GENERATOR DISAGREEMENT ACQUISITION FOR SAMPLE-EFFICIENT MOLECULAR DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Goal-directed molecular design often emphasizes the best molecule found, whereas practical discovery campaigns require multiple promising candidates under a limited budget of expensive oracle evaluations. In parent-conditioned search, however, a parent's observed reward need not reflect its ability to generate valuable offspring. This motivates selecting parents using predicted offspring reward distributions rather than their own scores alone. We introduce Generator Disagreement Acquisition (GDA), a closed-loop framework that allocates oracle queries at the parent level. GDA samples offspring from each candidate parent and combines the mean and dispersion of their surrogate-predicted rewards to select which parents supply candidates for oracle evaluation. This dispersion, termed generator disagreement, measures variation in predicted rewards across offspring rather than disagreement among predictors for a fixed molecule. Oracle feedback updates both the surrogate and the parent pool. We evaluate GDA on six tasks across DOCKSTRING and GEAM. All methods start from the same 512 labeled molecules and receive 800 additional oracle evaluations. With random-forest ensembles, GDA improves mean top-1% hit yield over a screened genetic algorithm using the same proposal operator and surrogate family by 38–61% on DOCKSTRING and 16–33% on GEAM. It also improves hit yield over Augmented Memory by 27–69% on DOCKSTRING, while Augmented Memory remains competitive on two GEAM tasks. These results support parent-level acquisition from predicted offspring distributions as a sample-efficient strategy for identifying multiple high-scoring candidates for downstream evaluation.

1 INTRODUCTION

Molecular discovery requires deciding which candidates should undergo an expensive evaluation. Generative methods can extend the search beyond a fixed library but their usefulness depends on what they find within the available oracle budget (García-Ortegón et al., 2022; Gao et al., 2022). Campaigns also need several promising candidates, since properties not included in the objective can rule out an otherwise high-scoring molecule. Hit yield and oracle burden capture this need by measuring how many distinct candidates meet a reward threshold and how many evaluations are needed to find them (Guo et al., 2026). In ligand discovery, candidate quality depends on binding affinity, selectivity, and molecular properties relevant to further development. We study these considerations through docking-based objectives and synthetic-accessibility (SA) filtering (García-Ortegón et al., 2022; Ertl & Schuffenhauer, 2009).

The observed reward of a parent molecule does not necessarily reflect its usefulness for generating new candidates. Parents with similar observed rewards may produce different offspring reward distributions under the same operator. Even when their offspring have comparable average rewards, one parent may offer a better chance of producing a high-reward child. Predictions across sampled offspring could therefore help assess parents before committing oracle calls. Reward feedback already guides context or parent selection in methods, such as, RGA, BOPRO, and TTT-Discover (Table 1). We investigate whether the predicted reward distribution of a parent's unevaluated offspring provides a useful signal for this decision.

We introduce *Generator Disagreement Acquisition* (GDA), a family of parent-level acquisition methods that uses this distribution. At each round, offspring from every parent in memory are

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

Table 1: **GDA selects parents through predicted offspring reward distributions.** The comparison describes core search decisions in representative methods. SGPO focuses on protein fitness optimization, while TTT-Discover addresses a broader range of scientific search problems. Rows do not imply an empirical ranking.

Method	Decision guided	Main reward signal
Graph GA (Jensen, 2019)	Parents and population	Evaluated molecular fitness.
RGA (Fu et al., 2022)	Parents and genetic actions	Learned target–ligand policies, updated using fitness improvements.
GEAM (Lee et al., 2024b)	Fragment vocabulary, assembly, and genetic search	Goal-aware fragment scores and reward-trained assembly policies.
f-RAG (Lee et al., 2024a)	Retrieved fragments and genetic population	Property-ranked fragment vocabularies and evaluated molecules.
Genetic GFN (Kim et al., 2024)	Generative policy and genetic population	Reward-ranked replay and GFlowNet learning from genetic search.
Augmented Memory / Saturn (Guo & Schwaller, 2024; Guo et al., 2026)	Generative policy and replay buffer	Evaluated rewards reused through replay and SMILES augmentation.
SGPO (Yang et al., 2025)	Guided sequence generation	Fitness predictors; ensemble-based adaptive guidance.
BOPRO (Agarwal et al., 2025)	Latent region and generation context	Bayesian acquisition in latent space; nearby evaluated solutions form the prompt.
TTT-Discover (Yuksekgonul et al., 2026)	Archived starting states	Best observed child reward, reward-rank prior, and visit counts.
GDA (ours)	**Parents supplying oracle queries**	**Mean and dispersion of predicted rewards across each parent's offspring.**

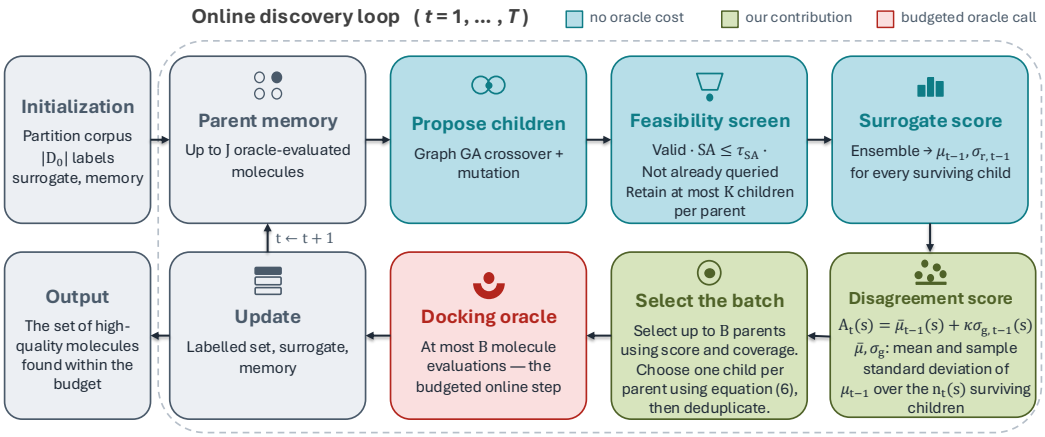

Figure 1: **One round of GDA**. Each parent proposes up to K structurally valid offspring. After screening, predicted offspring mean $\bar{\mu}$ and dispersion σ_g guide parent selection together with structural coverage. One child is chosen per selected parent. Deduplication and the remaining budget give at most B oracle queries. New labels update the surrogate and parent memory.

generated, screened, and scored by a surrogate. GDA combines the mean and dispersion of their predicted rewards to select parents. We call the dispersion *generator disagreement*: it measures variation across offspring of one parent-conditioned operator, which is distinct from ensemble uncertainty about a fixed molecule. Predicted reward, ensemble spread, and novelty then guide child selection. Oracle feedback updates the surrogate and parent memory (Figure 1). The same acquisition rule accommodates different parent-conditioned operators and surrogates. We evaluate GDA across proposal operators based on Graph GA and fragment recombination, and examine how

its performance varies between random-forest and graph-neural-network surrogates. GDA requires samples from the operator, without needing its gradients or likelihood. Our contributions are:

- **Acquisition from predicted offspring distributions.** We introduce generator disagreement for parent selection and use a standard sub-Gaussian bound on the expected best predicted offspring reward to motivate the $\sqrt{2 \log K}$ exploration scaling. The empirical acquisition score is distinguished from this population bound.
- **A family of closed-loop discovery methods in molecular graph space.** We integrate parent acquisition with child selection, structural coverage, and memory updates into a closed-loop search over molecular graphs. The formulation supports different parent-conditioned proposal operators under a shared acquisition rule, without requiring continuous latent-space optimization.
- **An empirical analysis of parent-level acquisition.** We compare GDA with pooled candidate selection and alternative parent-selection rules across DOCKSTRING and GEAM. GDA improves mean top-1% hit yield over pooled selection on all three DOCKSTRING tasks with both surrogate families, with larger relative gains for graph neural network ensembles than for random forest ensembles. On all three GEAM tasks, GDA with random forest ensembles improves mean top-1% and top-5% hit yields over GA+screen.

2 Related Work

Evolutionary and fragment-based search. Graph GA remains a competitive molecular optimizer (Jensen, 2019; Tripp & Hernández-Lobato, 2023). Subsequent methods learn how to guide the search: RGA learns policies for choosing parents and genetic actions (Fu et al., 2022), whereas Genetic GFN trains a generative policy on molecules produced by genetic search (Kim et al., 2024). GEAM and f-RAG adapt fragment vocabularies and generation (Lee et al., 2024b;a). Our fragment operator uses the rules of Breaking of Retrosynthetically Interesting Chemical Substructures (BRICS) (Degen et al., 2008). GDA compares parents through their predicted offspring rewards.

Generation with limited feedback. REINVENT adapts a molecular language model using reward feedback (Olivecrona et al., 2017). Augmented Memory uses experience replay and randomized SMILES to learn repeatedly from previously evaluated molecules (Guo & Schwaller, 2024). Saturn combines this training strategy with a Mamba-based molecular generator (Guo et al., 2026). Saturn also studies an optional GA on its replay buffer. SGPO examines adaptive guidance for protein generation (Yang et al., 2025), and SEIKO uses reward and uncertainty estimates to fine-tune diffusion models (Uehara et al., 2024). GDA guides search through parent selection without requiring generator-weight updates.

Acquisition and generation context. Bayesian acquisition uses predicted value and uncertainty to guide evaluations (Srinivas et al., 2010). BOPRO applies acquisition in latent space to select generation contexts (Agarwal et al., 2025). TTT-Discover selects starting states using observed child rewards and visit counts (Yuksekgonul et al., 2026). GDA evaluates parents using predicted reward variation across unqueried offspring, which differs from uncertainty about one candidate or evidence from past expansions.

Feasibility and evaluation. We use the synthetic accessibility (SA) score (Ertl & Schuffenhauer, 2009) as an inexpensive filter based on molecular structure before oracle evaluation. Retrosynthetic optimization (Guo & Schwaller, 2025) and pathway generation with PrexSyn (Luo & Coley, 2026) provide complementary approaches that assess feasibility through explicit synthesis routes. PMO emphasizes sample efficiency under limited oracle budgets (Gao et al., 2022), while Saturn uses hit yield and oracle burden to assess discovery performance (Guo et al., 2026). Molecular generation can produce candidates outside the corpus, extending the search beyond screening a fixed library (Thomas et al., 2026). Top-1% hit status is determined by reward, while corpus novelty indicates whether a molecule is absent from the corpus.

3 Method

Our proposed Generator Disagreement Acquisition (GDA) method selects parent molecules according to the predicted rewards of their offspring. Algorithm 1 summarizes the procedure. Starting from an initially labelled set, we fit a surrogate and initialize a parent pool. At each round, we

Algorithm 1 Generator Disagreement Acquisition

Input: $\mathcal{D}_0$, operator $\mathcal{O}$, online budget L, batch limit B, rollouts K, memory size J
Fit F_0 on $\mathcal{D}_0$; initialize $\mathcal{M}_0$ using observed rewards and equation 5; set $t = 1$
while $\sum_{u=1}^{t-1} |\mathcal{B}_u| < L$ **do**
  for each $s \in \mathcal{M}_{t-1}$ **do**
    Propose up to K valid children within the attempt limit; screen to obtain $\mathcal{R}_t(s)$
    If survivors remain, predict rewards and compute $A_t(s)$ by equation 3–equation 4
  Select up to B eligible parents greedily using equation 5
  Choose one child per selected parent using equation 6; remove duplicate finalists
  Truncate to the remaining budget, query $\mathcal{B}_t$, and append its labels to $\mathcal{D}_{t-1}$
  Refit F_t on $\mathcal{D}_t$; rebuild $\mathcal{M}_t$ using equation 7–equation 8; increment t
Return: the evaluated molecules and their oracle rewards

generate and screen offspring from every parent in the pool, then use the mean and dispersion of their predicted rewards to guide parent selection. We select one offspring from each chosen parent and evaluate the resulting batch after removing duplicates and accounting for the remaining oracle budget. The new labels update both the surrogate and the parent pool. We repeat this process until the oracle budget is exhausted. The following subsections formulate the problem and describe each component.

3.1 Problem Formulation

Let $\mathcal{G}$ denote the set of connected, neutral molecular graphs over a fixed atom and bond vocabulary, and let $y : \mathcal{G} \to \mathbb{R}$ be an expensive oracle reward, with larger values preferred. For example, y is the negative of the task's docking-based objective for DOCKSTRING. Every method starts with the same labelled set $\mathcal{D}_0$ of size n_0 and receives an online budget of L additional evaluations. At round t, it queries a batch $\mathcal{B}_t$ of at most B previously unqueried molecules and updates $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(x, y(x)) : x \in \mathcal{B}_t\}$. $\mathcal{X}_t$ represents the molecular identities in $\mathcal{D}_t$. The campaign stops when $\sum_t |\mathcal{B}_t| = L$.

We evaluate discovery using hit yield and oracle burden, following Guo et al. (2026), with corpus quantiles as reward thresholds. Let τ_q denote the q-quantile of rewards over the task corpus. Hit yield counts distinct newly evaluated molecules whose rewards meet or exceed this threshold:

$$\mathrm{HY}_q(t) = |\{x \in \mathcal{X}_t \setminus \mathcal{X}_0 : y(x) \geq \tau_q\}|, \qquad q \in \{0.99, 0.95\}. \tag{1}$$

The highest-scoring 1% of each task's corpus is withheld from the initially visible partitions, so top-1% hits measure discovery beyond the visible reward ceiling. A hit may be a rediscovered held-out molecule or a molecule absent from the corpus. This split does not make every top-5% molecule unavailable initially. For a requested number of hits n, oracle burden is the cumulative evaluation count at the first completed batch reaching that milestone, where $c_t = n_0 + \sum_{u=1}^{t} |\mathcal{B}_u|$:

$$\mathrm{OB}_q@n = \min\{c_t : \mathrm{HY}_q(t) \geq n\}. \tag{2}$$

We set this quantity to $+\infty$ if the requested number of hits is not reached. We additionally report best-so-far reward and top-10 and top-20 mean rewards. Appendix E defines these measures and the structural diversity and synthetic accessibility metrics.

3.2 Generating and screening offspring

GDA maintains a memory $\mathcal{M}_{t-1} \subseteq \mathcal{X}_{t-1}$ containing at most J parents and a surrogate fitted using $\mathcal{D}_{t-1}$. A parent-conditioned proposal operator $\mathcal{O}_t(\cdot \mid s)$ generates candidates from $s \in \mathcal{M}_{t-1}$. The index t in $\mathcal{O}_t$ indicates that the proposal distribution may depend on the current memory and generation history. Our main operator uses Graph GA crossover and mutation (Jensen, 2019; Tripp & Hernández-Lobato, 2023). A BRICS fragment-recombination operator (Degen et al., 2008) provides an alternative implementation. GDA requires this sampling interface, rather than a specific operator architecture. Operator details and parent context are given in Appendix C.

For each parent, we attempt to collect K distinct, structurally valid proposals. We then exclude previously queried molecules and apply the SA filter (Ertl & Schuffenhauer, 2009). The surviving

set is $\mathcal{R}_t(s) \subseteq \mathcal{C} \setminus \mathcal{X}_{t-1}$, where $\mathcal{C} = \{x \in \mathcal{G} : \mathrm{SA}(x) \le \tau_{\mathrm{SA}}\}$. Let $n_t(s) = |\mathcal{R}_t(s)| \le K$. Screening can reduce $n_t(s)$ even when the operator returns K valid proposals. Parents with no survivors are not eligible. All parents are sampled before parent selection.

3.3 Generator disagreement acquisition

For each surviving child x, the ensemble $F_{t-1} = \{f_{t-1,j}\}_{j=1}^{m}$ returns the predictive mean $\mu_{t-1}(x) = m^{-1}\sum_j f_{t-1,j}(x)$ and the sample standard deviation $\sigma_{r,t-1}(x)$ across members. This ensemble spread is an uncertainty proxy at one molecule. In contrast, generator disagreement measures variation in predicted mean rewards across different offspring of one parent. For $\mathcal{R}_t(s) = \{x_{s,1}, \ldots, x_{s,n}\}$, with $n = n_t(s) > 0$, we define

$$\bar{\mu}_{t-1}(s) = \frac{1}{n}\sum_{k=1}^{n} \mu_{t-1}(x_{s,k}), \qquad \sigma_{g,t-1}(s) = \left[\frac{1}{n-1}\sum_{k=1}^{n}\big(\mu_{t-1}(x_{s,k}) - \bar{\mu}_{t-1}(s)\big)^2\right]^{1/2}. \quad (3)$$

We set $\sigma_{g,t-1}(s) = 0$ when $n = 1$. GDA assigns the parent acquisition score

$$A_t(s) = \bar{\mu}_{t-1}(s) + \kappa\, \sigma_{g,t-1}(s), \qquad \kappa = c\sqrt{2\log K}. \quad (4)$$

The mean favors parents whose offspring are predicted to have high rewards. The spread favors parents whose offspring include a wider range of predicted rewards. Both quantities rely on the parent, the operator, and the current surrogate. Generator disagreement is therefore different from disagreement between generators, from the parent's own reward, and from σ_r at a fixed child. Throughout the paper, $\log$ denotes the natural logarithm.

Motivation for the dispersion bonus. Since each selected parent supplies at most one oracle query, high reward offspring can be useful even when their parent also produces weaker candidates. The following standard maximal inequality motivates a mean plus dispersion score.

Proposition 1 (Expected maximum of predicted offspring rewards)**.** *Condition on the search history and a fixed surrogate. Let $K \ge 1$ be an integer and $v_s \ge 0$. Suppose $V_1, \ldots, V_K$ have common mean m_s and satisfy $\mathbb{E}[e^{\lambda(V_k - m_s)}] \le e^{\lambda^2 v_s^2/2}$ for all $\lambda \in \mathbb{R}$. Then $\mathbb{E}[\max_{k\le K} V_k] \le m_s + v_s\sqrt{2\log K}$.*

The proof and its scope are given in the Appendix A. The population tail parameter v_s does not need to equal the standard deviation, and its empirical replacement in equation 4 is not a certified upper bound. We use the inequality to motivate the dependence on K. The coefficient uses the configured K even when fewer children survive screening. At $c = 0$, the score is the mean predicted offspring reward. It does not reduce to the parent's observed fitness. GDA summarizes predicted offspring rewards but does not directly optimize hit yield or guarantee improvement in oracle rewards.

3.4 Parent and child selection

Parent batch. To combine acquisition value with representation of the memory, we define

$$\Phi_U(S; v) = \zeta \sum_{s\in S} v(s) + (1-\zeta)\sum_{u\in U} \max_{s\in S} \mathrm{sim}(u, s), \qquad \max_{s\in\varnothing} \mathrm{sim}(u, s) := 0, \quad (5)$$

where sim is Tanimoto similarity between Morgan fingerprints and $\zeta \in [0, 1]$. Starting from the empty set, we repeatedly add the eligible parent with largest marginal gain in $\Phi_{\mathcal{M}_{t-1}}(S; A_t)$, until B parents are selected or none remain. This gives $\mathcal{S}_t$. The coverage term has diminishing returns when several selected parents represent the same part of the memory. The objective is submodular: the usual greedy $(1 - 1/e)$ guarantee applies when its quality scores are nonnegative (Nemhauser et al., 1978). Signed scores and the precise greedy rule are discussed in Appendix B.

Child selection. For each $s \in \mathcal{S}_t$, we select the child in $\mathcal{R}_t(s)$ maximizing

$$a_t(x) = \mu_{t-1}(x) + \beta_t \sigma_{r,t-1}(x) + \gamma_t \mathrm{nov}_t(x),$$
$$\beta_t = \sqrt{2\log\max\{N_t, 1\}}, \qquad N_t = \sum_{s\in\mathcal{M}_{t-1}} n_t(s), \quad (6)$$

where $\mathrm{nov}_t(x) = 1 - \max_{x'\in\mathcal{X}_{t-1}} \mathrm{sim}(x, x')$ rewards distance from previously evaluated molecules. We set $\gamma_t = \tilde{\gamma}\ \mathrm{med}(\sigma_r)/\ \mathrm{med}(\mathrm{nov}_t)$, with medians taken over the surviving offspring of the selected parents. The ratio scales the novelty bonus relative to ensemble spread, while $\tilde{\gamma}$ controls its

strength. To avoid division by a near-zero value, we set $\gamma_t = 0$ when the novelty median is at most a numerical tolerance ε_{nov}. We use ensemble spread as an uncertainty proxy for child selection in equation 6. The score ranks candidates but is not interpreted as a calibrated confidence bound. Duplicate finalists are removed without adding replacement candidates before querying the batch. Thus, σ_g guides which parents supply queries, while σ_r and novelty help choose their children.

3.5 Updating the memory and surrogate

After evaluating $\mathcal{B}_t$, we refit the surrogate using $\mathcal{D}_t$ and construct memory candidates $\mathcal{K}_t = \mathcal{M}_{t-1} \cup \mathcal{B}_t$. Let $\mathcal{L}_t(s)$ contain all labelled molecules directly generated from s by round t. A molecule generated from several parents contributes to each parent's record. We define

$$q_t(s) = \max\big(\{y(s)\} \cup \{y(x) : x \in \mathcal{L}_t(s)\}\big), \tag{7}$$

so a parent retains its own reward even before it produces a labelled child. Using the same marginal-gain rule and the same ζ as for the parent batch, we set

$$\mathcal{M}_t = \text{Greedy}_{\min\{J,|\mathcal{K}_t|\}}\big(\Phi_{\mathcal{K}_t}(\cdot; q_t)\big). \tag{8}$$

Initialization applies this rule to $\mathcal{X}_0$ with $q_0(s) = y(s)$. Memory quality is based on observed rewards, whereas parent acquisition uses predicted offspring rewards. Each round scores at most JK surviving offspring and selects at most B molecules for oracle evaluation.

4 Experiments

We design our experiments around the following three research questions (RQs):

- **RQ1**: Does GDA find more high value molecules within a fixed oracle budget than alternative parent selection rules and pooled candidate selection?
- **RQ2**: How much does the disagreement bonus contribute when the other components of GDA are held fixed?
- **RQ3**: How does the benefit change with the surrogate family?

4.1 Setup

Benchmarks. We retain the published objectives of DOCKSTRING (García-Ortegón et al., 2022) and GEAM (Lee et al., 2024b). DOCKSTRING adds a quantitative estimate of drug-likeness (QED) penalty to AutoDock Vina scores for F2 affinity, joint affinity across three PPAR subtypes, and JAK2/LCK selectivity. GEAM combines QuickVina2 docking scores, QED, and SA multiplicatively for PARP1, BRAF, and FA7.

Data and splitting. We use structurally filtered, deduplicated DOCKSTRING and ZINC250k (GEAM) corpora. Both follow the hidden-top-1% protocol, with a shared initial labelled set D_0 drawn from an initialization pool. GEAM initialization and online labels come from QuickVina2 docking. Hit thresholds use task-corpus reward quantiles for DOCKSTRING and ZINC250k reference reward quantiles for GEAM, solely for evaluation. Full objectives, docking settings, corpus sizes, and partition details appear in Appendix D.

Protocol. We compare using a cap of $1{,}312$ cumulative evaluations: 512 shared initial labels and up to 800 additional queries, in batches of at most eight. All methods share initialization, feasibility screening and evaluation thresholds. Results are means $\pm$ standard errors over five matched seeds.

Baselines. Pooled children selects the top batch from all deduplicated surviving offspring without parent acquisition. GA+screen samples parents by mean predicted offspring reward, omitting generator disagreement, then applies GDA's child-acquisition rule. Both share GDA's proposal operator and surrogate but use quality-only memory updates. We also compare against Graph GA (Jensen, 2019; Brown et al., 2019) without a surrogate, two strong reinforcement-learning baselines, REINVENT (Olivecrona et al., 2017) and Augmented Memory (Guo & Schwaller, 2024), and also Random sampling from the visible pretraining partition. Surrogates are random-forest (RF) ensembles except in RQ3.

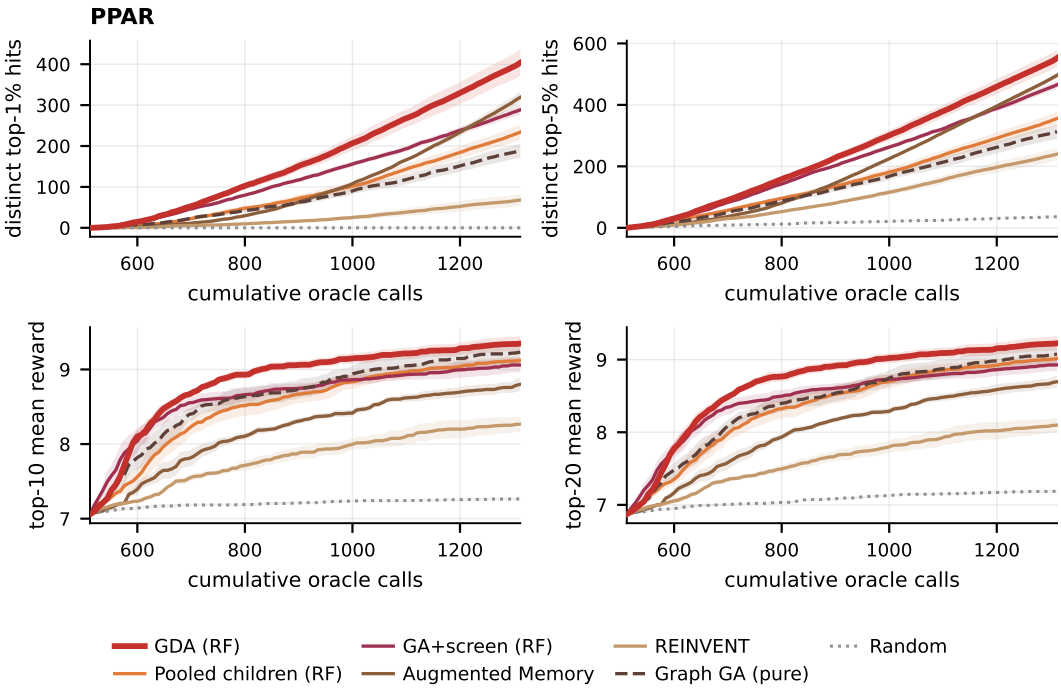

Figure 2: PPAR: hit yield at two thresholds and running top-10 and top-20 mean reward. The horizontal axis starts at 512 cumulative oracle calls, including the shared initialization followed by online evaluations. Hit counts include only online queries, whereas top-10 and top-20 mean rewards include both the initial molecules and subsequent evaluations.

4.2 RQ1: Discovery under a fixed budget

GDA achieves the highest mean top-1% and top-5% hit yields among the methods compared on all three DOCKSTRING tasks (Figure 2 and Appendix F). Averaged across three tasks and five matched random seeds per task, GDA yields 65.7 ± 26.7 more top-1% hits than Pooled children. The largest improvement occurs on PPAR, where GDA finds 404 ± 32 top-1% hits, compared with 234 ± 19 for Pooled children. The paired mean gain is 169.8 ± 33.6 hits, with a positive difference in all five seeds. The PPAR top-10 and top-20 mean reward curves also favor GDA, showing that its gains extend to the quality of the highest-scoring candidates. On GEAM, GDA-RF improves percentile hit yield over GA+screen on all three targets (Figure 3 and Appendix F). At the end of the 800-call online phase, GDA obtains 461.4 ± 39.0 top-1% hits and 659.0 ± 21.9 top-5% hits on FA7, compared with 346.2 ± 43.9 and 563.4 ± 33.8 for GA+screen. The multiplicative GEAM objective requires docking, QED, and SA terms to be simultaneously favorable. A gain in hit yield does not necessarily imply a higher single best reward. GDA's main advantage is broader accumulation of balanced high-reward molecules: it improves top-5% yield and oracle burden relative to GA+screen, while Augmented Memory retains the strongest extreme-reward tail on BRAF and PARP1. This distinction motivates reporting both percentile hit yield and top-k mean reward rather than relying on a single best-so-far value.

GDA achieves the lowest mean oracle burden for reaching 50 and 100 top-1% hits on all six tasks, averaging over successful runs (Appendix, Tables 3 and 4). It reaches both counts in every seed, whereas some baselines do not. These results support its use when seeking larger sets of high-reward molecules.

4.3 RQ2: What the disagreement term contributes

Since the preceding controls change more than the disagreement term, therefore, to isolate its contribution, we set $c = 0$, giving $A_t(s) = \bar{\mu}_{t-1}(s)$, while retaining GDA's coverage, child selection and memory rules. The dispersion therefore no longer affects acquisition here.

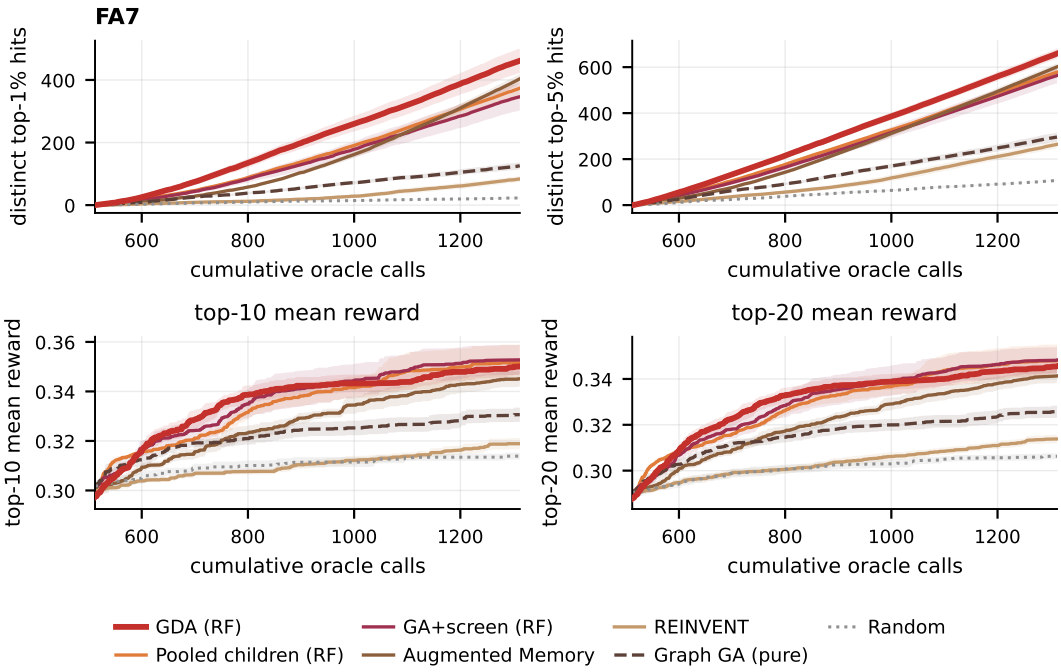

Figure 3: FA7: top-1% and top-5% hit yields (top row) and running top-10 and top-20 mean GEAM rewards (bottom row). Oracle call and molecule counting conventions follow Figure 2. Shading shows standard errors across five seeds.

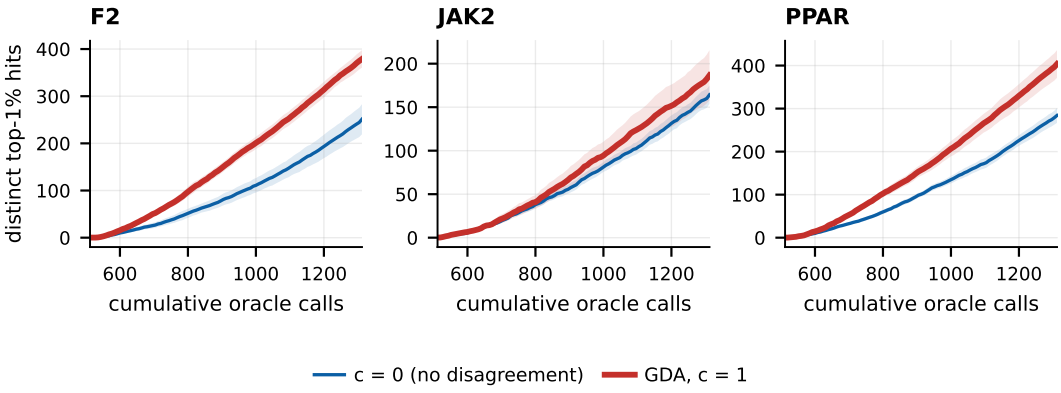

Figure 4: GDA with and without the disagreement bonus. Setting $c = 0$ retains the mean offspring score and the other selection components.

Removing the bonus reduces mean hit yield by 127.8 ± 31.7 on F2 and 119.6 ± 40.3 on PPAR (paired $t = 4.0$ and $t = 3.0$, respectively). GDA performs better in every seed on these two tasks. In the $c = 0$ ablation, the mean oracle burden for reaching 100 top-1% hits also increases by approximately 172 calls on F2 and 107 on PPAR (Figure 4). The JAK2 difference is smaller and less consistent, at 22.4 ± 37.6 hits in favor of GDA, with gains in three of five seeds.

On F2, GDA and Pooled children achieve similar final hit yields (Appendix F), yet removing GDA's disagreement bonus substantially reduces its yield. The controlled ablation therefore reveals a benefit of disagreement within GDA that is not apparent from the comparison with pooled selection alone.

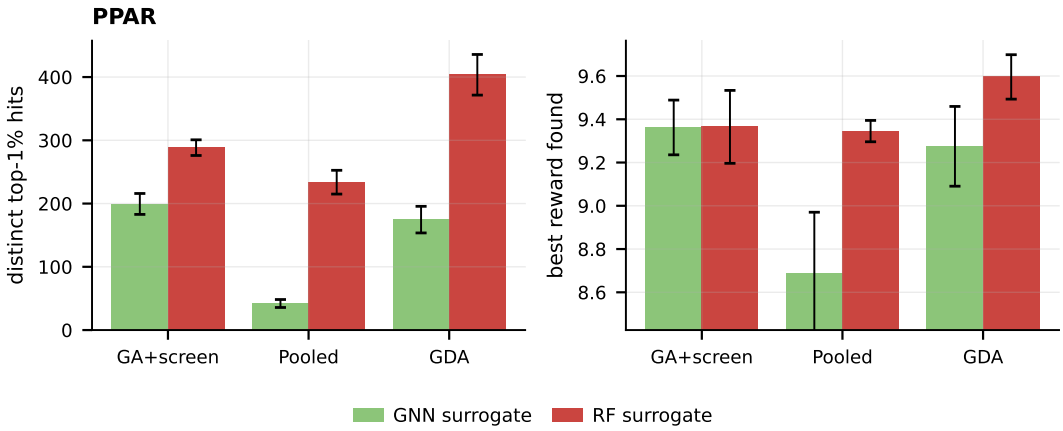

Figure 5: Acquisition comparisons with GNN and RF ensembles on PPAR. The best reward panel uses a truncated y-axis to make differences among methods visible.

4.4 RQ3: Dependence on the surrogate

Generator disagreement relies on the surrogate's predicted rewards. We compare GDA, Pooled children and GA+screen with two surrogate families: ensembles of random forests over Morgan fingerprints and ensembles of graph neural networks (GNN) (Figure 5; the other tasks are in Appendix F).

GDA has the highest mean hit yield in five of the six task–surrogate combinations. On PPAR with the GNN surrogate, GA+screen has a numerically slightly higher mean hit yield, although the standard-error bars overlap. GDA exceeds Pooled children in all six combinations. RF ensembles yield more hits than GNN ensembles for every matched task and acquisition rule.

4.5 Ablations

Proposal operator. Replacing Graph GA with BRICS fragment recombination leaves GDA's acquisition rule unchanged while changing the offspring distribution. On F2 and JAK2, the fragment operator increases internal diversity and scaffold count and reduces the mean SA score by 0.97 and 0.59, respectively, while producing fewer hits (Appendix J).

Feasibility filtering. Proposals that fail the SA cutoff are rejected before oracle evaluation. On F2, tightening the cutoff lowers the SA scores of queried molecules but also reduces hit yield and increases the calls needed for 100 hits (Appendix K). This quantifies the cost of stricter filtering under the tested settings.

5 Conclusion

We introduced GDA, a parent-level acquisition method that uses the mean and dispersion of predicted offspring rewards to guide molecular discovery. With RF ensembles, GDA achieves higher mean hit yields than Graph GA, REINVENT, and Augmented Memory on all three DOCKSTRING tasks. Its advantage over pooled candidate selection holds with both surrogate families investigated. Across all three GEAM tasks, GDA-RF improves top-1% and top-5% hit yields over GA+screen while reducing oracle burden. Ablations show that the disagreement bonus contributes to performance, with gains varying across tasks, while proposal and feasibility choices introduce trade-offs between hit yield, structural diversity, and synthetic-accessibility scores. Together, these findings support predicted offspring distributions as a useful basis for parent selection under limited oracle budgets. Our evaluation focuses on computational molecular objectives with SA filtering. Future work could extend GDA to learned parent-conditioned generators and experimentally evaluate promising candidates identified by the search.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the manuscript. We reviewed the AI-assisted material. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work uses public molecular benchmarks and computational evaluations, without human or animal experiments. Docking rewards and molecular-property filters do not establish biological efficacy, safety, or synthesizability. Generated candidates require further validation before practical use. Molecular optimization also has potential dual-use risks, which should be considered in downstream applications.

REPRODUCIBILITY STATEMENT

Section 3 specifies GDA and its algorithm. Appendices A and B provide the theoretical assumptions, proofs, and scope of the guarantees, while Appendix C describes the proposal operators. Experimental objectives, data processing, splits, seeds, budgets, and model settings are documented in Appendix D. Evaluation metrics are defined in Appendix E.

REFERENCES

Dhruv Agarwal, Manoj Ghuhan Arivazhagan, Rajarshi Das, Sandesh Swamy, Sopan Khosla, and Rashmi Gangadharaiah. Searching for optimal solutions with LLMs via bayesian optimization. In *International Conference on Learning Representations*, 2025.

Nathan Brown, Marco Fiscato, Marwin H. S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3): 1096–1108, 2019. doi: 10.1021/acs.jcim.8b00839.

Jörg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem*, 3(10):1503–1507, 2008. doi: 10.1002/cmdc.200800178.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1:8, 2009. doi: 10.1186/1758-2946-1-8.

Tianfan Fu, Wenhao Gao, Connor W. Coley, and Jimeng Sun. Reinforced genetic algorithm for structure-based drug design. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-0895.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W. Coley. Sample efficiency matters: A benchmark for practical molecular optimization. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1551.

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

Jeff Guo and Philippe Schwaller. Augmented memory: Sample-efficient generative molecular design with reinforcement learning. *JACS Au*, 4(6):2160–2172, 2024. doi: 10.1021/jacsau.4c00066.

Jeff Guo and Philippe Schwaller. Directly optimizing for synthesizability in generative molecular design using retrosynthesis models. *Chemical Science*, 16:6943–6956, 2025. doi: 10.1039/d5sc01476j.

Jeff Guo, Junwu Chen, Anthony GX-Chen, and Philippe Schwaller. Sample-efficient generative molecular design using memory manipulation. *Nature Machine Intelligence*, 8(3):449–460, 2026. doi: 10.1038/s42256-026-01200-4.

Jan H. Jensen. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science*, 10:3567–3572, 2019. doi: 10.1039/c8sc05372c.

Hyeonah Kim, Minsu Kim, Sanghyeok Choi, and Jinkyoo Park. Genetic-guided GFlowNets for sample efficient molecular optimization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 42618–42648, 2024. doi: 10.52202/079017-1350.

Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Saee Paliwal, Arash Vahdat, and Weili Nie. Molecule generation with fragment retrieval augmentation. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. doi: 10.52202/079017-4211.

Seul Lee, Seanie Lee, Kenji Kawaguchi, and Sung Ju Hwang. Drug discovery with dynamic goal-aware fragments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26731–26751, 2024b.

Shitong Luo and Connor W. Coley. Efficient and programmable exploration of synthesizable chemical space. *Transactions on Machine Learning Research*, 2026.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14:265–294, 1978. doi: 10.1007/BF01588971.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9:48, 2017. doi: 10.1186/s13321-017-0235-x.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 1015–1022, 2010.

Morgan Thomas, Raul Astudillo, Yasir S. Raouf, Carles Navarro, Gianni De Fabritiis, and Andreas Bender. Generative virtual screening: from search to generate and back. *Digital Discovery*, 2026. doi: 10.1039/d6dd00025h.

Austin Tripp and José Miguel Hernández-Lobato. Genetic algorithms are strong baselines for molecule generation. *arXiv preprint arXiv:2310.09267*, 2023.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M. Tseng, Sergey Levine, and Tommaso Biancalani. Feedback efficient online fine-tuning of diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48892–48918, 2024.

Jason Yang, Wenda Chu, Daniel Khalil, Raul Astudillo, Bruce J. Wittmann, Frances H. Arnold, and Yisong Yue. Steering generative models with experimental data for protein fitness optimization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4866.

Mert Yuksekgonul, Daniel Koceja, Xinhao Li, Federico Bianchi, Jed McCaleb, Xiaolong Wang, Jan Kautz, Yejin Choi, James Zou, Carlos Guestrin, and Yu Sun. Learning to discover at test time. In *International Conference on Machine Learning*, 2026.

A EXPECTED MAXIMUM AND THE EMPIRICAL ACQUISITION SCORE

Proof of Proposition 1. Fix the search history, surrogate and parent. Let $K \geq 1$ be an integer and $v_s \geq 0$. The variables V_k are predicted offspring rewards with common conditional mean m_s and the moment generating function bound in the proposition. For any $\lambda > 0$, Jensen's inequality gives

$$\exp\left(\lambda\left[\mathbb{E}\max_{k\leq K} V_k - m_s\right]\right) \leq \mathbb{E}\exp\left(\lambda \max_{k\leq K}(V_k - m_s)\right) \leq \sum_{k=1}^{K} \mathbb{E}e^{\lambda(V_k - m_s)} \leq Ke^{\lambda^2 v_s^2/2}.$$

Taking natural logarithms and dividing by λ yields

$$\mathbb{E}\max_{k\leq K} V_k \leq m_s + \frac{\log K}{\lambda} + \frac{\lambda v_s^2}{2}.$$

For $K > 1$ and $v_s > 0$, choosing $\lambda = \sqrt{2\log K}/v_s$ proves the claim. For $K = 1$, the expectation is m_s. For $v_s = 0$, letting $\lambda \to \infty$ gives the result. The bound does not require independence among $V_1, \ldots, V_K$. □

Scope. This inequality concerns the expected maximum of random predictions, conditional on a fixed surrogate. It neither bounds oracle rewards nor establishes a confidence interval. The coefficient follows from optimizing the exponential moment upper bound above. We do not claim that it is the smallest valid coefficient for every finite K or every offspring distribution.

The implementation uses the sample mean and sample standard deviation of the surviving predictions in equation 3–equation 4. These are not known population parameters: the sub-Gaussian scale v_s can exceed the population standard deviation, and sample estimates have sampling error. Screening and deduplication also affect the offspring distribution. The population assumptions must hold for the variables to which the proposition is applied. GDA does not estimate or certify such a scale. The bound motivates the $\sqrt{2\log K}$ dependence, while the empirical score remains an acquisition rule with multiplier c. Its coefficient uses the configured K and its sample statistics use the actual survivor count $n_t(s)$, with zero dispersion for a single survivor.

The empirical acquisition score can also be lower than the highest predicted reward among the sampled offspring. For $K = n = 16$, fifteen predictions equal to zero and one prediction equal to one have mean $1/16$ and sample standard deviation $1/4$. At $c = 1$, their acquisition score is about 0.651, below the sampled maximum of one. Dispersion can also reflect low predicted rewards, so a larger bonus alone does not establish a better parent.

B GREEDY SELECTION AND ITS SCOPE

The selection rule. For a finite reference set U, eligible set E, scores v, and requested size $h \leq |E|$, initialize $S_0 = \varnothing$ and $d_0(u) = 0$. At step $\ell = 1, \ldots, h$, choose

$$s_\ell \in \operatorname*{arg\,max}_{s\in E\setminus S_{\ell-1}} \left\{\zeta v(s) + (1-\zeta)\sum_{u\in U} \max\{\mathrm{sim}(u,s) - d_{\ell-1}(u), 0\}\right\}, \tag{9}$$

then set $S_\ell = S_{\ell-1}\cup\{s_\ell\}$ and $d_\ell(u) = \max\{d_{\ell-1}(u), \mathrm{sim}(u, s_\ell)\}$. A fixed candidate order resolves ties. Parent selection uses $U = \mathcal{M}_{t-1}$, $E = \{s \in U : n_t(s) > 0\}$, $v = A_t$, and $h = \min\{B, |E|\}$. Memory selection uses $U = E = \mathcal{K}_t$, $v = q_t$, and $h = \min\{J, |\mathcal{K}_t|\}$. If $h = 0$, the result is empty.

The classical guarantee. With $\mathrm{sim} \in [0,1]$ and $\zeta \in [0,1]$, the coverage marginal in equation 9 decreases as S grows. The quality term is modular, so Φ_U is submodular even when some quality scores are negative. If all scores are nonnegative, Φ_U is also normalized and monotone. Greedy then has the standard cardinality-constrained guarantee (Nemhauser et al., 1978):

$$\Phi_U(S_h; v) \geq \left[1 - \left(1 - \frac{1}{h}\right)^h\right] \max_{S\subseteq E,\,|S|\leq h} \Phi_U(S; v) \geq (1-1/e) \max_{S\subseteq E,\,|S|\leq h} \Phi_U(S; v), \qquad h \geq 1.$$

Indeed, submodularity and monotonicity imply that the best remaining marginal gain is at least $[\Phi_U(S^*; v) - \Phi_U(S_\ell; v)]/h$ for an optimal feasible set S^*. Applying this inequality for h greedy

steps proves the bound. It is a guarantee for this fixed set objective, not for discovery yield or the adaptive search as a whole.

For signed scores, the implementation still fills the requested size, even if remaining marginal gains are negative. The monotone guarantee above may not apply. Adding a common constant to make every score nonnegative preserves the greedy order and the optimum among sets of size h, but changes the objective by a term proportional to $|S|$. A guarantee for that shifted objective is not the same multiplicative guarantee for the original scores. No such guarantee is claimed for signed scores. The implementation uses the summed coverage term. Its balance with quality therefore depends on both reward scale and $|U|$.

Child selection. The ensemble spread $\sigma_{r,t-1}(x)$ in equation 6 measures disagreement among model predictions. We use it as an uncertainty proxy for ranking offspring, without assuming that it bounds the prediction error. The choice $\beta_t = \sqrt{2 \log \max\{N_t, 1\}}$ therefore specifies a ranking rule, not a simultaneous confidence guarantee. The medians for γ_t use the concatenated survivor lists of selected parents, retaining repeated entries when an offspring has several proposing parents. When $\mathrm{med}(\mathrm{nov}_t) > \varepsilon_{\mathrm{nov}}$, the median novelty bonus equals $\tilde{\gamma}\ \mathrm{med}(\sigma_r)$. This scales novelty relative to the unweighted ensemble spread σ_r, while β_t separately weights the ensemble spread term. Duplicate finalists are removed after child selection without adding replacements.

C Proposal operators and parent context

Graph GA. Crossover pairs the focal parent with another memory molecule sampled with weights proportional to $\exp(q_{t-1}(u)/\varsigma)$, followed by mutation (Jensen, 2019; Tripp & Hernández-Lobato, 2023). The temperature is derived from the spread of partner quality scores, with a positive numerical floor. The parent itself is used if no alternative partner exists. The operator rejects invalid products and duplicates within each parent's proposal list before the shared screening step. Thus even an operator without trainable weights changes its proposal distribution as the memory changes.

Fragment recombination. BRICS rules (Degen et al., 2008) decompose the parent and a support set of its labelled direct children. The support set contains at most r children ranked by predicted mean plus an ensemble spread bonus, using the previous round's candidate count to set that bonus. Additional source molecules are sampled from memory when needed. The operator deduplicates and caps the fragment pool, then recombines compatible fragments. Rebuilding can add further memory sources. Products do not need to retain a particular parent scaffold. The same structural, SA and previously queried screens apply. Both operators supply parent-conditioned proposals, but their proposal distributions differ.

D Experimental details

DOCKSTRING Objectives. Let $s(x, t)$ be the AutoDock Vina score of molecule x against target t. With QED denoting the quantitative estimate of drug-likeness, the DOCKSTRING objectives are

$$f_{\mathrm{F2}}(x) = s(x, \mathrm{F2}) + 10\,(1 - \mathrm{QED}(x)), \tag{10}$$

$$f_{\mathrm{PPAR}}(x) = \max_{t \in \{\mathrm{PPARA,PPARD,PPARG}\}} s(x, t) + 10\,(1 - \mathrm{QED}(x)), \tag{11}$$

$$f_{\mathrm{JAK2}}(x) = s(x, \mathrm{JAK2}) - \min\{s(x, \mathrm{LCK}) + 8.1, 0\} + 10\,(1 - \mathrm{QED}(x)). \tag{12}$$

We maximize $y = -f$ without reward normalization. Lower docking scores indicate stronger predicted binding. The F2 objective favors binding to F2, while the PPAR objective improves the least favorable docking score across the three PPAR receptors. The JAK2 objective favors binding to JAK2 while penalizing strong predicted binding to LCK, specifically when the LCK docking score is below -8.1. All three objectives include the same QED-based drug-likeness penalty and follow the DOCKSTRING reference implementation.

GEAM Objectives. For target t, let $d_t(x)$ denote the QuickVina2 docking score, $\mathrm{QED}(x)$ the quantitative estimate of drug-likeness, and $\mathrm{SA}(x)$ the synthetic-accessibility score. GEAM maximizes

the multiplicative reward

$$R_t(x) = \mathrm{clip}\left(-\frac{d_t(x)}{20}; 0, 1\right) \mathrm{clip}(\mathrm{QED}(x); 0, 1) \, \mathrm{clip}\left(\frac{10 - \mathrm{SA}(x)}{9}; 0, 1\right). \tag{13}$$

Thus, high reward requires favorable docking, high QED, and low SA simultaneously. QuickVina2 scores are used for all GEAM oracle labels. The source docking and property columns are not used as oracle labels.

Corpora and partitions. We retain molecules scored on every required target that pass structural screening: a connected, neutral graph with no radical electrons, elements C, N, O, F, P, S, Cl, Br or I, and at most 64 heavy atoms. Bond orders are 1–3 after Kekulization. The resulting corpora contain 244,544, 244,542 and 244,541 molecules for F2, JAK2 and PPAR. For each task, we withhold the 2,446 highest reward molecules (top 1%). Using the fixed split seed `20260101`, we randomly assign the remaining molecules to pretraining, initialization, and reference partitions that contain approximately 60%, 5%, and 34% of the complete filtered corpus, respectively.

The GEAM search corpus uses molecular identities from the official ZINC250k file. Structural filtering and identity deduplication leave 168,129 molecules per target. Using the fixed split seed `20260101`, the GEAM corpus is divided into hidden (1,682 molecules), pretraining (100,877), initialization (8,406), and reference (57,164) partitions. The shared D_0 contains 512 molecules sampled without replacement from the initialization partition. The pretraining and initialization partitions define the visible corpus. The hidden partition is reserved for post-hoc evaluation. Source docking and property columns are not used as oracle labels.

Budget and replication. All methods use the same 512 initial labels within a task and seeds $7, 11, 13, 17, 19$. Results use completed batches at 1,312 cumulative evaluations. One charged evaluation returns one molecule's task objective, whether supplied by the corpus table, a cache, or live docking. Failed evaluations also consume budget but do not count as hits. Oracle burden means include only runs that attain the requested hit count, as indicated in the tables.

Acquisition settings. GDA uses batch limit $B = 8$, rollouts $K = 16$, memory size $J = 64$, $\zeta = 0.7$, $c = 1$, $\tilde{\gamma} = 1$ and $\varepsilon_{\mathrm{nov}} = 10^{-12}$ on all the tasks. Thus $\kappa = \sqrt{2 \log K} \approx 2.355$, while $\beta_t = \sqrt{2 \log \max\{N_t, 1\}}$ changes with the screened candidate count. Novelty scaling follows equation 6. It matches the median novelty bonus to the unweighted ensemble spread. GA+screen and Pooled children use $\zeta = 1$ for memory selection. The $c = 0$ (Figure 4) ablation retains $\zeta = 0.7$ and changes only the disagreement multiplier.

Surrogates. Each ensemble has five members and is refit after every nonempty round. RF members contain 300 trees trained on bootstrap samples of 80% of the labelled set, using 2048-bit Morgan fingerprints with radius 2. GNN members have four message-passing layers of width 192 and dropout 0.1. They use bootstrap training samples, AdamW with learning rate 10^{-3} and weight decay 10^{-5}, batch size 64, and at most 30 epochs with validation-based early stopping (patience five). Predictions are returned in the original reward units.

Reinforcement-learning baselines. For all the tasks, our REINVENT and Augmented Memory implementations share a task-specific, three-layer GRU prior with hidden size 512, pretrained on SMILES from the pretraining partition for 40 epochs and selected by validation likelihood. Both agents are initialized from this prior and adapted to D_0 for five epochs before online search. Policy updates use the augmented-likelihood objective with $\sigma = 120$, Adam with learning rate 5×10^{-4}, and the transformed score $S(x) = \mathrm{clip}((y(x) - 4.25)/5.75, 0, 1)$. Evaluation metrics use the original rewards. Augmented Memory additionally maintains a buffer of the 100 highest-scoring molecules and performs ten replay updates with randomized SMILES after each oracle batch, without additional oracle evaluations.

Evaluation cost. For DOCKSTRING, a live objective evaluation requires one target docking for F2, two for JAK2 and three for PPAR. Corpus lookups and cached scores require no new docking. We used a fixed docking seed throughout. Mean recorded online runtime for the three RF-based selection policies ranges from 4.3 to 14.3 hours across tasks. Initialization, pretraining and checkpoint writing are not included in this timing measure. For GEAM, each live objective evaluation requires one docking calculation against the task's target. Failed docking attempts still consume budget.

Table 2: Final distinct GEAM top-5% hit yields after 800 online oracle calls. Values are mean ± standard error over five paired seeds.

Method	PARP1	BRAF	FA7
GDA (RF) (ours)	608.2 ± 25.9	676.4 ± 11.9	659.0 ± 21.9
GA+screen (RF)	543.2 ± 26.7	606.6 ± 16.0	563.4 ± 33.8
Pooled (RF)	509.2 ± 31.6	618.6 ± 15.4	578.2 ± 15.9
Graph GA	365.0 ± 10.1	416.0 ± 28.7	296.0 ± 19.0
REINVENT	253.0 ± 35.2	189.6 ± 36.4	264.2 ± 15.6
Augmented Memory	578.2 ± 11.1	662.8 ± 7.2	601.2 ± 15.1
Random	105.8 ± 0.8	105.8 ± 0.8	106.6 ± 0.8

E Evaluation metrics

Hit yield counts successful online queries with distinct InChIKeys, falling back to canonical SMILES when needed. Hit yield and oracle burden are defined in Section 3. We also report the following measures.

Best and top-k rewards. Best-so-far reward is the largest observed reward. Top-k mean reward averages the k largest rewards available at a given point, including D_0 and subsequent evaluations. We use $k = 10$ and $k = 20$. Together, these measure the strongest individual result and the quality of a small candidate set. Each reward curve starts at the value computed from the shared initialization.

Structural diversity. We measure structural diversity among the 100 highest-reward molecules from successful online queries in each run. IntDiv1 equals one minus the average Tanimoto similarity over all pairs of distinct molecules, using Morgan fingerprints with radius 2 and counting each pair once. We also report the number of distinct Bemis–Murcko scaffolds in the same set. Results for these metrics are reported in the proposal-operator comparison in Appendix J.

Synthetic accessibility. We report the mean fragment-based SA score (Ertl & Schuffenhauer, 2009) over the same top 100 queries in the proposal-operator comparison (Appendix J). Lower SA scores indicate lower estimated synthesis difficulty. Appendix K investigates how stricter SA cutoffs affect discovery performance and the median SA score of successful online queries.

F Per-task results

Figures 6 and 7 report hit yield and top-k reward for F2 and JAK2. GDA has the highest mean final top-1% yield in these RF comparisons although reward-based rankings differ: on JAK2, Pooled children has higher final top-10 and top-20 mean rewards. The surrogate comparisons in Figures 11 and 12 show higher mean hit yield for RF than GNN ensembles and gains for GDA over pooled selection with either surrogate. Figure 10 reports best-so-far reward, which complements the hit counts and top-k averages.

For GEAM, Figures 8–9 show top-1% and top-5% hit yields and running top-10 and top-20 mean rewards for PARP1 and BRAF. Table 2 reports the final top-5% hit yields for all three GEAM tasks.

G Oracle burden at the top-1% threshold

For DOCKSTRING, GDA has the lowest mean oracle burden in seven of the nine task–count combinations in Table 3. On PPAR, reaching 100 hits requires 807 cumulative evaluations, compared with 991 for Pooled children, a reduction of about 184 task evaluations. The exceptions to the overall ordering occur at 10 hits on JAK2 and PPAR. Random's zero top-1% yield is expected because it samples only the visible pretraining partition. Against the three principal baselines (Graph GA, REINVENT, and Augmented Memory), GDA achieves higher mean top-1% and top-5% hit yields on all three tasks and a lower mean oracle burden for every reported hit-count target reached by each baseline (Table 3 and Appendix F).

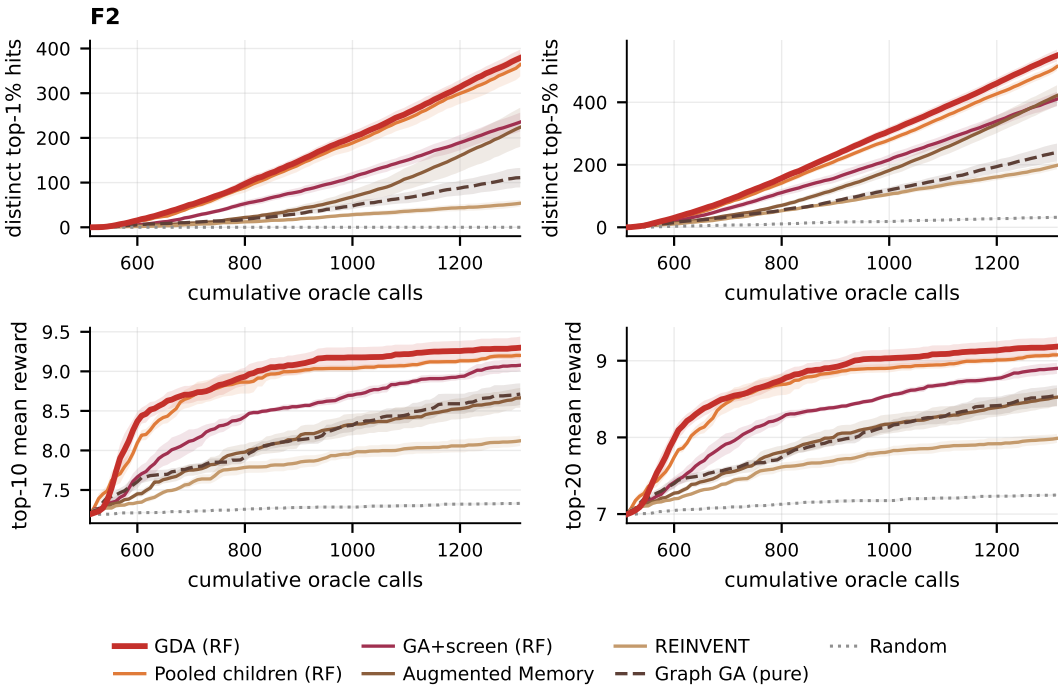

Figure 6: F2 DOCKSTRING: hit yield and top-k mean reward. Shading shows standard errors over five seeds.

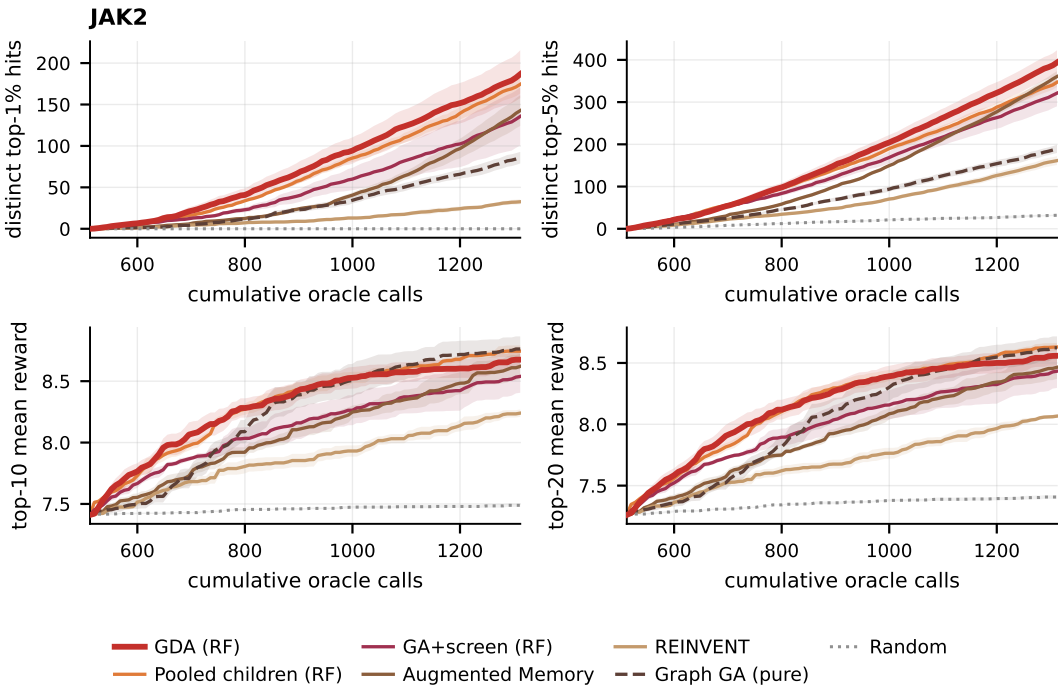

Figure 7: JAK2 DOCKSTRING: hit yield and top-k mean reward. Shading shows standard errors over five seeds.

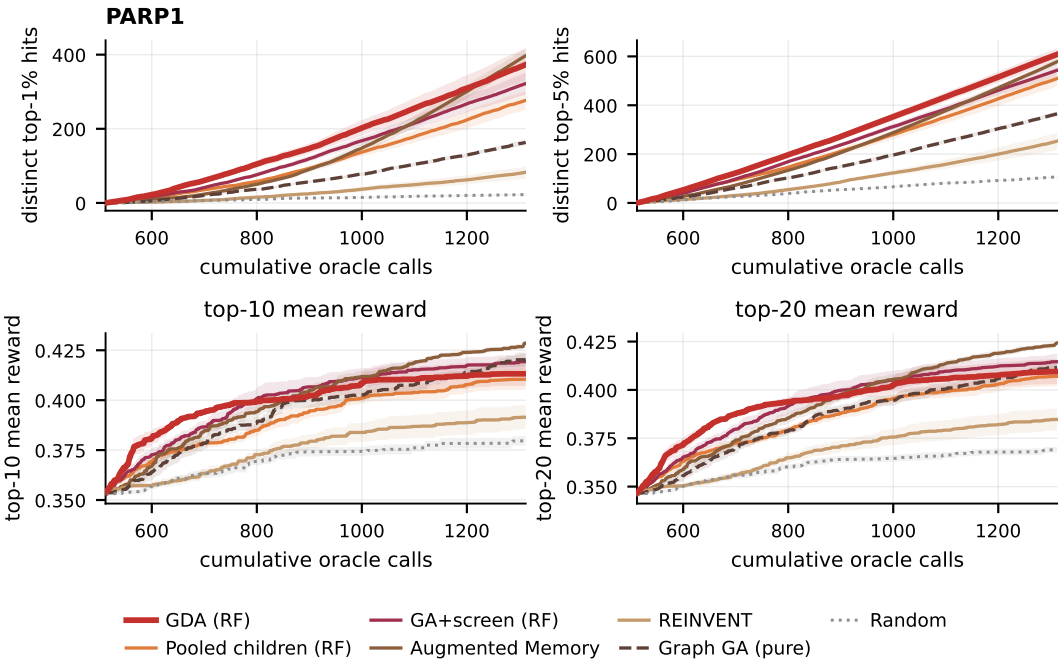

Figure 8: PARP1 GEAM: hit yield and top-k mean reward. Shading shows standard errors over five seeds.

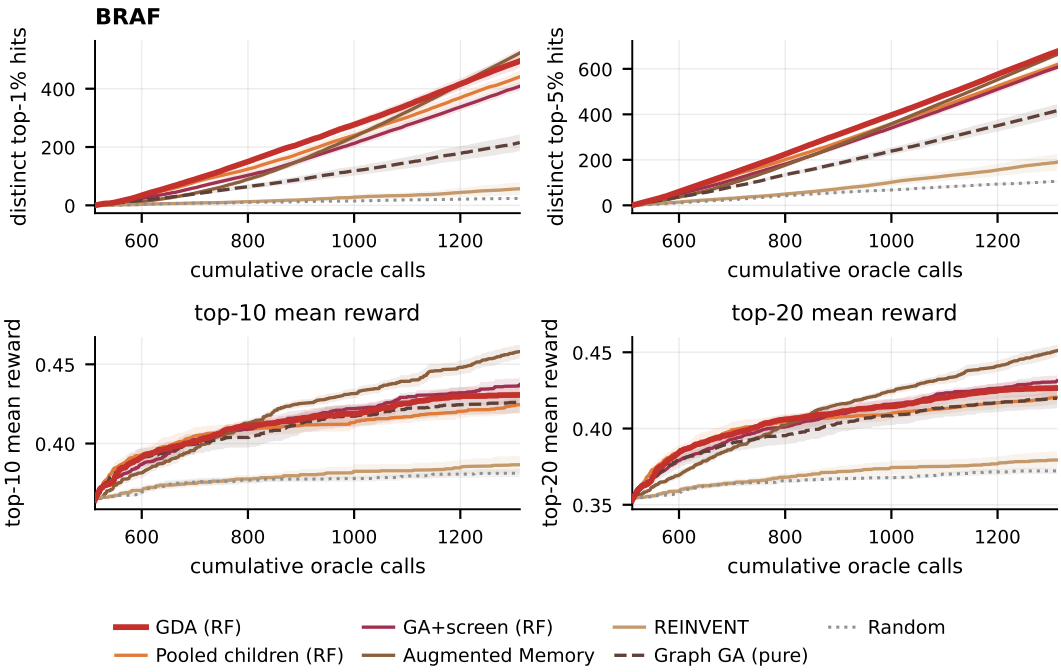

Figure 9: BRAF GEAM: hit yield and top-k mean reward. Shading shows standard errors over five seeds.

For GEAM, GDA also reaches the 50- and 100-hit top-1% milestones earlier than GA+screen on every target (Table 4). On BRAF, GDA obtains 495.0 ± 24.5 top-1% hits and 676.4 ± 11.9 top-5% hits. On PARP1, it obtains 373.2 ± 44.6 and 608.2 ± 25.9, respectively.

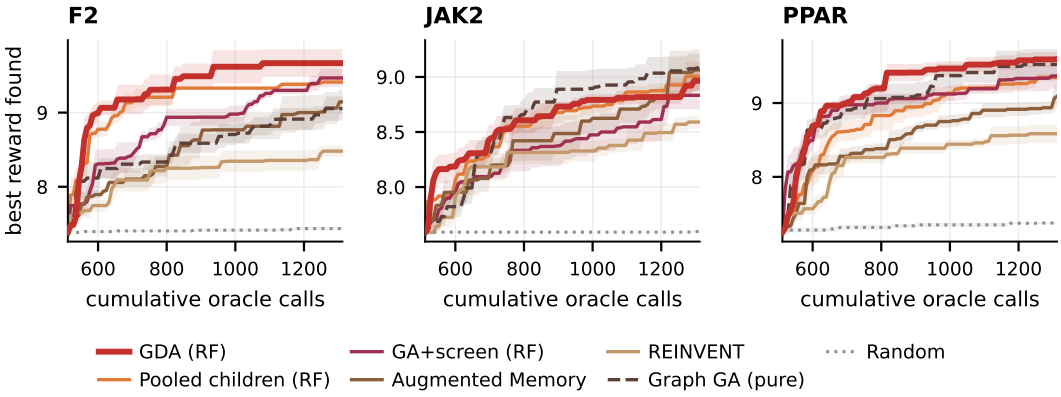

Figure 10: Best-so-far reward on all three DOCKSTRING tasks, with standard errors over five seeds.

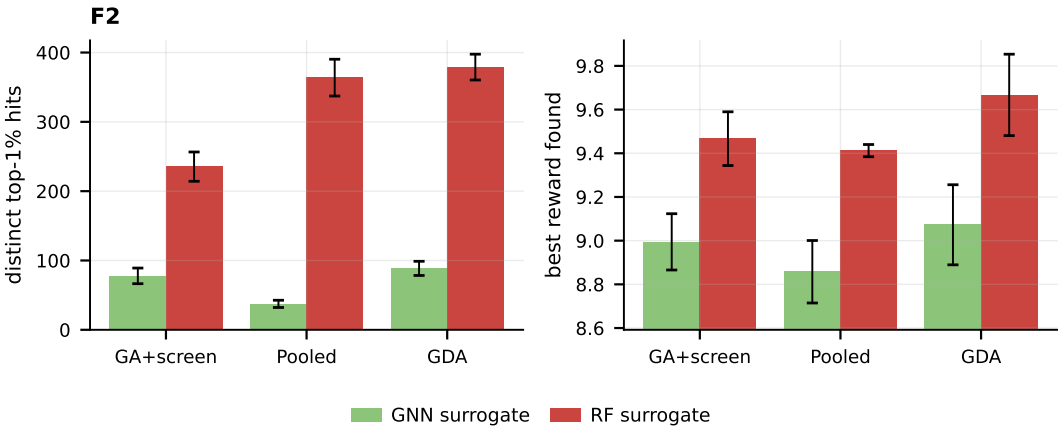

Figure 11: F2: acquisition rules with two surrogate families. The best reward axis is truncated.

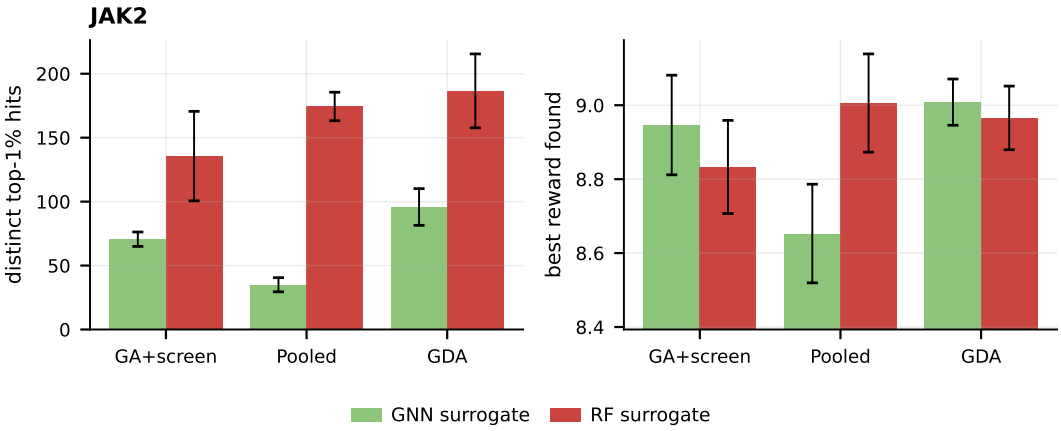

Figure 12: JAK2: acquisition rules with two surrogate families. The best reward axis is truncated.

Table 3: Cumulative oracle calls to reach 10, 50 and 100 distinct top-1% hits (lower is better) for DOCKSTRING. Means include only seeds that reach the requested count. Superscripts show the number of successful seeds when fewer than five do. A dash indicates no successful seeds. Random is omitted because it reaches none of these counts on any task within the budget. Bold identifies GDA, not column minima.

	F2			JAK2			PPAR		
Method	@10	@50	@100	@10	@50	@100	@10	@50	@100
GDA (ours)	**582**	**705**	**810**	**634**	**854**	**1050**	**590**	**699**	**807**
Pooled children	589	722	826	632	871	1063	645	822	991
GA+screen	647	802	973	662	993	1173^{3}	582	716	857
Augmented Memory	720	963	1118	742	1051	1194^{4}	688	870	987
REINVENT	786	1112^{2}	–	899	–	–	830	1166^{4}	1280^{1}
Graph GA	674	1030	1172^{2}	787	1109	1304^{2}	630	853	1054

Table 4: Cumulative oracle calls required to reach 10, 50, and 100 distinct top-1% hits for GEAM. Lower is better. Means are computed over seeds that reach the requested count. Superscripts show the number of successful seeds when fewer than five do. Random is omitted because it reaches none of these counts on any task within the budget. Bold identifies GDA.

	FA7			BRAF			PARP1		
Method	@10	@50	@100	@10	@50	@100	@10	@50	@100
GDA (ours)	**555**	**660**	**748**	**546**	**628**	**719**	**556**	**677**	**801**
Pooled children	545	706	821	543	642	752	579	766	921
GA+screen	581	723	851	558	676	799	577	721	859
Augmented Memory	598	777	898	591	724	821	614	801	925
REINVENT	766	1141	1272^{1}	802	966^{1}	1187^{1}	753	1085^{4}	1228^{2}
Graph GA	603	877	1179	571	764	956	634	869	1087

H ORACLE BURDEN AT THE TOP-5% THRESHOLD

Table 5 repeats the oracle burden analysis at the top-5% threshold for all the DOCKSTRING tasks. Every non-random method in the table reaches all three requested hit counts in all five seeds. Random reaches 10 hits but never 50 or 100. GDA has the lowest mean burden for 100 hits on each task, although its JAK2 mean is nearly identical to that of Pooled children and rounds to the same value. Other methods reach some of the smaller counts sooner.

Table 5: Cumulative oracle calls to reach 10, 50 and 100 distinct top-5% hits where lower is better. All five seeds reach each reported count. A dash means no seed reaches that count. Bold identifies GDA.

	F2			JAK2			PPAR		
Method	@10	@50	@100	@10	@50	@100	@10	@50	@100
GDA (ours)	**560**	**644**	**721**	**563**	**697**	**806**	**558**	**636**	**718**
Pooled children	554	655	739	556	681	806	555	684	820
GA+screen	571	689	788	557	691	852	550	648	734
Augmented Memory	582	750	869	590	774	901	598	742	834
REINVENT	595	787	992	616	910	1125	611	797	974
Graph GA	576	786	955	589	829	1019	566	702	843
Random	771	–	–	754	–	–	734	–	–

I WALL-CLOCK COST

Figure 13 compares the best reward found against recorded online runtime. For the RF-based selection policies, the oracle stage accounts for approximately 78–93% of mean per-run time fractions

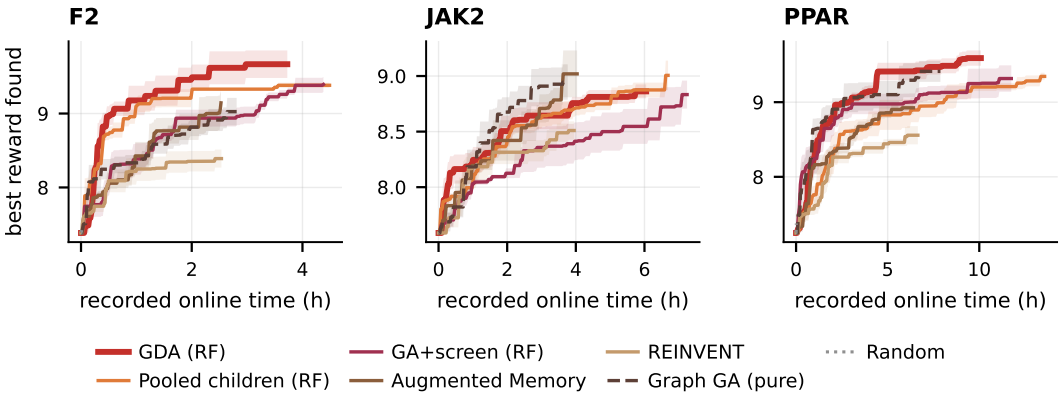

Figure 13: Best reward against recorded online time. Each method's curve ends at the shortest recorded duration among its five seeds. Shading shows standard errors.

across tasks. Runtime coefficients of variation across seeds range from approximately 5 to 11% for these policies. On JAK2, GDA averages 6.6 hours, compared with 7.2 for Pooled children and 7.7 for GA+screen, while achieving higher mean hit yield. These measurements include generation, selection and model updates, but exclude setup and checkpoint writing. They describe the recorded runs on shared hardware. The experiments were conducted on a Slurm cluster of NVIDIA A100-SXM4 (80 GB) nodes, with each run allocated 8 CPU cores, 80 GB of memory and at most one GPU.

J Proposal operator ablation

Figure 14 compares Graph GA with BRICS fragment recombination under the same GDA acquisition rule. Table 6 summarizes the changes between operators, pairing runs with the same seed.

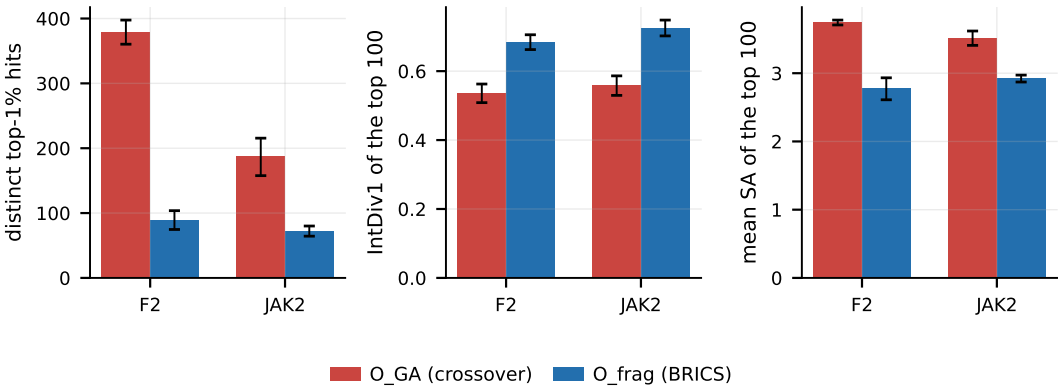

Figure 14: Graph GA and BRICS fragment recombination with the same GDA acquisition rule. Bars show means over five seeds. Error bars show standard errors of the means.

BRICS recombination yields fewer top-1% hits on both tasks, but its top-100 set has higher internal diversity, more distinct scaffolds, and a lower mean SA score (Table 6). These results show that the proposal operator affects the trade-off between hit yield, structural diversity, and estimated synthetic accessibility, even when the GDA acquisition rule is unchanged.

K Feasibility filtering

The SA gate rejects proposals before oracle evaluation. Table 7 reports the F2 results as the cutoff is tightened from 4.0 to 3.5 and 3.0.

Table 6: Changes from Graph GA to BRICS fragment recombination under GDA. Entries are mean paired differences (BRICS minus Graph GA) $\pm$ the standard error of the mean over five seeds. The top 100 are the highest-reward successful online queries in each run.

Metric	F2	JAK2
Distinct top-1% hits	-289.8 ± 26.0	-114.4 ± 25.5
IntDiv1 of the top 100	$+0.148 \pm 0.026$	$+0.167 \pm 0.036$
Scaffolds in the top 100	$+37.2 \pm 15.5$	$+30.8 \pm 6.8$
Mean SA of the top 100	-0.97 ± 0.19	-0.59 ± 0.07

Table 7: SA cutoff comparison for GDA on F2. Hit yields and best rewards are means $\pm$ standard errors over five seeds. Oracle burdens are mean cumulative calls to 100 top-1% hits, including the 512 initialization calls. SA medians pool successful online queries across all five seeds.

τ_{SA}	4.0	3.5	3.0
Distinct top-1% hits	379.0 ± 18.6	278.2 ± 44.1	185.8 ± 15.9
Best reward	9.67 ± 0.19	9.33 ± 0.12	8.92 ± 0.10
Mean calls to 100 hits	810	921	1013
Median SA of online queries	3.76	3.26	2.71

All five seeds reach 100 top-1% hits at each cutoff. Stricter filtering lowers the median SA score while reducing yield and increasing oracle burden. To compensate for screening losses, the configurations with $\tau_{\mathrm{SA}} = 3.5$ and 3.0 request $6K$ and $15K$ valid proposals per parent, respectively. After screening, we retain at most K survivors for surrogate scoring. The actual survivor count may be smaller: we set $\sigma_g = 0$ for a single survivor and exclude parents with none. Thus, this comparison changes both the SA cutoff and the proposal-generation allowance while keeping the oracle budget fixed.

L STRUCTURES OF THE DISCOVERED MOLECULES

Figures 15–17 show the ten highest reward distinct canonical SMILES found by GDA on DOCKSTRING F2, JAK2, and PPAR, pooled across the five seeds. Each molecule is annotated with its reward, SA score and maximum Tanimoto similarity to another molecule in the same task's displayed set.

The sets contain 10, 6 and 7 distinct Bemis–Murcko scaffolds for F2, JAK2 and PPAR, respectively, with mean pairwise similarities of 0.27, 0.27 and 0.34. F2 has a different scaffold for each displayed molecule although some pairs remain similar by fingerprint. PPAR includes several close analogues alongside less similar structures. The structures illustrate variation among the highest scoring candidates. Distinct scaffold counts alone do not imply that the molecules are structurally unrelated.

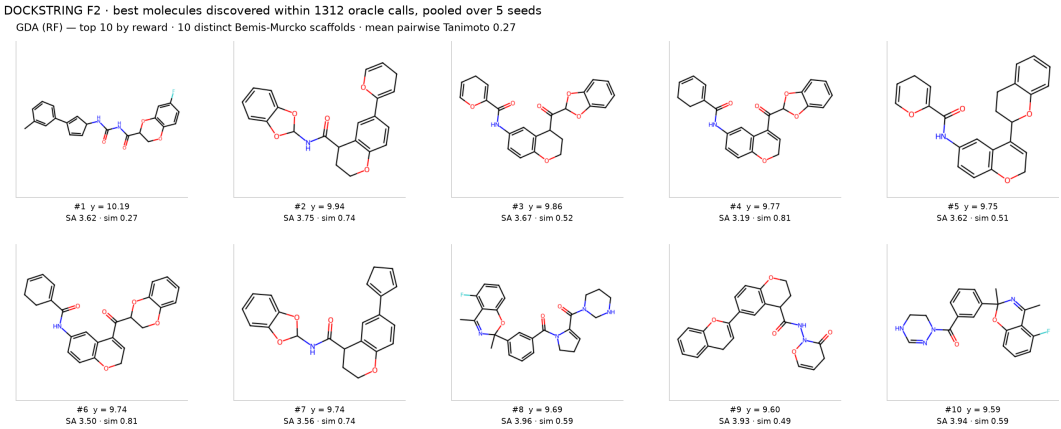

Figure 15: F2: the ten highest reward distinct molecules found by GDA.

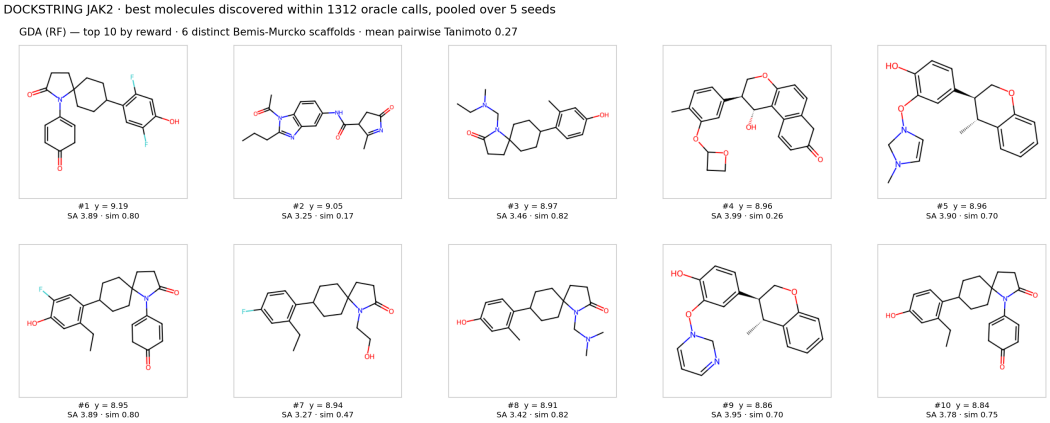

Figure 16: JAK2: the ten highest reward distinct molecules found by GDA.

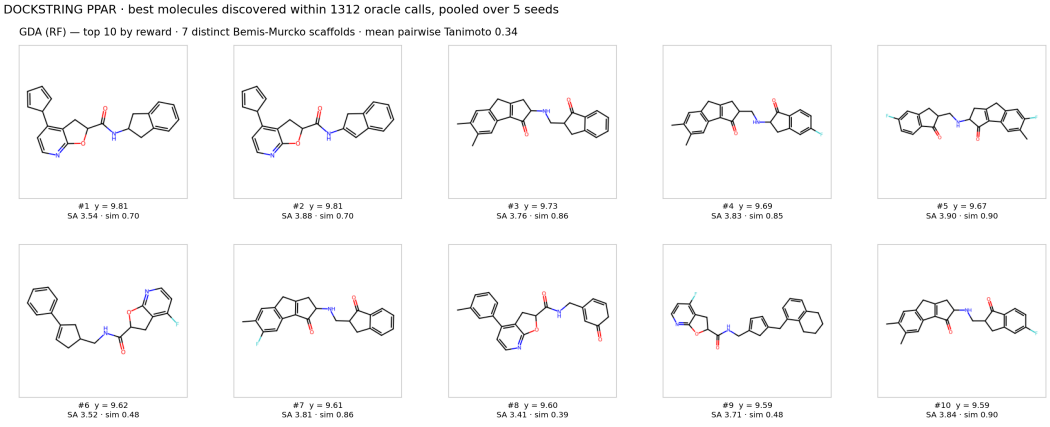

Figure 17: PPAR: the ten highest reward distinct molecules found by GDA.