PATHSEEK: MODEL- AND HARNESS-AGNOSTIC AGENTIC TRAJECTORY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Continued advances in models and agent execution frameworks (harness) have enhanced the capabilities of research agents, yet early-stage trajectory deviations and unsupported intermediate claims can still undermine research quality. We propose PathSeek, which introduces trajectory optimization as a capability-enhancement mechanism that is orthogonal to, and can be combined with, model enhancement and harness improvement, further extending the boundaries of agent capabilities. PathSeek encapsulates one or more consecutive rounds of Reasoning–Action–Observation interaction into a bounded Agent Event, uses executed trajectory prefixes as tree nodes and event-level continuations as search edges, and connects different model–harness combinations through a unified interface. To evaluate unfinished research trajectories, Claim–Evidence Auditing examines the evidence supporting key claims and the dependencies among them, while Middle-Candidate Verification assesses task relevance and overall quality. Together, they provide process-value assessments and corrective feedback at event boundaries to guide branch selection, evidence supplementation, and trajectory adjustment. Across four sets of Deep Research and scientific reasoning benchmarks and multiple combinations of five representative models and three harnesses, PathSeek achieves broad performance improvements and a score of 71.4 on SGI-Bench DeepResearch, demonstrating that our method can more effectively translate existing model capabilities and harnesses into high-quality agent research trajectories.

1 INTRODUCTION

The development of large language models (Singh et al., 2025; Anthropic, 2026; Team et al., 2023; Zeng et al., 2026; Team et al., 2025a; DeepSeek-AI, 2026b) and agent execution frameworks (harnesses) (OpenAI; DeepSeek-AI, 2026a) is driving agents from general-purpose task execution toward more challenging research work. Deep Research has become a representative frontier scenario for evaluating the capability boundaries of current agents. Such tasks require agents not only to perform long-horizon reasoning with tool usage around complex problems, but also to select and revise research trajectories based on continuously acquired evidence, producing conclusions with traceable and verifiable evidence. The core challenge lies not merely in individual reasoning or tool-use capabilities, but in whether agents can coordinate reasoning, action, and verification during long-horizon interactions and stably transform existing capabilities into high-quality research processes.

Existing work has primarily focused on model optimization (Ye et al., 2026; Chen et al., 2025; Li et al., 2026; Bai et al., 2026) and harness optimization (Team et al., 2025b; OpenAI, 2025; Wang et al., 2026c; OpenAI; DeepSeek-AI, 2026a). The former enhances agentic capabilities through model parameter scaling and post-training targeted at reasoning and tool interaction. For example, the Kimi (Team et al., 2025a) series models have continuously increased model parameter scales and achieved performance close to that of closed-source models, while ExoMind (Ye et al., 2026) and Agents-A1 (Bai et al., 2026) leverage various post-training approaches to enable smaller-scale models to surpass larger models in research capabilities. The latter optimizes agent execution processes through mechanisms such as task decomposition, tool orchestration, context management, and multi-agent collaboration. For example, Codex (OpenAI) and DeepSeek Harness (DeepSeek-AI, 2026a) provide general-purpose execution frameworks for long-horizon tasks. These two direc-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

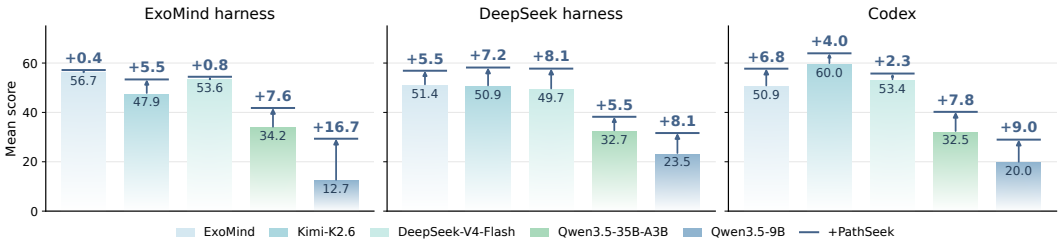

Figure 1: Baseline scores and PathSeek gains across five models and three harnesses. Means equally weight SGI-DR (LLM), PRL (Global), FS-O and FS-R.

tions respectively improve the underlying decision-making capabilities of models and the execution mechanisms of agents, providing stronger capability foundations for Deep Research agents.

However, we observe that even leading model-harness combinations today can exhibit the same failure pattern: erroneous or insufficiently supported intermediate claims continue to be used as premises for scientific reasoning, causing agents to repeatedly extend low-quality research trajectories (Wang et al., 2026a; Kirgis et al., 2026). Based on this observation, this paper further investigates how to transform the evidential support of claims and their dependency relationships into explicit feedback to guide the comparison, revision, and selection of subsequent agent trajectories. To support the joint development of models and harnesses, this paper seeks to address a central question: *Can we construct a trajectory optimization approach on top of existing models and harnesses that is orthogonal to both, enabling existing capabilities to be more effectively transformed into high-quality research processes?*

To this end, we propose PathSeek, an agent enhancement framework that directly optimizes research trajectories and can be layered on top of different model-harness combinations. PathSeek encapsulates consecutive Reasoning-Action-Observation interactions into bounded Agent Events, using executed trajectory prefixes as search nodes and event-level continuations as search edges. Through a unified interface, it enables trajectory optimization across different harnesses, while task planning, tool invocation, and context management within each event remain handled by the original harness. Furthermore, we design two complementary evaluation mechanisms, Claim-Evidence Auditing and Middle-Candidate Verification, which provide evidence-support diagnostics and overall trajectory quality assessment, respectively. Together, they provide fine-grained rewards without waiting for complete rollouts, enabling the search controller to dynamically allocate exploration resources under limited budgets, prevent trajectories that deviate from the objective or lack evidential support from being further expanded, and drive deeper exploration of high-quality research paths.

We evaluate PathSeek on multiple Deep Research and scientific reasoning benchmarks across combinations of different parameter scales, model families, and harnesses. As shown in Figure 1, PathSeek improves the equally weighted average across SGI-DR (LLM), PRL (Global), FS-O, and FS-R for all 15 evaluated model-harness combinations, with occasional regressions on individual tasks. The best-performing combination achieves a score of 71.4 on SGI-Bench DeepResearch (Xu et al., 2025), substantially exceeding the baseline score of 42.5 under the same setting. These results demonstrate PathSeek's applicability across models and frameworks, showing that our method can more effectively transform existing model capabilities and harness capabilities into high-quality agent research trajectories.

2 Related Work

Agents and Harnesses. Advances in the capabilities of Deep Research agents are driven primarily by improvements in models and harnesses. On the model side, ExoMind (Ye et al., 2026), Agents-A1 (Bai et al., 2026), and OpenAI Deep Research (OpenAI, 2025) enhance multi-turn research capabilities through post-training tailored to research tasks, while WebDancer (Wu et al., 2026) leverages research trajectory construction to train autonomous information-seeking policies. On the harness side, Codex (OpenAI) and DeepSeek Harness (DeepSeek-AI, 2026a) provide general-purpose task execution mechanisms, while Tongyi DeepResearch (Team et al., 2025b) and ExoMind

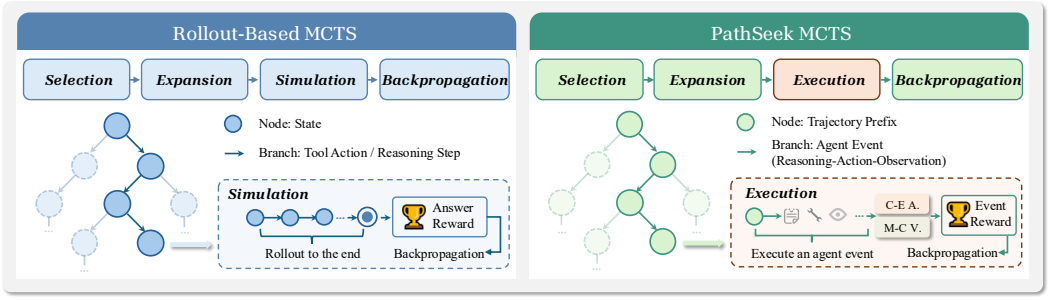

Figure 2: Rollout-based MCTS versus PathSeek. PathSeek executes bounded Agent Events and evaluates trajectory prefixes using Claim-Evidence Auditing and Middle-Candidate Verification, obtaining feedback without requiring complete rollouts.

Harness (Ye et al., 2026) organize research processes around information acquisition, long-horizon reasoning, and scientific verification. OpenEvolve (Sharma, 2025) iteratively improves candidate programs through program generation, automated evaluation, and population evolution. The Agent Lightning (Luo et al., 2025) series further decouples execution from training, enabling different harnesses to integrate with reinforcement learning workflows. PathSeek complements these efforts: it is structured as a reusable, model- and harness-agnostic enhancement layer that preserves the internal mechanisms of existing models and harnesses and uses a unified event execution interface to search over, evaluate, and improve research trajectories at the outer level.

Monte Carlo tree search (MCTS) for Trajectory Search. Monte Carlo tree search (MCTS) has been applied to language model reasoning, tool interaction, and trajectory optimization. RAP (Hao et al., 2023) uses language models as both reasoning policies and world models, while LATS (Zhou et al., 2023) combines value evaluation, reflection, and environmental feedback to organize search. WebPilot (Zhang et al., 2025) performs hierarchical planning and local search for web interaction, while MCTS-RAG (Hu et al., 2025) incorporates dynamic retrieval into the search action space. ReST-MCTS* (Zhang et al., 2024) and rStar-Math (Guan et al., 2025) use process rewards to guide search and construct training trajectories, while PRISM-MCTS (Cheng et al., 2026) reuses successful experiences and reflections on failures through shared memory. PathSeek explores how to extend this approach to a broad range of model–harness combinations. As shown in Figure 2, PathSeek incorporates carefully designed event-level rewards to avoid reliance on multi-step simulations for obtaining rewards in agent settings. It selects and revises research trajectory prefixes generated through actual execution and enables high-quality trajectories obtained through search to further support model learning.

3 Method

3.1 Preliminaries

Problem Formulation. A language model $\boldsymbol{\theta}$ and a harness H jointly define a research agent's interaction policy $\pi_{\boldsymbol{\theta},H}$. Given a question q, a trajectory of T interaction rounds is

$$\tau = (q, o_0, \rho_0, a_0, o_1, \ldots, \rho_{T-1}, a_{T-1}, o_T), \tag{1}$$

where o_0 is optional initial environment information, and ρ_t, a_t, and o_{t+1} denote the recorded reasoning, executed action, and resulting observation at round t. The history h_t is the trajectory prefix preceding round t; the harness constructs the model context from this history, potentially using truncation or summarization. With $\boldsymbol{\theta}$ and H fixed, PathSeek searches a candidate set $\mathbb{T}_B(q; \boldsymbol{\theta}, H)$ under budget B, selects a trajectory $\tau^\star$ using evidence-based evaluation, and returns its answer $y^\star$. The expanded interaction and selection definitions appear in Appendix A.1.

Framework Overview. As shown in Figure 3, an execution adapter connects existing model-harness combinations through bounded Agent Events. Claim-Evidence Auditing and Middle-

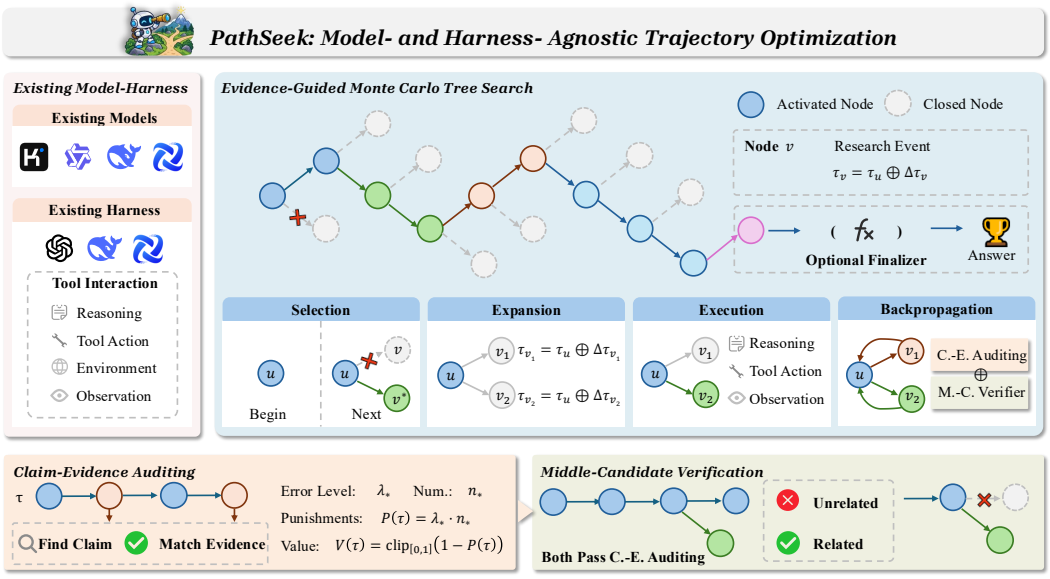

Figure 3: Overview of PathSeek. Existing model–harness combinations execute bounded Agent Events, while Claim–Evidence Auditing and Middle-Candidate Verification provide scores and corrective feedback to guide trajectory search and final answer selection.

Candidate Verification provide scores and corrective feedback for branch selection, expansion, and backpropagation. Selected trajectories can also be filtered offline to construct training supervision.

3.2 PathSeek

3.2.1 Agent Events and Harness Integration

Let node u store the trajectory prefix preceding round t_u. A bounded Agent Event executes one or more consecutive interaction rounds and extends this prefix to a child node v:

$$\begin{aligned} \Delta\tau_v &= \big((\rho_i, a_i, o_{i+1})\big)_{i=t_u}^{t_v-1}, \qquad t_v > t_u, \\ \tau_v &= \tau_u \oplus \Delta\tau_v, \end{aligned} \tag{2}$$

where $\oplus$ appends interaction records in execution order.

A harness adapter exposes the interface $\text{Exec}_{\boldsymbol{\theta},H}(q, s_v, d_v; b_v) \to (\Delta\tau_v, s'_v)$, taking the available execution state s_v, exploration direction d_v, and budget b_v, and returning the executed segment and updated state s'_v. The adapter normalizes harness outputs into reasoning, actions, observations, and completion status, while the original harness manages execution within each event. Search therefore evaluates prefixes grounded in actual tool interactions through a common interface across model-harness combinations. Appendix A.2 gives the expanded execution interface.

3.2.2 Trajectory Evaluation

We evaluate executed trajectory prefixes using two complementary signals.

Claim-Evidence Auditing. We first extract key claims, recording their occurrence and usage locations while distinguishing untested hypotheses from adopted conclusions. We then align each claim with traceable evidence, classifying its support as direct, weak, missing, or conflicting according to the claim type. Finally, we trace claim dependencies to locate the earliest consequential commitment to a flawed claim and the subsequent segments it affects. These diagnostics yield $V_{\text{CE}}(\tau) = \text{clip}_{[0,1]}(1 - P(\tau))$, where P aggregates penalties for erroneous commitments, affected segments, and evidence-support deficiencies (Appendix A.3). The score measures evidence reliabil-

ity rather than answer correctness probability; the audit also identifies evidence gaps and suggests further checks.

Middle-Candidate Verification. At event boundaries, a verifier evaluates the question, current trajectory prefix, and available tool records for task alignment, reasoning consistency, and evidence sufficiency. It returns $V_{\mathrm{MC}}(\tau) \in [0,1]$ and issue descriptions without requiring a completed rollout. The same verifier can assess completed candidates for final selection.

Combining Evaluation Signals. For valid evaluations, we combine the two scores into the verification feedback used by tree search:

$$r_v^{\mathrm{ver}} = S(\tau_v) = (1-\omega)V_{\mathrm{MC}}(\tau_v) + \omega V_{\mathrm{CE}}(\tau_v), \tag{3}$$

where $\omega \in [0,1]$ controls the contribution of claim-level auditing. Numerical feedback updates search values through backpropagation, while structured diagnostics guide evidence supplementation and trajectory revision.

3.2.3 Evidence-Guided Monte Carlo Tree Search

Search Tree and Node Representation. PathSeek adopts Monte Carlo Tree Search to organize candidate research directions. The following describes the trajectory continuation implementation in detail: nodes represent executed research prefixes, edges represent one Agent Event continuation, and each search iteration consists of selection, execution, verification, and backpropagation.

Selection. Each node v stores the visit count N_v, the mean local execution value $\bar{L}_v$, the number of valid verifications M_v, and the mean verification value $\bar{S}_v$. The combined value used for search is:

$$Q_v = \begin{cases} (1-\beta)\bar{L}_v + \beta\bar{S}_v, & M_v > 0, \\ \bar{L}_v, & M_v = 0. \end{cases} \tag{4}$$

For expandable child nodes of parent node u, the following UCT score is used to balance the current value and exploration:

$$\mathrm{UCT}(v \mid u) = \mathbf{1}_{\{N_v>0\}} Q_v + c\, p_v \sqrt{\frac{\log(\max(1, N_u)+1)}{\max(1, N_v)}}. \tag{5}$$

where $\mathbf{1}_{\{N_v>0\}}$ is the indicator function, $c \geq 0$ controls the exploration strength, and $p_v \in [0,1]$ is the candidate prior; autonomous checkpoint branches use a uniform prior. N_v, M_v are non-negative integers, $\bar{L}_v, \bar{S}_v, Q_v \in [0,1]$, and $\beta \in [0,1]$.

Expansion and Execution. In PathSeek, each expansion first retains an autonomous continuation branch, allowing the executor to independently select the next research action. When the verifier identifies specific defects, the controller can additionally generate feedback-guided branches by passing the claims requiring verification, evidence gaps, and corrective suggestions into the execution context. Subsequent actions may include supplementing sources, testing hypotheses, recalculating key quantities, searching for counterexamples, or revising intermediate conclusions. Verification feedback is treated as a suggestion to be validated, allowing the executor to retain or reject these suggestions based on the original evidence. The remaining branches maintain candidate diversity through alternative exploration directions.

Verification and Backpropagation. For a newly executed node, let the local execution signal be r_v^{loc}, and the valid verification signal be r_v^{ver}. When valid, both signals belong to $[0,1]$. Let $\eta_v \in \{0,1\}$ indicate whether the current verification is valid; when $\eta_v = 0$, the verification signal is unavailable. The candidate score of the node itself is:

$$R_v = \begin{cases} (1-\alpha)r_v^{\mathrm{loc}} + \alpha r_v^{\mathrm{ver}}, & \eta_v = 1, \\ r_v^{\mathrm{loc}}, & \eta_v = 0. \end{cases} \tag{6}$$

The controller backpropagates two signals along the path from the root node to the current node, and each ancestor node updates its visit count and local value mean. Only when the verification is valid

are the verification count and verification value mean updated, after which Q is recalculated to influence subsequent computation allocation. The running-average update is derived in Appendix A.4.

$$\bar{S}_u \leftarrow \begin{cases} S(\tau_v), & M_u = 0, \\ \dfrac{M_u \bar{S}_u + S(\tau_v)}{M_u + 1}, & M_u > 0, \end{cases} \qquad (7)$$
$$M_u \leftarrow M_u + 1, \qquad \eta_v = 1, \quad u \in \mathrm{Path}(\mathrm{root}, v).$$

$\alpha \in [0, 1]$ controls the verification weight in the current candidate score, while β controls the verification weight in the search value; the two are configured separately. Structured feedback and numerical backpropagation serve different purposes: numerical values determine which branches receive more search opportunities, while defect records help subsequent execution select specific evidence supplementation and trajectory correction operations. Under the `rank_only` configuration, low verification scores affect value estimation and ranking; only when `hard_prune` is enabled does the controller exclude corresponding branches based on valid verification results, risks, and score thresholds.

The trajectory implementation selects the eligible candidate with the highest node-level R_v, with verification availability handled through equation 6. The subtree value Q_v used for exploration aggregates feedback from subsequent branches and is therefore stored separately from the score of the current candidate answer. If no complete answer is available, implementations supporting finalization may generate a final response based on existing work within the reserved budget.

Constructing Supervision from Optimized Trajectories. In addition to being directly used for inference-time output, the research trajectories generated by PathSeek can also be used for offline supervised data construction. We retain the model messages, tool calls, and tool observations along the selected paths, and filter trajectories with correct answers and effective execution through an independent scoring stage. This process can be written as:

$$\mathbb{D}_{\mathrm{traj}} = \left\{ (q, \tau) \;\middle|\; \begin{array}{r} \mathrm{Valid}(\tau) = 1, \\ J(q, \mathrm{Answer}(\tau)) = 1 \end{array} \right\}, \qquad (8)$$

where Valid and J both return values in $\{0, 1\}$, representing trajectory validity and offline correctness judgment, respectively. J can adopt exact matching, reference-answer-assisted scoring, or other configured judgment rules based on the available information in the training data.

4 Experiments

4.1 Setup

Datasets. We conduct experiments on four sets of evaluation tasks: SGI-Bench DeepResearch (SGI-DR) (Xu et al., 2025), PRL-Bench (PRL) (Miao et al., 2026), FrontierScience-Olympiad (FS-O) (Wang et al., 2026b), and FrontierScience-Research (FS-R) (Wang et al., 2026b), covering multiple domains, including physics, chemistry, biology, engineering, and astronomy. FS-O and FS-R target competition-level scientific reasoning and PhD-level research subtasks, respectively, while SGI-DR and PRL further cover multi-step research reasoning and long-horizon tasks oriented toward real-world scientific research workflows. These benchmarks complement one another in disciplinary background, problem difficulty, and task structure, providing a multi-level evaluation ranging from scientific problem solving to complex research execution.

Models. We select five representative models: Kimi-K2.6 (Moonshot AI, 2026), DeepSeek-V4-Flash (DeepSeek-AI, 2026b), Qwen3.5-35B-A3B (Qwen Team, 2026), Qwen3.5-9B (Qwen Team, 2026), and ExoMind (Ye et al., 2026), covering different providers, parameter scales, and architectural designs, including dense and mixture-of-experts (MoE) models. This set includes both general-purpose models and models that have undergone specialized post-training for scientific research. This selection broadens the experimental coverage across model sources, scales, architectures, and training backgrounds, allowing us to assess the applicability of PathSeek under different underlying capability conditions.

Table 1: Main results across model-harness combinations. Parentheses show score changes from the corresponding baseline, computed before rounding; increases are green and decreases are dark red.

Model	Setting	SGI-DR		PRL	FrontierScience	
		Exact	LLM	Global	Olympiad	Research
ExoMind harness						
ExoMind	Baseline	34.3	57.6	26.4	86.0	57.0
	+PathSeek	35.9 (+1.6)	59.4 (+1.9)	28.9 (+2.5)	82.0 (-4.0)	58.4 (+1.4)
Kimi-K2.6	Baseline	23.6	52.2	28.9	69.0	41.3
	+PathSeek	33.3 (+9.8)	63.2 (+11.0)	33.8 (+4.9)	71.0 (+2.0)	45.4 (+4.1)
DeepSeek-V4-Flash	Baseline	30.5	55.7	30.6	73.0	55.1
	+PathSeek	30.8 (+0.3)	56.3 (+0.6)	30.9 (+0.3)	74.0 (+1.0)	56.6 (+1.4)
Qwen3.5-35B-A3B	Baseline	23.3	48.7	27.0	46.0	15.0
	+PathSeek	25.2 (+1.9)	49.1 (+0.4)	29.6 (+2.6)	52.0 (+6.0)	36.5 (+21.6)
Qwen3.5-9B	Baseline	5.7	13.5	4.9	23.0	9.2
	+PathSeek	25.5 (+19.8)	44.0 (+30.5)	18.8 (+13.9)	35.0 (+12.0)	19.5 (+10.3)
DeepSeek harness						
ExoMind	Baseline	35.2	60.1	33.4	62.0	50.0
	+PathSeek	36.2 (+0.9)	63.5 (+3.5)	35.3 (+1.9)	77.0 (+15.0)	51.7 (+1.8)
Kimi-K2.6	Baseline	26.7	47.2	35.2	69.0	52.4
	+PathSeek	33.3 (+6.6)	60.4 (+13.2)	39.8 (+4.6)	79.0 (+10.0)	53.6 (+1.2)
DeepSeek-V4-Flash	Baseline	21.6	42.5	32.2	70.0	54.0
	+PathSeek	43.1 (+21.5)	71.4 (+28.9)	35.3 (+3.1)	70.0 (+0.0)	54.3 (+0.3)
Qwen3.5-35B-A3B	Baseline	16.0	29.6	25.9	38.0	37.4
	+PathSeek	23.9 (+7.9)	42.8 (+13.2)	29.8 (+3.9)	41.0 (+3.0)	39.3 (+1.9)
Qwen3.5-9B	Baseline	11.0	24.2	17.9	22.0	30.0
	+PathSeek	21.1 (+10.1)	39.9 (+15.7)	19.5 (+1.6)	36.0 (+14.0)	31.1 (+1.1)
Codex						
ExoMind	Baseline	29.3	51.9	31.3	66.0	54.4
	+PathSeek	32.7 (+3.5)	54.4 (+2.5)	48.7 (+17.3)	72.0 (+6.0)	55.8 (+1.4)
Kimi-K2.6	Baseline	32.7	54.1	41.1	86.0	58.7
	+PathSeek	36.8 (+4.1)	61.3 (+7.2)	51.6 (+10.6)	84.0 (-2.0)	58.8 (+0.1)
DeepSeek-V4-Flash	Baseline	27.4	50.9	41.1	69.0	52.6
	+PathSeek	31.8 (+4.4)	53.4 (+2.5)	41.4 (+0.3)	72.0 (+3.0)	56.1 (+3.5)
Qwen3.5-35B-A3B	Baseline	25.5	44.3	25.4	37.0	23.1
	+PathSeek	29.6 (+4.1)	50.3 (+6.0)	25.7 (+0.3)	45.5 (+8.5)	39.4 (+16.3)
Qwen3.5-9B	Baseline	9.4	19.8	11.9	28.0	20.3
	+PathSeek	18.9 (+9.5)	34.0 (+14.2)	18.9 (+7.1)	38.0 (+10.0)	25.0 (+4.7)

Harness. We adopt three harnesses: Codex (OpenAI), DeepSeek Harness (DeepSeek-AI, 2026a), and ExoMind Harness (Ye et al., 2026), covering both general-purpose agent execution frameworks and frameworks tailored to scientific research. The former provide reusable mechanisms for tool interaction and task execution, while the latter organize research processes around scientific information acquisition, evidence processing, and executable verification. We evaluate different model–harness combinations, using the corresponding combinations without PathSeek as baselines. By covering multiple model capability configurations and execution mechanisms, we demonstrate PathSeek's applicability across combinations as a reusable enhancement layer.

Metrics and Settings. For SGI-DR (Xu et al., 2025) and FS-O (Wang et al., 2026b), which primarily involve question answering, we use LLM-as-a-judge results as our primary results to appropriately account for semantically equivalent answers. For PRL (Miao et al., 2026) and FS-R (Wang et al., 2026b), which require rubric-based scoring, we report rubric-based scores assigned by the LLM judge. The mean score for each dataset is computed as a weighted average of its domain-level scores. We use GPT-OSS-120B (OpenAI, 2025) as the benchmark judge throughout.

Table 2: Ablating Claim-Evidence Auditing and Middle-Candidate Verification on SGI-DR and PRL, using Kimi-K2.6 with ExoMind Harness.

Variant	SGI-DR		PRL					
	Exact	LLM	Astro	Cond	HEP	Quant.	Stat	Global
PathSeek	33.3	63.2	36.5	34.4	25.7	35.7	27.7	33.8
w/o claim-evidence auditing	26.1	53.5	28.5	30.8	24.5	29.4	28.6	29.1
w/o middle-candidate verification	28.9	52.5	28.1	29.1	20.6	32.9	23.7	29.0
w/o both (random final selection)	20.4	40.6	25.2	27.0	12.5	20.6	15.4	22.0
ExoMind harness[†]	23.6	52.2	24.5	30.4	22.3	32.4	24.7	28.9

[†]Baseline results for Kimi-K2.6 with the ExoMind harness.

4.2 Main Results

We report results in Table 1 for different model–harness combinations on the above benchmarks and use them as baselines for comparison with the same combinations augmented with PathSeek, demonstrating the applicability of our approach across models and harnesses. PathSeek yields average improvements of **6.2** points with ExoMind Harness, **6.9** points with DeepSeek Harness, and **6.0** points with Codex. Notably, on SGI-Bench DeepResearch, PathSeek substantially raises the best score among the evaluated configurations to **71.4**. This result suggests that trajectory optimization not only delivers broad performance gains but also pushes the empirical performance frontier of existing research agents. Together, these results show that PathSeek can provide additive performance gains through evidence-driven trajectory optimization, without requiring model retraining or harness redesign.

The gains also depend on the model-harness pairing. On SGI-DR, DeepSeek-V4-Flash with DeepSeek Harness achieves the highest LLM-judge score after applying PathSeek, improving from 42.5 to 71.4, compared with 56.3 under ExoMind Harness and 53.4 under Codex. Notably, its baseline score with DeepSeek Harness is lower than with either alternative (55.7 and 50.9, respectively). This reversal suggests that baseline performance alone does not predict how much a combination can benefit from trajectory optimization. The result is consistent with a favorable interaction between this model-harness pairing and PathSeek on SGI-DR, although it does not establish that shared model and harness provenance is the cause.

4.3 Ablation Study and Analysis

4.3.1 Component Ablations

We evaluate the two trajectory-evaluation modules using Kimi-K2.6 with the ExoMind harness (Table 2). Neither module accesses reference answers during inference.

Removing Claim-Evidence Auditing. Removing claim extraction, evidence alignment, dependency tracing, and their scores and feedback reduces the SGI-DR LLM-judge score from 63.2 to 53.5 and exact match from 33.3 to 26.1. PRL scores also decline. This supports the contribution of claim-level evidence diagnostics beyond the retained trajectory-level verifier. Because both scores and corrective feedback are removed, this ablation measures their joint contribution.

Removing Middle-Candidate Verification. Retaining Claim-Evidence Auditing while removing the independent assessment of task relevance and reasoning consistency reduces the SGI-DR LLM-judge score to 52.5 and exact match to 28.9. The PRL results show the same direction. Together, the two ablations support complementary evaluation signals: evidence support for individual claims does not by itself establish that the overall trajectory adequately addresses the task.

Removing Both Modules and Using Random Selection. Without either evaluation module or its feedback, we randomly select expandable branches and the final response from completed valid candidates. The SGI-DR LLM-judge score falls to 40.6, below the original harness's 52.2, while exact match falls to 20.4 versus 23.6. Thus, unguided exploration need not improve performance.

Table 3: Mean tokens, tool calls, trajectory length, and estimated cost with and without PathSeek across harnesses (pooled subset).

Setting	Tokens (k)	Calls	Rounds	Cost ($)
ExoMind Harness				
Baseline	702.9	6.2	6.4	0.402
+PathSeek	1125.1	4.4	5.2	0.341
DeepSeek Harness				
Baseline	214.8	3.8	1.0	0.018
+PathSeek	375.5	14.1	1.8	0.038
Codex				
Baseline	344.0	10.1	1.0	0.097
+PathSeek	282.5	5.2	2.2	0.091

Table 4: Mean tokens, tool calls, trajectory length, and estimated cost with and without PathSeek across harnesses (SGI-DR subset).

Setting	Tokens (k)	Calls	Rounds	Cost ($)
ExoMind Harness				
Baseline	680.7	6.5	6.6	0.258
+PathSeek	1026.9	4.2	5.0	0.250
DeepSeek Harness				
Baseline	241.1	4.3	1.0	0.019
+PathSeek	430.8	16.7	1.8	0.042
Codex				
Baseline	264.3	10.5	1.0	0.091
+PathSeek	335.6	6.0	2.2	0.111

This variant jointly changes branch and final-candidate selection; it does not isolate either stage or establish an equal-compute comparison.

4.3.2 Resource Usage and Cost

PathSeek allocates computation to evidence checks and bounded prefix exploration, using GPT-OSS-120B (OpenAI, 2025) as the verifier throughout. Unlike generating additional complete rollouts, evaluating partial trajectories does not require every explored branch to reach a final answer.

Tables 3 and 4 report per-question means. Tokens are in thousands; calls and rounds describe the recorded output trajectory, not the entire search tree. Cost combines rollout and verifier usage at their respective input/output rates, using a historical-composition approximation; it excludes external tool and hosting fees. Dashes denote unavailable comparable round counts. Full pricing, estimation assumptions, and trajectory analysis appear in Appendix B.2.

Resource implications. Estimated pooled verifier expenditure is approximately $0.090, $0.007, and $0.003 per question for ExoMind, DeepSeek, and Codex, respectively. Total estimated costs change from $0.402 to $0.341, $0.018 to $0.038, and $0.097 to $0.091. Although verifier expenditure is small in absolute terms, DeepSeek's total cost increase remains substantial proportionally. Dataset coverage also matters: Codex's estimated cost rises from $0.091 to $0.111 on SGI-DR.

For ExoMind, retained trajectories become shorter (6.4 to 5.2 rounds in the pooled subset), despite increased total token usage. Output-path length and search cost therefore capture different aspects of execution. Appendix B.2 discusses the harness-specific patterns and coverage limitations. Appendix C.1 further evaluates the selected trajectories as supervision for model training.

5 Conclusion

This work introduces PathSeek, a model- and harness-agnostic trajectory optimization framework for deep research agents. As model capabilities and agent execution frameworks (harness) continue to advance, existing systems still struggle to consistently produce high-quality research trajectories due to early path deviations, unsupported intermediate claims, and accumulated errors. PathSeek complements model enhancement and harness improvement by optimizing research trajectories on top of existing model–harness combinations, enabling more effective transformation of existing agent capabilities into reliable research processes.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and assistance with parts of the code implementation. We also used generative AI tools to assist in understanding domain-specific knowledge encountered in deep research trajectories, including explaining unfamiliar concepts, terminology, and background information. We have not used generative AI tools for developing research ideas, designing the methodology, conducting experiments, analyzing experimental results, or drawing scientific conclusions, and other required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to generate the PathSeek icon shown in Figure 3. We have reviewed all AI-assisted work. Specifically, all AI-assisted manuscript revisions were carefully checked and revised by the authors, AI-assisted code was verified and tested for correctness within our experimental pipeline, and the AI-generated icon was manually reviewed and refined to ensure that it does not introduce plagiarism or copyright concerns. We take responsibility for the final content of this work, including all text, claims, experimental results, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies trajectory optimization methods for autonomous research agents. Our experiments use publicly available benchmarks and do not involve human subjects, private data, or personally identifiable information. While PathSeek improves evidence-grounded research trajectory selection, language-model-based agents may still produce incorrect or unsupported information. Therefore, outputs generated by such systems should be independently verified before being used in high-impact domains. PathSeek is intended as a tool for improving agent reliability rather than a replacement for human judgment. We follow responsible research practices by documenting our methodology and evaluation settings. The authors take responsibility for all reported methods, results, analyses, and conclusions.

REPRODUCIBILITY STATEMENT

To facilitate reproduction, we provide the implementation of PathSeek, including the trajectory search controller, trajectory evaluation modules, harness adapters, and experimental configurations as supplementary materials. The main paper and appendix provide complete descriptions of the trajectory representation, Claim–Evidence Auditing, Trajectory-Level Verification, Monte Carlo tree search procedure, and training pipeline.

REFERENCES

Anthropic. Claude opus 4.8 system card, 2026. Accessed: 2026-06-19.

Lei Bai, Zongsheng Cao, Yang Chen, Zhiyao Cui, Shangheng Du, Yue Fan, Shiyang Feng, Zijie Guo, Haonan He, Liang He, Xiaohan He, Shuyue Hu, Yusong Hu, Songtao Huang, Yichen Jiang, Hao Li, Xin Li, Dahua Lin, Weihao Lin, Fenghua Ling, Dongrui Liu, Zhuo Liu, Runmin Ma, Chunjiang Mu, Haoyang Peng, Tianshuo Peng, Jinxin Shi, Luohe Shi, Boyuan Sun, Zelin Tan, Shengji Tang, Qianyi Wang, Yiming Wu, Yi Xie, Xiangchao Yan, Jingqi Ye, Peng Ye, Fangchen Yu, Jiakang Yuan, Bihao Zhan, Bo Zhang, Chen Zhang, Shufei Zhang, Shuaiyu Zhang, Wenlong Zhang, Yiqun Zhang, Junpeng Zhao, Zhijie Zhong, Bowen Zhou, and Yuhao Zhou. Scaling the horizon, not the parameters: Reaching trillion-parameter performance with a 35b agent. *arXiv preprint arXiv:2606.30616*, 2026.

Jiacheng Chen, Qianjia Cheng, Fangchen Yu, Haiyuan Wan, Yuchen Zhang, Shenghe Zheng, Junchi Yao, Qingyang Zhang, Haonan He, Yun Luo, et al. P1: Mastering physics olympiads with reinforcement learning. *arXiv preprint arXiv:2511.13612*, 2025.

Siyuan Cheng, Bozhong Tian, YanChao Hao, and Zheng Wei. Prism-mcts: Learning from reasoning trajectories with metacognitive reflection. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16392–16402, 2026.

DeepSeek-AI. Deepseek harness: Everything is a plugin. `https://github.com/deepseek-ai/deepseek-harness`, 2026a.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026b.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 8154–8173, 2023.

Yunhai Hu, Yilun Zhao, Chen Zhao, and Arman Cohan. Mcts-rag: Enhancing retrieval-augmented generation with monte carlo tree search. *arXiv preprint arXiv:2503.20757*, 2025.

Peter Kirgis, Sayash Kapoor, Andrew Schwartz, Stephan Rabanser, David Africa, Konstantinos Voudouris, Viet Nguyen, Toby Pilditch, Magda Dubois, Harry Coppock, et al. Can ai agents conduct open-ended ai research? early evidence from two case studies. *arXiv preprint arXiv:2607.27191*, 2026.

Yafu Li, Runzhe Zhan, Haoran Zhang, Shunkai Zhang, Yizhuo Li, Zhilin Wang, Jiacheng Chen, Futing Wang, Xuyang Hu, Yuchen Fan, et al. Achieving gold-medal-level olympiad reasoning via simple and unified scaling. *arXiv preprint arXiv:2605.13301*, 2026.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K Qiu, and Yuqing Yang. Agent lightning: Train any ai agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*, 2025.

Tingjia Miao, Wenkai Jin, Muhua Zhang, Jinxin Tan, Yuelin Hu, Tu Guo, Jiejun Zhang, Yuhan Wang, Wenbo Li, Yinuo Gao, et al. Prl-bench: A comprehensive benchmark evaluating llms' capabilities in frontier physics research. *arXiv preprint arXiv:2604.15411*, 2026.

Moonshot AI. Kimi K2.6: Advancing Open-Source Coding. `https://www.kimi.com/blog/kimi-k2-6`, 2026. Accessed: 2026-06-19.

OpenAI. Codex CLI. `https://github.com/openai/codex`.

OpenAI. Introducing deep research. `https://cdn.openai.com/API/docs/deep_research_blog.pdf`, February 2025. OpenAI blog, February 2, 2025.

OpenAI. gpt-oss-120b gpt-oss-20b model card, 2025. URL `https://arxiv.org/abs/2508.10925`.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Asankhaya Sharma. Openevolve: an open-source evolutionary coding agent, 2025. URL `https://github.com/algorithmicsuperintelligence/openevolve`.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025a.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025b.

Jiaming Wang, Ziteng Feng, Jiangtao Wu, Ruihao Li, Qianqian Xie, Yuxiang Ren, He Zhu, Xueming Han, Fanyu Meng, Junlan Feng, et al. Where do deep-research agents go wrong? span-level error localization in agent trajectories. *arXiv preprint arXiv:2606.02060*, 2026a.

Miles Wang, Robi Lin, Kat Hu, Joy Jiao, Neil Chowdhury, Ethan Chang, and Tejal Patwardhan. Frontierscience: Evaluating ai's ability to perform expert-level scientific tasks. *arXiv preprint arXiv:2601.21165*, 2026b.

Yibo Wang, Lei Wang, Yue Deng, Keming Wu, Yao Xiao, Huanjin Yao, Liwei Kang, Hai Ye, Yongcheng Jing, and Lidong Bing. Deepresearcheval: An automated framework for deep research task construction and agentic evaluation. *arXiv preprint arXiv:2601.09688*, 2026c.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhenglin Wang, Zhengwei Tao, Ding-Chu Zhang, Zekun Xi, Robert Tang, et al. Webdancer: Towards autonomous information seeking agency. *Advances in Neural Information Processing Systems*, 38:120957–120985, 2026.

Wanghan Xu, Yuhao Zhou, Yifan Zhou, Qinglong Cao, Shuo Li, Jia Bu, Bo Liu, Yixin Chen, Xuming He, Xiangyu Zhao, et al. Probing scientific general intelligence of llms with scientist-aligned workflows. *arXiv preprint arXiv:2512.16969*, 2025.

Peng Ye, Zhuo Liu, Jingqi Ye, Fangchen Yu, Shengji Tang, Yichen Jiang, Haonan He, Zongsheng Cao, Tao Chen, Bo Zhang, et al. Exomind: Democratizing scientific intelligence via extended-mind-inspired agentic system. 2026.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772, 2024.

Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23378–23386, 2025.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.

A MATHEMATICAL DETAILS OF PATHSEEK

This appendix expands the interaction, event, and scoring definitions used in the main text and derives the running-average update used in search. The evaluation scores are search surrogates, not calibrated correctness probabilities or guarantees of optimality.

A.1 INTERACTION HISTORY AND TRAJECTORY SELECTION

For a fixed model parameter vector $\boldsymbol{\theta}$ and harness H, the history preceding round t is

$$h_t = \big(q, o_0, (\rho_i, a_i, o_{i+1})_{i=0}^{t-1}\big). \tag{9}$$

Here q is the question, o_0 is optional initial environment information, and ρ_i, a_i, and o_{i+1} are the recorded reasoning, executed action, and returned observation, respectively. The harness constructs the model context from this history, potentially through truncation or summarization. The joint interaction policy admits the conditional factorization

$$\begin{aligned} \pi_{\boldsymbol{\theta},H}(\rho_t, a_t \mid h_t) &= \pi_{\boldsymbol{\theta},H}(\rho_t \mid h_t)\, \pi_{\boldsymbol{\theta},H}(a_t \mid h_t, \rho_t), \\ h_{t+1} &= (h_t, \rho_t, a_t, o_{t+1}). \end{aligned} \tag{10}$$

This factorization describes the interaction distribution; it does not require two separate model calls. After T rounds, the accumulated record is the trajectory defined in the main text.

Let $\mathbb{T}_B(q; \boldsymbol{\theta}, H)$ denote the candidate trajectories explored under budget B, and let $\mathcal{C}_B$ be the corresponding nodes with eligible complete answers. The trajectory implementation selects using the node-level candidate score R_v:

$$\begin{aligned} v^\star &\in \arg\max_{v \in \mathcal{C}_B} R_v, \\ \tau^\star &= \tau_{v^\star}, \qquad y^\star = \mathrm{Answer}(\tau^\star). \end{aligned} \tag{11}$$

This selection is over the candidates actually explored, rather than all possible trajectories. It applies when $\mathcal{C}_B \neq$; otherwise, an implementation may use the reserved finalization budget as described in the main text. The underlying model and harness remain fixed during this search.

A.2 EVENT-LEVEL EXECUTION AND PREFIX EXTENSION

An edge from parent u to child v records an executed continuation between interaction boundaries t_u and t_v. In expanded form,

$$\begin{aligned} \Delta\tau_v &= \big((\rho_i, a_i, o_{i+1})\big)_{i=t_u}^{t_v-1}, \qquad t_v > t_u, \\ \tau_v &= \tau_u \oplus \Delta\tau_v. \end{aligned} \tag{12}$$

The operator $\oplus$ concatenates records in execution order. Each event therefore contains one or more interaction rounds; an event boundary need not coincide with the end of the task.

The adapter exposes the execution interface

$$\mathrm{Exec}_{\boldsymbol{\theta},H}(q, s_v, d_v; b_v) \longrightarrow (\Delta\tau_v, s_v'), \tag{13}$$

where s_v is the available state before executing the child continuation, d_v is its exploration direction, b_v is its execution budget, and s_v' is the resulting state. The adapter converts these inputs into native harness requests and normalizes the returned reasoning, actions, observations, and completion status. In implementations supporting continuation, the state may include the trajectory prefix, tool observations, unresolved issues, and recoverable workspace files. The segment $\Delta\tau_v$ comes from actual execution rather than a simulated tool response.

A.3 CLAIM-EVIDENCE SCORING DETAILS

Claim-Evidence Auditing aggregates its structured diagnostics into a normalized score:

$$\begin{aligned} V_{\mathrm{CE}}(\tau) &= \mathrm{clip}_{[0,1]}\big(1 - P(\tau)\big), \\ P(\tau) &= \lambda_{\mathrm{first}} z_{\mathrm{first}} + \lambda_{\mathrm{err}}(n_{\mathrm{err}} - 1)^+ \\ &\quad + \lambda_{\mathrm{weak}} n_{\mathrm{weak}} + \lambda_{\mathrm{miss}} n_{\mathrm{miss}} + \lambda_{\mathrm{conf}} n_{\mathrm{conf}}, \end{aligned} \tag{14}$$

where $P(\tau)$ is the total penalty, $z_{\text{first}} \in \{0, 1\}$ indicates whether an erroneous commitment has been localized, and n_{err} counts erroneous trajectory segments. The quantities n_{weak}, n_{miss}, and n_{conf} count claims with weak, missing, and conflicting support, respectively. We use $a^+ = \max(0, a)$ and $\text{clip}_{[0,1]}(a) = \min(1, \max(0, a))$. All penalty coefficients are non-negative and control the contribution of each defect to the trajectory score.

Claim records retain their occurrence and usage locations and distinguish untested hypotheses from adopted conclusions. Evidence records preserve traceable sources and original text, with support assessed according to claim type. Dependency tracing identifies the earliest consequential commitment to a flawed claim and the subsequent segments affected by it. The resulting score compares evidence-support and consistency risks, rather than estimating the probability of a correct final answer. Evidence gaps, affected claims, and suggested checks are returned alongside the numerical score.

A.4 From Single-Trajectory Scores to Node Statistics

A valid evaluation of the current trajectory gives the single feedback value

$$r_v^{\text{ver}} = S(\tau_v) = (1 - \omega)V_{\text{MC}}(\tau_v) + \omega V_{\text{CE}}(\tau_v). \tag{15}$$

For $\omega \in [0, 1]$, both component scores lie in $[0, 1]$, so their convex combination also lies in $[0, 1]$.

To distinguish this single score from the accumulated node statistic, let $\mathcal{I}_u$ index the valid evaluation events whose feedback has been backpropagated to node u. Write $\tau^{(k)}$ for the evaluated trajectory in event k. These indices refer to evaluation events, not necessarily distinct descendant nodes. Then

$$M_u = |\mathcal{I}_u|, \qquad \bar{S}_u = \frac{1}{M_u} \sum_{k \in \mathcal{I}_u} S(\tau^{(k)}), \qquad M_u > 0. \tag{16}$$

For a new valid evaluation of τ_v, every node u on the root-to-v path receives the same single feedback value. When $M_u > 0$, adding that event gives

$$\begin{aligned} \bar{S}_u^{\text{new}} &= \frac{\sum_{k \in \mathcal{I}_u} S(\tau^{(k)}) + S(\tau_v)}{M_u + 1} \\ &= \frac{M_u \bar{S}_u + S(\tau_v)}{M_u + 1}, \\ M_u^{\text{new}} &= M_u + 1. \end{aligned} \tag{17}$$

When $M_u = 0$, the first valid event initializes $\bar{S}_u = S(\tau_v)$ and $M_u = 1$. An invalid evaluation leaves both statistics unchanged. Thus the recursive update in the main text is exactly the arithmetic mean of valid feedback received by the node; it does not average invalid evaluations as zero scores.

The distinction also explains the roles of R_v and Q_v. The candidate score R_v combines the current node's local signal r_v^{loc} and its single verification score $S(\tau_v)$, with weight α. The search value Q_v combines the accumulated local mean $\bar{L}_v$ and verification mean $\bar{S}_v$, with weight β. When no valid verification is available, each uses the local-only fallback defined in the main text. Consequently, Q_v can reflect evaluations of subsequent branches, whereas R_v scores the current candidate itself. Bounded local signals, valid verification scores, and weights $\alpha, \beta \in [0, 1]$ imply $R_v, Q_v \in [0, 1]$; this boundedness does not imply that either score equals answer correctness probability.

B Detail Results

B.1 Complete Main Results

We report SGI-DR, PRL, and FrontierScience results, including domain breakdowns, for the 15 model-harness combinations. Within each harness, the baseline models are listed first, followed by the same models with PathSeek. Scores are rounded to one decimal place using round-half-up. Aggregate formulas retain the source domain weights: $(16, 30, 10, 36, 8)/100$ for PRL, $(10, 40, 50)/100$ for FrontierScience-Olympiad, and equal weights across the three FrontierScience-Research domains. Explicitly entered aggregate scores are preserved.

Table 5: Complete results across model-harness combinations. FS-O: FrontierScience-Olympiad; FS-R: FrontierScience-Research. Astro: astrophysics; Cond: condensed matter; HEP: high-energy physics; Quant: quantum physics; Stat: statistical physics. Bio, Chem, and Phys denote biology, chemistry, and physics.

Model	SGI-DR		PRL						FS-O				FS-R			
	Exact	LLM	Astro	Cond	HEP	Quant	Stat	Global	Bio	Chem	Phys	Total	Bio	Chem	Phys	Total
ExoMind Harness																
ExoMind	34.3	57.6	44.6	17.0	39.0	23.6	21.5	26.4	60.0	95.0	84.0	86.0	50.8	70.3	49.9	57.0
Kimi-K2.6	23.6	52.2	24.5	30.4	22.3	32.4	24.7	28.9	50.0	77.5	66.0	69.0	40.3	40.6	43.1	41.3
DeepSeek-V4-Flash	30.5	55.7	28.2	31.8	36.1	33.8	9.8	30.6	60.0	80.0	70.0	73.0	45.2	58.9	61.3	55.1
Qwen3.5-35B-A3B	23.3	48.7	28.8	31.5	24.7	24.8	19.2	27.0	30.0	52.5	44.0	46.0	15.7	13.4	15.8	15.0
Qwen3.5-9B	5.7	13.5	4.3	1.8	1.3	8.9	4.5	4.9	0.0	30.0	22.0	23.0	12.1	10.5	5.1	9.2
ExoMind Harness ***+PathSeek***																
ExoMind	35.9	59.4	25.7	25.4	36.7	29.2	36.8	28.9	80.0	87.5	78.0	82.0	60.1	61.6	53.4	58.4
Kimi-K2.6	33.3	63.2	36.5	34.4	25.7	35.7	27.7	33.8	70.0	80.0	64.0	71.0	48.9	47.0	40.4	45.4
DeepSeek-V4-Flash	30.8	56.3	23.4	36.5	28.5	30.1	31.9	30.9	70.0	85.0	66.0	74.0	53.8	68.6	47.3	56.6
Qwen3.5-35B-A3B	25.2	49.1	29.2	36.5	23.0	29.0	15.1	29.6	30.0	60.0	50.0	52.0	35.3	35.9	38.4	36.5
Qwen3.5-9B	25.5	44.0	19.3	24.6	13.0	17.6	9.3	18.8	20.0	52.5	24.0	35.0	27.8	8.5	22.2	19.5
DeepSeek Harness																
ExoMind	35.2	60.1	38.9	37.5	23.7	32.9	21.5	33.4	20.0	60.0	72.0	62.0	51.4	53.0	45.5	50.0
Kimi-K2.6	26.7	47.2	24.5	43.3	37.5	36.1	19.0	35.2	30.0	87.5	62.0	69.0	54.2	55.7	47.3	52.4
DeepSeek-V4-Flash	21.6	42.5	27.9	35.6	29.2	30.7	39.3	32.2	50.0	85.0	62.0	70.0	55.6	61.3	45.1	54.0
Qwen3.5-35B-A3B	16.0	29.6	27.2	27.0	20.6	28.5	14.8	25.9	0.0	50.0	36.0	38.0	40.9	39.8	31.5	37.4
Qwen3.5-9B	11.0	24.2	19.8	16.1	8.4	23.7	6.8	17.9	10.0	37.5	12.0	22.0	31.8	36.3	21.9	30.0
DeepSeek Harness ***+PathSeek***																
ExoMind	36.2	63.5	46.2	38.3	32.5	31.5	22.5	35.3	50.0	100.0	64.0	77.0	50.1	55.3	49.9	51.7
Kimi-K2.6	33.3	60.4	31.0	43.6	31.4	42.7	40.2	39.8	40.0	90.0	78.0	79.0	48.4	57.0	55.3	53.6
DeepSeek-V4-Flash	43.1	71.4	32.2	36.3	36.7	35.3	35.8	35.3	50.0	75.0	70.0	70.0	56.3	68.3	38.3	54.3
Qwen3.5-35B-A3B	23.9	42.8	25.1	34.3	16.6	33.7	21.5	29.8	20.0	47.5	40.0	41.0	39.9	48.2	29.9	39.3
Qwen3.5-9B	21.1	39.9	22.9	17.2	19.5	23.1	6.0	19.5	10.0	40.0	38.0	36.0	35.9	41.0	16.4	31.1
Codex																
ExoMind	29.3	51.9	28.0	28.6	22.9	34.0	47.1	31.3	50.0	82.5	56.0	66.0	56.4	66.4	40.4	54.4
Kimi-K2.6	32.7	54.1	31.2	47.7	18.8	46.6	39.3	41.1	70.0	95.0	82.0	86.0	59.8	62.6	53.7	58.7
DeepSeek-V4-Flash	27.4	50.9	37.9	45.0	39.8	41.4	33.9	41.1	30.0	75.0	72.0	69.0	49.4	60.4	47.6	52.6
Qwen3.5-35B-A3B	25.5	44.3	20.9	29.5	10.0	29.1	21.4	25.4	20.0	37.5	40.0	37.0	25.4	25.7	17.9	23.1
Qwen3.5-9B	9.4	19.8	8.7	14.4	8.9	12.1	11.0	11.9	0.0	30.0	32.0	28.0	26.6	19.5	15.0	20.3
Codex ***+PathSeek***																
ExoMind	32.7	54.4	41.8	61.8	32.5	47.0	41.0	48.7	30.0	82.5	72.0	72.0	55.9	72.1	39.4	55.8
Kimi-K2.6	36.8	61.3	32.3	53.1	87.5	54.4	27.9	51.6	50.0	100.0	78.0	84.0	57.3	64.8	54.3	58.8
DeepSeek-V4-Flash	31.8	53.4	36.4	45.0	47.0	40.7	34.6	41.4	50.0	82.5	68.0	72.0	56.0	62.3	50.1	56.1
Qwen3.5-35B-A3B	29.6	50.3	23.3	28.7	18.3	28.3	16.6	25.7	30.0	52.5	43.0	45.5	44.7	47.4	26.2	39.4
Qwen3.5-9B	18.9	34.0	20.4	16.7	34.6	16.0	17.4	18.9	50.0	40.0	34.0	38.0	24.1	26.6	24.3	25.0

B.2 Resource Accounting and Trajectory Analysis

This appendix provides the accounting assumptions and detailed interpretation of Tables 3 and 4.

Measurement and cost estimation. Tables 3 and 4 report mean token usage, output-trajectory tool calls, and interaction rounds where comparable counters are available. We estimate rollout and

verifier costs separately, using input/output prices in dollars per million tokens: Kimi-K2.6 (Moonshot AI, 2026) (0.958575/3.9818), DeepSeek-V4-Flash (DeepSeek-AI, 2026b) (0.22121/0.663629), Qwen3.5-35B-A3B (Qwen Team, 2026) and ExoMind (Ye et al., 2026) (0.058989/0.471914), Qwen3.5-9B (Qwen Team, 2026) (0.073737/0.589892), and GPT-OSS-120B (OpenAI, 2025) (0.117978/0.471914). Cached input tokens are charged at the supplied input price; no cache discount is assumed.

Token means do not include a complete model-by-role input/output breakdown. We therefore approximate that breakdown using historical usage composition, estimated from up to five usable small logs per model-dataset-setting combination, or compact per-question summaries for the ExoMind baseline. Model and dataset token weights are derived from historical question pairs with token records on both sides. We rescale this composition to each mean token total:

$$\widehat{C} = \frac{\overline{T}}{10^6} \sum_{m,g,t} \widehat{f}_{m,g,t}\, p_{m,t}, \qquad \sum_{m,g,t} \widehat{f}_{m,g,t} = 1. \tag{18}$$

Here, $\overline{T}$ is the mean total token count, m identifies the serving model, g distinguishes rollout-side and verifier usage, t distinguishes input and output tokens, $\widehat{f}$ is the estimated token fraction, and p is the corresponding price. Auditing is priced as GPT-OSS-120B (OpenAI, 2025) usage; generation, repair, checkpoint, and finalization calls are assigned the rollout model's rate. Input/output components are normalized when their sum differs from the logged total. The single cost column reports the sum of the two cost components, excluding external tool fees and hosting costs.

Tokens are in thousands. Calls and rounds describe the recorded output trajectory. Cost combines rollout and GPT-OSS-120B (OpenAI, 2025) verification under the historical-composition approximation described in the text; external tool fees are excluded. A dash denotes unavailable round counts.

Search usage and output-trajectory length. Table 3 shows that additional search computation does not necessarily produce a longer output trajectory. For ExoMind Harness, mean total token usage increases from 702.9k to 1,125.1k, while output-trajectory tool calls decrease from 6.2 to 4.4 and interaction rounds decrease from 6.4 to 5.2. Codex also exhibits fewer output-trajectory calls, from 10.1 to 5.2. DeepSeek Harness shows a different pattern: token usage increases from 214.8k to 375.5k and output-trajectory calls increase from 3.8 to 14.1. These differences indicate that retained-path length is harness-dependent. Fewer retained calls should not be interpreted as fewer calls across the full search tree, and additional calls alone do not establish more useful evidence acquisition.

Verifier overhead and total estimated cost. Under the stated prices and usage-composition approximation, pooled verifier expenditure is approximately \$0.090 per question for ExoMind Harness, \$0.007 for DeepSeek Harness, and \$0.003 for Codex. This isolates the modest absolute expense of the verifier from the potentially larger cost of generating and executing research continuations. Total estimated cost changes from \$0.402 to \$0.341 for ExoMind, from \$0.018 to \$0.038 for DeepSeek, and from \$0.097 to \$0.091 for Codex. The ExoMind estimate illustrates why a token increase need not translate proportionally into a dollar-cost increase when the model and request-type composition changes. However, DeepSeek's increase of approximately two cents per question remains substantial in relative terms. The estimates therefore support discussing absolute verification overhead separately from total cost, rather than claiming uniformly negligible cost growth.

Dataset-specific behavior. Table 4 restricts the analysis to SGI-DR. ExoMind again produces shorter retained trajectories, with tool calls decreasing from 6.5 to 4.2 and rounds from 6.6 to 5.0; estimated cost changes from \$0.258 to \$0.250. DeepSeek's output-trajectory calls increase from 4.3 to 16.7 and estimated cost from \$0.019 to \$0.042. For Codex, token usage increases from 264.3k to 335.6k and estimated cost from \$0.091 to \$0.111, despite a decrease in the pooled cost estimate. This change in direction highlights the influence of dataset composition and available question coverage. Although the SGI-DR resource records cover all five models for each harness, metric-specific coverage differs, so the table does not establish a controlled cross-harness efficiency ranking.

Table 6: Evaluation of supervised fine-tuning on trajectories generated with and without PathSeek. PathSeek trajectories generally provide stronger supervision signals and lead to improved performance on the overall evaluation.

Variant	SGI-DR		PRL					
	Exact	LLM	Astro	Cond	HEP	Quant.	Stat	Global
Qwen3.5-35B-A3B	23.3	48.7	28.8	31.5	24.7	24.8	19.2	27.0
ExoMind Harness (100 steps)	26.1	46.9	28.8	27.9	19.1	27.1	20.4	26.2
PathSeek (100 steps)	29.9	51.6	26.2	35.9	22.5	35.7	17.4	31.4
ExoMind Harness (170 steps)	26.7	48.7	30.4	31.5	23.7	28.1	14.4	27.9
PathSeek (170 steps)	27.0	49.4	22.8	33.4	29.9	30.8	17.8	29.2

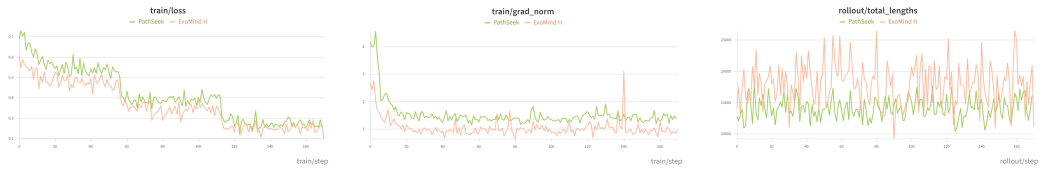

Figure 4: From left to right: training loss, gradient norm, and rollout length.

C Training Details

C.1 Learning from Optimized Trajectories

In addition to improving inference-time trajectory selection, PathSeek can also provide high-quality research trajectories for model training. To evaluate whether optimized trajectories can be effectively transferred into model capabilities, we conduct an additional supervised fine-tuning (SFT) experiment.

Specifically, we collect trajectories for the same set of initial questions using ExoMind Harness with and without PathSeek. These trajectories are then used to fine-tune Qwen3.5-35B-A3B under identical training configurations, including the same optimization procedure and training settings. The only difference lies in the source of supervision trajectories.

Table 6 reports the evaluation results after SFT. Compared with trajectories collected from the original harness, trajectories generated with PathSeek generally provide stronger supervision signals and lead to improved performance on the overall evaluation. These results indicate that PathSeek not only improves inference-time research trajectory selection, but also discovers higher-quality research behaviors that can be transferred into the model through supervised learning.

As shown in Figure 4, both runs show decreasing training loss, while PathSeek exhibits generally shorter rollouts and fewer large length excursions.

D Case Studies Across Harnesses

We present one paired example per harness, keeping the model and research question fixed within each pair. Gray boxes summarize the task, red boxes show the original harness output, and blue boxes show the PathSeek search, executed evidence, and selected result. **Presentation and omissions.** The boxes are abridged, editorially organized summaries of saved records, not complete transcripts or verbatim quotations. The marker . . . , followed by an italic explanation, explicitly identifies omitted or compressed material, such as task background, intermediate discussion, code, retrieval text, repeated checks, and detailed audit fields. It is an editorial marker, not an agent output, a timeout, or evidence of an unexecuted branch. A marker may describe compression within a displayed passage as well as a gap between passages. Numerical results and execution states are retained as reported. The ExoMind and DeepSeek baseline boxes summarize the evaluated response rather than a reconstructed full interaction history; the Codex baseline additionally uses its saved

execution log. The explanatory paragraphs outside the boxes are our analysis. Executed branches, unexecuted candidates, and alternative calculations within a single branch are distinguished explicitly. Benchmark scores are reported separately from internal verification scores. These selected examples illustrate mechanisms and failure modes; they are not equal-compute ablations.

D.1 ExoMind

Task: constrained energy minimization

. . . Task background and full subtask wording are condensed below.

Model: Kimi-K2.6. **Benchmark:** PRL, `prlbench_091`.
Given $E = (0, 1, 2, 4, 7, 11)$, $M = (4, 9, 16, 25, 36, 49)$, and $m = 2$, minimize $\sum_i p_i E_i$ subject to $\sum_i p_i = 1$ and $\sum_i \sqrt{p_i/M_i} \geq 1/\sqrt{m}$. The task requests feasibility decisions, the stationarity root and its residual, the optimal energy and distribution, constraint residuals, a support-size bound, an exponential-form test, and coarse-graining results.

Original ExoMind harness: partial answer, 11/100

. . . Only the evaluated response is summarized below; the full baseline interaction is not reproduced.

Recorded output progression. The baseline discusses the failure of an exponential fit, concludes that the answer to that subtask is "no," and starts the coarse-graining calculation:

$$a_1 = \frac{1}{4} + \frac{1}{9}, \qquad a_2 = \frac{1}{16} + \frac{1}{25} + \frac{1}{36} + \frac{1}{49},$$
$$\mu = \frac{a_1\{2(a_1 + a_2) - 1\}}{a_2}.$$

It then states that it needs to compute μ and the corresponding probability, without providing those final numerical values in the evaluated output.
Stopping condition: the available model-call budget is exhausted.
Evaluator outcome: incorrect, with a rubric score of **11/100**. The judge credits the exponential-form decision but reports that the other required results are missing. This record shows incomplete delivery; it does not establish that all preceding calculations were incorrect.

PathSeek: executed calculations and alternative completed candidates

. . . Selected path and candidate summaries follow; the full 64-node search and all messages are not reproduced.

1. Establish a numerical solution. The selected trajectory contains ten Python tool calls. It checks feasibility, then defines the task-specific sums

$$S_1(\nu) = \sum_{i=1}^{6} \frac{1}{M_i(E_i - \nu)}, \qquad S_2(\nu) = \sum_{i=1}^{6} \frac{1}{M_i(E_i - \nu)^2}.$$

A root calculation solves $S_2(\nu)/m = S_1(\nu)^2$. The observed output is

$$\nu^* = -10.662067374051205, \qquad \left|S_2(\nu^*)/m - S_1(\nu^*)^2\right| \simeq 2.17 \times 10^{-19}.$$

. . . Full Python programs, intermediate arrays, and additional numerical output are omitted.

2. Test the result by a second numerical method. The trajectory constructs $p_i = [M_i(E_i - \nu^*)^2 S_2(\nu^*)]^{-1}$, evaluates the energy, and runs direct constrained optimization with SLSQP. Tool outputs give

$$E_{\min} = 0.9243015292, \qquad |E_{\text{constructed}} - E_{\text{SLSQP}}| \simeq 4.44 \times 10^{-16}.$$

The largest coordinate difference between the two probability vectors is approximately 1.17×10^{-9}. Normalization and compression residuals are printed as zero at the displayed precision.

. . . Repeated verification calls and intervening discussion are omitted; only the final candidate comparison is shown below.

3. Compare alternative finalizations. The search summary contains 64 nodes and 44 iterations. Three completed candidates share the recorded prefix ending at `tt32`. Let P denote the abridged shared prefix `tt0` $\rightarrow$ `...` $\rightarrow$ `tt32`; the omitted node identifiers belong to that common prefix. The saved candidate identifiers establish the following structure:

$P \to F_3$	Finalize the depth-11 prefix; **selected**.
$P \to \texttt{tt38} \to F_2$	Finalize after one further continuation.
$P \to \texttt{tt38} \to \texttt{tt42} \to F_1$	Finalize after two further continuations.

These are competing finalizations of related prefixes, not three independent scientific hypotheses. Their recorded selection scores are:

Candidate	Depth	Verifier	Claim-evidence	Selection
F_3 (selected)	11	55.0	67.0	55.00
F_2	12	30.0	67.0	42.95
F_1	13	20.0	67.0	36.45

Scores use 100-point scales; "Selection" is the recorded score after the final gating checks. All three candidates report the same principal numerical answers. Their different internal scores therefore do not establish that the unselected answers are mathematically wrong. The run stops at its time limit; ground-truth use is disabled in the online selection configuration.

. . . The full derivations and candidate responses are omitted; the selected final results are summarized below.

4. Selected final results (condensed).

- Feasible set nonempty: yes; optimum above E_1: yes.
- $\nu^* = -10.6620673741$; $E_{\min} = 0.9243015292$.
- Optimal distribution:

$$p = (0.5904482822,\ 0.2193466966,\ 0.1046635538,\\ 0.0499567606,\ 0.0239077731,\ 0.0116769337).$$

- Support size: 6; harmonic-mean lower bound: 5.8616984748, giving integer bound 6.
- Exact exponential form: no.
- Coarse-graining: $\mu = 0.0565420580$, probability 0.8328904313.

The saved final response includes the required solution and answer sections and matches the evaluated output. **Benchmark outcome: correct, 100/100.**

The useful change is the completion and consolidation of checkable research work: the selected trajectory connects a stationarity calculation to constraint checks, an independent optimization, and all requested final results. The baseline's evaluated output stops before that complete deliverable. The branch comparison also shows that extra continuation is not automatically useful: PathSeek selects the shorter of three related completed candidates. However, the candidates share the main numerical results, and the internal verifier flags include noisy assessments; this case supports reproducible answer completion, not perfect branch discrimination or an equal-budget causal claim.

D.2 DeepSeek Harness

Task: infer dry biomass from a volumetric regression

. . . The original question is paraphrased below; surrounding instructions are omitted.

Model: DeepSeek-V4-Flash. **Benchmark:** SGI-DR, `SGI_DeepResearch_0059`.
Estimate dry grassland biomass for a volume of $2.0\,\mathrm{m}^3$ and report two decimal places in $\mathrm{kg/m}^2$. The question text does not supply the numerical regression coefficients.

Original DeepSeek harness: unsupported substitution

. . . Only the evaluated response is summarized below; the full baseline interaction is not reproduced.

Evaluated reasoning. The response notices that the coefficients are missing, then assumes that the biomass estimate equals the supplied volume. The resulting calculation effectively uses an identity mapping rather than an established biomass model.
Final output: 2.00.
Benchmark outcome: exact match = 0; LLM judge = 0.

PathSeek: evidence acquisition, checkpoint alternatives, and numerical answer

. . . Selected observations and checkpoint records follow; retrieval text, code, and intermediate discussion are condensed.

1. Retrieve the relation missing from the question. The retained audit input records retrieval of *Estimation of forage biomass and vegetation cover in grasslands using UAV imagery*. Tool observations contain OCR of the dry-biomass figure, including the relation

$$\text{Volume} = 17.446 \times \text{dry biomass} + 0.3707.$$

Thus, the slope and intercept have an observed source rather than being supplied by an unsupported assumption.

. . . Search results, article text, figure-processing commands, and repeated OCR output are omitted.

2. Invert the relation and execute the arithmetic. The trajectory identifies volume as the dependent variable in the retrieved regression and calculates

$$\text{dry biomass} = \frac{2.0 - 0.3707}{17.446} \simeq 0.09339 \quad \Longrightarrow \quad 0.09\,\mathrm{kg/m^2}.$$

The final audit aligns the regression with figure-derived observations and the inversion with an executed computation, assigning direct support to both claims.

. . . Intermediate reasoning and full claim-evidence records are omitted; the checkpoint structure is summarized below.

3. Distinguish executed continuations from proposed alternatives. The checkpoint tree contains the following forks:

Parent	Executed continuation	Other generated candidate
Root	`n001`	`n002`: alternative derivation/check
`n001`	`n003`	`n004`: audit-guided continuation
`n003`	`n005`	`n006`: audit-guided continuation

The selected path is

$$\text{Root} \rightarrow \texttt{n001} \rightarrow \texttt{n003} \rightarrow \boxed{\texttt{n005}}.$$

Only `n001`, `n003`, and `n005` are executed. Their audit scores are respectively 0.41, 0.35, and 0.59. The 0.09 answer is already present in the first executed segment; the later continuations preserve it. Nodes `n002`, `n004`, and `n006` are generated but not executed before the rollout budget is exhausted, so no results are attributed to them.

. . . Repeated answer text in the executed continuations and detailed audit fields are omitted.

4. Retain the answer with the remaining audit concern visible. The last audit still flags missing support for the output units. Verification is configured for ranking, so its pruning recommendation does not remove this path. The saved selection chooses `n005` with audit score 0.59.
Final output: $0.09\,\mathrm{kg/m^2}$.
Benchmark outcome: exact match $= 1$; LLM judge $= 1$.

The numerical success follows from replacing the baseline's unsupported identity mapping with retrieved regression coefficients and then solving for the requested variable. The trajectory exposes the dependency chain from source relation to arithmetic to answer. It does not show an audit-guided branch correcting a wrong intermediate answer: the successful calculation precedes the final audit, and the proposed audit-guided alternatives are unexecuted. The residual unit flag further shows why benchmark acceptance and complete evidence support should be reported separately.

D.3 CODEX

Task: maximum near-field scanning interval

. . . Device background and full question wording are condensed below.

Model: Kimi-K2.6. **Benchmark:** SGI-DR, `SGI_DeepResearch_0108`.
At $f = 2\,\mathrm{GHz}$ and $c = 3 \times 10^8\,\mathrm{m/s}$, the wavelength is $\lambda = 150\,\mathrm{mm}$. The scanner-to-device distance is $4\,\mathrm{mm}$, with the probe $1\,\mathrm{mm}$ below the scanner surface. Determine the probe-to-device distance h and

evaluate

$$\Delta_{\max}(h) = \frac{\lambda}{2\sqrt{1 + (\lambda/h)^2}},$$

using the requested intermediate rounding and reporting the final interval to two decimal places.

Original Codex: correct arithmetic for the wrong input interpretation

. . . Execution-log excerpt summarized below; full code and surrounding messages are omitted.

Executed calculation. The baseline sets $h = 0.003\,\mathrm{m}$, evaluates $\lambda/h = 50$, and rounds the square-root term to 50.0100. Its tool output gives

$$\Delta_{\max} = \frac{150}{2 \times 50.0100} \simeq 1.499700\,\mathrm{mm}.$$

Final output: $1.50\,\mathrm{mm}$.
Benchmark outcome: exact match $= 0$; LLM judge $= 0$. Successful execution verifies this substitution, not the geometric choice of h.

PathSeek: two search branches and two calculations within the selected branch

. . . Branch records and selected calculations follow; full tool calls and intermediate discussion are omitted.

1. Expand the root into two executed branches.

- **Root → node 1: selected.** The continuation investigates probe placement, compares candidate distances, and produces a complete numerical answer. Its audit score is 1.0.
- **Root → node 2: excluded.** This alternative is executed but terminates with the recorded status `non_workspace_local_access`; it is pruned as an invalid path. It has no accepted final answer. This exclusion is an execution-policy outcome, not a claim-evidence rejection of a scientific conclusion.

. . . Full branch transcripts and search responses, including unsuccessful retrieval attempts, are omitted.

2. Compare two input interpretations inside node 1. The selected branch performs searches about probe placement and runs Python calculations for both distances:

Distance choice	Computed interval (mm)	Rounded answer (mm)
$h = 3\,\mathrm{mm}$	1.499700	1.50
$h = 5\,\mathrm{mm}$	2.498612	2.50

These are two calculations within one executed branch, not two additional MCTS nodes. The trajectory then adopts $h = 4 + 1 = 5\,\mathrm{mm}$, matching the benchmark's geometric interpretation.

. . . Further retrieval attempts and intermediate discussion are omitted; the final arithmetic check is retained below.

3. Recompute with the requested rounding. A final decimal-arithmetic tool call records

$$\lambda/h = 30,$$
$$\sqrt{1 + (\lambda/h)^2} = \sqrt{901} \simeq 30.0167,$$
$$\Delta_{\max} = \frac{150}{60.0334} \simeq 2.4986091076\,\mathrm{mm}.$$

The audit reports no weak, missing, or conflicting support risks. Final selection retains node 1 under the recorded reason that it has a complete answer without new descendant evidence.
Final output: $2.50\,\mathrm{mm}$.
Benchmark outcome: exact match $= 1$; LLM judge $= 1$.

The consequential difference is the input interpretation: the selected trajectory explicitly compares two distances before fixing $h = 5\,\mathrm{mm}$, whereas the baseline directly substitutes $h = 3\,\mathrm{mm}$. It then supplies a reproducible computation with the required rounding. The numerical checks establish what follows from each distance; they do not by themselves prove the physical interpretation. This example therefore illustrates reconsideration of an early modeling choice and checked finaliza-

tion, while keeping the unrelated execution failure of the second branch separate from the scientific analysis.