H2L-Bench: A Benchmark for Agentic Planning Under Uncertainty in Drug Discovery

Pablo Lemos[1] **Nhung T.H. Nguyen**[1] **Kevin Kühl**[2] **Aditya Kashyap**[1]
Ahmed Elnaggar[1] **Dong Ding**[1] **Majd Mustapha**[1] **Umut Eser**[1] **Timothy Schultz**[1]
[1]Johnson & Johnson Innovative Medicine [2]School of Mathematics and Statistics, UNSW Sydney

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 clearing multiple hidden thresholds simultaneously, with costs and noise 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 actions by decision-theoretic expected information gain per dollar. Comparing VoI to a greedy baseline and three frontier Claude LLMs (Haiku 4.5, Sonnet 4.5, Opus 4.5) at 50 seeds per cell, we find a consistent ordering (greedy $<$ Haiku $<$ Sonnet $\approx$ Opus $<$ VoI) with the largest gap on easy tier (VoI 88% vs Opus 63%) and no statistically detectable difference from Sonnet on medium (54% vs 52%). Beyond the headline win rates, the environment lets us watch where LLMs and a Bayesian agent diverge action-by-action: on a subset of seeds, LLMs recover the correct compound using two-turn patterns, or taking a risky bet and committing budget to an expensive singular assay, that a depth-1 per-dollar VoI ranking cannot propose.

1 Introduction

LLM-based agents have made striking recent progress on multi-step tasks in software engineering [1], web navigation [2, 3] and workflow orchestration [4, 5]. Much of the excitement around scientific applications extrapolates these gains to open-ended discovery pipelines like drug design [6, 7, 8, 9] and materials discovery [10, 11]. But standard agent benchmarks reward completing a task correctly, while 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 discussed below in Section 2.

We introduce the **H2L-Bench**, a compact benchmark modelling the hit-to-lead phase of small-molecule drug discovery. The agent starts with a library of 24 candidate compounds and a fixed budget, and must nominate one lead that clears hidden hard thresholds (potency, selectivity, program-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 program archetypes and three difficulty tiers span realistic variation.

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Workshop: AI for Drug Discovery: Bridging the Translation Gap. Corresponding author: plemospo@its.jnj.com.

Because the environment's noise model is known, we can build a strong decision-theoretic reference policy: a Bayesian value-of-information (VoI) agent [12, 13, 14] that plans by expected reduction in nomination regret 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/cost rather than evaluating action sequences — and so it is a reference, not an upper bound; LLMs occasionally match or narrowly beat it on specific cells.

Our contributions are as follows: **(i)** a lightweight benchmark for agentic decision-making under uncertainty with realistic cost structure; **(ii)** a computationally-tractable Bayesian VoI agent as a reference; **(iii)** a five-agent comparison (greedy, Claude Haiku/Sonnet/Opus 4.5, VoI) and a consequent analysis of their performances, which reveals a concrete failure mode of depth-1 VoI that LLM agents can exploit.

2 Related Work[1]

Agentic benchmarks and evaluation: Recent LLM-agent benchmarks have focused on task completion in deterministic environments, such as WebArena [15] for web navigation, SWE-bench [16] for code generation, τ-bench [17] for tool-use in customer service. There are also general-knowledge benchmarks, such as MMLU [18], BIG-Bench [19] 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 [6] and Coscientist [7] evaluate LLM agents on real chemistry design, showing they can orchestrate real tools; ScienceAgentBench [20], HeurekaBench [21], LifeSciBench [22], BixBench [23], LAB-Bench [24] and PaperBench [25] test data-driven scientific reasoning. These benchmarks judge success by task completion or answer correctness, not by how well the agent trades observation cost against information gain. They are also not designed to have a computable optimal policy.

Sequential decision-making under uncertainty: Bayesian value-of-information dates to Howard [12] and Lindley [26], and has been applied to sequential decision-making in a variety of domains. Modern reviews [13, 14] cover Bayesian experimental design broadly. VoI has been applied to clinical-trial planning [27] and active-learning-based hit discovery [28], but not to our knowledge as a benchmark reference agent for LLM comparison.

Environments with theoretical optima: A few benchmarks admit computable optimal policies. For example, small POMDPs [29], gridworlds like BabyAI [30], 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.

Our Contribution: The H2L-Bench combines all four items: 24-compound sequential decisions, hard budget cap, eight assays with distinct cost/noise tradeoffs, and 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

A *scenario* consists of (i) a pool of $N = 24$ candidate compounds, each with 10 hidden numeric traits; (ii) a set of thresholds (a subset are "hard" and must all be cleared to win); and (iii) 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. Each (compound, wet-lab assay) pair may only be run once per scenario, mimicking the practice that medicinal chemists confirm suspicious results by orthogonal assays rather than serial replicates, and to stop agents from trivially averaging out noise by repeatedly running the same assay on the same compound.

Archetypes. We include five program archetypes with distinct hard-axis structures (*kinase-oncology*, *GPCR-CNS*, *antibiotic-bacterial*, *PPI-oncology*, and *protease-chronic*). Each specifies its own hard

[1]This field is large and growing rapidly. While we have tried to best of our knowledge to cover the most relevant work, we apologize for any omissions.

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 C).

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 score
`in_vivo_pk`	8,000	oral bioavailability + half-life

thresholds and, optionally, biased priors on selected traits (e.g. GPCR selectivity uses a lower-mean Beta prior for aminergic promiscuity). Full definitions are in Appendix A.

Difficulty tiers. Three tiers (easy, medium, hard) scale the per-assay noise, the artifact rate, and the total budget ($60k / $35k / $20k). Harder tiers keep the compound distribution unchanged; only the observation channel and budget change. Full parameters in Appendix C.

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 program. 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 Maintains point estimates (no uncertainty), applies a static heuristic (silico first, then dose-response on top scorers, then admet), and nominates the compound with highest point-estimate P(win).

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 can compute exact Bayesian updates, but it gives a principled per-cell reference against which LLM and heuristic behavior 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

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

via $N = 256$ Monte Carlo draws from the posterior predictive, each followed by the exact Bayesian update. Assay updates use closed-form Kalman when the observation model is linear-Gaussian, and 1-D or 2-D grid integration for the three assays (`in_vivo_pk`, `cell_potency`, `admet_panel` whose nonlinear observation models made extended Kalman filter (EKF) linearisation biased in practice. Actions are ranked by VoI/dollar; VoI nominates when no positive-VoI action remains in budget. Full posterior specification and derivation are in Appendix D; approximations and their consequences in Appendix E. The most consequential is depth-1 myopia: VoI picks the single best action rather than the best sequence.

LLM agents. We test three Claude models: Haiku 4.5, Sonnet 4.5, and Opus 4.5 [2]. 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. 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), and like VoI must infer program character from them. The

[2] At the time this work was started, these were the latest Claude models available

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 [4] loop at the granularity our environment supports: each atomic game step is one Thought–Action–Observation triple, and we ablate the Thought half with an explicit chain-of-thought toggle — somewhat surprisingly, enabling CoT *reduces* Sonnet's win rate by 5–13 percentage points per tier (see Appendix H). Higher-level scaffolds such as reflexion-style self-critique [31], Tree-of-Thoughts branching [32], or explicit planning [33] are left to future work.

5 Results

Each of the five agents plays 50 seeds per (archetype, difficulty) cell, totalling 750 games per agent. Table 2 reports the aggregate results per tier; Figure 1 shows the per-cell breakdown. Greedy $<$ Haiku $<$ Sonnet $\approx$ Opus $<$ VoI on every tier. To bound what is achievable under noise, we also report an *informative oracle* that runs every wet-lab assay once per compound and aggregates via the same posterior VoI uses (Appendix B).

Table 2: Aggregate win rates (250 games per tier). Oracle is the non-budgeted informative-oracle noise ceiling (Appendix B); VoI is the depth-1 Bayesian reference at fixed budget.

Tier	greedy	Haiku 4.5	Sonnet 4.5	Opus 4.5	VoI	oracle
easy	27%	37%	60%	63%	88%	**89%**
medium	18%	32%	52%	47%	54%	**84%**
hard	10%	21%	31%	29%	41%	**76%**

Reading the aggregates. Because our sweep runs every agent on the same seed within a cell, we use McNemar's exact test for paired binary outcomes [34] — the paired analogue of Fisher's exact for two-sample proportions — and report tier-level pooled results (250 seeded pairs per tier). VoI significantly beats every LLM on easy ($p < 10^{-3}$) and hard ($p < 0.01$). On medium, VoI beats Haiku ($p < 10^{-3}$) and beats Opus (52 VoI-only wins vs 31 Opus-only, $p = 0.028$) but we cannot distinguish VoI from Sonnet (52 vs 45 discordant pairs, $p = 0.54$); we report this as absence of evidence for a difference, not evidence of equivalence. Within the LLM cluster, Sonnet 4.5 beats Haiku 4.5 by 20–28 pp per tier, while Sonnet and Opus land within $\leq$3 pp of each other in aggregate despite differing per-cell (Opus stronger on kinase-easy, Sonnet on kinase-medium). See Figure 1.

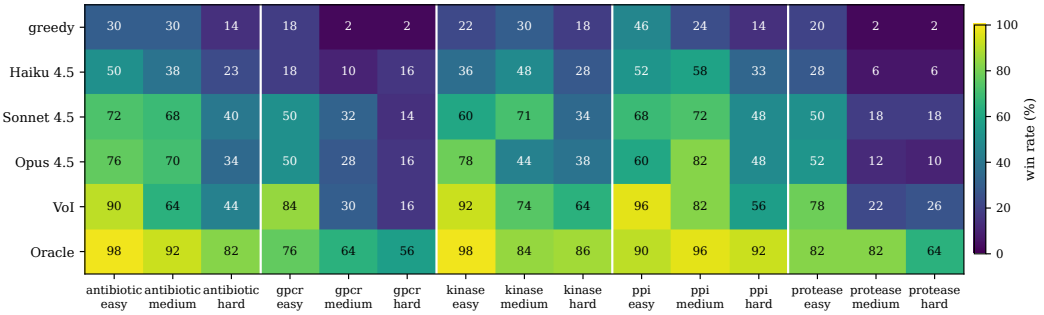


Figure 1: Per-cell win rates (%) at 50 seeds per cell. Rows are agents, columns are (archetype, tier) cells. Lighter means higher win rate. The bottom row (Oracle) shows the informative-oracle noise ceiling, a non-budgeted reference. Kinase-oncology and PPI-oncology are the cleanest cells for VoI; GPCR and protease remain hard for every agent. VoI leads the budgeted agents on most cells but sits well below the oracle on medium and hard — where a budgeted agent could still improve substantially.

Two strategies VoI misses. On two cells at the hard tier — antibiotic-hard (VoI 44% vs Sonnet 40%) and gpcr-hard (VoI 16% vs Opus 16%) — VoI and the leading LLM land within a few percentage points of each other, and on a subset of individual seeds the LLM wins games where depth-1 VoI does not. Trajectories on those divergent seeds reveal the same underlying limitation: value that only materializes across two turns, invisible to a marginal per-turn ranking.

Strategy 1 (Sonnet, antibiotic-hard) — head-to-head comparison. Sonnet's winning recipe on the seeds VoI loses: screen, then run `selectivity_panel` on *both* top candidates and pick the one

clearly above threshold. The other candidate is an artifact whose apparent potency the panel exposes. Neither panel alone is individually attractive to VoI, so it runs at most one and nominates the artifact. What Sonnet does is a two-turn commitment; what our VoI evaluates is one turn at a time.

Strategy 2 (Opus, gpcr-hard) — expensive but singular assays. On gpcr-hard three hard axes each need a different assay, and `in_vivo_pk` at $8k is 40% of the budget. Opus's winning games run `in_vivo_pk` three times as often as its losing games; VoI almost never runs it at all. In per-dollar terms it looks bad; conditional on a plan to fully profile one compound it is mandatory. Opus commits to that plan from a domain heuristic; VoI keeps deferring the assay in favour of cheaper information until budget is spent.

Both are the same shape of failure. Figure 2 shows action mixes on the antibiotic-hard divergent seeds; a detailed per-seed trajectory analysis is in Appendix F.

Does depth-2 help? We re-ran VoI with an approximate 2-step lookahead (imagined outcomes o_1 from the posterior predictive, best follow-up a_2 per outcome, rank each a_1 by $\mathbb{E}_{o_1}[\max_{a_2}$ terminal $P(\text{win})]$ per dollar). On a held-out draw of 30 fresh seeds per cell, depth-2 yields no improvement over depth-1 on antibiotic-hard (16/30 vs 16/30, paired McNemar's $p = 1.0$) and no improvement on gpcr-hard (3/30 vs 8/30, $p = 0.13$, directionally worse). Both failure modes therefore appear structural to the depth-1 per-dollar rule rather than fixable by a single extra step of lookahead. See Appendix G for details, including an earlier exploration on the divergent-seed subset that suggested partial recovery but was selection-biased.

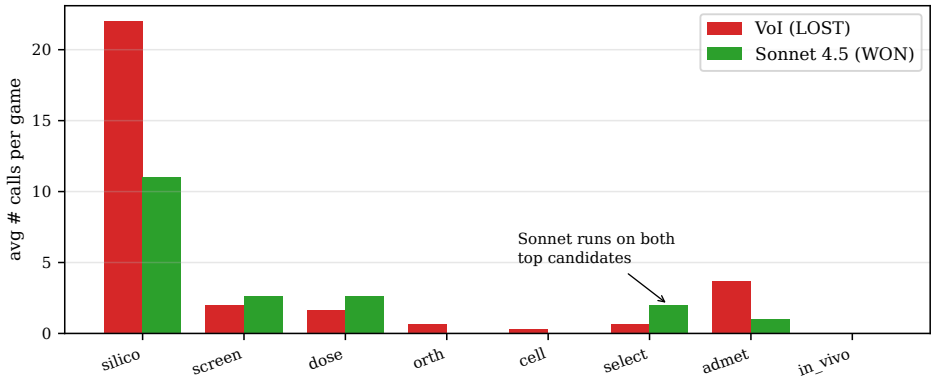


Figure 2: Action distributions on antibiotic-hard, averaged over three representative seeds (3, 7, 10) drawn from the six seeds in the original 20-seed draw where Sonnet wins but depth-1 VoI loses. Sonnet spends fewer `predict_in_silico` calls and runs `selectivity_panel` on both top candidates (2 calls); VoI runs many more silicos and ≤ 1 selectivity_panel.

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 $<$ Sonnet $\approx$ Opus $<$ VoI. Trajectory analysis on the seeds where LLMs occasionally beat VoI surfaces two two-turn patterns (head-to-head selectivity comparison; commitment to an expensive singular assay) that a depth-1 per-dollar ranking cannot propose by construction, and approximate 2-step lookahead on fresh seeds does not fix them — so the failures reflect the action space and scoring rule rather than lookahead depth. The reference is not an upper bound: an informative oracle running every assay once per compound reaches 89 / 84 / 76% per tier (Appendix B), leaving substantial headroom on medium and hard for a policy that handles joint measurement or non-per-dollar objectives.

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); LLMs run without external chemistry tools (docking, retrieval); LLM outputs are non-deterministic at temperature 1, so per-cell effects below $\sim$10 pp are not significant at $n = 50$; and our LLM sample is predominantly one family (Claude); a preliminary run of Llama 3.3 70B on two archetypes places it below Haiku 4.5 on every tier (Appendix K), consistent with a capability-scaling trend but not yet a thorough cross-family comparison. H2L-Bench measures agent reasoning, not real chemistry — deployment in drug discovery must still clear standard regulatory validation.

References

[1] 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.

[2] 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.

[3] 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.

[4] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[5] 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.

[6] 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.

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

[8] 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*, pages 1–3, 2026.

[9] 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.

[10] 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.

[11] 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.

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

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

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

[15] 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*, volume 2024, pages 15585–15606, 2024.

[16] 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*, volume 2024, pages 54107–54157, 2024.

[17] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. *arXiv preprint arXiv:2406.12045*, 2024.

[18] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[19] 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. *arXiv preprint arXiv:2206.04615*, 2022.

[20] 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, pages 96934–96990, 2025.

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

[22] 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*, pages 2026–08, 2026.

[23] 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.

[24] 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.

[25] 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. *arXiv preprint arXiv:2504.01848*, 2025.

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

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

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

[29] 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.

[30] 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. *arXiv preprint arXiv:1810.08272*, 2018.

[31] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

[32] 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, 2023.

[33] 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. *arXiv preprint arXiv:2305.16291*, 2023.

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

A Program archetypes: full specification

Table 3 lists the hard thresholds and artifact rate multiplier for each archetype. All archetypes share the same compound prior for non-hard-axis traits (Beta$(3, 3)$) and the same observation models.

Table 3: Archetype hard thresholds. Traits not listed default to easy floors of 0.2. GPCR-CNS additionally uses a Beta$(2, 3)$ prior on selectivity to model aminergic promiscuity.

Archetype	Hard thresholds	Off-target pool
kinase-oncology	potency ≥ 0.75, selectivity ≥ 0.75, cardiotox ≥ 0.60	kinase_A/B, cyp3a4
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	PPI_A/B
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.

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)

On medium and hard the scenario generator injects exactly one winning compound; the task is literally to find the needle in a 24-compound haystack under noisy assays. On easy, most cells have 1–3 winners, occasionally more. Random-nomination win rates correspond to $\sim$1/24 = 4% on medium/hard (empirically 0–4% per cell in our sweep) and $\sim$4–11% on easy.

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 wet-lab assay once on every compound (bypassing the budget), 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:

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 (88 / 54 / 41% aggregate), our best budgeted agent has 1 pp of headroom on easy but 30–35 pp on medium and hard. This falsifies one alternative interpretation of the paper's near-ties on gpcr-hard (16% for both VoI and Opus): the informative oracle achieves 56% on that cell, so the near-tie is not a feasibility floor — it is a shared *failure* of budgeted agents to approach a substantially higher achievable rate.

C Assay noise models

Assay observations depend on the compound's hidden latent traits plus Gaussian noise on the appropriate scale. Table 4 lists the noise parameters per difficulty tier.

Table 4: Per-difficulty noise parameters. Silico and admet noise are logit-scale σ; dose-response and PK are $\log_{10}$-scale σ. $P(\text{artifact})$ is per-compound; artifact bias enters silico and dose_response as additive logit shifts.

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	0.10	0.35	0.50
$P(\text{artifact})$	0.05	0.12	0.20
Silico artifact bias	0.5	0.6	0.9

D 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 10 continuous latents. Four correlated ADMET traits (lipophilicity, solubility, permeability, cardiotox_safety) live in a 4-D multivariate Gaussian — capturing the "lipophilicity $\rightarrow$ (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.

Five of the eight assays admit a closed-form Gaussian posterior update (Kalman filter for linear-Gaussian observations; scalar tobit for right-censored IC_{50}s; 1-D grid integration for the Bernoulli screen). 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. 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.

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.

E VoI approximations: what makes it a reference, not an upper bound

Three approximations mean the VoI agent is only approximately optimal. *(i) Depth-1 myopia*: VoI picks the single best next action, not the best action sequence. On hard tiers where each individual action has modest absolute VoI, this bites; see Appendix F for a concrete failure mode. *(ii) Composite screening*: batched `single_shot_screen` is scored as a top-k composite rather than jointly optimised over all subsets of the library. *(iii) Off-target abstraction*: off-target activities are folded into the selectivity marginal rather than being tracked as separate latents. The most consequential is depth-1 myopia; the other two are unlikely to change the ranking of an action against its per-dollar competitors in the regime we tested.

F VoI failure-mode analysis on antibiotic-bacterial hard

On antibiotic-bacterial hard at $n = 50$, VoI wins 22/50 (44%) and Sonnet 4.5 wins 20/50 (40%). The paired same-seed comparison on the first 20 seeds (where VoI wins 8 and Sonnet wins 12) reveals a specific trade-off in their action-selection policies that we believe is worth documenting. LLM stochasticity across seeds means an individual seed's outcome is a random draw, but the mechanistic pattern reproduces across the two-run comparisons we ran.

Concrete case (seed 7). The true winner is compound C-05 (potency=0.86, selectivity=0.83, metabolic_stability=0.72, not an artifact). Sonnet nominates C-05 and wins. VoI nominates C-14 (potency=0.69 near threshold, selectivity=0.78 close, metabolic stability=0.56 below threshold, and *an artifact*) and loses.

Action counts on this seed:

Assay	Sonnet	VoI
predict_in_silico	8	23
single_shot_screen	2	2
dose_response	2	1
orthogonal_confirm	0	0
selectivity_panel	2	1
admet_panel	1	4
cell_potency	0	1
Total wet-lab	15	32

VoI spent 22 of its 32 turns on `predict_in_silico` because each one has a small but strictly positive per-dollar VoI — the cheapest ($50) action almost always wins the ratio, even against $5,000 confirmatory assays with much larger absolute VoI. Sonnet allocated more turns to expensive per-compound confirmatory work.

Pattern across divergent seeds. On the paired first-20-seed draw, six seeds are Sonnet-only wins (3, 5, 7, 8, 10, 16) and two are VoI-only wins (0, 1); the rest are concordant. On the six Sonnet-only seeds, VoI runs 19–23 silicos and Sonnet runs 7–13. Both agents follow their usual behavioural

template regardless of which one wins the seed; when Sonnet does win, the mechanism is its more balanced budget allocation — specifically, running `selectivity_panel` on the top-2 candidates head-to-head so that the artifact is exposed by its below-threshold selectivity readout.

G 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 $\mathbb{E}_{o_1}[\max_{a_2} \text{terminal } P(\text{win})]$ per dollar rather than by marginal one-turn VoI. To keep runtime tractable we prune both the a_1 and a_2 candidate sets by depth-1 VoI rank (top-10 union of per-dollar and raw); an exhaustive enumeration is infeasible at typical turn counts.

Initial exploration on divergent seeds. We first evaluated depth-2 on the nine seeds identified in a preliminary run where an LLM won and depth-1 VoI lost (four antibiotic-hard, five gpcr-hard). Depth-2 recovered 1 of 4 antibiotic-hard divergent seeds and 0 of 5 gpcr-hard divergent seeds. 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 antibiotic-hard, depth-2 wins 16/30 and depth-1 wins 16/30, with 4 depth-1-only and 4 depth-2-only pairs (McNemar's exact $p = 1.0$). On gpcr-hard, depth-2 wins 3/30 and depth-1 wins 8/30, with 6 depth-1-only and 1 depth-2-only pairs (McNemar's exact $p = 0.13$, directionally worse). We conclude that depth-2 lookahead does not improve on depth-1 on either cell; the earlier partial-recovery result was regression to the mean.

We also verified that the "silico dominance" pattern (Section 5) is not a Monte-Carlo-noise artifact 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).

H Chain-of-thought ablation

We ran a chain-of-thought variant of the Sonnet 4.5 agent that prepends an explicit "*think step-by-step before answering*" instruction to each turn's prompt. The rest of the scaffolding is identical: same state summary, same JSON output format, same retry policy. The intuition was that explicit reasoning would help on the harder cells (which reward patient budget allocation). We ran CoT-Sonnet on the same 50 seeds per cell used in the main-body comparison.

Empirically CoT does not help, and hurts on easy and medium:

Tier	Baseline Sonnet (n=50)	CoT Sonnet (n=50)	Delta
easy	60	49	−11
medium	52	45	−7
hard	31	30	−1

Aggregate win rate over 250 games per tier, in percent. The easy-tier hit is significant (McNemar's $p < 0.02$); the medium and hard tiers are within noise. This is the opposite of what we expected — we had thought explicit deliberation would help most on the harder cells — and instead CoT's cost falls on the tiers where baseline Sonnet is already reasoning well enough.

Trace mining. Post-hoc analysis of the 14,918 CoT turns:

Tier	retry rate	deliberation/action mismatch	truncation
easy	1.3%	46%	0%
medium	1.8%	49%	0%
hard	1.8%	52%	0%

Retry rate counts turns whose returned JSON failed shape/K=1 validation and triggered a re-prompt (max 3 retries per turn). Truncation counts turns whose response exceeds 3,000 characters and ends without a balanced JSON block — a proxy for hitting `max_tokens`; it is zero at all tiers, so token limits are not the failure mode. The mismatch column is a text heuristic: for each turn, we compare the *last* action-verb keyword mentioned in the model's prose against the `type` in the returned JSON; roughly half of turns show a disagreement. The heuristic is fuzzy — CoT prose often weighs multiple options before committing — but the magnitude is striking. A concrete example (kinase-oncology medium, seed 7): the final CoT turn before nomination runs `predict_in_silico` on `C-10` while its own prose focuses on picking a winner; it then nominates `C-04` on the next turn, whereas baseline Sonnet on the same seed methodically profiled `C-15` and won.

Plausible mechanisms: (i) explicit CoT prompting nudges the model toward longer verbal deliberation that spends output tokens on option-weighing prose whose conclusion the final JSON does not always match; (ii) baseline Sonnet already reasons implicitly at inference time, so the explicit CoT scaffold appears to compete for output-token budget with the actual action call rather than to add capability.

I VoI hyperparameter sensitivity

We ablated the three main VoI hyperparameters on the same 20-seed $\times$ 15-cell grid used in the paper, one factor at a time. All numbers are aggregate win rate per tier over 100 games; baseline values (marked $*$) match the main-body VoI agent.

N (imagined observations per candidate)	easy	medium	hard
16	85%	62%	38%
64	76%	61%	40%
256*	86%	58%	43%
1024	82%	53%	52%

Aggregate is roughly flat in N; $N = 256$ leads on easy and hard by a narrow margin, larger N helps hard slightly but not enough to change the ordering. $N = 64$ is a viable cheaper alternative ($\sim$4$\times$ speedup with $\leq$5 pp cost).

α (cost exponent in VoI/cost$^\alpha$)	easy	medium	hard
0.3	82%	43%	35%
0.7	83%	55%	37%
1.0*	86%	58%	43%
1.5	80%	60%	42%

$\alpha = 1.0$ wins on aggregate; smaller α (0.3, 0.7) hurts medium-tier consistently by 3–15 pp because the agent burns budget on expensive assays too early. $\alpha = 1.5$ is within noise. This supports the paper's Strategy 2 argument: relaxing α toward zero does not recover the gpcr-hard failure mode without paying an aggregate cost.

`RULEOUT_REL_TO_MAX`	easy	medium	hard
0	84%	63%	48%
0.02*	86%	58%	43%
0.05	82%	62%	48%
0.10	77%	51%	47%

Rule-out (the relative-P(win) floor below which surviving compounds are pruned) is the most consequential of the three. Disabling rule-out entirely ($= 0$) matches or slightly beats the baseline 0.02 on medium and hard, at a $\sim$2 pp cost on easy; 0.10 hurts everywhere. The baseline choice is defensible but a lower or zero threshold is a plausible near-equivalent that we did not commit to in the main body.

J LLM system prompt

The system prompt used for all LLM agents is reproduced below verbatim. It is the same across Haiku, Sonnet, Opus, and Llama; 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 total).

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

## Assay mechanics

Every assay produces a noisy readout. Each (compound, paid assay) pair
may only be run ONCE per game. To confirm a suspicious result, run an
orthogonal assay. `predict_in_silico` is deterministic per compound
and cached: repeat calls return the same readout at $0 cost.
```

```
- ‘predict_in_silico‘ ($50): computational potency prediction + coarse
  ADMET summary. Both noisy on logit scale. Artifact compounds get
  their potency estimate inflated by an additive bias.
- ‘single_shot_screen‘ ($200/batch): binary hit/miss at 10 uM.
  Difficulty-dependent false-positive floor; artifacts inflate hit
  probability.
- ‘dose_response‘ ($1,500): observed IC50 in uM, or ">LOQ" if >30 uM.
  Log-normal noise. Artifact compounds have observed IC50 biased down.
- ‘orthogonal_confirm‘ ($4,000): same shape as ‘dose_response‘, but
  artifact bias is zeroed and log-noise sigma halved.
- ‘cell_potency‘ ($3,000): cellular EC50, or ">LOQ". Effective potency
  = potency * (0.3 + 0.7 * permeability).
- ‘selectivity_panel‘ ($5,000): target IC50, numeric selectivity_score
  in [0,1], and observed off-target IC50s.
- ‘admet_panel‘ ($2,000): solubility, permeability, metabolic_stability,
  cardiotox_safety, each in [0,1], plus binary hERG flag.
- ‘in_vivo_pk‘ ($8,000): oral_bioavailability and half_life, each in
  [0,1]. Both readouts share a per-call random effect.

## Nomination

Nomination is one-shot: no partial credit, no re-nominations. Once you
nominate, the game ends and the compound is scored.

## Output format

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

The chain-of-thought variant (Appendix H) 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.
```

K Llama 3.3 70B: cross-family pilot

To test whether the capability-scaling trend observed within the Claude family generalises across model families, we ran Meta's Llama 3.3 70B Instruct on two complete archetypes (antibiotic-bacterial and gpcr-cns) at 20 seeds per cell, using the same prompt and scaffolding as the Claude agents. Llama was served via AWS Bedrock with cross-region inference.

Archetype	Agent	easy	medium	hard
antibiotic-bacterial	Llama 3.3 70B	30%	35%	15%
	Haiku 4.5	50%	38%	23%
	Sonnet 4.5	72%	68%	40%
	VoI	90%	64%	44%
gpcr-cns	Llama 3.3 70B	5%	0%	0%
	Haiku 4.5	18%	10%	16%
	Sonnet 4.5	50%	32%	14%
	VoI	84%	30%	16%

Llama sits below Haiku on 5 of 6 cells and roughly at the greedy baseline level on gpcr-cns (greedy scores 18/2/2% there). This is consistent with the model-capability scaling trend observed within the

Claude family, but from a smaller open-source model. The gap is largest on easy tier, where stronger models have enough budget headroom to compensate for suboptimal action selection.

Caveats. Llama was run on only two of five archetypes (40 games per tier vs. 250 for the Claude agents), so per-tier aggregates are noisier. Bedrock throttling on the Llama endpoint caused a higher error rate during the sweep ($\sim$25% of initial runs failed and had to be retried); all reported numbers are from successfully completed games. We plan to expand the cross-family comparison to additional model families and all five archetypes in a future version.