PREDICTION-REASONING FUSION FOR MOLECULAR HIT DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular hit discovery requires turning imperfect activity predictions into useful assay choices. We study prediction-reasoning fusion, in which a predictive model estimates molecular activity, a language model proposes interpretable acquisition policies, and a policy evaluator measures their utility on development data. The language model requests numerical summaries relating predictions and molecular features to observed outcomes, proposes decision trees and uses evaluation feedback to guide subsequent proposals. A tree guides the next assay batch only when it improves development utility over model-based ranking. Before each subsequent acquisition round, the predictive model is refitted using the initial and acquired labels, and a new tree-development session begins. This automated, agentic procedure produces decision rules that scientists can examine alongside their development support. A theoretical bound relates the transfer of development gains to adaptive data reuse, differences between candidate populations and the extent of policy changes. Experiments on HIV and ALDH1 show higher mean acquisition hits and early discovery than ranking by a multilayer perceptron ensemble and random-tree search, although higher development scores do not always lead to better acquisition outcomes. A supplementary exploratory held-out assessment finds significant hit gains on HIV.

1 INTRODUCTION

Advances in game playing, protein structure prediction and mathematical discovery illustrate how learned models and search can extend scientific and computational capabilities (Silver et al., 2016; Abramson et al., 2024; Romera-Paredes et al., 2024). In molecular hit discovery, decisions must often be made with limited information about the biological processes underlying activity. An assay reveals activity under particular experimental conditions, while the mechanism responsible may remain unknown. Compound activity benchmarks also expose difficulties in generalizing across chemical and assay contexts (Tian et al., 2024). This makes molecular screening a useful setting for studying how predictions and reasoning can jointly inform experimental choices.

Molecular hit discovery seeks experimentally active compounds within a large candidate collection. With a limited assay budget, the order of testing affects which active compounds are discovered and how early they are found. A useful strategy should discover active compounds while making productive use of the earliest experiments. Predictive models provide a natural basis for prioritization, and uncertainty-aware molecular learning offers ways to guide discovery using more than point predictions (Soleimany et al., 2021). Selecting molecules with the highest predicted activity is a strong reference policy. Any additional decision layer must justify departures from that ranking through improved discovery.

We study *prediction-reasoning fusion*, which combines outcome prediction with language-model reasoning about experimental choices. A predictive model estimates candidate outcomes from labeled observations. A large language model (LLM) uses these predictions and contextual information to reason and propose acquisition policies. A policy evaluator checks the proposed rules, measures their utility on development data and returns feedback for subsequent proposals. Work on program and heuristic search, including FunSearch and Evolution of Heuristics, provides examples of coupling LLM proposals with numerical evaluation (Romera-Paredes et al., 2024; Liu et al., 2024a). Our focus

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.

is how such interaction can construct interpretable molecular acquisition policies and whether their development improvements carry into subsequent discovery.

We implement this framework using an ensemble of multilayer perceptrons (MLPs) and LLM-generated decision trees. We use "MLP ranking" to refer to ranking by the ensemble's mean predicted activity. At each acquisition round, the LLM conducts a sequence of query-proposal pairs. It requests summaries relating model predictions and molecular features to observed development outcomes, then proposes a tree whose leaves assign ordinal acquisition priorities. Molecules are ordered first by tree priority and then by MLP score within each priority group. The policy evaluator operates within an evaluation harness that checks each candidate, measures its development hits and returns feedback that can guide the next query and proposal. The best eligible tree is selected when it improves development utility over the current MLP ranking. Otherwise, acquisition uses that ranking. The algorithm automates policy construction, with the LLM choosing evidence queries and policy proposals in response to feedback. This control over successive actions gives the procedure its agentic role.

Each acquisition round selects one batch using the chosen policy. Before the next round, the MLP ensemble is fitted from initialization using the initial labeled collection and all outcomes acquired so far. A new tree-development session then begins with the updated predictive model and the same query-proposal budget. Previous trees and dialogues are excluded from the new session. Acquired outcomes contribute through the predictive model and summaries of known molecular neighbors. Policy construction therefore uses the model and information available at the current decision step.

The LLM receives structured numerical summaries and feature definitions covering activity predictions, ensemble variation, similarity to known active and inactive molecules, and molecular descriptors. Dataset and compound identities, raw molecular structures, unassayed outcomes and held-out test labels are withheld. The harness provides no external lookup tools or target-specific biochemical instructions. These boundaries constrain the information available to policy construction while retaining chemically meaningful descriptors. Development outcomes guide policy construction and selection and are excluded from predictive-model training. Acquisition outcomes measure discovery on molecules whose labels were unavailable when selected and become available for subsequent training. The held-out test set assesses the final deployed model and policy without further fitting or policy development.

Decision trees give the policy layer a practical role in scientific work. A scientist can follow the feature conditions that change a molecule's priority and examine their development support and counterexamples. The tree is a statistical decision model, with its structure and priorities proposed by the LLM and selected through empirical evaluation. Its interpretation concerns the acquisition rule. Biological explanations suggested by its branches remain hypotheses for further investigation. This representation connects automated policy development with scientific judgment without requiring the practitioner to inspect every language-model exchange.

We also study how development gains can transfer to acquisition utility. Under sampling assumptions, a theoretical bound relates expected acquisition gain to development gain, dependence induced by adaptive policy selection, differences between candidate populations and the extent of changes to model-based selections. The result helps explain why a higher development score may not give a larger acquisition benefit.

Our experiments compare MLP ranking, random-tree search and LLM-guided tree development on HIV and ALDH1 screening benchmarks. Random search uses the same tree representation, available features, evaluator and candidate-evaluation budget. It provides a control for the complete LLM-guided search procedure. Campaigns begin with the same labeled collection across methods within each replicate. Final hit counts measure discovery after a specified number of assays. We assess early discovery using the area under the cumulative discovery curve (AUCDC), scaled so that a campaign in which every assay gives a hit scores one (Appendix B.3). This scaling expresses discovery performance on a common zero-to-one scale, while preserving differences in the number and timing of hits. Across HIV and ALDH1, both LLM workflows achieve higher mean acquisition hits and early discovery than MLP ranking and random-tree search. Higher development scores do not always lead to better acquisition outcomes. A supplementary exploratory held-out assessment finds significant hit gains on HIV after multiplicity correction. Section 4 reports the acquisition and held-out comparisons separately.

2 Related Work

Machine learning for molecular screening. Models of molecular activity can guide the selection of compounds for experimental testing. Stokes et al. (2020) used deep learning to prioritize antibacterial candidates and validated selected compounds experimentally. Active learning combines repeated candidate selection with model updating. van Tilborg & Grisoni (2024) examine how acquisition strategies affect hit discovery with limited labeled data. MolPAL applies pool-based active learning to virtual screening, evaluating the recovery of highly ranked docking candidates (Graff et al., 2021). Language models have also been incorporated into statistical optimization. LLAMBO uses contextual information through in-context learning to support initialization, surrogate modeling and candidate sampling in Bayesian optimization (Liu et al., 2024b). GOLLuM learns language-model representations through a Gaussian-process marginal-likelihood objective to guide experimental discovery under uncertainty (Ranković et al., 2026). Our prediction-reasoning fusion framework uses language-model reasoning to develop interpretable acquisition trees from model predictions and molecular context. Acquisition outcomes assess whether these policies improve the number of molecular hits discovered and how early they are found.

Language-guided policy search. Numerical evaluation can guide the search over rules and programs proposed by language models. FunSearch combines program generation with evaluation and evolutionary selection (Romera-Paredes et al., 2024). Evolution of Heuristics develops natural-language descriptions and their corresponding code within an evolutionary search (Liu et al., 2024a), while ReEvo uses reflective feedback to guide heuristic generation (Ye et al., 2024). These methods illustrate how numerical feedback can guide language-model proposals. In our setting, the search produces a compact acquisition tree using current model predictions and molecular summaries. Before each acquisition round, the predictive model is refitted and a new policy-development session begins. Repeated use of development feedback connects this search to adaptive data analysis and its study of selection bias and generalization (Dwork et al., 2015; Russo & Zou, 2016). Our theoretical analysis examines how adaptive selection, population differences and changes to model-based selections affect the transfer of development gains to acquisition utility. Random-tree search provides a control under the same representation and candidate-evaluation budget.

Interpretable trees and language-model reasoning. Recent work uses language models to construct or improve trees for tabular prediction. Talking Trees equips a reasoning model with tools for building, evaluating and revising decision trees (Yakushev et al., 2025). DeLTa uses LLM-generated rules to correct predictions from existing trees (Ye et al., 2025). Our trees assign acquisition priorities using predictions from a model that is retrained as assay outcomes accumulate. Their branches can be examined alongside the development examples and feedback supporting their selection. Evaluation concerns the discovery utility of this sequential procedure, including how generated rules transfer from development to acquisition. Interpretability supports scientific scrutiny of those rules, while biochemical explanations remain hypotheses.

3 Framework

3.1 Prediction-reasoning fusion

We consider *prediction-reasoning fusion* as a framework for combining outcome prediction with language-model reasoning to construct decision policies. A predictive model provides predictions, while a language model uses these predictions and available contextual information to propose rules for choosing actions. A policy evaluator checks the proposed rules, measures their utility on development data and returns feedback that informs subsequent proposals. Predictions thus inform the search for useful decisions, while empirical evaluation guides policy selection. Our aim is to improve decisions based on model predictions and contextual information while keeping the decision rules interpretable.

At decision step t, predictive-model training fits the predictive model $f(\,\cdot\,;\widehat{\theta}_t)$ to the available labeled observations L_t, where $\widehat{\theta}_t$ denotes the fitted parameters. Policy development uses separate development data D to construct, evaluate and select policies. These data are excluded from predictive-model training. A policy maps model predictions and contextual information to an action, whose

utility specifies the task. In molecular screening, the action is a batch of candidates and utility is the number of observed hits. The model-based policy ranks candidates by predicted activity. The language model proposes conditions under which contextual information may justify changing that selection. The framework separates policy construction on development data from measurement of its benefit through subsequent acquisition.

3.2 Policy development and sequential acquisition

Each development session alternates numerical queries with policy proposals. The language model receives model predictions and contextual summaries, then requests further comparisons relating candidate characteristics to development outcomes. Molecular inputs include ensemble disagreement, descriptors and similarity to molecules with known assay outcomes. The evaluator returns the requested summaries and evaluates proposed policies, reporting development utility, branch support and gains or losses relative to model-based selection. Subsequent turns use the queries, proposals and feedback from that session. This permits revision of a hypothesis without updating the predictive model or language-model parameters. Unassayed acquisition outcomes are unavailable to this process.

Our implementation uses decision trees whose branches compare features with thresholds and whose leaves assign ordinal priorities. Candidates are ordered by leaf priority, followed by model score and a specified candidate identifier to resolve ties. A tree can therefore change ordering between priority groups while preserving model-score ordering within them. Among valid proposals, the evaluator selects the tree with the greatest development utility if it improves on the current model-based policy. Ties favor fewer leaves, smaller depth and earlier proposal. If no tree improves development utility, acquisition uses the model-based policy. Each selected branch can be examined alongside its support and counterexamples. This makes the decision rule interpretable while leaving biochemical explanations subject to further investigation.

The selected policy is applied for one acquisition round. Before the next round, L_{t+1} contains the previous labeled observations and newly observed outcomes. The predictive model is refitted on L_{t+1}, and a new policy-development session begins. Earlier trees and dialogues are excluded. Acquired labels contribute through the updated predictive model and contextual information. The development set remains unchanged across rounds, while its predictions and neighbor features are updated. A development budget of K sequential query-proposal pairs allows up to K evaluated policies at each checkpoint. Each pair consists of one query turn and one proposal turn. Random-tree search evaluates K new candidates under the same representation and evaluation rules. Appendix B specifies the implementation. This procedure makes acquisition utility the criterion for assessing whether development gains transfer.

3.3 From development gains to acquisition utility

Adaptive feedback can improve the policies found while increasing their dependence on the development observations. Acquisition candidates may also differ from those used in development. We study these effects at one checkpoint, treating the predictive model and contextual information as constants obtained independently of the development data D. Let $Z = (X, Y)$ be a labeled candidate set, with X containing candidate features, model predictions and contextual information available for selection and $Y_i \in [0, 1]$ the outcome of candidate i. Each candidate set contains at least b candidates, and a policy π selects b using X. For additive utility $U(B, Y) = \sum_{i \in B} Y_i$, define its gain over the model-based policy π_0 by

$$g_\pi(Z) = \frac{U(\pi(X), Y) - U(\pi_0(X), Y)}{b}.$$

Binary outcomes give the difference in hit fractions. Suppose $D = (Z_1, \ldots, Z_n)$ consists of independent candidate sets from a development distribution P, and let Q be the distribution of an independent acquisition candidate set.

Only candidates selected by one policy but not the other contribute to their utility difference. Let $\rho(\pi) \in [0, 1]$ be a measurable bound satisfying $|\pi(X) \setminus \pi_0(X)|/b \leq \rho(\pi)$ throughout the supports of P and Q. We take $\rho(\pi_0) = 0$. For other policies, $\rho(\pi) = 1$ permits unrestricted replacement. Write $\delta = \mathrm{TV}(P, Q)$, with total variation defined as $\sup_A |P(A) - Q(A)|$. The following result applies the information-theoretic analysis of adaptive selection (Russo & Zou, 2016) to relative acquisition utility.

Proposition 1 (Transfer of an adaptively selected policy)**.** *Let n and b be positive integers. Let $D = (Z_1, \ldots, Z_n)$ consist of independent candidate sets with common distribution P, and let Q be the distribution of an independent acquisition set. Each set $Z = (X, Y)$ contains at least b candidates with outcomes $Y_i \in [0, 1]$. The predictive model, external context and baseline policy π_0 are specified independently of D and treated as constants. Consider a policy class specified independently of D, containing the baseline π_0. Every policy in this class selects b candidates using X. Suppose a measurable function ρ, specified independently of D with $\rho(\pi_0) = 0$, satisfies $0 \le \rho(\pi) \le 1$ and $|\pi(X) \setminus \pi_0(X)| \le b\rho(\pi)$ for every such policy and every candidate set in the supports of P and Q.*

For the gain per selected candidate $g_\pi(Z) = b^{-1}(\sum_{i\in\pi(X)} Y_i - \sum_{i\in\pi_0(X)} Y_i)$, assume joint measurability in (π, Z). Write $\widehat{\Delta}_D(\pi) = n^{-1}\sum_{j=1}^n g_\pi(Z_j)$ and $\Delta_Q(\pi) = \mathbb{E}_{Z\sim Q}[g_\pi(Z)]$. Let $\widehat{\pi}$ be a measurable policy selected from this class using D, allowing adaptive queries and randomization. Set $\delta = \sup_A |P(A) - Q(A)|$, where the supremum is over measurable sets, and suppose the mutual information $\mathcal{I} = I(\widehat{\pi}; D)$ is finite and computed using natural logarithms. Then

$$\mathbb{E}[\Delta_Q(\widehat{\pi})] \ge \mathbb{E}[\widehat{\Delta}_D(\widehat{\pi})] - \sqrt{\frac{2\mathcal{I}\,\mathbb{E}[\rho(\widehat{\pi})^2]}{n}} - 2\delta\,\mathbb{E}[\rho(\widehat{\pi})], \tag{1}$$

where expectations are over development sampling and policy construction.

The first subtracted term bounds optimism from using the same data to construct and evaluate a policy. Mutual information measures the dependence of the selected policy on those data. The second term bounds the change in expected gain between development and acquisition populations. Both terms depend on the extent of the correction. With the other quantities unchanged, a smaller replacement bound tightens the guarantee, although it can also limit attainable gains. The lower bound is positive when expected development gain exceeds the combined terms. The proof is provided in Appendix A.

The number of evaluated candidates does not determine their dependence on development data. For at most K candidates generated independently of D, selection with fallback to the model-based policy gives $\mathcal{I} \le \log(K+1)$. The additional choice is the model-based policy. Adaptive proposal generation can introduce further dependence before final selection. Appendix A derives this relationship and discusses feature-dependent candidate construction. Equal candidate-evaluation budgets therefore control one part of the comparison. They do not establish equal development optimism. More reasoning may improve the policies found, while its effect on transfer requires empirical evaluation.

Proposition 1 provides an explanation of possible development-acquisition gaps under sampling assumptions. The independent candidate sets counted by n are the units of the theoretical analysis. Our reused development folds and sequential campaign initializations do not supply those independent units, and later predictive models can depend on earlier development feedback through acquisition choices. We do not estimate $\mathcal{I}$ or δ, or use the bound as a numerical guarantee. These considerations motivate reporting acquisition outcomes separately from development scores, alongside the rules and feedback that produced each policy.

4 Experiments

We evaluate prediction-reasoning fusion through retrospective molecular screening. The datasets' activity labels serve as assay outcomes. We compare MLP ranking, random-tree search and LLM-guided tree development using the same MLP architecture and training procedure. The comparison with MLP ranking tests the benefit of adding a policy layer, while random search provides a control for guided proposal generation within the same tree representation. Development scores describe policy selection, and subsequent acquisition outcomes assess discovery utility.

The development set guides tree construction and selection, while the acquisition pool provides new assay outcomes that update the predictive model and inform the next round of tree development. A separate held-out test set is excluded from both processes and assesses each replicate's final deployed predictive model and policy without further fitting or policy development. Each campaign begins with 1,000 labeled molecules from the acquisition pool, and the remaining pool labels are concealed until selection. Known and remaining molecules within this pool can share scaffolds. Appendix B specifies the partitions and information available at each stage.

Each campaign comprises ten acquisition rounds, with 30 molecules selected per round. We use ten paired initializations per dataset, with methods sharing the initial labeled set within each replicate.

Before each round, each method fits its five-member MLP ensemble using the labels available to that method. Tree-based methods then begin a new policy-development session. At each checkpoint, the LLM conducts four sequential query-proposal pairs, and random search evaluates four sampled trees. Both tree-based methods use the same feature definitions, representation constraints and selection criterion. The current MLP ranking is used if no eligible tree improves development utility. GPT Astra uses medium reasoning, and Qwen3.8-27B follows the same eight-call procedure. Both receive model predictions, ensemble disagreement, structural-neighbor evidence and molecular descriptors. Dataset identities and target-specific biochemical instructions are excluded.

We report newly discovered hits after 300 assays and AUCDC, scaled as defined in Appendix B.3. The latter combines the number of hits with how early they are discovered. Development utility counts hits among 150 selections across five search folds at each checkpoint. We average these checkpoint scores within a campaign when summarizing development. Paired acquisition comparisons use whole campaigns as their statistical units. The two-sided sign-flip tests and four-test Holm adjustment are applied separately for each dataset and LLM workflow, as described in Appendix B.3.

Table 1 summarizes acquisition outcomes across both datasets. Figure 1 shows their variation across campaign initializations, while Figure 2 compares development and acquisition gains over MLP ranking. Table 2 reports the supplementary held-out test comparisons.

4.1 HIV SCREENING

The HIV benchmark contains 41,127 molecules, including 1,443 labeled active (3.51%). It originates from the NCI AIDS antiviral screen and is distributed through MoleculeNet and Therapeutics Data Commons (Wu et al., 2018; Huang et al., 2021). We retain the benchmark's binary activity labels. Each campaign draws its 1,000 initial labeled molecules from an acquisition pool of 28,816 molecules, leaving 27,816 candidates for acquisition. A separate development partition provides policy-search feedback. Appendix C describes preparation and the full partition allocation.

Qwen-generated policies discover a mean of 196.9 hits and Astra-generated policies 195.7, compared with 188.1 for MLP ranking and 190.3 for random-tree search (Table 1). The mean gains over MLP are 8.8 hits for Qwen and 7.6 for Astra. Mean AUCDC is also higher for both LLM workflows, at 0.7095 for Qwen and 0.6952 for Astra, compared with 0.6874 for both MLP and random trees. The top row of Figure 1 shows the variation across campaigns. Appendix C.3 reports the paired tests and intervals for both outcomes.

Interpreting a decision. A depth-two Astra policy from the second replicate illustrates how the generated rules can be interpreted. It uses similarity to known actives, with different similarity thresholds above and below a fraction of sp^3 carbon atoms of 0.5. Molecules receive a priority, and MLP scores order candidates with equal priority. A scientist can follow the branch behind a promotion and examine its development support. The example improves development hits from 18 to 34 and discovers the same 19 first-batch hits as MLP ranking. Appendix C.4 gives the full rule and its uneven foldwise support. Its conditions express an acquisition preference, without establishing a biochemical mechanism.

4.2 ALDH1 SCREENING

The ALDH1 benchmark concerns inhibitors of aldehyde dehydrogenase 1A1, an enzyme studied as a potential cancer-treatment target because of its association with cancer stem cells and tumor recurrence (Yasgar et al., 2017). This benchmark provides a larger candidate pool under the same acquisition budget. We use the full LIT-PCBA ALDH1 files (Tran-Nguyen et al., 2020), retain their activity labels and apply structural deduplication. The resulting dataset contains 142,858 molecules, including 7,123 actives (4.99%). Each campaign starts with 1,000 labeled molecules from an acquisition pool of 100,002 molecules, leaving 99,002 candidates for acquisition. Appendix D documents the source, preparation and partitions.

Qwen discovers a mean of 115.7 hits and Astra 113.9, compared with 108.1 for MLP and 107.7 for random trees (Figure 1, bottom row). Their mean gains over MLP are 7.6 and 5.8 hits, respectively. Mean AUCDC is 0.3844 for Qwen and 0.3813 for Astra, compared with 0.3701 for MLP and 0.3611

Table 1: Acquisition outcomes after 300 assays, averaged over ten paired campaign initializations per dataset. Figures show campaign variation. Paired uncertainty intervals and Holm-adjusted tests appear in Appendices C.3 and D.3.

Dataset	Acquisition method	Mean hits	Mean AUCDC
HIV	MLP	188.1	0.6874
	MLP + random tree	190.3	0.6874
	MLP + Qwen tree	196.9	0.7095
	MLP + Astra tree	195.7	0.6952
ALDH1	MLP	108.1	0.3701
	MLP + random tree	107.7	0.3611
	MLP + Qwen tree	115.7	0.3844
	MLP + Astra tree	113.9	0.3813

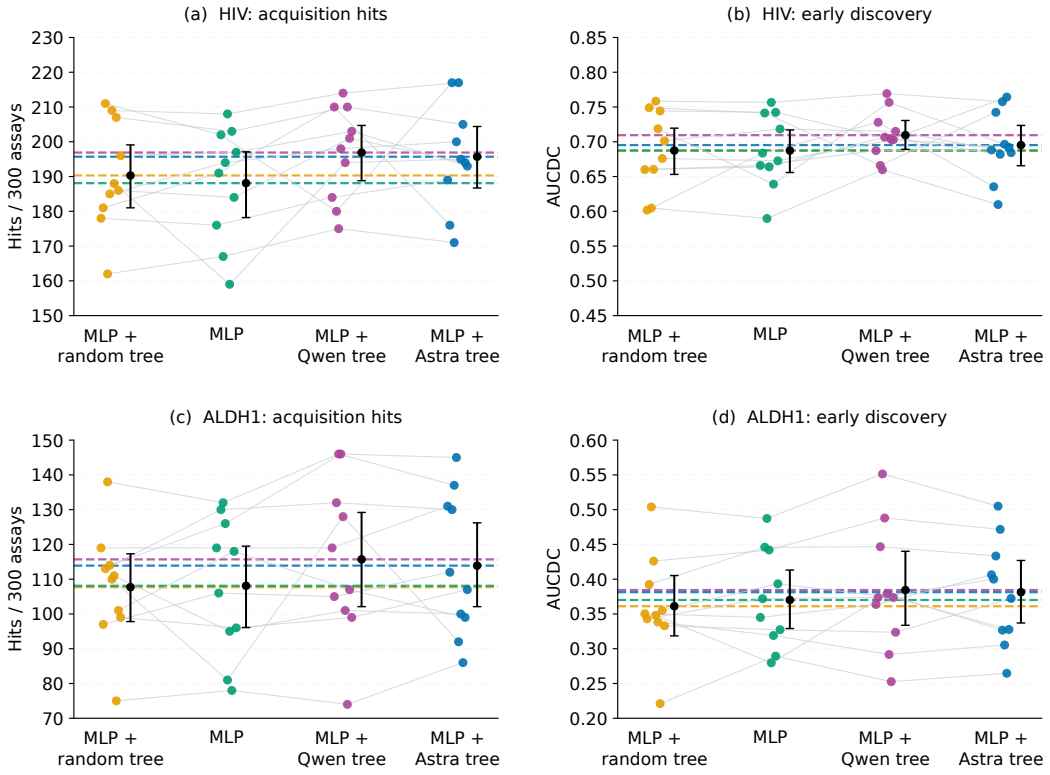

Figure 1: Acquisition outcomes for HIV (top) and ALDH1 (bottom). Left panels show hits after 300 assays, and right panels show AUCDC across ten sequential acquisition rounds. Corresponding panels use equal vertical spans with different limits across datasets. Each point represents one of ten paired campaign initializations. Thin solid gray lines connect the same initialization across methods. Black dots and bars show means and descriptive 95% bootstrap intervals. Colored dashed lines extend across each panel at the means indicated by the black dots. Intervals describe variability across initializations of the same dataset.

for random trees. Both LLM workflows improve mean discovery utility under this acquisition budget. Appendix D.3 gives the full paired comparisons.

4.3 Findings across datasets

Both LLM workflows improve mean acquisition hits and AUCDC over MLP ranking and random-tree search in both datasets. None of these acquisition comparisons reaches the 5% significance level

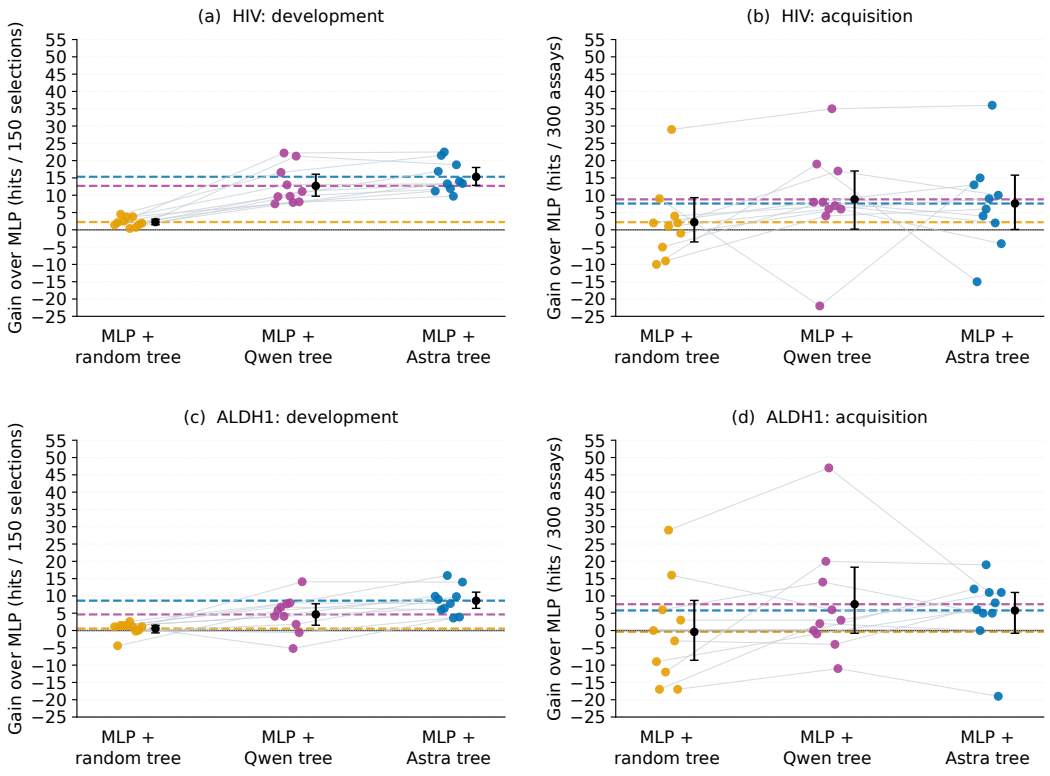

Figure 2: Paired gains over the MLP campaign during development (left) and acquisition (right), for HIV (top) and ALDH1 (bottom). Development scores average ten checkpoints, each selecting 150 molecules across five search folds. Acquisition counts cover 300 assays per campaign. The zero line denotes equal outcomes, and five hits have the same vertical length in all panels. Points, connections, means and descriptive intervals follow Figure 1, with intervals computed from paired differences. Development gains compare each tree workflow with the MLP workflow at corresponding checkpoints. Each workflow uses its own acquired labels and fitted predictive models. Development scores guide policy selection. The stages use different populations and budgets, so the difference between their gains is not an estimate of overfitting.

within its four-test Holm family. The positive mean gains and campaign variation support further evaluation, while the current samples leave uncertainty about the size and consistency of the benefit. Random-tree search changes mean discovery little relative to MLP, but this control evaluates four sampled candidates per checkpoint and does not establish the value of more extensive statistical search.

The ordering of the two LLM workflows reverses between development and acquisition on both datasets (Figure 2). Astra has higher mean development scores than Qwen, with 39.5 versus 36.9 hits on HIV and 33.5 versus 29.5 on ALDH1. Qwen has slightly higher mean acquisition hits and AUCDC on both datasets. Higher development scores therefore do not order the two workflows by subsequent discovery. This pattern seems consistent with overfitting during policy development, although differences between development and acquisition populations may also contribute. Table 3 gives all development means. This comparison concerns the configured workflows and does not isolate reasoning ability or establish an advantage for either workflow through a statistical test.

The possible effects of adaptive development reuse and population differences are consistent with the considerations in Proposition 1, although these experiments do not identify their contributions. Later checkpoints use each method's acquired labels and predictive model, so the figure compares evolving campaigns. The pattern across both datasets motivates testing how reasoning affects policy transfer under controlled settings.

Table 2: Held-out test comparisons across ten paired campaign initializations. Mean gains count additional hits among 300 selections made by each final deployed predictive model and policy. Tree methods operate above their respective MLPs. Two-sided sign-flip tests use all $2^{10} = 1,024$ sign assignments for the ten paired replicates, with Holm correction across the two comparators separately for each LLM and dataset. All four HIV comparisons meet the 5% statistical significance threshold; none does for ALDH1.

Dataset	LLM tree	Comparator	Mean hit gain	Holm p
HIV	Qwen	MLP	$+12.8$	0.00781
	Qwen	Random tree	$+13.0$	0.00977
	Astra	MLP	$+18.0$	0.00391
	Astra	Random tree	$+18.2$	0.00391
ALDH1	Qwen	MLP	$+6.0$	0.25000
	Qwen	Random tree	$+7.6$	0.25000
	Astra	MLP	$+4.1$	0.36719
	Astra	Random tree	$+5.7$	0.36719

These sequential results raise a further question about transfer beyond the acquisition pool. We apply each replicate's final deployed predictive model and policy to the held-out test set, selecting 300 molecules without further fitting or policy development. On HIV, Qwen and Astra give mean gains of 12.8 and 18.0 hits over MLP ranking, and 13.0 and 18.2 over random trees. All four comparisons meet the 5% significance level after Holm correction within each LLM's two-comparator family (Table 2). These exploratory findings support the promise of prediction-reasoning fusion for transfer to held-out molecules. On ALDH1, the corresponding gains are 6.0 and 4.1 hits over MLP and 7.6 and 5.7 over random trees, without statistical significance. This additional assessment was designed after inspecting acquisition results and remains exploratory. It evaluates transfer of the completed predictive model and policy to a separate molecular pool. Its gains concern a different selection task from sequential acquisition. Appendix B.5 gives the protocol and results.

5 Discussion

Higher mean acquisition utility and significant gains in the exploratory HIV held-out assessment support further study of prediction-reasoning fusion. Variation across initializations and evaluation settings leaves uncertainty about when the approach will be useful. Under what conditions does language-model reasoning identify useful departures from model-based ranking that transfer to new candidates?

The information available to the language model constrains the hypotheses it can support. Molecular activity labels describe assay outcomes without identifying their biochemical mechanisms. Our structural descriptors and local activity summaries capture only part of the context behind those outcomes. Interpretable branches allow scientists to examine the conditions, supporting examples and counterexamples behind a selection, while mechanistic conclusions require further evidence. Incorporating assay conditions or mechanistic measurements could help formulate more relevant hypotheses. Their value would then be assessed by evaluating the resulting policies on outcomes unavailable during policy construction.

Proposition 1 relates transfer to development dependence, population differences and the extent of changes to model-based selections. These considerations suggest factors for controlled investigation, without identifying the cause of the observed differences. Varying the reasoning budget within one language model could test whether additional search improves acquisition utility or increases adaptation to development examples. Such a comparison should retain the same evidence, policy representation and candidate-evaluation budget. Astra and Qwen differ in model characteristics, inference settings and response reliability, so their comparison leaves open whether a smaller reasoning budget acts as regularization. Computational effort should therefore be assessed against acquisition benefit.

AI USE STATEMENT

Generative AI tools assisted with experimental design, data preparation, code development, and checking and polishing the manuscript. The authors are responsible for the final content.

ETHICS STATEMENT

This study uses public molecular screening datasets and involves no new experiments with human participants or animals. The generated policies prioritize molecules for testing. Their rules do not establish biochemical mechanisms, therapeutic efficacy or safety. Prospective use would require experimental validation and scientific oversight.

REPRODUCIBILITY STATEMENT

Section 3 describes the framework, and Appendix A provides the theoretical proof. Appendix B specifies data partitioning, predictive-model training, policy development, language-model configurations and statistical analysis. Appendices C and D describe dataset sources, molecular preparation and supplementary results. Language-model outputs may vary across requests and changes to the serving infrastructure. The code will be made available upon publication.

REFERENCES

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

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

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

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Fei Liu, Xialiang Tong, Mingxuan Yuan, Xi Lin, Fu Luo, Zhenkun Wang, Zhichao Lu, and Qingfu Zhang. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32201–32223. PMLR, 2024a.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In *International Conference on Learning Representations*, pp. 31252–31284, 2024b.

Bojana Ranković, Ryan-Rhys Griffiths, and Philippe Schwaller. Large language models as uncertainty-calibrated optimizers for experimental discovery. *Nature Machine Intelligence*, 8(9):1466–1477, 2026.

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

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 1232–1240. PMLR, 2016.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Ava P. Soleimany, Alexander Amini, Samuel Goldman, Daniela Rus, Sangeeta N. Bhatia, and Connor W. Coley. Evidential deep learning for guided molecular property prediction and discovery. *ACS Central Science*, 7(8):1356–1367, 2021.

Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackermann, Victoria M. Tran, Anush Chiappino-Pepe, Ahmed H. Badran, Ian W. Andrews, Emma J. Chory, George M. Church, Eric D. Brown, Tommi S. Jaakkola, Regina Barzilay, and James J. Collins. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702.e13, 2020.

Tingzhong Tian, Shuya Li, Ziting Zhang, Lin Chen, Ziheng Zou, Dan Zhao, and Jianyang Zeng. Benchmarking compound activity prediction for real-world drug discovery applications. *Communications Chemistry*, 7(1):127, 2024.

Viet-Khoa Tran-Nguyen, Célien Jacquemard, and Didier Rognan. LIT-PCBA: An unbiased data set for machine learning and virtual screening. *Journal of Chemical Information and Modeling*, 60(9): 4263–4273, 2020.

Derek van Tilborg and Francesca Grisoni. Traversing chemical space with active deep learning for low-data drug discovery. *Nature Computational Science*, 4(10):786–796, 2024.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018.

George Yakushev, Alina Shutova, Ivan Rubachev, Natalia Bereberdina, Renat Sergazinov, and Artem Babenko. Talking trees: Reasoning-assisted induction of decision trees for tabular data, 2025. arXiv:2509.21465v3, revised May 2026.

Adam Yasgar, Steven A. Titus, Yuhong Wang, Carina Danchik, Shyh-Ming Yang, Vasilis Vasiliou, Ajit Jadhav, David J. Maloney, Anton Simeonov, and Natalia J. Martinez. A high-content assay enables the automated screening and identification of small molecules with specific ALDH1A1-inhibitory activity. *PLOS ONE*, 12(1):e0170937, 2017.

Hangting Ye, Jinmeng Li, He Zhao, Dandan Guo, and Yi Chang. LLM meeting decision trees on tabular data. In *Advances in Neural Information Processing Systems*, volume 38, pp. 118095–118131. Curran Associates, Inc., 2025.

Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. ReEvo: Large language models as hyper-heuristics with reflective evolution. In *Advances in Neural Information Processing Systems*, volume 37, pp. 43571–43608. Curran Associates, Inc., 2024.

A DEVELOPMENT-TO-ACQUISITION TRANSFER

A.1 PROOF OF PROPOSITION 1

The predictive model, external context and baseline policy π_0 are specified independently of the development sample and treated as constants. All policies considered below select b candidates using the observable part X of a candidate set $Z = (X, Y)$. Candidate outcomes lie in $[0, 1]$. Dependence between candidates within a set is permitted. The independence assumption concerns the n sets in D. Summaries and queries computed from D are part of policy construction and may depend on those data. We use natural logarithms and the total-variation convention $\mathrm{TV}(P, Q) = \sup_A |P(A) - Q(A)|$.

Proof. For any policy π under consideration, the sets $\pi(X) \setminus \pi_0(X)$ and $\pi_0(X) \setminus \pi(X)$ have the same cardinality, at most $b\rho(\pi)$. Outcomes of candidates in $\pi(X) \cap \pi_0(X)$ cancel. Since outcomes lie in $[0, 1]$,

$$-\rho(\pi) \le g_\pi(Z) \le \rho(\pi). \tag{2}$$

Write $\Delta_P(\pi) = \mathbb{E}_{Z\sim P}[g_\pi(Z)]$. Hoeffding's lemma applied to (2), followed by independence across the development candidate sets, gives

$$\mathbb{E}_D \exp\left\{\lambda\big[\widehat{\Delta}_D(\pi) - \Delta_P(\pi)\big]\right\} \le \exp\left\{\frac{\lambda^2\rho(\pi)^2}{2n}\right\} \qquad (\lambda \in \mathbb{R}). \tag{3}$$

This inequality holds for each policy considered separately. To account for adaptive selection, let ν be the joint law of $(D, \widehat{\pi})$, and let μ be the product of their marginal laws. Thus D and $\widehat{\pi}$ are independent under μ, and $\mathrm{KL}(\nu\|\mu) = I(\widehat{\pi}; D) = \mathcal{I}$. For $\lambda > 0$, set

$$F_\lambda(D, \pi) = \lambda\big[\widehat{\Delta}_D(\pi) - \Delta_P(\pi)\big] - \frac{\lambda^2\rho(\pi)^2}{2n}.$$

Integrating (3) over the policy marginal gives $\mathbb{E}_\mu[e^{F_\lambda}] \le 1$. The variational inequality for relative entropy states that

$$\mathbb{E}_\nu[F_\lambda] \le \mathrm{KL}(\nu\|\mu) + \log \mathbb{E}_\mu[e^{F_\lambda}].$$

For completeness, this follows by introducing the probability law with density $e^{F_\lambda}/\mathbb{E}_\mu[e^{F_\lambda}]$ relative to μ, and applying nonnegativity of the KL divergence from ν to that law. The function F_λ is bounded, and finite $\mathcal{I}$ ensures the required absolute continuity. Consequently,

$$\mathbb{E}\big[\widehat{\Delta}_D(\widehat{\pi}) - \Delta_P(\widehat{\pi})\big] \le \frac{\mathcal{I}}{\lambda} + \frac{\lambda}{2n}\mathbb{E}[\rho(\widehat{\pi})^2]. \tag{4}$$

When both $\mathcal{I}$ and $\mathbb{E}[\rho(\widehat{\pi})^2]$ are positive, minimize the right-hand side at $\lambda = \sqrt{2n\mathcal{I}/\mathbb{E}[\rho(\widehat{\pi})^2]}$. This gives

$$\mathbb{E}\big[\widehat{\Delta}_D(\widehat{\pi}) - \Delta_P(\widehat{\pi})\big] \le \sqrt{\frac{2\mathcal{I}\,\mathbb{E}[\rho(\widehat{\pi})^2]}{n}}. \tag{5}$$

If $\mathcal{I} = 0$, take λ to zero in (4). If $\mathbb{E}[\rho(\widehat{\pi})^2] = 0$, then $\rho(\widehat{\pi}) = 0$ with probability one, so both $\widehat{\Delta}_D(\widehat{\pi})$ and $\Delta_P(\widehat{\pi})$ vanish and (5) also holds.

It remains to account for the acquisition distribution. For any policy π with $\rho(\pi) > 0$, the function $(g_\pi + \rho(\pi))/(2\rho(\pi))$ takes values in $[0, 1]$. The defining property of total variation therefore gives

$$|\Delta_Q(\pi) - \Delta_P(\pi)| \le 2\rho(\pi)\delta. \tag{6}$$

The same inequality holds when $\rho(\pi) = 0$. Averaging (6) over the selected policy and combining it with (5) proves (1). □

A.2 CANDIDATE EVALUATION BUDGETS

Let K denote the proposal budget. Write $C = (\pi_0, \pi_1, \ldots, \pi_K)$ for the candidate list and $J \in \{0, \ldots, K\}$ for the selected index, with $\widehat{\pi} = \pi_J$. Lists with fewer than K valid proposals can be completed by repeating π_0. The baseline thus contributes one additional selection option. If C is independent of D, data processing and the chain rule give

$$\begin{aligned} I(\widehat{\pi}; D) &\le I(C, J; D) \\ &= I(C; D) + I(J; D \mid C) \\ &\le H(J \mid C) \le \log(K + 1), \end{aligned} \tag{7}$$

where H denotes Shannon entropy computed using natural logarithms. If candidate construction uses development observations, the corresponding bound is

$$I(\widehat{\pi}; D) \leq I(C; D) + \log(K + 1).$$

The candidate list can therefore carry dependence on development data in addition to the choice of its final index. This observation applies to any data-dependent generator. It does not assert that adaptive construction increases actual bias or reduces acquisition utility.

Independence from development outcomes is weaker than independence from the full development sample. Our random generator uses feature-derived thresholds and branch-support constraints. Consequently, (7) is an illustrative bound for candidates constructed independently of the development sample, rather than a numerical bound established for that comparator. Conditioning on observed features could isolate dependence on labels, but would require a corresponding conditional sampling analysis. The comparison in the experiments controls representation, available features and candidate evaluations, with the LLM additionally using outcome-dependent feedback to guide proposals.

A.3 Scope of the result

The proposition concerns expected gain on an independent acquisition candidate set at one checkpoint. The baseline π_0 uses the same predictive model as the selected policy at the checkpoint under analysis. Experimental campaign comparisons also include differences in acquired training data and subsequent model fits. Its unit of independence is a complete labeled candidate set, rather than a selected molecule, a proposal or an acquisition round. It allows dependence among candidates within a set. The development folds in our implementation are reused for adaptive feedback, and the ten campaign initializations reuse the underlying benchmark. We do not identify either count with n in the proposition.

The bound $\rho(\pi)$ applies across candidate sets in the supports of both populations. A small replacement fraction observed in one batch does not establish such a bound. Our priority trees can reorder the full pool, so the unrestricted choice $\rho = 1$ is available for nonbaseline policies. The proposition introduces no replacement cap or additional acceptance rule into the implemented algorithm. Similarly, $\mathcal{I}$ describes statistical dependence rather than model size, token use or reasoning effort. Neither it nor δ is estimated in the reported experiments, and the resulting bound may be too loose to establish a positive gain.

The argument requires the predictive model and external contextual information to be obtained independently of the development sample under analysis. Later campaign predictive models use labels acquired through earlier development-guided choices. Restarting the dialogue removes earlier trees and messages from the prompt, but does not remove this statistical dependence. A guarantee for the full campaign would require an analysis of that sequential dependence and the changing acquisition pool. Proposition 1 motivates separate development and acquisition evaluation without claiming such a campaign guarantee.

B Shared experimental protocol

B.1 Preparation and information access

Molecular preparation uses RDKit 2023.09.6. Structures are parsed and sanitized, and canonical isomeric SMILES identify duplicates and conflicting labels. Dataset-specific handling of conflicting labels and duplicates is given in Appendices C and D. Salts and disconnected fragments are retained without modification. We do not select a parent fragment, neutralize molecules or standardize tautomers. Binary Morgan fingerprints use radius two, 2,048 bits, chirality and bond types (Rogers & Hahn, 2010).

Molecules sharing a nonempty Bemis-Murcko scaffold are grouped, and groups are merged transitively through identical fingerprints. Molecules with empty scaffolds remain separate unless fingerprint identity connects them to another group. A seeded, 20-fold stratified group allocation is combined into 14 acquisition-pool folds, three development folds and three held-out test folds, balancing molecule, active and scaffold-empty counts. Each nonempty scaffold and each fingerprint vector

belongs to a single partition. Other forms of structural similarity can cross partitions, particularly among acyclic molecules.

The acquisition-pool partition contains both the initial labeled set and the molecules available for subsequent selection. Each campaign begins with 1,000 molecules selected from this partition without consulting their activity labels and with identical-fingerprint groups kept together. The remaining molecules form the unassayed pool. Methods share the initial labeled set and configuration within a replicate. Once acquisition choices differ, subsequent training data and pools can differ. Neighbor features use initial and subsequently acquired labels. Development and held-out test outcomes are excluded from neighbor labels.

Within a replicate, development structural groups are assigned by deterministic, label-blind hashes to a search portion and a withheld audit portion. The hash includes the dataset, campaign seed, structural-group identifier and a prespecified split seed. Each group has a nominal 20% probability of audit assignment, so the withheld molecule fraction depends on group sizes. A separate hash assigns search groups to five folds. Membership and fold assignments remain unchanged throughout the campaign and are shared across methods. Predictions and neighbor information are refreshed at each checkpoint. Audit outcomes are excluded from policy generation and selection. We make no audit-performance claim in this analysis.

Acquisition labels become available after a batch is selected. The campaign takes place within the acquisition-pool partition, where known and remaining molecules can share scaffolds. This evaluates sequential discovery within the acquisition pool. The held-out test set is excluded from these campaigns and is evaluated separately after their completion (Appendix B.5). In a prospective screening workflow, each cycle would wait for the previous batch's assay outcomes.

B.2 Predictive model, evidence and policy search

Each MLP takes 2,048 fingerprint bits and has hidden layers of 128 and 64 units, ReLU activations and dropout 0.2. Training uses unweighted binary cross-entropy, Adam with learning rate 10^{-3} and weight decay 10^{-4}, batches of 128 and 25 epochs. The five ensemble members use different random seeds, with the same five seeds used across methods and campaigns. The ensemble mean provides the activity score. The population standard deviation across members measures disagreement on the probability scale. It is not a calibrated error probability. All methods retrain this ensemble from initialization before each acquisition round, using the 1,000 initial labels plus the outcomes acquired in previous rounds. There are no warm-started MLP weights.

The policy receives the raw ensemble mean, ensemble standard deviation, maximum Tanimoto similarity to known actives and inactives, the difference between these similarities, the active fraction among the five nearest known molecules, and a scaffold-empty indicator. Twelve further descriptors are molecular weight, logP, topological polar surface area, hydrogen-bond donor and acceptor counts, heavy-atom count, rotatable bonds, formal charge, aromatic rings, fraction of sp^3 carbon atoms, amide count and carboxyl count. These descriptors and neighbor summaries are inputs to the policy layer. The MLP uses only fingerprints. Both tree-search methods receive the same feature definitions and threshold vocabulary. No probability recalibration is applied.

At each checkpoint, a candidate tree selects 30 molecules from each of the five search folds. Its objective is the total number of actives among these 150 selections. The folds provide repeated adaptive feedback. They are dependent parts of policy development. The MLP is not retrained during this checkpoint's search. Trees have depth at most three and integer leaf priorities in $\{0, \ldots, 7\}$. Every leaf must contain at least 50 search molecules overall and at least five in each fold. Molecules are ordered by decreasing tree priority, then decreasing raw MLP probability, then a specified candidate identifier. A priority change can reorder the full candidate pool. There is no top-rank cap, and a foldwise loss does not veto a positive aggregate gain.

LLM development alternates four evidence queries with four tree proposals. Queries request numerical feature contrasts within model-score strata and can condition on another feature. Evaluation returns the tree's hit count, gains or losses within folds, leaf support, and hits added or displaced relative to the current MLP. Subsequent turns receive all actions and evaluator feedback from the same checkpoint. The LLM can revise a tree or propose a different hypothesis. JSON schemas specify the

permitted query or tree response, and the evaluator checks the candidate. Invalid responses consume their turn without an additional proposal allowance.

A new dialogue starts after every predictive-model retraining. Earlier trees, dialogues and acquisition summaries are excluded from the new dialogue. Acquired labels still affect the current predictive model and neighbor evidence. Among valid candidates that improve aggregate development hits by at least one, selection prefers the most hits, then fewer leaves, smaller depth and the earlier proposal. If no candidate improves on the current MLP, the current MLP ranking is used. A selected tree is applied unchanged to that round's 30 acquisitions.

The Astra backend uses the served model identifier `gpt-6-astra`, medium reasoning, and a maximum of 32,768 generated tokens per call. The Qwen backend uses `qwen3.8-27b-6000pro` through an OpenAI-compatible endpoint, temperature 0.6, the same total ceiling, and an 8,192-token reasoning cap. These settings are practical allocations, not a claim of equal reasoning effort. A completed campaign schedules up to 80 provider calls, with eight at each of ten checkpoints. Conversations, evaluator responses, selected policies and token usage are saved. Unresolved transport failures pause execution. Requests marked as abandoned consume their slot when the run resumes.

Provider messages omit the dataset and target names, molecular identifiers, raw structures, and unavailable assay outcomes. The LLM receives numerical summaries and feature definitions, with no external lookup tools or target-specific biochemical library. It is instructed not to infer the target and to treat biochemical explanations as hypotheses that require support in the development evidence. Omitting identifiers limits target disclosure. Relevant prior chemical knowledge may still be present in a pretrained model.

The random comparator evaluates four new candidates at each checkpoint under the same constraints and selection rule. It samples target depth from one to three, can terminate an intermediate branch with probability 0.25, and samples feasible features, thresholds and priorities without outcome-dependent proposal generation. The evaluator uses development labels to select among these candidates. The comparison controls for access to features, representation and candidate-evaluation budget. Total computation and evidence processing can differ. It also does not compare the LLM with every possible tree optimizer.

B.3 Outcomes and statistical analysis

Consider a campaign with R acquisition rounds and b assays per round. Let h_r be the number of new active molecules in round r and $C_r = \sum_{j=1}^{r} h_j$, with $C_0 = 0$. Initial known actives are excluded. Final hits are C_R. We define AUCDC as the trapezoidal area under cumulative hits against assays, scaled by the area of a campaign in which every assay gives a hit:

$$\mathrm{AUCDC} = \frac{\sum_{r=1}^{R}(C_{r-1} + C_r)}{bR^2} = \frac{\sum_{r=1}^{R}\big[2(R-r)+1\big]h_r}{bR^2}.$$

Our experiments use $R = 10$ and $b = 30$. The score lies between zero and one. Both final discovery and timing affect it. Ordering within an assay batch is not measured.

The reported cohort contains ten replicates, using the first ten campaign seeds from the saved seed pool. Comparisons use a common completed cohort. The reporting target was reduced from twenty to ten for time and cost reasons after exploratory runs began. Repeated inspection and previous methodological development on these benchmarks make the results exploratory. The multiplicity adjustment below does not account for that broader research process.

Acquisition tests use whole campaign replicates as their units, not individual rounds. A two-sided sign-flip test compares the absolute mean paired difference with 99,999 random sign assignments, with Monte Carlo correction $(k+1)/(99{,}999+1)$. The count k includes sampled sign assignments whose absolute mean difference is at least as large as the observed value. Under the null reference distribution, the joint distribution of the paired differences is unchanged by independent sign reversals. For each dataset and LLM workflow, the four comparisons are LLM trees minus MLP and LLM trees minus random trees, each for hits and AUCDC. Holm adjustment covers this family jointly at $\alpha = 0.05$. Separate families do not provide omnibus error control across LLM workflows or datasets.

Descriptive percentile-bootstrap intervals use 20,000 resamples of whole replicates, preserving pairing. Figure 1 shows intervals for method means, and Figure 2 shows intervals for mean paired

gains. Tables in the dataset appendices give paired differences and their intervals. These intervals are not multiplicity-adjusted and are not inversions of the sign-flip test. Consequently, excluding zero in a bootstrap interval is not interchangeable with significance under the reported test, particularly with ten replicates. Inference concerns variation across campaign initializations for the chosen dataset, its development, acquisition and held-out test partitions, and the MLP architecture and training procedure. The campaigns are not independent biological experiments.

The MLP baseline discovers hits in 62.7% of its 300 acquisitions on HIV and 36.0% on ALDH1, substantially above the respective dataset activity fractions of 3.51% and 4.99%. This establishes a productive reference within the current screening protocol. Published benchmark scores use other partitions, metrics and acquisition budgets, so these hit counts do not establish comparative predictive performance against published models.

B.4 Development summaries

Table 3 summarizes the scores used during policy search. Each value averages ten checkpoint scores within each campaign, followed by an average across ten campaigns. These optimized measurements are descriptive. They do not provide independent evidence of generalization.

B.5 Held-out test assessment

The held-out test set contains 6,155 HIV molecules and 21,428 ALDH1 molecules (Tables 4 and 8). Its labels are excluded from predictive-model training, neighbor evidence, policy development and acquisition. We evaluate the final deployed predictive model and policy from every one of the ten campaign initializations, with no selection of a preferred replicate. This supplementary transfer assessment was designed after inspecting acquisition results.

The predictive model used in round ten was trained on 1,270 labels: the initial 1,000 molecules and the 270 acquired in rounds one through nine. We reconstruct that ensemble using its saved member seeds, training settings and software version. Its training history and predictions on the development and remaining acquisition molecules must reproduce the saved checkpoint without numerical differences before proceeding. The round-ten tree is then applied unchanged, including the MLP fallback when that was the deployed policy. Neighbor summaries use only the 1,270 labels available at that checkpoint. Round-ten acquisition labels are not added to this predictive model.

Each predictive model and its policy are used to rank the full test set once. We save the first 300 selections before reading their labels and count their hits. There is no further model fitting, LLM request, tree selection or update after observing these outcomes. This evaluates the complete resulting workflow, including its acquired training samples, rather than isolating a tree's effect on a common predictive model. Mean test hits for MLP, random trees, Qwen and Astra are 57.2, 57.0, 70.0 and 75.2 on HIV, and 44.2, 42.6, 50.2 and 48.3 on ALDH1.

Table 2 reports paired hit gains. The two-sided sign-flip tests use the same statistic and null assumptions as the acquisition analysis, with all $2^{10} = 1{,}024$ sign assignments evaluated instead of Monte Carlo sampling. These assignments calculate one p-value per comparison. Holm correction covers two comparisons per LLM and dataset, against MLP and random trees, at $\alpha = 0.05$. This family structure follows the acquisition analysis, with only hits evaluated in the one-off test. The families were aligned after the initial test statistics had been inspected. All four HIV comparisons meet the corrected threshold; none does for ALDH1. The ten evaluations share one test partition, so inference concerns campaign-initialization variation conditional on that partition, rather than variation across independently sampled test sets. These test results do not alter the acquisition comparisons.

C HIV data and supplementary results

C.1 Source and molecular preparation

We load the HIV high-throughput-screening task through Therapeutics Data Commons using `HTS(name='HIV')` (Huang et al., 2021), retaining the MoleculeNet benchmark's binary labels (Wu et al., 2018). The downloaded table contains 41,127 rows, with 1,443 active and 39,684 inactive. Quality checks find no missing structures or labels, duplicate identifiers, duplicate input SMILES,

Table 3: Mean development hits among 150 selections per checkpoint. Tree methods operate above the MLP. Each method follows its own acquisition trajectory and subsequently fitted predictive models.

Dataset	MLP	Random tree	Qwen tree	Astra tree
HIV	24.2	26.4	36.9	39.5
ALDH1	24.9	25.4	29.5	33.5

or invalid labels. All structures pass RDKit parsing and sanitization. Canonical isomeric SMILES change the textual representation of nine rows but introduce no duplicate structures or conflicting labels. All rows are retained.

There are 3,087 rows (7.51%) containing multiple fragments, which are retained without modification. Preparation follows Appendix B. There are 40,764 distinct fingerprints. The 281 repeated-fingerprint groups cover 644 molecules. No fingerprint is all zero. Identical hashed fingerprints are treated as potential representation duplicates, rather than assumed to imply chemical identity.

C.2 Partitions

The grouped allocation in Appendix B produces the partitions in Table 4. There are 1,582 molecules with empty Murcko scaffolds. No nonempty scaffold or identical fingerprint crosses the outer partitions. Each campaign begins with 1,000 labeled molecules from the acquisition-pool partition, leaving 27,816 unassayed candidates. The development search portion contains between 4,733 and 5,116 molecules across the ten campaign initializations. The held-out test set is excluded from sequential discovery and supports the separate transfer assessment in Appendix B.5.

C.3 Acquisition comparisons

Tables 5 and 6 report the Astra and Qwen comparisons using the same paired procedure. Each LLM workflow has a separate four-test Holm family, as defined in Appendix B.3. None of these acquisition comparisons reaches the 5% significance level after Holm correction. The intervals describe paired campaign variability and are not adjusted for multiple comparisons.

Qwen exceeds MLP ranking in nine of ten replicates and Astra in eight. Qwen's paired differences range from 22 fewer to 35 more hits, while Astra's range from 15 fewer to 36 more. The descriptive 95% intervals for their mean hit gains over MLP are $[0.2, 17.0]$ and $[0.1, 15.8]$, respectively. These observed gains motivate the policy layer, with uncertainty about their consistency across initializations.

C.4 An interpretable Astra policy

We illustrate a compact depth-two tree selected at the fourth proposal (dialogue call eight) before the first acquisition round of replicate two. The example combines a chemical descriptor with local observed activity evidence. It was chosen for its readable conditional structure, not as evidence of acquisition superiority. Let c denote the fraction of carbon atoms with sp^3 hybridization and s the maximum fingerprint Tanimoto similarity to an initially known active. Table 7 gives the complete policy. Similarity thresholds are displayed to four decimal places; execution uses their stored precision.

The rule permits a lower similarity threshold in the higher-sp^3 group and gives that intersection first priority. It expresses a conditional acquisition preference rather than a claim that high sp^3 fraction is universally favorable. Carbon hybridization is chemically interpretable, but this descriptor does not specify a three-dimensional conformation, target interaction or mechanism of antiviral activity. Likewise, fingerprint proximity to an active is empirical evidence, not proof of shared activity.

The selected tree scores 34 development hits against 18 for MLP ranking among 150 selections. Its foldwise changes are $0, 2, 11, 5, -2$. Eleven of the sixteen additional hits come from one fold. The highest-priority leaf contains 90 search molecules, including 24 actives. Of these, 88 molecules are selected during development and account for 23 of the 34 hits. These figures expose both the basis of

Table 4: HIV data partitions shared across campaign replicates. Counts refer to molecules. The acquisition pool includes the initial 1,000 labeled molecules.

Partition	Total	Active	Active (%)
Development	6,156	216	3.51
Acquisition pool	28,816	1,011	3.51
Held-out test	6,155	216	3.51

Table 5: HIV paired acquisition comparisons for Astra trees across ten campaign initializations. Gains are LLM-tree outcomes minus the comparator. Holm adjustment covers these four tests jointly. No test rejects at the 5% level. Bootstrap intervals are descriptive and unadjusted.

Comparator	Outcome	Mean gain	95% interval	Holm p
MLP	Hits	$+7.6$	$[0.1, 15.8]$	0.30813
Random trees	Hits	$+5.4$	$[1.5, 9.1]$	0.15236
MLP	AUCDC	$+0.0079$	$[-0.0105, 0.0248]$	0.46374
Random trees	AUCDC	$+0.0078$	$[-0.0035, 0.0190]$	0.46374

the promotion and its concentrated support. Reusing these folds to formulate and select the rule does not independently validate it.

In the subsequent first acquisition batch, both the tree and MLP ranking discover 19 hits. The full Astra campaign later finishes with 176 hits versus 191 for MLP, but different trees are developed at later checkpoints, so that campaign difference cannot be attributed to this initial rule. The example illustrates how a scientist can identify the branch behind a promotion, examine its support and counterexamples, and question its transfer to new candidates. Interpretation applies to the acquisition policy. The internal computations of the MLP and language model remain outside this account.

D ALDH1 DATA AND SUPPLEMENTARY RESULTS

D.1 SOURCE AND MOLECULAR PREPARATION

We use the full ALDH1 files released by the LIT-PCBA authors, associated with PubChem assay identifier 1030 (Tran-Nguyen et al., 2020). The download contains 145,133 rows, comprising 7,168 rows labeled active and 137,965 labeled inactive. We retain these binary labels and construct our own grouped partitions from the full files. We do not use the benchmark's original partitions. The archive, source files, retrieval metadata and checksums are retained with the prepared data.

All downloaded structures pass RDKit parsing and sanitization. Canonicalization identifies 142,880 unique structures, including 22 with conflicting labels. We remove every row belonging to a conflicting structure and retain one row for each remaining structure. The analysis therefore uses 142,858 molecules, of which 7,123 are active (4.99%). No structure contains disconnected fragments. Molecular preparation and fingerprint construction follow Appendix B.

D.2 PARTITIONS

The grouped split gives the partitions in Table 8. No nonempty scaffold or identical fingerprint occurs in more than one partition. The 549 molecules with empty scaffolds follow the separate-group policy described in Appendix B. Each campaign begins with 1,000 labeled molecules from the acquisition-pool partition, leaving 99,002 candidates for acquisition. The initial selection keeps identical-fingerprint groups together without consulting activity labels. The same campaign initialization is used across methods. The held-out test set is excluded from sequential discovery and supports the separate transfer assessment in Appendix B.5.

Table 6: HIV paired acquisition comparisons for Qwen trees across ten campaign initializations. Gains are LLM-tree outcomes minus the comparator. Holm adjustment covers these four tests jointly. No test rejects at the 5% level. Bootstrap intervals are descriptive and unadjusted.

Comparator	Outcome	Mean gain	95% interval	Holm p
MLP	Hits	$+8.8$	$[0.2, 17.0]$	0.34820
Random trees	Hits	$+6.6$	$[-3.2, 14.5]$	0.65526
MLP	AUCDC	$+0.0221$	$[-0.0099, 0.0536]$	0.65526
Random trees	AUCDC	$+0.0221$	$[-0.0108, 0.0547]$	0.65526

Table 7: Astra's first-round policy for the illustrative replicate. The variable c is the fraction of carbon atoms with sp^3 hybridization, and s is the maximum fingerprint Tanimoto similarity to an initially known active. Counts describe all search molecules in each leaf, not the acquired batch. Higher priority takes precedence. Raw MLP scores order molecules with equal priority, including across different leaves.

Condition	Priority	Search molecules	Active
$c < 0.5$, $s < 0.3000$	0	3,665	105
$c < 0.5$, $s \geq 0.3000$	1	127	18
$c \geq 0.5$, $s < 0.2361$	0	1,234	29
$c \geq 0.5$, $s \geq 0.2361$	2	90	24

Table 8: ALDH1 data partitions shared across campaigns and methods. Counts refer to molecules. The acquisition pool includes the initial 1,000 labeled molecules. The development partition provides the search and withheld audit portions described in Appendix B.

Partition	Total	Active	Active (%)
Development	21,428	1,068	4.98
Acquisition pool	100,002	4,987	4.99
Held-out test	21,428	1,068	4.98

Table 9: ALDH1 paired acquisition comparisons for Astra trees across ten campaign initializations. Gains are LLM-tree outcomes minus the comparator. Holm adjustment covers these four tests jointly. No test rejects at the 5% level. Bootstrap intervals are descriptive and unadjusted.

Comparator	Outcome	Mean gain	95% interval	Holm p
MLP	Hits	$+5.8$	$[-0.8, 11.0]$	0.44428
Random trees	Hits	$+6.2$	$[-2.4, 14.6]$	0.44428
MLP	AUCDC	$+0.0112$	$[-0.0126, 0.0329]$	0.44428
Random trees	AUCDC	$+0.0202$	$[-0.0017, 0.0406]$	0.44428

Table 10: ALDH1 paired acquisition comparisons for Qwen trees across ten campaign initializations. Gains are LLM-tree outcomes minus the comparator. Holm adjustment covers these four tests jointly. No test rejects at the 5% level. Bootstrap intervals are descriptive and unadjusted.

Comparator	Outcome	Mean gain	95% interval	Holm p
MLP	Hits	$+7.6$	$[-0.8, 18.3]$	0.34318
Random trees	Hits	$+8.0$	$[1.2, 15.5]$	0.23373
MLP	AUCDC	$+0.0143$	$[-0.0096, 0.0408]$	0.34349
Random trees	AUCDC	$+0.0233$	$[0.0030, 0.0407]$	0.21188

D.3 ACQUISITION COMPARISONS

Tables 9 and 10 report the Astra and Qwen comparisons using the same paired procedure. Each LLM workflow has a separate four-test Holm family, as defined in Appendix B.3. None of these acquisition comparisons reaches the 5% significance level after Holm correction. The intervals describe paired campaign variability and are not adjusted for multiple comparisons.

Qwen improves on MLP in six replicates, ties in one and discovers fewer hits in three, with differences ranging from 11 fewer to 47 more hits. Astra improves in eight replicates, ties in one and loses 19 hits in the remaining replicate. The descriptive 95% intervals for the mean hit gains over MLP are $[-0.8, 18.3]$ for Qwen and $[-0.8, 11.0]$ for Astra. Qwen's larger mean therefore accompanies less consistent improvement across these initializations.