SCIENCEFLOW: EXECUTABLE-STATE RE-ANCHORING FOR LONG-HORIZON AUTONOMOUS RESEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon LLM agents can improve research outcomes, but their gains often rely on expensive model calls and poorly targeted computation. We introduce **ScienceFlow**, an end-to-end auto-research framework designed to convert limited inference and execution budgets into sustained research progress. ScienceFlow organizes research around recoverable executable states, allowing validated artifacts and intermediate results to persist across iterations. Its Executable-State Transition with Re-Anchoring (ESTRA) mechanism jointly selects an execution anchor from the current or archived states and an extend-or-redirect research direction, enabling continued exploration or redirection from recoverable executable states. An evidence-aware execution controller further admits and monitors jobs according to remaining budget, resource availability, and validated progress. We evaluate ScienceFlow across machine learning, scientific modeling, and mathematical optimization. On the full 75-task MLE-bench, ScienceFlow with DeepSeek-V4-Flash-Preview achieves a $\mathbf{70.22 \pm 1.18}\%$ **Any-Medal rate** under a 24-hour budget per task run, evaluated over three independent benchmark sweeps. ScienceFlow also achieves a low mean API expenditure of **$76.4** per benchmark sweep under our cost-accounting protocol. Source code is provided in the supplementary material.

1 INTRODUCTION

Autoresearch agents have evolved from isolated reasoning and tool use toward complete, long-horizon workflows that formulate hypotheses, modify code, run experiments, interpret evidence, and revise decisions across machine learning and scientific discovery (Lu et al., 2024b; Yamada et al., 2025; Tie et al., 2026; Wei et al., 2025). AlphaEvolve (Novikov et al., 2025), for instance, shows that iterative code refinement guided by evaluator feedback can meaningfully improve algorithms. As these workflows lengthen, the central bottleneck shifts from executing discrete steps to sustaining productive, stable, and goal-aligned progress. Effective agents must preserve research continuity and make adaptive decisions as evidence accumulates, directions evolve, and computational resources become constrained.

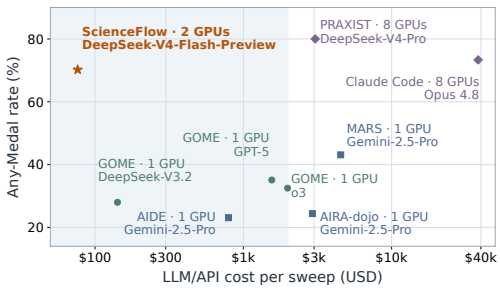


Figure 1: Reported API cost–performance on MLE-bench. PRAXIST and Claude Code report single sweeps; ScienceFlow reports the three-run mean. Details are provided in Appendix D.1.

Sustained progress is complicated by the expanding research state, which grows into a rich executable workspace containing code, datasets, caches, checkpoints, solver states, and evaluation outputs. Textual summaries cannot faithfully reconstruct the executable state needed to resume or revisit prior work, motivating structured memory and persistent workspace designs (Zhu et al., 2026; Chen et al., 2026a; Qian et al., 2026). Moreover, promising directions often consume significant resources before revealing their flaws; apparent progress can stem from noisy signals, implementation artifacts, or strategies that have exhausted their local potential (Toledo et al., 2025). Decisions to continue, branch, or revert directly govern resource deployment, so effective long-horizon research requires state persistence, exploration choices, and execution control to evolve jointly as evidence

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.

accumulates. Prior work tackles isolated facets through disparate representations: structured memory preserves project knowledge (Qian et al., 2026; Chen et al., 2026a), trajectory-centric methods support branching and recovery (Chen et al., 2026c; Zhang et al., 2026b), and resource-aware schedulers manage heterogeneous workloads (Wang et al., 2026; Lu & Reda, 2026). Yet these remain loosely connected, leaving research state, trajectory decisions, and physical execution across different views, a fragmentation that impedes coherent, adaptive, and efficient long-horizon research.

To address this challenge, we introduce **ScienceFlow**, a workspace-grounded autonomous research system supporting one or more homogeneous workers. Each archived state couples executable artifacts with research memory, validation evidence, and resource records. Executable-State Transition with Re-Anchoring (ESTRA) selects both an execution anchor and an extend-or-redirect direction at segment boundaries. Selecting an archived anchor restores its associated workspace and context while resource accounting remains cumulative, enabling continuation or redirection from preserved research states. Parallel workers maintain isolated trajectories and exchange compact progress summaries at these boundaries. An evidence-aware execution controller separately governs job admission, allocation, and runtime review using resource availability, remaining budget, and validated progress. These mechanisms connect state recovery, research-route selection, and physical execution within a common research process.

A key challenge is whether a unified set of long-horizon research abstractions can transfer effectively across fundamentally different executable tasks while using LLM inference cost-effectively. We investigate this question across three domains: machine learning, scientific modeling, and mathematical optimization. On the full 75-task MLE-bench (Chan et al., 2025), ScienceFlow with DeepSeek-V4-Flash-Preview achieves a $70.22 \pm 1.18\%$ Any-Medal rate under a 24-hour budget and reports a low mean API expenditure of \$76.4 per benchmark sweep under our cost-accounting protocol. Figure 1 places this result alongside reported API cost–performance of existing systems under differing evaluation settings. ScienceFlow also achieves strong group-balanced performance on SciModelingBench (SciModelingBench, 2026) and competitive results across multiple mathematical optimization problems. These results demonstrate robust cross-domain performance with modest reported API expenditure. Overall, our contributions are threefold:

- We identify that effective long-horizon research depends not simply on extending interaction time or compute, but on jointly maintaining executable state, adapting research trajectories, and aligning execution with validated progress.
- ScienceFlow realizes this principle through recoverable executable workspaces, ESTRA-governed transitions between research segments, and evidence-aware execution control, enabling research to continue, redirect, or recover without losing useful progress.
- ScienceFlow demonstrates strong and consistent performance across a diverse range of tasks, including machine learning, scientific modeling, and mathematical optimization, with substantial gains on MLE-bench and SciModelingBench and state-of-the-art results across multiple optimization problems.

2 Method

2.1 Preliminaries

We formalize executable research tasks whose outcome quality can be quantified by a goal-dependent utility. A task is defined as $\mathcal{T} = (\mathcal{G}, U_{\mathcal{G}}, V_{\mathcal{G}}, B, W_0)$, where $\mathcal{G}$ specifies the research goal and success criteria, $U_{\mathcal{G}}(\cdot)$ denotes the final task utility, $V_{\mathcal{G}}(\cdot)$ is the validation signal observable during research, B specifies the available resource budget, and W_0 is the initial executable workspace. Starting from W_0, the agent explores a growing collection of recoverable executable states. Let $\mathcal{A}_T$ denote the archive of checkpointed executable states created over a research horizon T. Each executable state $s_v \in \mathcal{A}_T$ represents a concrete checkpoint of the evolving research process, including the executable workspace and its associated artifacts, memory, validation evidence, and resource records. The state archive therefore provides a set of recoverable states from which the agent can continue, revisit, or redirect its research trajectory. Since $U_{\mathcal{G}}$ is often unavailable during execution, the agent uses $V_{\mathcal{G}}$ to guide search and seeks the best validated state within the available budget:

$$\hat{s}_T = \arg\max_{s_v \in \mathcal{A}_T} V_{\mathcal{G}}(s_v) \quad \text{s.t.} \quad C_T = \sum_{t=1}^{T} \rho_t \preceq B, \tag{1}$$

where ρ_t denotes the resource-consumption vector at step t, and C_T is the cumulative resource expenditure. The budget vector $B = (B_1, \ldots, B_K)$ specifies K component-wise limits, such as wall-clock time, compute, and storage. For any $t \leq T$, let $C_t = \sum_{\tau=1}^{t} \rho_\tau$ and $B_t = B - C_t$ denote the cumulative expenditure and remaining resource budget, respectively. The selected state is ultimately assessed by $U_{\mathcal{G}}$.

2.2 System Overview

Long-Horizon Research Process. ScienceFlow extends tool-using inference loops into long-horizon executable research by organizing the process around recoverable executable states. Each state integrates an executable workspace, compact research memory, validation evidence, and resource records, enabling research continuation, trajectory adaptation, and resource-aware execution across iterations. During exploration, a task-specific stage gate determines when to checkpoint the current research state, preserving workspace artifacts and execution results. ScienceFlow then employs ESTRA to transition between research segments by selecting the next research anchor and an extend-or-redirect direction based on available states and accumulated evidence. When an archived state is selected, the corresponding workspace and context are restored before continuing exploration. The resulting framework maintains a persistent research trajectory while enabling dynamic adaptation under evolving evidence and resource constraints. Figure 2 illustrates the overall architecture.

System and Worker Model. ScienceFlow supports one or more homogeneous research workers, each operating within an isolated executable workspace and maintaining its own state archive. Multiple workers explore independent research trajectories and exchange only compact validated progress at research-segment boundaries. They share a physical resource pool governed by a common execution controller, which manages resource availability and runtime execution without participating in scientific route selection.

2.3 Executable Research Process

Long-horizon research is organized as a sequence of research segments. Within each segment, the worker follows its local tool-use policy to inspect, modify, and execute the active workspace. Whenever ScienceFlow detects a task-specific result signal, the stage gate evaluates the result, records the resulting evidence, collects a compact progress summary, and checkpoints the corresponding executable state in the archive. Multiple such checkpoint events may occur within one segment. A text-only response or an approaching context-capacity limit closes the current segment and invokes ESTRA to initialize the next segment from a selected research state and direction. Each design is described in detail in the following subsections.

2.3.1 Research State Representation

We define a research state s_v as an archived milestone in a long-running research process, where v is a unique identifier that enables retrieval from the archive $\mathcal{A}$. Each state records four components: an executable workspace snapshot W_v, a structured research memory m_v, validation evidence $e_v = V_{\mathcal{G}}(s_v)$, and resource records ℓ_v. Specifically,

$$s_v = (W_v, m_v, e_v, \ell_v), \tag{2}$$

where W_v contains executable artifacts and environment metadata required for recovery, m_v maintains persistent long-horizon research memory (detailed in Section 2.3.4), e_v records the validated performance of the state, and ℓ_v tracks resource usage at checkpointing time. States are stored in and retrieved from the archive $\mathcal{A}$ to support future research continuation or re-anchoring.

2.3.2 Forward Research

Each research segment executes a forward research process that may produce multiple checkpointed research states. During this process, the worker follows a local reasoning–action–observation policy to interact with the active workspace. For segment n, ESTRA initializes the fixed base context P_n and workspace $W_{n,0}$ from the selected anchor: a current anchor retains the live workspace, while an archived anchor restores its corresponding snapshot. The context combines anchor-specific memory, evidence, and the selected direction; the first segment starts from the task instructions and initial

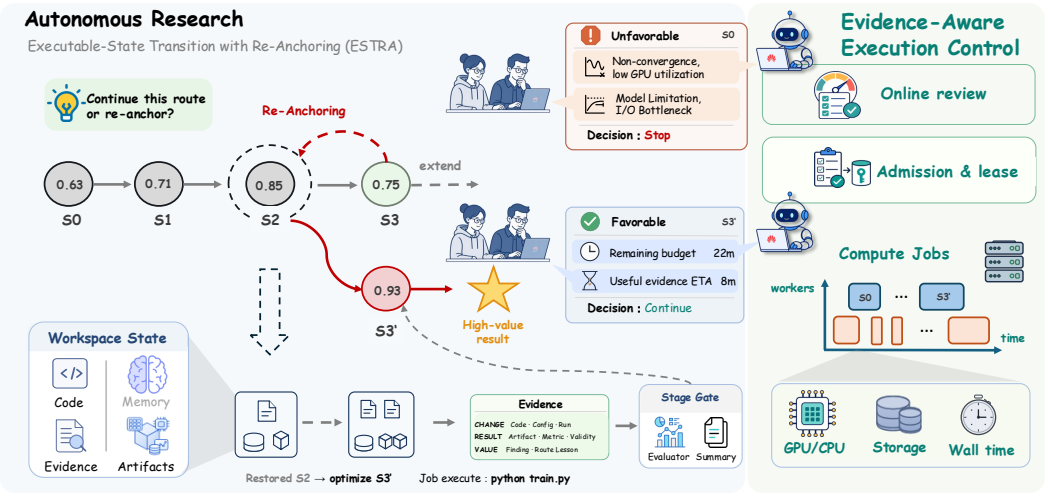


Figure 2: ScienceFlow system architecture. Research workers operate over recoverable executable states and adapt long-horizon trajectories through ESTRA-governed transitions between research segments, while an evidence-aware execution controller coordinates physical resource allocation and runtime execution.

workspace W_0. $W_{n,j}$ denotes the active workspace at round j. $h_{n,j}$ represents the segment-local interaction history (agent context) accumulated up to round j. It captures the reasoning content, tool actions, and observations across rounds and is initialized as $h_{n,0} = \varnothing$. At each round j the policy generates a reasoning output $r_{n,j}$ and tool action $\alpha_{n,j}$:

$$(r_{n,j}, \alpha_{n,j}) \sim \pi_{\text{LLM}}\left(\cdot \mid P_n, h_{n,j}\right). \tag{3}$$

If $\alpha_{n,j} \neq \varnothing$, ScienceFlow executes the action, producing an observation $o_{n,j}$ and an updated workspace:

$$(o_{n,j}, W_{n,j+1}) = \text{Exec}\left(\alpha_{n,j}; W_{n,j}\right). \tag{4}$$

If $\alpha_{n,j} = \varnothing$, the step is text-only; the workspace remains unchanged, $(W_{n,j+1} = W_{n,j})$ and no observation is produced $(o_{n,j} = \varnothing)$. After each round, the local history is updated by appending the reasoning, action, and observation:

$$h_{n,j+1} = h_{n,j} \oplus \left(r_{n,j}, \alpha_{n,j}, o_{n,j}\right), \tag{5}$$

where $\oplus$ denotes ordered concatenation.

Independently of the local interaction policy, ScienceFlow monitors the updated workspace for a task-specific result signal specified by the task adapter. Upon detection, the stage gate collects the result, invokes the evaluator to obtain validation evidence, and requests a compact progress summary from the worker. It then snapshots the workspace as $W_v = \text{Snapshot}(W_{n,j+1})$, records the associated memory m_v, validation evidence e_v, and resource records ℓ_v, and inserts the resulting state $s_v = (W_v, m_v, e_v, \ell_v)$ into the archive $\mathcal{A}_n$. Temporary stage-gate exchanges are excluded from the segment-local history $h_{n,j+1}$, and the worker resumes forward research within the same segment.

Separately, the current research segment closes and transfers control to ESTRA when either of the following conditions for a research-segment boundary is met: 1) The policy produces a text-only response $\alpha_{n,j} = \varnothing$; or 2) The accumulated context exceeds the configured capacity threshold $|P_n \oplus h_{n,j+1}| \geq L_{\text{ctx}}$. If neither condition holds, the forward loop proceeds to round $j + 1$. Otherwise, the live workspace at the trigger defines the current state s_n^{cur}, and ScienceFlow initiates the route-level deliberation described next. When a result signal and the context-capacity condition coincide, ScienceFlow completes stage-gate processing first, making the newly checkpointed state available to the subsequent ESTRA decision.

2.3.3 Re-Anchoring and Exact Restoration

At each research-segment boundary, ScienceFlow invokes ESTRA for route-level deliberation by the research worker. The worker policy π_{LLM} produces an anchor-and-direction decision (a_n, d_n).

This decision operates on two independent axes: a_n selects the execution anchor, either the current trigger state s_n^{cur} or a previously archived state $s_v \in \mathcal{A}_n$, while d_n determines whether to extend the current trajectory or redirect onto a new branch from that anchor. The worker bases this decision on the compact memory view, available anchor candidates with their validation evidence, and the current resource envelope. Formally:

$$a_n \in \{s_n^{\text{cur}}\} \cup \mathcal{A}_n, \qquad d_n \in \{\textsc{extend}, \textsc{redirect}\}. \tag{6}$$

Combining these two choices yields four possible outcomes: extending or redirecting from the current state, and extending or redirecting from an archived state. After the ESTRA decision, the next research segment $n+1$ is initialized by defining the segment-level base context P_{n+1} and workspace $W_{n+1,0}$ as:

$$(P_{n+1}, W_{n+1,0}) = \text{ESTRA}\left(a_n, d_n; \mathcal{A}_n\right). \tag{7}$$

If a_n selects the current executable state, ESTRA retains the current workspace $W_{n,j+1}$ as $W_{n+1,0}$. If a_n selects an archived executable state (namely $s_v \in \mathcal{A}_n$), ESTRA restores the full state, i.e., workspace W_v, memory m_v, validation evidence e_v, and resource records ℓ_v, with resource accounting remaining cumulative across restoration, including resources consumed after archiving. The anchor thus determines the workspace from which research resumes, while d_n determines whether research extends the anchor's existing route or redirects from it. The resulting P_{n+1} carries the anchor's research summary, relevant evidence, and selected direction, while $W_{n+1,0}$ provides the corresponding executable artifacts. ESTRA preserves the full archive while cleanly setting both the starting state and research direction for the next segment.

2.3.4 Memory and Context Assembly

ScienceFlow maintains persistent research memory through four operations: Add, Fold, Unfold, and Assemble. At each stage-gate event, Add appends a compact result card containing evaluated results, relevant artifacts, and current progress while preserving the segment-local interaction history. Fold keeps the memory view within the context budget by retaining recent, best-validated, and anchor-relevant information and summarizing older records; the complete history remains archived and indexed for recovery. It also summarizes completed exploration whenever an ESTRA transition closes a research segment, providing historical evidence for future decisions.

During anchor selection, Unfold retrieves detailed records associated with archived summaries or result cards when additional evidence is needed, temporarily augmenting the compact view while preserving the persistent memory. Assemble combines the selected memory view, retrieved history, evidence, execution logs, and workspace state with stable task instructions and transition directions to support continuation or redirection from the selected anchor. Together, these operations preserve a recoverable research trajectory within a bounded context. Formal definitions and implementation details are provided in Appendix B.

2.4 Evidence-Aware Execution Control

A key distinction that separates ScienceFlow from existing autonomous research agents is the explicit separation of research-route selection from physical execution control. Specifically, the research worker proposes executable jobs, while a dedicated evidence-aware execution controller determines whether, where, and for how long each job runs. This controller evaluates pending and running jobs based on current resource availability, remaining budget, validated research progress, and the recoverable state of execution artifacts. Accordingly, the control process operates in two stages: admission and allocation before a job is launched, and online review and replanning during execution.

2.4.1 Execution Admission and Allocation

For each proposed job b, ScienceFlow constructs a resource request specifying its command, device class and count, estimated runtime, memory requirements, and worker-provided research-value hints. These hints and estimates from the execution context determine admission priority based on expected research value, lineage diversity, proximity to a deliverable, worker starvation, duplication, and runtime risk. The admission-and-lease module evaluates the request against remaining budget B_t and live resource state R_t, including GPU/CPU capacity, storage constraints, queue state, GPU

pressure, and active leases. Confident preflight results are applied directly; otherwise, an isolated admission LLM reviews the request, research-value information, and preflight result to produce:

$$\delta_b^{\text{adm}} = \text{Admit}(b; R_t, B_t) \in \{\texttt{RUN_NOW}, \texttt{OBSERVE_THEN_RUN}, \texttt{PENDING}, \texttt{REPLAN}\}. \tag{8}$$

Each decision determines the execution policy: RUN_NOW executes immediately after acquiring a device lease, OBSERVE_THEN_RUN performs execution with early validation under uncertainty, PENDING defers execution in the priority queue, and REPLAN triggers plan revision using blocking evidence. Device leases ensure controlled resource access and are released upon job completion, termination, or resource release.

2.4.2 ONLINE REVIEW AND REPLAN

After admission, ScienceFlow reviews jobs at observation boundaries using timestamped process, log, metric, artifact, utilization, checkpoint, and budget evidence. Liveness, log growth, and sustained utilization indicate activity; improving metrics, validated results, and recoverable artifacts indicate research progress. The controller estimates the time to the next comparable metric or recoverable artifact (useful-evidence ETA), favoring continuation when it fits the remaining budget and recent progress supports it. Persistent non-convergence, low GPU utilization, model limitations, or I/O bottlenecks support stop-and-replan when useful evidence is unlikely within budget and no recoverable value has emerged. If research value remains uncertain, the worker provides an advisory—CONTINUE, TIMEBOX CONTINUE, SAFE TO STOP, REPLAN, or UNKNOWN—with confidence, a measurable commitment, and the expected next artifact. The controller retains authority over leases, timeboxing, and termination, combining execution evidence $y_{0:t}$, the optional advisory g_t, and remaining budget B_t to determine:

$$c_t = \text{Review}(y_{0:t}, g_t; B_t) \in \{\texttt{CONTINUE}, \texttt{TIMEBOX}, \texttt{STOP_AND_REPLAN}\}. \tag{9}$$

`CONTINUE` keeps the job running; `TIMEBOX` grants a bounded window with a measurable progress condition; `STOP_AND_REPLAN` terminates execution after preserving recoverable artifacts and releasing leases. Both `REPLAN` and `STOP_AND_REPLAN` return blocking or runtime evidence to the research worker for scientific plan revision, while ESTRA separately selects the anchor a_n and direction d_n. Together, Admit before launch and repeated Review at observation boundaries form a closed execution-control loop that allocates compute according to research value until completion or handoff to the worker.

3 EXPERIMENTS

3.1 EXPERIMENT SETTINGS

On 75-task MLE-bench (Chan et al., 2025), ScienceFlow uses DeepSeek-V4-Flash-Preview for 24 hours with two research workers sharing a pool of up to two GPUs; Any-Medal rates are mean $\pm$ SEM over three independent runs, with test labels withheld from the agent. Circle Packing, Ratio Minimization, and Uncertainty Inequality use 12-hour runs with two workers and artifact-based evaluation; we test DeepSeek-V4-Flash-Preview and openPangu-2.0-Pro and rerun OpenEvolve (Sharma, 2025) locally. SpOC4 KTTSP follows the official competition protocol. The 12-task SciModelingBench (SciModelingBench, 2026) uses offline observations, two-hour budgets, and 8–20 batch evaluations; metrics are best-K mean, normalized enrichment, or global NDCG, with six DrugMatrix endpoints grouped for aggregation. On SciModelingBench, each system–task pair is evaluated once, reporting its best valid submission. Protocols appear in Appendices D.1, E.2, F.1, and G.

3.2 MAIN RESULTS

MLE-bench Results. Table 1 presents representative MLE-bench results, emphasizing repeated evaluations; Figure 1 separately includes configurations with matched public API-cost records, including single-sweep results. Across the full 75-task benchmark, ScienceFlow achieves an Any-Medal rate of $70.22 \pm 1.18\%$, exceeding the strongest baseline in Table 1 by 4.92 pp. The three independent runs obtain medals on 54, 53, and 51 tasks, respectively, demonstrating consistent

Table 1: Representative results on the full 75-task MLE-bench. Any-Medal rates (%) are reported as mean ± SEM when three runs are available. Best results in each column are shown in **bold**.

Agent	LLM(s)	Lite (%)	Medium (%)	High (%)	All (%)
		Other reported systems			
Famou-Agent	Gemini-2.5-Pro	75.76 ± 1.52	57.89 ± 1.52	40.00 ± 0.00	59.56 ± 0.89
CAIR MARS+	Gemini-3-Pro-Preview	78.79 ± 1.52	60.53 ± 1.52	44.44 ± 2.22	62.67 ± 0.77
AIBuildAI	Claude-Opus-4.6	77.27 ± 0.00	61.40 ± 0.88	46.67 ± 0.00	63.11 ± 0.44
Famou-Agent 2.0	Gemini-3-Pro-Preview	**80.30 ± 1.52**	64.04 ± 2.32	42.22 ± 2.22	64.44 ± 1.18
Iris	Claude-Opus-4.6	**80.30 ± 1.50**	64.00 ± 0.90	44.40 ± 2.20	64.90 ± 0.40
		Open-source baselines			
ML-Master 2.0	DeepSeek-V3.2-Speciale	75.76 ± 1.51	50.88 ± 3.51	42.22 ± 2.22	56.44 ± 2.47
MLEvolve	Gemini-3-Pro-Preview	**80.30 ± 1.52**	57.89 ± 1.52	42.22 ± 2.22	61.33 ± 1.33
PiEvolve	Gemini-3-Pro-Preview	**80.30 ± 1.52**	58.77 ± 0.88	40.00 ± 0.00	61.33 ± 0.77
MLEvolve	Gemini-3.1-Pro-Preview	**80.30 ± 1.50**	64.00 ± 0.90	**46.70 ± 0.00**	65.30 ± 0.80
ScienceFlow (Ours)	**DeepSeek-V4-Flash-Preview**	**80.30 ± 1.52**	**74.56 ± 0.88**	44.44 ± 2.22	**70.22 ± 1.18**

Table 2: Audited mathematical optimization scores. Bold and underline indicate the best and second-best results at the displayed precision.

Method	Model	Circle packing ↑	Ratio min. ↓	Hermite C_4 ↓
AlphaEvolve	Gemini Pro/Flash	2.6358627564	3.590162407473	0.352099104423
ShinkaEvolve	Sonnet 4/GPT-4.1	2.6359828390	–	–
ThetaEvolve	DS-R1/Qwen3-8B	2.6359830774	–	–
MLEvolve	Gemini 3.1 Pro	2.6359830395	**3.590157365311**	0.352099104416
FM Agent	Gemini 2.5 Pro	–	3.590157406159	0.352099104416
OpenEvolve	DS-V4-Flash	2.6344194866	3.658667107456	0.352581132500
ScienceFlow	openPangu-2.0-Pro	2.6359824748	3.590157365325	**0.343293122432**
ScienceFlow	DS-V4-Flash	**2.6359830849**	**3.590157365311**	0.348200107555

Notes: Packing uses a unit square, and the ratio task uses 16 planar points. DS denotes DeepSeek. Gemini 3.1 Pro and DS-V4-Flash are Preview variants. AlphaEvolve uses Gemini 2.0 Pro + Flash for the first two tasks.

performance across runs. The improvement is particularly strong on the Medium tier, where ScienceFlow reaches $74.56 \pm 0.88\%$, 10.52 pp above the strongest baseline in Table 1. It also achieves $80.30 \pm 1.52\%$ on Lite, matching the best listed result, and $44.44 \pm 2.22\%$ on High, within 2.26 pp of the strongest listed baseline. Complete MLE-bench results and evaluation details appear in Appendix D.1, including the full leaderboard in Table A5. Case-study records are provided in Appendix D.2.

Math Optimization Results. Table 2 compares audited scores on three mathematical optimization tasks. ScienceFlow achieves the highest Circle Packing score, although its small margin over ThetaEvolve (Wang et al., 2025b) is best interpreted as a numerical near tie. It also matches MLEvolve (Du et al., 2026) on Ratio Minimization at the displayed precision and reduces the best listed Hermite C_4 upper bound by 2.5%. DeepSeek-V4-Flash-Preview yields the stronger ScienceFlow results on the first two tasks, while openPangu-2.0-Pro performs better on the uncertainty-bound task.

KTTSP Competition Results. ScienceFlow searches target order and transfer timing in ESA's SpOC4 KTTSP challenge, using official evaluations, archived results, and peer summaries to guide refinement. In the July 6, 2026 leaderboard snapshot, it achieves 116.911, 234.929, and 393.229 mission days on the easy, medium, and hard tracks, ranking 10th, 8th, and 3rd, respectively. The hard track uses a ten-day campaign. The detailed results of KTTSP are provided in Appendix F.

Table 3: SciModelingBench group-balanced pass1 results.

Agent	Score ↑	Rank ↓
Pi	51.39	2.79
Claude Code	51.37	3.69
Codex	50.39	3.64
OpenCode	44.21	3.33
ScienceFlow	**54.41**	**1.55**

Scientific Modeling Results. Table 3 reports the group-balanced pass1 results on SciModelingBench. ScienceFlow achieves the highest normalized score of 54.41, exceeding the strongest baseline, Pi (51.39), by 3.02 points. It also obtains the best average rank of 1.55, compared with 2.79 for Pi. At the task level, ScienceFlow

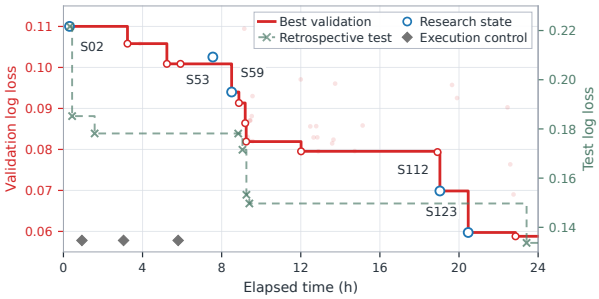


Figure 3: A 24-hour ScienceFlow trajectory on the Statoil Iceberg task.

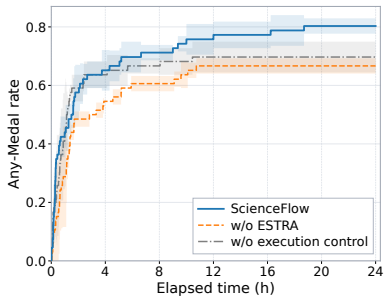


Figure 4: MLE-bench Lite ablation: mean ± SD (3 seeds).

ranks first on five tasks and second on six, placing in the top two on 11 of the 12 tasks. For example, on UTR MRL, which selects RNA sequences with high mean ribosome load, ScienceFlow achieves a normalized enrichment of 0.7605, ahead of Pi's 0.7533. On the DrugMatrix Phosphorus task, which ranks treatment conditions by the magnitude of their phosphorus response, ScienceFlow reaches a global NDCG of 0.5985, exceeding the runner-up Pi (0.4917) by 0.1068. Creatinine is its only task outside the top two.

3.3 Case Study and Ablations

Long-horizon case study. On the *Statoil Iceberg Classifier Challenge*, ScienceFlow distinguishes icebergs from ships in satellite SAR imagery under a 24-hour budget. Figure 3 traces one run, showing validation progress alongside recoverable workspace states, ESTRA transitions, and evidence-aware execution control. Retrospective test scores are used only for analysis. Appendix D.2 provides the corresponding search and resource-control records.

ESTRA and Execution Control. We ablate ESTRA and Execution Control on the official 22-task MLE-bench Lite split under a 24-hour budget. All other settings are fixed, with variant definitions in Appendix C.4. Figure 4 shows the cumulative first-medal rate, where a task is counted once it first reaches Bronze, Silver, or Gold. Curves report the mean over three seeds, with shaded bands indicating sample standard deviation. At 24 hours, the full system reaches an Any-Medal rate of $80.30 \pm 2.62\%$, compared with $66.67 \pm 2.62\%$ without ESTRA and $69.70 \pm 5.25\%$ without execution control. The difference emerges early: by 12 hours, ScienceFlow reaches $77.27 \pm 4.55\%$, while the two ablations remain at $66.67 \pm 2.62\%$ and $69.70 \pm 5.25\%$, respectively. The full system therefore improves both the final Any-Medal rate and the speed of medal acquisition.

State-matched re-anchoring study. To examine re-anchoring effects in this single-task case study, we replay seven historical ESTRA decisions from captured executable states, crossing current or archived anchors with extend or redirect directions. Three seeds and matched four-hour budgets yield 84 replay branches (Figure 5). The historical action achieves the best mean score, including ties, at five of seven points, with zero median regret and a mean regret of 1.725×10^{-3} AUROC against the replay oracle. The two exceptions highlight execution reliability and remaining search horizon: at `P04`, archived-redirect underperforms in all three replays, although the original trajectory improves after 5.26 hours; at `P07`, extending the current trajectory near the budget limit is outperformed by restoring a strong archived state. These results suggest that re-anchoring should jointly consider transition reliability, remaining budget, and archived-state quality. A four-backbone comparison on the same task examines ScienceFlow's sensitivity to model choice and performance–cost trade-offs (Appendix D.3).

4 Related Work

Scientific Agents. Existing scientific agents can be characterized by their pipeline structures and levels of autonomy (Tie et al., 2026; Chen et al., 2025; Wei et al., 2025; Eger et al., 2026). Systems span ML (Lu et al., 2024b; Yamada et al., 2025), wet-lab research (Boiko et al., 2023; Swanson et al., 2025), and self-evolving pipelines (Novikov et al., 2025), using refinement (Baek et al., 2025), reflection (Shinn et al., 2023), and skill libraries (Wang et al., 2023) across software engineering (An-

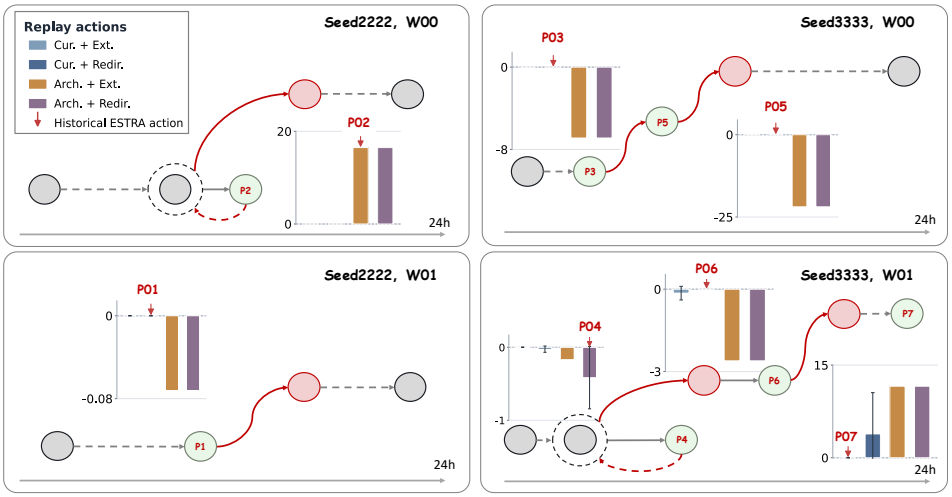


Figure 5: **State-matched replay on Tabular Playground Series May 2022.** Bars show mean private-test AUROC changes from the source state; error bars indicate sample SD across three seeds. Red arrows mark historical ESTRA actions.

toniades et al., 2025), algorithm design (Romera-Paredes et al., 2024; Lu et al., 2024a; Faldor et al., 2025; Hazra et al., 2025), and open-ended discovery (O'Neill et al., 2025). Reproducibility, novelty validation, modularity, and experimental memory remain challenges (Tie et al., 2026; Wei et al., 2025; Eger et al., 2026).

MLE Agents. MLE-bench (Chan et al., 2025) evaluates agents on 75 Kaggle tasks under 24-hour GPU budgets, motivating advances beyond AIDE (Jiang et al., 2025) along three directions. Search-based agents use MCTS (Liu et al., 2025), reflection (MARS) (Chen et al., 2026b), graph recombination (MLEvolve) (Du et al., 2026), and epistemic loops (Iris) (Fu et al., 2026). Multi-agent systems organize research and execution through specialized roles (Yang et al., 2025a; InternAgent Team, 2025; Zhang et al., 2026a), island models (Li et al., 2025), and web-search with ablation (Nam et al., 2025). Memory architectures use cognitive caches (Zhu et al., 2026), Git-native graphs (Nadafian et al., 2026), file-based coordination (Chen et al., 2026a), hypothesis trees (Jin et al., 2026), and environment-centric designs (Xin et al., 2026), alongside work on ideation, memory tiers, and risks (Zhang et al., 2026d; Chopde et al., 2025; Ward et al., 2025). Related mechanisms include artifact storage (AutoSci) (Qian et al., 2026), state-tree revision (MAGE) (Chen et al., 2026c), trajectory refinement (PIVOT) (Zhang et al., 2026b), and telemetry scheduling (Wang et al., 2026; Lu & Reda, 2026).

Optimization Agents. Optimization agents tackle numerical and combinatorial problems through history-based prompting (OPRO) (Yang et al., 2024), algorithm design and evolution (Pandit et al., 2025; Yang et al., 2025b; Suwandi et al., 2025; Lange et al., 2024), and hierarchical multi-agent workflows (Guo et al., 2025; Baumann & Kramer, 2025).

Despite progress in scientific agents, existing approaches mainly improve specific capabilities, such as search, collaboration, memory, or optimization. However, sustaining long-horizon research requires coordinating persistent states, adaptive trajectory decisions, and resource-aware execution throughout an evolving process. These challenges remain important for developing agents that can support continuous research beyond isolated iterations.

5 Conclusion

We introduced **ScienceFlow**, a workspace-grounded autonomous research system with recoverable states, ESTRA-governed transitions, evidence-aware execution control, and configurable isolated workers. Evaluated on ML engineering, optimization, and scientific modeling, ScienceFlow achieves $70.22 \pm 1.18\%$ Any-Medal on MLE-bench, matches top optimization results (improving a Hermite bound by 2.5%), and obtains a group-balanced score of 54.41 on SciModelingBench, demonstrating strong observed performance across the evaluated domains.

AI Use Statement

We used generative AI tools to support research ideation and execution, including methodological clarification, and interpretation of experimental results. AI tools also assisted with literature retrieval and discovery, reference formatting, drafting selected sections, and improving manuscript clarity and fluency.

The authors reviewed and revised AI-assisted drafts and experiments, verified references against original sources, and checked reported numerical comparisons against experimental results.

References

Anthropic. Claude Code. npm software release, 2026. URL https://www.npmjs.com/package/@anthropic-ai/claude-code/v/2.1.72. Version 2.1.72; accessed August 12, 2026.

Antonis Antoniades, Albert Örwall, Kexun Zhang, Yuxi Xie, Anirudh Goyal, and William Yang Wang. SWE-search: Enhancing software agents with monte carlo tree search and iterative refinement. In *International Conference on Learning Representations*, 2025.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002. URL https://api.semanticscholar.org/CorpusID:207609497.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 conference of the nations of the Americas chapter of the association for computational linguistics: human language technologies (volume 1: long papers)*, pp. 6709–6738, 2025.

Luis A. Barrera, Anastasia Vedenko, Jesse V. Kurland, et al. Survey of variation in human transcription factors reveals prevalent DNA binding changes. *Science*, 351(6280):1450–1454, 2016. doi: 10.1126/science.aad2257.

Jill Baumann and Oliver Kramer. An llm-based multi-agent framework for evolutionary blackbox optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '25 Companion, pp. 671–674, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714641. doi: 10.1145/3712255.3726575.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. ISSN 1476-4687.

Dmitry Bolotin. Local fitness landscape of the green fluorescent protein. figshare dataset, 2016. URL https://doi.org/10.6084/m9.figshare.3102154.v1. Dataset; CC BY 4.0.

Sai Kiran Botla, Kirubanath Sankar, Abhishek Chopde, and Fardeen Pettiwala. Pi-Evolve: Long-horizon evolutionary optimization for autonomous scientific discovery. GitHub software, 2025. URL https://github.com/FractalAIResearchLabs/PiEvolve. Accessed August 12, 2026.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. Mle-bench: Evaluating machine learning agents on machine learning engineering. *International Conference on Learning Representations*, 2025.

Guoxin Chen, Jie Chen, Lei Chen, Jiale Zhao, Fanzhe Meng, Wayne Xin Zhao, Ruihua Song, Cheng Chen, Ji-Rong Wen, and Kai Jia. Toward autonomous long-horizon engineering for ml research. *arXiv preprint arXiv:2604.13018*, 2026a.

Jiefeng Chen, Bhavana Dalvi Mishra, Jaehyun Nam, Rui Meng, Tomas Pfister, and Jinsung Yoon. Mars: Modular agent with reflective search for automated ai research. *arXiv preprint arXiv:2602.02660*, 2026b.

Qiguang Chen, Mingda Yang, Libo Qin, Jinhao Liu, Zheng Yan, Jiannan Guan, Dengyun Peng, Yiyan Ji, Hanjing Li, Mengkang Hu, Yimeng Zhang, Yihao Liang, Yuhang Zhou, Jiaqi Wang, Zhi Chen, and Wanxiang Che. AI4Research: A Survey of Artificial Intelligence for Scientific Research, 2025. arXiv:2507.01903.

Yaoqi Chen, Haibin Lai, Yuru Feng, Chuyu Han, et al. Beyond semantic organization: Memory as execution state management for long-horizon agents. *arXiv preprint arXiv:2606.06090*, 2026c.

Abhishek Chopde, Fardeen Pettiwala, Sankar Kirubananth, Sai Kiran Botla, and Pachipulusu Ayyappa Kethan. PiML: Automated machine learning workflow optimization using LLM agents. In *Proceedings of the Fourth International Conference on Automated Machine Learning*, volume 293 of *Proceedings of Machine Learning Research*, pp. 1/1–42. PMLR, 2025. URL https://proceedings.mlr.press/v293/chopde25a.html.

Shangheng Du, Xiangchao Yan, Jinxin Shi, Zongsheng Cao, Shiyang Feng, Zichen Liang, Boyuan Sun, Tianshuo Peng, Yifan Zhou, Xin Li, Jie Zhou, Liang He, Bo Zhang, and Lei Bai. Mlevolve: A self-evolving framework for automated machine learning algorithm discovery. *arXiv preprint arXiv:2606.06473*, 2026.

Steffen Eger, Yong Cao, Jennifer D'Souza, Andreas Geiger, Christian Greisinger, Stephanie Gross, Yufang Hou, Brigitte Krenn, Anne Lauscher, Yizhi Li, Chenghua Lin, Nafise Sadat Moosavi, Wei Zhao, and Tristan Miller. Transforming Science with Large Language Models: A Survey on AI-assisted Scientific Discovery, Experimentation, Content Generation, and Evaluation, 2026. arXiv:2502.05151.

European Space Agency, Advanced Concepts Team. SpOC 4: Space Logistics. European Space Agency competition webpage, 2026. URL https://www.esa.int/gsp/ACT/news/spoc-2026/. Accessed August 12, 2026.

Maxence Faldor, Jenny Zhang, Antoine Cully, and Jeff Clune. OMNI-EPIC: Open-endedness via models of human notions of interestingness with environments programmed in code. In *International Conference on Learning Representations*, 2025.

Fordyce Lab. BET-seq Processed Data. figshare dataset, 2018. URL https://doi.org/10.6084/m9.figshare.5728467.v1. Dataset; CC BY 4.0.

Shaokang Fu, Yulong Tao, Linbo Jin, Jiarong Zhao, Qiming Shi, Tianjun Pan, Haonan Li, Chengyu Wang, Jia Wu, and Chengfu Huo. Beyond solution-centric search: Adaptive inquiry and knowledge revision for autonomous ML engineering. *arXiv preprint arXiv:2608.02143*, 2026.

Felipe Gonçalves, Diogo Oliveira e Silva, and Stefan Steinerberger. Hermite polynomials, linear flows on the torus, and an uncertainty principle for roots. *Journal of Mathematical Analysis and Applications*, 451(2):678–711, 2017. doi: 10.1016/j.jmaa.2017.02.030. URL https://doi.org/10.1016/j.jmaa.2017.02.030.

Hongshu Guo, Zeyuan Ma, Yining Ma, Xinglin Zhang, Wei-Neng Chen, and Yue-Jiao Gong. DesignX: Human-competitive algorithm designer for black-box optimization. *Advances in Neural Information Processing Systems*, 38:6582–6615, 2025.

Kam Hamidieh. A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154:346–354, 2018a. doi: 10.1016/j.commatsci.2018.07.052.

Kam Hamidieh. Superconductivty data. UCI Machine Learning Repository, 2018b. URL https://doi.org/10.24432/C53P47. Dataset; CC BY 4.0.

Rishi Hazra, Alkis Sygkounas, Andreas Persson, Amy Loutfi, and Pedro Zuidberg Dos Martires. REvolve: Reward evolution with large language models using human feedback. In *International Conference on Learning Representations*, 2025.

HeyNeo Team. Neo: Next-generation ai agents. https://heyneo.so/blog, 2025.

Maxwell A. Hume, Luis A. Barrera, Stephen S. Gisselbrecht, and Martha L. Bulyk. UniPROBE, update 2015: New tools and content for the online database of protein-binding microarray data on protein–DNA interactions. *Nucleic Acids Research*, 43(D1):D117–D122, 2015. doi: 10.1093/nar/gku1045.

InternAgent Team. Internagent: When agent becomes the scientist – building closed-loop system from hypothesis to verification. *arXiv preprint arXiv:2505.16938 [cs.AI]*, 2025.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. Aide: Ai-driven exploration in the space of code. *arXiv preprint arXiv:2502.13138*, 2025.

Jiajie Jin, Yuyang Hu, Kai Qiu, Qi Dai, Chong Luo, Guanting Dong, Xiaoxi Li, Tong Zhao, Xiaolong Ma, Gongrui Zhang, et al. Toward generalist autonomous research via hypothesis-tree refinement. *arXiv preprint arXiv:2606.11926*, 2026.

Robert Lange, Yingtao Tian, and Yujin Tang. Large language models as evolution strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 579–582, 2024.

Daniel D. Le, Tyler C. Shimko, Arjun K. Aditham, Allison M. Keys, Scott A. Longwell, Yaron Orenstein, and Polly M. Fordyce. Comprehensive, high-resolution binding energy landscapes reveal context dependencies of transcription factor binding. *Proceedings of the National Academy of Sciences*, 115(16):E3702–E3711, 2018. doi: 10.1073/pnas.1715888115.

Annan Li, Chufan Wu, Zengle Ge, Yee Hin Chong, Zhinan Hou, Lizhe Cao, Cheng Ju, Jianmin Wu, Huaiming Li, Haobo Zhang, Shenghao Feng, Mo Zhao, Fengzhi Qiu, Rui Yang, Mengmeng Zhang, Wenyi Zhu, Yingying Sun, Quan Sun, Shunhao Yan, Danyu Liu, Dawei Yin, and Dou Shen. The FM agent. *CoRR*, abs/2510.26144, 2025.

Jin Li, Ahmed Murtadha, Zhiyu Wang, Qiwen Chen, William Chen, Yifei Wu, Guan Wang, Andy L. Siy, Jiayi Yang, Mengsha Huang, Wenhao Li, Yixuan Liu, Shuailin Pan, Mingli Yuan, Sen Song, and Yuhao Sun. Praxist: From experimental artifacts to solution lineages. *arXiv preprint arXiv:2608.25955*, 2026. URL https://arxiv.org/abs/2608.25955.

Zexi Liu, Yuzhu Cai, Xinyu Zhu, Yujie Zheng, Runkun Chen, Ying Wen, Yanfeng Wang, Siheng Chen, et al. Ml-master: Towards ai-for-ai via integration of exploration and reasoning. *arXiv preprint arXiv:2506.16499*, 2025.

Chris Lu, Samuel Holt, Claudio Fanconi, Alex James Chan, Jakob Nicolaus Foerster, Mihaela van der Schaar, and Robert Tjarko Lange. Discovering preference optimization algorithms with and for large language models. In *Advances in Neural Information Processing Systems*, 2024a.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery, 2024b. arXiv:2408.06292.

Tianxi Lu and Sherief Reda. Agentic cpu-gpu scheduling for heterogeneous ai workloads. *arXiv preprint arXiv:2607.22242*, 2026.

Alireza Nadafian, Alireza Mohammadshahi, and Majid Yazdani. Kapso: A knowledge-grounded framework for autonomous program synthesis and optimization. *arXiv preprint arXiv:2601.21526*, 2026.

Jaehyun Nam, Jinsung Yoon, Jiefeng Chen, Jinwoo Shin, Sercan Ö Arık, and Tomas Pfister. Mle-star: Machine learning engineering agent via search and targeted refinement. *arXiv preprint arXiv:2506.15692*, 2025.

National Toxicology Program (NTP). DrugMatrix. Chemical Effects in Biological Systems (CEBS), 2023. URL https://doi.org/10.22427/NTP-DATA-107-022-001-000-3.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. Alphaevolve: A coding agent for scientific and algorithmic discovery, 2025. URL https://arxiv.org/abs/2506.13131.

Charles O'Neill, Tirthankar Ghosal, Roberta Răileanu, Mike Walmsley, Thang Bui, Kevin Schawinski, and Ioana Ciucă. Sparks of science: Hypothesis generation using structured paper data. *arXiv preprint arXiv:2504.12976*, 2025.

OpenAI. Codex CLI. npm software release, 2026a. URL https://www.npmjs.com/package/@openai/codex/v/0.144.5. Version 0.144.5; accessed August 12, 2026.

OpenAI. MLE-bench Leaderboard. GitHub repository, 2026b. URL https://github.com/openai/mle-bench. Accessed August 12, 2026.

OpenCode Contributors. OpenCode: The open source AI coding agent. npm software release, 2026. URL https://www.npmjs.com/package/opencode-ai/v/1.18.4. Version 1.18.4; accessed August 12, 2026.

Sujay Pandit, Akanksha Jain, Rami Cohen, Zhijie Deng, Sagar Karandikar, Sagi Perel, Anand Raghunathan, and Parthasarathy Ranganathan. LLM-box : An agentic framework for guided black-box optimization in mapping LLMs onto specialized hardware accelerators. In *Machine Learning for Systems 2025*, 2025.

Weitong Qian, Beicheng Xu, Zhongao Xie, Bowen Fan, et al. Autosci: A memory-centric agentic system for the full scientific research lifecycle. *arXiv preprint arXiv:2605.31468*, 2026.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. ISSN 1476-4687.

Paul J. Sample, Ban Wang, and Georg Seelig. Human 5' UTR design and variant effect prediction from a massively parallel translation assay. NCBI Gene Expression Omnibus, 2018. URL https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE114002. Dataset; accession GSE114002.

Paul J. Sample, Ban Wang, David W. Reid, Vlad Presnyak, Iain J. McFadyen, David R. Morris, and Georg Seelig. Human 5' UTR design and variant effect prediction from a massively parallel translation assay. *Nature Biotechnology*, 37(7):803–809, 2019. doi: 10.1038/s41587-019-0164-5.

Karen S. Sarkisyan, Dmitry A. Bolotin, Margarita V. Meer, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016. doi: 10.1038/nature17995.

SciModelingBench. SciModelingBench Design-Bench Data. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/sci-modeling-bench/design-bench. Hugging Face dataset, version 0.10.0.

Asankhaya Sharma. OpenEvolve: An open-source evolutionary coding agent. GitHub software, 2025. URL https://github.com/codelion/openevolve. Accessed August 12, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023.

Richard Cornelius Suwandi, Feng Yin, Juntao Wang, Renjie Li, Tsung-Hui Chang, and Sergios Theodoridis. Adaptive kernel design for bayesian optimization is a piece of CAKE with LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, and James Zou. The virtual lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 646:716–723, 2025.

Guiyao Tie, Pan Zhou, and Lichao Sun. A Survey of AI Scientists, 2026. arXiv:2510.23045.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Baldwin, Alexis Audran-Reiss, Michael Kuchnik, Despoina Magka, Minqi Jiang, Alisia Maria Lupidi, et al. Ai research agents for machine learning: Search, exploration, and generalization in mle-bench. *arXiv preprint arXiv:2507.02554*, 2025.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-Bench: Benchmarks for data-driven offline model-based optimization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21658–21676. PMLR, 2022. URL https://proceedings.mlr.press/v162/trabucco22a.html.

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.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Openhands: An open platform for ai software developers as generalist agents. *International Conference on Learning Representations*, 2025a.

Yifei Wang, Hancheng Ye, Yechen Xu, Cong Guo, Chiyue Wei, Qinsi Wang, Dongting Li, Tingjun Chen, Hai Li, Danyang Zhuo, and Yiran Chen. Mars: Efficient, adaptive co-scheduling for heterogeneous agentic systems. *arXiv preprint arXiv:2604.26963*, 2026.

Yiping Wang, Shao-Rong Su, Zhiyuan Zeng, Eva Xu, Liliang Ren, Xinyu Yang, Zeyi Huang, Xuehai He, Luyao Ma, Baolin Peng, Hao Cheng, Pengcheng He, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Thetaevolve: Test-time learning on open problems. *arXiv preprint arXiv:2511.23473*, 2025b.

Francis Rhys Ward, Teun van der Weij, Hanna Gábor, Sam Martin, Raja Mehta Moreno, Harel Lidar, Louis Makower, Thomas Jodrell, and Lauren Robson. CTRL-ALT-DECEIT: Sabotage evaluations for automated AI R&D. *Advances in Neural Information Processing Systems*, 2025.

Jiaqi Wei, Yuejin Yang, Xiang Zhang, Yuhan Chen, Xiang Zhuang, Zhangyang Gao, Dongzhan Zhou, Guangshuai Wang, Zhiqiang Gao, Juntai Cao, Zijie Qiu, Ming Hu, Chenglong Ma, Shixiang Tang, Junjun He, Chunfeng Song, Xuming He, Qiang Zhang, Chenyu You, Shuangjia Zheng, Ning Ding, Wanli Ouyang, Nanqing Dong, Yu Cheng, Siqi Sun, Lei Bai, and Bowen Zhou. From AI for Science to Agentic Science: A Survey on Autonomous Scientific Discovery, 2025. arXiv:2508.14111.

Amy Xin, Jiening Siow, Junjie Wang, Zijun Yao, Jian Song, Lei Hou, Juanzi Li, and Fanjin Zhang. Eurekagent: Agent environment engineering is all you need for autonomous scientific discovery. *arXiv preprint arXiv:2606.13662*, 2026.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search, 2025. arXiv:2504.08066.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*, 2024.

Xu Yang, Xiao Yang, Shikai Fang, Yifei Zhang, Jian Wang, Bowen Xian, Qizheng Li, Jingyuan Li, Minrui Xu, Yuante Li, Haoran Pan, Yuge Zhang, Weiqing Liu, Yelong Shen, Weizhu Chen, and Jiang Bian. R&D-Agent: An LLM-agent framework towards autonomous data science. *arXiv preprint arXiv:2505.14738*, 2025a.

Zhuo Yang, Daolang Wang, Lingli Ge, Beilun Wang, Tianfan Fu, and Yuqiang Li. Reasoning bo: Enhancing bayesian optimization with long-context reasoning power of llms. *arXiv preprint arXiv:2505.12833*, 2025b.

Mario Zechner. Pi Coding Agent. npm software release, 2026. URL https://www.npmjs.com/package/@earendil-works/pi-coding-agent/v/0.81.1. Version 0.81.1; accessed August 12, 2026.

Ruiyi Zhang, Peijia Qin, Qi Cao, Li Zhang, and Pengtao Xie. Aibuildai: An ai agent for automatically building ai models. *arXiv preprint arXiv:2604.14455*, 2026a.

Tuo Zhang, Alin-Ionut Popa, Yan Xu, Rui Song, and Dimitrios Dimitriadis. Pivot: Bridging planning and execution in llm agents via trajectory refinement. *arXiv preprint arXiv:2605.11225*, 2026b.

Yifei Zhang, Xu Yang, Xiao Yang, Bowen Xian, Qizheng Li, Shikai Fang, Jingyuan Li, Jian Wang, Mingrui Xu, Yuge Zhang, Weiqing Liu, and Jiang Bian. Reasoning as gradient: Scaling mle agents beyond tree search. *arXiv preprint arXiv:2603.01692*, 2026c. URL https://arxiv.org/abs/2603.01692.

Yunxiang Zhang, Kang Zhou, Zhichao Xu, Kiran Ramnath, Yun Zhou, Sangmin Woo, Haibo Ding, and Lin Lee Cheong. Learning to ideate for machine learning engineering agents. *arXiv preprint arXiv:2601.17596*, 2026d.

Xinyu Zhu, Yuzhu Cai, Zexi Liu, Bingyang Zheng, Cheng Wang, Rui Ye, Jiaao Chen, Hanrui Wang, Wei-Chen Wang, Yuzhi Zhang, Linfeng Zhang, Weinan E, Di Jin, and Siheng Chen. Toward ultra-long-horizon agentic science: Cognitive accumulation for machine learning engineering. *arXiv preprint arXiv:2601.10402*, 2026.

A Theory: Why Biased Validation Can Mislead Search

A.1 Main Idea

Consider an archive containing two candidate research anchors. The first, s^*, has the greatest latent scientific utility, but a biased automatic validator assigns a larger expected score to the inferior anchor s_b. The underlying problem therefore satisfies

$$U(s^*) > U(s_b) \qquad \text{but} \qquad \mathbb{E}[V_k(s^*)] < \mathbb{E}[V_k(s_b)].$$

This is a systematic ranking error rather than ordinary evaluation noise. Repeated evaluations estimate the two biased expectations more accurately, so collecting more validation data cannot recover the true ordering. It can instead **make a validation-driven search algorithm increasingly confident in the inferior anchor**.

The two classical search results formalize this effect. At a fixed MCTS branching point, UCB learns that s^* is suboptimal according to the proxy and expands it only $O(\log T)$ times in expectation over T rounds. Its expected fraction of expansions allocated to the truly best anchor therefore converges to zero. In the elitist Genetic Algorithm (GA) setting, once the proxy-optimal but truly inferior state s_b is generated, proxy-based elitism preserves it; under global reachability, this absorption occurs with probability approaching one. Thus both algorithms can continue searching around states favored by the validator even when those states are not favored by the true objective.

ESTRA differs in what information is used to choose the next anchor. Its LLM judge can condition on the research problem, the candidate archive, and the semantic content of each candidate rather than only on validation scores. We model this judge as a contextual decision policy, not as a classifier with a fixed accuracy. The policy may change over time and its decisions may be correlated. For the allocation and node-selection-regret comparison, we require only that its average probability of choosing s^* remains bounded away from zero. To guarantee that repeated LLM judgments return s^* as the final plurality winner, we separately require a positive average voting margin over every competitor.

The comparison is conditional on these assumptions. It explains why biased validation can make proxy-driven search concentrate on suboptimal parents, but it does not claim that every MCTS or GA implementation fails, that every LLM judge satisfies the contextual-policy conditions, or that the judge can discover a strong candidate that is absent from the archive. Theorems A.1 and A.2 make these two conditional comparisons precise.

A.2 Main Theorems

A.2.1 UCB-Based Selection

Theorem A.1 (Fixed-context UCB failure under proxy-order reversal)**.** *Fix a decision context x and an archive $\mathcal{A}$ with*

$$X_t = x, \qquad \mathcal{A}_t = \mathcal{A}, \qquad |\mathcal{A}| = n \geq 2, \qquad s^* := \arg\max_{s \in \mathcal{A}} U(s),$$

where $U(s) = U(x, s) \in [0, 1]$ and the maximizer is unique. The kth proxy evaluation of state s is

$$V_k(s) = U(s) + \varepsilon_k(s) \in [0, 1], \qquad \mu_s := \mathbb{E}[V_k(s)],$$

and, for each state, the sequence $\{V_k(s)\}_{k \geq 1}$ is i.i.d., with the sequences independent across states. Assume the proxy ordering reverses the true ordering, namely

$$\mu^+ := \max_{s \in \mathcal{A}} \mu_s, \qquad \Delta := \mu^+ - \mu_{s^*} > 0.$$

UCB1 evaluates every state once and, thereafter, selects

$$S_t \in \arg\max_{s \in \mathcal{A}} \left\{ \widehat{\mu}_{t-1}(s) + \sqrt{\frac{2 \log t}{N_{t-1}(s)}} \right\},$$

where $N_t(s)$ and $\widehat{\mu}_t(s)$ are respectively the visit count and empirical proxy mean. Given the same archive, the LLM judge selects

$$J_t \mid X_t, \mathcal{H}_{t-1} \sim \pi_t^{\mathrm{LLM}}(\cdot \mid X_t, \mathcal{H}_{t-1}),$$

where $\mathcal{H}_{t-1}$ is the pre-round history. The policy may be time-varying and history-dependent; no fixed accuracy or independence of the judgments is assumed. Define the conditional and marginal true-optimal selection probabilities and their running average by

$$q_t \coloneqq \pi_t^{\mathrm{LLM}}(s^* \mid X_t, \mathcal{H}_{t-1}), \qquad p_t \coloneqq \mathbb{E}[q_t] = \mathbb{P}(J_t = s^*), \qquad \overline{p}_T \coloneqq \frac{1}{T}\sum_{t=1}^{T} p_t,$$

and suppose

$$\liminf_{T\to\infty} \overline{p}_T = \overline{p} > 0.$$

For a sequence of selected states $A_1, \ldots, A_T$, define its normalized node-selection regret by the zero-one loss

$$\mathfrak{R}_T(A) \coloneqq \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\mathbf{1}\{A_t \neq s^*\}].$$

Let $C = 2 + \pi^2/3$. Then, for every $T \geq n$, the expected fraction of UCB expansions allocated to the true optimum obeys

$$\mathbb{E}\left[\frac{N_T(s^*)}{T}\right] \leq B_T \coloneqq \frac{1}{T}\left(\frac{8\log T}{\Delta^2} + C\right) \xrightarrow[T\to\infty]{} 0, \tag{A1}$$

whereas the expected fraction of LLM-selected expansions allocated to s^* is

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{J_t = s^*\}\right] = \overline{p}_T, \qquad \liminf_{T\to\infty} \overline{p}_T = \overline{p} > 0. \tag{A2}$$

Consequently, the LLM policy has a strictly higher expected true-optimal expansion rate for all sufficiently large T. Equivalently, their normalized node-selection regrets satisfy

$$\mathfrak{R}_T(S) = 1 - \mathbb{E}\left[\frac{N_T(s^*)}{T}\right] \geq 1 - B_T \longrightarrow 1, \qquad \limsup_{T\to\infty} \mathfrak{R}_T(J) = 1 - \overline{p} < 1. \tag{A3}$$

Thus UCB has strictly higher node-selection regret for all sufficiently large T under the stated rank reversal.

If UCB returns its most-visited state,

$$\widehat{s}_{\mathrm{UCB}}(T) \in \arg\max_{s\in\mathcal{A}} N_T(s),$$

then

$$\mathbb{P}(\widehat{s}_{\mathrm{UCB}}(T) = s^*) \leq nB_T \xrightarrow[T\to\infty]{} 0. \tag{A4}$$

For the LLM, let

$$Y_t(s) \coloneqq \mathbf{1}\{J_t = s\}, \qquad M_T(s) \coloneqq \sum_{t=1}^{T} Y_t(s), \qquad \widehat{s}_{\mathrm{LLM}}(T) \in \arg\max_{s\in\mathcal{A}} M_T(s).$$

If there is a $\gamma > 0$ such that, for all sufficiently large T,

$$\min_{s\neq s^*} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[Y_t(s^*) - Y_t(s) \mid \mathcal{H}_{t-1}] \geq \gamma \qquad \textit{almost surely},$$

then

$$\mathbb{P}(\widehat{s}_{\mathrm{LLM}}(T) \neq s^*) \leq (n-1)\exp\left(-\frac{T\gamma^2}{2}\right) \xrightarrow[T\to\infty]{} 0. \tag{A5}$$

Thus plurality-based LLM selection eventually has higher final true-selection accuracy than the most-visited UCB recommendation.

A.2.2 Elitist Genetic Search

Consider an elitist genetic algorithm (GA) that uses the proxy reward as its fitness function. Assume that a truly suboptimal state $s_b \neq s^*$ is the unique maximizer of expected proxy fitness:

$$U(s^*) > U(s_b), \qquad \mu_{s_b} > \max_{s \neq s_b} \mu_s, \qquad \Gamma := \mu_{s_b} - \max_{s \neq s_b} \mu_s > 0. \tag{A6}$$

Let G_t be the elite retained after generation t. Each generation produces at least $\lambda \geq 1$ offspring and preserves the candidate with the largest proxy fitness among the current population and its offspring. The variation operator is globally reachable: conditional on any history in which s_b is absent, the probability that s_b is generated in the next generation is at least $\rho > 0$. First suppose that selection uses the stable proxy fitness $F(s) = \mu_s$.

Theorem A.2 (Elitist GA convergence to a proxy-optimal bad state)**.** *Under equation A6, elitist proxy-fitness selection, and global reachability with probability ρ, the GA satisfies*

$$\mathbb{P}(G_T = s_b) \geq 1 - (1-\rho)^T \xrightarrow[T\to\infty]{} 1 \tag{A7}$$

and

$$\mathbb{P}(G_T = s^*) \leq (1-\rho)^T \xrightarrow[T\to\infty]{} 0. \tag{A8}$$

Its normalized elite-selection regret therefore obeys

$$\mathfrak{R}_T(G) := \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}(G_t \neq s^*) \geq 1 - \frac{1}{\rho T} \xrightarrow[T\to\infty]{} 1. \tag{A9}$$

An LLM contextual policy satisfying equation A17 has $\limsup_{T\to\infty} \mathfrak{R}_T(J) = 1 - \overline{p} < 1$ and hence has strictly lower average node-selection regret for all sufficiently large T. If equation A23 also holds, its plurality recommendation converges to s^* while the GA elite converges to s_b.

To compare efficiency, let one proxy evaluation cost $c_V > 0$ and one LLM judgment cost $c_J > 0$. Let $\mathcal{C}_{\mathrm{GA}}(T)$ be the GA's actual proxy-evaluation cost. Since at least λ offspring are evaluated per generation, $\mathcal{C}_{\mathrm{GA}}(T) \geq \lambda T c_V$, and

$$\eta_{\mathrm{GA}}(T) := \frac{\sum_{t=1}^{T} \mathbb{P}(G_t = s^*)}{\mathcal{C}_{\mathrm{GA}}(T)} \leq \frac{1}{\lambda c_V \rho T} \xrightarrow[T\to\infty]{} 0, \tag{A10}$$

whereas T LLM judgments satisfy

$$\eta_{\mathrm{LLM}}(T) := \frac{\sum_{t=1}^{T} \mathbb{P}(J_t = s^*)}{T c_J} = \frac{\overline{p}_T}{c_J}, \qquad \liminf_{T\to\infty} \eta_{\mathrm{LLM}}(T) = \frac{\overline{p}}{c_J} > 0. \tag{A11}$$

Thus the LLM contextual policy is asymptotically more efficient under this definition for any fixed finite positive costs c_V and c_J.

Corollary A.2.1 (Stochastic proxy-fitness evaluations)**.** *Suppose at most K candidates participate in selection at generation t, and every participating candidate is rescored using the mean of m_t fresh proxy-reward samples. If, for every $t \geq 2$,*

$$m_t \geq \frac{8}{\Gamma^2}\left(3\log t + \log(2K)\right), \tag{A12}$$

then

$$G_t \longrightarrow s_b \quad \textit{almost surely}, \qquad \mathbb{P}(G_t = s^*) \longrightarrow 0. \tag{A13}$$

Consequently, under equation A17, the LLM remains asymptotically more efficient in expected true-optimal selections per unit evaluator cost. Under equation A23, it is also asymptotically more accurate as a final selector.

A.3 Implications and Scope

The two theorems identify objective misalignment rather than an intrinsic weakness of MCTS or genetic search. When validation reverses the true ordering, additional proxy evaluations reinforce the wrong preference: UCB assigns a vanishing expansion fraction to the true optimum, and the specified elitist GA is absorbed by a proxy-optimal bad state. A contextual LLM policy with $\overline{p} > 0$

has strictly lower asymptotic zero-one node-selection regret in both comparisons. This formalizes the mechanism behind the claim that validation-driven search can repeatedly refine suboptimal parents when the validation signal is biased.

The assumptions also mark the limits of the conclusion. A positive average success probability supports the allocation and regret comparisons but does not by itself make repeated LLM voting consistent; the latter requires the stronger margin condition. At the symmetric chance level $\overline{p} = 1/n$, the selector still outperforms biased UCB in allocation rate, but its final accuracy remains $1/n$. The UCB theorem treats a fixed set of root children with stationary bounded rewards, and the GA theorem treats a globally reachable elitist algorithm. They do not prove that every MCTS or GA implementation enters a local minimum, nor do they model the ability to generate a semantic "jump" to a state absent from the archive. Action-dependent archive evolution requires additional tree-transition or MDP assumptions. In particular, the semantic advantage of ESTRA is encoded in the contextual policy conditions, not derived from first principles.

A.4 Formal Setting and Auxiliary Results

In automated research and self-evolution, a search procedure repeatedly chooses which research state to develop next. UCB-based MCTS makes this choice from observed validation rewards, whereas a direct LLM judge can also use the semantic content of the candidates. The comparison below asks which procedure more often develops the state with the highest latent scientific value when the validation reward is systematically misaligned.

Contextual decision model. At round t, let X_t denote the information available before selecting a research head. The context may include the current archive $\mathcal{A}_t$, semantic representations of its candidates, the research query, and task metadata. Each candidate $s \in \mathcal{A}_t$ has latent scientific utility $U(X_t, s) \in [0, 1]$, and we assume a unique context-dependent optimum

$$s_t^* \coloneqq \arg\max_{s \in \mathcal{A}_t} U(X_t, s). \tag{A14}$$

An LLM judge is modeled as a stochastic contextual decision policy

$$J_t \mid X_t, \mathcal{H}_{t-1} \sim \pi_t^{\mathrm{LLM}}(\cdot \mid X_t, \mathcal{H}_{t-1}), \tag{A15}$$

where $\mathcal{H}_{t-1}$ is the history before round t. The policy may be fixed, time-varying, or history-dependent; it need not learn online from bandit feedback. Thus "contextual policy" is the precise general term, whereas "contextual-bandit algorithm" applies only when the policy itself is updated from reward feedback.

The policy induces, rather than assumes as a primitive parameter, the conditional and marginal success probabilities

$$q_t \coloneqq \pi_t^{\mathrm{LLM}}(s_t^* \mid X_t, \mathcal{H}_{t-1}), \qquad p_t \coloneqq \mathbb{E}[q_t] = \mathbb{P}(J_t = s_t^*), \tag{A16}$$

and the average contextual success probability

$$\overline{p}_T \coloneqq \frac{1}{T} \sum_{t=1}^{T} p_t, \qquad \liminf_{T \to \infty} \overline{p}_T = \overline{p} > 0. \tag{A17}$$

This condition permits heterogeneous archives, queries, and correlated decisions. A constant-accuracy model is the special case $q_t = p$ almost surely for every t; it is not required by the analysis.

Fixed-context reduction. The main theorem isolates selection within one decision segment by fixing $X_t = x$ and $\mathcal{A}_t = \mathcal{A}$ for all rounds, where $\mathcal{A}$ contains $n \geq 2$ research states. Write $U(s) = U(x, s)$ and

$$s^* \coloneqq \arg\max_{s \in \mathcal{A}} U(s). \tag{A18}$$

This reduction matches the root-child bandit solved by the UCB step of MCTS. We evaluate a selection rule by its true-optimal expansion rate, namely the fraction of decisions allocated to s^*. When a method returns one final state, we instead measure its true-selection accuracy $\mathbb{P}(\widehat{s} = s^*)$.

Biased validation signal. The true utility is unavailable during search. The kth evaluation of state s instead produces a bounded proxy reward

$$V_k(s) = U(s) + \varepsilon_k(s) \in [0,1], \qquad \mu_s \coloneqq \mathbb{E}[V_k(s)]. \tag{A19}$$

For each state, the proxy rewards are independent and identically distributed, and the reward sequences are independent across states. The noise $\varepsilon_k(s)$ may have a state-dependent nonzero mean, allowing systematic evaluation bias. We study the rank-reversal regime in which the true-optimal state is suboptimal in proxy expectation:

$$\mu^+ \coloneqq \max_{s\in\mathcal{A}} \mu_s, \qquad \Delta \coloneqq \mu^+ - \mu_{s^*} > 0. \tag{A20}$$

Selection rules. UCB1 evaluates every state once and then selects the state with the largest empirical proxy mean plus its exploration bonus:

$$S_t \in \arg\max_{s\in\mathcal{A}} \left\{ \widehat{\mu}_{t-1}(s) + \sqrt{\frac{2\log t}{N_{t-1}(s)}} \right\}, \tag{A21}$$

where $N_t(s)$ and $\widehat{\mu}_t(s)$ are the visit count and empirical proxy mean. Given the same candidates, the LLM action J_t is drawn from equation A15. Under the fixed-context reduction, $s_t^* = s^*$, so equation A16–equation A17 measure its average probability of selecting the same true-optimal state as the UCB comparison.

Repeated voting needs an additional condition because correlated errors need not average away. Let

$$Y_t(s) \coloneqq \mathbf{1}\{J_t = s\}, \qquad M_T(s) \coloneqq \sum_{t=1}^{T} Y_t(s), \tag{A22}$$

We use the following optional average conditional-margin condition: for some $\gamma > 0$ and all sufficiently large T,

$$\min_{s\neq s^*} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[Y_t(s^*) - Y_t(s) \mid \mathcal{H}_{t-1}] \geq \gamma \qquad \text{almost surely.} \tag{A23}$$

This condition allows time-varying and dependent judgments, but requires the true optimum to retain a positive average voting margin over every competitor.

Corollary A.2.2 (Chance-level selection can outperform biased UCB)**.** *Under the setting of Theorem A.1, suppose the selector has chance-level average accuracy*

$$\overline{p}_T = \frac{1}{n} \qquad \textit{for every } T. \tag{A24}$$

Then its expected true-optimal expansion rate is exactly $1/n$, and hence is strictly larger than the UCB rate for all sufficiently large T.

If, in addition, the joint distribution of its state counts is invariant under permutations of the n state labels and plurality ties are broken uniformly, then

$$\mathbb{P}(\widehat{s}_{\mathrm{LLM}}(T) = s^*) = \frac{1}{n}, \qquad \mathbb{E}[U(\widehat{s}_{\mathrm{LLM}}(T))] = \frac{1}{n}\sum_{s\in\mathcal{A}} U(s) \tag{A25}$$

for every T. This symmetry condition holds, for example, for independent uniform choices and for the persistent-error model

$$\theta \sim \mathrm{Dirichlet}\left(\frac{\alpha_0}{n}\mathbf{1}_n\right), \qquad J_t \mid \theta \overset{\text{i.i.d.}}{\sim} \mathrm{Categorical}(\theta). \tag{A26}$$

Here $\alpha_0 > 0$ controls persistence; in this model the within-instance correlation of each state-choice indicator is $1/(\alpha_0+1)$. Consequently, the UCB final-selection accuracy converges to zero while the chance-level selector remains at $1/n$. The result shows that UCB driven by a sufficiently misaligned proxy can be worse than random selection; it does not show that chance-level LLM judgments contain useful semantic information.

Corollary A.2.3 (Finite-context policy extension). *Let $\mathcal{X}$ be a finite context set. Each context $x \in \mathcal{X}$ has an archive $\mathcal{A}(x)$ of size $n_x \geq 2$, a unique true optimum*

$$s_x^* \coloneqq \arg\max_{s \in \mathcal{A}(x)} U(x, s), \tag{A27}$$

and proxy-reward streams that are bounded, i.i.d. within each context-action pair, and independent across pairs, with means $\mu_x(s)$. Suppose the proxy ordering reverses the true optimum in every context, with gaps

$$\Delta_x \coloneqq \max_{s \in \mathcal{A}(x)} \mu_x(s) - \mu_x(s_x^*) > 0. \tag{A28}$$

Let $C = 2 + \pi^2/3$. Assume the context sequence is exogenous: it is deterministic or independent of the proxy-reward streams. Consider tabular contextual UCB, which maintains a separate UCB1 instance for every context and uses that instance's local visit count as its clock. Let

$$T_x(T) \coloneqq \sum_{t=1}^{T} \mathbf{1}\{X_t = x\}, \qquad K_T^{\mathrm{UCB}} \coloneqq \sum_{t=1}^{T} \mathbf{1}\{S_t = s_{X_t}^*\}. \tag{A29}$$

Conditionally on the realized context sequence,

$$\mathbb{E}\left[\frac{K_T^{\mathrm{UCB}}}{T}\middle| X_1, \ldots, X_T\right] \leq \frac{1}{T}\sum_{x \in \mathcal{X}}\left[n_x + \frac{8\log(\max\{T_x(T), 2\})}{\Delta_x^2} + C\right] \xrightarrow[T\to\infty]{} 0. \tag{A30}$$

For the LLM contextual policy equation A15, define

$$K_T^{\mathrm{LLM}} \coloneqq \sum_{t=1}^{T} \mathbf{1}\{J_t = s_{X_t}^*\}. \tag{A31}$$

Then

$$\mathbb{E}\left[\frac{K_T^{\mathrm{LLM}}}{T}\right] = \overline{p}_T. \tag{A32}$$

Therefore, if equation A17 holds, the contextual policy has a strictly higher expected true-optimal selection rate for all sufficiently large T.

A.5 Derivation of the UCB-Based Result

We first state the standard finite-time UCB1 pull-count bound in the notation used above Auer et al. (2002).

Lemma A.3 (Proxy-suboptimal pull count). *If a state s has proxy gap $\Delta_s = \mu^+ - \mu_s > 0$, then UCB1 satisfies*

$$\mathbb{E}[N_T(s)] \leq \frac{8\log T}{\Delta_s^2} + C, \qquad C = 2 + \frac{\pi^2}{3}, \tag{A33}$$

for every $T \geq n$.

Proof. Fix a proxy-optimal state $s^+ \in \arg\max_{a \in \mathcal{A}} \mu_a$ and let

$$m = \left\lceil \frac{8\log T}{\Delta_s^2} \right\rceil. \tag{A34}$$

Once s has been evaluated at least m times, selecting it at round t implies that at least one of the following events occurs:

$$\widehat{\mu}_{t-1}(s^+) \leq \mu^+ - \sqrt{\frac{2\log t}{N_{t-1}(s^+)}}, \tag{A35}$$

$$\widehat{\mu}_{t-1}(s) \geq \mu_s + \sqrt{\frac{2\log t}{N_{t-1}(s)}}, \tag{A36}$$

$$\mu^+ < \mu_s + 2\sqrt{\frac{2\log t}{N_{t-1}(s)}}. \tag{A37}$$

Event equation A37 is impossible because $N_{t-1}(s) \geq m$ and $t \leq T$. For any fixed sample count, Hoeffding's inequality bounds the probabilities of equation A35 and equation A36 by t^{-4}. Summing over rounds and possible sample counts gives the standard UCB1 remainder

$$2\sum_{t=1}^{\infty} t^{-2} = \frac{\pi^2}{3}. \tag{A38}$$

Consequently,

$$\mathbb{E}[N_T(s)] \leq m + 1 + \frac{\pi^2}{3} \leq \frac{8\log T}{\Delta_s^2} + 2 + \frac{\pi^2}{3}, \tag{A39}$$

which proves equation A33. □

A.5.1 Proof of Theorem A.1

By equation A20, s^* is proxy-suboptimal with proxy gap Δ. Applying Lemma A.3 to s^* and dividing by T gives

$$\mathbb{E}\left[\frac{N_T(s^*)}{T}\right] \leq \frac{1}{T}\left(\frac{8\log T}{\Delta^2} + C\right) = B_T. \tag{A40}$$

Because $\log T/T \to 0$, we have $B_T \to 0$, proving equation A1. By linearity of expectation,

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{J_t = s^*\}\right] = \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}(J_t = s^*) \tag{A41}$$

$$= \overline{p}_T, \tag{A42}$$

which proves equation A2 without requiring fixed probabilities or independence among the judge outputs.

For zero-one node-selection regret, taking complements in equation A1 and equation A2 gives

$$\mathfrak{R}_T(S) = 1 - \mathbb{E}\left[\frac{N_T(s^*)}{T}\right] \geq 1 - B_T, \tag{A43}$$

$$\mathfrak{R}_T(J) = 1 - \overline{p}_T. \tag{A44}$$

The first expression converges to one, while $\limsup_{T\to\infty}(1 - \overline{p}_T) = 1 - \overline{p} < 1$. This proves equation A3 and the eventual strict regret comparison.

By equation A17, there exists a finite T_J such that $\overline{p}_T \geq \overline{p}/2$ for every $T \geq T_J$. For an explicit finite allocation comparison, use $\log T \leq \sqrt{T}$ for $T \geq 1$:

$$B_T \leq \frac{8}{\Delta^2\sqrt{T}} + \frac{C}{T}. \tag{A45}$$

Therefore, the sufficient threshold

$$T_{\text{sel}} = \left\lceil \max\left\{ n, T_J, \left(\frac{32}{\overline{p}\Delta^2}\right)^2, \frac{4C}{\overline{p}} \right\} \right\rceil \tag{A46}$$

ensures $B_T < \overline{p}/2 \leq \overline{p}_T$ for every $T > T_{\text{sel}}$.

It remains to analyze the final recommendations. If $\widehat{s}_{\text{UCB}}(T) = s^*$, then s^* has the largest of n visit counts whose sum is T, and therefore $N_T(s^*) \geq T/n$. Markov's inequality gives

$$\mathbb{P}(\widehat{s}_{\text{UCB}}(T) = s^*) \leq \mathbb{P}\left(N_T(s^*) \geq \frac{T}{n}\right) \tag{A47}$$

$$\leq \frac{n}{T}\mathbb{E}[N_T(s^*)] \tag{A48}$$

$$\leq \frac{n}{T}\left(\frac{8\log T}{\Delta^2} + C\right) = nB_T. \tag{A49}$$

This proves equation A4.

For the LLM plurality result, fix any $s \neq s^*$ and define

$$D_t(s) \coloneqq Y_t(s^*) - Y_t(s), \qquad Z_t(s) \coloneqq D_t(s) - \mathbb{E}[D_t(s) \mid \mathcal{H}_{t-1}]. \tag{A50}$$

The sequence $\{Z_t(s)\}$ is a martingale-difference sequence, and each increment lies in a conditionally centered interval of length two. Under equation A23,

$$\sum_{t=1}^{T} \mathbb{E}[D_t(s) \mid \mathcal{H}_{t-1}] \geq T\gamma \tag{A51}$$

for every sufficiently large T. If $M_T(s^*) \leq M_T(s)$, then $\sum_{t=1}^{T} D_t(s) \leq 0$, which requires $\sum_{t=1}^{T} Z_t(s) \leq -T\gamma$. The Azuma–Hoeffding inequality gives

$$\mathbb{P}(M_T(s^*) \leq M_T(s)) \leq \exp\left(-\frac{T\gamma^2}{2}\right). \tag{A52}$$

A union bound over the $n-1$ competitors proves equation A5. Together with equation A4, this completes the proof.

A.5.2 Proof of Corollary A.2.2

Substituting equation A24 into equation A2 gives

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{J_t = s^*\}\right] = \frac{1}{n}. \tag{A53}$$

Because $B_T \to 0$, there is a finite T_0 such that $B_T < 1/n$ for all $T \geq T_0$. Equation equation A1 therefore proves the strict allocation comparison.

Now assume label invariance of the count vector and symmetric tie breaking. The plurality rule is equivariant under permutations: permuting the state labels in the counts applies the same permutation to the distribution of its output. Label invariance therefore implies that every state has the same probability of being returned. These n probabilities sum to one, so each equals $1/n$. In particular,

$$\mathbb{P}(\widehat{s}_{\mathrm{LLM}}(T) = s^*) = \frac{1}{n}. \tag{A54}$$

Taking expectation over the returned state also gives

$$\mathbb{E}[U(\widehat{s}_{\mathrm{LLM}}(T))] = \sum_{s\in\mathcal{A}} \frac{1}{n} U(s) = \frac{1}{n}\sum_{s\in\mathcal{A}} U(s). \tag{A55}$$

Both independent uniform choices and the symmetric Dirichlet mixture in equation A26 have the required label invariance. Finally, equation A4 shows that the UCB final-selection accuracy converges to zero, completing the comparison.

This chance-level construction does not satisfy the premise needed for plurality consistency. For independent uniform judgments, every conditional difference in equation A23 is exactly zero. Under the symmetric Dirichlet mixture, persistent profiles that favor a state other than s^* have positive probability, so no deterministic $\gamma > 0$ can satisfy equation A23 almost surely. Accordingly, equation A5 is not invoked, and the final accuracy remains $1/n$ rather than approaching one.

A.5.3 Proof of Corollary A.2.3

For each $x \in \mathcal{X}$, define the context-action count

$$N_{T,x}(s) \coloneqq \sum_{t=1}^{T} \mathbf{1}\{X_t = x,\, S_t = s\}. \tag{A56}$$

Condition on the exogenous context sequence and write $m_x = T_x(T)$. Because tabular contextual UCB maintains separate statistics and a separate local clock for each context, its decisions on the m_x appearances of context x are exactly those of an ordinary UCB1 instance run for m_x rounds.

Exogeneity ensures that conditioning on the context sequence does not alter the bounded i.i.d. proxy-reward streams used by that instance.

If $m_x \geq n_x$, Lemma A.3 applied with proxy gap Δ_x gives

$$\mathbb{E}[N_{T,x}(s_x^*) \mid X_1, \ldots, X_T] \leq \frac{8 \log m_x}{\Delta_x^2} + C. \tag{A57}$$

If $m_x < n_x$, the deterministic bound $N_{T,x}(s_x^*) \leq m_x < n_x$ applies instead. Both cases are covered by the looser uniform inequality

$$\mathbb{E}[N_{T,x}(s_x^*) \mid X_1, \ldots, X_T] \leq n_x + \frac{8 \log(\max\{m_x, 2\})}{\Delta_x^2} + C. \tag{A58}$$

Since

$$K_T^{\mathrm{UCB}} = \sum_{x \in \mathcal{X}} N_{T,x}(s_x^*), \tag{A59}$$

summing equation A58 over contexts and dividing by T proves the finite-time inequality in equation A30. Moreover, $m_x \leq T$ and $\mathcal{X}$ is finite, so its right-hand side is at most

$$\frac{1}{T} \sum_{x \in \mathcal{X}} \left[n_x + \frac{8 \log(\max\{T, 2\})}{\Delta_x^2} + C \right] = O\left(\frac{\log T}{T} \right), \tag{A60}$$

which converges to zero uniformly over all exogenous context sequences. Taking expectation over a random exogenous sequence therefore preserves the same asymptotic conclusion.

For the contextual LLM policy, linearity of expectation and equation A16 give

$$\mathbb{E}\left[\frac{K_T^{\mathrm{LLM}}}{T} \right] = \frac{1}{T} \sum_{t=1}^{T} \mathbb{P}(J_t = s^*_{X_t}) \tag{A61}$$

$$= \frac{1}{T} \sum_{t=1}^{T} p_t = \overline{p}_T, \tag{A62}$$

which proves equation A32. Finally, equation A17 implies $\overline{p}_T \geq \overline{p}/2$ for all sufficiently large T, whereas the uniform UCB upper bound converges to zero. The contextual policy therefore has a strictly higher expected true-optimal selection rate eventually.

A.6 Derivation of the Genetic-Search Result

A.6.1 Stable Proxy Fitness

If s_b belongs to the initial population, unique proxy optimality and elitism make it the elite immediately, so the claimed bounds are trivial. Assume henceforth that s_b is absent initially.

Let A_T be the event that s_b is not generated in generation T, let $H_T = \bigcap_{t=1}^{T} A_t$, and let $\mathcal{F}_T$ denote the history through generation T. On H_{T-1}, global reachability implies

$$\mathbb{P}(A_T \mid \mathcal{F}_{T-1}) \leq 1 - \rho. \tag{A63}$$

Taking expectations and applying this inequality recursively gives

$$\mathbb{P}(H_T) = \mathbb{E}[\mathbf{1}\{H_{T-1}\}\mathbb{P}(A_T \mid \mathcal{F}_{T-1})] \tag{A64}$$

$$\leq (1 - \rho)\mathbb{P}(H_{T-1}) \tag{A65}$$

$$\leq (1 - \rho)^T. \tag{A66}$$

Whenever s_b is generated, equation A6 makes it the unique candidate with maximal proxy fitness. Elitist selection therefore makes it the elite, and no later candidate can replace it. Hence

$$\{G_T \neq s_b\} \subseteq H_T, \tag{A67}$$

which yields

$$\mathbb{P}(G_T = s_b) = 1 - \mathbb{P}(G_T \neq s_b) \geq 1 - \mathbb{P}(H_T) \geq 1 - (1 - \rho)^T. \tag{A68}$$

Because $s_b \neq s^*$,

$$\{G_T = s^*\} \subseteq \{G_T \neq s_b\}, \tag{A69}$$

and thus

$$\mathbb{P}(G_T = s^*) \leq (1-\rho)^T. \tag{A70}$$

This proves equation A7 and equation A8.

For the final-selection comparison, equation A5 implies that the LLM plurality recommendation selects s^* with probability converging to one under equation A23. Equation equation A8 shows that the GA elite selects s^* with probability converging to zero. Hence the LLM has strictly higher final true-selection accuracy for all sufficiently large T.

For the efficiency comparison, sum equation A8 over the first T generations:

$$\sum_{t=1}^{T} \mathbb{P}(G_t = s^*) \leq \sum_{t=1}^{T} (1-\rho)^t \leq \frac{1}{\rho}. \tag{A71}$$

Therefore,

$$\mathfrak{R}_T(G) = 1 - \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}(G_t = s^*) \tag{A72}$$

$$\geq 1 - \frac{1}{\rho T} \xrightarrow[T\to\infty]{} 1, \tag{A73}$$

which proves equation A9. Together with equation A3, this also proves the stated eventual regret comparison with the LLM policy.

Since $\mathcal{C}_{\mathrm{GA}}(T) \geq \lambda T c_V$, even this optimistic lower bound on the GA's actual cost gives

$$\eta_{\mathrm{GA}}(T) \leq \frac{1}{\lambda c_V \rho T}. \tag{A74}$$

By equation A17, the LLM efficiency satisfies

$$\eta_{\mathrm{LLM}}(T) = \frac{\overline{p}_T}{c_J}, \qquad \liminf_{T\to\infty} \eta_{\mathrm{LLM}}(T) = \frac{\overline{p}}{c_J} > 0. \tag{A75}$$

Since $\eta_{\mathrm{GA}}(T) \to 0$, the LLM contextual policy is more efficient for all sufficiently large T. This completes the proof of Theorem A.2.

A.6.2 Stochastic Proxy Fitness

We now prove Corollary A.2.1. Let $\widehat{\mu}_t(s)$ be the mean of the m_t fresh proxy samples used to score candidate s at generation t, and define the ranking-error event

$$E_t := \left\{ \max_{s \text{ scored at } t} |\widehat{\mu}_t(s) - \mu_s| \geq \frac{\Gamma}{4} \right\}. \tag{A76}$$

Conditioning on the history and the identities of the candidates, Hoeffding's inequality and a union bound over at most K scored candidates give

$$\mathbb{P}(E_t) \leq 2K \exp\left(-\frac{m_t \Gamma^2}{8}\right). \tag{A77}$$

Under the batch schedule equation A12,

$$\mathbb{P}(E_t) \leq t^{-3}, \qquad \sum_{t=2}^{\infty} \mathbb{P}(E_t) < \infty. \tag{A78}$$

The first Borel–Cantelli lemma therefore implies that only finitely many events E_t occur almost surely. Thus, with probability one, there is a finite random generation τ after which every scored proxy mean is within $\Gamma/4$ of its expectation.

If s_b participates in any selection after τ, then

$$\widehat{\mu}_t(s_b) > \widehat{\mu}_t(s) \qquad \text{for every } s \neq s_b, \tag{A79}$$

because the true proxy-fitness gap is at least Γ while the two estimation errors sum to at most $\Gamma/2$.

For any deterministic generation M, iterating the global-reachability bound shows that the probability that s_b never participates in selection after M is at most

$$\lim_{r\to\infty} (1-\rho)^r = 0. \tag{A80}$$

With probability one, Borel–Cantelli supplies some integer M after which no ranking error occurs. The event that s_b never participates after this error-free tail is contained in the countable union, over $M \geq 1$, of events that s_b never participates after generation M; every event in this union has probability zero. Hence s_b participates during the error-free tail almost surely. At that generation it becomes the elite, and all later error-free selections retain it. Therefore

$$G_t \longrightarrow s_b \qquad \text{almost surely.} \tag{A81}$$

Since $s_b \neq s^*$, the bounded indicators $\mathbf{1}\{G_t = s^*\}$ converge almost surely to zero. The dominated convergence theorem gives

$$\mathbb{P}(G_t = s^*) = \mathbb{E}[\mathbf{1}\{G_t = s^*\}] \longrightarrow 0. \tag{A82}$$

Its Cesàro mean also converges to zero:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{P}(G_t = s^*) \longrightarrow 0. \tag{A83}$$

The GA performs at least λT proxy evaluations over T generations (and generally many more under equation A12), so its expected true-optimal selections per unit of actual proxy-evaluation cost converge to zero. Equation equation A11 remains bounded below by $\overline{p}/(2c_J)$ for all sufficiently large T, completing the proof of Corollary A.2.1.

B Memory and Context Assembly

This appendix formalizes the four memory operations introduced in Section 2.3.4.

Add: accumulating progress. At each stage-gate event, the worker summarizes the evaluated result, relevant artifacts, and current progress into a compact result card q_v. This card is added to the persistent memory via $m_v = \mathrm{Add}(m_v, q_v)$.

Fold: compressing memory. When persistent memory m_v exceeds its allocated context space B_{mem}, Fold automatically compresses it: $(\widetilde{m}_v, \iota_v) = \mathrm{Fold}(m_v; B_{\mathrm{mem}})$, where $\widetilde{m}_v$ is a compact view that keeps recent, best-validated, and anchor-relevant records in full while summarizing older ones, and ι_v is an index that maps each folded summary to its original records in the archive. The complete memory m_v remains preserved in the archive.

ScienceFlow also applies Fold whenever an ESTRA transition closes a research segment, independently of B_{mem}. If the selected anchor is the current state ($a_n = s_n^{\mathrm{cur}}$), Fold summarizes the research segment just ended. If the anchor is an archived state ($a_n = s_v \in \mathcal{A}_n$) it summarizes the exploration performed since last time that state was restored. This summary is carried forward as historical evidence, while the raw result cards remain addressable via Unfold.

Unfold: retrieving history for anchor selection. Unfold retrieves complete records associated with a requested identifier (z_v a folded summary or an indexed result card): $u_v = \mathrm{Unfold}(m_v, \iota_v, z_v)$. The retrieved records u_v temporarily augment the compact memory view for the current deliberation, while m_v remains the persistent source.

Assemble: constructing the anchor-specific context. The assembly order and update scopes are specified in Table A1; the selected memory view and retrieved records form the anchor-specific component.

C Implementation Details

C.1 Context Construction and Interaction Pipeline

Table A1 specifies the model-facing assembly order and update scopes used by the forward process in Section 2.3.2.

Table A1: Model-facing context components in assembly order. The first four components form the cache-friendly prefix; subsequent components vary with the recoverable state or current research segment.

Component	Contents	Update scope
P_{worker}	Generic research-worker and workspace-interaction policy	Worker lifecycle
P_{runtime}	Runtime behavior, safety, and tool-use policy	Worker lifecycle
P_{tools}	Tool names, descriptions, argument schemas, and call policy	Worker lifecycle
P_{task}	Objective, evaluator contract, budget envelope, task adapter, and initial analysis guidance	Task run
$P_{\text{anchor}}(a_n)$	Workspace view, folded or unfolded memory, validation evidence, and resource status for the selected anchor	Research segment
$P_{\text{dir}}(d_n)$	Extend-or-redirect instruction selected by ESTRA	Research segment
$h_{n,j}$	Reasoning outputs, tool calls, and compact observations	Tool-use round

Initial segment. The initial segment combines the stable context P_{stable} with a compact view of the initial workspace W_0. The task component specifies the objective, evaluator contract, resource budget, and required artifact interface. For MLE-bench, the initial guidance prioritizes file and modality inspection, leakage-safe validation, and a bounded baseline before expensive training. For optimization tasks, it specifies the objective, constraints, evaluator, and candidate-solution contract. These instructions define the initial inspection priorities, while subsequent actions remain selected by the worker's local policy.

Schematic context views for initial and subsequent segments

```
[stable prefix: P_stable]
worker:  generic research-worker policy
runtime: execution and tool-use policy
tools:   read / search / edit / write / execute / inspect
task:    objective + evaluator + budget + artifact contract

[initial segment]
workspace_view: View(W_0)
instruction: inspect task -> analyze data -> build bounded baseline

[subsequent segment]
anchor:         current_state | archived_state
workspace_view: View(W(a_n))
memory_view:    Fold(m(a_n); B_mem) [+ optional Unfold(...)]
validation:     e(a_n)
resources:      l(a_n) + remaining budget B_t
direction:      extend | redirect
closed_branch:  Fold(completed exploration)

[segment-local history: h_n,j]
reasoning -> tool action -> observation -> workspace update -> ...
```

Stable-prefix caching. The stable prefix $P_{\text{stable}} = P_{\text{worker}} \oplus P_{\text{runtime}} \oplus P_{\text{tools}} \oplus P_{\text{task}}$ is placed before all anchor-specific and segment-local content. The worker policy, runtime instructions, and tool schemas remain fixed over the worker lifecycle, while the task contract remains fixed within a task run. Workspace views, memory, validation evidence, resource status, and ESTRA directions are appended afterward and updated at research-segment boundaries. This ordering preserves a common prefix across tool rounds and research segments. Backend-reported cache telemetry is recorded when available, while cache availability does not alter the research-state transitions.

C.2 ESTRA Decision and Re-Anchoring Pipeline

Table A2 and the decision contract below specify the evidence and structured output used by ESTRA in Section 2.3.3.

Table A2: Evidence provided to ESTRA at a research-segment boundary. Optional components are included only when corresponding records are available.

Component	Contents
Current state	Active workspace summary, latest result card, and current validation evidence
Archived anchors	Recoverable states with workspace summaries and associated validation evidence
Memory view	Folded research history, recent result cards, and indexed evidence available through Unfold
Peer evidence	Optional validated scores and compact method summaries from other workers
Resource envelope B_t	Remaining wall-clock budget and current execution constraints

Decision prompt. ESTRA evaluates candidate anchors using their validation evidence, recoverable workspace contents, recent progress, failure history, peer evidence, and the remaining resource budget. The validation metric is treated as one signal rather than the sole decision criterion, allowing ESTRA to preserve promising but not yet leading routes or redirect from saturated ones. Tool use is disabled during this deliberation, and the research worker returns one structured decision specifying the anchor a_n, direction d_n, supporting evidence, and intended search focus.

Schematic ESTRA decision contract

```
decision axes:
  anchor    = current_state | archived_state
  direction = extend | redirect

evidence:
  current_state
  archived_anchor_candidates
  memory_view
  validation_history
  optional_peer_evidence
  remaining_resource_budget

requirements:
  select exactly one anchor and one direction
  identify the current bottleneck
  justify the decision using recorded evidence
  specify the next search focus

structured output:
  anchor, target_state, direction,
  bottleneck, supporting_evidence,
  decision_reason, next_search_focus
```

Folding and segment initialization. The following contract implements the transition-triggered folding defined in Appendix B and records the context used to initialize the next segment.

Schematic ESTRA folding and segment initialization

```
[completed exploration]
terminal_state: Syy
selected_anchor: current_state | archived_state Sxx
direction: extend | redirect
folded_evidence:
  methods attempted
  validated outcomes
  observed failures
  avoid-repeat guidance

[next segment]
stable_prefix: P_stable
workspace: current workspace | Restore(W(Sxx))
anchor_context:
```

```
    memory view
    validation evidence
    resource records
    workspace view
  direction_context: extend | redirect
  optional_context: peer evidence
  local_history: empty
```

Recorded folding and peer-evidence transitions appear in Appendices F.3 and F.4.

C.3 Execution-Control Configuration

Table A3 records the public implementation defaults at commit `c1d2d0e`. The modular YAML files under `scienceflow/config/defaults/lnr/` define memory, monitoring, arbitration, and GPU allocation; `defaults/agent.yaml` defines the model roles. These defaults describe the public implementation rather than a verified configuration shared by all historical runs. The reported task budgets are 24 hours per MLE-bench task-run, 12 hours with two workers for mathematical optimization, and two hours with two workers for SciModelingBench. KTTSP-hard uses a separate ten-day campaign. Run manifests can override the defaults; the task protocols take precedence.

Table A3: Public implementation defaults for memory and execution control. Time settings are in seconds unless otherwise stated.

Setting	Default
Control mode / profile	`resource_smart_llm` / `normal`
Memory-view budget	24,000 characters
Telemetry polling / GPU sampling	10 / 30
Review-state heartbeat	60
Warmup / inactivity / value windows	10 / 3 / 5 heartbeats
Initial trial / hard review	300 / 900
Timebox min. / max. / budget fraction	60 / 1,800 / 0.10
Maximum proof windows	2
GPU queue timeout / heartbeat	1,800 / 15
Admission LLM / arbiter call timeout	60 / 90
Worker advisory min. interval / timeout	600 / 60
Arbiter LLM / advisory call caps per job	12 / 3
Worker / feedback model	`deepseek-v4-flash`
Worker / feedback temperature	0.7 / 0.3
Worker / feedback maximum output tokens	32,768 / 16,384

Telemetry polling and review-state heartbeats update execution evidence; they do not invoke an LLM at every tick. Admission and runtime arbitration use the feedback-model role. Timeboxes adapt to observed progress intervals and contention, using the configured remaining-budget fraction and minimum/maximum bounds. The memory-view budget is measured in characters and is distinct from the model's full context capacity.

C.4 Mechanism Ablation Protocol

We compare the full system with two variants on the 22-task MLE-bench Lite split using three seeds, two research workers per task, and a 24-hour budget. All three configurations use the same backbone, worker count, tools, task instructions, and resource budgets, with both workers sharing the task-level pool of up to two GPUs; only the specified mechanism is disabled.

Without ESTRA. The worker runs a standard agent loop and always continues from the current workspace. Memory, Fold, and checkpointing remain enabled, but ESTRA performs no anchor or research-direction selection: the only transition is continuation, with no re-anchoring to archived states.

Without execution control. The monitoring agents are disabled (`resource_control_mode=off`), leaving jobs to execute and compete freely for shared

resources. Jobs terminate naturally or when their execution timeout is reached; there is no monitoring-agent intervention in admission, allocation, or runtime termination. ESTRA and the memory and checkpoint mechanisms remain enabled.

D MLE-bench Supplementary Results

D.1 Evaluation on the Full MLE-bench Set

We report per-task results on all 75 MLE-bench competitions, grouped by the official Lite, Medium, and High complexity splits. Each **Score** is the mean task-specific benchmark score over three independent runs. Because evaluation metrics differ across competitions, these raw scores should not be compared across tasks.

A checkmark in **Any-Medal** indicates that at least one of the three runs reached a bronze, silver, or gold threshold. For each run, **Cost($)** is the cumulative LLM/API expenditure up to the first medal-producing result; if no medal is obtained, it is the total expenditure over the complete run. The accounting covers end-to-end LLM/API calls across all roles, including research workers, ESTRA decisions, execution control, and auxiliary calls within the stated cutoff. The reported cost is averaged over the three runs and excludes accelerator infrastructure costs. The 75 per-task means sum to $76.4044 per sweep (approximately $229.21 across three sweeps), based on rounded table entries and excluding post-medal expenditure. **First Medal Time (h)** is averaged only over medal-producing runs and is reported as "–" when no run earns a medal. A value of 0 in **GPUs** denotes CPU-only execution. All runs use two research workers with DeepSeek-V4-Flash-Preview, sharing up to two GPUs, 16–32 logical CPU cores, and 256 GB RAM per task-run.

The per-task Any-Medal indicator reports the best observed outcome across runs and is therefore descriptive. It differs from Table 1, which computes the medal rate independently for each run and reports mean ± SEM across three runs. Collapsing the archive by competition yields 54 of 75 tasks (72.0%) with at least one medal: 22 gold, 17 silver, and 15 bronze. This cross-run union differs from the three-run mean plotted in Figure 1.

Figure 1 sources and accounting. Figure 1 includes configurations with reported full-suite performance, corresponding API costs, and identifiable compute resources. GOME points use its 12-hour, single-GPU evaluations and cost table (Zhang et al., 2026c). AIDE, AIRA-dojo, and MARS use the Gemini-2.5-Pro controlled comparison in the MARS cost analysis (Chen et al., 2026b), rather than their other leaderboard configurations. PRAXIST (Li et al., 2026) and Claude Code use the single-sweep results and finalized cost ledgers in the PRAXIST study. Per-task costs are multiplied by 75; directly reported sweep totals are retained. Infrastructure costs are excluded.

Table A4: Sources of the points in Figure 1. Costs are USD per 75-task sweep. G, M, and P denote the GOME, MARS, and PRAXIST studies cited above.

System	Backbone	Hours	GPUs	Medal (%)	Cost
GOME (G)	DeepSeek-V3.2	12	1	28.0	141.75
GOME (G)	GPT-5	12	1	35.1	1,555.50
GOME (G)	o3	12	1	32.5	1,989.75
AIDE (M)	Gemini-2.5-Pro	24	1	23.1	795.00
AIRA-dojo (M)	Gemini-2.5-Pro	24	1	24.4	2,925.00
MARS (M)	Gemini-2.5-Pro	24	1	43.1	4,537.50
PRAXIST (P)	DeepSeek-V4-Pro	24/36	8	80.0	3,054.00
Claude Code (P)	Opus 4.8	24/36	8	73.3	38,370.00
ScienceFlow	DeepSeek-V4-Flash-Preview	24	2	70.22	76.40

GOME and the MARS comparison report three runs. PRAXIST and Claude Code each report a single 75-task sweep, without repeated full-suite evaluations or an estimate of run-to-run variability. They use a shared H100 pool, with up to eight concurrent single-GPU experiments per task; the GPU annotation is a concurrency ceiling. Their task caps are 24 hours for 70 tasks and 36 hours for five tasks. ScienceFlow's plotted 70.22% is the mean over three complete sweeps, matching Table 1,

and is paired with the mean first-medal-cutoff cost over the same three sweeps; unsuccessful task-runs retain full-run costs. The performance uses full-run outcomes, while expenditure is truncated at the first medal for successful task-runs. The figure retains the higher-performance cost-reported configuration when multiple budgets are available for one system. Models, hardware, budgets, repeat counts, and stopping rules vary, so the figure provides reported cost–performance context rather than a controlled Pareto comparison.

Four-hour archival cutoff. Archived first-medal timestamps characterize continued progress beyond a short horizon. By the four-hour cutoff, 35 of 75 tasks (46.67%) had a documented medal-producing result; the final archive contains at least one medal for 54 tasks. The four-hour value is a descriptive best-observed cutoff reconstructed from available task logs, rather than the mean of three independent four-hour evaluations. It should therefore be interpreted as trajectory evidence that additional tasks continue to reach medal quality after four hours, not as a compute-normalized comparison or a causal estimate of any individual ScienceFlow mechanism. The headline three-run 24-hour result is $70.22 \pm 1.18\%$.

Official-leaderboard entries below follow the public MLE-bench leaderboard (OpenAI, 2026b); available system papers are cited alongside the corresponding agent names.

Table A5: Full-set MLE-bench Any-Medal results. Values are percentages reported as mean ± SEM over three runs. Baseline results retain their published reporting protocols. Systems are grouped by source, and the best result in each column is bolded.

Agent	LLM(s) used	Lite (%)	Medium (%)	High (%)	All (%)
ScienceFlow (Ours)[5]	DeepSeek-V4-Flash-Preview	**80.30 ± 1.52**	**74.56 ± 0.88**	44.44 ± 2.22	**70.22 ± 1.18**
Public study					
Iris (Fu et al., 2026)[4]	Claude-Opus-4.6	**80.30 ± 1.50**	64.00 ± 0.90	44.40 ± 2.20	64.90 ± 0.40
MLEvolve (Du et al., 2026)[6]	Gemini-3.1-Pro-Preview	**80.30 ± 1.50**	64.00 ± 0.90	**46.70 ± 0.00**	65.30 ± 0.80
Official leaderboard					
Famou-Agent 2.0 (Li et al., 2025)	Gemini-3-Pro-Preview	**80.30 ± 1.52**	64.04 ± 2.32	42.22 ± 2.22	64.44 ± 1.18
AIBuildAI (Zhang et al., 2026a)	Claude-Opus-4.6	77.27 ± 0.00	61.40 ± 0.88	46.67 ± 0.00	63.11 ± 0.44
CAIR MARS+ (Chen et al., 2026b)	Gemini-3-Pro-Preview	78.79 ± 1.52	60.53 ± 1.52	44.44 ± 2.22	62.67 ± 0.77
MLEvolve[4]	Gemini-3-Pro-Preview	**80.30 ± 1.52**	57.89 ± 1.52	42.22 ± 2.22	61.33 ± 1.33
PiEvolve (Botla et al., 2025)	Gemini-3-Pro-Preview[3]	**80.30 ± 1.52**[2]	58.77 ± 0.88[2]	40.00 ± 0.00[2]	61.33 ± 0.77[2]
Famou-Agent 2.0	Gemini-2.5-Pro	75.76 ± 1.52	57.89 ± 1.52	40.00 ± 0.00	59.56 ± 0.89
ML-Master 2.0 (Zhu et al., 2026)	DeepSeek-V3.2-Speciale	75.76 ± 1.51	50.88 ± 3.51	42.22 ± 2.22	56.44 ± 2.47
CAIR MARS	Gemini-3-Pro-Preview	74.24 ± 1.52	52.63 ± 3.04	37.78 ± 2.22	56.00 ± 1.54
PiEvolve[4]	Gemini-3-Pro-Preview[3]	74.24 ± 3.03[2]	45.61 ± 0.88[2]	35.55 ± 2.22[2]	52.00 ± 0.77[2]
Leeroo (Nadafian et al., 2026)	Gemini-3-Pro-Preview[3]	68.18 ± 2.62[2]	44.74 ± 1.52[2]	40.00 ± 0.00[2]	50.67 ± 1.33[2]
Thesis	gpt-5-codex	65.15 ± 1.52	45.61 ± 7.18	31.11 ± 2.22	48.44 ± 3.64
CAIR MLE-STAR-Pro-1.5 (Nam et al., 2025)	Gemini-2.5-Pro	68.18 ± 2.62	34.21 ± 1.52	33.33 ± 0.00	44.00 ± 1.33
Famou-Agent	Gemini-2.5-Pro	62.12 ± 1.52	36.84 ± 1.52	33.33 ± 0.00	43.56 ± 0.89
Operand ensemble	gpt-5 (low verbosity/effort)[1]	63.64 ± 0.00	33.33 ± 0.88[2]	20.00 ± 0.00[2]	39.56 ± 0.44[2]
CAIR MLE-STAR-Pro-1.0[4]	Gemini-2.5-Pro	66.67 ± 1.52	25.44 ± 0.88	31.11 ± 2.22	38.67 ± 0.77
InternAgent (InternAgent Team, 2025)[4]	DeepSeek-R1	62.12 ± 3.03	26.32 ± 2.63	24.44 ± 2.22	36.44 ± 1.18
R&D-Agent (Yang et al., 2025a)[4]	gpt-5	68.18 ± 2.62	21.05 ± 1.52	22.22 ± 2.22	35.11 ± 0.44
Neo (HeyNeo Team, 2025) multi-agent[4]	undisclosed	48.48 ± 1.52	29.82 ± 2.32	24.44 ± 2.22	34.22 ± 0.89
AIRA-dojo (Toledo et al., 2025)	o3	55.00 ± 1.47	21.97 ± 1.17	21.67 ± 1.07	31.60 ± 0.82
R&D-Agent	o3 + GPT-4.1	51.52 ± 4.01	19.30 ± 3.16	26.67 ± 0.00	30.22 ± 0.89
ML-Master (Liu et al., 2025)[4]	DeepSeek-R1	48.48 ± 1.52	20.18 ± 2.32	24.44 ± 2.22	29.33 ± 0.77
R&D-Agent	o1-preview	48.18 ± 1.11	8.95 ± 1.05	18.67 ± 1.33	22.40 ± 0.50
AIDE (Jiang et al., 2025)	o1-preview	35.91 ± 1.86	8.45 ± 0.43	11.67 ± 1.27	17.12 ± 0.61
AIDE	gpt-4o-2024-08-06	18.55 ± 1.26	3.06 ± 0.33	8.15 ± 0.84	8.63 ± 0.54
AIDE	claude-3-5-sonnet	19.70 ± 1.52	2.63 ± 1.52	2.22 ± 2.22	7.56 ± 1.60
OpenHands (Wang et al., 2025a)	gpt-4o-2024-08-06	12.12 ± 1.52	1.75 ± 0.88	2.22 ± 2.22	4.89 ± 0.44
AIDE	llama-3.1-405b	10.23 ± 1.14	0.66 ± 0.66	0.00 ± 0.00	3.33 ± 0.38
MLAB	gpt-4o-2024-08-06	4.55 ± 0.86	0.00 ± 0.00	0.00 ± 0.00	1.60 ± 0.27

[1] Uses light assistance from Gemini-2.5-Pro, Grok-4, and Claude 4.1 Opus, distilled by Gemini-2.5-Pro.
[2] For incomplete three-run evaluations, missing runs are counted as Any-Medal failures when computing the reported mean and SEM.
[3] Uses Gemini-3-Pro-Preview primarily, with selected modules using GPT-5 and GPT-5-mini.
[4] Uses a reported 12 h or 36 h per-task budget rather than the standard 24 h budget; the exact setting follows the cited source.
[5] ScienceFlow evaluates `tensorflow-speech-recognition-challenge` using the corrected setup from MLE-bench issue #63; baseline aggregates are retained as reported.
[6] MLEvolve with Gemini-3.1-Pro-Preview uses a 12 h per-task budget.

Table A6: Per-task results on the 22-task MLE-bench Lite split. Field definitions follow Section D.1.

Competition Name	Year	Score	Cost($)	GPUs	Any-Medal	First Medal Time (h)
detecting-insults-in-social-commentary	2012	0.9588	0.0618	1	✓	2.54
the-icml-2013-whale-challenge-right-whale-redux	2013	0.9486	1.4995	1	✓	0.12
mlsp-2013-birds	2013	0.9099	0.1483	1	✓	1.95
random-acts-of-pizza	2015	0.7713	1.0641	1	✓	0.35
denoising-dirty-documents	2015	0.0116	0.0141	1	✓	1.08
text-normalization-challenge-russian-language	2017	0.9791	0.3177	1	✓	1.35
dogs-vs-cats-redux-kernels-edition	2017	0.0150	0.0150	1	✓	0.48
spooky-author-identification	2017	0.2800	0.2299	1	✓	3.66
text-normalization-challenge-english-language	2017	0.9965	0.1472	1	✓	1.59
leaf-classification	2017	0.0121	0.0603	0	✓	5.76
jigsaw-toxic-comment-classification-challenge	2018	0.9865	0.1521	1	✓	2.93
new-york-city-taxi-fare-prediction	2018	4.8527	0.2986	0	-	-
nomad2018-predict-transparent-conductors	2018	0.0550	0.1805	0	✓	0.09
dog-breed-identification	2018	0.3525	0.5283	1	-	-
aerial-cactus-identification	2019	1.0000	0.2885	1	✓	2.11
aptos2019-blindness-detection	2019	0.9204	0.3543	1	✓	4.84
histopathologic-cancer-detection	2019	0.9970	0.0214	1	✓	1.07
siim-isic-melanoma-classification	2020	0.9353	1.9233	2	✓	11.98
plant-pathology-2020-fgvc7	2020	0.9928	0.0283	1	✓	0.46
ranzcr-clip-catheter-line-classification	2021	0.9124	0.2221	1	-	-
tabular-playground-series-dec-2021	2021	0.9622	0.0106	0	✓	0.42
tabular-playground-series-may-2022	2022	0.9882	0.5283	0	-	-

Table A7: Per-task results on the 38-task MLE-bench Medium split. Field definitions follow Section D.1.

Competition Name	Year	Score	Cost($)	GPUs	Any-Medal	First Medal Time (h)
AI4Code	2022	0.7911	1.6478	1	-	-
alaska2-image-steganalysis	2020	0.8992	1.6478	2	✓	22.27
billion-word-imputation	2014	4.4668	0.1794	2	✓	7.87
cassava-leaf-disease-classification	2021	0.8995	0.3477	2	✓	11.77
cdiscount-image-classification-challenge	2017	0.7248	0.4307	2	✓	23.04
chaii-hindi-and-tamil-question-answering	2021	0.7458	0.0056	1	✓	6.11
champs-scalar-coupling	2019	0.6758	0.4361	1	-	-
facebook-recruiting-iii-keyword-extraction	2013	0.5037	1.6478	0	-	-
freesound-audio-tagging-2019	2019	0.7346	0.2054	1	✓	2.04
google-quest-challenge	2020	0.3806	0.1189	1	✓	1.05
h-and-m-personalized-fashion-recommendations	2022	0.0248	0.0309	0	✓	10.20
herbarium-2020-fgvc7	2020	0.4146	0.1594	1	✓	2.97
herbarium-2021-fgvc8	2021	0.1656	0.4524	1	✓	4.06
herbarium-2022-fgvc9	2022	0.6994	0.2894	1	✓	6.88
hotel-id-2021-fgvc8	2021	0.1116	0.1905	1	✓	2.57
hubmap-kidney-segmentation	2021	0.9681	0.0891	1	✓	0.45
icecube-neutrinos-in-deep-ice	2023	1.3585	0.2270	1	-	-
imet-2020-fgvc7	2020	0.6544	0.6925	1	✓	22.33
inaturalist-2019-fgvc6	2019	0.2972	0.2171	1	✓	3.49
iwildcam-2020-fgvc7	2020	0.7329	0.1496	1	✓	2.63
jigsaw-unintended-bias-in-toxicity-classification	2019	0.8494	0.3405	1	-	-
kuzushiji-recognition	2019	0.9527	0.0462	1	✓	3.01
learning-agency-lab-automated-essay-scoring-2	2024	0.8355	0.6734	1	✓	1.27
lmsys-chatbot-arena	2024	1.0010	0.4159	1	✓	9.55
multi-modal-gesture-recognition	2013	0.2177	0.1462	1	✓	7.96
osic-pulmonary-fibrosis-progression	2020	-6.7620	1.5486	2	✓	11.87
petfinder-pawpularity-score	2022	16.9877	0.9553	1	✓	12.03
plant-pathology-2021-fgvc8	2021	0.9072	0.1213	1	✓	1.42
seti-breakthrough-listen	2021	0.8002	2.1684	1	✓	6.25
statoil-iceberg-classifier-challenge	2018	0.1349	3.0067	1	✓	23.42
tensorflow-speech-recognition-challenge	2018	0.9841	1.2587	1	✓	1.52
tensorflow2-question-answering	2020	0.5691	0.1996	1	-	-
tgs-salt-identification-challenge	2018	0.7704	1.1963	1	-	-
tweet-sentiment-extraction	2020	0.7178	0.8099	1	✓	11.80
us-patent-phrase-to-phrase-matching	2022	0.8707	0.4126	1	✓	4.01
uw-madison-gi-tract-image-segmentation	2022	0.6902	2.4634	1	-	-
ventilator-pressure-prediction	2021	0.3384	0.5535	1	-	-
whale-categorization-playground	2018	0.4783	0.8077	1	✓	3.70

Table A8: Per-task results on the 15-task MLE-bench High split. Field definitions follow Section D.1.

Competition Name	Year	Score	Cost($)	GPUs	Any-Medal	First Medal Time (h)
3d-object-detection-for-autonomous-vehicles	2019	0.0597	0.1874	1	✓	1.73
bms-molecular-translation	2021	4.8469	1.0800	1	-	-
google-research-identify-contrails-reduce-global-warming	2023	0.4474	4.1043	1	-	-
hms-harmful-brain-activity-classification	2024	0.8100	4.1043	1	-	-
iwildcam-2019-fgvc6	2019	0.2107	0.1442	1	✓	0.68
nfl-player-contact-detection	2023	0.6585	4.1043	1	✓	1.63
predict-volcanic-eruptions-ingv-oe	2021	3,068,390.3330	2.8259	1	✓	4.57
rsna-2022-cervical-spine-fracture-detection	2022	0.5613	4.1043	1	-	-
rsna-breast-cancer-detection	2023	0.1225	4.1043	1	-	-
rsna-miccai-brain-tumor-radiogenomic-classification	2021	0.6163	4.4492	1	✓	0.59
siim-covid19-detection	2021	0.4527	4.1043	1	-	-
smartphone-decimeter-2022	2022	4.6753	4.1043	0	-	-
stanford-covid-vaccine	2020	0.2272	0.0580	1	✓	0.07
vesuvius-challenge-ink-detection	2023	0.1415	4.1043	1	-	-
vinbigdata-chest-xray-abnormalities-detection	2021	0.3344	0.4418	1	✓	1.43

D.2 MLE-bench Long-Horizon Case Study

Figure 3 shows one of the three medal-producing Statoil Iceberg runs. The records below detail its restoration, artifact reuse, and resource-control events.

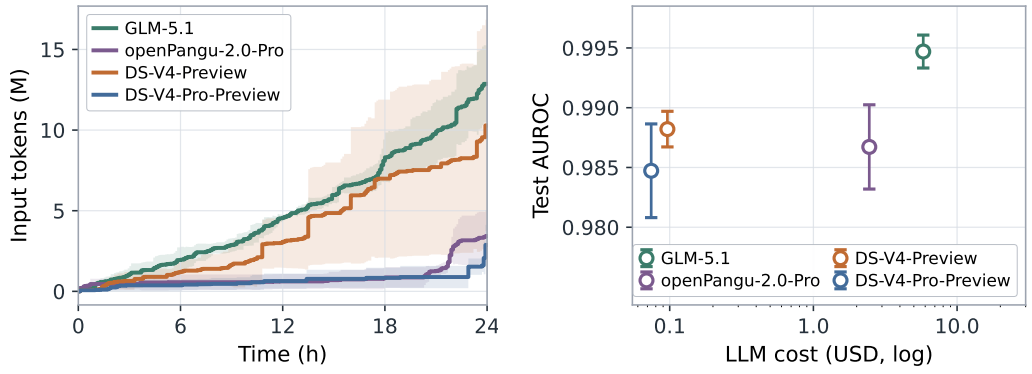
The trajectory exhibits both state restoration and selective artifact reuse. After subsequent CNN-only exploration fails to improve the incumbent, ScienceFlow restores `S59` as an execution anchor and redirects the search toward tree models over saved CNN features. A later unproductive stacking branch similarly returns to `S112` before exploring a new route. Within these branches, the agent selectively reuses earlier workspace artifacts: `S59` combines fold checkpoints from `S02` and `S53`, `S112` trains a LightGBM model on CNN logits and angle features derived from `S53`, and `S123` combines the wide-64 and wide-96 assets from `S53`/`S55`. These operations reduce recomputation while improving validation loss from 0.1100 at `S02` to 0.0597 at `S123`.

Resource control operates concurrently with trajectory search. Two workers share a single-GPU device pool, and the marked events show three controller decisions: preserving exclusive GPU access after a sharing review, terminating a CPU-active job that reports 0% GPU utilization, and falling back from a blocked heavy execution to lightweight inference, which produces a valid submission in 13.91 seconds. Research-route selection remains with the worker, while the controller admits and monitors the corresponding physical execution under the available resources.

State-matched replay. The largest replay-oracle regret in Figure 5 is 11.572×10^{-3} AUROC at `P07`, complementing the summary statistics in Section 3.3.

D.3 Backbone Sensitivity and Efficiency

We compare GLM-5.1, openPangu-2.0-Pro, DeepSeek-V4-Flash-Preview, and DeepSeek-V4-Pro-Preview on *Tabular Playground Series May 2022*, using two workers, a 24-hour budget, and three independent runs per backbone. Figure A1 shows cumulative input tokens and mean best observed test AUROC, evaluated retrospectively. Costs use model-specific inference pricing: GLM and openPangu costs are reconstructed from research-worker call traces, while DeepSeek costs use frozen experiment records. openPangu runs locally without prefix/KV-cache optimization; its cost is estimated from uncached Huawei Cloud Pro pricing at the applicable request-length tier.



(a) Cumulative input tokens.

(b) Test AUROC and LLM cost.

Figure A1: Backbone sensitivity on one MLE-bench task over three runs. Token bands show sample SD; AUROC error bars show population SD. Cost variability is omitted. DS denotes DeepSeek.

GLM-5.1 achieves the highest mean test AUROC (0.9947) at \$5.81, approximately 60× the cost of DeepSeek-V4-Flash-Preview (0.9882, \$0.096). openPangu-2.0-Pro reaches 0.9867 at \$2.45, while DeepSeek-V4-Pro-Preview has the lowest cost (\$0.074) and lowest mean AUROC (0.9847). These results illustrate task-specific quality, throughput, and cost trade-offs: Flash offers a favorable balance among the lower-cost backbones, while GLM attains higher quality at greater expenditure.

The three-run, single-task comparison is descriptive and does not establish a benchmark-wide model ranking.

D.4 Operational Telemetry Coverage

All 75 tasks are included in the performance evaluation under the same ScienceFlow protocol. The operational analysis is retrospective and requires both ESTRA-event records and checkpoint/snapshot telemetry. Both streams are available for 54 tasks; the remaining 21 tasks retain benchmark outcomes but lack complete mechanism-level records. The smaller denominator therefore reflects telemetry availability rather than task selection or a different system configuration.

E Mathematical Optimization Details

E.1 Formal Task Definitions

Circle packing. For $n = 26$, a candidate consists of circle centers $p_i = (x_i, y_i) \in [0,1]^2$ and radii $r_i \geq 0$. The task is

$$\max_{\{p_i, r_i\}_{i=1}^{26}} \sum_{i=1}^{26} r_i \quad \text{subject to} \quad r_i \leq x_i, y_i \leq 1 - r_i, \qquad \|p_i - p_j\|_2 \geq r_i + r_j \quad (i \neq j). \tag{A84}$$

The first constraint keeps every circle inside the unit square, and the second prevents pairwise overlap. The reported score is the sum of radii of a feasible configuration, so larger values are better (Novikov et al., 2025).

Ratio minimization. A candidate is a set of $n = 16$ distinct points $P = \{p_1, \ldots, p_{16}\} \subset \mathbb{R}^2$. Define

$$d_{\min}(P) = \min_{i<j} \|p_i - p_j\|_2, \qquad d_{\max}(P) = \max_{i<j} \|p_i - p_j\|_2. \tag{A85}$$

The objective is

$$\min_{P:\, d_{\min}(P)>0} \rho(P), \qquad \rho(P) = \frac{d_{\max}(P)}{d_{\min}(P)}. \tag{A86}$$

The objective is invariant to translation, rotation, and uniform scaling. AlphaEvolve defines the objective as $\rho(P)$ but follows the source packing tables in reporting $\rho(P)^2$ (Novikov et al., 2025). To match Table 2, we take the positive square root of such published values and report $\rho(P)$ for every method; lower values are better.

Uncertainty inequality. For an integrable real-valued function $f : \mathbb{R} \to \mathbb{R}$, let

$$\widehat{f}(\xi) = \int_{\mathbb{R}} f(x) e^{-2\pi i x \xi}\, \mathrm{d}x, \qquad A(f) = \inf\{r > 0 : f(x) \geq 0 \text{ for all } |x| \geq r\}. \tag{A87}$$

For nonzero even functions satisfying $\max\{f(0), \widehat{f}(0)\} \leq 0$, with $\widehat{f} \in L^1(\mathbb{R})$ and finite $A(f)$ and $A(\widehat{f})$, the Fourier sign-uncertainty constant is (Gonçalves et al., 2017)

$$C_4 = \inf_f A(f) A(\widehat{f}). \tag{A88}$$

Our benchmark follows the first, Hermite-polynomial formulation used by Novikov et al. (2025), which refines the construction of Gonçalves et al. (2017). It searches over

$$Q_{\boldsymbol{c}}(z) = \sum_{k=0}^{m} c_k H_{4k}(z), \qquad f_{\boldsymbol{c}}(x) = Q_{\boldsymbol{c}}(\sqrt{2\pi}\, x) e^{-\pi x^2}, \tag{A89}$$

where H_j is the physicists' Hermite polynomial. The degrees $4k$ make $f_{\boldsymbol{c}}$ invariant under the Fourier transform. As in the official AlphaEvolve verifier, the final coefficient is chosen so that $Q_{\boldsymbol{c}}(0) = 0$, so $f_{\boldsymbol{c}}(0) = \widehat{f}_{\boldsymbol{c}}(0) = 0$, an admissible boundary case (Gonçalves et al., 2017). The polynomial is

oriented to be positive at infinity. After removing the resulting factor z^2, let $r_{\max}$ be the largest positive real root across which $Q_{\boldsymbol{c}}(z)/z^2$ changes sign. The evaluator returns

$$U(\boldsymbol{c}) = \frac{r_{\max}^2}{2\pi}, \qquad C_4 \le U(\boldsymbol{c}). \tag{A90}$$

Thus the reported score is the verified Hermite-construction upper bound on C_4, and lower values are better. This benchmark does not include the separate Laguerre-polynomial refinement also discussed by AlphaEvolve.

E.2 Evaluation and Feasibility Audit

ScienceFlow candidates are re-evaluated from the JSON artifacts stored in immutable workspace snapshots rather than from rounded stage-ledger metrics. The audit parses the saved decimal literals, verifies the task-specific structure and feasibility conditions, and recomputes the displayed objective. Unreadable artifacts, malformed candidates, and candidates that fail the corresponding feasibility test are excluded from comparison.

Circle packing. The strict audit treats every saved coordinate and radius as an exact decimal rational and applies the inequalities in Equation A84 with zero numerical tolerance. It records the minimum boundary slack and the minimum squared pairwise slack. For a candidate with a small negative slack, we also compute the smallest common radius reduction $\delta \ge 0$ that makes its fixed centers feasible; the repaired score is $\sum_i r_i - 26\delta$. This repair is used only to produce a conservative feasible score and never to improve a candidate. The artifact underlying the reported value 2.6359830849 requires $\delta = 9.26 \times 10^{-15}$ and yields the strictly feasible score 2.63598308491745569, which is unchanged at the precision shown in Table 2.

Ratio minimization. The audit parses coordinates with 60-digit decimal precision, requires exactly 16 two-dimensional points, recomputes all 120 pairwise Euclidean distances, and rejects a candidate when $d_{\min} = 0$. It then evaluates $d_{\max}/d_{\min}$ directly, avoiding the six-decimal inverse-squared metric stored in the stage ledger. The best audited DeepSeek-V4-Flash-Preview candidate gives 3.590157365310609, while the best openPangu-2.0-Pro candidate gives 3.590157365325370478. Both values are reported as the unsquared ratio $d_{\max}/d_{\min}$.

Uncertainty inequality. We reconstruct the public AlphaEvolve B.4 Hermite verifier exactly. The audit enforces $Q_{\boldsymbol{c}}(0) = 0$ and a positive limit at infinity, divides out the factor z^2, computes the real roots symbolically, and selects the largest positive sign-changing root using 200-digit root approximations. The resulting bound is evaluated by Equation A90. The reported ScienceFlow bounds 0.348200107555 and 0.343293122432 both pass this verifier.

Baseline provenance. Published entries in Table 2 are transcribed from their cited papers or released repositories and retain the construction and precision reported by those sources; they are not presented as independent reruns. The AlphaEvolve uncertainty entry is independently reproduced with its public Hermite verifier. OpenEvolve is run locally as described in Section 3.1; its retained scores come from the local task evaluator. In particular, its circle-packing output does not preserve the final evaluated coordinates for a separate exact-decimal audit. The local paper archive contains the derived audit summaries used here; some corresponding raw run artifacts remain in the original experiment storage and are not duplicated in the paper repository.

F SpOC4-KTTSP Supplementary Details

F.1 Competition, Tracks, and Evaluation Protocol

SpOC4 is the fourth Space Optimisation Competition organized by ESA's Advanced Concepts Team. Its Keplerian Tomato Traveling Salesperson Problem asks a spacecraft to collect all targets in lunar orbit in minimum mission time while respecting orbital-transfer and maneuverability constraints (European Space Agency, Advanced Concepts Team, 2026). We evaluate the three official instances. The easy, medium, and hard tracks contain 50, 182, and 1,052 targets, respectively; their minimum

transfer times are 0.001, 0.01, and $1/1440$ days, and their maximum mission durations are 200, 500, and 3,000 days.

For an N-target instance, a submission contains a permutation of all targets, $N-1$ departure epochs, and $N-1$ transfer durations. The evaluator checks the permutation, time bounds, and chronological consistency before solving each transfer with a Lambert solver in both directions and with at most 20 revolutions. A nominal leg may use at most $100\,\mathrm{m\,s^{-1}}$ of ΔV; at most five exception legs may exceed this threshold, and no leg may exceed $600\,\mathrm{m\,s^{-1}}$. The score is the arrival time at the final target in days, so lower is better. Candidates are serialized as an official-format JSON decision vector, and invalid candidates receive no score.

Trajectory evaluation is CPU-only. The easy and medium results were obtained through staged continuation runs, whereas KTTSP-hard used the separate ten-day campaign described in Section 3.2, with DeepSeek-V4-Pro-Preview, DeepSeek-V4-Flash-Preview, and GLM-5.1. Because resumed stages and worker counts differed across tracks, we report the public leaderboard outcomes without treating them as a compute-normalized comparison.

Long-horizon memory folding Figure A2 traces worker W01 during the ten-day KTTSP-hard campaign. Across 188 persisted snapshots, the result-card ledger in `.run_results.md`, which serializes the persistent memory m_v, grows from 0.65k to 87.9k characters. In contrast, the 114 recorded context packets assembled at research-segment boundaries remain between 4.4k and 13.4k characters, with a median size of 8.8k and a median paired packet-to-ledger ratio of 13.7%. This separation reflects the ADD, FOLD, and ASSEMBLE operations in Section 2.3.4: accumulated stage records remain persistent, while each new research segment receives a bounded, anchor-specific memory view. The reported packet sizes characterize this dynamic memory component rather than the complete context P_{n+1} of the next research segment.

The trace contains 185 valid evaluation stages and 26 incumbent updates. W01 reduces the objective from 1928.39 mission days at `S01` to 393.229 at `S112`, a 79.6% reduction. Improvement continues late in the campaign: after the incumbent remains at 1641.42 days on June 25, phase-aware route construction coincides with the `S80` improvement to 573.03 days, and subsequent route and timing refinement reaches 420.27 at `S97`, 415.56 at `S109`, and 393.229 at `S112`. At this final incumbent, the most recent paired context packet contains 6.37k characters, compared with an 80.17k-character result-card ledger (7.9%). Fourteen subsequent valid trials do not improve the incumbent, indicating late-stage saturation rather than termination at the first strong solution. This single-run trace does not establish that memory folding causes the score improvements; it instead shows that late-stage improvement and continued validation can coexist with a persistent result-card history and bounded anchor-specific context packets.

Peer-guided search The same trace illustrates cross-worker coordination at research-segment boundaries. At W01's `S25` research-segment boundary, W00 had reached 521.9054 mission days while W01 remained at 1746.8935; W00's validated score prompted W01 to redirect from timing-only refinement toward route ordering. After W01 improved to 1321.8824 days, the `S79` research-segment boundary supplied a compact method summary of W00's search, prompting an operator-level redirect toward route swaps and continuous retiming. W01 subsequently reached a validated score of 420.2742 within 12 hours and 17 minutes of the `S79` decision and later improved to 393.229. The workers exchanged no executable artifacts or workspace state; all subsequent candidates were generated and validated within W01's isolated workspace. The trace therefore illustrates how peer evidence can guide route selection without direct solution reuse, rather than providing a controlled estimate of the coordination effect. Appendix F.4 provides the underlying records and checkpoint timeline.

Route and timing refinement A KTTSP candidate contains two coupled decision components: a discrete visit permutation $\pi = (v_1, \ldots, v_N)$ and a continuous schedule $\tau = \{(t_i, \mathrm{tof}_i)\}_{i=1}^{N-1}$ of departure epochs and flight durations. Figure A3 compares two validated candidates generated during W01's search: an inclination-binned phase ordering and a phase-proximal construction followed by multiresolution retiming. The procedures below summarize their decision structures while omitting implementation-specific optimization loops. These strategies are task-specific artifacts produced by the research worker rather than fixed components of ScienceFlow.

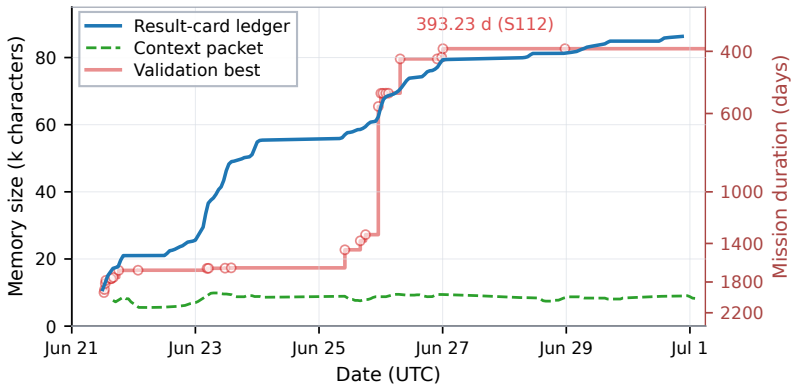
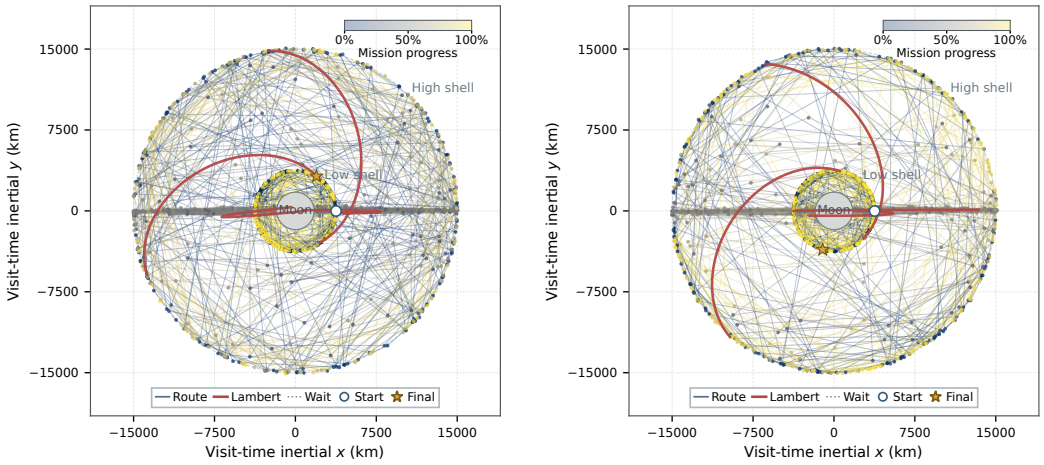


Figure A2: Persistent-memory growth and validation progress for worker W01 during the ten-day KTTSP-hard campaign. The right axis uses $\log_{10}(\text{score}/2000)$ coordinates with raw mission-day tick labels; lower is better. Invalid evaluation outputs are excluded.



(a) Initial strategy at `S78`: inclination-binned phase ordering (1321.8824 days).

(b) Refined strategy at `S112`: phase-proximal construction and retiming (393.229 days).

Figure A3: Route and timing refinement within the W01 KTTSP-hard trajectory. Figures (a) and (b) visualize two evaluator-valid candidates that visit the same target set. Target color progresses from blue to yellow with scheduled visit order, and thin links connect consecutive targets.

Initial strategy: Inclination-binned phase ordering.
`BIN-INCLINATION` → `SORT-PHASE` → `CONCATENATE` → `FIXED-ROUTE RETIME`
The worker groups targets by inclination, orders each group by orbital phase, concatenates the groups, and then retimes the fixed route. This construction provides a feasible starting route but can retain unfavorable adjacencies at group boundaries.

Refined strategy: Phase-proximal construction and retiming.
`PHASE-NEIGHBORS` → `LAMBERT-FILTER` → `APPEND-ROUTE` → `MULTIRESOLUTION-RETIME` → `FORWARD-PASS`
At each construction step, the worker considers phase-proximal successors, filters infeasible candidates with the Lambert evaluator, and appends a transfer-aware feasible choice. Coarse-to-fine retiming and a forward feasibility pass then update the schedule under the temporal and transfer constraints.

Between the evaluator-valid `S78` and `S112` candidates, fewer than 7% of directed consecutive target pairs are shared, while mission duration decreases from 1321.8824 to 393.229 days, a 70.25%

reduction. This descriptive comparison associates the improvement with joint route reordering and retiming rather than isolating individual search operators.

F.2 Context-Folding Trace on KTTSP-hard

This section instantiates the memory operations in Section 2.3.4 with a recorded W01 checkpoint from the KTTSP-hard run. At `S115`, the append-only stage ledger m_v contained 115 result cards and occupied 73,964 characters. When the memory view exceeded B_{mem}, Fold compressed cards `S01--S112` into the addressable summary `L_S01_S112`. The resulting view $\widetilde{m}_v$ occupied 4,661 characters, a 93.7% reduction relative to the raw ledger. At the same research-segment boundary, the runtime emitted a 6,773-character memory-bearing state packet for the next research segment. This telemetry covers the folded view and associated state metadata, rather than the complete context P_{n+1}; the stable prefix, resource provenance, workspace view, and direction instruction are composed during final context assembly.

Before context folding: persistent ledger. The memory m_v keeps each stage as an independently addressable result card in `.run_results.md`. The excerpts below reproduce selected fields from three decision-critical cards; line wrapping is added only for page layout. The `validity` field denotes ScienceFlow's internal evidence tier, not the KTTSP track difficulty or the official feasibility decision. The initial feasible result, an unverified intermediate attempt, and the best validated result are all retained rather than keeping improvements alone.

Selected fields from stage cards before context folding

```
S01  metric=1928.39350319; validity=medium
BRIEF: Bridge strategy achieves 0 infeasible transfers with
       4 exceptions and 1928.39d duration.
WHY:  First fully feasible candidate.

S18  metric=1761.93330934; validity=low
NOTE:  invalid_submission: no submission artifact found.
BRIEF: Finer local-window timing refinement reached 1761.93d.
WHY:  Timing resolution helped, but the result was not eligible
      as a verified comparable entry.

S112 metric=393.229; validity=high
BRIEF: Forward-pass timing refinement improved 696 legs and
       saved 7.70d locally, but yielded only 0.001d net gain.
WHY:  Five frozen exception legs absorbed the upstream savings;
      further progress requires re-timing or re-routing them.
FILES: code=refine_timing_fast.py
```

After context folding: budgeted view. In $\widetilde{m}_v$, the historical portion is replaced by a compact summary with an addressable identifier, while cards needed for verification and immediate continuation remain raw. The generated summary records the folded range, its best stage, representative early evidence, the number of omitted intermediate cards, and the last folded stage:

Folded stage-memory view

```
Historical summary: L_S01_S112  [S01--S112]
best_stage=S112
- S01:  metric=1928.39350319; validity=medium; first feasible
- S02:  metric=1928.39350319; validity=medium; safe baseline
- S03:  metric=1928.39350319; validity=high; validated baseline
- S04:  metric=1891.24195011; validity=medium; timing re-opt
... 107 intermediate stages summarized ...
- S112: metric=393.229; validity=high;
        frozen exception legs block the timing cascade

key_stage_index: S112=best_valid
available_expand_id: L_S01_S112
verification_raw_cards: S01, S18, S112
current_raw_segment: S113, S114, S115
```

Active review through unfolding. The runtime field `available_expand_id` exposes the identifier used by the formal Unfold operation. The following view illustrates an active comparison of

the unverified timing attempt and the best validated route. This is an observed UNFOLD event from the archived W01 runtime trace, shown using the corresponding indexed cards:

On-demand unfolded memory view

```
request: Unfold(L_S01_S112, stages={S18, S112})
index:   L_S01_S112 -> .run_results.md [S01--S112]

returned_raw_cards:
- S18:  metric=1761.93330934; validity=low
  NOTE:  invalid_submission: no submission artifact found
  BRIEF: Finer local-window timing refinement reached 1761.93d.
  WHY:  Timing resolution helped, but the result was not eligible
        as a verified comparable entry.
- S112: metric=393.229; validity=high
  BRIEF: Forward-pass timing refinement improved 696 legs and
         saved 7.70d locally, but yielded only 0.001d net gain.
  WHY:  Five frozen exception legs absorbed the upstream savings;
        further progress requires re-timing or re-routing them.
  FILES: code=refine_timing_fast.py

context_effect: temporary augmentation of folded view
ledger_effect:  none (m_v remains append-only)
```

The index resolves the request to the original cards in `.run_results.md`. In the implementation, this runtime-executed UNFOLD is realized through indexed workspace reads rather than a separate memory service.

F.3 ESTRA-TRIGGERED FOLDING TRACE ON KTTSP-HARD

This section instantiates the ESTRA-triggered fold in Sections 2.3.3 and 2.3.4 with a recorded W01 transition. At 17:17 UTC on June 21, at a research-segment boundary caused by the context limit, the worker selected the runtime action `switch_stage`, moving from terminal stage `S27` to the archived anchor at `S23`. The transition folded the post-anchor branch `S24--S27` into an 845-character summary and assembled a 3,583-character state packet for the restored trajectory. Unlike capacity-based folding of the persistent memory view, this fold scope was determined by the selected archived anchor: it closes the abandoned branch and carries its diagnosis into the next research segment.

Before ESTRA-triggered folding: abandoned tail. The four cards refer to the same 1746.8935-day artifact, and no new route is produced. Their internal evidence tiers vary because the repeated evaluations provide different amounts of verification evidence; this variation does not indicate different outcomes under the official KTTSP feasibility checks:

Selected fields from abandoned-tail cards

```
S24 metric=1746.89350787; validity=medium
BRIEF: Existing artifact re-scored end-to-end with pykep;
       1046 feasible, 4 exceptions, 0 infeasible.

S25 metric=1746.89350787; validity=high
BRIEF: Direct pykep recomputation validates the artifact before
       route-ordering experiments.

S26 metric=1746.89350787; validity=medium
BRIEF: Baseline survives the transition but remains far behind peer W00.
WHY:  Timing-only refinement is exhausted on the sorted route.

S27 metric=1746.89350787; validity=medium
WHY:  Route ordering is the dominant bottleneck; the next route
      should use proxy-prefiltered construction and bounded checks.
```

After ESTRA-triggered folding: restored-state evidence. The generated summary preserves the branch-level conclusion rather than carrying all four cards verbatim into the next segment's agent-facing state packet:

ESTRA-folded abandoned-tail view

```
runtime_action: switch_stage
restore: terminal S27 -> target S23
fold_scope: abandoned tail S24--S27

tail_summary:
- Repeated pykep rescoring only revalidated the existing 1746.89d
  artifact; no new candidate was produced.
- Validity fluctuated between medium and high on the same artifact;
  treat the repeated rescoring as non-informative.
- Timing-only refinement on the sorted route is saturated.
- The 1746.89d baseline remains far behind W00 at 521.91d;
  route-ordering or structural changes are required.
- Avoid redundant validation passes and further timing polish
  without a new ordering or construction strategy.

summary_chars: 845
restored_state_packet_chars: 3583
preserved: raw cards, snapshot, logs, terminal archive
```

After restoration, the worker resumes from the executable workspace snapshot at `S23`, while the folded `S24--S27` branch is retained as completed negative evidence. The original cards remain addressable in the archived memory through UNFOLD. Thus, the ESTRA-triggered fold changes the active research route and agent-facing context without altering the persistent research record.

F.4 PEER-GUIDED SEARCH TRACE ON KTTSP-HARD

The W01 trajectory in Figure A2 contains a concrete cross-worker coordination episode. W00 established an early validated solution of 521.9054 mission days at `S60`, while W01 remained at 1746.8935 days. Each worker maintained an isolated executable workspace, and no artifact or workspace state was transferred between them. Instead, compact peer evidence comprising W00's validated score and method summary entered W01's ESTRA context. The worker used this evidence to redirect its search from timing-only refinement toward route-ordering strategies.

The archived runtime records preserve both the peer-evidence payload and the resulting direction instruction. We identify each archived ESTRA decision by its worker and stage as `ESTRA-Wxx-Syy`.

At W01's `S25` research-segment boundary, compact peer evidence comprising W00's validated score and a brief windowed-2-opt summary entered the worker's ESTRA decision (event `ESTRA-W01-S25`). The worker used the performance gap to diagnose route ordering as the bottleneck, but redirected toward its own proxy-prefiltered nearest-neighbor construction rather than copying W00's procedure. PyKEP supplies the Lambert-transfer evaluation.

Selected fields from ESTRA-W01-S25

```
Route order is the bottleneck not timing; need faster NN using
orbital-element proxy pre-filter then pykep verify only top-3,
not full grid on 30 candidates.

Bottleneck: Route ordering is the dominant factor: 1746d vs peer
521d, but greedy NN with pykep evaluation is too slow to complete
within resource limits.

Exploration summary: Timing refinement on original sorted route
converged at 1746.89d; greedy NN route rewrite attempted but too
slow at 47s/10 legs. Peer at 521.91d via windowed 2-opt.

Redirect: Proxy-pre-filtered NN: rank by orbital distance,
pykep-verify only top 3-5 with coarse grid.
```

After several route-ordering attempts, W01 reached 1321.8824 days but still trailed W00 by approximately 802 days. At `S79`, the peer summary provided more explicit method-level evidence, producing a second redirect at the operator level (event `ESTRA-W01-S79`).

Selected fields from ESTRA-W01-S79

```
Worker decision: Adapt the peer-validated 2-opt+SLSQP operators;
inclination-bin sorting appears saturated.
```

```
Bottleneck: Static inclination-bin sorting cannot match W00's
route-reordering and timing-optimization strategy (519 days).

Evidence: An approximately 802-day gap remains; finer bins improve
the objective by only 54 days, while greedy variants remain infeasible.

Next action: Apply 2-opt route swaps and SLSQP timing refinement
to W01's 1321.8824-day incumbent.
```

Table A9: Validated checkpoints and ESTRA decisions in the KTTSP-hard peer-guided overtaking case. Mission duration is measured in days; lower is better.

Time (UTC)	Worker	Stage	Duration	Search event
Jun 21 16:12	W00	S60	521.9054	Early peer best
Jun 21 16:20	W01	S25	1746.8935	Structural ESTRA redirect
Jun 25 18:04	W01	S78	1321.8824	Fine inclination bins
Jun 25 19:12	W01	S79	1321.8824	Operator-level ESTRA redirect
Jun 25 23:01	W01	S80	573.0305	Phase-greedy narrow bins
Jun 26 00:00	W01	S85	525.9948	Near peer best
Jun 26 07:29	W01	S97	**420.2742**	First validated lead over W00
Jun 27 00:06	W01	S110	393.2352	Ultrafine timing grid
Jun 28 23:29	W01	S112	393.229	Final refinement

Between the first peer-aware ESTRA decision at `S25` and its first validated lead over W00 at `S97`, W01 reduced its incumbent from 1746.8935 to 420.2742 days, a 75.9% improvement over 4 days and 15 hours. More directly, W01 produced the 420.2742-day result within 12 hours and 17 minutes of the operator-level redirect at `S79`. It ultimately reached 393.229 days, outperforming W00's final validated 519.6255-day result by 24.32%.

The trace supports peer-guided adaptation rather than direct solution reuse. Applying windowed SLSQP to W01's existing route at `S94` improved the objective by only 0.02 days. W01 established the subsequent lead after combining a 10-degree inclination-bin route with phase-proximal construction and finer timing grids. Peer evidence therefore served as search guidance: it revealed the structural limitations of the current route family and identified promising operators, while W01 independently instantiated and validated a distinct route and schedule.

G SciModelingBench Results and Evaluation Details

This appendix records the data sources and hidden evaluation boundary for the 12 SciModelingBench tasks in Section 3.2. We use the public SciModelingBench 0.10.0 release and its released dataset. In Table A12, N is the number of candidates in each submission, K is the summary size used by the shared diagnostics, and Q is the total query limit for one system–task pair. ScienceFlow's two workers split this limit. The external baselines are OpenCode 1.18.4 (OpenCode Contributors, 2026), Pi 0.81.1 (Zechner, 2026), Codex 0.144.5 (OpenAI, 2026a), and Claude Code 2.1.72 (Anthropic, 2026). All four use DeepSeek-V4-Flash-Preview through non-interactive clients.

Pass1 protocol and aggregation. Each system–task pair is run once (pass1), with the same two-hour limit, DeepSeek-V4-Flash-Preview backbone, and task-level quota of 8–20 batch submissions. We report the best valid submission within these limits; ScienceFlow's two workers share the quota. Each run uses eight logical CPU cores, no accelerator, and less than 32 GiB peak RSS. These are single-run comparisons, without estimates of run-to-run variability.

Primary metrics. For an ordered submission $B = (b_1, \ldots, b_N)$ and its task-defined reference domain P, let $u(b)$ be the trusted objective oriented so that larger is better. BKM is the mean objective of the best K candidates in B, independent of their submitted order. Normalized enrichment is

$$\mathrm{NE}(B) = \frac{\bar{u}_B - \bar{u}_P}{\bar{u}_{\mathrm{Top}_N(P)} - \bar{u}_P}.$$

For global NDCG, define $g(b) = (u(b) - u_{\min,P})/(u_{\max,P} - u_{\min,P})$ and let $p_1^*, \ldots, p_N^*$ be the ideal top-N candidates from P, sorted by utility. Then

$$\mathrm{NDCG}(B) = \frac{\sum_{i=1}^{N} g(b_i)/\log_2(i+1)}{\sum_{i=1}^{N} g(p_i^*)/\log_2(i+1)}.$$

The denominator uses the full declared reference domain rather than only the submitted candidates. For measured-pool tasks, "global" refers to the frozen evaluation pool, not the entire scientific design space. These definitions follow the v0.10.0 metric contract; reference scopes and N, K, Q appear in Table A12.

For task t, let $s_{a,t}$ be agent a's reported score, r_t the mean of 5,000 uniformly sampled task-sized batches (randomly ordered for ranking tasks), and o_t the attainable oracle optimum. The normalized score is $z_{a,t} = 100(s_{a,t} - r_t)/(o_t - r_t)$. The six standalone tasks each form one group, and the six DrugMatrix endpoints form a seventh group. For these seven groups $\mathcal{G}$, we report

$$\mathrm{Score}_a = \frac{1}{7}\sum_{g\in\mathcal{G}}\frac{1}{|g|}\sum_{t\in g} z_{a,t}, \qquad \mathrm{Rank}_a = \frac{1}{7}\sum_{g\in\mathcal{G}}\frac{1}{|g|}\sum_{t\in g} \rho_{a,t},$$

where $\rho_{a,t}$ ranks the five evaluated agents by task score, with lower rank better. Thus DrugMatrix has the same total weight as each standalone task, rather than six times its weight.

Complete pass1 results. Table A10 reports all 12 pass1 tasks. All five systems exceed Random on every task. ScienceFlow leads on TFBind8, UTR MRL, GFP, MCH, and Phosphorus; its MCH margin over the runner-up is 0.0551, whereas TFBind8 is nearly tied at 0.0005.

Table A10: Complete pass1 task-level results under a two-hour budget with DeepSeek-V4-Flash-Preview. BKM, NE, and NDCG denote best-K mean, normalized enrichment, and global NDCG. Best and second-best agent scores are **bolded** and underlined.

Task	Metric	Random	OpenCode	Pi	Codex	Claude Code	ScienceFlow (Ours)
Standalone scientific design tasks							
TFBind8	BKM	0.7533	0.9764	0.9730	0.9718	0.9648	**0.9769**
Superconductor	NDCG	0.2800	0.4929	**0.8671**	0.8188	0.8132	0.8583
UTR MRL	NE	-0.0006	0.6523	0.7533	0.7424	0.7367	**0.7605**
TFBind10 Pho4	NE	0.0000	0.2467	0.2473	0.2024	**0.3197**	0.2543
GFP	NE	0.0001	0.2600	0.2692	0.2586	0.1977	**0.2732**
Hopper Controller	NDCG	0.1698	**0.3917**	0.3248	0.3755	0.3718	0.3756
Rat clinical pathology tasks (DrugMatrix)							
MCHC	NDCG	0.1733	**0.7344**	0.3720	0.6178	0.6503	0.7220
MCH	NDCG	0.2669	0.4147	0.3673	0.4242	0.3623	**0.4793**
Creatinine	NDCG	0.1955	0.6081	0.7638	0.5681	**0.8038**	0.6995
Sodium	NDCG	0.1519	0.7619	0.8283	0.7696	**0.8498**	0.8488
Chloride	NDCG	0.1689	0.5106	0.4551	**0.5589**	0.5397	0.5488
Phosphorus	NDCG	0.2040	0.4130	0.4917	0.3749	0.3999	**0.5985**

Table A11 separates the standalone-design and DrugMatrix scores underlying the main-text aggregate. ScienceFlow leads in both categories, while each external agent leads on at least one individual task.

Table A11: Complete pass1 group-balanced results. Scores map Random to 0 and the attainable optimum to 100. The DrugMatrix endpoints form one group, weighted equally with each standalone task. Average rank uses the same seven groups. Best and second-best values are **bolded** and underlined; lower rank is better.

Agent	Standalone Design (6)	Rat Clinical Pathology (6)	Group-balanced Score ↑	7-group Avg. Rank ↓	Task Wins
Pi	52.77	43.12	51.39	2.79	1/12
Claude Code	51.65	49.64	51.37	3.69	3/12
Codex	51.49	43.79	50.39	3.64	1/12
OpenCode	43.84	46.44	44.21	3.33	2/12
ScienceFlow (Ours)	**54.16**	**55.91**	**54.41**	**1.55**	**5/12**

Table A12: SciModelingBench task metadata. Visible scale describes the data given to the agent; evaluation scale describes the exact reference domain or label-hidden pool. Values in parentheses are canonical candidates represented by row-level observations. DrugMatrix contains six endpoint tasks from one study collection. BKM, NE, and NDCG denote *best-K mean*, *normalized enrichment*, and *global NDCG*. Data terms come from the original source, while the Hopper row separately identifies simulator software licenses.

Task group	Visible	Evaluation	$N/K/Q$	Trusted evaluator	Source / terms
TFBind8 *BBO*	32,768 sequences	65,536 sequences	32/5/20	Normalized PBM E-score lookup; BKM	PBM/UniPROBE (Barrera et al., 2016; Hume et al., 2015); source-specific academic-use terms
TFBind10 Pho4 *BBO*	2,087,323 rows (524,300 seq.)	4^{10} sequences	128/16/20	Four-replicate affinity posterior; NE	BET-seq (Le et al., 2018; Fordyce Lab, 2018); CC BY 4.0
Superconductor *ranking*	16,795 records (12,179 groups)	2,985 groups	32/5/20	Median critical temperature by composition; NDCG	UCI Superconductivity Data (Hamidieh, 2018b;a); CC BY 4.0
UTR MRL *ranking*	76,877 50-mers	5,043 50-mers	128/16/10	Two-replicate mean ribosome load; NE	GEO GSE114002/mRNABench (Sample et al., 2018; 2019); terms unknown
GFP *ranking*	41,372 proteins	10,343 proteins	128/16/20	Protein-level median log brightness; NE	Sarkisyan Figshare (Bolotin, 2016; Sarkisyan et al., 2016); CC BY 4.0
Hopper Controller *ranking*	1,920 policies	1,280 policies	32/5/10	Mean of 500 frozen Hopper-v5 rollouts; NDCG	Design-Bench policy assets (terms not separately stated); Gymnasium (MIT), MuJoCo (Apache-2.0) (Trabucco et al., 2022; Todorov et al., 2012)
DrugMatrix (6) *ranking*	9,442 animal rows	390 conditions	16/5/8	Absolute log treatment–control deviation; NDCG	NIEHS CEBS DrugMatrix (National Toxicology Program (NTP), 2023); terms unspecified

Reconstruction principle SciModelingBench takes its scientific settings from Design-Bench (Trabucco et al., 2022), but does not automatically treat the packaged arrays as ground truth. For every retained setting, we return to the original experiment or simulator assets. We then define the candidate identity, visible observations, hidden reference data, and trusted objective separately. Agents may train predictive models to rank candidates, but those predictions are not used as ground truth outside the training distribution. We omit settings that do not support reproducible candidates and evaluation, rather than keep them only to match a historical learned oracle.

Transcription-factor binding For TFBind8, the original PBM table stores an 8-mer and its reverse complement in separate columns that share one E-score. The historical preprocessing joins these columns, producing 65,792 rows and duplicating the 256 reverse-complement palindromes. We remove only these exact duplicates, keep non-palindromic reverse complements as distinct valid sequences, and verify that the result contains all 4^8 8-mers exactly once. We also retain both the published and normalized E-score scales. The resulting complete measured landscape supports direct lookup without a learned oracle.

For TFBind10 Pho4, the legacy table contains replicate-level binding estimates. The same 10-mer can therefore have conflicting values, and using row order as a lookup rule silently selects one replicate. We rebuild the task from four BET-seq bound/input count replicates. The lower-half observations expose the raw replicate measurements, while a deterministic affinity posterior covers the complete 4^{10} domain and is oriented so that higher values are better. This keeps replicate disagreement visible instead of hiding it behind the final row.

Superconducting materials Design-Bench uses a fitted random forest to score arbitrary composition vectors. Composition alone, however, omits crystal structure, phase, pressure, defects, and processing. The source data also contain repeated compositions with different measured critical temperatures. We first realign the two UCI tables and normalize the amounts of the 86 elements. We then group proportional formulas under one canonical composition while retaining every measurement. The group median is the ranking target, and the measurement range and dispersion remain visible to the agent. Evaluation is limited to the measured high-temperature pool, so composition-only extrapolations are not treated as physical ground truth.

UTR and GFP sequence measurements The original UTR setting uses a learned ResNet to score arbitrary sequences. We instead use the unique 50-nucleotide variable regions measured in the eGFP reporter assay. We select the unmodified-RNA condition, use the mean ribosome load over two replicates, and hold out one group defined by (uAUG presence, Kozak quality) as a finite measured pool.

For GFP, translating the legacy nucleotide rows produces 56,086 rows but only 51,715 unique proteins. Among the 1,202 duplicated protein groups, 1,201 have conflicting labels, and the wild type alone appears 534 times. We instead use the authors' protein-level aggregate, verify the unique 237-residue sequences, and split the data by protein identity before adding nucleotide and barcode observations. This prevents synonymous encodings of the same protein from appearing on both sides of the split. Both tasks therefore evaluate measured candidates rather than predictions from historical neural-network oracles.

Hopper controller relabeling We keep Design-Bench's 3,200 policy vectors as candidates but replace the historical labels, each of which came from a single return. Every policy is evaluated over 500 stochastic episodes in Hopper-v5. Policies share the same reset-seed schedule but use separate action-noise seeds. The released data include raw returns, episode lengths, termination flags, and uncertainty summaries; mean return is the frozen target. From the offline observations, the agent ranks a held-out pool of higher-performing measured policies. It cannot access the simulator, train PPO, or request online rollouts during the task.

DrugMatrix reconstruction The legacy ChEMBL arrays represent each condition only by a molecular token, dropping its dose, duration, route, vehicle, study, sex, and animal-group context. Nearly every molecule then maps to several endpoint labels, and a row-level split can place the same molecular identity on both sides. We therefore rebuild the task from the NIEHS CEBS individual-animal clinical-pathology release instead of reusing these arrays. We preserve the treatment context and controls, map chemical identity by strict name and CASRN matching, and use ChEMBL only to link structures rather than supply endpoint labels.

Candidates are five-day treatments at the highest observed dose, with complete endpoints and controls matched by study, duration, route, vehicle, and sex. We hide the treatment-animal rows for these candidates but keep their matched controls and observations from other doses or times visible. All six tasks use the same candidate conditions and rank them by the absolute log difference between the endpoint mean and the matched-control mean. This score measures the size of a biological change, not drug safety or quality.

Protocol and metric audit Before running any agents, we fixed the lower-score thresholds, structured holdouts, submission sizes, and primary metrics using source-data analyses and task-specific random, simple, and cross-validated audits. These audits helped us avoid saturated metrics, distinguish strong submissions, and keep the queried fraction small relative to the hidden domain. The choices were not based on ScienceFlow or external-agent results. Here, an "exact" evaluator returns the same score whenever it is applied to the same frozen data. The underlying experimental measurements may still contain noise.

Runtime leakage controls and contamination scope The agent workspace contains only the exported observations, candidate view, manifest, task contract, and submission directory. Hidden labels, trusted objectives, evaluator caches, full dataset copies, and submission records stay outside the workspace, and external network access is disabled. The evaluator returns one score for the full batch rather than a score for each candidate. The harness separately records query use and the best valid artifact. These controls block runtime answer lookup through the benchmark infrastructure.

They do not show that public papers or upstream data were absent from model pretraining. We therefore claim runtime leakage controls and audited provenance, not a contamination-free evaluation. General scientific knowledge is allowed, although in some cases it is difficult to distinguish inference from memorized candidate labels.

Licensing and availability SciModelingBench and Design-Bench software use the MIT license, but this does not relicense the underlying experimental data. Table A12 therefore lists the original source-specific, CC BY 4.0, or unknown data terms. Because these terms differ, the dataset card lists

the license as "other" rather than assigning one license to the full collection. Processing records and task-specific details are available in the public suite documentation and dataset card.