wins 53 of 57 targets because neighbor expansion from top hits concentrates the remaining budget on productive regions. The crossover occurs between 500 and 1000 calls.

5.2 TDC DRD3: the nonparametric floor in context

Table 4 compares expansion to TDC leaderboard methods on DRD3. At 100 calls, expansion places first (−8.00 vs SMILES-LSTM −7.59). At 500 calls, Graph-GA overtakes it by 0.29 kcal/mol. At 5000 calls, Graph-GA scores −14.81—far beyond the best-in-data ceiling of −12.08—producing novel molecules absent from ZINC 250K. The crossover lies between 100 and 500 calls on this target.

Table 4: Expansion vs. TDC DRD3 leaderboard. Top-100 mean docking score (kcal/mol, lower is better). Best-in-data (−12.08) bounds any retrieval method.

Method	100 calls	500 calls	5000 calls
Expansion (ours)	**−8.002**	−9.746	−10.672
Graph-GA [6]	−7.222	**−10.036**	**−14.810**
SMILES-LSTM [20]	−7.594	−9.419	—
GCPN [21]	3.860	−8.119	—
MARS [22]	−5.928	−7.278	—
MolDQN [23]	−5.178	−6.357	—
Best-in-data		−12.080	

5.3 Crossover atlas (57 standard targets)

Exploration–exploitation crossover. At budget 100, iterative refinement (10 seeds, 3 expansion rounds) wins on 45 of 57 targets (79%, 95% CI: 68–90%) over expansion (15 seeds, single expansion). At budget 500, the relationship reverses: expansion (40 seeds) wins on 54% of targets (CI: 42–67%) because broader initial coverage discovers more diverse hit neighborhoods. This micro-scale crossover within the retrieval floor itself parallels the macro-scale crossover between retrieval and generation at higher budgets.

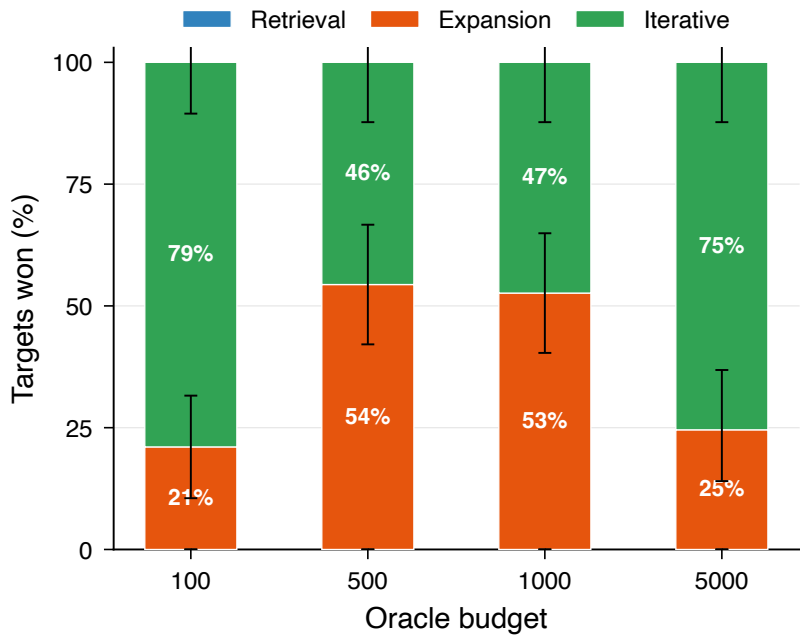


Figure 2: Fraction of 57 standard DOCKSTRING targets won by each CBR variant at four oracle budgets (10 seeds per condition). Iterative dominates at $B = 100$ (79%), expansion wins at $B = 500$ (54%), and iterative regains the lead at $B = 5000$ (75%). Error bars show bootstrap 95% CIs. Retrieval-only never wins at any budget.

Per-target variation. At budget 100, iterative refinement achieves the strongest top-100 mean on targets with deep binding pockets, such as ABL1 (-9.00), PDE5A (-9.95), and ACHE (-9.68). Shallow or solvent-exposed targets such as BACE1 (-7.30), CASP3 (-7.53), and AR (-7.54) yield weaker scores, reflecting fewer potent binders in the ExCAPE-DB library.

Ceiling recovery. The best-in-data ceiling (top-100 mean over all ∼260K molecules, no filter) ranges from -10.56 (BACE1) to -14.36 (ACHE). At budget 100, iterative refinement recovers $69.4\% \pm 1.8\%$ of the ceiling across all 57 standard targets. At budget 500, this rises to $80.9\% \pm 2.1\%$. The narrow variance shows that ceiling recovery is approximately constant across targets, independent of absolute difficulty.

At budgets 1000 and 5000, expansion and iterative converge to within 0.1 kcal/mol of each other (mean top-1: -12.0 and -12.5 kcal/mol respectively), both 1.5 kcal/mol ahead of retrieval-only. The crossover between expansion and iterative is budget-dependent: iterative wins 79% of targets at $B = 100$ (CI: 68–90%), expansion wins at intermediate budgets ($B = 200$–1000, 53–67%), and iterative regains a clear lead at $B = 5000$ (75%, CI: 63–86%). Figure 2 summarizes this pattern; retrieval-only never wins at any budget.

5.4 Oracle-ranker gap analysis

For each target, we decompose the gap between the method's top-1 and the library ceiling into two components: the *retrieval gap* (library ceiling minus best score in the full retrieved candidate pool, before budget truncation) and the *budget gap* (best score in the full pool minus best score actually evaluated within the budget). A large retrieval gap indicates that the method's neighborhood

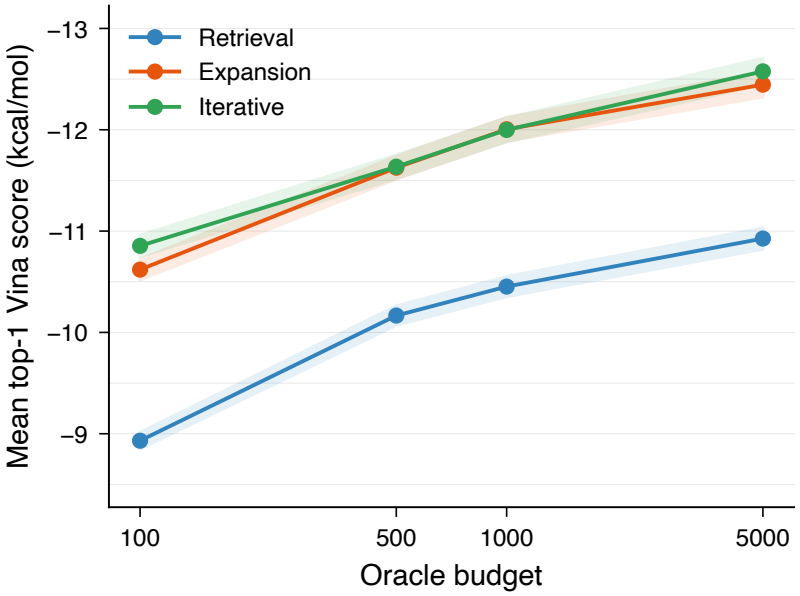


Figure 3: Mean top-1 Vina score across DOCKSTRING targets as a function of oracle budget. Shaded regions show ±1 SEM. Expansion and iterative track each other closely, both 1.5 kcal/mol ahead of retrieval-only across all budgets. The gap narrows with budget as retrieval-only samples more of the library.

search misses the library's strongest binders—generation of novel molecules would help. A large budget gap indicates that the retrieved pool contains strong molecules that went unevaluated due to budget constraints—a surrogate ranker would help.

Table 5 reports the decomposition at budget 100 across all 57 standard DOCKSTRING targets (10 seeds each).

Table 5: Oracle-ranker gap decomposition at budget 100 (mean ± std across 57 targets, kcal/mol). Pool size is the number of candidates retrieved before budget truncation.

Method	Retrieval gap	Budget gap	Pool size
Retrieval	−4.81	0.00	10
Expansion	−1.50	−1.62	9,182
Iterative	−1.61	−1.27	9,158

Retrieval-only has zero budget gap (all filtered seeds are docked) but a large retrieval gap (−4.81 kcal/mol): the pool is too small to contain strong binders. Expansion and iterative retrieve ∼9,200 candidates via neighbor expansion but can only dock ∼85 within the remaining budget. For expansion, 52% of the total gap is budget-limited and 48% is retrieval-limited; for iterative, the budget fraction drops to 44%, consistent with multi-round refinement concentrating the budget on productive neighborhoods.

Figure 4 shows the per-target decomposition. Targets at the top (AR, THRB, ABL1) are predominantly budget-limited: the retrieved pool already contains strong binders, but the budget is exhausted before they can be evaluated. A surrogate ranker [12] would help these targets. Targets

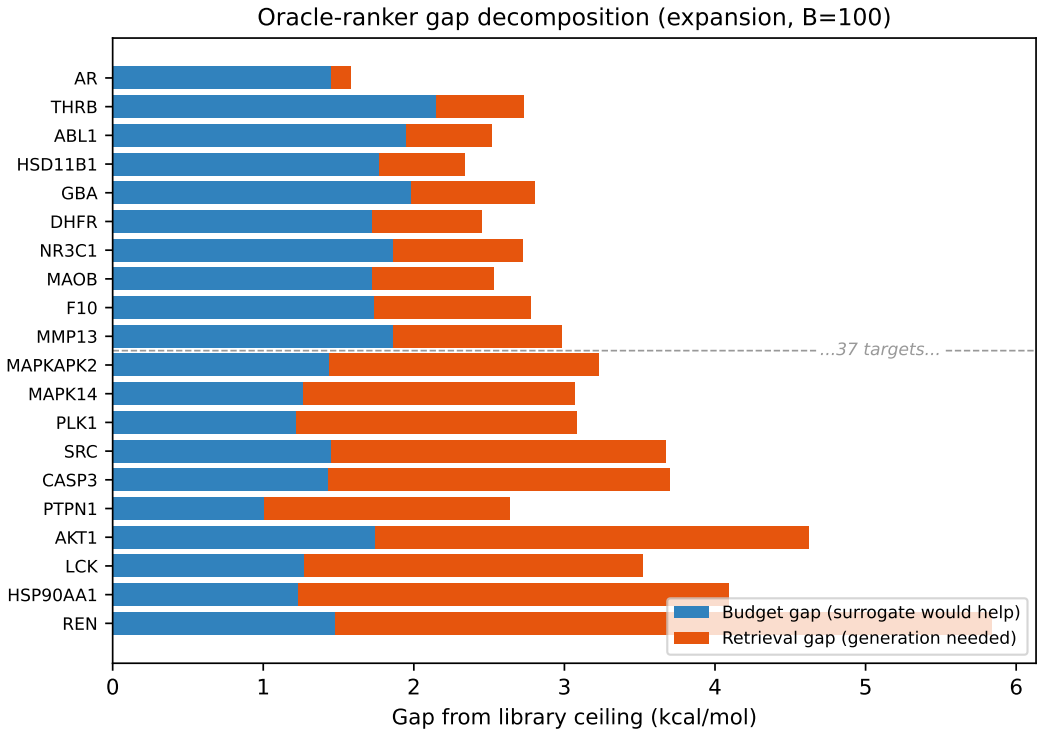


Figure 4: Oracle-ranker gap decomposition (expansion, $B = 100$, 10 seeds): 10 most budget-limited targets (top) and 10 most retrieval-limited (bottom), separated by a dashed line (37 intermediate targets omitted; full figure in Appendix, Figure S4). Each bar decomposes the gap between the method's top-1 and the library ceiling into a budget gap (blue; a surrogate ranker would help) and a retrieval gap (orange; generation needed). Targets sorted by budget-limited fraction.

at the bottom (REN, HSP90AA1, LCK) are retrieval-limited: the fingerprint neighborhood misses the library's strongest binders, and generation of novel molecules would be required.

Pool sizes are remarkably stable across targets (std = 29 for expansion, 14 for iterative), confirming that pool size is determined by fingerprint geometry rather than target identity. The green diamonds show that random sampling's gap from the ceiling tracks expansion's total gap on most targets, reinforcing that library composition drives performance.

5.5 Selectivity and multi-objective tasks

We tested retrieval on four composite objectives at budget 100 to bound where it fails. On kinase selectivity (JAK2 minus LCK), both methods find selective molecules (iterative -0.56 ± 0.27, expansion -0.37 ± 0.06). On promiscuous PPAR binding (worst score across three subtypes), iterative finds pan-subtype binders 0.33 kcal/mol stronger than expansion. On F2 docking with a QED penalty, iterative achieves -9.85 ± 0.09 versus expansion's -9.35 ± 0.32.

DRD3-over-DRD2 selectivity is the honest failure: both methods yield positive scores (expansion +0.22, iterative +0.11), meaning neither finds DRD3-selective molecules. Library compounds are promiscuous across dopamine receptor subtypes. This aligns with PMO-Dock's specificity tasks [9], where retrieval baselines should underperform.

5.6 TDC 9-target validation

Table 6 shows expansion and iterative results on 9 TDC docking targets using live Vina on ZINC 250K. Iterative refinement wins on all 9 targets at budget 100, with improvements of −0.26 to −0.78 kcal/mol (average −0.47), matching the DOCKSTRING result (−0.42) and confirming cross-library transfer.

Table 6: Expansion vs. iterative on 9 TDC targets (live docking, ZINC 250K, budget 100, mean of 3 seeds). Top-100 mean docking score (kcal/mol).

Target	PDB	Expansion	Iterative	Δ
CDK2	4UNN	−7.31	−7.85	−0.54
DRD2	3NY8	−8.07	−8.53	−0.46
DRD3	3PBL	−7.68	−7.94	−0.26
EGFR	7L11	−6.23	−6.59	−0.35
IGF1R	1IEP	−7.86	−8.20	−0.35
JAK2	4RLU	−8.18	−8.55	−0.37
LCK	5MO4	−7.31	−7.82	−0.51
PPARG	3EML	−7.87	−8.65	−0.78
SRC	2RGP	−7.51	−8.08	−0.57
Average		−7.56	−8.02	−0.47

Five targets (CDK2, DRD2, DRD3, EGFR, JAK2) appear in both benchmarks. Absolute scores differ between libraries (DOCKSTRING uses ExCAPE-DB, 0.5–2.5 kcal/mol stronger), and rank ordering also differs: EGFR ranks last on ZINC 250K (−6.59) but second-best on ExCAPE-DB (−9.13). Target solvability depends on library composition, supporting the atlas's per-target, per-library framing over single aggregate metrics.

6 Discussion

6.1 Three regimes of molecular optimization

The crossover map reveals three regimes separated by oracle budget:

1. **Nonparametric retrieval** (budget $\lesssim$ a few hundred). Too few oracle calls to train any model. Fingerprint-guided retrieval from a precomputed library dominates because it converts all calls into exploitation. Our three methods operate here.

2. **Surrogate-guided screening** (budget $\sim 10^3$–10^5). Enough labeled data to fit a surrogate model that guides pool-based screening. MolPAL [12] operates here.

3. **Generative search** (budget $\gtrsim 10^3$–10^4). Methods propose molecules outside the library. Graph-GA, REINVENT, and SEISMO operate here [3, 8].

The crossover between regimes 1 and 3 varies by target. On DRD3, Graph-GA surpasses the retrieval floor between 100 and 500 calls. On targets where the library already contains strong

binders (ACHE ceiling −14.36, iterative at $B = 5000$: −12.5), the crossover budget is higher because retrieval captures most of the available affinity. On targets with poor library coverage (BACE1 ceiling −10.56, iterative at $B = 5000$: −9.8), generators that propose novel scaffolds would surpass the floor sooner. Within regime 1, the expansion-to-iterative crossover at budget 500 illustrates a micro-scale version of the same tradeoff (Figure 2).

We tested whether the expansion-to-iterative crossover budget is predictable from target properties (library ceiling, pool coverage, filter pass rate). Pool coverage is the only significant predictor (Spearman $r = 0.27$, $p = 0.042$), but a linear model combining all three features explains less than 1% of variance (OLS $R^2 = 0.006$). The crossover budget is not well-predicted by simple target features, suggesting that it depends on the fine-grained geometry of the fingerprint neighborhood around high-scoring molecules rather than on aggregate library statistics.

6.2 Library composition dominates low-budget performance

The random baseline reveals that most of the retrieval floor's height comes from the library itself. When the library is curated from bioactivity databases (as ExCAPE-DB is), even uniform random sampling recovers AUC within 1.6% of targeted expansion. Fingerprint-guided retrieval adds value only beyond 500 calls, where neighbor expansion from top hits concentrates the budget on productive chemical neighborhoods. Below this threshold, the "retrieval strategy" is secondary to the "retrieval source." This implies that benchmark comparisons between retrieval and generation are confounded by library quality: a retrieval method on a rich library will beat a generator on a poor one regardless of the generator's capacity.

A pre-registered prediction that targets with better library ceilings would have smaller retrieval gaps was falsified (Spearman $r = +0.50$, $p < 0.001$): targets with the most extreme ceilings actually have *larger* retrieval gaps. The explanation is that targets with very strong binders in the library have extreme ceilings that are difficult to reach via fingerprint neighbor expansion, even though the library contains strong binders overall. Pool size is target-independent (std = 29 across 57 targets), but pool *location* in chemical space varies with which seeds score best on each target.

6.3 Why the atlas compares retrieval variants, not generators

DOCKSTRING provides precomputed Vina scores for a fixed library of 260K molecules. Learned generators propose molecules *outside* this library, so their scores cannot be looked up—they require live docking. The atlas therefore measures the retrieval ceiling per target: the best score achievable by any method restricted to the library. Any generator that exceeds this ceiling on a given target has demonstrated genuine generative value on that target. The DRD3 comparison (Table 4) illustrates this: Graph-GA reaches −14.81 kcal/mol at 5000 calls, far beyond the best-in-data ceiling of −12.08, because it proposes novel molecules absent from ZINC 250K.

6.4 The nonparametric floor as standard practice

Docking benchmarks at low budgets primarily measure library coverage rather than generative capacity. We propose that future benchmarks report a fingerprint-retrieval floor alongside all learned methods, as information retrieval benchmarks report BM25. A method that cannot surpass this floor at a given budget has failed to demonstrate generative capability—it has underperformed a lookup table.

Benchmarks could further separate retrieval from generation by requiring proposed molecules to differ structurally from the library (minimum Tanimoto distance to any library member), or by evaluating on targets where the library contains few strong binders.

6.5 Limitations

Retrieval proposes only molecules already present in the search library and cannot produce novel scaffolds. The retrieval floor is library-dependent: the ExCAPE-DB library underlying DOCKSTRING was curated from bioactivity databases and is enriched for drug-like molecules, so our retrieval benefits from a library assembled by domain experts. A different library (e.g., a random subset of ZINC) would produce a different atlas, and the floor's height should be understood as a property of the *library-target pair*, not of retrieval in general.

All scores in this study are AutoDock Vina predictions with fixed docking parameters. Vina's correlation with experimental binding affinity is moderate ($r \approx 0.3$–0.5), so the crossover atlas maps where retrieval beats generation *on Vina scores*, not necessarily on real binding. This limitation applies equally to PMO-Dock [9] and SEISMO [8], which use the same scoring framework.

The DRD3-over-DRD2 selectivity experiment shows that retrieval fails on subtype-selectivity objectives because library compounds tend to be promiscuous across related receptors. PMO-Dock's specificity tasks would provide a systematic test of this limitation.

Our three methods instantiate Retrieve, Reuse, and Retain but omit Revise—the CBR step that modifies molecular structures. The absence of Revise is precisely the gap where learned generators earn their parameters: scaffold hopping, fragment merging, and structure-guided optimization all correspond to a Revise operation that retrieval cannot perform. Hybrid approaches such as f-RAG [27], which retrieves molecular fragments and feeds them to a generative model, demonstrate that retrieval and generation can be composed rather than treated as alternatives.

The iterative variant was selected on a synthetic oracle whose inductive bias (Tanimoto locality) favors retrieval strategies; the multi-target sweep tests whether this advantage transfers. Our AUC metric is computed from 4 pre-registered budget snapshots (100, 500, 1000, 5000), yielding a coarser approximation than per-call trajectories reported by SEISMO [8]. Post-hoc evaluation at intermediate budgets (200, 300) confirms the crossover pattern but these were not pre-registered. Finer-grained logging would enable direct AUC comparison.

7 Conclusions

We constructed a budget-crossover atlas across 67 docking targets that maps where zero-parameter retrieval stops winning and learned generation begins to earn its computational cost. Three retrieval methods—instantiating successive steps of the CBR cycle [19]—establish a nonparametric floor that any learned optimizer should surpass before claiming generative capability.

The atlas reveals a U-shaped crossover between expansion and iterative refinement: iterative wins 79% of targets at budget 100 and 75% at budget 5000, while expansion wins at intermediate budgets (54–67%). On TDC's DRD3 target, expansion places first at 100 oracle calls with zero learned parameters. Generation surpasses the floor only beyond a few hundred calls, and on most targets only beyond 1000.

These findings suggest a practical recommendation: future molecular optimization benchmarks should report a fingerprint-retrieval floor alongside all learned methods, as information retrieval benchmarks report BM25. The atlas is publicly available and extensible to new targets and libraries.

Declarations

Ethics approval and consent to participate

Not applicable. This study uses only publicly available benchmark datasets (DOCKSTRING, TDC) with no human subjects research.

Consent for publication

Not applicable.

Availability of data and materials

DOCKSTRING precomputed scores are available from García-Ortegón et al. [2]. TDC docking targets are available from Huang et al. [1]. All results files, pre-registration commits (SHA `94989e0` for expansion; SHA `bacc0ff` for iterative), deviation log, and experimental scripts are available at `https://github.com/elliottower/generation-vs-retrieval-docking`. The repository was public before the pre-registration freeze. A citable archive of the code, data, and manuscript is deposited on Zenodo at `https://doi.org/10.5281/zenodo.21281732`. To reproduce the full atlas:

```
python experiments/dockstring_sweep.py --all-targets
```

Competing interests

The author declares no competing interests.

Funding

This work received no external funding.

Authors' contributions

ET conceived the study, designed the CBR framing, conducted all experiments, and wrote the manuscript.

Use of AI-assisted technologies

Claude (Anthropic) was used as a writing and analysis assistant during manuscript preparation. The author directed all analytical decisions and verified all numerical results against source data. The CBR framework, pre-registered protocols, experimental results, and interpretations are the author's own. The author takes full responsibility for the content. No AI tool is listed as an author.

Acknowledgements

Not applicable.

Authors' information

ET is an independent researcher focused on benchmarking methodology and validity assessment for computational chemistry.

List of abbreviations

AUC: area under the curve; CBR: case-based reasoning; CI: confidence interval; ECFP4: extended-connectivity fingerprint (radius 2); QED: quantitative estimate of drug-likeness; RL: reinforcement learning; SA: synthetic accessibility; SEM: standard error of the mean; TDC: Therapeutics Data Commons.

Additional files

Additional file 1: Supplementary figures. Per-target crossover heatmap (Figure S1), per-target score distributions at $B = 5000$ (Figure S2), improvement margin over retrieval-only (Figure S3), and full 57-target gap decomposition (Figure S4). (Format: PDF)

References

[1] K. Huang, T. Fu, W. Gao, Y. Zhao, Y. Roohani, J. Leskovec, C. W. Coley, C. Xiao, J. Sun, and M. Zitnik. Therapeutics Data Commons: Machine learning datasets and tasks for drug discovery and development. In *NeurIPS Datasets and Benchmarks*, 2021.

[2] M. García-Ortegón, G. N. C. Simm, A. J. Tripp, J. M. Hernández-Lobato, A. Bender, and S. Bacallado. DOCKSTRING: Easy molecular docking yields better benchmarks for ligand design. *J. Chem. Inf. Model.*, 62(15):3486–3502, 2022.

[3] W. Gao, T. Fu, J. Sun, and C. W. Coley. Sample efficiency matters: A benchmark for practical molecular optimization. In *NeurIPS*, 2022.

[4] M. Thomas, A. Bender, and C. de Graaf. Generative models should at least be able to design molecules that dock well: A new benchmark for molecular design. *arXiv:2308.05028*, 2023.

[5] M. Olivecrona, T. Blaschke, O. Engkvist, and H. Chen. Molecular de-novo design through deep reinforcement learning. *J. Cheminform.*, 9(1):48, 2017.

[6] J. H. Jensen. A graph-based genetic algorithm and generative model / Monte Carlo tree search for the exploration of chemical space. *Chem. Sci.*, 10:3567–3572, 2019.

[7] S. Lee et al. GenMol: A drug discovery generalist with discrete diffusion. *arXiv*, 2025.

[8] F. P. Krüger, A. Hunklinger, A. Wolny, T. J. Adler, I. Tetko, and S. D. Villalba. SEISMO: Increasing sample efficiency in molecular optimization with a trajectory-aware LLM agent. *arXiv:2602.00663*, 2026.

[9] K. Simonyan et al. PMO-Dock: Benchmarking docking, specificity and generalization in molecular optimization. In *ICLR 2026 Workshop GEM*, 2026.

[10] R. Das, A. Godbole, A. Naik, E. Tower, R. Jia, M. Zaheer, H. Hajishirzi, and A. McCallum. Knowledge base question answering by case-based reasoning over subgraphs. In *ICML*, 2022.

[11] R. Das, A. Godbole, N. Monath, M. Zaheer, and A. McCallum. A simple approach to case-based reasoning in knowledge bases. *arXiv:2006.14198*, 2021.

[12] D. E. Graff, E. I. Shakhnovich, and C. W. Coley. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chem. Sci.*, 12:7866–7881, 2021.

[13] A. J. Tripp, E. A. Daxberger, and J. M. Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. In *NeurIPS*, 2020.

[14] J. Guo and P. Schwaller. Saturn: Sample-efficient generative molecular design using memory manipulation. *arXiv:2405.17066*, 2024.

[15] D. Rogers and M. Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, 2010.

[16] P. Ertl and A. Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.*, 1(1):8, 2009.

[17] J. Eberhardt, D. Santos-Martins, A. F. Tillack, and S. Forli. AutoDock Vina 1.2.0: New docking methods, expanded force field, and Python bindings. *J. Chem. Inf. Model.*, 61(8):3891–3898, 2021.

[18] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pp. 335–336, 1998.

[19] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, 1994.

[20] M. H. S. Segler, T. Kogej, C. Tyrchan, and M. P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.*, 4(1):120–131, 2018.

[21] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *NeurIPS*, 2018.

[22] Y. Xie, C. Shi, H. Zhou, Y. Yang, W. Zhang, Y. Yu, and L. Li. MARS: Markov molecular sampling for multi-objective drug discovery. In *ICLR*, 2021.

[23] Z. Zhou, S. Kearnes, L. Li, R. N. Zare, and P. Riley. Optimization of molecules via deep reinforcement learning. *Sci. Rep.*, 9(1):10752, 2019.

[24] H. Kim, M. Kim, S. Ahn, and J. Shin. Genetic-GFN: Genetic algorithm-inspired GFlowNets for molecular optimization. In *NeurIPS*, 2024.

[25] S. Nikolenko et al. Chemlactica: Large language model for small molecule generation and optimization. *arXiv:2407.12541*, 2024.

[26] D. E. Graff, M. Aldeghi, J. A. Morrone, K. E. Jordan, E. O. Pyzer-Knapp, and C. W. Coley. Self-focusing virtual screening with active design space pruning. *J. Chem. Inf. Model.*, 62(16):3854–3862, 2022.

[27] Y. Chen, Z. Wang, and L. Xie. f-RAG: Fragment retrieval-augmented generation for molecular optimization. In *NeurIPS*, 2024.

[28] Z. Wang, Y. Wen, A. Pandy, H. Liu, and K. Ding. MolMem: Memory-augmented agentic reinforcement learning for sample-efficient molecular optimization. In *ACL*, 2026.