H2L-BENCH: A BENCHMARK FOR AGENTIC PLANNING UNDER UNCERTAINTY IN DRUG DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

AI agents have made rapid progress on tool use, multi-step planning and open-ended reasoning, yet the benchmarks that drive this progress rarely capture what matters in scientific decision making: acting under uncertainty, choosing between costly experiments, and committing to an answer with limited information. We introduce *H2L-Bench*, a lightweight decision-theoretic sandbox that simulates the hit-to-lead phase of small-molecule drug discovery as a budget-constrained sequential-decision task with a mechanistic ground truth. An agent chooses among eight assays of varying cost and information content, then nominates one compound from a library. Winning requires the compound's hidden traits to clear multiple thresholds simultaneously, with assay noise and budget scaled across three difficulty tiers. As a principled reference we implement a Bayesian value-of-information (VoI) agent that maintains an analytical Gaussian posterior over each compound's latent traits and picks the assay that most increases the best compound's probability of winning per dollar spent. Comparing VoI to four frontier LLMs (Claude Haiku 4.5, Sonnet 5, Opus 5, and OpenAI GPT-5.4) at 50 seeds per cell, we find that VoI has the highest win rate on every tier, but in paired same-seed tests its lead over the best LLM on each tier is not significant after correction for multiple comparisons. Beyond the headline win rates, the environment lets us analyse where LLMs and a Bayesian agent diverge action-by-action: on a subset of seeds, LLMs recover the correct compound by committing budget to a confirmatory profile of one compound, or to an expensive singular assay, that a depth-1 per-dollar VoI ranking keeps deferring. An approximate two-step lookahead does not close this gap on fresh seeds, so the shortfall reflects VoI's per-dollar scoring rule rather than its planning horizon. Test-time scaffolds do not help: on the hard tier, chain-of-thought and tree-of-thoughts leave Sonnet 5's win rate statistically unchanged, and explicit planning lowers it (32% to 23%).

1 INTRODUCTION

LLM-based agents have made striking recent progress on multi-step tasks in software engineering (Yang et al., 2024), web navigation (Nakano et al., 2021; Deng et al., 2023) and workflow orchestration (Yao et al., 2023b; Schick et al., 2023). Much of the excitement around scientific applications extrapolates these gains to open-ended discovery pipelines like drug design (Bran et al., 2024; Boiko et al., 2023; Gottweis et al., 2026; Lu et al., 2024) and materials discovery (Choudhary et al., 2022; Szymanski et al., 2023). While standard agent benchmarks reward completing a task correctly, scientific discovery requires answering uncertain questions: *Which experiment should I run next given what I already know? When have I gathered enough evidence to commit? How do I trade cheap noisy readouts against expensive definitive ones?* There are few controlled, reproducible benchmarks for these capabilities, as we discuss in Section 2.

In this work, we introduce **H2L-Bench**, a compact benchmark modelling the hit-to-lead phase of small-molecule drug discovery (Figure 1). The agent starts with a library of 24 candidate compounds and a fixed budget, and must nominate one lead whose hidden traits clear the hard thresholds (potency, selectivity, programme-specific ADMET/PK). It may pay for any of eight noisy assays, from coarse in-silico prediction to expensive in-vivo pharmacokinetics, each with distinct cost and information content. Five programme archetypes and three difficulty tiers span realistic variation.

Because the environment's noise model is known, we can build a strong decision-theoretic reference policy: a Bayesian value-of-information (VoI) agent (Howard, 1966; Chaloner & Verdinelli, 1995;

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.

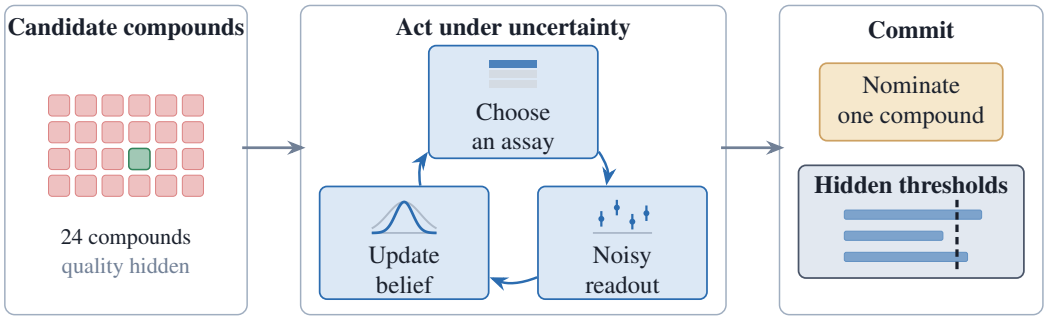


Figure 1: The H2L-Bench decision problem. From a library of 24 candidate compounds whose quality is hidden (left), the agent runs a per-turn loop under uncertainty (centre): it chooses one of eight noisy assays, from cheap in-silico predictions ($50) to expensive in-vivo pharmacokinetics ($8k), observes a noisy readout, and updates its belief, repeating until the budget is spent. It then commits a single nomination (right), which wins only if the chosen compound's hidden traits clear every hard threshold (potency, selectivity, programme-specific ADMET/PK). Five programme archetypes $\times$ three difficulty tiers (budgets $60k/$35k/$20k) span realistic variation.

Rainforth et al., 2024) that picks the assay with the largest expected increase in the best compound's win probability per dollar. This gives a principled reference against which LLM and heuristic agents can be benchmarked directly, rather than only against each other. The VoI agent is depth-1 myopic (it ranks single $(assay, compound)$ pairs by VoI divided by cost, rather than evaluating action sequences) and so it is a reference, not an upper bound.

Our contributions are as follows: **(1)** a lightweight benchmark for agentic decision-making under uncertainty with realistic cost structure; **(2)** a computationally-tractable Bayesian VoI agent as a reference; **(3)** a comparison of VoI with greedy, four frontier LLMs (Claude Haiku 4.5, Sonnet 5, and Opus 5; OpenAI GPT-5.4), three test-time scaffolds, Thompson sampling and PPO, and an analysis of the seeds where an LLM beats VoI, which reveals a concrete failure mode of depth-1 VoI. Beyond this benchmark, these results showcase the potential of LLMs to reason under uncertainty and make sequential decisions in a scientific context, and highlight the tradeoff between "safe bets" and "high-risk high-reward" strategies in such scenarios.

2 Related Work[1]

Agentic benchmarks and evaluation: Recent LLM-agent benchmarks have focused on task completion in deterministic environments, such as WebArena (Zhou et al., 2024b) for web navigation, SWE-bench (Jimenez et al., 2024) for code generation, τ-bench (Yao et al., 2025) for tool-use in customer service. There are also general-knowledge benchmarks, such as MMLU (Hendrycks et al., 2021) and BIG-Bench (Srivastava et al., 2023) which test factual recall or reasoning without any tool use. These evaluate whether an agent can produce a correct output, not whether it can decide what to measure next.

Scientific-agent benchmarks: A growing set of benchmarks target scientific tasks: ChemCrow (Bran et al., 2024) and Coscientist (Boiko et al., 2023) evaluate LLM agents on real chemistry design, showing they can orchestrate real tools; ScienceAgentBench (Chen et al., 2025), HeurekaBench (Panigrahi et al., 2026), LifeSciBench (Liu et al., 2026), BixBench (Mitchener et al., 2025), LAB-Bench (Laurent et al., 2024) and PaperBench (Starace et al., 2025) test data-driven scientific reasoning; BoxingGym (Gandhi et al., 2025) and DiscoveryWorld (Jansen et al., 2024) test experimental design and discovery, without per-experiment costs or a budget. These benchmarks judge success by task completion or answer correctness, not by how well the agent trades observation cost against information gain. They also have no computable reference policy to compare agents against.

Sequential decision-making under uncertainty: Bayesian value-of-information dates to Howard (1966) and Lindley (1956), and has been applied to sequential decision-making in a variety of domains. Modern reviews (Chaloner & Verdinelli, 1995; Rainforth et al., 2024) cover Bayesian

[1]This field is large and growing rapidly. We have tried to cover the most relevant work and apologise for any omissions.

experimental design broadly. VoI has been applied to clinical-trial planning (Willan & Pinto, 2005) and active-learning-based hit discovery (Reker & Schneider, 2015), but not to our knowledge as a benchmark reference agent for LLM comparison. Structurally, the task is fixed-budget best-arm identification (Audibert & Bubeck, 2010) with constraints and heterogeneous measurement costs, a setting in which LLMs are known to explore poorly (Krishnamurthy et al., 2024). In Bayesian Optimization terms, VoI is the knowledge gradient (Frazier et al., 2008) on $P(\text{win})$ per dollar (Snoek et al., 2012), with cheap assays as low fidelities (Kandasamy et al., 2016) and a constrained multi-objective target (Gelbart et al., 2014; Daulton et al., 2020). GP acquisitions (EI (Jones et al., 1998), GP-UCB (Srinivas et al., 2010)) need a kernel over compound features, which H2L-Bench does not expose, so our BO baseline is Thompson sampling (Thompson, 1933) on VoI's posterior. Molecular BO searches chemical space (Griffiths & Hernández-Lobato, 2020); H2L-Bench fixes the library and asks which measurements to buy. The value of information is non-concave in the number of measurements, so myopic knowledge-gradient policies undervalue measurements that pay off only in combination (Frazier & Powell, 2010), as in the failure mode of §5.2.

Environments with theoretical optima: A few benchmarks admit computable optimal policies. For example, small POMDPs (Kaelbling et al., 1998), gridworlds like BabyAI (Chevalier-Boisvert et al., 2019), and single-shot Bayesian experimental design tasks. None of these combine (a) rich, multi-step sequential decisions, (b) explicit budget constraints and (c) noisy heterogeneous measurements with different costs and information content. H2L-Bench has no computable optimum either; what its known generative model allows is a Bayesian reference policy.

Our Contribution: H2L-Bench combines sequential decisions over 24 compounds, a hard budget cap, and eight assays with distinct cost/noise tradeoffs, in a hit-to-lead scientific framing, while remaining synthetic and small enough that a Bayesian VoI reference policy is computable. This lets us benchmark LLM agents against a principled reference agent rather than only against each other.

3 The H2L-Bench Environment

In the H2L environment, a *scenario* consists of **(1)** a pool of $N = 24$ candidate compounds, each with nine hidden numeric traits in $[0, 1]$ (potency, selectivity, solubility, permeability, metabolic stability, cardiotox safety, oral bioavailability, half-life and lipophilicity) and a hidden artifact flag that biases some assays; **(2)** a threshold on every numeric trait (a subset are "hard" and set by the archetype, the rest are easy floors of 0.2); and **(3)** a per-scenario budget in dollars. On each turn the agent chooses one of the eight assays in Table 1, receives a noisy observation, and updates its beliefs. When done, it nominates one compound, which is considered a winner if all thresholds clear. Lipophilicity is never observed directly: it drives the correlation between solubility, permeability and cardiotox safety, so ADMET readouts carry indirect evidence about it, and its floor alone decides 2–9 of each agent's 750 games. Each (compound, wet-lab assay) pair may only be run once per scenario, a rule that mimics the practice that medicinal chemists confirm suspicious results by orthogonal assays rather than serial replicates, and stops agents from trivially averaging out noise by repeatedly running the same assay on the same compound.

We include five programme *archetypes* with distinct hard-axis structures (*kinase-oncology*, *GPCR-CNS*, *antibiotic-bacterial*, *PPI-oncology*, and *protease-chronic*). Each specifies its own hard thresholds. Full definitions are in Appendix A. H2L also spans three different difficulty tiers (easy, medium, hard) that scale the per-assay noise, the artifact rate, and the total budget (\$60k / \$35k / \$20k). Medium and hard draw compounds from the same priors, so on a given seed the two tiers have the same library and differ only in the observation channel and budget. Easy also raises the priors on five traits (potency, selectivity, metabolic stability, oral bioavailability, half-life) from $\text{Beta}(3, 3)$ to $\text{Beta}(5, 2)$, so its libraries contain more winners. Full parameters can be found in Appendix E.

Scope and fidelity. H2L-Bench is a stylised decision testbed, not a chemistry-accurate simulator. Assay costs, Beta priors, and coupling coefficients are order-of-magnitude choices that produce distinct difficulty tiers, not calibrated on any specific programme. Our claims are about relative ordering and divergence mechanisms of agents, not absolute performance on real drug discovery.

4 Agents

All agents share the same public interface: see the game state, return an action.

Greedy. A fixed hit-to-lead programme that takes single readings at face value (no uncertainty). It screens the whole library with `single_shot_screen` and runs `dose_response` on the first six

Table 1: The eight assays. Costs are approximate industry-scale figures; each observes one or more of a compound's latent traits with characteristic noise. Some observations are biased by an artifact flag (see Appendix E).

Assay	Cost (USD)	Observes
`predict_in_silico`	50	coarse potency, coarse ADMET summary
`single_shot_screen`	200	Bernoulli hit/miss (batched)
`dose_response`	1,500	biochemical IC_{50} (artifact-biased)
`admet_panel`	2,000	4 ADMET traits + hERG flag
`cell_potency`	3,000	cellular EC_{50} (potency $\times$ permeability)
`orthogonal_confirm`	4,000	artifact-free IC_{50}
`selectivity_panel`	5,000	target IC_{50}, selectivity, off-target IC_{50}s
`in_vivo_pk`	8,000	oral bioavailability + half-life

hits in library order. Compounds whose IC_{50} is at most 10 μM go on to `orthogonal_confirm`, and those that stay at or below 10 μM on this artifact-free assay survive. It profiles the survivors on the assays that measure the scenario's hard thresholds (`selectivity_panel`, `admet_panel`, and `in_vivo_pk` if the budget covers it for every survivor), and nominates the survivor whose readings clear the most hard thresholds.

Value of Information (VoI). The VoI agent is a depth-1 Bayesian reference policy. It is not a direct comparison to LLMs since it has access to the ground-truth generative model and updates its posterior with it, but it gives a principled per-cell reference against which LLM and heuristic behaviour can be measured. Each compound has a Bayesian posterior: A mixture over the discrete `is_artifact` latent, with continuous latents represented as Gaussians on the appropriate transformed scale (logit for bounded traits, log for IC_{50}-type readouts). Four correlated ADMET traits live in a 4-D multivariate Gaussian to capture lipophilicity coupling; the rest are independent 1-D Gaussians. Prior parameters are moment-matched to the generator's Beta distributions.

At each turn, for each candidate action a we compute

$$\mathrm{VoI}(a) = \mathbb{E}_{o\sim p(\cdot|a,\text{posterior})}\Big[\max_{c'} P(\text{win}|c', o)\Big] - \max_{c'} P(\text{win}|c'),$$

via $N = 256$ Monte Carlo draws from the posterior predictive, each followed by a Bayesian update of the posterior. The update is a closed-form Kalman step for `predict_in_silico`, whose potency readout is linear-Gaussian on the logit scale; an extended Kalman filter (EKF) linearisation for the three IC_{50} assays (`dose_response`, `orthogonal_confirm`, `selectivity_panel`); and 1-D or 2-D grid integration for the screen and for the three assays (`in_vivo_pk`, `cell_potency`, `admet_panel`) whose nonlinear observation models made EKF linearisation biased in practice. After each update the posterior is projected back to a Gaussian by moment matching. Actions are ranked by VoI divided by dollar; VoI keeps acting until no affordable un-run assay remains, so it spends its entire budget by construction (Appendix G). Compounds are pruned as evidence accumulates: once a compound's $P(\text{win})$ falls below 2% of the current front-runner's, it is ruled out for the rest of the game and no longer evaluated or nominated. When no affordable action remains, VoI nominates the surviving compound with the highest posterior $P(\text{win})$. Appendix L varies the rule-out threshold, the number of Monte Carlo draws, and an exponent α on the cost in the per-dollar ratio ($\mathrm{VoI}/\mathrm{cost}^{\alpha}$; $\alpha = 1$ in all main results). Full posterior specification and derivation are in Appendix F; approximations and their consequences in Appendix G. The main approximation is depth-1 myopia: VoI picks the single best action rather than the best sequence. An approximate two-step variant does not improve on it (Appendix I), so we attribute VoI's losses to the per-dollar scoring rule rather than to lookahead depth.

LLM agents. We test four LLMs: Claude Haiku 4.5, Sonnet 5, and Opus 5, and OpenAI's GPT-5.4. Sonnet 5 and Opus 5 run with their default adaptive extended thinking, GPT-5.4 with medium reasoning effort; Haiku 4.5 answers directly. All use the same scaffolding: each turn the agent gets a state summary (target, hard thresholds, remaining budget, per-compound observation rollup) and returns a single JSON action. All LLM agents are sampled at temperature 1, so repeated runs on a seed are non-deterministic. We do not tell the LLM which archetype it is playing; it sees the same state fields as the VoI agent (target name, hard thresholds, per-compound observation history). What differs is knowledge of the generative model. VoI is given the trait priors (including each archetype's

prior overrides, which it looks up from the target name and hard thresholds), noise scales, artifact rates, and the map from latent potency to IC_{50} ($\log_{10} IC_{50}/\mu M = 2 - 4 \cdot$ potency); the LLM's prompt (Appendix M) describes each assay's noise only qualitatively and gives none of these values. The prompt describes the rule of not repeating (compound, experiment) pairs in the same session. Illegal actions get up to three retries before we nominate a random compound so the game terminates. This setup is functionally a ReAct (Yao et al., 2023b) loop at the granularity our environment supports: each atomic game step is one Thought–Action–Observation triple, with the Thought carried by the model's hidden reasoning (for Sonnet 5 on the hard tier, about 930 thinking tokens per turn) rather than written into the reply. We also test three test-time scaffolds on Sonnet 5 at the hard tier. ***CoT*** asks the model to write its reasoning before the action. ***Tree-of-Thoughts*** (Yao et al., 2023a) asks for three candidate actions, scores each from 0 to 10 in a separate call, and plays the highest-scoring one. ***Planner*** (Wang et al., 2024) asks at the first turn for a full plan of actions up to nomination and plays its steps in order, skipping any that have become invalid. It asks for a new plan from the current state every five turns or when the plan runs out, so each plan commits to up to five assays before any of their results are seen.

5 Results

5.1 Main Results

Each agent plays 50 seeds per (archetype, difficulty) cell, totalling 750 games per agent. Table 2 reports the aggregate results per tier; Figure 2 shows the per-cell breakdown. The ordering is stable across tiers: greedy is weakest, Haiku next, then a cluster of stronger LLMs (Sonnet 5, Opus 5, GPT-5.4), with VoI highest on every tier. Its lead over the best LLM is 7 points on easy (Opus 5), 5 on medium (GPT-5.4) and 3 on hard (Opus 5); the paired tests below find none of these three gaps significant. To show how much of the gap to a perfect score is due to assay noise rather than budget, we also report an *informative oracle*. It ignores the budget, runs every assay once on every compound (the most any agent can observe, since repeats are not allowed), combines the readouts with VoI's posterior, and nominates the compound with the highest $P(\text{win})$. It still loses games because a single noisy readout per assay does not always single out the winner. It estimates what is achievable under this noise, but it is not a per-game upper bound: it draws its own assay noise, so an agent can win a game that the oracle loses (Appendix B).

We also evaluate two additional agents, described with their results in Appendix D: a from-scratch PPO policy trained directly on the environment ($\sim$157k parameters, one model per tier; Appendix D.2), and a Thompson sampling agent, which each turn draws one sample from VoI's posterior and runs the cheapest assay on the compound that looks best in that draw, targeting the hard trait that compound is least likely to clear (Appendix D.1). Neither beats VoI on any tier. Thompson sampling is level with Opus 5 and VoI on easy (82%) but falls to Haiku 4.5's level on hard (22%); PPO, trained from scratch, matches Sonnet 5 and GPT-5.4 on hard (34%).

Table 2: Aggregate win rates (250 games per tier). Oracle runs every assay once on every compound, ignoring the budget (Appendix B); VoI is the depth-1 Bayesian reference at fixed budget.

Tier	greedy	Haiku 4.5	Sonnet 5	Opus 5	GPT-5.4	VoI	oracle
easy	27%	37%	65%	80%	70%	87%	89%
medium	18%	32%	43%	51%	54%	59%	84%
hard	10%	21%	32%	40%	33%	43%	76%

Because every agent plays the same seeds within a cell, we compare agents with McNemar's exact test for paired binary outcomes (McNemar, 1947) (an exact binomial test on the discordant pairs) pooled per tier (250 seeded pairs). Each tier involves 15 pairwise tests, and with that many tests some would fall below $p = 0.05$ by chance alone, so we apply Holm's correction (Holm, 1979) within each tier; every p reported below is the corrected value (all 45 tests in Appendix C).

Four patterns stand out (Figure 2; all p-values are Holm-corrected paired tests from Appendix C).

(1) On every tier, the results follow a pattern: greedy $<$ Haiku 4.5 $<$ each of Sonnet 5, Opus 5 and GPT-5.4.

(2) On each tier the best LLM (Opus 5 on easy and hard, GPT-5.4 on medium) trails VoI by 7, 5 and 3 points, and none of these gaps is significant ($p = 0.073, 0.46, 1.0$). The weaker frontier LLMs

do fall behind VoI on the easier tiers: GPT-5.4 and Sonnet 5 on easy ($p < 0.001$) and Sonnet 5 on medium ($p = 0.001$). On hard, neither is significantly behind ($p \geq 0.06$).

(3) The gap to the informative oracle widens with difficulty. On easy the best agent is within 2 points of the informative oracle, but on medium and hard it is 25 and 33 points below it (Table 2).

(4) The LLMs become harder to tell apart as difficulty rises. On easy, Opus 5 beats both Sonnet 5 and GPT-5.4 ($p \leq 0.014$); on medium, only GPT-5.4 beats Sonnet 5 ($p = 0.025$); on hard, no pair of LLMs differs significantly.

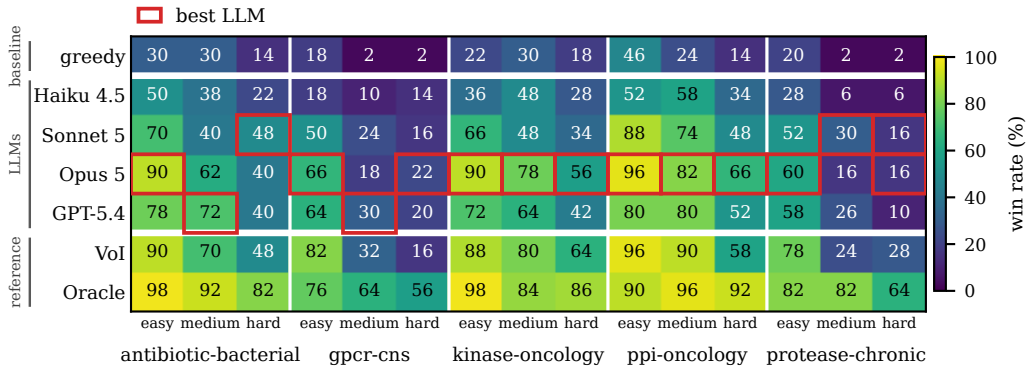


Figure 2: Per-cell win rates (%) at 50 seeds per cell. Rows are agents, grouped into three bands (baseline, LLMs, reference policies); columns are (archetype, tier) cells. Lighter means higher win rate. A red outline marks the best-performing LLM in each column (both cells when tied). The reference band shows VoI (depth-1 Bayesian, fixed budget) and the informative-oracle noise ceiling (non-budgeted). Kinase-oncology and PPI-oncology are the cleanest cells for VoI; GPCR and protease remain hard for every agent. VoI has the highest win rate on 7 of 15 cells and ties the best LLM on 3 more, but sits well below the oracle on medium and hard — where a budgeted agent could still improve substantially.

5.2 Where VoI loses and an LLM wins

VoI loses 278 of its 750 games. We want to focus particularly on games where VoI loses, and at least one of the LLM wins. Therefore, we take seeds in which VoI lost, and at least one of the two strongest Claude models, Sonnet 5 and Opus 5, won: $N = 134$, pooled across all cells (per-cell rates in Figure 2). From the action logs and the ground-truth scenario we ask, for each, why VoI missed a winner the LLM found.

The diagnosis: in these losses, VoI's targeting is no better than chance. VoI can afford confirmatory (wet-lab) assays on only $\sim$7 of 24 compounds; the overlap between the compounds it confirms and the true winners is $0.82\times$ the random base rate. Its per-dollar ranking does not preferentially steer deep testing towards winners. As a result VoI never confirmatory-tests *any* true winner in 74% of these losses (we call this a *coverage* failure); in the remaining 26% it does test a winner but still nominates a loser (we call this a *ranking* failure).

We also find that the missed winners are not just unlucky: they are systematically hard to distinguish from near-misses. For 74% of these seeds the LLM-found winner clears its tightest hard threshold by a razor margin (median 0.026 in trait units). These are exactly the compounds that must be confirmatory-tested to be told apart from near-misses, and exactly the ones VoI's per-dollar ranking fails to prioritise, so they are systematically under-ranked both when VoI chooses what to confirm and when it nominates. Table 3 crosses the two axes. Behaviourally, in these losses VoI hedges confirmatory budget across $\sim$7 candidates and spends 99% of its budget; the winning LLMs concentrate on $\sim$4 and stop at 79%, committing early to a shortlist. Unspent budget earns nothing; what differs is where the budget goes.

This explains why VoI loses on these seeds, but then it is natural to ask what the LLMs do instead. Two cells show this concentration in detail: ppi-hard (VoI 58% vs Opus 5 66%) and antibiotic-medium (VoI 70% vs GPT-5.4 72%, a one-game lead). On the seeds VoI loses, the LLM's winning trajectories share a shape: budget committed to an expensive confirmatory assay that VoI's per-dollar, one-turn-at-a-time ranking keeps deferring in favour of cheaper information.

Table 3: Why VoI loses, on the $N = 134$ seeds where VoI lost and at least one frontier LLM (Sonnet 5 / Opus 5) won. Rows: whether VoI ever ran a confirmatory assay on a true winner. Columns: whether the LLM-found winner clears its tightest hard threshold by < 0.05 (marginal) or by more (comfortable).

	marginal	comfortable	total
coverage failure (never tested a winner)	73	26	99
ranking failure (tested, nominated loser)	26	9	35
total	99	35	134

Strategy 1 (Opus 5, ppi-hard) — profile one compound to confirmation. On the seeds VoI loses, Opus 5 concentrates its budget on `admet_panel` (3.7 vs 1.4 calls/game) and `selectivity_panel` (1.0 vs 0.2), fully profiling its lead compound, and runs essentially no `cell_potency` (0.0 vs 4.2). VoI instead spreads `cell_potency` ($3k) across compounds and never confirms a winner: each confirmatory call is individually unattractive per dollar, so VoI defers them until the budget is gone.

Strategy 2 (GPT-5.4, antibiotic-medium) — pay for the decisive assay. `in_vivo_pk` at $8k is 23% of the budget and looks bad per dollar. GPT-5.4's winning games run it four times as often as VoI does (0.8 vs 0.2 calls/game), trading down cheap `predict_in_silico` (12.8 vs 20.8) for the one measurement that resolves the hard axis. VoI keeps sampling cheap in-silico prediction and rarely runs the assay at all.

Both are the same shape of failure: value that only materialises once budget is committed to a compound's confirmatory profile, invisible to a marginal per-turn ranking. Figure 3 shows the action mixes on the divergent seeds of both cells.

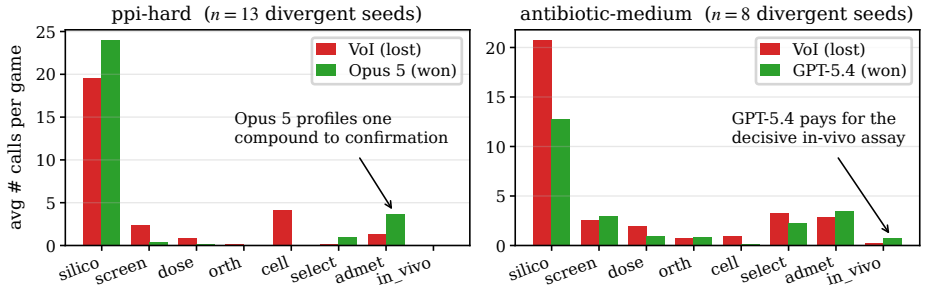


Figure 3: Action distributions on the divergent seeds of two cells where a frontier LLM beats depth-1 VoI. *Left:* ppi-hard — Opus 5 spends its budget on `admet_panel` and `selectivity_panel` to confirm one compound, where VoI runs `cell_potency` instead. *Right:* antibiotic-medium — GPT-5.4 pays for the expensive `in_vivo_pk` assay VoI defers. Bars average per-assay call counts over the seeds where the LLM won and depth-1 VoI lost (13 and 8 seeds respectively).

Ruling out other explanations. Appendix H checks three other explanations for VoI's losses and finds none is the main cause: assay noise, VoI's Gaussian posterior approximation, and too little lookahead (an approximate depth-2 VoI wins fewer games than depth-1).

5.3 Test-time scaffolds and the base model's commitment

We define the *commitment* as the share of an agent's confirmatory (wet-lab) budget spent on its single most-tested compound, averaged over games. It is a simple, interpretable summary of how much an agent hedges its budget across candidates. The commitment axis also helps explain a negative result on the LLM side: none of three test-time reasoning scaffolds improves on the plain agent (Table 19 in Appendix K). On the hard tier, chain-of-thought (CoT), Tree-of-Thoughts (ToT; three proposals scored per turn), and an explicit up-front Planner — all on Sonnet 5 — win 33%, 29%, and 23% against base Sonnet 5's 32%. Only Planner's drop is significant in paired tests ($p = 0.033$; one test per scaffold on all 250 hard-tier games, Holm-corrected over the three scaffolds). The ToT we test is shallow (best-of-three at each turn, with no lookahead or backtracking), so these results do not cover tree search over action sequences such as RAP (Hao et al., 2023) or LATS (Zhou et al., 2024a).

Within base Sonnet 5's hard-tier games, wins concentrate confirmatory (wet-lab) budget on fewer compounds than losses (2.8 vs 3.7 compounds; 0.68 vs 0.54 of that budget on the single most-tested

compound) and stop with budget to spare — the same contrast as VoI versus the frontier LLMs above, now within a single agent (Figure 4, right).

CoT keeps the base commitment profile (0.60 concentration on 3.4 compounds, vs. 0.58 on 3.4) and the base win rate. The written reasoning adds little deliberation: the base model already thinks for about 930 tokens per turn before answering, and CoT's total output is about the same (990 tokens, about 150 of them visible), so CoT moves part of the existing reasoning into the reply rather than adding more.

On the other hand, ToT and Planner move the commitment in opposite directions. ToT reverts the model towards VoI-style hedging: it spreads confirmatory budget across 4.6 compounds at 0.48 concentration but without VoI's calibrated posterior, so it lands below both base Sonnet and VoI. Planner does the reverse, concentrating harder than base (0.79 on 2.6 compounds) but fixing its target before cheap screening can inform the choice; within Planner concentration no longer predicts winning (0.74 for wins vs 0.80 for losses). Figure 4 (left) places all five agents on the commitment axis.

Overall, in the base model's per-turn judgement, adaptive commitment (concentrate late, once cheap screening has identified a candidate) goes with winning. This is a correlation, not a causal test: a game with a clear early leader favours both. Writing that judgement out (CoT) changes neither commitment nor win rate. Replacing it with an explicit control structure moves commitment away from the base setting, towards too little (ToT) or too much (Planner); only Planner's larger shift costs a significant amount of win rate.

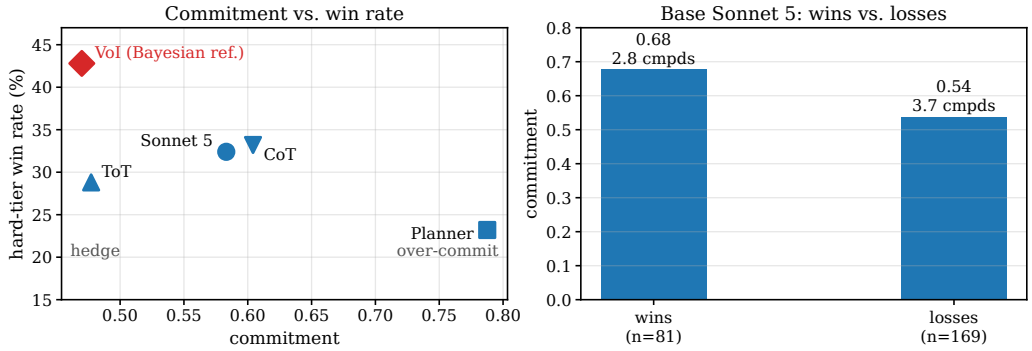


Figure 4: Test-time scaffolds and commitment (hard tier, 250 games per agent). *Commitment* is the share of an agent's confirmatory (wet-lab) assay budget spent on its single most-tested compound, averaged over games. *Left:* hard-tier win rate vs. commitment. Among the LLM-family agents, base Sonnet 5 and CoT sit together at moderate commitment; ToT hedges (spreading budget like VoI) and Planner over-commits, both below base (only Planner significantly; Table 19). VoI wins while hedging on the strength of its calibrated posterior — commitment helps the LLM, but does not substitute for a good prior. *Right:* within base Sonnet 5, winning games have higher commitment than losing games (0.68 vs 0.54) and spread confirmatory budget over fewer compounds (2.8 vs 3.7); the association is correlational.

5.4 Scaling with budget and compound count

To study how the results vary with the number of compounds to screen, and with the available budget, we sweep two resources on kinase-oncology / medium (50 seeds per point): the per-game budget, and the number of candidate compounds at a fixed $35k budget (Figure 5).

Budget. At the smallest budgets the LLMs are level with or ahead of VoI: at $5k Opus 5 and Sonnet 5 win 42% and 40% against VoI's 34%, and at $10k all four are within three games of each other (42–48%). With almost no room for confirmatory assays, a good prior beats sequential information-gathering. From $20k to $100k VoI leads every LLM, and it reaches 84%, the informative oracle's level, at $60k. Sonnet 5 and Opus 5 level off lower. GPT-5.4 is the exception among the LLMs: it tracks Opus 5 up to $35k but keeps climbing where the others flatten, reaching 86% at $200k, one game above both VoI and the informative oracle (84%). As previously discussed, an agent can beat the oracle because the oracle's noise comes from a separate seed, from the noise in the agent's game. At $200k the budget rarely limits GPT-5.4 (it spends $70k on average), so it plays near the noise

limit, and whether it lands a game above or below the oracle is chance: it wins 6 seeds the oracle loses and loses 5 that the oracle wins.

Compound count. Holding the budget at $35k and enlarging the candidate set makes the search harder for every agent, but VoI degrades slowest. With 8 or 16 compounds it wins 96%, against 72–84% for the LLMs. At 24 and 48 compounds Opus 5 comes close to VoI. At 96 compounds VoI still wins 54%, while Opus 5, Sonnet 5 and GPT-5.4 fall to 28%, 16% and 16%, and greedy falls from 66% at 8 compounds to 2%. VoI's advantage therefore grows with the size of the library and is not specific to the default 24-compound setting.

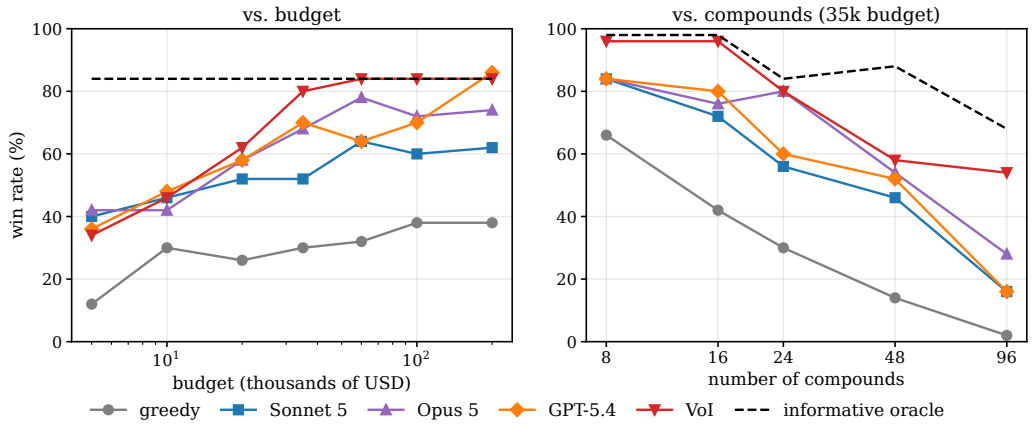


Figure 5: Win rate vs. per-game budget (left) and vs. number of candidate compounds at a fixed $35k budget (right), on kinase-oncology / medium (50 seeds per point). The dashed line is the informative oracle, a noise-limited reference from one noise draw that an agent can exceed by chance (Appendix B); it does not depend on the budget, so it is flat (84%), and it falls as the compound set grows; its rise from 84% at 24 compounds to 88% at 48 is two games, within sampling noise. The LLMs are level with or ahead of VoI at the two smallest budgets; VoI leads from $20k to $100k and degrades slowest as the compound set grows; GPT-5.4 is the only LLM still improving at $200k.

6 Conclusions and Limitations

H2L-Bench packages a scientific decision-under-uncertainty problem into a small open sandbox where a Bayesian reference agent, LLMs, and any future method can be compared action-by-action. In our sweep, the ordering is greedy $<$ Haiku 4.5 $<$ a cluster of Sonnet 5, Opus 5, and GPT-5.4 $<$ VoI, although in paired same-seed tests VoI is not significantly ahead of the best LLM on any tier. Trajectory analysis on the seeds where LLMs occasionally beat VoI surfaces two commitment patterns that a depth-1 per-dollar ranking keeps deferring. More importantly, this benchmark helps us understand the behaviours of LLMs in a scientific decision-making context. For example, the LLMs' emergent ability to commit to a candidate compound and concentrate confirmatory assays on it, once cheap screening has identified it, leads, on the seeds where an LLM beats VoI, to better outcomes than a purely Bayesian approach that is more conservative and hedges across multiple candidates. We hope that H2L-Bench will be a useful tool for the community to explore and understand the capabilities and limitations of LLMs in scientific decision-making tasks, and to ultimately improve the design of future LLMs for scientific applications.

Limitations and broader impacts. Beyond "Scope and fidelity" (§3): the task is single-shot nomination on a fixed library (real hit-to-lead iterates through many design-make-test cycles); compounds carry no structural features, so agents cannot transfer what they learn between similar compounds, as structure–activity reasoning does in real campaigns (we leave that for future work); LLMs run without external chemistry tools (docking, retrieval); LLM outputs are non-deterministic (temperature 1; §4) and each LLM plays each seed once, so run-to-run variance on a fixed seed is not measured and per-cell effects below $\sim$10 pp are not significant at $n = 50$; and our LLM sample covers only two proprietary model families (Claude and GPT), with no open-weight model; extending it to more models is future work. H2L-Bench measures agent reasoning, not real chemistry — deployment in drug discovery must still clear standard regulatory validation.

AI USE STATEMENT

Large language models are themselves subjects of this study: the LLM agents we evaluate (§4), their scaffolding, and the exact prompt (Appendix M) are described in full. Separately, we used generative AI tools to assist with editing the manuscript and to help with writing the code. All AI-assisted text and code were reviewed and tested by the authors, who take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

To make the benchmark reimplementable from the paper, we specify the compound traits (§3), the assays and their costs (Table 1), the compound priors and archetype thresholds (Appendix A), the noise parameters (Appendix E), VoI's posterior and update rules (Appendix F), and the complete LLM prompt (Appendix M). The LLM agents are sampled at the provider-default temperature (1 for all four models), so a rerun produces different games.

REFERENCES

Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *COLT-23th Conference on learning theory-2010*, pp. 13–p, 2010.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical science*, pp. 273–304, 1995.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, volume 2025, pp. 96934–96990, 2025.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: A platform to study the sample efficiency of grounded language learning. In *International Conference on Learning Representations*, 2019.

Kamal Choudhary, Brian DeCost, Chi Chen, Anubhav Jain, Francesca Tavazza, Ryan Cohn, Cheol Woo Park, Alok Choudhary, Ankit Agrawal, Simon JL Billinge, et al. Recent advances and applications of deep learning methods in materials science. *npj Computational Materials*, 8(1):59, 2022.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2020.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Peter I Frazier and Warren B Powell. Paradoxes in learning and the marginal value of information. *Decision Analysis*, 7(4):378–403, 2010.

Peter I. Frazier, Warren B. Powell, and Savas Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008. doi: 10.1137/070693424.

Kanishk Gandhi, Michael Y Li, Lyle Goodyear, Agam Bhatia, Louise Li, Aditi Bhaskar, Mohammed Zaman, and Noah D Goodman. Boxinggym: Benchmarking progress in automated experimental design and model discovery. *arXiv preprint arXiv:2501.01540*, 2025.

Michael A. Gelbart, Jasper Snoek, and Ryan P. Adams. Bayesian optimization with unknown constraints. In *Conference on Uncertainty in Artificial Intelligence*, 2014.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, et al. Accelerating scientific discovery with Co-Scientist. *Nature*, 2026.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 11(2):577–586, 2020. doi: 10.1039/C9SC04026A.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173, 2023. doi: 10.18653/v1/2023.emnlp-main.507.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Ronald A Howard. Information value theory. *IEEE Transactions on systems science and cybernetics*, 2(1):22–26, 1966.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *Advances in neural information processing systems*, 37:10088–10116, 2024.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, pp. 54107–54157, 2024.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998. doi: 10.1023/A:1008306431147.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Kirthevasan Kandasamy, Gautam Dasarathy, Junier B. Oliva, Jeff Schneider, and Barnabás Póczos. Gaussian process bandit optimisation with multi-fidelity evaluations. In *Advances in Neural Information Processing Systems*, 2016.

Akshay Krishnamurthy, Keegan Harris, Dylan J Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? *Advances in Neural Information Processing Systems*, 37:120124–120158, 2024.

Jon M Laurent, Joseph D Janizek, Michael Ruzo, Michaela M Hinks, Michael J Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D White, and Samuel G Rodriques. Lab-bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024.

Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

Amelia Liu, Andrew Ho, Anne Marie Droste, David Martin, Edmund Wong, Edward Zhou, Isabelle Zhou, Joshua Park, Joy Jiao, Katie-Rose Skelly, et al. LifeSciBench: Evaluating language models on realistic, expert-level tasks in the life sciences. *bioRxiv*, 2026.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Ludovico Mitchener, Jon M Laurent, Alex Andonian, Benjamin Tenmann, Siddharth Narayanan, Geemi P Wellawatte, Andrew White, Lorenzo Sani, and Samuel G Rodriques. Bixbench: a comprehensive benchmark for llm-based agents in computational biology. *arXiv preprint arXiv:2503.00096*, 2025.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Siba Smarak Panigrahi, Jovana Videnović, and Maria Brbic. Heurekabench: A benchmarking framework for ai co-scientist. In *International Conference on Learning Representations*, volume 2026, pp. 11029–11061, 2026.

Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. Modern bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024.

Daniel Reker and Gisbert Schneider. Active-learning strategies in computer-assisted drug discovery. *Drug discovery today*, 20(4):458–465, 2015.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2012.

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

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, et al. PaperBench: Evaluating AI's ability to replicate AI research. In *International Conference on Machine Learning*, 2025.

Nathan J Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E Kumar, Tanjin He, David Milsted, Matthew J McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, et al. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*, 624(7990):86–91, 2023.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3–4):285–294, 1933. doi: 10.1093/biomet/25.3-4.285.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.

Andrew R Willan and Eleanor M Pinto. The value of information and optimal clinical trial design. *Statistics in medicine*, 24(12):1791–1806, 2005.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. In *International Conference on Learning Representations*, 2025.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *International Conference on Machine Learning*, 2024a.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, pp. 15585–15606, 2024b.

A Programme archetypes: full specification

Table 4 lists the hard thresholds and off-target pool for each archetype. On medium and hard, all archetypes draw every trait from Beta$(3, 3)$ except for the ppi-oncology overrides in the caption. On easy, potency, selectivity, metabolic stability, oral bioavailability and half-life are drawn from Beta$(5, 2)$ instead, so easy libraries contain more strong compounds. All archetypes share the same observation models, and use the base per-difficulty artifact rate (Appendix E) except ppi-oncology, whose aggregator artifacts are twice as common ($2\times$ multiplier).

Table 4: Archetype hard thresholds. Traits not listed default to easy floors of 0.2. PPI-oncology additionally uses weaker priors on potency (Beta$(2, 5)$) and permeability (Beta$(2, 4)$), reflecting the harder PPI chemotype.

Archetype	Hard thresholds	Off-target pool
kinase-oncology	potency ≥ 0.80, selectivity ≥ 0.70, cardiotox ≥ 0.60	kinase_X/Y, hERG_lite
antibiotic-bacterial	potency ≥ 0.75, selectivity ≥ 0.75, metabolic_stability ≥ 0.65	host_ortholog, cyp3a4
gpcr-cns	selectivity ≥ 0.70, permeability ≥ 0.65, half_life ≥ 0.65	GPCR_A/B/C
ppi-oncology	potency ≥ 0.55, selectivity ≥ 0.65, permeability ≥ 0.60	hERG_lite
protease-chronic	selectivity ≥ 0.70, cardiotox ≥ 0.70, half_life ≥ 0.65	protease_A/B, cyp3a4

B Library feasibility per scenario

For each (archetype, tier) cell we count how many of the 24 library compounds actually clear every hard threshold and every easy floor — i.e., how many "true" winners exist for the agent to find. Mean over 50 seeds, min–max in parentheses.

Table 5: True winners per 24-compound library: compounds that clear every hard threshold and every easy floor. Mean over 50 seeds, min–max in parentheses.

Archetype	easy	medium	hard
antibiotic-bacterial	2.7 (1–6)	1.0 (1–1)	1.0 (1–1)
gpcr-cns	2.0 (1–5)	1.0 (1–1)	1.0 (1–1)
kinase-oncology	1.7 (1–4)	1.0 (1–1)	1.0 (1–1)
ppi-oncology	1.0 (1–2)	1.0 (1–1)	1.0 (1–1)
protease-chronic	1.8 (1–5)	1.0 (1–1)	1.0 (1–1)

No winner is injected. The generator draws the library at random and, while it contains no winner, replaces a randomly chosen compound with a fresh draw, stopping as soon as one winner exists. On medium and hard this replacement was needed in 237 of 250 scenarios, and every library ends with

exactly one winner, so the task is to find one winner among 24 compounds under noisy assays. On easy (83 of 250 scenarios needed replacement), most cells have 1–3 winners, occasionally more.

Information ceiling under noise. The feasibility count above tells us how many compounds *are* winners on ground truth. It does not tell us how many the agent could reliably *identify* given the actual assay noise. We measure the latter with an off-budget *informative oracle*: it runs every assay once on every compound, including the in-silico prediction and the screen (bypassing the budget), draws each readout from the observation model VoI uses rather than through the game engine, aggregates the noisy observations into a per-compound posterior using the same Bayesian machinery as the VoI agent, and nominates $\arg\max_c P(\text{win} \mid c)$. Win rates over 50 seeds per cell:

Table 6: Informative-oracle win rate (every assay run once on every compound, budget bypassed), 50 seeds per cell.

Archetype	easy	medium	hard
antibiotic-bacterial	98%	92%	82%
gpcr-cns	76%	64%	56%
kinase-oncology	98%	84%	86%
ppi-oncology	90%	96%	92%
protease-chronic	82%	82%	64%
tier aggregate	89%	84%	76%

This is the noise-limited ceiling: even given every noisy readout in the environment, a Bayesian aggregator misses 11–24% of scenarios per tier because the observations are consistent with multiple candidate winners. Comparing to the main-body VoI numbers (87 / 59 / 43% aggregate), our best budgeted agent has 2 pp of headroom on easy but 25 and 33 pp on medium and hard. This falsifies one alternative interpretation of the low win rates on gpcr-hard (VoI 16%, Opus 5 22%): the informative oracle achieves 56% on that cell, so the low rates are not a feasibility floor — it is a shared *failure* of budgeted agents to approach a substantially higher achievable rate.

C PAIRED SIGNIFICANCE TESTS FOR THE MAIN ROSTER

Table 8 lists all 45 paired tests behind the "Reading the aggregates" paragraph. For each pair (A vs B) and tier we pool the 250 same-seed games, count the discordant pairs (A wins and B loses, b; B wins and A loses, c), and apply McNemar's exact (binomial) test to b vs c. Each p is Holm-corrected over the 15 tests within the tier (Holm, 1979). Win rates per tier are in Table 2. Every agent has 50 complete games in every cell, so each tier has 250 pairs.

Table 7 reports the mean of the fractional score that the LLM prompt describes (§4) over the same 250 games per tier.

Table 7: Mean fractional score per tier (0 if any easy floor fails, else the fraction of hard thresholds cleared; 1 = win).

Agent	easy	medium	hard
greedy	0.60	0.29	0.23
Haiku 4.5	0.65	0.45	0.35
Sonnet 5	0.81	0.59	0.48
Opus 5	0.91	0.68	0.57
GPT-5.4	0.85	0.68	0.47
VoI	0.93	0.71	0.55

Table 8: Paired exact McNemar tests, main roster. b/c: discordant pairs (row's first agent wins only / second agent wins only). p: two-sided exact p, Holm-corrected over the 15 tests within the tier.

	easy		medium		hard	
Pair	b/c	p	b/c	p	b/c	p
greedy vs Haiku 4.5	16/40	0.007	15/51	<0.001	9/36	<0.001
greedy vs Sonnet 5	12/107	<0.001	17/81	<0.001	8/64	<0.001
greedy vs Opus 5	4/137	<0.001	12/96	<0.001	9/84	<0.001
greedy vs GPT-5.4	2/110	<0.001	11/103	<0.001	11/68	<0.001
greedy vs VoI	5/154	<0.001	7/111	<0.001	8/90	<0.001
Haiku 4.5 vs Sonnet 5	15/86	<0.001	30/58	0.022	24/53	0.009
Haiku 4.5 vs Opus 5	6/115	<0.001	21/69	<0.001	27/75	<0.001
Haiku 4.5 vs GPT-5.4	7/91	<0.001	20/76	<0.001	25/55	0.008
Haiku 4.5 vs VoI	5/130	<0.001	16/84	<0.001	20/75	<0.001
Sonnet 5 vs Opus 5	17/55	<0.001	32/52	0.123	36/55	0.234
Sonnet 5 vs GPT-5.4	20/33	0.098	33/61	0.025	42/43	1.000
Sonnet 5 vs VoI	10/64	<0.001	35/75	0.001	37/63	0.060
Opus 5 vs GPT-5.4	49/24	0.014	39/47	0.460	52/34	0.234
Opus 5 vs VoI	18/34	0.073	29/49	0.123	46/53	1.000
GPT-5.4 vs VoI	16/57	<0.001	36/48	0.460	31/56	0.058

D ADDITIONAL AGENTS

D.1 THOMPSON SAMPLING

Why Thompson sampling. Standard Bayesian-optimisation acquisitions (GP-UCB, expected improvement) rely on a surrogate that shares information between candidates through a kernel on their features. H2L-Bench compounds carry no structural features: the agent sees compound IDs and assay readouts only. Without a kernel the surrogate reduces to independent per-compound posteriors, which VoI already maintains. Thompson sampling on that posterior is therefore the BO-style policy we can build without inventing features. It uses the same model, observations and nomination rule as VoI and differs only in how it picks the next action.

Algorithm. Each turn TS (1) draws one full latent state per compound from its posterior, marginalising over `is_artifact`; (2) ranks compounds by the draw, first those that clear every threshold, then by the number of hard thresholds cleared, with ties broken at random; (3) on the top-ranked compound, orders its hard traits by posterior $P(\text{clear})$, lowest first, and runs the cheapest affordable un-run assay whose readout depends on that trait (potency: `predict_in_silico`, `dose_response`, `cell_potency`, `orthogonal_confirm`; selectivity: `selectivity_panel`; ADMET traits: `admet_panel`, and `cell_potency` for permeability; PK traits: `in_vivo_pk`), moving down the ranking if that compound has none left; and (4) nominates with VoI's rule once no compound has such an assay left. TS has no tuned parameter and does not simulate assay outcomes, so a turn costs about as much computation as a greedy turn. It plays the same 750 seeded games as the roster.

Results. Tables 9–11. TS beats greedy on every tier. On easy, where the budget is large enough to measure most candidates, it also beats Haiku 4.5, Sonnet 5 and GPT-5.4 ($p \leq 0.004$) and is level with Opus 5 and VoI ($p = 0.71$ and 0.24). On medium it beats Haiku 4.5 ($p = 0.038$), is level with Sonnet 5 ($p = 0.69$), and is below Opus 5, GPT-5.4 and VoI ($p \leq 0.032$). On hard it is level with Haiku 4.5 ($p = 0.82$) and below the other three LLMs and VoI ($p \leq 0.018$). The gap to VoI opens as the budget tightens, and it lies in where the budget goes. We call the expensive per-compound wet-lab assays (`dose_response`, `orthogonal_confirm`, `cell_potency`, `selectivity_panel`, `admet_panel`, `in_vivo_pk`) *confirmatory*, and measure how much a game commits to one compound in two ways, as in §5.3: the fraction of confirmatory spend that goes to the single most-assayed compound, and the number of compounds that receive any confirmatory assay. On the hard tier, TS spreads its confirmatory spend over slightly more compounds than VoI (4.9 vs. 4.5 per game) and puts a smaller fraction on its top compound (0.37 vs. 0.47). The draw selects a compound that might be the winner, not the measurement that would most change the nomination, and at the start of a game, when all posteriors are equal, the draw picks compounds at random. TS does worst on gpcr-cns, where it has its lowest medium and hard win rates (12% and 4%; 64% on easy). Two of its three hard traits, selectivity and half-life, are read only by the $5k selectivity panel and the $8k PK study. On gpcr-cns hard, all 50 games open with a selectivity panel on an unscreened compound, 39 of them spend the whole $20k budget on four panels, and 91% of the budget goes to that assay.

Robustness to the assay rule. Step (3) is the one design choice. Replacing it with "the cheapest un-run assay that reads any hard trait of the top-ranked compound" gives 74/41/25%. Like the default rule, this variant is above Haiku 4.5 on medium ($p = 0.047$), level with it on hard ($p = 0.26$), and below VoI on medium and hard ($p < 0.001$; p corrected as in Table 10). Unlike the default rule, it is also below VoI on easy ($p < 0.001$), and it spreads its spend more thinly still (0.32 over 7.4 compounds). The conclusion does not depend on the rule: with either rule, TS falls well short of VoI once the budget cannot measure every candidate.

Table 9: TS against VoI and the base Sonnet 5 agent. Win rate per tier, and two measures of commitment on the hard tier, averaged over the 250 hard games. *share*: fraction of the game's spend on confirmatory (expensive per-compound wet-lab) assays that goes to the single most-assayed compound. *cpds*: number of compounds that receive at least one confirmatory assay. Higher share and fewer cpds mean more commitment. TS-cheapest is the robustness variant.

	win rate (%)			hard commitment	
Agent	easy	med	hard	share	cpds
TS	82	41	22	0.37	4.9
TS-cheapest	74	41	25	0.32	7.4
Sonnet 5	65	43	32	0.58	3.4
VoI	87	59	43	0.47	4.5

Table 10: Paired exact McNemar tests, TS against each roster agent, as in Table 8. b: TS wins only; c: the other agent wins only. p: two-sided exact p, Holm-corrected over the 6 tests within each tier.

	easy		medium		hard	
TS vs	b/c	p	b/c	p	b/c	p
greedy	145/8	<0.001	82/23	<0.001	49/19	0.001
Haiku 4.5	121/8	<0.001	56/33	0.038	42/39	0.824
Sonnet 5	62/20	<0.001	49/54	0.694	33/59	0.018
Opus 5	34/30	0.708	32/57	0.032	29/74	<0.001
GPT-5.4	53/24	0.004	34/67	0.005	32/59	0.018
VoI	19/31	0.238	24/69	<0.001	27/79	<0.001

Table 11: TS win rate (%) per cell, for both assay rules.

	TS			TS-cheapest		
Archetype	easy	med	hard	easy	med	hard
antibiotic-bacterial	88	60	38	66	50	34
gpcr-cns	64	12	4	60	12	8
kinase-oncology	96	52	18	84	54	28
ppi-oncology	100	66	32	98	72	46
protease-chronic	62	16	18	62	18	10

D.2 A FROM-SCRATCH RL BASELINE: PPO

To complement the reasoning-based (LLM) and decision-theoretic (VoI, greedy) agents with a *learned* policy, we trained a proximal-policy-optimisation (PPO) agent directly on the environment. We summarise the setup and results here; the qualitative trajectory analysis is the most informative part.

Setup. The policy is a shared per-compound actor–critic. It maps the native public state into a $1{,}398$-dimensional observation (24 ordered compound rows of 57 features plus a 30-feature global block; hidden latents and the true winner set are excluded) and passes each row, concatenated with the global block, through the same $87 \to 256 \to 256$ encoder, so evidence learned at one compound position transfers to every other. Shared heads score all 24 compounds, giving a masked categorical policy over 195 discrete actions (7 assays $\times$ 24 compounds, 3 fixed eight-compound screen batches,

and 24 nominations), for 156,682 trainable parameters. Training uses clipped PPO (learning rate 3×10^{-4}, clip 0.2, four update epochs, minibatches of 64, value coefficient 0.5, entropy coefficient 0.01, gradient-norm cap 0.5, $\gamma = 1$, GAE $\lambda = 0.95$) with action masking for affordability, K=1 per-pair use, and horizon. Intermediate assay actions receive zero reward; at nomination the training reward is the native fractional score (zero if any easy floor fails, otherwise the fraction of hard thresholds cleared). Three independent models are trained, one per difficulty tier, each on 100,000 episodes (1,000 iterations $\times$ 100 balanced episodes, 20 per archetype) using training seeds 100,000–199,999. Evaluation is deterministic masked-argmax on held-out seeds 0–49 across all five archetypes (250 games per tier), disjoint from training.

Aggregate results. Binary win rate decreases monotonically across tiers, from 48.0% on easy to 41.2% on medium and 33.6% on hard, as does mean fractional score. Budget use is not monotone: medium spends the largest fraction of its budget (81%), whereas the hard policy spends only about half (52%).

Table 12: PPO (argmax policy) aggregate results, 250 games per tier.

Tier	Win rate	Wins	Mean score	Mean turns	Mean spend
easy	48.0%	120/250	0.7133	24.52	$44,885
medium	41.2%	103/250	0.5547	23.60	$28,347
hard	33.6%	84/250	0.4947	21.69	$10,409

Placed against the paper's agents (win rates are descriptive, not a paired comparison — see caveats), PPO beats the greedy baseline and Haiku 4.5 at every tier; on easy and medium it sits at roughly Haiku-to-Sonnet level, but on the hard tier, where budgets bind, it is competitive with the strongest LLMs and trails only Opus 5 and the VoI reference:

Table 13: PPO win rate against the main-roster agents (reported numbers from Table 2; not a paired comparison).

Method	easy	medium	hard
greedy (reported)	27%	18%	10%
Haiku 4.5 (reported)	37%	32%	21%
PPO (argmax)	**48%**	**41%**	**34%**
Sonnet 5 (reported)	65%	43%	32%
Opus 5 (reported)	80%	51%	40%
GPT-5.4 (reported)	70%	54%	33%
VoI (reference)	87%	59%	43%

Per archetype (win % / mean score), PPI-oncology is strongest in every tier (98/0.99, 70/0.75, 56/0.65 from easy to hard); antibiotic-bacterial is weakest on easy (22%), while protease-chronic is weakest on medium and hard (12% and 14%) and hard GPCR-CNS is similarly hard (16%). Individual archetypes are not monotone across tiers: the medium and hard policies both exceed easy on antibiotic-bacterial and kinase-oncology.

PPO learns real structure, but coverage and completeness stay decoupled. Analysing the 750 evaluation transcripts (with compound and batch identities removed, and consecutive repeats of one action type collapsed into a phase) shows PPO does learn coherent, archetype-specific assay templates rather than one generic action mixture — it recognises the archetype, triages economically, and selects plausible assay families. What it does not reliably learn is a *binary-win-optimal* policy: because the training reward gives fractional credit for clearing only one or two of the three hard axes, incomplete templates still receive a strong learning signal. The sharpest symptom is that direct *breadth* (how many compounds get any wet-lab evidence) and *completeness* (whether any one candidate gets an all-hard-axis profile) stay decoupled (Table 14). On easy, several archetypes profile 13–21 compounds and usually complete a profile; but on the hard tier essentially no archetype completes an all-hard-axis profile of any candidate (0%, except PPI at 8%), even when many compounds are touched. The clearest broad-but-single-family failure is hard protease-chronic, whose modal trajectory spends the entire $20k budget on ten ADMET panels (Ax10–nominate, 100% support) while never running

selectivity or PK; hard GPCR-CNS collapses to a P-A-nominate prefix (78%) that likewise omits the required axes.

Table 14: Candidate coverage over 50 episodes per cell (from the PPO report). Entries are mean directly-profiled compounds / fraction of episodes with at least one complete all-hard-axis profile. A compound counts as directly profiled only under dose-response, orthogonal, cell-potency, selectivity, ADMET, or PK testing (not prediction or batch screen).

Archetype	easy	medium	hard
antibiotic-bacterial	1.98 / 100%	5.14 / 98%	1.26 / 0%
gpcr-cns	18.10 / 94%	12.74 / 4%	3.26 / 0%
kinase-oncology	12.84 / 100%	6.76 / 96%	1.46 / 0%
ppi-oncology	4.84 / 100%	2.64 / 100%	2.50 / 8%
protease-chronic	20.76 / 92%	5.32 / 0%	10.00 / 0%

This mirrors the allocation failures we see elsewhere: like the scaffolds that displace adaptive commitment (§5.3) and VoI's per-dollar scoring rule (Appendix I), the hard part is not executing assays but collecting complementary evidence on enough distinct candidates and knowing when to stop — PPI-oncology retains 42–52% of its budget across tiers yet remains the strongest archetype, while the weakest cells exhaust their budgets on incomplete templates.

Caveats. These numbers are descriptive, not a controlled comparison. PPO acts through a fixed 195-action encoded interface, whereas every other agent uses the native turn-by-turn interface, so the win rates are not paired and PPO is excluded from Table 2. Also, PPO is trained and reported on the native *fractional* score, whereas the paper's headline metric is binary `is_winner`; the win-count columns above are binary and comparable, but the mean-score column is not.

E ASSAY NOISE MODELS

Assay observations depend on the compound's hidden latent traits plus Gaussian noise on the appropriate scale. Table 15 lists the noise parameters per difficulty tier.

Table 15: Per-difficulty noise parameters. Silico noise is logit-scale σ; ADMET is log-scale; dose-response is $\log_{10}$-scale and PK natural-log-scale σ. $P(\text{artifact})$ is per-compound; the artifact bias enters silico as an additive logit shift and dose_response as an additive shift in $\log_{10}$ IC_{50}. In the screen, a compound's hit probability is a sigmoid in potency plus the false-positive floor, plus the artifact bump if it is an artifact.

Parameter	Easy	Medium	Hard
Budget	\$60,000	\$35,000	\$20,000
Silico logit noise	0.15	0.60	0.80
Dose-response noise	0.03	0.18	0.26
ADMET log-scale noise	0.06	0.22	0.30
PK log-scale noise	0.12	0.35	0.45
$P(\text{artifact})$	0.05	0.12	0.20
Silico artifact bias	0.4	0.6	0.9
Dose-response artifact bias	-0.2	-0.4	-0.5
Screen false-positive floor	0.03	0.08	0.15
Screen artifact bump	0.15	0.30	0.40

F VOI AGENT: POSTERIOR REPRESENTATION

Each compound's posterior is a mixture over the binary `is_artifact` latent, with each branch a factored Gaussian approximation over the nine continuous latents. Four correlated ADMET traits (lipophilicity, solubility, permeability, cardiotox_safety) live in a 4-D multivariate Gaussian — capturing the "lipophilicity → (perm, cardiotox, solubility)" coupling built into the generator. The other five traits (potency, selectivity, metabolic_stability, oral_bioavailability, half_life) are

represented as independent 1-D Gaussians. Prior parameters are moment-matched to the generator's Beta(α, β) distributions by matching mean and variance on the logit scale.

The eight assays use three kinds of update. The potency readout of `predict_in_silico` is linear-Gaussian on the logit scale, so its update is an exact Kalman step (VoI ignores that assay's coarse ADMET summary). The IC_{50}-type readouts (`dose_response`, `orthogonal_confirm`, `selectivity_panel`) use an extended-Kalman (linearised) update, with a moment-matched Gaussian projection for the scalar tobit update on right-censored IC_{50}s. The Bernoulli screen uses 1-D grid integration. The remaining three (`in_vivo_pk`, `cell_potency`, `admet_panel`) have observation models nonlinear enough that extended-Kalman linearisation introduced a persistent martingale-violating bias in the posterior over $P(\text{win} \mid c)$ under imagined observations. We replaced these with exact 1-D or 2-D numerical grid integration, using the same pattern as the Bernoulli update. After every update the posterior is moment-matched back to a Gaussian. This makes the posterior update honest at the cost of a $\sim 5\times$ slowdown per affected update.

Empirical bias check. We measured the martingale violation directly. For each of $N = 40$ random posterior states (sampled by generating a scenario, applying 0–3 random assay observations to warm the prior, then querying), we drew $K = 200$ imagined observations from the posterior predictive and applied both the naive EKF update and the grid update to each. Under a correct Bayesian update, the mean of $P(\text{win} \mid c, \text{after observing } o)$ across imagined o equals $P(\text{win} \mid c)$ before the observation (martingale property). Any systematic deviation is bias.

Table 16: Martingale check for the three nonlinear assays, grid vs. EKF update: mean signed bias in $P(\text{win} \mid c)$ across imagined observations, and fraction of states with negative computed VoI.

Assay	method	mean signed bias	frac. of states with VoI < 0
`in_vivo_pk`	grid	$+0.0003$	8%
	EKF	-0.0013	50%
`admet_panel`	grid	-0.0006	20%
	EKF	-0.0071	68%
`cell_potency`	grid	$+0.0004$	20%
	EKF	$+0.0009$	5%

On `in_vivo_pk` and `admet_panel` — both log-sigmoid observation models — EKF's linearisation biases the posterior update so much that half to two-thirds of states produce strictly-negative VoI estimates for further measurement, which is impossible under a correct posterior. Grid integration stays close to unbiased. On `cell_potency` the 2-D linearisation happens to be innocuous and either method works; we still use grid for consistency.

G VoI APPROXIMATIONS: WHAT MAKES IT A REFERENCE, NOT AN UPPER BOUND

Two approximations mean the VoI agent is only approximately optimal. *Depth-1 myopia*: VoI picks the single best next action, not the best action sequence. We tested this: an approximate two-step variant wins fewer games than depth-1 on fresh seeds (Appendix I). *Composite screening*: batched `single_shot_screen` is scored as a top-k composite rather than jointly optimised over all subsets of the library; we did not test an exact alternative. The selectivity panel also reports off-target IC50s. These are drawn independently of every scored trait and do not enter the score, so they carry no information about whether a compound wins, and VoI ignores them without loss.

VoI spends its entire budget by construction. A consequence of the per-dollar ranking is that VoI never nominates while any affordable, un-run $(assay, compound)$ pair remains: when no action clears the (near-zero) positive-VoI floor, it falls back to the highest-VoI-per-dollar action of *any* sign — the Bayesian martingale guarantees true VoI is non-negative, so tiny-negative estimates are Monte-Carlo noise, not evidence that measuring is value-destroying — and it nominates only once the cheapest un-run assay exceeds the remaining budget. We verified this empirically: across 350 games spanning budgets from \$5k to \$200k (kinase-oncology medium, 7 budgets $\times$ 50 seeds), VoI

nominated with an affordable un-run action still available in *0 of 350* games. The leftover budget at nomination was always smaller than the cheapest un-run assay. VoI's failures are therefore pure *misallocation* (spending on the wrong assays; Figure 3), never under-spending. This is why "the LLM wins by saving budget or stopping early" cannot be the mechanism, and why a policy that merely spends *more* of the budget cannot help — there is no budget to save.

H RULING OUT OTHER EXPLANATIONS FOR VOI'S LOSSES

Not noise. Were VoI's losses noise, seeds where only one of the two LLMs won should look more random than seeds where both did. They do not: the failure profile is invariant to agreement (coverage 73% vs 74%, marginal-winner 76% vs 73%, for both-won [$n = 41$] vs one-won [$n = 93$]). The losses are systematic consequences of the depth-1 per-dollar rule, not variance. *Not the posterior.* VoI's Gaussian-on-logit approximation (Appendix F) tracks the true Beta generative prior closely and to the same degree across tiers (checked on kinase-oncology), and at nomination it costs 3 of 50 hard games, so it does not drive the hard-tier gap (Appendix J). *Not lookahead depth.* These commitments look like they call for lookahead, so we re-ran VoI with an approximate two-step scheme: for each first action it imagines outcomes from the posterior predictive, picks the best follow-up for each outcome, and ranks first actions by their two-step value of information per dollar, the depth-1 objective extended by one step. On 30 fresh seeds for each of the two cells in §5.2 (ppi-hard and antibiotic-medium), depth-2 wins fewer games than depth-1 on both (Table 17). Pooled over both cells, depth-1 wins 17 seeds that depth-2 loses and depth-2 wins only 5 that depth-1 loses (paired exact McNemar $p = 0.017$). The extra step makes VoI worse, so the failure is structural to the per-dollar rule, not fixable by one more step of lookahead. Appendix I gives details, including an earlier divergent-seed exploration that suggested partial recovery but was selection-biased.

Table 17: Depth-1 vs. approximate depth-2 VoI on 30 fresh seeds per cell (games won).

Cell	depth-1	depth-2
ppi-hard	22/30	17/30
antibiotic-medium	25/30	18/30

I DEPTH-2 LOOKAHEAD: EXPLORATION AND FRESH-SEED EVALUATION

Motivated by the two mechanistic patterns above, we built a 2-step lookahead variant of VoI. For each candidate first action a_1, it samples imagined outcomes o_1 from the posterior predictive; for each o_1 it identifies the best follow-up a_2; and it ranks each a_1 by its *marginal two-step value of information per dollar*:

$$\text{score}(a_1) \;=\; \frac{\mathbb{E}_{o_1}\big[\max_{a_2} \mathbb{E}_{o_2}[\max_c P(\text{win})]\big] \;-\; \max_c P(\text{win})}{\text{cost}(a_1)},$$

the expected best terminal win probability after the imagined (a_1, o_1, a_2, o_2) trajectory, minus the current best win probability, per dollar. This is the two-step generalisation of the depth-1 rule — both score an action by the marginal gain in win probability it buys per dollar, so depth-1 and depth-2 differ only in horizon, not in objective. The denominator is $\text{cost}(a_1)$ alone, matching depth-1's per-action normalisation: the follow-up a_2 must be affordable after a_1 but its cost does not enter the price, so depth-2 is not penalised for the second step it anticipates. To keep runtime tractable we prune both the a_1 and a_2 candidate sets by depth-1 VoI rank (the union of the top five by VoI per dollar and the top five by raw VoI), and exclude the batch screen from both; an exhaustive enumeration is infeasible at typical turn counts. Unlike depth-1, depth-2 has no fallback that spends leftover budget: it nominates as soon as no first action has positive two-step value. The expectations are Monte Carlo estimates (16 imagined outcomes per first action), so the scheme is an approximation to exact two-step lookahead.

Initial exploration on divergent seeds. We first evaluated depth-2 on the divergent seeds of the two cells featured in §5.2, where an LLM won and depth-1 VoI lost (9 antibiotic-medium seeds and 14 ppi-hard seeds). Depth-2 won 3 of the 9 and 8 of the 14; depth-1, rerun on the same seeds, won 1 of each. This was suggestive but selection-biased: the seeds were chosen conditional on depth-1 loss, so any alternative agent is guaranteed to appear better in expectation via regression to the mean.

Fresh-seed evaluation. We drew a held-out sample of 30 new seeds per cell (seeds 100–129, disjoint from the paper's main 0–49) and ran depth-1 and depth-2 in a paired same-seed comparison on the same two cells (ppi-hard, Opus 5; antibiotic-medium, GPT-5.4). Depth-2 does not improve on depth-1: it wins 17/30 against depth-1's 22/30 on ppi-hard, with 7 depth-1-only and 2 depth-2-only pairs, and 18/30 against 25/30 on antibiotic-medium, with 10 depth-1-only and 3 depth-2-only pairs. Pooled across the two cells, depth-2 loses the discordant pairs 5 to 17 (McNemar's exact $p = 0.017$). The earlier partial-recovery result was regression to the mean.

We also verified that the "silico dominance" pattern (VoI repeatedly buying cheap in-silico predictions; §5.2) is not a Monte-Carlo-noise artefact of the per-dollar ranker: rerunning seed 7 on antibiotic-hard at $N \in \{256, 1024, 4096\}$ MC samples per action gives essentially the same action distribution (24 of 33–35 turns spent on `predict_in_silico` at every N).

J Is the Gaussian posterior approximation to blame?

VoI tracks each trait as logit(trait) $\sim \mathcal{N}(\mu, \sigma^2)$, whereas the generative model draws traits from a Beta prior. A natural hypothesis for VoI's hard-tier losses is that this Gaussian-on-logit approximation misestimates $P(\text{win})$ in the tail, where hard-tier winners live. We test it directly. For 200 sampled scenarios per tier on kinase-oncology we compute, for every compound, $P(\text{win})$ under (a) the true Beta prior and (b) VoI's Gaussian-on-logit representation at the prior, and compare both the per-trait $P(\text{trait} \geq \text{threshold})$ divergence and the downstream compound selection after in-silico screening.

The approximation error is small and, crucially, does not grow with difficulty. Per-trait $P(\text{trait} \geq \text{threshold})$ differences peak at ± 0.096 (on permeability, cardiotoxicity safety, and solubility — the traits coupled to lipophilicity, where the transformed Gaussian is least Beta-like) and are nearly identical across easy, medium, and hard (Figure 6, left). Downstream the effect is negligible: the fraction of scenarios whose top-ranked compound after in-silico screening is a true winner (or an artifact) is the same under the Gaussian and the Beta posterior at every tier, to within 2 pp on easy and < 1 pp on medium and hard (Figure 6, right).

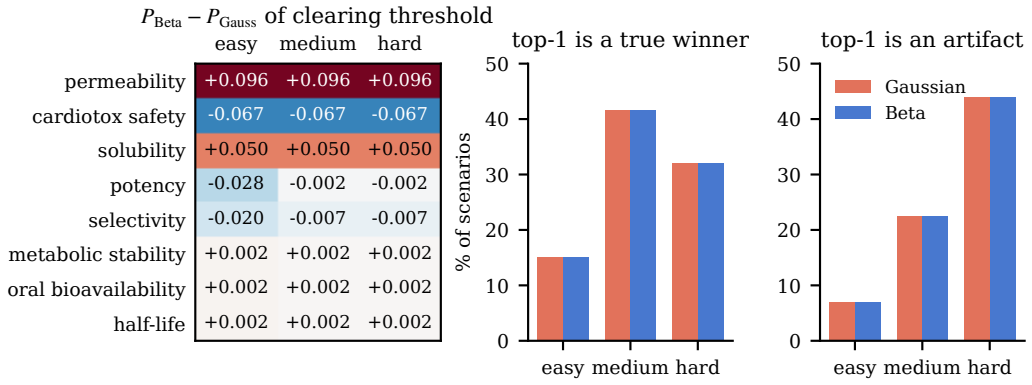


Figure 6: *Left:* per-trait divergence in $P(\text{trait} \geq \text{threshold})$ between the true Beta prior and VoI's Gaussian-on-logit representation; small and tier-independent. *Middle and right:* post-screening top-1 selection quality is unchanged between the two representations at every tier. The Gaussian approximation is not the main source of the hard-tier gap.

Posterior error at nomination. The comparison above is made at the prior. We also test VoI's posterior at the moment it nominates, in real VoI games (50 per tier: 5 archetypes $\times$ 10 seeds). For every compound we compute a reference $P(\text{win})$ under the exact generative model: we draw 20,000 latent trait vectors from the true Beta priors, weight each by the exact likelihood of the measurements the compound received, and take the weighted fraction that clears every threshold. When measurements are precise, nearly all the weight falls on one sample, so we reach the posterior in 24 tempering steps, resampling and jittering the samples at each step. Table 18 compares this reference with VoI's $P(\text{win})$.

The error in VoI's $P(\text{win})$ is largest on easy (mean absolute error 0.115, up to 0.60 in single games) and an order of magnitude smaller on hard (0.011). Nomination depends only on which compound ranks first, and on easy the winners stand clear of the rest of the library, so the large easy-tier error costs no games. On hard the compounds are close together and the small error is enough to reorder them: in 3 of 50 hard games (6%) VoI nominates a non-winner that the exact model ranks below the true winner. This never happens on easy or medium. The approximation therefore costs VoI a few hard-tier wins, about 6 of the roughly 57 hard games in 100 that VoI loses.

Table 18: VoI's $P(\text{win})$ versus the exact generative-model value at nomination (5 archetypes $\times$ 10 seeds per tier). The error is on the nominated compound. "Top-1 differs": VoI's highest-$P(\text{win})$ compound is not the exact model's. "Winnable game lost": of those, games where the exact model's top compound is a true winner and VoI's is not; ranges are 95% Wilson intervals.

Tier	Games	Mean abs. error	Top-1 differs	Winnable game lost
easy	50	0.115	1 (2%)	0 (0%; 0–7%)
medium	50	0.038	4 (8%)	0 (0%; 0–7%)
hard	50	0.011	8 (16%)	3 (6%; 2–16%)

We conclude that the Gaussian posterior approximation is not the main source of VoI's hard-tier failures. This agrees with the depth-2 result (Appendix I) and points to the per-dollar scoring rule and the single-assay action space, rather than the representation of uncertainty, as the likely limit on VoI; we have not tested this attribution directly.

K TEST-TIME SCAFFOLDS: IMPLEMENTATION DETAILS

All three scaffolds run on Sonnet 5 on the hard tier (5 archetypes $\times$ 50 seeds, the same seeds as the base agent) and share its state summary, JSON action format and retry policy. Results are in §5.3 and Table 19. Each p-value there is a paired exact McNemar test against the base agent on the 250 same-seed games, Holm-corrected over the three scaffolds; the discordant counts are given below.

Table 19: Test-time scaffolds on Sonnet 5, hard tier (50 games per archetype). p: paired exact McNemar vs. base over the 250 same-seed games, Holm-corrected over the three scaffolds. Commitment: mean share of confirmatory budget on the single most-tested compound, and mean number of compounds given a confirmatory assay (Figure 4).

	base	CoT	ToT	Planner
antibiotic-bacterial	48%	30%	36%	28%
gpcr-cns	16%	22%	12%	16%
kinase-oncology	34%	42%	40%	18%
ppi-oncology	48%	54%	36%	44%
protease-chronic	16%	18%	20%	10%
hard tier	32%	33%	29%	23%
p vs. base	—	0.91	0.82	0.033
commitment	0.58	0.60	0.48	0.79
compounds confirmed	3.4	3.4	4.6	2.6

Chain-of-thought. Our CoT variant appends a "Reasoning first" block to the system prompt (Appendix M), after the base prompt's closing instruction to return a bare JSON object with no prose. The block asks the model, before emitting the JSON, to write 2–5 sentences (under 200 words) that summarise its top 2–3 candidate compounds, identify the most uncertain axis separating them, and pick the action that best resolves it given the remaining budget; it adds that prose around the JSON is accepted. The two instructions conflict, and the later one wins: every one of the 5,050 hard-tier CoT turns opens with a reasoning block. Of the discordant seeds CoT wins 44 and base 42 ($p = 0.91$).

Tree-of-thoughts. At each turn the model proposes three candidate actions with rationales, scores each valid candidate on a 0–10 scale in a separate call, and executes the highest-scored one (up to four LLM calls per turn, about $4\times$ the base call volume). A candidate whose score cannot be parsed gets 0. If no proposed candidate is valid, ToT tries to read the proposal reply as a single action, and if

that also fails it nominates a random compound, which ends the game. Of the discordant seeds ToT wins 42 and base 51 ($p = 0.82$).

Planner. At the first turn the model emits a full action plan up to nomination; the agent then plays its steps in order, skipping any that have become invalid, and asks for a new plan from the current state every five turns or when the plan runs out. Of the discordant seeds Planner wins 26 and base 49 ($p = 0.033$).

L VoI HYPERPARAMETER SENSITIVITY

We ablated the three main VoI hyperparameters one factor at a time on a 20-seed $\times$ 15-cell grid. All numbers are aggregate win rate per tier over 100 games. Baseline values (marked $*$) are the main-body VoI run restricted to seeds 0–19; they differ from Table 2 (87/59/43%), which is over 50 seeds. With 100 games per tier, differences below about 10 pp are within sampling noise.

Table 20: VoI ablation: imagined observations N per candidate. Win rate over 100 games per tier (20 seeds $\times$ 5 archetypes); $*$ marks the main-body setting.

N (imagined observations per candidate)	easy	medium	hard
16	84%	65%	40%
64	78%	63%	44%
256*	83%	66%	46%
1024	83%	55%	52%

Win rate is roughly flat in N on easy and medium, except that $N = 1024$ loses 11 pp on medium. On hard it rises with N (40, 44, 46, 52%), so no single N is best on every tier. $N = 64$ is a cheaper alternative ($\sim$4$\times$ speedup), 5 pp below baseline on easy and 2–3 pp below on medium and hard.

Table 21: VoI ablation: cost exponent α. Same grid as Table 20.

α (cost exponent in VoI/cost$^\alpha$)	easy	medium	hard
0.3	82%	45%	34%
0.7	83%	56%	42%
1.0*	83%	66%	46%
1.5	82%	63%	44%

$\alpha = 1.0$ is best on medium and hard, and all four values are within 1 pp on easy. Smaller α (0.3, 0.7) costs 10–21 pp on medium and 4–12 pp on hard because the agent spends on expensive assays too early. $\alpha = 1.5$ is within 3 pp of baseline. Lowering α makes expensive assays more attractive, which is what Strategy 2 needs (§5.2), but it lowers the win rate on medium and hard, so changing the cost exponent does not fix depth-1 VoI.

Table 22: VoI ablation: rule-out threshold `RULEOUT_REL_TO_MAX`. Same grid as Table 20.

`RULEOUT_REL_TO_MAX`	easy	medium	hard
0	86%	68%	46%
0.02*	83%	66%	46%
0.05	83%	64%	45%
0.10	78%	52%	45%

Rule-out (the relative-P(win) floor below which surviving compounds are pruned) matters little up to 0.05: thresholds of 0, 0.02 and 0.05 are within 3 pp of each other on every tier, with no rule-out nominally best on easy and medium. A threshold of 0.10 costs 5 pp on easy and 14 pp on medium. Hard is flat (45–46%) at every setting.

M LLM SYSTEM PROMPT

The system prompt used for all LLM agents is reproduced below verbatim. It is the same across Haiku, Sonnet, and Opus; the CoT variant appends the "Reasoning first" suffix that follows. The per-turn state summary (target name, hard thresholds, remaining budget, per-compound observation rollup) is provided as the user message on each turn.

```
You are an autonomous drug-discovery agent playing a hit-to-lead
optimization scenario. Each turn you receive a state dict describing the
current library, budget, thresholds, and prior observations. You must
respond with EXACTLY ONE JSON object naming your next action.

## Your goal

Pick one compound from the library that clears every scoring threshold.
You have a fixed research budget to spend on the assays below before
committing. There is NO reward for saving budget -- unspent dollars do
not contribute to your score.

## Action shapes and costs

Paid assays (single compound):

  {"type": "predict_in_silico",  "cid": "C-01"}   $50
  {"type": "dose_response",      "cid": "C-01"}   $1,500
  {"type": "orthogonal_confirm", "cid": "C-01"}   $4,000
  {"type": "cell_potency",       "cid": "C-01"}   $3,000
  {"type": "selectivity_panel",  "cid": "C-01"}   $5,000
  {"type": "admet_panel",        "cid": "C-01"}   $2,000
  {"type": "in_vivo_pk",         "cid": "C-01"}   $8,000

Paid batch action:

  {"type": "single_shot_screen", "cids": ["C-01","C-02", ...]}  $200 flat
    (batch of 1 to 10 compounds, no duplicates)

Free actions:

  {"type": "review_log"}         -- returns the full chronological log
  {"type": "query_thresholds"}   -- returns the scoring thresholds

Terminal action (free, ends the game):

  {"type": "nominate", "cid": "C-01"}

## Scoring rule

You get one shot at nomination. Your `score` is computed as follows:

1. If the nominated compound MISSES any easy floor, `score = 0.0`. Floors
   are table stakes; missing one disqualifies the compound entirely
   regardless of how well it does on hard axes.
2. Otherwise, `score = (hard thresholds cleared) / (hard thresholds total)`.

Maximum score is 1.0, achieved by clearing EVERY hard threshold AND every
easy floor. `is_winner = True` is the same condition.

Concrete example: hard thresholds = {potency >= 0.80, selectivity >= 0.70,
cardiotox_safety >= 0.60}, plus 6 easy floors at 0.20 each. A compound
with (potency=0.85, selectivity=0.72, cardiotox_safety=0.65, all floors
>= 0.20) scores 1.0 and wins. Same compound but with solubility=0.15
scores 0.0 -- the floor miss zeroes it.

## Assay mechanics

Every assay produces a noisy readout. Each (compound, paid assay) pair
may only be run ONCE per game -- running the same paid assay on the same
```

```
compound a second time raises an error. To confirm a suspicious result,
run an orthogonal assay (e.g. dose_response then orthogonal_confirm,
or cell_potency after biochemical dose_response). `predict_in_silico`
is deterministic per compound and is cached: repeat calls return the
same readout at $0 cost. What each paid assay returns:

- `predict_in_silico` ($50): a computational prediction of potency and a
  coarse ADMET summary. Both are noisy on the logit scale. Compounds
  flagged as ARTIFACT (a hidden attribute) get their potency estimate
  inflated by an additive bias here.
- `single_shot_screen` ($200/batch): a binary hit/miss for each compound
  in the batch, evaluated at 10 µM. Has a difficulty-dependent
  false-positive floor. Artifact compounds have an inflated hit
  probability.
- `dose_response` ($1,500): observed IC50 in µM, or the string ">LOQ" if
  the true IC50 exceeds 30 µM. Log-normal noise. Artifact compounds have
  their observed IC50 biased downward (they look more potent).
- `orthogonal_confirm` ($4,000): same output shape as `dose_response`,
  but the artifact bias is zeroed and the log-noise sigma is halved.
- `cell_potency` ($3,000): observed cellular EC50 in µM, or ">LOQ".
  Effective potency = potency × (0.3 + 0.7 × permeability).
- `selectivity_panel` ($5,000): observed target IC50, a numeric
  `selectivity_score` in [0, 1], and observed off-target IC50s.
- `admet_panel` ($2,000): observed solubility, permeability,
  metabolic_stability, cardiotox_safety (each in [0, 1]), plus a binary
  hERG flag that fires when observed cardiotox_safety is low.
- `in_vivo_pk` ($8,000): observed oral_bioavailability and half_life,
  each in [0, 1]. Both readouts share a common per-call random effect.

Numeric readouts on assays that can return ">LOQ" require guarding --
the value is a string in that case, not a number.

## Nomination

Nomination is one-shot: no partial-credit signal, no re-nominations.
Once you nominate, the game ends and the compound is scored. Nominating
a compound with no measurements is legal but produces `score = 0` in
almost every scenario, because clearing all thresholds by chance is
rare.

## Output format

Respond with a single JSON object. Nothing else -- no prose, no code
fences, no explanation. Just the JSON.
```

The "no partial-credit signal" clause under `## Nomination` refers to the nomination mechanic — it is one-shot and final, with no re-nominations — and to the binary `is_winner` outcome, which requires clearing *every* threshold. It does not contradict the fractional `score` defined under `## Scoring rule`: that fraction is a secondary diagnostic (hard thresholds cleared, conditional on the easy floors passing), while all headline win rates in the paper use the all-or-nothing `is_winner`.

The chain-of-thought variant (Appendix K) appends:

```
## Reasoning first

Before emitting the JSON, write a brief reasoning block (2--5 sentences,
under 200 words) that: (1) summarises the current state of your top
2--3 candidate compounds; (2) identifies the most-uncertain axis that
would separate them; (3) picks the action that best resolves that
uncertainty given remaining budget. Then emit the JSON action object.
The JSON extractor accepts prose surrounding the JSON --- just make
sure exactly one JSON object appears in your response.
```