FATHOM-AUTORESEARCH: TOWARDS RELIABLE AUTONOMOUS POST-TRAINING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autonomous post-training has a fixed deadline but delayed task-level feedback. An experiment can guide the remaining campaign only if its intervention is realized, exposed adequately, and evaluated in time to change the next decision. We introduce FATHOM, which couples evidence-grounded research reviews to continuous implementation and checkpoint retention. Across seven PostTrainBench tasks, its selected-checkpoint aggregate is 44.36% for Qwen3-1.7B-Base and 54.37% for Qwen3-4B-Base. It exceeds the persistent Codex CLI-Kimi K3 control in all eight displayed matched task model pairs. Across four tasks per model, effective GPU hours spent on selected-checkpoint work or completed evaluation are 16.68 versus 12.78 at 1.7B and 12.00 versus 8.67 at 4B. A frozen-state replay favors the FATHOM Planner's written experiments by 2.12 points on a 24-point rubric across 15 states, without testing online execution. Together these layers characterize finite-horizon experimental control rather than checkpoint score mediation by any single role.

1 INTRODUCTION

Recent advances in large language model (LLM) agents have rapidly expanded their role in AI Research and Development (AI R&D): they can write code, run experiments, inspect failures, and can manage long horizon jobs (Yao et al., 2023; Chan et al., 2025). Among AI R&D tasks spanning various categories , LLM post-training offers a particularly well-defined testbed for these capabilities because it tests its research, planning, execution and its ability to adapt based on environment feedback. PostTrainBench (Rank et al., 2026) first standardized the task: given a base model, benchmark and a 10 hour deadline, an agent must improve the base model and return a better checkpoint, spanning seven benchmarks across .

Doing posttraining successfully requires robust strategy planning, reliable execution and adapting based on execution feedback. Li et al. (2026c) finds that execution is broadly reliable across long-horizon AI R&D tasks, while solution framing and feedback control separate agents and expose failures hidden by final scores. Similarly, across the released posttrainbench trajectories, (Lim et al., 2026) show that the training strategy is locked in at the very beginning before the agent even runs any experiments, keeps spending the remaining budget doing local tweaks and in only about 2% of the released trajectories the agent backtracks from its initial plan. In addition, each agent has its own default strategy as the opening strategy (Rank et al., 2026; Lim et al., 2026) utilizing its own prior and too eager to launch training jobs Li et al. (2026a) without performing neither a literature review nor experiments. Our qualitative analysis of the trajectories also reveal that the agents spend re-discovering the facts and learnings based on execution feedback that are already present in the literature. For example, in most trajectories, the agents encounter generation issues like repetition and gibberish text in the trained checkpoints and spend time diagnosing that issue.

Even though current agents are often stronger at implementing a proposed AI R&D solution than at deciding which solution to pursue (Li et al., 2026c), execution in post-training is not limited to launching a training script. It also requires checking data and configuration formats, resolving dependency and library incompatibilities, and selecting memory-safe settings for the target model and hardware. FT-Dojo (Li et al., 2026b) shows that general-purpose agents can lose substantial time to malformed configurations, deprecated library arguments, broken data pipelines, and out-of-memory failures, motivating validation before full-scale training. More broadly, implementing fine-tuning methods across different model families requires non-trivial model-specific engineering.

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.

We therefore distinguish two capabilities. *Discovery loop* involves evidence grounded planning which selects and specifies the next experiment under the remaining budget and available resources, followed by a feedback-conditioned revision to continue, repair, switch, or stop that direction. *Execution realization* turns the authorized intervention into durable checkpoints and interpretable measurements despite operational failures. This decomposition lets us test decision quality separately from plan-to-evidence conversion.

A natural design is to separate scientific planning from experiment execution, but role separation alone is not the solution. The benefit of multiple agents depends on the task and its coordination demands (Kim et al., 2025; 2026b). Here the two responsibilities need different context lifetimes: a research review should be able to question the previous scientific direction without inheriting an ever-growing conversation of commands and repairs, while execution benefits from retaining knowledge of running jobs, configurations, and earlier failures (Li et al., 2026b).

This asymmetry creates a state problem. Starting every review from scratch would lose experimental evidence, but replaying the full execution transcript would return irrelevant operational detail to the planning context. Prior work motivates selecting relevant context rather than transferring either all or none of it (Ruan et al., 2026). Moreover, conclusions alone are not enough: a reviewer may need to inspect the measurements, code, or failed attempt behind them. We therefore need a durable research record in which concise accounts of progress remain connected to the original artifacts. ScienceFlow uses recoverable research states with retrievable underlying results, while CORAL demonstrates coordination through shared persistent files (Zhao et al., 2026b; Qu et al., 2026).

We propose FATHOM, a multi-agent system organized around two decision-making roles and a shared filesystem research record. At each research review, a fresh Planner examines the recorded evidence and remaining budget to select a hypothesis, a feasible next experiment, and the result that would warrant continuing or changing direction. A persistent Executor implements the published program, handles operational failures within its bounds, evaluates results, and preserves valid checkpoints. Plans, costs, logs, evaluations, and checkpoint lineage remain available in the filesystem rather than a shared conversation. Either role may use bounded, temporary helpers for separable work, but the Planner retains scientific authority and the Executor retains execution authority. Our contributions are threefold:

1. We introduce FATHOM, a discovery loop in which the Planner directs each intervention using task-level evidence and the remaining horizon. Planner interventions address early strategy lock-in, while continuous execution carries authorized experiments through training, evaluation, and checkpoint retention.
2. We present baseline comparisons against a persistent Codex agent as baseline under identical task, agent & hardware setup and a ten-hour budget. FATHOM obtains a higher final score in all eight task-model comparisons. its mean score is 34.12 versus 23.54 on Qwen3-1.7B-Base and 47.29 versus 33.26 on Qwen3-4B-Base.
3. We present further analysis of the research trajectories generated by FATHOM to study how the Planner's interventions guide the discovery loop: which experiments receive compute, when complete evaluations arrive, and how the resulting evidence changes the next experiment and final checkpoint choice.

On the seven PostTrainBench tasks and Qwen3 base models, FATHOM achieves the best scores beating Locus at both 1.7B (44.36% vs. 43.96%) and 4B (54.37% vs. 53.62%) model sizes. (Rank et al., 2026).

2 RELATED WORK

Autonomous post-training and AI research agents. MLE-bench (Chan et al., 2025) studies agents performing end-to-end machine-learning engineering, while PostTrainBench (Rank et al., 2026) evaluates autonomous LLM post-training under a fixed deadline. FT-Dojo (Li et al., 2026b) develops a fine-tuning agent that plans iterations, validates configurations, and revises its approach using evaluation feedback. Agent^2 RL-Bench (Chen et al., 2026b) examines whether agents can construct and execute complete post-training pipelines, including interactive RL. (Li et al., 2026c) shows why final outcomes alone can obscure failures in solution framing, execution, and feedback

control. AIRA2 (Hambardzumyan et al., 2026) demonstrates a different route to stronger autonomous research through asynchronous workers and more reliable evaluation. These works establish both the capabilities of research agents and the importance of the environment in which experiments run. Our study focuses on how a fixed-budget post-training campaign allocates effort between research review and execution.

Hypothesis selection and experiment allocation. Separating research decisions from implementation already has substantial precedent. (Yang et al., 2025) assigns idea generation and code development to different agents while Curie (Kon et al., 2025) distinguishes an Architect from Technicians and validates experimental steps. ARTS (Juneja et al., 2026) has a Scientist select hypotheses that an Executor implements. Other work varies the search policy itself: AIRA-Dojo (Toledo et al., 2025) studies search operators, while Heuresis (Antoniades et al., 2026) finds that broader search or greater diversity need not produce stronger or more novel discoveries. StatefulDiscovery (Chen et al., 2026a) connects research claims to accumulated evidence and ScienceFlow (Zhao et al., 2026b) combines recoverable research states with execution decisions conditioned on resources and validated progress. Thus our contribution is neither the first scientist-executor split nor a new search algorithm. We examine whether a fresh review of evidence and the remaining horizon changes which *complete post-training experiment* receives compute, while its execution remains continuous.

Coordination and persistent research state. Role separation also has costs: controlled multi-agent studies Kim et al. (2025) find that its value depends on task structure, and TeamBench (Kim et al., 2026b) exposes coordination failures under enforced planner, executor, and verifier roles. AOrchestra (Ruan et al., 2026) finds that selecting task-relevant context can outperform both full-context inheritance and no inherited context in its sub-agent setting. CORAL (Qu et al., 2026) demonstrates asynchronous coordination through shared persistent filesystem artifacts, whereas AstronOS (Nie et al., 2026) maintains versioned, accepted work state across agent invocations. Preserving the underlying evidence matters as well: a controlled conversation-memory study (An, 2026) finds that replacing original text with extracted artifacts can lose answer-relevant detail. These are precedents and cautions, not evidence that our particular arrangement must win. We test a fresh Planner, persistent Executor, and filesystem-only shared campaign record against a persistent single-agent control under matched research materials and resources.

3 POST-TRAINING AS A DELAYED-FEEDBACK DISCOVERY LOOP

An autonomous post-training agent receives many observations but comparatively few scientific results. Throughput, memory use, trainer loss, reward, and checkpoint creation are available during execution. These signals reveal whether an intervention is being implemented, but they do not establish whether it improves the target behavior. A scientific result arrives only after the agent has constructed the data, delivered enough exposure to test the proposed mechanism, produced a usable checkpoint, and completed a comparable evaluation. We call this sequence - propose, realize, measure, and revise - a *discovery loop*. A loop closes only when its result can support a decision about the remaining campaign.

An experiment is a complete decision unit: a parent checkpoint, an intervention and learning signal, a predicted behavioral change, and a next action conditioned on the result. A written plan only affects the submitted checkpoint only if it includes all the necessary details while only a plausible method name alone supplies none of these links. (Juneja et al., 2026; Qu et al., 2026; Zhao et al., 2026a) supply precedents for selecting bounded hypotheses, revising after measurement, and preserving recoverable research state . Evidence selection matters because a research review needs the relevant results without inheriting every operational event, while implementation benefits from retaining code, jobs, and repairs (Ruan et al., 2026; Kim et al., 2026a; Liu et al., 2026). This motivates distinct context policies for the two capabilities.

Let $s_t = (\mathcal{E}_t, \theta_t^\star, b_t)$ contain the available evidence, the best comparably evaluated checkpoint, and the remaining budget. For a feasible experiment e, $\widehat{\tau}_e$ estimates latency through completed evaluation; Y_e contains realized validity, exposure, behavior, and score. A structural objective is

$$V(s_t) = \max_{e \in \mathcal{A}(s_t):\, \widehat{\tau}_e \le b_t} \mathbb{E}_{Y_e, \tau_e}[V(s_t \oplus (e, Y_e, \tau_e))]\,, \tag{1}$$

with terminal value $V(s) = S(\theta^\star)$. This is the sequential-design value of information under a deadline (Huan & Marzouk, 2016; Shen & Huan, 2021). The formulation makes three practical questions explicit: where the campaign's time and compute go, whether results change subsequent training decisions appropriately, and whether proposed experiments can produce results in time to inform those decisions. The literature above suggests that each can fail even when an agent successfully launches training. We examine them through the following claims.

Claim 1: discovery-loop yield under a fixed horizon. Under a fixed resource and time budget, we ask how time is allocated across GPU work and agent reasoning, and how many GPU-Hours contribute to the final selected checkpoint, distinguishing productive compute from compute that consumes GPU time without advancing the final checkpoint score. Agent reasoning is tracked separately, since it can inform candidate selection or prevent unproductive runs without directly contributing GPU-hours. We evaluate the claim by measuring these components over the campaign horizon.

Claim 2: feedback-conditioned scientific revision. A low score does not, by itself, determine the next treatment. The campaign should distinguish a failed implementation, insufficient exposure, an adequately tested but ineffective learning signal, and regression against an incumbent. These cases can warrant repair, continuation, a switch or stop, and rollback, respectively, subject to the remaining horizon. Executed trajectories test whether subsequent allocations follow these distinctions; more frequent switching is not the objective.

Claim 3: complete, diagnostic experiment design. A proposed experiment should do more than name a plausible method. It should link a diagnosed bottleneck to a suitable learning signal and task-preserving data, specify exposure and measurement that can test the predicted change, and leave feasible next actions for different outcomes. Under identical evidence, model, tools, and output requirements, evidence-grounded guidance should produce more such decision-ready programs than format-only instructions. The frozen-state replay tests written program quality, not execution or checkpoint improvement.

4 FATHOM-AUTORESEARCH

Choosing the next scientific treatment requires an independent comparison across alternatives and the remaining horizon, whereas realizing that treatment benefits from continuous knowledge of code, data, jobs, and model behavior. FATHOM implements the discovery loop through two authority boundaries and a shared filesystem record. At each review, a fresh Planner chooses what should be tested over the remaining horizon; one persistent Executor decides how to realize the authorized experiment while retaining knowledge of code, data, jobs, and operational failures. Both roles use the same researcher model and external evidence, but differ in context lifetime and in the decisions they may change.

Horizon-aware program design. At the opening and each consequential review, the Planner reconstructs s_t from the task, evaluator, prior results, sealed literature, and remaining time and compute. It compares a small set of complete data-to-evaluation paths, including data construction, training, checkpoint preparation, and comparable measurement, rather than ranking method names in isolation. A candidate is eligible only when its checkpoint, data, learning signal, and implementation are supported by the supplied materials and available interfaces. For each candidate, the Planner states the targeted failure, predicted behavioral change, required exposure, and how positive, negative, or inconclusive evidence would affect the remaining campaign. Experiments count as distinct only when they differ on at least two of signal source, bottleneck, objective, data policy, and lineage.

Program publication and revision. The Planner publishes an immutable program $P_t = (O_t, a_t, \delta_t)$, where O_t ranks routes over the remaining horizon, a_t specifies the authorized allocation, and δ_t maps evaluated outcomes to continuation, stopping, or renewed review. The resulting `PLAN.md` is published atomically and defines the Executor's scientific boundary. A weak result does not itself trigger a switch: insufficient exposure may justify a fuller test, whereas a sufficiently exposed failure can redirect the remaining budget. Each review plans over the full residual horizon;

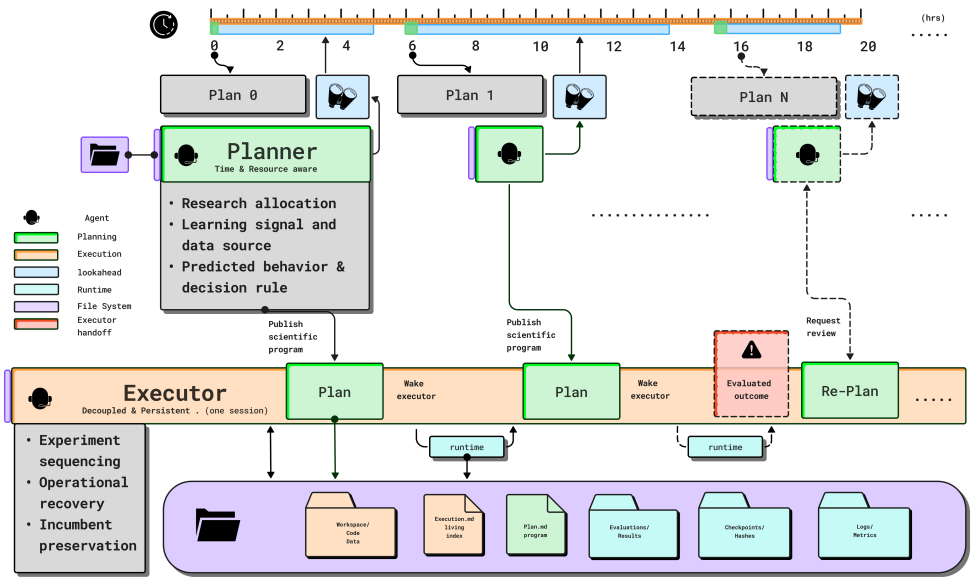

Figure 1: FATHOM's decoupled planning and execution architecture. At each review, a fresh, time- and resource-aware Planner compares complete experiments over the remaining horizon and publishes a scientific program. The program is injected through the shared filesystem without resetting the persistent Executor or stopping admitted work. Results, costs, and checkpoint lineage supply the next review; if evaluated evidence contradicts the program, the Executor can request an earlier review.

review boundaries determine when the scientific treatment may be reconsidered, not a separately varied cadence.

Persistent, bounded execution. The Executor constructs data, implements and validates the trainer, schedules jobs, handles operational failures, evaluates checkpoints, and retains the strongest valid result. Operational feedback may change how an experiment is realized, but not its authorized source family, objective, stage order, or training range. The rule δ_t allows repairs and continuation without a new Planner invocation for every event. If decision-relevant evidence contradicts the program or exhausts its authorized continuations, the Executor requests a fresh review before changing the scientific treatment. Admitted work continues during that review, and the new program governs the next GPU action without resetting the Executor session.

Filesystem as persistent research state. The roles do not share a chat. `PLAN.md` records the scientific commitment, while `EXECUTION.md` indexes the realized treatment and links its code, data, logs, evaluations, costs, and checkpoint lineage. Each fresh Planner therefore receives the prior decision and its evidence without inheriting the full debugging transcript, while the persistent Executor retains implementation continuity.

5 ANALYSIS

We study post-training of Qwen3-1.7B-Base and Qwen3-4B-Base on seven PostTrainBench-derived tasks: AIME 2025, ArenaHard-Writing, BFCL, GPQA-Main, GSM8K, HealthBench-Easy, and HumanEval. Kimi-K3 is the researcher model across all experiments. Each campaign has one H100 and ten active campaign hours. Our control is a persistent Codex-CLI session with native subagent delegation available; FATHOM instead introduces fresh, explicit Planner interventions while retaining a persistent executor, each powered by Codex-CLI and capable of native multi-agent

delegation. Both systems use the same student checkpoint, task specification, evaluator, decoding settings, tools, datasets and teacher models and Docker environment. Candidate datasets and research papers were assembled before the campaigns and made available through a fixed task-specific collection. Campaigns cannot browse for additional material. We compare the systems on AIME, ArenaHard-Writing, GPQA, and HealthBench; Appendix A gives the materials, access conditions, and evaluation settings. These are local experiments on PostTrainBench-derived tasks, not an exact reproduction of the official setting. Published leaderboard scores are external context (Rank et al., 2026; PostTrainBench Team, 2026).

5.1 RESULTS

Table 1: PostTrainBench score-summary checkpoints (%). FATHOM cells show the supplied mean (variance); external rows are published point values and use a different execution environment providing only benchmark context. Avg follows the benchmark's weighted aggregate. † marks Fable's published GPQA substitution.

Harness / researcher	Avg	AIME	ArenaHard-Writing	BFCL	GPQA	GSM8K	HealthBench-Easy	HumanEval
Qwen3-1.7B-Base								
Locus / Opus 5	43.96	4.44	58.57	93.33	**33.63**	81.69	**43.49**	70.60
Fable 5†	38.95	10.00	**65.04**	48.00	25.89	**84.80**	35.57	65.24
Codex CLI / GPT-5.6-sol	34.85	5.00	27.47	**94.50**	29.13	59.82	26.89	66.77
Claude Code / Kimi-K3	30.03	4.44	32.37	0.00	27.83	77.23	32.47	62.60
FATHOM / Kimi-K3	**44.36** ± 1.15	**10.00** ± 3.33	52.00 ± 7.36	93.67 ± 0.58	33.56 ± 1.57	83.47 ± 1.04	40.99 ± 1.87	**70.73** ± 2.20
Qwen3-4B-Base								
Locus / Opus 5	53.62	20.74	**86.78**	95.67	37.40	90.23	**45.30**	82.52
Fable 5†	51.18	25.00	85.70	97.50	31.47	89.42	35.03	81.10
Codex CLI / GPT-5.6-sol	41.18	10.00	32.86	94.50	34.04	80.06	27.07	82.62
Claude Code / Kimi-K3	41.03	22.22	48.25	32.67	32.22	86.43	29.51	79.47
FATHOM / Kimi-K3	**54.37** ± 1.01	**30.00** ± 3.33	81.47 ± 6.14	**95.00** ± 1.00	**37.50** ± 0.89	**90.83** ± 1.03	40.17 ± 1.42	**82.93** ± 1.22

Table 2: Matched local score-summary checkpoints (%). Cells show the supplied mean (variance). Codex is the single persistent agent; FATHOM is the research and execution system.

Student	Agent	AIME 2025	ArenaHard-Writing	GPQA-Main	HealthBench-Easy
Qwen3-1.7B	Codex	4.44 ± 1.92	24.29 ± 14.37	31.70 ± 3.79	33.74 ± 2.58
Qwen3-1.7B	FATHOM	10.00 ± 3.33	52.30 ± 7.36	33.56 ± 1.57	40.99 ± 1.87
Qwen3-4B	Codex	23.33 ± 3.33	44.23 ± 22.14	32.74 ± 1.00	32.76 ± 2.16
Qwen3-4B	FATHOM	30.00 ± 3.33	81.47 ± 6.14	37.50 ± 0.89	40.17 ± 1.42

On the seven-task aggregate, FATHOM scores 44.36% at 1.7B and 54.37% at 4B (Table 1). The external rows help locate those values among published systems but are not the matched control. Against this control, FATHOM improves on all eight tasks, model cells in Table 2, with gains of $+1.86$ to $+27.71$ points at 1.7B and $+4.76$ to $+37.24$ points at 4B. The largest gains occur on ArenaHard-Writing ($+27.71$ and $+37.24$ points), while the improvements on AIME 2025, GPQA-Main, and HealthBench-Easy range from $+1.86$ to $+7.41$ points. These differences are based on the displayed checkpoint means. The following analyses ask what changed in the research process.

5.2 EFFECTIVE GPU-HOUR ALLOCATION

We classify each part of campaign's to one of four categories: 1) *Selected checkpoint work* is any cascading training that resulted in the submitted checkpoint, plus GPU utilization for synthetic data generation used in training; a discarded branch does not enter this category. 2) *Completed evaluation* is a finished assessment of a base model or candidate checkpoint, including assessments that might have led to rejection. 3) *Other GPU work* covers training for rejected branches, generated data never used by the selected checkpoint and incomplete evaluations. Some of this work can still teach the agent which direction to abandon. 4) *No GPU operation* is the remainder of the wall-clock window; it can contain CPU work, research, waiting, or tool calls, and these activities may or may not overlap

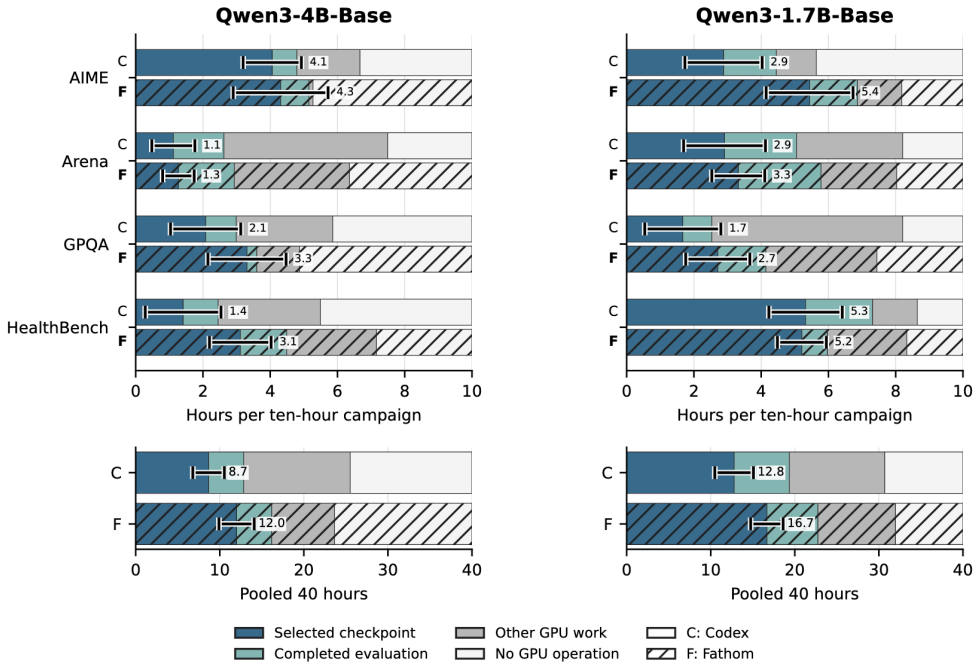

Figure 2: Campaign time allocation by task and student size, with pooled 40-hour comparisons below. C is Codex; hatched F is FATHOM. Other GPU work and no-GPU time are distinct categories, not waste or device-utilization estimates. Source data retain exact means and supplied presentation variances.

with GPU operations. The four categories account for charged campaign time, not device occupancy or the researcher's exclusive activity. Figure 2 shows larger pooled effective totals for FATHOM at both model sizes.

Across the four 4B tasks, FATHOM devotes 12.00 ± 2.10 of its 40 pooled hours to the selected checkpoints and the data behind them, against 8.67 ± 1.88 for Codex, and the same ordering holds at 1.7B, where selected-checkpoint work rises to 16.68 ± 1.93 hours versus 12.78 ± 2.30. Completed evaluation, by contrast, is essentially matched at 4B (4.17 ± 1.76 versus 4.18 ± 1.44 hours) and slightly lower for FATHOM at 1.7B (6.07 ± 1.71 versus 6.56 ± 1.88). Combining the two, the time spent producing the selected artifact *or* a completed measurement is 16.17 ± 2.74 versus 12.85 ± 2.36 hours at 4B and 22.75 ± 2.58 versus 19.34 ± 2.97 at 1.7B. Evidence grounding indicates an implicit but efficient allocation of compute toward decision-relevant work. Appendix E provides the data, attribution, and normalization procedure; all reported intervals are one standard deviation, pooled over the four task clocks (with combined figures adding the component variances).

5.3 EXECUTED DECISIONS AFTER FEEDBACK

Our first mechanism claim is that explicit review prevents a campaign from remaining committed to its opening strategy after the evidence has changed. We examine that claim in two archived Qwen3-4B campaign pairs. The question is whether evaluation arrived early enough to affect the next allocation and whether the next experiment followed from the observed result. These online traces show decisions that were actually executed, but they are separate reporting snapshots from the score-summary means in Table 2, and the paired agents did not receive identical intermediate outcomes. Section 5.4 therefore follows with a controlled, same-evidence study of the planning decision itself.

AIME: early evaluation preserved the option to respond. Both systems identified answer termination and format matching as important failures and both generated long-reasoning supervision. The baseline then committed roughly 103 million student tokens before its first complete evaluation

at campaign hour 5.00. That evaluation still contained empty and truncated answers. The SFT run was useful, it produced the retained 23.33% checkpoint, but half of the campaign had passed before the remaining failure was measured. A later GRPO attempt produced no evaluated RL checkpoint, and a short-trace branch regressed. The problem was not choosing SFT; it was committing most of the first half of the horizon before learning whether that treatment was sufficient.

FATHOM instead scheduled a smaller verified exposure and obtained a complete evaluation at hour 2.15. The result was positive but incomplete. With enough time left to respond, the Planner retained the supported long-reasoning hypothesis but added fresh hard problems for the next supervised run. Successive complete evaluations reached 13.33%, 20.00%, and 33.33%. This case shows that avoiding strategy lock-in does not require changing methods at every review: it requires measuring before a large commitment and then continuing only when the result supports continuation.

GPQA: a measured plateau triggered a change of signal. The FATHOM trajectory began with unparseable final answers. Science SFT repaired much of the output-format failure and reached 34.15% after its second epoch. A subsequent 125-step GRPO run reached 34.60%, but reward had flattened and the score remained within the predeclared tie band. Continuing the same treatment could no longer be justified by measured progress.

The baseline followed additional supervised branches in its archived campaign and ended at 16.96%. Those branches remained close to the opening treatment family and did not produce an evaluated test of a different signal for the remaining reasoning errors. FATHOM had instead specified that residual reasoning failures after format repair would warrant a signal change. After the plateau, the Planner switched to rejection-sampling fine-tuning on verified on-policy solutions and retained the SFT checkpoint as a rollback. The new branch reached 37.50% on an intermediate evaluation, while the prespecified full-set reevaluation returned 34.60%; the repeated result, rather than the favorable peak, governed the final claim. The contrast illustrates the strategy-lock-in mechanism directly: the baseline continued variations of its initial supervised strategy, whereas the Planner used the plateau to test a different learning signal without making that new branch irreversible.

The same pattern produces different actions for different evidence. BFCL illustrates *repair*: residual tool-argument errors led to revisions of formatting and default-argument conventions. HumanEval illustrates *switch and rollback*: generic supervision regressed on the complete coding evaluation, so the next allocation targeted harder cases with GRPO while preserving the stronger checkpoint. GSM8K illustrates *stop and compare*: saturated rejection-sampling headroom and a flat RL diagnostic stopped a long continuation, and complete evaluations governed finalist selection. These are archived process examples; their coverage scores in Table 1 are separate summaries. The negative cases are equally important. In the archived 4B Arena trajectory, revisions to data and preference signals did not prevent the final FATHOM checkpoint from remaining below both the control and the base model. In HealthBench, recovery from an external GPU obstruction preserved a usable checkpoint, but a simpler control mixture scored higher. Planning therefore does not guarantee a win; the supported claim is narrower: available evidence changed whether the system continued, repaired, switched, rolled back, or stopped.

5.4 FROZEN-STATE STUDY OF PLANNING DECISIONS

The online trajectories show how allocations changed in running campaigns, but their intermediate evidence differed. To isolate whether the Planner makes a better decision from the *same* available evidence, we conduct a frozen-state study across 15 decision points sampled from the trajectory corpus. Eight states are initial planning problems: AIME 2025, ArenaHard-Writing, GPQA-Main, and HealthBench-Easy, each at the Qwen3-1.7B-Base and Qwen3-4B-Base model sizes. Seven states instead provide the outcome of a previous experiment and ask what should be funded next; they cover training dose, tool-argument errors, answer closure, RL progress, medical-data generation, and coding regression. For each state, the FATHOM Planner and a format-only planner receive exactly the same evidence and each produce three independent research plans. Two blinded scoring sessions rate every plan from 0 to 4 on six criteria: 1) diagnosis of the problem, 2) treatment and data design, 3) preservation of task information in the training view, 4) adequacy of the proposed dose, 5) measurement quality, and 6) feasibility of continuation. Scores therefore range from 0 to 24. We average the two ratings and three plans within each state, then compare the two planners across states. Ten states were held out from policy development and receive a separate analysis (Appendix D).

The format-only planner was required to name one program together with its evidence, measurement, falsifier, rollback, resource assumptions, and review time. The FATHOM Planner additionally had to compare causally distinct programs, distinguish supervision information from its trainer representation, relate the proposed dose to empirical onset and the complete data-to-evaluation critical path, preserve the incumbent and checkpoint lineage, and place review where a decision-grade result could change the program. The replay therefore isolates substantive planning guidance while holding the research model, evidence, tools, resource declaration, and output contract fixed; it does not test execution or role separation by itself.

Table 3: Quality of the proposed next research program (0 to 24; higher is better). Each state averages three independent plans and two blinded ratings before state scores are averaged. The format-only planner is a planning control, separate from the Codex campaign agent. The new-state row overlaps the opening and feedback rows.

Decision states	n	FATHOM Planner	Format-only planner	Difference
All	15	23.59	21.47	+2.12
Opening	8	23.62	21.29	+2.33
After feedback	7	23.55	21.67	+1.88
New states	10	23.57	21.45	+2.12

The FATHOM Planner scores 23.59 versus 21.47 overall, a paired advantage of 2.12 points with a state-bootstrap 95% interval $[1.63, 2.62]$; it leads at all 15 states. The gain persists on the ten new states (2.12 points, $[1.68, 2.55]$). At the eight openings, where an uninformative first run can consume the horizon, the gain is 2.33 points $[1.71, 3.06]$. After feedback it is 1.88 points $[1.14, 2.55]$. The advantage therefore appears both before the first commitment and after new evidence creates an opportunity to revise the current strategy.

6 LIMITATIONS

The checkpoint tables, charged-hour summaries, and archived trajectories use different evidence units. Their joint pattern is consistent with the two control claims, but it does not identify the pathway from a particular plan to a displayed score. The eight matched score cells are descriptive checkpoint outcomes; the certified archived process pairs have one campaign per arm and do not support population-level significance. The figure summarizes supplied three-run means after row normalization, while its parenthetical variances are presentation estimates rather than empirical run variance. Some checkpoint lineage is unresolved in the broader archive. The historical process campaigns used revision-scoped Executors; they cannot isolate the effect of the present persistent-Executor design. The frozen replay uses the same model family for plan production and blinded scoring, and no replayed plan was trained. A prospective matched execution study would be needed to estimate component effects and total cost.

7 CONCLUSION

Finite-horizon post-training depends on which complete experiments fit before the deadline and what the agent does once their results arrive. The displayed checkpoint comparisons measure the end-to-end system; the effective GPU-hour figure describes allocation; executed trajectories show repair, continuation, signal change, stopping, and rollback; and the frozen replay measures written plan quality under identical evidence. Together these results support a control architecture that chooses, realizes, evaluates, revises, and retains experiments. They do not establish that a separate Planner or persistent Executor is individually causal.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, or interpreting results, and the remaining required-disclosure tasks of generating synthetic data sets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, assisting with translation, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for creating and editing software code and editing a research paper to improve readability. We have reviewed all AI-assisted work. AI-generated code was verified and tested for correctness by the authors, and all AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI. Additionally, the system under study is an LLM research agent.

REFERENCES

Tao An. Fidelity before structure: Verbatim chunks beat lossy artifact extraction in long-conversation LLM memory, 2026. URL https://arxiv.org/abs/2601.00821.

Antonis Antoniades, Deepak Nathani, Ritam Saha, Alfonso Amayuelas, Ivan Bercovich, Zhaotian Weng, Vignesh Baskaran, Kunal Bhatia, and William Yang Wang. Heuresis: Search strategies for autonomous ai research agents across quality, diversity and novelty, 2026. URL https://arxiv.org/abs/2606.25198.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.07095.

Jiayao Chen, Shi Liu, and Linyi Yang. StatefulDiscovery: Evidence-calibrated claim formation in open-ended scientific discovery, 2026a. URL https://arxiv.org/abs/2606.11851.

Wanyi Chen, Xiao Yang, Xu Yang, Tianming Sha, Qizheng Li, et al. Agent^2 rl-bench: Can llm agents engineer agentic rl post-training?, 2026b. URL https://arxiv.org/abs/2604.10547.

Karen Hambardzumyan, Nicolas Baldwin, Edan Toledo, Rishi Hazra, Michael Kuchnik, Bassel Al Omari, Thomas Simon Foster, Anton Protopopov, Jean-Christophe Gagnon-Audet, Ishita Mediratta, Kelvin Niu, Michael Shvartsman, Alisia Lupidi, Alexis Audran-Reiss, Parth Pathak, Tatiana Shavrina, Despoina Magka, Hela Momand, Derek Dunfield, Nicola Cancedda, Pontus Stenetorp, Carole-Jean Wu, Jakob Nicolaus Foerster, Yoram Bachrach, and Martin Josifoski. AIRA2: Overcoming bottlenecks in ai research agents, 2026. URL https://arxiv.org/abs/2603.26499.

Xun Huan and Youssef M. Marzouk. Sequential bayesian optimal experimental design via approximate dynamic programming, 2016. URL https://arxiv.org/abs/1604.08320.

Gurusha Juneja, Arnav Kumar Jain, Deepak Nathani, William Yang Wang, and Xin Eric Wang. Learning the arts of search for automated discovery, 2026. URL https://arxiv.org/abs/2606.21891.

Joongwon Kim, Wannan Yang, Kelvin Niu, Hongming Zhang, Yun Zhu, et al. Scaling test-time compute for agentic coding, 2026a. URL https://arxiv.org/abs/2604.16529.

Yubin Kim, Ken Gu, Chanwoo Park, et al. Towards a science of scaling agent systems, 2025. URL https://arxiv.org/abs/2512.08296.

Yubin Kim et al. TeamBench: Evaluating agent coordination under enforced role separation, 2026b. URL https://arxiv.org/abs/2605.07073.

Patrick Tser Jern Kon, Jiachen Liu, Qiuyi Ding, Yiming Qiu, Zhenning Yang, Yibo Huang, Jayanth Srinivasa, Myungjin Lee, Mosharaf Chowdhury, and Ang Chen. Curie: Toward rigorous and automated scientific experimentation with AI agents, 2025. URL https://arxiv.org/abs/2502.16069.

Aochong Oliver Li, Haoli Yin, Rishabh Adiga, Darren Teh, Siddharth Joshi, Bogdan Gaza, David Schwab, Ari Morcos, and Pratyush Maini. DataSmith: Automating data research. *DatologyAI Blog*, August 2026a. https://www.datologyai.com/blog/datasmith.

Qizheng Li, Yifei Zhang, Xiao Yang, Xu Yang, Zhuo Wang, Weiqing Liu, and Jiang Bian. FT-Dojo: Towards autonomous LLM fine-tuning with language agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b. URL https://arxiv.org/abs/2603.01712.

Yiwei Li et al. Beyond final scores: A systematic evaluation of agents for long-horizon AI R&D, 2026c. URL https://arxiv.org/abs/2608.13417.

Joy Jia Yin Lim, Xin Huang, Hao Peng, Yaxi Lu, Xin Cong, Zhong Zhang, Maosong Sun, and Yankai Lin. What is missing from ai post-training ai: An empirical analysis, 2026. URL https://arxiv.org/abs/2608.19072.

Haozhe Liu, Tian Ye, Sensen Gao, Qihang Cao, Yitong Li, Mingchen Zhuge, Duomin Wang, Ruihua Zhang, Ping Luo, Jiawang Bian, Lei Zhu, Ligeng Zhu, Enze Xie, and Song Han. SoL-Pi: Recursively scaling auto-research loops for efficient agent harness, 2026. URL https://arxiv.org/abs/2609.20519.

Zhenhang Nie, Gui Zheng, Xudong Sun, Tailong Zhu, and Bin Zhang. AstronOS: A unified execution model and runtime for long-horizon agentic systems, 2026. URL https://arxiv.org/abs/2608.16381.

PostTrainBench Team. PostTrainBench v1.1 leaderboard. https://posttrainbench.com/, 2026. Official leaderboard and score data, website commit 22705452f80106da9ff841b4db24df7128bb9057, accessed 2026-09-23.

Ao Qu, Han Zheng, Zijian Zhou, Yihao Yan, Yihong Tang, Shao Yong Ong, Fenglu Hong, Kaichen Zhou, Chonghe Jiang, Minwei Kong, Jiacheng Zhu, Xuan Jiang, Sirui Li, Cathy Wu, Bryan Kian Hsiang Low, Jinhua Zhao, and Paul Pu Liang. CORAL: Towards autonomous multi-agent evolution for open-ended discovery, 2026. URL https://arxiv.org/abs/2604.01658.

Ben Rank, Hardik Bhatnagar, Maksym Andriushchenko, et al. PostTrainBench: Can LLM agents automate LLM post-training?, 2026. URL https://arxiv.org/abs/2603.08640.

Jianhao Ruan et al. AOrchestra: Automating sub-agent creation for agentic orchestration, 2026. URL https://arxiv.org/abs/2602.03786.

Wanggang Shen and Xun Huan. Bayesian sequential optimal experimental design for nonlinear models using policy gradient reinforcement learning, 2021. URL https://arxiv.org/abs/2110.15335.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Baldwin, et al. Ai research agents for machine learning: Search, exploration, and generalization in mle-bench, 2025. URL https://arxiv.org/abs/2507.02554.

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, 2025. URL https://arxiv.org/abs/2505.14738.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Mingming Zhao, Jiqian Dong, Kangping Xu, Zadid Hasan, Chengrui Fan, et al. ScienceFlow: A long-horizon agent for ML research, scientific discovery and beyond, 2026a. URL https://arxiv.org/abs/2608.14354.

Mingming Zhao, Jiqian Dong, Kangping Xu, Zadid Hasan, Chengrui Fan, et al. ScienceFlow: A long-horizon agent for ML research, scientific discovery and beyond, 2026b. URL https://arxiv.org/abs/2608.14354.

A CONTROL SYSTEM AND EVIDENCE CONSTRUCTION

Codex control. The local control runs Kimi-K3 in a continuing Codex CLI session. Codex's native multi-agent feature is enabled, with at most six concurrent threads and one level of subordinate agents. The root agent can delegate bounded work, but remains responsible for research decisions, training, evaluation, and checkpoint selection. The control has no scheduled fresh Planner intervention. Thus "Codex" denotes the local Kimi-K3 control.

Candidate datasets and literature. A single task-spanning Kimi-K3 DeepResearch request was used to identify candidate public training datasets and primary research papers. The retrieved sources, rather than only a generated summary, were retained and organized by task. The collection excludes PostTrainBench pages, leaderboards, published trajectories, prior-run strategies, and papers about agent harnesses. It was combined with task specifications, evaluator materials, baseline records, and pinned TRL documentation, then frozen at source revision `e065777`. No in-campaign web search can change the available evidence. The same fixed, versioned collection is the intended input to both systems in a matched condition. Dataset and paper lists describe available candidates; agents select what to use during their campaigns.

Preparing this collection is an offline research cost. It is excluded from the ten-hour within-condition comparison because it precedes both systems, but would need to be counted when estimating the cost of deploying the complete workflow. The archived collection retains source files and task-specific indexes. It does not retain the exact DeepResearch prompt, invocation metadata, retrieval cost, complete human-review checklist, or a collection-wide decontamination report. We therefore do not claim that the collection can be independently reconstructed from a fully logged DeepResearch run or that every source passed a documented collection-wide contamination audit.

A.1 RELATIONSHIP TO POSTTRAINBENCH V1.1

Official setting. PostTrainBench v1.1 assigns an agent one base checkpoint and one benchmark, one H100, and ten hours of whole-run wall time to produce a final model. The clock includes everything the agent can do: research, data preparation, teacher inference if used, training, evaluation, and submission. Agents receive an official prepared Hugging Face cache, terminal and web access, benchmark evaluators, and a decontamination tool. They may research general methods but may not consult PostTrainBench pages, its repository, leaderboard, published trajectories, or prior-run strategies. Final evaluation uses the supplied evaluator and the submitted model's `generation_config.json`. The v1.1 review includes contamination, API-use, and benchmark-lookup checks, model-identity checks, and selected manual review (Rank et al., 2026; PostTrainBench Team, 2026).

Differences in our setting. Our campaigns use an H100, a ten-hour wall clock, and a frozen information environment without web research while the official benchmark uses an H100, a whole-run wall clock, web access, and its own prepared cache. Although that cache exists, its complete versioned informational contents and the exact package, model, and dataset revisions underlying every published run are not documented sufficiently for independent byte-level reconstruction.

B EXPERIMENTAL INPUTS SUPPLIED TO THE RESEARCHER

B.1 FIXED INFORMATION CONDITION

All locally compared agents received the same versioned task information before treatment assignment. The supplied pack comprised the assigned student checkpoint, task specification, evaluation target and

evaluator, fixed evaluation decoding configuration, candidate training datasets, task-specific primary literature, and a snapshot of the Hugging Face TRL documentation together with the papers cited by that documentation. Internet access was disabled during the campaign. The literature and candidate datasets were evidence available to the researcher, not a prescribed training recipe; the researcher remained responsible for selecting, filtering, combining, or rejecting them. The experimental contrast therefore conditions on a common information set and does not measure open-web search ability.

The researcher model was Kimi-K3, requested with maximum reasoning effort, a maximum response length of 32,768 tokens, and a 3,600-second turn timeout. The student checkpoints were `Qwen/Qwen3-1.7B-Base` at revision `ea980cb0a6c2ae4b936e82123acc929f1cec04c1` and `Qwen/Qwen3-4B-Base` at revision `906bfd4b4dc7f14ee4320094d8b41684abff8539`.

B.2 Local teacher

Where a campaign included a local teacher, it used `Qwen/Qwen3.8-27B` at revision `1d4bf0f2ff6012fd82039f2fa52739d0dd7c60c0`. The vLLM serving configuration used bfloat16 weights, an 8,192-token model context, `gpu_memory_utilization=0.92`, prefix caching, and the Triton GDN prefill backend. Teacher inference consumed the campaign's assigned GPU allocation, and teacher-generated examples were prohibited from using evaluation items. Sampling and filtering for data generation were selected by the researcher and recorded with each generated dataset; no single teacher sampling configuration was fixed across campaigns.

B.3 Evaluation targets and decoding

All Qwen evaluations enabled thinking and used BOS/PAD token 151643 and EOS tokens `[151645, 151643]`. "Greedy" denotes temperature 0, top-$p = 1$, and top-$k = -1$. Table 4 summarizes the seven local targets.

Table 4: Public evaluation sources, splits, and per-target decoding.

Target	Public source and split	Rows	Decoding and scoring
AIME 2025	`math-ai/aime25`, test	30	T=0.6, top-p=0.95, top-k=20; 16,000 new tokens.
ArenaHard-Writing	`CohereLabs/m-ArenaHard-v2.0`, multilingual writing subset	250	T=0.7, top-p=0.8, top-k=20, rep. penalty 1.05; 16,384 new tokens; `gpt-5-mini` judge at medium reasoning effort.
BFCL	Berkeley Function Calling Leaderboard, `exec_simple`	100	Greedy; 16,000 new tokens; Hermes tool parser with executable/AST matching.
GPQA-Main	`Idavidrein/gpqa`, `gpqa_main`	448	T=0.6, top-p=0.95, top-k=20; 16,000 new tokens; answer choices shuffled.
GSM8K	`openai/gsm8k`, `main/test`	1,319	Greedy, rep. penalty 1.0; 4,000 new tokens; fixed eight-shot prompt.
HealthBench-Easy	PostTrainBench 245-example HealthBench-Easy subset	245	T=0.6, top-p=0.95, top-k=20; 16,384 new tokens; `gpt-5-mini` judge at T=0 with shuffle seed 42.
HumanEval	`openai/openai_humaneval`, test	164	Greedy, rep. penalty 1.0; 4,000 new tokens; local execution sandbox.

B.4 Candidate datasets supplied to the researcher

Table 5 lists the public origins or, where no public artifact exists, the names of frozen prepared derivatives that were supplied per target. Dataset availability did not imply that an agent used a given dataset in its final training mixture. The staged dataset trees were content-hashed; most upstream Hugging Face revision SHAs were not retained. Public identifiers therefore describe provenance but do not by themselves reproduce the exact staged bytes. The released supplement includes the staged-tree SHA-256 manifest.

Table 5: Candidate datasets supplied to the researcher per target.

Target	Candidate datasets
AIME 2025	`EleutherAI/hendrycks_math`; `AI-MO/NuminaMath-CoT`; `nvidia/OpenMathInstruct-2`; `open-r1/OpenR1-Math-220k`; `nvidia/AceReason-Math`; `simplescaling/s1K-1.1`; `GAIR/LIMO`; `nvidia/OpenMathReasoning`; `AI-MO/NuminaMath-TIR`; `zwhe99/DeepMath-103K`.
ArenaHard-Writing	Prepared Tülu-3 instruction-constraint subset; `HuggingFaceH4/no_robots`; NVIDIA HelpSteer3 preference, feedback, and edit subsets; `zai-org/LongWriter-6k`; `OpenAssistant/oasst2`; `allenai/tulu-3-pref-personas-instruction-following`; `SAA-Lab/LitBench-Train`.
BFCL	`NousResearch/hermes-function-calling-v1` (single- and multi-turn subsets); `argilla/Synth-APIGen-v0.1`; `Team-ACE/ToolACE`; `allenporter/assist-llm-function-calling`; `nvidia/When2Call`.
GPQA-Main	`allenai/ai2_arc` (ARC-Challenge train); `allenai/openbookqa` (additional train); `allenai/sciq` (train); prepared WildSci, MegaScience, Nemotron-RL science, Nemotron-SFT science/StackExchange, Nemotron-SFT science/vendor, and SCP-116K science derivatives; `allenai/qasc`; `xw27/scibench`; `TianHongZXY/CHIMERA`.
GSM8K	`openai/gsm8k` (main train); `EleutherAI/hendrycks_math`; `TIGER-Lab/MathInstruct`; `meta-math/MetaMathQA`; `nvidia/OpenMathInstruct-2`; `open-r1/OpenR1-Math-220k`; `microsoft/orca-math-word-problems-200k`; `chilleD/SVAMP`.
HealthBench-Easy	Frozen local MedRedQA training preparation; frozen local medical-case-reasoning preparation; `GBaker/MedQA-USMLE-4-options`; `Itaykhealth/K-QA`; `har1/MTS_Dialogue-Clinical_Note`; `datajuicer/RealMedConv`.
HumanEval	`bigcode/self-oss-instruct-sc2-exec-filter-50k`; `OpenCoder-LLM/opc-sft-stage2`; `m-a-p/CodeFeedback-Filtered-Instruction`; `nuprl/stack-dedup-python-testgen-starcoder-filter-v2`.

B.5 Task-specific primary literature

Table 6 lists the task-specific and benchmark-defining sources supplied per target. Compact identifiers of the form `YYMM.NNNNN` resolve at arxiv.org/abs/<identifier> unless a venue or web URL is stated. Some papers appear in more than one pack.

Table 6: Task-specific and benchmark-defining literature supplied per target.

Target	Supplied sources
AIME 2025	Qwen3 technical report (2505.09388) and the official Qwen3 blog; s1 (2501.19393); LIMO (2502.03387); OpenMathReasoning (2504.16891); AceReason (2505.16400); OpenThoughts (2506.04178); DeepSeek-R1 (2501.12948); Distilling Step-by-Step (2305.02301); TailSFT (2608.25756).
ArenaHard-Writing	Qwen3 report/blog; Arena-Hard/BenchBuilder (2406.11939); LongWriter (2408.07055); Suri (Findings of EMNLP 2024, paper 94); WritingBench (2503.05244); WritingPreferenceBench (2510.14616); HelpSteer3 feedback/edits (2503.04378); HelpSteer3 preferences (2505.11475); Art or Artifice (2309.14556); Tülu 3 (2411.15124); DPO (2305.18290); Anchored Preference Optimization (2408.06266); Magpie (2406.08464); DEITA (2312.15685).
BFCL	Qwen3 report/blog; Berkeley Function Calling Leaderboard (ICML 2025); APIGen (2406.18518); ToolACE (2409.00920); Hammer (2410.04587); When2Call (NAACL 2025 long paper 174); ActionStudio (2503.22673); Tülu 3 (2411.15124); Self-Instruct (2212.10560); Magpie (2406.08464).
GPQA-Main	Qwen3 report/blog; GPQA (2311.12022); WildSci (2601.05567); MegaScience (2507.16812); SciInstruct (2401.07950); ChemPile (2505.12534); CHIMERA (2603.00889); DEITA (2312.15685); Distilling Step-by-Step (2305.02301); Tülu 3 (2411.15124).
GSM8K	Qwen3 report/blog; Training Verifiers to Solve Math Word Problems (2110.14168); MetaMath (2309.12284); MAmmoTH (2309.05653); OpenMathInstruct 2 (2410.01560); process- versus outcome-based feedback (2211.14275); DeepSeekMath (2402.03300); ReST-EM (2312.06585); small-model SFT empirical guide (2412.13337); Open-R1 update 3; Tülu 3 (2411.15124); DeepSeek-R1 (2501.12948); TailSFT (2608.25756).
HealthBench-Easy	Qwen3 report/blog; HealthBench (2505.08775); K-QA (2401.14493); MedCaseReasoning (2505.11733); Med-PaLM (2212.13138); Med-PaLM 2 (2305.09617); Magpie (2406.08464); Tülu 3 (2411.15124).
HumanEval	Qwen3 report/blog; Evaluating Large Language Models Trained on Code (2107.03374); SelfCodeAlign (2410.24198); OpenCoder (2411.04905); RLTF (2307.04349); CodeRL (2207.01780); Tülu 3 (2411.15124); DEITA (2312.15685); Distilling Step-by-Step (2305.02301).

In addition, every campaign received the TRL 1.12.0 documentation snapshot at commit `59c4a8e104413fa9f4ca1a54eaf2ff93c0f299be` together with the 79 distinct papers linked by that documentation, whose arXiv identifiers are:

```
1606.02647  1707.06347  1910.02054  1910.10683  2104.08691  2106.09685  2211.14275
2305.10425  2305.14314  2305.18290  2306.08543  2306.13649  2307.09288  2309.06657
2309.16240  2310.12036  2312.00886  2312.09244  2401.08417  2402.00856  2402.01306
2402.03300  2402.04792  2402.05369  2402.09353  2402.14740  2403.00409  2403.07691
2403.17031  2404.04656  2404.09656  2404.10830  2404.19733  2405.00675  2405.14734
2405.16436  2405.16681  2405.21046  2406.05882  2406.08414  2406.11827  2408.06266
2409.06411  2411.10442  2411.15124  2501.03262  2501.03884  2501.12948  2502.03373
2503.14476  2503.20783  2503.24290  2504.11442  2505.07291  2505.20686  2505.22312
2506.01939  2506.13585  2507.18071  2507.20673  2508.00180  2508.05629  2508.08221
2510.00977  2510.13786  2510.18855  2510.26788  2511.20347  2512.02556  2601.05242
2601.19897  2601.20802  2602.05261  2602.10693  2603.00729  2604.01193  2605.22765
2606.22600  2606.30406
```

C Researcher-model cost accounting

C.1 Method

We report researcher-model costs for both the published PostTrainBench systems and our campaigns using a single, transparent schedule. For each of our completed campaigns, we sum the token counters over every researcher turn, define fresh input as total input minus cache-read and cache-write tokens, and apply the common price schedule

$$C = \left\{ 3\,N_{\text{fresh}} + 0.3\,N_{\text{cache-read}} + 3\,N_{\text{cache-write}} + 15\,N_{\text{output}} \right\} / 10^6 \quad \text{(USD)}. \tag{2}$$

For external systems, we downloaded the PostTrainBench trajectory index (Rank et al., 2026; PostTrainBench Team, 2026) on 26 September 2026 and, for each system and student size, retained completed records with a non-null score and a positive reported `total_cost_usd`; a zero reported cost is treated as missing accounting rather than free inference. The resulting summaries use different provider prices and cache policies and are therefore descriptive positioning rather than a controlled treatment comparison. Costs exclude GPU rental and judge-API charges throughout.

C.2 Published-system trajectory costs

Table 7 reports arithmetic mean $\pm$ sample standard deviation over the retained trajectories. Codex CLI / GPT-5.6-sol has 14 completed trajectories at each model size but reports zero dollars and is therefore omitted from dollar means; one completed Locus / Opus 5 Qwen3-4B trajectory also reports zero dollars and is omitted, and one Fable Qwen3-4B trajectory is incomplete and is excluded. The large Qwen3-1.7B dispersion for Locus / Opus 5 includes a single reported $503.53 outlier that is retained.

Table 7: Reported researcher-model cost per PostTrainBench trajectory (USD, mean $\pm$ sample standard deviation). n is the number of completed, positive-cost trajectories across all seven tasks.

System / researcher	Qwen3-1.7B	n	Qwen3-4B	n
Locus / Opus 5	79.60 ± 152.64	14	18.21 ± 7.09	13
Fable 5	33.58 ± 50.14	12	18.09 ± 8.37	12
Codex CLI / GPT-5.6-sol	not reported	0	not reported	0
Claude Code / Kimi-K3	126.43 ± 46.33	21	125.88 ± 69.75	21

C.3 Local seven-task cost and token accounting

Table 8 reports our per-campaign researcher-model cost and token accounting under the schedule in Equation 2. Token columns are in millions of tokens. The seven-task rows are an archival coverage audit rather than a controlled cost treatment: AIME (1.7B), GSM8K (both sizes), and HealthBench-Easy (4B) contain heavy repeated status- and polling-turn patterns, which inflate total input and cost despite high cache-read fractions. The parenthetical dispersion is across tasks, not seed uncertainty, and does not include GPU rental or judge-API charges.

C.4 Four-task matched cost

Restricting to the four-task matched pool (AIME 2025, ArenaHard-Writing, GPQA-Main, HealthBench-Easy) used in the local score-summary controls gives a cleaner cost comparison at Qwen3-4B (Table 9).

Table 8: FATHOM researcher-model cost and token totals, one campaign per task student cell. Token columns are millions of tokens; costs use Equation 2.

Student	Task	Cost (USD)	Input	Cache-read	Fresh input	Output
1.7B	AIME 2025	65.85	200.69	199.35	1.334	0.0795
1.7B	ArenaHard-Writing	47.90	120.91	117.70	3.216	0.196
1.7B	BFCL	8.35	18.91	18.44	0.473	0.093
1.7B	GPQA-Main	31.57	70.36	67.61	2.751	0.202
1.7B	GSM8K	174.51	536.34	532.37	3.968	0.193
1.7B	HealthBench-Easy	37.81	85.03	81.68	3.357	0.216
1.7B	HumanEval	55.72	151.37	148.58	2.789	0.186
1.7B mean ± sample SD		60.24 ± 53.66	169.09 ± 172.18	166.53 ± 171.53	2.56 ± 1.22	0.167 ± 0.056
4B	AIME 2025	36.53	94.37	92.86	1.511	0.276
4B	ArenaHard-Writing	105.50	305.91	301.87	4.042	0.188
4B	BFCL	110.23	329.61	326.52	3.091	0.200
4B	GPQA-Main	34.22	80.90	79.33	1.571	0.381
4B	GSM8K	275.51	838.53	831.90	6.623	0.404
4B	HealthBench-Easy	170.54	523.91	520.09	3.814	0.204
4B	HumanEval	51.01	115.80	111.88	3.924	0.378
4B mean ± sample SD		111.93 ± 87.25	327.00 ± 276.94	323.49 ± 275.44	3.511 ± 1.740	0.290 ± 0.096

Table 9: Four-task matched researcher-model cost and token totals at Qwen3-4B, one campaign per task arm cell. Token columns are in millions of tokens; costs use Equation 2.

Arm	n tasks	Mean cost (USD)	Mean input	Mean fresh input	Mean output
Persistent single agent	4	30.42 ± 8.71	69.56	2.68	0.155
FATHOM	4	32.93 ± 3.03	78.82	1.62	0.328

C.5 COST COMPARISON ACROSS POSTTRAINBENCH CAMPAIGN

Across both student-model sizes, FATHOM operates with substantially less fresh input than the Claude Code comparators while still supporting long-horizon agentic exploration through extensive cache reuse. For Kimi-K3, FATHOM uses only 2.56 ± 1.22M fresh input tokens at 1.7B and 3.51 ± 1.74M at 4B, compared with roughly 24M of fresh input for the corresponding Claude Code runs. At the same time, its output footprint is orders of magnitude smaller (0.17 ± 0.06K and 0.29 ± 0.10K tokens), indicating that much of the computational burden is concentrated in reusable context rather than repeated generation. Importantly, FATHOM supports this regime while remaining applicable across model scales and preserving the full agentic post-training workflow.

Table 10: External PTB comparator cost and token accounting, together with GPT-5.6 Sol. Values are run-weighted mean ± sample standard deviation. Input-token columns are in millions of tokens; output is in thousands. A dash indicates that the corresponding cost or token telemetry is unavailable.

Agent	Student	Harness	Cost (USD)	Fresh input (M)	Cache read/write (M)	Output (k)
Kimi-K3	1.7B	Claude Code	126.43 ± 46.33	23.66 ± 10.58	0	96.19 ± 29.32
Opus 5	1.7B	Claude Code	79.60 ± 152.64	- ± -	142.46 ± 298.63 / 0.47 ± 0.25	146.34 ± 92.16
Fable 5	1.7B	Claude Code	33.58 ± 50.14	0.02 ± 0.01	57.89 ± 116.74 / 0.20 ± 0.14	99.64 ± 84.56
Kimi-K3	1.7B	FATHOM	60.24 ± 53.66	2.56 ± 1.22	166.53 ± 171.53/0.00 ± 0.00	0.167 ± 0.056
Kimi-K3	4B	Claude Code	125.88 ± 69.75	24.68 ± 13.85	0	97.92 ± 23.37
Opus 5	4B	Claude Code	18.21 ± 7.09	0.00 ± 0.00	20.94 ± 12.70 / 0.43 ± 0.29	97.50 ± 38.31
Fable 5	4B	Claude Code	18.09 ± 8.37	0.01 ± 0.01	27.06 ± 23.94 / 0.20 ± 0.17	97.68 ± 82.72
Kimi-K3	4B	FATHOM	111.93 ± 87.25	3.511 ± 1.740	323.49 ± 275.44/0.00 ± 0.00	0.290 ± 0.096

Interpretation of historical score runs. The archived score runs include both literature-enabled and no-paper conditions, as well as teacher-enabled and no-teacher conditions. Some displayed Codex FATHOM pairs differ on those access conditions. Their score differences describe the recorded systems and should not all be interpreted as an isolated effect of Planner intervention under identical evidence. The separately archived trajectory analyses also have their own checkpoint and cost records; a trajectory timing is not assigned to a score from another reporting snapshot.

D FROZEN-STATE PLANNING STUDY

D.1 QUESTION AND COMPARISON

This study asks whether the scientific guidance given to the FATHOM Planner improves the next written research program relative to a format-only planning baseline. It is a controlled decision study rather than a training experiment. In every comparison, the research model, decision state, mounted evidence, tools, output files, and declared resource envelope are held fixed. The only manipulated variable is the role instruction.

The **FATHOM Planner** receives the complete planning policy described below. The **format-only baseline** receives only the following requirements: inspect the supplied task, evaluator, data descriptions, papers, baseline measurements, runtime facts, and current state; write one plan and one review-time record; do not train, evaluate, delegate, or generate data; and state one next scientific program together with its evidence, measurement, falsifier, rollback, resource assumptions, and review time. Thus, the baseline is not the single-agent campaign system in Table 2. It controls for the model, evidence, tools, and required output structure while removing FATHOM's substantive research and resource-allocation policy.

The reported comparison contains 15 decision states, two policies, and three independent model calls per state and policy, for 90 plans. A separate resource-envelope experiment is described below.

D.2 PLANNER AND EXECUTOR POLICIES

FATHOM Planner policy. Each invocation starts a fresh Planner session. The Planner reconstructs the campaign from the evidence available at that decision and publishes one remaining-horizon research outlook together with one executable scientific tranche. The outlook preserves useful later options but does not authorize them. The tranche must identify the incumbent and parent lineage, diagnosed failure, causal hypothesis, selected data and supervision signal, trainer-facing representation, objective, dose, resource assumptions, expected behavioral change, comparable measurement, falsifier, rollback meaning, and the observation that would require a new scientific decision. It may include evidence-supported, result-conditioned continuations that test the same causal claim.

The Planner compares materially distinct programs by the fidelity and supply of their supervision, transfer distance from empirical precedent, useful exposure achievable within the horizon, complete latency to a comparable result, and the option value left after that result. It distinguishes the information in a source from the representation expected by the trainer; a mechanically repairable template mismatch is not treated as absence of semantic supervision. Dose is specified as a coupled quantity: for example, unique examples, useful tokens, updates, effective batch, epochs, online reuse, sequence-length distribution, model scale, and learning-rate schedule rather than as row count alone. A treatment below the evidence-supported onset of effect is labelled a feasibility probe and is not credited with the anchor treatment's endpoint claim.

The policy explicitly treats response length and answer closure as part of the supervision distribution when they affect deployment behavior. It requires a complete dependency path through data construction or generation, training, checkpoint publication, and comparable evaluation. Review timing is chosen from that critical path: review observes admitted work but neither defines its dose nor stops a valid active operation. The Planner may inspect evidence and specify scientific bounds, but it may not implement data, reserve GPUs, launch jobs, or change immutable evaluation software.

Corresponding Executor policy. The latest system policy uses one Executor session that persists across plan revisions. The Executor implements only the current tranche; the outlook is context, not execution authority. The Planner owns the scientific hypothesis, signal, objective, dose bounds, comparison, and decision rule. The Executor owns data construction, trainer implementation, concrete values inside authorized bounds, operation ordering, topology, parallelism, throughput and memory measurement, mechanical repair, checkpointing, evaluation, and checkpoint selection.

For every accepted training record, the Executor preserves source and transformation lineage, generator identity when applicable, and the final trainer columns. Raw, intermediate, accepted, and trainer-facing datasets remain distinct; rejects and filtering decisions are retained. Evaluation items

may be inspected for diagnosis but may not enter training or synthetic generation. Before expensive work, the Executor resolves cheap mechanical uncertainty with proportional import, constructor, data, throughput, or memory checks.

After each consequential result, the Executor compares intended and realized data, transforms, parent, objective, dose, resources, proximal behavior, and checkpoint lineage. It classifies failure as mechanical failure, invalid measurement, under-realized treatment, valid negative evidence, or valid positive evidence. A valid evaluation is a continuation decision: authorized same-claim work may continue, whereas contradiction of the scientific premise or exhaustion of authorized continuations requests a fresh Planner review. Training reward or loss alone cannot establish movement in the endpoint. The strongest valid incumbent remains eligible throughout. A new plan does not discard or reparent active work; it governs the next GPU action after a safe operation boundary. This Executor policy describes the complete system, but no Executor was invoked in the frozen-state study, which isolates written planning only.

D.3 Decision states and information boundary

Eight states are openings with no post-training outcome: AIME 2025, ArenaHard-Writing, GPQA-Main, and HealthBench-Easy, each paired with Qwen3-1.7B-Base and Qwen3-4B-Base. Seven states are later decision points constructed from observations that were available at the corresponding campaign boundary. The complete decision-relevant content of those junctions is given below. Five states: the 1.7B Arena and Health openings, the 4B AIME and GPQA openings, and the 4B AIME dose junction were available during policy development. The other ten states were withheld from policy development and are also analyzed separately.

Arena 1.7B termination. Decide whether to repair, continue, switch, or roll back. A generic 32k-row SFT mixture consumed 44.1M student tokens in 86 updates; held-out generations frequently terminated immediately or produced malformed tails; measured end-token probability was approximately 0.001; the base incumbent remained available.

HealthBench 1.7B dose. Choose the next funded action. Teacher-generated medical SFT passed a canary, but generation was roughly three times slower than planned; the accepted pool fell from 7,395 to 2,500 rows; a 72-step run at learning rate 10^{-5} still produced rambling answers; accepted data and the incumbent remained available.

AIME 4B dose. Choose the next funded action. Long-reasoning SFT used 22,172 rows, 132.40M student tokens, 339 updates, and 1.202 epochs; the complete score was 6/30, checkpoint 120 scored 5/30, 19/30 final generations reached the 16k-token cap, and 5.28 hours remained.

GPQA 4B closure. Decide whether the next action should change information, representation, or closure. Contract-faithful science SFT used 122.2M tokens and scored 29.02%; accuracy conditional on non-capped answers was 40.4%; all 126 parse failures were 16k-token wait loops rather than wrong answer letters.

HumanEval 4B regression. Choose the next program while protecting the incumbent. The base scored 130/164 (79.27%); three broad-SFT checkpoints scored 68.90%, 73.17%, and 75.00%; an RFT checkpoint scored 78.66% and generic GRPO scored 79.88%; all checkpoints remained available.

GSM8K 4B plateau. Decide whether to continue, switch, compare, or stop. Verified SFT scored 92% on a 300-item diagnostic; rejection-sampling pass@8 was 99.4%; GRPO checkpoints at 60, 120, and 180 updates scored 92.0%, 92.0%, and 91.33% on the same diagnostic.

BFCL 4B residual errors. Choose a bounded action or stop. Schema-faithful SFT solved the output-call format and complete evaluations were 95 to 97%; remaining errors concerned optional or default arguments and strict argument equality; additional epochs and broad preference training were flat or worse.

Each call received the same packet-specific task and evaluator description, student-model and resource facts, mounted data descriptions, benchmark papers, trainer documentation, baseline measurements, and campaign-neutral research library. The following were withheld: outcomes after the frozen boundary, historical oracle plans or recipes, scoring rubrics, policy identities, and outputs from other calls. An opening packet contained no post-training result. A feedback packet contained only the

observations listed above. The resource statement was one NVIDIA H100 80GB and ten active hours. This was a declared planning constraint, not hardware used by the model calls.

D.4 Generation and scoring procedure

Every plan was produced by a fresh Kimi-K3 session through the Fireworks Responses API in the same container image. The study did not set temperature or top-p overrides; those fields therefore followed the provider/client defaults. Both policies had the same read-only inputs, tool authority, and requirement to publish one plan plus a review time. Calls were replaced only for provider failure or truncation before a plan was produced, using the same replacement rule for both policies.

Before generation, two authors independently specified a rubric that allowed multiple scientifically defensible programs rather than requiring the action taken in the historical trajectory. Candidate plans were randomly shuffled within each state and assigned anonymous identifiers. Two fresh Kimi-K3 scorer sessions independently scored every plan without access to policy identity, future outcomes, or other study files. Scorers evaluated each plan independently rather than ranking candidates against one another.

The six domains, each scored from 0 to 4, were:

1. diagnosis and grounding in the supplied evidence;
2. treatment, supervision signal, and data choice;
3. preservation of task-relevant information in the trainer representation;
4. coupling of dose, empirical onset, resources, and latency;
5. measurement, falsifier, rollback, and checkpoint lineage; and
6. horizon management, review timing, switching, and recovery.

A zero contradicts the supplied evidence or omits the domain; two is plausible but incomplete; three is complete and executable; and four compares alternatives, handles uncertainty, and specifies a decision-changing branch. A plan is admissible when it contains no fatal violation, identifies one executable next program, preserves the incumbent and lineage, includes a complete comparable endpoint within the declared envelope, totals at least 18/24, and scores at least 3 in treatment/data, dose/resources, measurement/rollback, and horizon/recovery. Fatal violations include an infeasible critical path, presenting a below-onset probe as an endpoint test, review-induced underdose, updating scientific belief from a mechanical failure, avoidable destruction of supervision information, evaluation contamination, or failure to preserve the incumbent.

For each plan, the two domain-score vectors were averaged. The three plan scores were then averaged within state and policy. Paired differences were computed at the state level; repeated calls estimate generation variation but are not treated as independent states. Confidence intervals use 10,000 state-cluster bootstrap resamples with random seed 20260925. The same procedure was applied separately to opening states, feedback states, and the ten policy-development holdouts.

Domain decomposition. Mean FATHOM/baseline scores were 3.97/3.63 for diagnosis and evidence, 3.93/3.68 for treatment and data, 3.89/3.42 for information representation, 3.81/3.26 for dose, resources, and latency, 3.99/3.86 for measurement and rollback, and 4.00/3.62 for horizon and recovery. The largest differences therefore concern dose and resource reasoning, preserving information across trainer representations, and horizon-aware continuation.

D.5 Resource-envelope intervention

To test whether a written program responds to available compute rather than merely repeating a recipe, we selected four states: the 1.7B Arena and HealthBench openings, the 4B GPQA opening, and the 4B AIME dose junction. For each state, we crossed one versus four declared H100s with ten versus 25 hours. This gives 16 state resource conditions and 96 reported plans (16 conditions, two policies, three calls); the one-H100, ten-hour plans reuse 24 plans from the core study.

Across the 16 conditions, the FATHOM Planner scored 23.67/24 and the format-only baseline scored 21.33/24, a difference of 2.34 points; FATHOM scored higher in all 16 conditions. Its admissibility

was 100.0%, compared with 89.6% for the baseline. Within the FATHOM plans, changing one H100 to four increased median planned useful exposure by an average of 17.1M tokens and reduced the predicted critical path by 0.83 hours. Changing ten hours to 25 increased planned exposure by 30.5M tokens, increased the funded critical path by 1.90 hours, and preserved 8.79 additional hours for evaluation. Plan quality itself changed little with scale: four versus one H100 changed the score by $-0.38/24$, and 25 versus ten hours changed it by $+0.08/24$. These quantities were extracted from written plans and blinded ratings. They are not realized throughput, GPU utilization, or model-score scaling results.

D.6 SCOPE AND LIMITATIONS

The controlled contrast isolates the effect of FATHOM's written Planner guidance relative to a format-only prompt. It does not isolate the value of role separation relative to a skilled single agent, test the persistent Executor, or show that a higher-rated plan improves a benchmark score. Kimi-K3 generated and judged the plans in separate sessions, so shared-model bias is possible. The rubric directly measures properties emphasized by the FATHOM policy, and the resulting scores are close to the upper bound. Five states informed policy development; the ten-state holdout reduces but does not eliminate dependence on the same task families and historical campaigns. No plan was executed, and the H100 counts in the resource study are declared constraints only. Human or different-model adjudication and prospective execution are therefore required before making claims about realized GPU efficiency, online adaptation, final checkpoint quality, or benchmark improvement.

D.7 EXACT ROLE INSTRUCTIONS

For completeness, the following blocks render the complete operative Planner instruction used by the reported FATHOM arm and the corresponding Executor instruction. Generated comments, command strings, and filesystem spellings are normalized for print; no scientific or decision-policy requirement is omitted. The Planner block was active in the frozen-state study. The Executor block specifies the other role in the full system but was not invoked in this Planner-only study.

D.7.1 PLANNER INSTRUCTION

```
---
name: planner
description: Researches evidence and publishes a ranked, horizon-aware
  post-training plan.
disable-model-invocation: true
---

# Planner

Publish one evidence-conditioned, revisable scientific program. Use the
configured campaign lifetime for lookahead, and authorize one current causal
treatment program for execution. You own its hypotheses, experiment design,
resource strategy, and publication. Each revision starts a fresh Planner
session and reconstructs the campaign from durable evidence.

## Authority and output

Open the exact invocation paths first. The invocation is authoritative for the
plan revision, treatment, lifetime, review policy, prior publications, and
output destinations. The outlook spans the useful program supported by the
current evidence and configured lifetime. The executable tranche is
experiment-scoped: it lasts through the first causal decision boundary where
realized evidence can change the scientific claim. It includes the dependent
data, training exposure, intermediate measurements, evidence-supported
same-claim continuations or repairs, and comparable endpoint needed to test one
scientific treatment at an informative dose. A scheduled review observes and
reconstructs this work without defining its dose or turning an intermediate
score into a handoff. Evidence may justify an earlier handoff when it changes
the scientific choice; publication never stops admitted work or shortens the
campaign lifetime.

Publish exactly one fresh, non-empty `PLAN.md` at the supplied destination.
Use free-form prose shaped by evidence rather than a fixed schema. Write
nothing else unless the invocation names a separate review selection; finish
`PLAN.md` before writing that selection.

Specify data and trainer views, but do not implement them, reserve devices,
operate jobs, or edit prior publications.

## Lifetime-aware outlook and executable tranche

The outlook and executable tranche are separate layers of the same
publication. The outlook preserves useful lookahead across the configured
lifetime without authorizing distant work. For a finite campaign that means
the remaining horizon; for an operator-finalized campaign it means the durable
program implied by current evidence. The current tranche is the only
scientific work the Executor may start, and it can contain result-conditioned
steps that remain inside the same causal claim.

For the tranche, make the current hypothesis, parent lineage, intended
intervention, intended signal and exposure, expected observation, measurement,
falsifier, rollback meaning, and stopping or handoff condition clear. Close
choices that current evidence supports, including same-claim continuation
rules whose inputs and consequences are already knowable, and leave later
scientific choices open when they depend on new results. State the resource
and time assumptions needed to realize the treatment and obtain its
decision-grade result.

When a new publication arrives while an operation is active, state its
scientific disposition: how its eventual result bears on the program, parent
eligibility, and remaining evidence needs. The Executor owns whether and how
the operation continues or stops safely. Preserve the incumbent and exact
lineage; never imply that replanning itself stops, discards, or reparents
in-flight work.

Choose sources, mixtures, methods, objectives, dose and scaling strategy,
comparisons, and scientifically material parameter bounds from the available
evidence. Leave concrete implementation values, plan-equivalent ordering,
```

```
topology, and runtime optimization to the Executor when they depend on
mechanical feasibility or execution feedback. The harness does not prescribe
a search procedure, trainer family, stage order, challenger, or recipe.

## Dynamic-review Planner-Executor treatment

Each immutable publication contains a long-range outlook and a shorter
executable tranche. Each Planner publication selects the next review timing
and records it in `REVIEW_CADENCE.toml`. The duration reflects the full
dependency chain to its next decision-grade result, including the data,
training exposure, and comparable measurement needed to test the claim. Select
a boundary that fits that evidence-grounded critical path, so review timing
follows the treatment needed to test the claim rather than defining its dose.
That duration sets the next cumulative review boundary. The review does not
cap treatment dose or operation duration and does not stop admitted work.
Evidence may close the tranche earlier through Planner handoff. At the
selected boundary a fresh Planner reviews the persistent Executor's evidence
and publishes a new tranche. The campaign invocation states whether its
lifetime is finite or operator-finalized.

## Evidence and operating frame

Open the invocation first, then the supplied task descriptor. The invocation
owns the current role, mode, time, paths, and publication facts. The task
descriptor contains the requested outcome and direct paths to every configured
resource. Inspect the underlying resources when their contents matter.

The role's references document is a neutral map of original TRL pages,
configuration dimensions, and locally mounted papers. Open the named
documentation or paper before relying on a summarized mechanism or API.

The runtime-capabilities record is the image-bound account of installed
interfaces and explicit exceptions. The runtime contract is the sole contract
for writable paths, operations, deadlines, services, waiting, checkpoints,
contamination controls, and evaluation.

The API-availability inventory and image-smoke evidence answer different
questions. A method listed as API-available but not image-smoke-tested remains
available; the latter means only that the harness has not recorded the named
one-step image smoke. Missing smoke evidence is not a prohibition. Resolve
material uncertainty proportionally through installed-interface inspection
and, when runtime evidence is needed, an executing-role constructor or canary.

## Online access

Internet research is enabled for this campaign. Use external access only
within this role's scientific authority, keep benchmark material, private
runtime data, and credentials local, and prefer primary sources. Record URLs
and access dates for consequential external claims.

## Research strategy

Build and revise one causal account that connects the measured failure to a
constructible supervision signal, a learning mechanism, a realizable exposure,
an expected behavioral change, and a comparable measurement. Reconstruct the
objective, immutable evaluator, starting checkpoint, baseline or incumbent,
student and teacher resources, hardware, and campaign lifetime from exact
records. Inspect item-level outputs, source schemas and rows, model files,
primary papers, trainer documentation, prior plans, operations, checkpoints,
and trajectories deeply enough to support the load-bearing links in that
account.

Compare serious candidate programs on expected scientific value under the
actual lifetime. The comparison combines the strength and transfer distance of
the evidence, the fidelity and useful supply of the supervision, the exposure
that can be realized, the complete latency to a comparable result, and the
option value of what remains afterward. The closest empirical treatment is a
strong anchor when its model lineage, signal, objective, dose, stage, and
measurement transfer; it is evidence to analyze rather than a recipe to copy.
Where precedent is weak, a compact discriminating experiment earns priority
when its possible outcomes lead to materially different programs.

Study data as the treatment's information channel. Characterize how each
source covers the diagnosed failure, the quality and granularity of its
targets, its representative content, acceptance or yield, generation cost,
trainer-facing form, and useful-token supply. Justify mixtures in terms of
useful exposure and complementary signal rather than row availability. Native
records, teacher-generated supervision, student-generated supervision, and
preference or reward signals carry different information; rank them by the
causal claim they can identify.
```

```
Separate the information carried by a source from its trainer-facing
representation. A mechanically transformable marker, template, or row-shape
mismatch does not erase otherwise strong semantic supervision. Compare
preserving and rendering existing evidence against regenerating it, accounting
for source provenance, generator capability, independent verification,
target-distribution fit, transformation risk, cost, and achievable scale.
Conversely, rendering cannot supply task-relevant information that a generic
source lacks.

When a deterministic transform could change the source ranking, inspect or
construct a representative candidate trainer view and compare the resulting
views rather than their raw encodings. Check joint coverage of task-critical
attributes: separate sources that cover difficulty, language, format, or
domain individually do not establish supervision for their interaction.
Generated or native signal is valuable when it supplies that missing joint
coverage at credible fidelity and scale.

Represent dose as a coupled realization rather than a single large number.
Relate unique examples or prompts, useful supervised or generated tokens,
optimizer updates, effective batch, epochs or online reuse, learning-rate
schedule, sequence-length distribution, and model scale to measured
throughput and the closest empirical anchor. Intermediate checkpoints can map
this response curve. The planned treatment is informative only when its
realized exposure can test the predicted mechanism and still leave time for
its comparable measurement.

Treat response length and closure as part of the supervision distribution
when they affect the measured behavior. Compare the training distribution to
the deployment budget and expected task difficulty, and supply useful
exposure across the depths at which the model must learn to continue,
conclude, and terminate. A maximum-length cutoff alone does not establish that
distribution.

Separate difficulty from response length when they covary in the available
sources. An informative mixture exposes hard examples at multiple closure
depths rather than silently making greater difficulty synonymous with longer
targets. When the deployment budget can bind, include constructible
budget-conditioned closure signal and state the response-distribution change
it is intended to identify.

Estimate the minimum meaningful realization of each candidate from its closest
empirical evidence. A substantially smaller run can answer a feasibility
question, but it is not the same intervention or a test of the reported
mechanism. Rank that probe by the later treatment it enables and its
opportunity cost against a complete treatment that fits.

For a candidate whose support comes from an empirical treatment, compare the
planned exposure to the anchor's observed onset of effect in
mechanism-relevant units such as generated samples, optimizer updates, tokens,
or epochs. State the exposure ratio and the evidence for any scale adjustment.
An exposure below the anchor's observed effect regime is a feasibility probe;
the anchor's endpoint gain does not become that probe's expected treatment
effect. Make this classification before ranking complete programs: a treatment
test reaches an evidence-supported onset regime, while lower exposure remains
a feasibility probe regardless of optimizer activity or the number of
proximal samples. Rank and scope the candidate's causal claim according to
that classification.

For every empirical treatment anchor used to justify a selected candidate,
record an exposure ledger in the scientific decision record: the anchor's
observed onset or dose, the planned exposure in the same units, their ratio,
the evidence for any scale adjustment, and the resulting treatment-test or
feasibility-probe classification. When inspection finds no reported onset,
mark it unknown and scope transfer claims to that uncertainty.

Update the program in proportion to the evidence. Interpret endpoint scores
with their uncertainty and with the realized data, transforms, parent,
objective, dose, proximal behavior, and measurement. A supported mechanism
remains eligible for scaling or same-claim repair while its predicted
behavior is developing and the intended exposure remains unresolved. A
causally different program becomes preferred when decision-grade evidence
contradicts the leading mechanism or a better-supported signal offers greater
expected value. This is a comparison of explanations and interventions, not a
novelty or method-count objective.

Treat a response curve as multidimensional. Saturation along token count,
steps, or one mixture does not establish that a trainer family is exhausted
when signal fidelity, joint coverage, length-difficulty balance, or another
coupled exposure dimension remains unresolved.

Publish the selected causal claim, why it outranks the material alternative,
```

```
the observation that would reverse that ranking, and the lineage and
measurement needed to interpret the result. Treat the configured evaluator as
fixed measurement software and record its settings as reproducibility
metadata.

## Dedicated Planner research

The Planner's comparative role is an evidence-acquisition and decision turn,
not an additional narration of Executor work. Reconstruct the baseline or
incumbent failure and the validity of any realized treatment independently
from exact records. Use item-level behavior and proximal measurements to
separate measurement, signal coverage, representation, optimization,
termination, and capability explanations. Treat the Executor's interpretation
as evidence to verify.

On the initial revision, survey the available signals, trainer mechanisms,
model resources, and closest empirical treatments only far enough to identify
the material contenders, then deepen the evidence that determines their
ranking. The publication should emerge from this comparison with a leading
causal program, an informative coupled dose, a complete critical path, and a
material alternative whose ranking could change under stated evidence.

Value a preliminary stage by the decision it enables, not by how quickly it
produces a score. When its plausible outcomes would leave the same
evidence-supported treatment preferred, compare the complete integrated
program against the staged route and authorize the coupled treatment when it
has greater endpoint value within the lifetime. Keep a stage separate when
its result can genuinely reverse the ranking or protect validity.

On later revisions, compare intended and realized source mixture, transforms,
parent, objective, dose vector, runtime, proximal behavior, and endpoint
measurement before changing the scientific program. Relate each new result to
the predicted response curve and its uncertainty. This decides whether the
next high-value action is same-claim continuation, realization repair,
rollback, or a causally different intervention. The accumulated survey remains
available, so the fresh turn can concentrate on evidence that changes this
choice.

Resource reasoning is part of scientific ranking. Estimate the complete
dependency chain through data construction or generation, trainer realization,
checkpoint publication, and comparable evaluation using measured throughput
when available. Include the cost of a feasibility probe and the remaining
decision opportunity after it. Scientific requirements and meaningful bounds
belong in the plan; constructor details, plan-equivalent topology, retries,
and runtime optimization remain Executor choices.

Read-only helpers may inspect independent sources, papers, trainers,
baselines, and trajectories in parallel. The Planner integrates their evidence
and closes the knowable scientific choices. Research depth is sufficient when
additional inspection is unlikely to change the ranked program. Decision
boundaries belong where an observation changes a scientific choice, with
thresholds grounded in empirical uncertainty and treatment realization.

# Data and supervision design

For each selected source, connect the signal to the diagnosed failure and
specify the evidence needed to establish its schema, representative content,
available scale, coverage, target granularity, quality, and trainer-facing row
shape. Define source mixtures, balancing or curricula when scientifically
material, expected acceptance or yield, generation cost, useful-token supply,
lineage, contamination constraints, and dose units. When the selected method
needs a missing signal, identify an available student or mounted local teacher
and define how generated records will be accepted and what information they
add beyond mounted data. Leave construction, verification, and runtime
operation to the Executor.

## Planner application

Keep `PLAN.md` free-form. It must distinguish non-executable outlook from the
current executable tranche and make the tranche's scientific claim,
measurement, rollback, and handoff authority unambiguous. Cite the exact
records that support load-bearing choices and separate observation,
publication evidence, inference, transfer assumption, and unresolved
uncertainty.

Express tranche closure entirely in terms of realized scientific evidence.
Describe scheduled review solely as observation and republication cadence,
without making it a closure, handoff, or dose condition.

The plan should expose the result of the dedicated research pass, not reproduce
its entire working history. Make the leading path and causally distinct data
```

```
or mechanism alternatives that remain material legible, explain why evidence
ranks them that way, and authorize only the current evidence-producing tranche.
Show how the selected signal, method, coupled exposure, and measurement produce
a valid test, and include supported same-claim continuations whose selection
can be determined from stated intermediate evidence. Keep speculative later
work in the outlook rather than converting it into a rigid recipe for the
Executor.

## Replanning boundary

State what realized evidence closes or falsifies the executable tranche:
completion of its discriminating measurement, contradiction of a scientific
premise, exhaustion of its authorized result-dependent paths, or another
result that makes a new scientific choice necessary. A measurement whose
published rule still selects a same-claim continuation remains inside the
tranche. When a tranche closes before scheduled review, the Executor preserves
the incumbent and requests a Planner handoff. Keep engineering repair,
plan-equivalent resource choices, and scheduled review mechanics out of the
handoff decision.
```

D.7.2 EXECUTOR INSTRUCTION

```
---
name: executor
description: Persists across plan revisions and executes bounded scientific
  work through engineering, analysis, delegation, and supervised runtime
  operations.
disable-model-invocation: true
---

# Executor

Execute the current Planner publication's authorized tranche through
implementation, supervised work, analysis, repair, evaluation, and closeout.
Use its remaining-horizon outlook as context, not execution authority. You own
the realized treatment and the campaign's next action inside the tranche.

## Authority and record

Open the exact invocation paths first. Trust its plan revision, treatment,
Planner hash, campaign lifetime, current review boundary, mode, and writable
paths. Read the immutable `PLAN.md` as one coherent publication containing a
remaining-horizon outlook and one authorized executable tranche. Respect each
closed choice and bounded range in the tranche. Do not start an outlook-only
path.

The Planner owns the scientific choices it closed or bounded for the tranche.
You own data implementation, code, ordering, parallelism, plan-equivalent
realization, runtime measurements, retries, repairs, and checkpoint selection.
When the tranche leaves a choice to new evidence, run the stated measurement
and apply its decision rule.

Maintain `EXECUTION.md` as a concise living decision and evidence index. Link
exact code, data, configurations, operations, logs, checkpoints, and
evaluations rather than duplicating them. Record intended versus realized
treatment, validity, causal interpretation, active work, resource decisions,
incumbent, unresolved evidence, and one next action.

Continue useful tranche-authorized work while it fits the finite campaign
deadline, when one exists, and has expected decision value, including
independent implementation, analysis, safe preparation, and
result-conditioned steps that the tranche explicitly authorizes while long
jobs run. A plan publication does not stop or reparent active jobs. After each
terminal, the newest `PLAN.md` executable tranche is the only source of the
next GPU action; rewrite `EXECUTION.md` Authority to that path and hash and do
not launch the predecessor's next-action list or an outlook-only path. When a
new publication arrives during active work, use runtime evidence and safe
stopping points to decide whether finishing or supervised stopping best
realizes its scientific disposition. Record the choice and retain terminal
output as evidence even when it is not adopted as the next parent. As a finite
deadline shrinks, prefer publishing and validating the incumbent and closing
the record over work that cannot finish before admission closes.

Write only `EXECUTION.md`, the workspace, checkpoints, evaluations, and
records exposed through the supervised runtime. These paths and this Executor
session persist across Planner publications. Inputs and Planner files are
read-only.

End the activation naturally only when the next useful decision depends on
future job evidence or a real revisit/review time and no independent useful
```

```
work remains. The campaign owner will resume this exact chat on each
individual terminal job, committed plan revision, or due timer. Never poll for
controller liveness or compensate with detached shell processes, checkpoint
copies, or harness repair.

Persistence and resumability preserve session, evidence, and checkpoint
lineage; they do not justify idling, repeating settled work, deferring a
supported decision, or consuming unbounded compute. Each activation should
advance or decisively prepare the authorized tranche unless it is waiting for
already submitted evidence.

Request a Planner handoff when realized evidence closes or falsifies the
current executable tranche and the next scientific action is not already
authorized, or when evidence contradicts a premise on which the tranche
depends.

This is an evidence-boundary action, not a last-resort escape. Do not request
it for every checkpoint, for a measurement whose published rule still selects
an authorized same-claim continuation, for scheduled review closure, or for
code, data-path, allocation, OOM, model-server, throughput, or runtime repair.
After requesting handoff, preserve active jobs and the incumbent but start no
new work until the new publication arrives.

For a definitive campaign-wide infrastructure or immutable-input blocker,
stop owned operations and record the literal evidence and impact before
marking the campaign failed.

## Dynamic-review Planner-Executor treatment

Each immutable publication contains a long-range outlook and a shorter
executable tranche. Each Planner publication selects the next review timing
and records it in `REVIEW_CADENCE.toml`. The duration reflects the full
dependency chain to its next decision-grade result, including the data,
training exposure, and comparable measurement needed to test the claim. Select
a boundary that fits that evidence-grounded critical path, so review timing
follows the treatment needed to test the claim rather than defining its dose.
That duration sets the next cumulative review boundary. The review does not
cap treatment dose or operation duration and does not stop admitted work.
Evidence may close the tranche earlier through Planner handoff. At the
selected boundary a fresh Planner reviews the persistent Executor's evidence
and publishes a new tranche. The campaign invocation states whether its
lifetime is finite or operator-finalized.

## Evidence and operating frame

Open the invocation first, then the supplied task descriptor. The invocation
owns the current role, mode, time, paths, and publication facts. The task
descriptor contains the requested outcome and direct paths to every configured
resource. Inspect the underlying resources when their contents matter.

The role's references document is a neutral map of original TRL pages,
configuration dimensions, and locally mounted papers. Open the named
documentation or paper before relying on a summarized mechanism or API.

The runtime-capabilities record is the image-bound account of installed
interfaces and explicit exceptions. The runtime contract is the sole contract
for writable paths, operations, deadlines, services, waiting, checkpoints,
contamination controls, and evaluation.

The API-availability inventory and image-smoke evidence answer different
questions. A method listed as API-available but not image-smoke-tested remains
available; the latter means only that the harness has not recorded the named
one-step image smoke. Missing smoke evidence is not a prohibition. Resolve
material uncertainty proportionally through installed-interface inspection
and, when runtime evidence is needed, an executing-role constructor or canary.

## Online access

Internet research is enabled for this campaign. Use external access only
within this role's scientific authority, keep benchmark material, private
runtime data, and credentials local, and prefer primary sources. Record URLs
and access dates for consequential external claims.

# Data construction

When constructing a trainer-facing dataset, determine the consumer row shape
from the selected installed interface or its documentation.

## Inspect sources

Inspect the actual schema and representative rows of every source selected for
```

```
use. If a required mount is missing or unreadable, record the literal path and
stop dependent work.

Evaluation JSONL, answers, and rubrics may be inspected. They must not enter a
training or synthetic record. Run the pack's exact-normalized target-prompt
overlap check before a derived artifact is used.

## Trainer view

Define and verify the trainer view before training. Keep raw, intermediate,
accepted, and trainer-view files separate.

Every accepted row needs lineage: source id, transform identity, generator or
judge identity when used, and the final trainer columns. Log filters and
retain rejects. Acceptance rate is not the objective. Dose in useful tokens,
unique prompts, or updates---not only row count.

The trainer's expected columns are not evidence that the labels are
trustworthy.

## Missing-signal generation

Use only a generator the campaign actually provides: the student or a mounted
local teacher. Record generator identity and decoding. A transport success is
not a valid row; verify generated records before training and never seed
generation from evaluation items.

A local teacher, if used, follows the campaign teacher contract.

## Implement the trainer view

Implement the specified transforms in ordinary Python: render, filter, mix
with explicit weights, verify, deduplicate, and publish a trainer view. Keep
raw, intermediate, accepted, and trainer-view files separate. Write lineage
with each accepted row.

Serve a local teacher only through the campaign teacher contract and
supervised runtime.

## Implementation and control loop

Inspect and edit the workspace code, data builders, configurations, scripts,
and tests needed for the authorized treatment. Build trainer views through the
data procedure above. Use the runtime-capabilities record for exact installed
interfaces and the runtime contract for all operation, service, checkpoint,
and evaluation mechanics.

Before expensive work, close cheap mechanical uncertainty with imports,
constructors, tiny data samples, or short throughput and memory measurements.
Keep these checks proportional to the unknown. Preserve the scientific
identity of the treatment while repairing its implementation.

Maintain a quantitative dependency, resource, and latency model from realized
execution. For finite campaigns, include remaining time; for
operator-finalized campaigns, include time to decision-changing evidence and
any scheduled owner-timed revisit. After each result or revisit, update whether
useful training, checkpoint publication, comparable evaluation, continuation,
and eventual closeout remain practical. Choose placement, concurrency,
scaling, and equivalent topology from memory fit, dependencies, throughput,
comparability, and feedback latency. Record realized allocation, dose,
throughput, memory, and important departures in `EXECUTION.md`.

## Bounded delegation

At each meaningful work boundary, identify evidence or code tasks that are
genuinely separable. Delegate them when independent model work improves
coverage or the critical path. Give each helper one deliverable, exact read
paths, a read-only scope or non-overlapping writable paths, fixed scientific
invariants, bounded checks, and required evidence.

Batch uniform mechanical work with ordinary processes when no independent
reasoning is needed. Review every helper result and diff, run integration
checks, and retain scientific, resource, checkpoint, publication, and
execution authority in the parent.

## Interpret and continue

For failed work inspect the literal command, logs, metrics, realized progress,
configuration, and artifacts. Classify it as mechanical failure, invalid
measurement, underdose, valid negative, or valid positive before choosing the
next action within current authority. Engineering repair preserves scientific
```

```
identity.

For every consequential result compare intended and realized rows, source
mixture, transforms, code, configuration, parent, dose, resources, proximal
behavior, and checkpoint lineage. Analyze comparability, uncertainty,
trajectories, slices, and confounds. Use intermediate checkpoints as dose
evidence when they are informative, without substituting a proxy or subset
result for comparable endpoint selection. Update the incumbent, remaining-time
model, and one next action in `EXECUTION.md`: continue or scale, repair,
evaluate, roll back, switch within authority, or close out.

Evaluation material remains evaluation-only. Trainer-facing artifacts require
source and transformation lineage plus the pack's overlap check. Use the
unchanged evaluator and decode through the runtime, and preserve the parent,
data, code, configuration, operation, dose, and evaluation lineage of every
checkpoint considered for selection.

## Executor application

On a fresh Planner publication, read its immutable `PLAN.md` and establish
parent, data, and checkpoint lineage before acting. On continuation, confirm
the immutable Planner identity and reconstruct current state from operations,
logs, metrics, checkpoints, evaluations, and `EXECUTION.md`.

Choose concrete values inside authorized bounds from measured evidence.
Choose ordering, parallelism, equivalent topology, retries, repairs,
authorized alternatives, and checkpoints. Never change a tranche's closed
choice or exceed a bounded range silently. A newly published `PLAN.md`
supersedes the prior publication; follow its executable tranche without
waiting for another handoff.

Treat every valid evaluation as a continuation decision. Update the incumbent,
remaining-time model, and realized signal, mixture, parent, dose, and proximal
behavior, then continue or scale, roll back, repair, or select an alternative
when the tranche's decision rule authorizes it. Intermediate checkpoints may
establish scaling behavior, but a proxy or subset result does not replace the
comparable endpoint used for selection. If the result closes or falsifies the
tranche, preserve the best valid incumbent and request a Planner handoff.
Training-metric movement alone does not establish that the evaluation failure
moved.
```

E CAMPAIGN TIME ATTRIBUTION

Table 11 contains the campaign time attribution of Qwen3-4B-Base for Codex and Fathom is present and Table 12 contains the campaign time attribution of Qwen3-1.7B-Base.

Table 11: Qwen3-4B-Base campaign hours. Each task row shows the supplied mean (variance), with category means normalized to a ten-hour campaign. Pooled rows sum the four task means to 40 hours.

Task	Agent	Selected checkpoint work	Completed evaluation	Other GPU work	No GPU operation
AIME 2025	Codex	4.06 ± 0.75	0.73 ± 0.68	1.88 ± 1.27	3.33 ± 1.79
	FATHOM	4.31 ± 1.99	0.84 ± 0.81	0.12 ± 0.60	4.73 ± 0.54
ArenaHard-Writing	Codex	1.12 ± 0.41	1.50 ± 0.44	4.87 ± 0.01	2.51 ± 0.44
	FATHOM	1.27 ± 0.22	1.66 ± 1.33	3.43 ± 1.43	3.64 ± 0.83
GPQA-Main	Codex	2.08 ± 1.09	0.91 ± 0.77	2.87 ± 1.33	4.14 ± 0.37
	FATHOM	3.31 ± 1.35	0.29 ± 0.42	1.27 ± 0.97	5.13 ± 1.30
HealthBench-Easy	Codex	1.41 ± 1.27	1.04 ± 0.18	3.04 ± 0.46	4.51 ± 1.46
	FATHOM	3.11 ± 0.84	1.38 ± 0.53	2.67 ± 1.11	2.84 ± 1.70
Four tasks	Codex	8.67 ± 3.52	4.18 ± 2.07	12.66 ± 3.07	14.49 ± 4.06
	FATHOM	12.00 ± 4.40	4.17 ± 3.09	7.49 ± 4.11	16.34 ± 4.37

Table 12: Qwen3-1.7B-Base campaign hours, with the categories and units of Table 11. Each task row sums to ten hours; the pooled rows sum to 40 hours.

Task	Agent	Selected checkpoint work	Completed evaluation	Other GPU work	No GPU operation
AIME 2025	Codex	2.88 ± 1.32	1.57 ± 0.94	1.19 ± 1.17	4.36 ± 1.10
	FATHOM	5.44 ± 1.67	1.42 ± 1.00	1.32 ± 1.22	1.82 ± 0.41
ArenaHard-Writing	Codex	2.91 ± 1.48	2.14 ± 0.97	3.16 ± 0.96	1.79 ± 0.95
	FATHOM	3.32 ± 0.62	2.46 ± 0.14	2.25 ± 0.09	1.97 ± 0.80
GPQA-Main	Codex	1.67 ± 1.28	0.86 ± 0.63	5.68 ± 1.46	1.79 ± 1.29
	FATHOM	2.71 ± 0.91	1.43 ± 0.73	3.30 ± 0.66	2.56 ± 0.56
HealthBench-Easy	Codex	5.32 ± 1.19	1.99 ± 1.00	1.33 ± 0.40	1.36 ± 0.32
	FATHOM	5.21 ± 0.53	0.76 ± 1.07	2.36 ± 0.38	1.67 ± 0.53
Four tasks	Codex	12.78 ± 5.27	6.56 ± 3.54	11.36 ± 3.99	9.30 ± 3.66
	FATHOM	16.68 ± 3.73	6.07 ± 2.94	9.23 ± 2.35	8.02 ± 2.30

F UNTRAINED STARTING POINTS

The base checkpoints are evaluated under the fixed local task evaluators before post-training. The Qwen3-4B model revision is held fixed across local tasks.

Table 13: Untrained base-model scores (%) under the fixed task evaluators.

Target	1.7B-Base	4B-Base
AIME 2025	3.3	3.3
GPQA-Main	23.4	8.0
HealthBench-Easy	17.3	10.3
ArenaHard-Writing	5.7	30.5
GSM8K	33.6	79.8
HumanEval	0.0	79.3
BFCL exec-simple	0.0	0.0
HealthBench-Hard	0.0	0.0