TADA-BIO: A REPLAYABLE BIOMEDICAL AI-AGENT BENCHMARK FOR BUDGETED DIRECTED EVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific agents in biological discovery campaigns must coordinate sequential operations: choosing which branch to open, how much evidence to collect, which candidates to validate, and when to stop under finite time and budget. Existing scientific evaluations in protein engineering largely emphasize sequence prediction or candidate ranking, leaving the role of an agent as the operator of a budgeted, long-horizon campaign underexplored. We introduce TadA-Bio, a replayable benchmark for evaluating such decision policies. It is built from measured sequences, activities, and provenance from a historical 31-round TadA directed-evolution campaign, but adds a controlled replay layer: each episode exposes a partial public state, hides future candidates and activities, and uses fixed benchmark costs, times, screening noise, and reward rules. An agent can prepare seeds, open libraries, screen, confirm hits, acquire BioSkills, or stop; terminal cash is the primary outcome. Across heuristic policies, three LLM-agent harnesses (Codex, OpenCode, and Claude Code), budget sweeps, and BioSkill settings, the benchmark separates productive operation from cash preservation, wasteful exploration, runtime failures, and unproductive skill purchase. TadA-Bio is a playable, discriminative, and diagnostic offline benchmark, providing a grounded example and potential foundation for future automated biological laboratories.

1 INTRODUCTION

Scientific agents in biological discovery must coordinate sequences of experimental commitments. In a directed-evolution campaign, an agent must decide which branch to open, what evidence to collect, which hit to confirm, when to acquire a capability, and when to stop under finite time and budget. Actions consume resources, so their value depends on when they are taken and which opportunities remain afterward. A useful computational evaluation should therefore test whether a scientific agent can turn limited time, evidence, and budget into validated progress.

Existing benchmarks cover long-horizon tool use, state maintenance, web or software interaction, and delayed consequences (Mialon et al., 2024; Zhou et al., 2024; Jimenez et al., 2024; Andon Labs, 2026); Vending-Bench 2 makes resource management explicit through terminal bank balance (Andon Labs, 2026). Scientific evaluations around protein engineering largely emphasize sequence prediction, candidate ranking, protocol planning, or local closed-loop optimization (Mirza et al., 2025; Notin et al., 2023; O'Donoghue et al., 2023; Rapp et al., 2024). The evaluated object is usually a sequence, a candidate, or a local optimization step; the role of an agent as the operator of a budgeted, long-horizon campaign remains less explored. Screens, confirmations, and purchased workflows consume cash and time before delayed feedback, so trial-and-error can exhaust a campaign before validation. The missing evaluation unit is a replayable campaign trajectory.

We introduce TadA-Bio, a replayable benchmark for evaluating decision policies as budgeted operators of a directed-evolution campaign. It uses measured sequences, activities, and round/library provenance from a historical 31-round TadA wet-lab campaign, the same assay corpus that supports TadABench-1M (Gao et al., 2026). TadA supplies branching programs, sequential library dependencies, noisy screening, and downstream confirmation. TadA-Bio turns this structure into a partially observed replay with a fixed action and accounting protocol. Each episode starts with an initial TadA seed, limited cash, and public campaign state; the policy prepares seeds, opens libraries, mines evidence, confirms hits, purchases BioSkills, or stops. Future candidates and true activities remain

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.

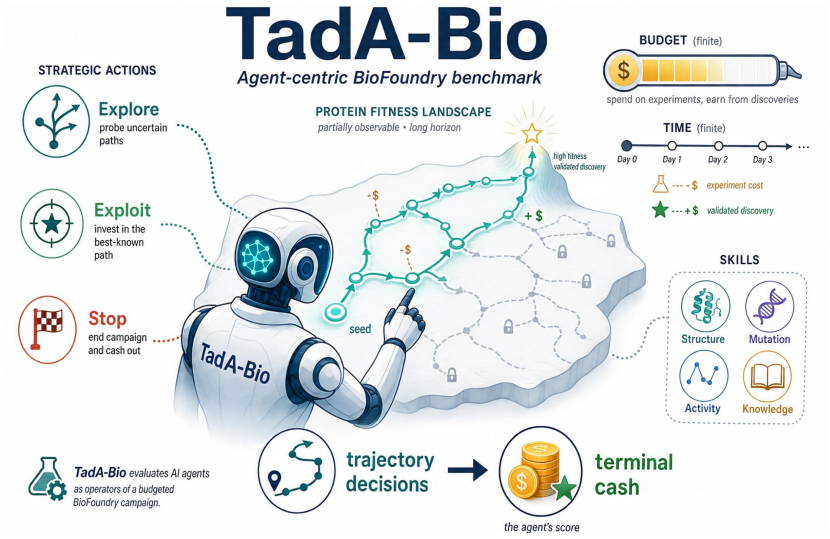

Figure 1: TadA-Bio converts a real 31-round TadA directed-evolution campaign into a replayable, budgeted agent benchmark. The agent observes only the public campaign state, chooses among branch exploration, noisy screening, hit confirmation, BioSkill acquisition, and stopping, while future pools and true activities remain hidden. Rewards are realized only through validated improvements, so terminal score reflects the full operating trajectory: planning, evidence use, cost control, and timely stopping.

hidden behind the host-side judge, while every policy uses the same interface, costs, screening process, and settlement rules.

This design makes cost, scale, and capability acquisition part of the task state. Terminal cash couples biological improvement to the price paid to reach it: exploration may reveal opportunity, but confirmation realizes value, and every action reduces runway. Initial capital defines the operating regime; more cash can open deeper branches and broader evidence collection but can also fund waste, while less cash forces selectivity and may leave no validation reserve. In the market setting, BioSkills represent priced expertise rather than free tools; their value depends on whether the policy purchases them at the right time and converts them into branch choice, triage, confirmation, or stopping. A host-side judge keeps hidden assays, future candidates, screening noise, legality, costs, rewards, and termination outside the container; the policy sees only public JSON observations and legal actions, making trajectories replayable and auditable. The resulting benchmark is an offline replay of a measured campaign, not a prospective test of live-laboratory control.

Our experiments show that productive operation is achievable under the fixed TadA-Bio protocol, while the tested public-information heuristics follow distinct cash-preservation and exploration strategies. Capital sweeps show that runway changes branch reachability and confirmation affordability, not just score scale. Comparisons across three execution harnesses (Codex, OpenCode, and Claude Code) expose differences in branch choice, evidence use, confirmation timing, stopping, and runtime stability; these are end-to-end configuration results. BioSkill studies show that selected preinstalled skills can improve campaign outcomes, while market access couples capability value to purchase timing and downstream use. Together, these results position TadA-Bio as a grounded reference for resource-aware operation under delayed feedback in future automated biological laboratories.

Contributions. We make three contributions.

1. We introduce TadA-Bio, a replayable benchmark for budgeted operation of a real TadA directed-evolution campaign, extending evaluation beyond static sequence scoring.
2. We define a replay protocol with hidden assays, noisy evidence, costly confirmation, BioSkill acquisition, explicit stopping, and auditable cost/reward trajectories.
3. We validate TadA-Bio across agents, budgets, harnesses, and skills, separating campaign strategy, resource allocation, capability pricing, confirmation, stopping, and runtime validity.

2 Related Work

2.1 Long-Horizon Agent Benchmarks

Agent benchmarks now test multi-step interaction, tool use, memory, and stateful execution beyond single-turn question answering (QA) (Liu et al., 2024; Mialon et al., 2024). Web, app, operating-system, text-world, and embodied environments expose agents to realistic interfaces or grounded action sequences (Zhou et al., 2024; Deng et al., 2023; Trivedi et al., 2024; Xie et al., 2024; Shridhar et al., 2021; Wang et al., 2022). Software and machine-learning engineering tasks stress code changes, experiment loops, and recovery from intermediate failures (Jimenez et al., 2024; Huang et al., 2024; Chan et al., 2025). Long-duration and tool-use evaluations show failures under delayed consequences, hidden state, or cumulative cost (Liu et al., 2024; Mialon et al., 2024; Andon Labs, 2026). Domain-specific science benchmarks probe biological protocol planning and chemistry knowledge, but usually not budgeted wet-lab campaign operation (O'Donoghue et al., 2023; Mirza et al., 2025). Most settings remain digital; TadA-Bio instead uses a real directed-evolution trajectory with noisy proxy evidence, costly confirmation, branch management, and optional local tool acquisition.

2.2 Closed-Loop Scientific Optimization and Autonomous Labs

Directed evolution, ML-guided directed evolution, and active learning frame protein engineering as sequential design, screening, and selection over accumulating measurements (Arnold, 1996; Romero & Arnold, 2009; Yang et al., 2019; Biswas et al., 2021; Pandi et al., 2022; Yang et al., 2025). Phage-assisted evolution has also produced genome-editing enzyme improvements through iterative wet-lab campaigns (Miller et al., 2020; Richter et al., 2020; Zhang et al., 2024). Self-driving laboratories extend this design-build-test-learn loop to automated chemistry, materials, and protein experiments with adaptive policies (Granda et al., 2018; Burger et al., 2020; MacLeod et al., 2020; Szymanski et al., 2023; Rapp et al., 2024). Tool-augmented scientific agents and protocol-planning benchmarks motivate planning-oriented evaluation (Boiko et al., 2023; Bran et al., 2024; O'Donoghue et al., 2023; Mirza et al., 2025). Their hardware, assay, or optimizer coupling can complicate replay and controlled comparison (Burger et al., 2020; MacLeod et al., 2020; Rapp et al., 2024). TadA-Bio separates replayable policy evaluation from live laboratory execution while preserving branching, partial observability, resource constraints, and a terminal cash objective.

2.3 Protein Benchmarks from Static Prediction to Trajectory Evaluation

Protein task suites and fitness-landscape resources evaluate fixed datasets, mutation-effect prediction, and offline ranking (Rao et al., 2019; Capel et al., 2022; Xu et al., 2022; Dallago et al., 2021; Notin et al., 2023). Zero-shot and low-data design can prioritize variants, but usually does not score budgeted multi-round experimental policies (Meier et al., 2021; Biswas et al., 2021). Directed-evolution campaigns make this gap concrete because mutation, screening, selection, and confirmation interact over many rounds (Arnold, 1996; Romero & Arnold, 2009; Packer & Liu, 2015). TadABench-1M is closest because it provides a large wet-lab TadA trajectory with realistic temporal shift for future-round ranking (Gao et al., 2026). TadA-Bio builds on that asset and changes the evaluation unit from sequence scorer to trajectory-level experimental policy.

Across these literatures, TadA-Bio occupies a niche: a replayable benchmark for biomedical language-model agents grounded in a real wet-lab campaign, with explicit budget accounting, multi-fidelity evidence, branching search, and optional BioSkill workflows.

3 TadA-Bio: A Replayable Agent Environment for Protein-Discovery Campaigns

TadA-Bio evaluates trajectory policies, with protein scoring serving as a distinct evaluation object. We build it from a historical directed-evolution campaign because it combines branching, partial observability, noisy screens, costly validation, finite resources, and replayable records. Its fixed accounting protocol follows the source wet-lab preparation–screening–confirmation workflow: action durations, charges, screening perturbations, and reward settlement are specified before the sweeps

and shared by every policy. The reward schedule uses a piecewise increasing-marginal-value utility to represent regime-dependent process consequences of activity improvements. The construction has three components. Section 3.1 describes how the historical TadA campaign becomes a partially observable replay environment. Section 3.2 defines the action interface, including BioSkill acquisition as one legal action. Section 3.3 specifies the host-side judge, hidden-state boundary, ledger scoring, and termination rules.

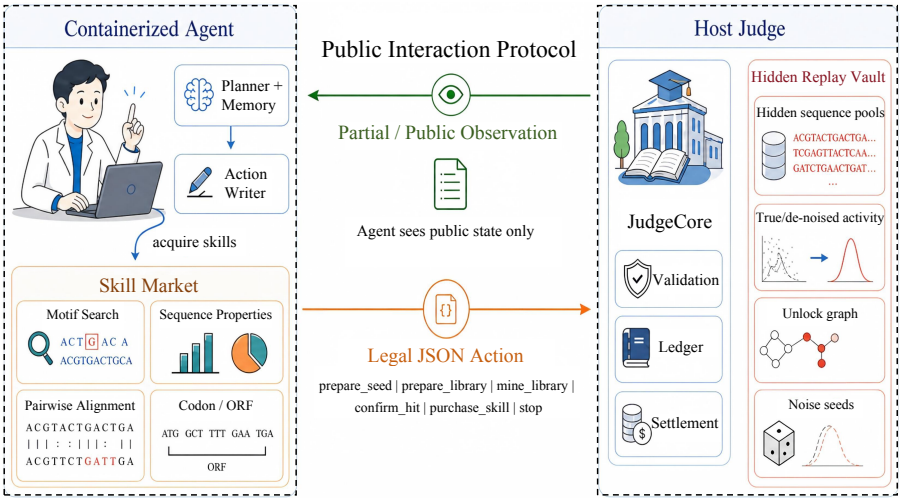

Figure 2: TadA-Bio exposes only public observations and legal actions to the containerized agent, while the host-side replay judge holds hidden replay pools, assay outcomes, noise seeds, unlock logic, time, and ledger state. BioSkills affect the agent workspace; mining returns noisy proxy evidence, and confirmation settles validated reward.

3.1 Campaign Replay as an Agent Environment

We instantiate the benchmark using a historical 31-round TadA directed-evolution campaign (Gao et al., 2026). TadA provides measured activities, temporal provenance, branching seed programs, downstream library dependencies, noisy screening, and confirmation outcomes. These properties expose the key campaign decisions: reaching a branch, allocating evidence budget, selecting candidates for validation, and preserving runway. The seed-program graph organizes the historical sequence–activity and round/library records into a common protocol; Appendix A.5 describes the data flow. During an episode, future candidates, true activities, ordering seeds, noise seeds, and settlement internals remain hidden behind the host-side judge.

The same assay corpus can also support static future-round sequence generalization, but TadA-Bio uses it for a different evaluation view: agent campaign replay. The distinction is summarized in Appendix A.1. Activities are represented as fold-change values relative to the campaign reference construct, with the initial confirmed best normalized to 1.0. A variant matters only if the policy first spends enough to encounter it, decides whether noisy evidence justifies confirmation, and reaches settlement before budget or time runs out. The resulting output is terminal cash and trajectory diagnostics, grounded in the underlying sequence–activity records.

3.2 Action Interface and Capability Acquisition

The public action interface casts the replay as a partially observed decision process. Each episode shows the agent the current day, cash, and public campaign state: visible branches, prior screens, confirmed hits, legal actions, and BioSkill preview cards. The judge keeps ground-truth activities and future campaign outcomes hidden. From this state, the agent chooses whether to prepare search space, screen for noisy evidence, confirm a hit, buy a BioSkill from the Skill Market, or stop. The sequence matters: early actions determine reachability, middle actions trade cheap noisy evidence against costly validation, and late actions spend or preserve the remaining runway. Figure 2 shows this public interaction boundary around the host-side replay judge.

Table 1: Compact action taxonomy in the public agent interface. Actions are grouped by the scale of decision they expose, and each role states the agent ability being tested. The complete action schema, direct costs, time costs, batch sizes, and reward rules are given in Appendix A.3; the BioSkill panel is detailed in Appendix A.4.

Decision scale	Public action	Agent ability tested
Program	`prepare_seed`: activate a reachable sequence background.	Tests when an agent commits resources to a branch before downstream screening is available.
Library	`prepare_library`: open one mutational window under a prepared seed.	Tests branch timing and exploration–exploitation tradeoffs.
Screening	`mine_library`: buy a proxy-screening batch.	Provides cheap, uncertain observed activity without direct reward.
Candidate	`confirm_hit`: validate a provisional hit.	Removes screening noise and is the only action that can realize incremental reward.
Workflow	`purchase_skill`: buy a local BioSkill workflow from Skill Market preview cards.	Tests capability acquisition: whether the agent can identify a useful tool, pay for it at the right time, and use it to improve downstream decisions.
Episode	`stop`: end the episode and keep the current ledger state.	Tests stopping under uncertainty and prevents reward from unconstrained action volume.

Table 1 groups the actions by campaign role. `prepare_seed` and `prepare_library` create reachable search space before any new measurement appears. `mine_library` and `confirm_hit` form the evidence–validation pair: mining buys noisy proxy observations without immediate reward, while confirmation spends extra resources to remove screening noise and settle any validated improvement. `purchase_skill` tests capability acquisition by letting the agent buy a BioSkill from preview cards and use it in later decisions. `stop` makes ending the campaign an explicit policy choice because it preserves the current ledger.

3.3 Host-Side Judge, Scoring, and Termination

TadA-Bio enforces evaluation through a host-controlled replay judge. The agent runs inside a containerized workspace and sees only the public campaign state, current budget, legal actions, and visible skill cards. The judge remains outside the workspace. It holds hidden replay data and handles the runtime mechanics: legality, accounting, noisy screens, confirmation settlement, ledger updates, and logging. This boundary lets the agent choose any policy while keeping hidden data access and scoring under evaluator control.

The judge distinguishes screening from confirmation by whether proxy noise is present. During screening, `mine_library` returns reproducible noisy observations for latent activities using multiplicative clipped Gaussian noise with zero mean, standard deviation 0.10, and clipping at plus/minus 0.20. During confirmation, `confirm_hit` settles against the corresponding recorded assay activity. This two-stage design keeps screening as an inexpensive proxy and confirmation as the validation operation that realizes measured biological value. Profit can be realized only at this confirmation stage, and only when the confirmed candidate improves on the previous confirmed best.

The evaluation combines terminal cash with a qualitative audit log. Terminal cash is the underlying objective, reported as capital-adjusted profit, $\mathrm{cash}_{\mathrm{final}} - \mathrm{cash}_0$; confirmed best activity and the remaining trajectory statistics are diagnostics. Every non-terminal action advances time by its duration and updates cash by the direct action cost plus the fixed daily burn, while screening itself has zero immediate reward. The audit log records the step-by-step observations, actions, costs, confirmations, failures, and stopping rationale for each agent, which lets us inspect how a final score was produced

rather than treating it as an opaque number. The tiered payout ledger in Appendix A.3 defines the exact confirmation settlement rule used by the runtime.

Episodes end under a small set of terminal conditions. The agent may voluntarily submit `stop`; the judge may terminate when cash or remaining time prevents any further legal action; the episode ends at the 180-day horizon; and unrecoverable invalid-action or backend failures are recorded as failure terminals. These modes keep ordinary stopping, budget or timeout exhaustion, and interface/runtime failures separate in the final diagnostics. A full summary of the seed graph, costs, screening details, reward tiers, and BioSkills panel appears in Appendix A.2–A.4.

4 Experiments

We organize the evaluation around three questions. First, can the replay distinguish productive campaign operation from cash preservation and unguided spending? Second, does it separate agents by execution harness and operating behavior under the same budget? Third, does access to BioSkills improve campaign outcomes when skills are preinstalled or offered for purchase? These experiments examine achievable profit relative to public-information heuristics, differences between deployed system configurations, and the effect of optional capabilities.

4.1 Experimental Settings

Each episode places the agent in a partially observed TadA replay. The public state begins with `seed1` visible but not yet prepared, and the agent chooses whether to prepare seeds, open libraries, mine noisy evidence, confirm hits, purchase BioSkills, or stop. The default initial cash is \$100k for the policy/reference, agent, single-skill, and market settings. The main cash sweep uses \$30k, \$50k, \$75k, \$100k, \$150k, \$200k, \$300k, \$500k, and \$700k, with the same cost model, noise process, and replay seeds. Terminal cash is the underlying objective, reported as capital-adjusted profit, $\mathrm{cash}_{\mathrm{final}} - \mathrm{cash}_0$. We use confirmed best activity, target success at $\geq 5\times$, terminal status, and screen volume as diagnostics. Each condition uses ten replay seeds, varying candidate ordering and screening-noise draws within the same campaign, not independent biological tasks or accounting rules. Figure error bars summarize variability across these seeds. Exploratory paired comparisons use nominal pointwise 95% bootstrap intervals (Appendix A.5); other rankings are descriptive. All LLM-agent experiments were run through provider API calls rather than local model training or GPU inference; the replay judge and plotting scripts require only ordinary CPU-side orchestration and paper-generation compute.

4.2 Comparing Agents against Heuristic Policies

We first test whether an agent can operate effectively in TadA-Bio by comparing the Codex / GPT-5.5 reference agent with five heuristic baselines across the capital sweep. The baselines are random legal action, static ranking with a confirmation reserve, bounded scouting, aggressive upside-seeking, and empirical expected-value planning. Together, they cover unguided exploration, reserve management, selective evidence collection, and explicit cost–benefit planning, providing reference behaviors against which to assess whether an agent can turn campaign resources into validated progress.

We define baseline fairness through information equivalence. Each policy receives the same public observation, legal actions, and responses to its own actions, and uses the same action interface, evidence access, costs, screening, confirmation, and termination rules. Future candidates and unrevealed assays remain outside the policy interface. History-dependent rules, rollout procedures, search methods, and other planners are compatible with this interface when they use legally acquired observations; a solver that loads the complete replay table would receive hidden information. Our reference suite focuses on interpretable public-information policies whose decisions can be audited from the observation and action history.

Figure 3 shows that positive profit is achievable but is not consistently recovered by the tested heuristics. The mean profit of the Codex / GPT-5.5 reference agent is negative when the initial budget is \$30k, becomes positive by \$50k, and reaches \$394.0k mean profit at \$700k, indicating that additional runway can be converted into confirmable scientific value. In contrast, the heuristic policies remain close to cash preservation or negative across the same budgets. Random legal and upside-

greedy policies spend capital without reliably producing target-level confirmations; static ranking and scout-confirm policies reduce waste but largely fail to improve the confirmed activity; and the empirical expected-value (EV) planner is the strongest heuristic while still remaining near break-even at $100k budget ($-12.7k).

Starting capital is subtracted from the score, so the $700k condition does not receive credit simply for starting with more cash. It permits more exploration while retaining a confirmation reserve; at low budgets these expenditures compete directly. Budget therefore changes the feasible trajectories, not simply the scale of the reported profit. The break-even point and profit values remain specific to the published cost and reward calibration.

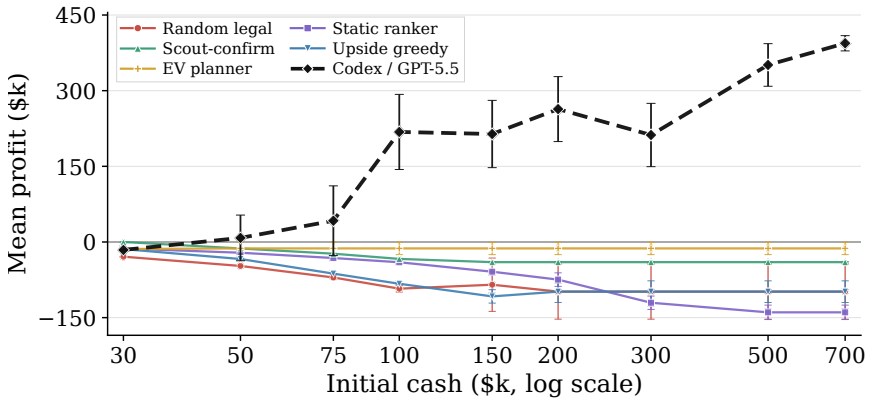

Figure 3: The Codex / GPT-5.5 reference policy achieves positive mean profit at larger budgets, unlike the tested heuristics. Across the initial-cash sweep, active rule-based policies mostly remain near break-even or lose cash, showing that spending budget or greedily following apparent upside is not enough to create value. The Codex / GPT-5.5 reference agent becomes profitable once the campaign has enough runway. Markers show mean profit over ten replay seeds at each initial cash level on a log scale, with error bars denoting the standard error of the mean (SEM); barely visible error bars indicate low or zero variation across seeds.

4.3 Differentiating Agent Performance and Behaviors

A long-horizon R&D benchmark should distinguish deployed agents by branch choice, evidence use, confirmation timing, and stopping. Since the execution harness shapes planning, tool invocation, context handling, and recovery of an agent, harness and model can both be the factors that influence the behaviors of the agent. We evaluate fifteen different harness/model settings spanning three execution harnesses—Codex, OpenCode, and Claude Code—under the same $100k replay campaign.

Figure 4A shows substantial differences between the tested system configurations. Codex / GPT-5.4 is the strongest setting, reaching $412.5k mean profit above the $100k starting cash, and every Codex-harness agent still ranks ahead of every OpenCode-harness agent. The Claude Code harness lands between these families: Claude Code / Sonnet 4.6 is the strongest non-Codex setting, ranking third overall at $274.1k mean profit with target success on seven of ten seeds—ahead of the Codex / GPT-5.5 baseline and every OpenCode setting—while Claude Code / Opus 4.8 sits mid-table at $136.0k. These are end-to-end system results, encompassing model behavior as well as context handling, tool interfaces, and backend reliability.

The matched-model comparisons help separate this gap from backend failures. Codex exceeds OpenCode by $373.6k mean profit for GPT-5.4 (95% paired interval [$244.6k, $460.4k]) and $295.7k for GPT-5.3-codex ([$123.7k, $435.3k]). Neither comparison contains backend-error episodes, so those failures cannot explain these gaps. These comparisons concern complete configurations; they do not isolate the effect of any individual harness component.

Within harnesses, GPT-5.4 has the highest mean profit in Codex and DeepSeek-V4-Pro in OpenCode. Claude Code / Sonnet 4.6 has a higher mean than Opus 4.8, but this is not evidence of a general model-size relationship. Panel B distinguishes low-value confirmations, lack of target success, and backend/tool failures. These terminal outcomes complement profit without by themselves identifying why a policy failed. Fine-grained rankings should remain descriptive unless supported by a specific uncertainty analysis.

The aggregate comparison is also inspectable at the trajectory level. The supplementary audit in Appendix B illustrates how spending, validated reward, and stopping unfold within a single episode; the main comparisons below focus on aggregate behavior across replay seeds.

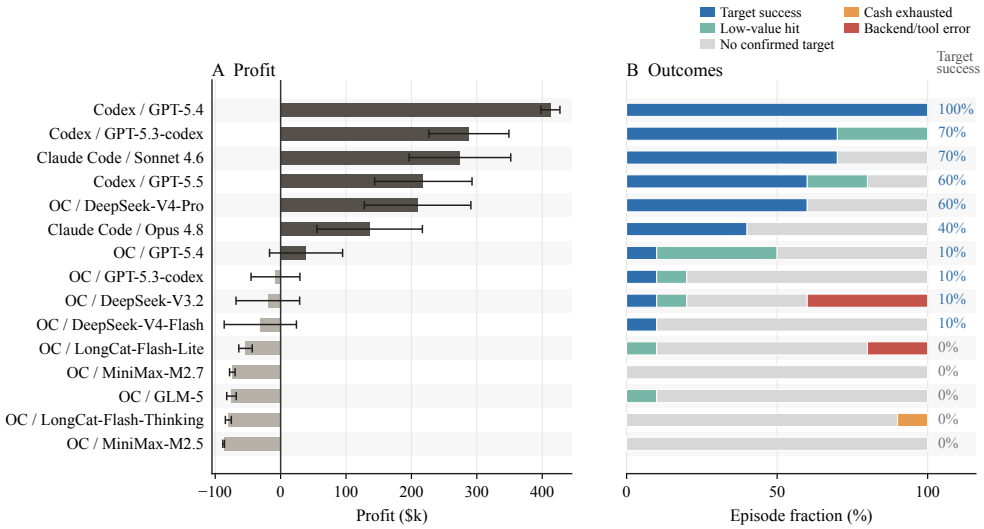

Figure 4: TadA-Bio separates the operational capabilities of different execution harnesses and underlying models across three harness families (Codex, OpenCode, Claude Code). Labels follow Harness / Model; `OC` abbreviates `OpenCode`. (A) Bars show mean profit relative to the $100k initial cash, with SEM over ten seeds; Codex / GPT-5.4 is strongest at $412.5k mean profit and Claude Code / Sonnet 4.6 is the strongest non-Codex setting, with matched-model differences quantified in Appendix A.5. Within Claude Code, Sonnet 4.6 has higher mean profit than Opus 4.8. (B) Stacked bars show terminal outcome composition; Target success means confirmed activity $\geq 5\times$.

4.4 Evaluating BioSkill Utility and Market Selection

The BioSkill experiments distinguish capability utilization from capability acquisition within the same budgeted campaign. In the forced single-skill setting, exactly one BioSkill is installed at episode start and the market is disabled, testing its contribution to the agent's policy without a purchase decision. In the Skill Market setting, the agent instead receives preview cards for the same skill panel and decides whether and when to pay the purchase cost and incur the one-day delay. This setting evaluates acquisition and subsequent use jointly.

All skill conditions are paired with the no-skill control using the same starting cash, replay seeds, hidden candidate pools, and reward schedule. Confirmed best activity measures validated biological progress, whereas capital-adjusted profit measures the net value retained after campaign costs. These complementary outcomes assess whether capability access improves resource-aware experimental decisions rather than merely increasing the volume of analysis.

Figure 5 shows heterogeneous mean effects of preinstalled skills. Relative to the Codex / GPT-5.5 no-skill baseline, `motif search` improves both mean confirmed activity and mean profit, whereas `distance calculations` improve mean profit. Most other forced-skill conditions show little gain or lower mean profit.

In contrast, marketing-framed and neutral market access reduce mean profit by $174.9k and $176.1k relative to the matched no-market control; both nominal paired intervals lie below zero (Ap-

pendix A.5). Under the tested preview cards, price, delay, and Codex / GPT-5.5 configuration, this contrast separates the benefit of selected preinstalled capabilities from the challenge of acquiring and using capabilities profitably. The market results reflect acquisition and downstream use jointly, rather than the intrinsic utility of an individual BioSkill. TadA-Bio thus evaluates capability access as part of campaign operation, where tool use must be coordinated with experimental commitments and remaining resources.

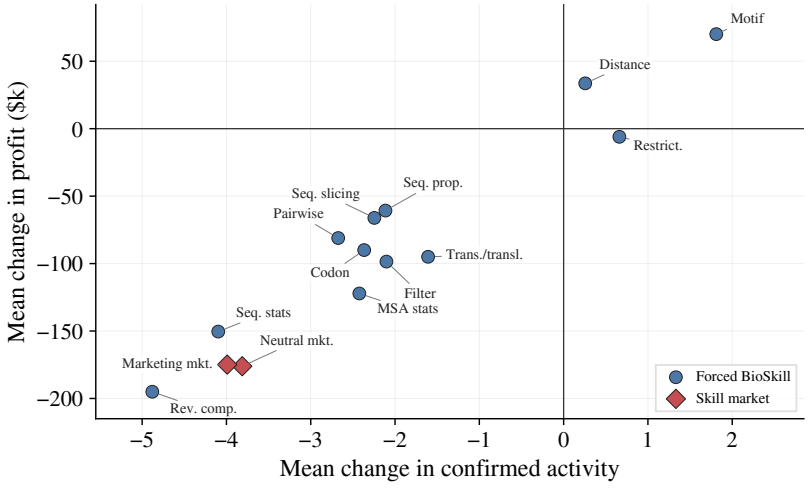

Figure 5: TadA-Bio tests whether agents can utilize domain-specific tools (BioSkills). Each marker reports the mean effect over ten matched seeds relative to the corresponding `no_skill` control. Motif search improves both confirmed activity and profit; distance calculations improve profit on average. Both market variants have lower mean profit under the tested preview, price, and delay conditions; this is not a general claim about agents' tool-selection ability.

5 Conclusion

We introduced TadA-Bio to evaluate scientific agents as budgeted campaign operators, extending evaluation beyond static sequence scoring. Historical TadA records become a partially observed replay where exploration, screening, confirmation, capability acquisition, and stopping consume explicit resources. A host-side judge enforces the information boundary, and capital-adjusted profit and trajectories test whether agents turn finite runway into validated progress. The three experimental questions jointly establish the benchmark's operating scope: the capital sweep shows how an agent converts runway into confirmed value under the shared accounting protocol; harness/model comparisons separate deployed configurations by profit, target success, failures, and trajectories; and BioSkill studies show how capability access interacts with purchase timing and downstream campaign decisions. Together, these results establish TadA-Bio as a playable, discriminative, and diagnostic benchmark for budgeted directed evolution.

The replay uses TadA directed evolution as a grounded biological setting with measured sequences, activities, and round/library provenance from a 31-round campaign. Its branching programs, sequential library dependencies, noisy screens, and confirmation outcomes expose the decisions required for trajectory-level operation. Batch scales, action durations, confirmation costs, relative charges, and reward magnitudes follow the source wet-lab preparation–screening–validation sequence and were fixed before all sweeps. Shared accounting preserves the trade-offs among exploration, validation, and stopping under one declared protocol. TadA-Bio provides a complete first benchmark instance and a basis for extending campaign-level evaluation to additional proteins and workflows.

Ethics Statement

This paper introduces TadA-Bio, a replayable evaluation harness for long-horizon experimental planning constructed from existing TadA directed-evolution assay records. We use "biomedical" to denote experimental biological optimization relevant to therapeutic protein and genome-editing workflows, not patient-facing clinical decision support, diagnosis, treatment recommendation, or clinical deployment. The study is retrospective and computational only: it does not involve human subjects, private data, animal experiments, or new wet-lab assays. Agents interact with an offline environment that replays outcomes for previously assayed variants under a bounded action space and budget model. The reported results should therefore be interpreted as benchmark evidence about AI planning and evaluation, not as validated biological discoveries or actionable recommendations for synthesis, experimentation, or therapeutic use. Because the domain concerns biological optimization, any real-world follow-up would require separate expert review and appropriate institutional biosafety oversight.

Reproducibility Statement

The reproducibility assets are split between this paper-source package and the companion `TadABio` repository, which will be released as open source under a license. The paper package contains the manuscript source and plotted figure assets; `TadABio` contains the replay runtime, data-preparation command, launch scripts, configs, docs, tests, and output schemas needed to reproduce the reported figures and trajectories.

LLM Use Disclosure. We used a large language model, GPT-6 Pro, to assist with drafting, rewriting, and restructuring portions of the manuscript text, including substantial English prose revisions for clarity, organization, and presentation. The model was used as a writing assistant based on author-provided technical content, outlines, and experimental results. It was not used to run experiments, generate the reported numerical results, or determine the scientific conclusions of the paper. All technical claims, figures, tables, citations, and references were reviewed and verified by the authors, who take full responsibility for the content of this manuscript.

References

Andon Labs. Vending-Bench 2. https://andonlabs.com/evals/vending-bench-2, 2026. Accessed: 2026-05-05.

Frances H Arnold. Directed evolution: creating biocatalysts for the future. *Chemical Engineering Science*, 51(23):5091–5102, 1996.

Surojit Biswas, Grigory Khimulya, Ethan C. Alley, Kevin M. Esvelt, and George M. Church. Low-N protein engineering with data-efficient deep learning. *Nature Methods*, 18(4):389–396, 2021. doi: 10.1038/s41592-021-01100-y.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, et al. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6:525–535, 2024. doi: 10.1038/s42256-024-00832-8.

Benjamin Burger, Phillip M. Maffettone, Vladimir V. Gusev, Catherine M. Aitchison, Yang Bai, et al. A mobile robotic chemist. *Nature*, 583:237–241, 2020. doi: 10.1038/s41586-020-2442-2.

Henriette Capel, Robin Weiler, Maurits Dijkstra, Reinier Vleugels, Peter Bloem, and K. Anton Feenstra. ProteinGLUE multi-task benchmark suite for self-supervised protein modeling. *Scientific Reports*, 12:16047, 2022. doi: 10.1038/s41598-022-19608-4.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, et al. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *The Thirteenth International Conference on Learning Representations*, 2025.

Christian Dallago, Jody Mou, Kadina E. Johnston, Bruce J. Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K. Yang. FLIP: Benchmark tasks in fitness landscape inference for proteins. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samual Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jin Gao, Juntu Zhao, Zirui Zeng, Jiaqi Shen, Junhao Shi, Dukun Zhao, Yuming Lu, and Dequan Wang. TadABench-1M: A large-scale wet-lab protein benchmark for rigorous OOD evaluation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Jaroslaw M. Granda, Liva Donina, Vincenza Dragone, De-Liang Long, and Leroy Cronin. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature*, 559 (7714):377–381, 2018. doi: 10.1038/s41586-018-0307-8.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309. PMLR, 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 *The Twelfth International Conference on Learning Representations*, 2024.

Joana Lima-Ramos, Pär Tufvesson, and John M. Woodley. Application of environmental and economic metrics to guide the development of biocatalytic processes. *Green Processing and Synthesis*, 3(3):195–213, 2014. doi: 10.1515/gps-2013-0094.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*, 2024. arXiv:2308.03688.

B. P. MacLeod, F. G. L. Parlane, T. D. Morrissey, Florian Häse, L. M. Roch, et al. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*, 6(20):eaaz8867, 2020. doi: 10.1126/sciadv.aaz8867.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29287–29303, 2021.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*, 2024.

Shannon M Miller, Tina Wang, and David R Liu. Phage-assisted continuous and non-continuous evolution. *Nature Protocols*, 15(12):4101–4127, 2020.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, et al. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17:1027–1034, 2025. doi: 10.1038/s41557-025-01815-x.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. *Advances in Neural Information Processing Systems*, 36:64331–64379, 2023.

Odhran O'Donoghue, Aleksandar Shtedritski, John Ginger, Ralph Abboud, Ali Essa Ghareeb, and Samuel G. Rodriques. BioPlanner: Automatic evaluation of LLMs on protocol planning in biology. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2676–2694, 2023. doi: 10.18653/v1/2023.emnlp-main.162.

Michael S. Packer and David R. Liu. Methods for the directed evolution of proteins. *Nature Reviews Genetics*, 16(7):379–394, 2015. doi: 10.1038/nrg3927.

Amir Pandi, Christoph Diehl, Ali Yazdizadeh Kharrazi, Scott A. Scholz, Elizaveta Bobkova, et al. A versatile active learning workflow for optimization of genetic and metabolic networks. *Nature Communications*, 13:3876, 2022. doi: 10.1038/s41467-022-31245-z.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating protein transfer learning with TAPE. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jacob T. Rapp, Bennett J. Bremer, and Philip A. Romero. Self-driving laboratories to autonomously navigate the protein fitness landscape. *Nature Chemical Engineering*, 1:97–107, 2024. doi: 10.1038/s44286-023-00002-4.

Michelle F Richter, Kevin T Zhao, Elliot Eton, Audrone Lapinaite, Gregory A Newby, B. W. Thuronyi, Christopher Wilson, Luke W Koblan, Jing Zeng, Daniel E Bauer, et al. Phage-assisted evolution of an adenine base editor with improved Cas domain compatibility and activity. *Nature Biotechnology*, 38(7):883–891, 2020.

Philip A. Romero and Frances H. Arnold. Exploring protein fitness landscapes by directed evolution. *Nature Reviews Molecular Cell Biology*, 10(12):866–876, 2009. doi: 10.1038/nrm2805.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *The Ninth International Conference on Learning Representations*, 2021.

Nathan J. Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E. Kumar, Tanjin He, et al. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*, 624:86–91, 2023. doi: 10.1038/s41586-023-06734-w.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850.

Pär Tufvesson, Joana Lima-Ramos, Naweed Al Haque, Krist V. Gernaey, and John M. Woodley. Advances in the process development of biocatalytic processes. *Organic Process Research & Development*, 17(10):1233–1238, 2013. doi: 10.1021/op4001675.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022. doi: 10.18653/v1/2022.emnlp-main.775.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Chang Ma, Runcheng Liu, and Jian Tang. PEER: A comprehensive and multi-task benchmark for protein sequence understanding. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2022.

Jason Yang, Ravi G. Lal, James C. Bowden, Raul Astudillo, Mikhail A. Hameedi, et al. Active learning-assisted directed evolution. *Nature Communications*, 16:714, 2025. doi: 10.1038/s41467-025-55987-8.

Kevin K. Yang, Zachary Wu, and Frances H. Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694, 2019. doi: 10.1038/s41592-019-0496-6.

Emily Zhang, Monica E Neugebauer, Nicholas A Krasnow, and David R Liu. Phage-assisted evolution of highly active cytosine base editors with enhanced selectivity and minimal sequence context preference. *Nature Communications*, 15(1):1697, 2024.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, et al. WebArena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024.

Table 2: Evaluation views over the TadA substrate. TadA-Bio keeps the same sequence–activity basis but evaluates campaign operation rather than only candidate-level prediction.

Axis	TadA sequence-generalization view	TadA-Bio campaign replay
Scientific substrate	Historical TadA wet-lab assay corpus	Same corpus organized as replay state
Evaluated object	Sequence-level predictor or ranker	Trajectory-level experimental policy
Observation	Labeled variants with future rounds held out	Public campaign state revealed through agent actions
Decision	Score or rank candidate variants	Prepare seeds, open libraries, screen, confirm, acquire skills, or stop
Feedback	Offline labels for held-out variants	Noisy and delayed observations mediated by host-side replay
Constraints	Temporal shift and out-of-distribution generalization	Budget, time, branch unlocks, assay cost, and validation risk
Primary output	Ranking or prediction quality	Final profit, confirmed best activity, target success, terminal status, and trajectory trace

A Supplementary Benchmark Specification and Provenance

This appendix follows the construction of the benchmark from evaluation semantics to runtime evidence. We first distinguish campaign replay from static sequence evaluation, then specify the public topology and hidden-state boundary, formalize cost and reward settlement, document the BioSkill and experiment manifests, and finally audit how spending and validated activity unfold along trajectories.

A.1 Evaluation View over the TadA Substrate

TadA-Bio uses the TadA directed-evolution records as the scientific substrate for a campaign-level evaluation. The underlying measurements remain sequence–activity assays from the historical campaign, but the benchmark exposes them through a replay process in which the evaluated system must operate under partial observability, finite capital, delayed confirmation, and stopping risk. The key change, summarized in Table 2, is therefore the unit of evaluation: from predicting or ranking variants in a fixed table to managing a budgeted experimental trajectory.

The replay setting changes how the same biological measurements enter evaluation. A high-activity variant does not determine performance by its latent value alone; it affects the episode only if the policy reaches the relevant branch, screens enough evidence to nominate the candidate, pays for confirmation, and stops with value left on the ledger. Final profit and confirmed best activity therefore summarize an operating trajectory, not only the quality of a variant ranking.

A.2 Public Replay Topology

The public replay state is organized around seven buildable seed programs derived from the historical TadA campaign. The root program, `seed1`, is visible at episode start, but the agent must still spend budget to prepare it. Downstream programs are revealed only through specific upstream libraries, making reachability part of the experimental decision process. Table 3 reports the seed-level topology; the complete 31-row library manifest is included in the released runtime artifact.

The sparse topology creates the main operating trade-off. Conservative spending can leave later branches unreached, whereas aggressive expansion can exhaust the budget required to validate promising candidates. Effective policies must therefore coordinate reachability, evidence acquisition, and confirmation capacity.

A.3 Formal Runtime, Accounting, and Reward Settlement

The replay protocol fixes the experimental degrees of freedom before any sweep is run. Table 4 specifies the episode horizon, initial-state visibility, cash grid, mining batch menu, replay seeding, score, and diagnostics. The action durations and accounting charges are grounded in the source wet-lab

Table 3: Canonical seed-program graph. Sparse unlocking makes early branch choice consequential: most libraries expand local search, while a small subset reveal downstream programs.

Seed	Hist. rounds	Revealed by	Local libs	Decision role
`seed1`	r1–r8	root, visible at start	8	Entry program; visible initially but still subject to preparation cost.
`seed2`	r9–r16	`seed1_library3`	8	Primary downstream branch opened from early local exploration.
`seed3`	r17–r20	`seed1_library3`	4	Parallel downstream branch that tests breadth beyond the first expansion path.
`seed4`	r21	`seed2_library4`	1	Narrow bridge program requiring retained expansion and confirmation capacity.
`seed5`	r22–r28	`seed4_library1`	7	Later high-upside region reachable only after multiple prior commitments.
`seed6`	r29–r30	`seed5_library7`	2	Downstream breakout branch observed in many profitable trajectories.
`seed7`	r31	`seed6_library2`	1	Terminal downstream program, typically reachable only with substantial remaining runway.

Table 4: Formal replay protocol used unless otherwise noted. These settings are fixed before policy, agent, budget, and BioSkill sweeps are evaluated.

Item	Value
Horizon	180 simulated days
Initial visibility	`seed1` visible but not yet prepared
Initial cash	Main sweep: \$30k, \$50k, \$75k, \$100k, \$150k, \$200k, \$300k, \$500k, \$700k; default agent, skill, and market setting: \$100k
Daily burn	\$800 per simulated day
Mining batch menu	780, 880, 1020, or 1280 candidates per `mine_library` action
Replay control	Ten replay seeds per condition; hidden ordering and screening noise are seeded per run
Primary score	Capital-adjusted profit, $\mathrm{cash}_{\mathrm{final}} - \mathrm{cash}_0$
Diagnostics	Confirmed best activity, target success at $\geq 5\times$, terminal status, failure taxonomy, screening volume, and trajectory log

protocol and its preparation–screening–confirmation sequence, then fixed as part of the benchmark specification. A different schedule would define a new benchmark scenario; it does not change the internal comparison among policies under the reported protocol. With these settings held constant, changes in profit are interpreted as differences in how a policy converts the same hidden TadA substrate into confirmed value under limited runway. Cash values are benchmark accounting units whose role is to impose a consistent ordering over preparation, library construction, screening, confirmation, skill acquisition, and daily runway consumption.

The action ledger in Table 5 defines the local trade-offs the policy repeatedly faces. Preparation and library construction buy reachability, mining buys noisy evidence, confirmation is expensive and delayed but is the only route to reward, BioSkill purchase exchanges cash and time for optional analysis, and stopping preserves the current ledger. Because each non-terminal action consumes both cash and simulated time before any reward can be settled, exploration volume has opportunity cost; profitable trajectories must pair search breadth with a validation reserve.

If a sequence has latent activity a, mining returns a reproducible noisy proxy

$$\widetilde{a} = a(1+\epsilon), \qquad \epsilon \sim \mathcal{N}(0, 0.10^2) \text{ clipped to } [-0.20, 0.20]. \tag{1}$$

Table 5: Compact action ledger. Total charge equals direct action cost plus daily burn over the action duration. Screening returns noisy evidence; confirmation is the only action that can settle reward.

Action	Total charge	Days	Operational interpretation
`prepare_seed`	\$2,765	3	Activate a reachable sequence background.
`prepare_library`	\$1,742	2	Open one mutational window under a prepared seed.
`mine_library` (780)	\$8,400	9	Run a lower-throughput noisy proxy-screening batch.
`mine_library` (880)	\$8,460	9	Run a moderate noisy proxy-screening batch.
`mine_library` (1020)	\$9,340	10	Run a larger noisy proxy-screening batch.
`mine_library` (1280)	\$9,400	10	Run the largest default noisy proxy-screening batch.
`confirm_hit`	\$10,067	12	Validate one provisional hit; removes screening noise and may realize reward.
`purchase_skill`	\$1,800	1	Acquire a local BioSkill after observing only its preview card.
`stop`	\$0	0	End the episode and preserve the current ledger state.

Table 6: Piecewise reward calibration in the default formal setting. The increasing multipliers represent regime-dependent process value over the TadA activity scale; the resulting dollar values are benchmark accounting units rather than universal monetary valuations of protein activity.

Activity interval	Multiplier	Segment payout cap	Interpretation
$[1.00, 1.314)$	1.5	\$35.8k	Marginal gain regime; profitable only when confirmation cost is controlled.
$[1.314, 3.78)$	2.4	\$119.2k	Moderate upside regime; selective confirmation remains important.
$[3.78, 7.44)$	18.0	\$234.3k	Strong gain regime in which validation can finance continued search.
$[7.44, \infty)$	30.0	unbounded	Breakout regime; high value when screening evidence supports confirmation.

Confirmation settles against the corresponding campaign-record activity. Reward is credited only when a confirmed candidate improves on the previous confirmed best, and the runtime splits the improvement interval across the tiers in Table 6. Because one mining action followed by one confirmation costs roughly \$19k–\$20k after time burn, screening volume alone is not a sufficient objective; the policy must also preserve validation capacity.

The reward schedule is a process-value surrogate rather than a claim that assay activity has a universal monetary exchange rate. In biocatalytic process development, the operational value of an improved catalyst depends jointly on activity, catalyst loading, reaction time, productivity, yield, and downstream constraints, so equal fold-changes in activity need not have equal process value (Tufvesson et al., 2013; Lima-Ramos et al., 2014). Over the TadA activity scale used here, crossing an activity regime can represent a qualitatively different operating opportunity, such as making a confirmation worthwhile or enabling continued search after settlement. We therefore use a piecewise, increasing-marginal-value schedule with marginal, moderate-upside, strong-gain, and breakout regimes. The thresholds and multipliers are fixed benchmark calibration choices selected relative to the TadA activity scale and the declared screen–confirmation cost structure; the resulting dollar values are accounting units, not procurement prices or universal biological valuations.

Table 7: BioSkill panel summarized by decision role. Each market purchase has a $1,000 direct price and one simulated day of delay before burn is added. Full preview cards and workflow files are released with the benchmark artifact.

Decision role	BioSkills in panel	Potential operational effect
Motif and local-pattern search	Motif Search; Sequence Slicing	Triage candidates and interpret branches when local sequence patterns recur among provisional hits.
Sequence sanity checks	Sequence Properties; Sequence Statistics; Reverse Complement; Transcription/Translation	Reduce basic interpretation errors involving strand direction, stop codons, abnormal lengths, or composition outliers.
Similarity and diversity analysis	Pairwise Alignment; MSA Statistics; Distance Calculations	Inform the choice between local exploitation and diversity-seeking exploration before confirmation.
Expression and constructability checks	Codon Usage; Restriction Sites	Identify practical sequence liabilities that may change which provisional hit is worth validating.
Filtering and cleanup	Filter Sequences	Remove low-quality or ambiguous candidates before downstream reasoning.

Table 8: Design-level experiment matrix for the reported figures. Counts are by experimental setting, with ten replay seeds per setting. The shared Codex / GPT-5.5 no-skill condition is counted once when reused as a matched baseline.

Family	Settings counted	Settings	Runs
Shared baseline	Codex / GPT-5.5, $100k initial cash, no skill	1	10
Policy sweep	Five active heuristic policies crossed with nine initial-cash levels	45	450
Capital sweep	Eight non-baseline Codex / GPT-5.5 cash levels, with the $100k baseline reused	8	80
Agent sweep	Fourteen non-baseline harness/model settings across the Codex, OpenCode, and Claude Code harnesses; Codex / GPT-5.5 is the baseline row	14	140
Forced-skill sweep	Twelve single-skill BioSkill settings; no-skill reuses the shared baseline	12	120
Skill-market sweep	Neutral full market and marketing-framed full market	2	20

A.4 BioSkills and Experiment Provenance

BioSkills are controlled interventions on the agent-facing analysis workspace. They do not alter hidden candidate pools, true activities, screening noise, reward rules, or replay seeds. Table 7 groups the skill panel by the operating decision each capability can affect, such as candidate triage, sequence sanity checking, diversity analysis, constructability assessment, and filtering. In the forced single-skill setting, one skill is installed at episode start and the market is disabled, isolating the effect of that procedure on downstream decisions. In the market setting, the same fixed panel is exposed during the campaign, so the agent must decide whether a previewed capability justifies its cost and delay.

The experiment manifest in Table 8 fixes the provenance of the reported aggregates. It separates the active policy/capital grid from the agent, forced-skill, and skill-market sweeps, and records when matched comparisons reuse the shared Codex / GPT-5.5 no-skill baseline.

This convention prevents two common ambiguities in interpreting the sweeps. First, matched controls reuse the same hidden replay seeds, so paired effects should be interpreted relative to their control trajectories. Second, skill access is evaluated through operating decisions–branch choice, candidate triage, confirmation timing, and stopping–while the replay substrate remains fixed. The current reference panel emphasizes interpretable public-information policies; future extensions can

Table 9: Exploratory paired comparisons computed from the paper's selected-run export. Differences are in benchmark dollars; positive values favor the first condition.

Comparison	Mean difference	95% interval	n
Codex GPT-5.4 − OpenCode GPT-5.4	\$373.6k	[\$244.6k, \$460.4k]	10
Codex GPT-5.3-codex − OpenCode GPT-5.3-codex	\$295.7k	[\$123.7k, \$435.3k]	10
Marketing market − no market	−\$174.9k	[−\$300.3k, −\$54.3k]	10
Neutral market − no market	−\$176.1k	[−\$324.7k, −\$34.5k]	10

broaden it with additional public-observation planners, including limited-horizon search, belief-state planning, or Monte Carlo tree search.

A.5 Data Provenance and Paired Comparisons

The reported benchmark combines historical assay records with an explicit replay layer. The historical component supplies candidate sequences, measured activities, and round/library provenance; the replay layer supplies hidden-state access, candidate ordering, screening perturbations, action costs, time advancement, reward settlement, and BioSkill interventions. The released runtime artifact contains the machine-readable conversion manifest and the exact configuration used for the paper sweeps.

For the paired comparisons reported in Section 4.3 and Section 4.4, we matched rows by replay seed and computed nominal percentile intervals from 200,000 paired bootstrap resamples. The intervals are descriptive, pointwise, and not corrected for multiple comparisons. They do not establish cross-campaign generalization or robustness to alternative cost and reward calibrations.

B Supplementary Trajectory Audit

The trajectory audit examines how operating decisions unfold before settlement. The operating frontier relates confirmed best activity, final profit, and screening volume across valid episodes, revealing when biological progress is converted into cost-adjusted value.

The forced-skill cash trajectories measure when each condition spends budget and whether that spending is followed by confirmed value. Declines in cash reflect preparation, library opening, screening, confirmation, skill acquisition, and daily burn; upward jumps occur only when confirmed improvements settle reward. These trajectories show that a BioSkill improves the ledger only when its analysis changes experimental commitments in a productive way; a skill can also be neutral or detrimental if it adds delay or encourages unproductive spending.

B.1 Operating Frontier

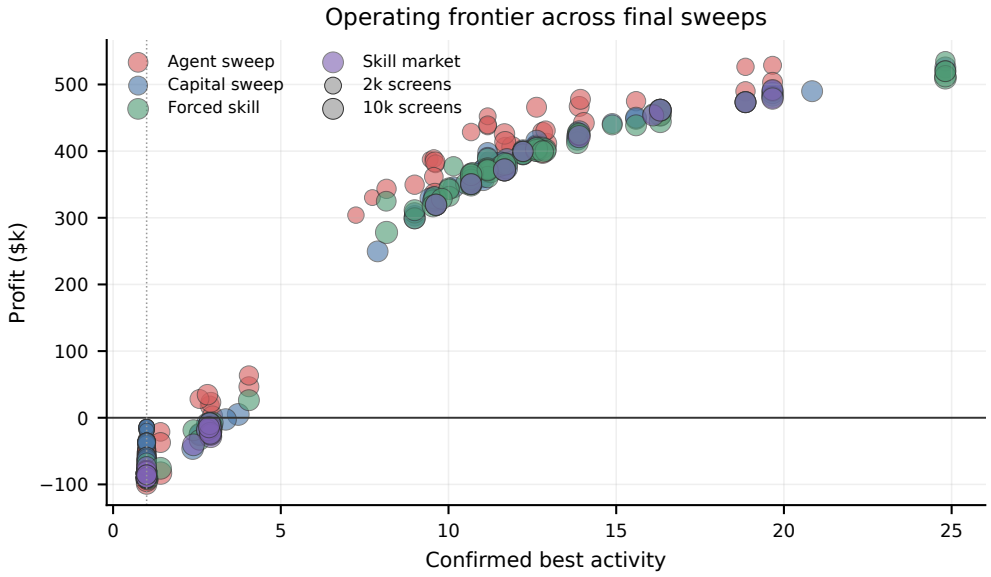

Figure 6: Supplementary operating frontier across capital, skill, market, and agent sweeps. The source CSV includes all final sweep rows; the plotted frontier excludes backend-error episodes so that marker positions summarize completed operating trajectories. Marker size indicates screening volume.

Figure 6 pools valid episodes across the final sweep collection and plots confirmed best activity against final profit, with marker size indicating screening volume. Backend-error episodes are excluded so that the frontier reflects completed operating trajectories rather than runtime failures. This audit complements the main experiments: the capital sweep studies reachability, the harness comparison studies agent separation, and the frontier checks whether biological progress is converted into cost-adjusted value.

The frontier separates three quantities that are otherwise easy to conflate. Confirmed activity records biological progress, but not the cost of reaching, screening, confirming, and stopping around that activity. Screening volume records exploration effort, but not whether the agent preserved enough budget for validation or retained value after settlement. Final profit therefore measures validated progress under cost rather than activity alone.