Beyond Outcome Filtering: Diagnosing and Correcting Behavioral Bias in Agentic SFT

Anonymous authors
Paper under double-blind review

Abstract

Supervised fine-tuning (SFT) for agentic language models is often built from demonstrations selected by task outcome. This creates a mismatch: curation evaluates whether a trajectory succeeds, while SFT learns from the entire trajectory. Successful demonstrations can therefore retain undesirable behaviors such as redundant retrieval, repetition, excessive verbosity, and failures to use available evidence. We introduce DiaCurate (Diagnose-and-Curate), a diagnostic-driven framework for agentic SFT data curation that identifies behavioral failures, applies targeted edits to demonstrations, and re-diagnoses the policy after retraining to detect collateral regressions. On a deep-research agent evaluated with HLE and BrowseComp, SFT on an outcome-filtered reference corpus improves task accuracy while increasing search inefficiency by 25.9 and 17.2 points and verbosity by 19.2 and 26.1 points on corresponding development sets. Directly shortening trajectories reduces unnecessary search but increases premature commitment and degrades accuracy. Re-diagnosis motivates an evidence-use refinement that reduces search inefficiency while preserving task performance, reaching 35.2 and 34.1 avg@3 versus 33.5 and 33.3 for the reference. These results show that successful trajectories are not necessarily good demonstrations: agentic SFT curation should account for both task outcomes and the behaviors learned from training trajectories.

1 Introduction

Large language models are increasingly used as agents that interact with external tools—including web search, page retrieval, and code execution—over extended multi-turn trajectories to solve complex tasks (Yao et al., 2023; Schick et al., 2023; Nakano et al., 2022). A common training pipeline uses supervised fine-tuning (SFT) on demonstration trajectories to establish agentic competence, followed by reinforcement learning (RL) or other post-training methods to further refine the agent's behavior (Ouyang et al., 2022; Bai et al., 2022).

For agentic SFT, selecting good demonstrations is critical but non-trivial. A trajectory may contain dozens of decisions: what to search for, which results to inspect, when to stop retrieving, how to use evidence, and how to formulate the final answer. Outcome-based curation evaluates whether the trajectory ultimately succeeds, whereas SFT learns from the entire retained trajectory. Consequently, a demonstration can receive a successful outcome while still containing behaviors that we would not want the student policy to imitate.

For example, a research agent may retrieve answer-bearing evidence early in a trajectory but continue searching for many additional turns, repeat queries or reasoning, and eventually produce an unnecessarily long answer. Such a trajectory can pass a correctness filter even though its research process is inefficient. More generally, successful demonstrations can retain undesirable behaviors such as redundant retrieval, repetition, protocol violations, excessive verbosity, and failures to recognize or use available evidence. Correctness filtering places no explicit constraint on these behaviors.

This creates a mismatch between *outcome selection* and *behavioral imitation*: SFT supervises intermediate agent actions along with the final answer. Successful demonstrations can

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.

therefore transmit undesirable behaviors even when their outcomes are correct. Data curation for agentic SFT must therefore consider not only whether a demonstration succeeds, but also which behaviors its trajectory teaches the student policy to imitate.

We introduce DiaCurate (Diagnose-and-Curate), a diagnostic-driven framework for agentic SFT data curation. DiaCurate closes the loop between policy behavior and training data: diagnose behavioral failures in the current policy, modify demonstrations to target those failures, retrain, and re-diagnose the resulting policy. A curation intervention is therefore evaluated by how it changes the trained policy, including any collateral failures it introduces.

We instantiate DiaCurate on a deep-research agent and evaluate it using the HLE and BrowseComp benchmark families (Phan et al., 2025; Wei et al., 2025). Our experiments reveal three main findings. First, SFT on the outcome-filtered reference corpus improves task accuracy, but also worsens several behavioral dimensions: avg@3 increases from 22.5 to 33.5 on HLE-500 and from 21.8 to 33.3 on BrowseComp-400, while search inefficiency increases by 25.9 and 17.2 points and verbosity by 19.2 and 26.1 points on the corresponding development sets. Second, directly editing trajectories to stop once sufficient evidence is found reduces unnecessary search but increases premature commitment and degrades task accuracy, showing that correcting one behavioral failure can introduce another. Third, re-diagnosis motivates TE3, which explicitly connects retrieved evidence to answer commitment; it reduces search inefficiency by 4.5 and 4.2 points while remaining within the prescribed error tolerance, and reaches 35.2 and 34.1 avg@3 compared with 33.5 and 33.3 for the outcome-filtered reference. Answer compression provides a complementary result, reducing verbosity by up to 9.4 and 16.0 points while improving pass@3 on both benchmarks.

We make three contributions:

1. **We identify a behavioral blind spot in outcome-based curation for agentic SFT.** Outcome-based selection constrains the final result but not the sequence of actions that SFT learns to imitate. We show empirically that improved task performance can coexist with substantial regressions in learned agent behavior, demonstrating that task success alone is insufficient to characterize the quality of agentic supervision.
2. **We introduce DiaCurate, a multidimensional behavioral diagnostic for agentic SFT.** DiaCurate separates process-level failures—such as repetition, protocol violations, search inefficiency, and verbosity—from outcome-conditioned failures to acquire or use answer-bearing evidence. The diagnostics are reproducible and can be applied consistently to training trajectories and fresh policy rollouts.
3. **We show how behavioral diagnosis can guide iterative data curation.** We use diagnosed failure modes to design targeted edits to successful demonstrations, retrain the policy, and re-diagnose its behavior. This feedback loop detects collateral regressions introduced by curation and guides subsequent refinement. We further show that editing the behavioral content of demonstrations can change the learned policy in ways that selection-only filtering does not reproduce.

Together, these results show that successful trajectories are not necessarily good demonstrations. Agentic SFT data should therefore be curated not only for whether trajectories reach correct outcomes, but also for the behaviors they teach the student policy to imitate.

2 Related Work

Research agents build on browser-assisted question answering, tool-use supervision, and interleaved reasoning and action (Nakano et al., 2022; Schick et al., 2023; Yao et al., 2023). Recent systems extend these approaches to long-horizon web research through task synthesis and trajectory processing (Li et al., 2025; Chu et al., 2026; Xie et al., 2026; Venus Team et al., 2026).

Data selection and adaptive curation. Instruction-tuning work emphasizes carefully curated examples (Zhou et al., 2023), LLM-based quality assessment (Chen et al., 2024), quality, complexity, and diversity (Liu et al., 2024), and influence on target capabilities (Xia et al., 2024). Other curation mechanisms include deduplication (Lee et al., 2022) and reward-

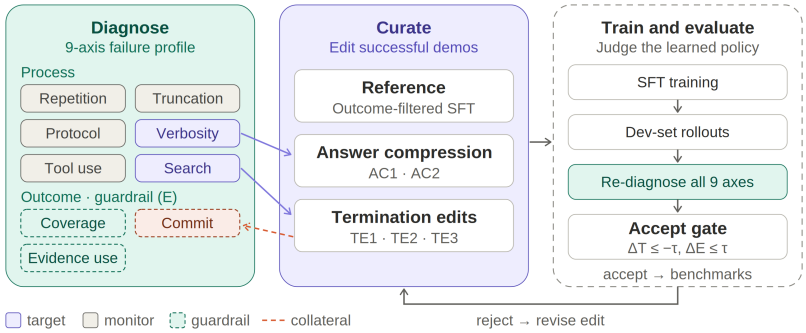

Figure 1: **Overview of DiaCurate.** *Diagnose*: a nine-axis profile scores the policy's rollouts. *Curate*: termination edits and answer compression revise successful demonstrations in the outcome-filtered reference corpus. *Train and evaluate*: each curated corpus is used for SFT, the resulting policy is re-diagnosed on all nine axes, and the edit is accepted when $\Delta T \leq -\tau$ and $\Delta E \leq \tau$.

ranked sampling (Touvron et al., 2023). Failure-driven approaches use counterexamples (Vejendla, 2025) or slice-level mixture adjustment (Dou & Yang, 2026); studies of incomplete SFT learning further motivate examining what models internalize (Xue et al., 2026). CurateEvo (Wang et al., 2026) evolves executable strategies from held-out failures for filtering, augmentation, and fine-grained trajectory refinement, constructing SFT data, RL data, and inference-time memory. DiaCurate shares this feedback-driven perspective but focuses on behavioral bias in successful demonstrations and the collateral failures introduced by correcting it. We assess targeted edits through the resulting SFT policy rather than demonstration quality alone.

Diagnosing agent failures. Agent diagnosis spans failure taxonomies (Cemri et al., 2025), annotated error traces (Deshpande et al., 2025), and agent- and step-level attribution (Zhang et al., 2025b). AgenTracer (Zhang et al., 2025a) constructs attribution supervision through counterfactual replay and fault injection. Other approaches use structured behavioral invariants (Bi et al., 2026) or dynamics learned from successful trajectories (Yeh et al., 2026). AgentDebug (Zhu et al., 2025) connects diagnosis to corrective feedback for agent recovery. Here, diagnosis guides offline SFT curation and re-diagnosis on fresh rollouts. Separating process-level measurements from outcome-conditioned errors exposes cases where reduced activity accompanies premature commitment or evidence-use failure, without claiming causal attribution to a single step.

Search efficiency and stopping. Excessive deliberation can impede agent interaction (Cuadron et al., 2025). DAS (Zhang et al., 2026) diagnoses over-search and under-search through factual and counterfactual trajectories and constructs preference pairs for direct preference optimization. We instead edit successful SFT demonstrations to target stopping, evidence use, and final-answer verbosity. Termination editing can reduce the student's search inefficiency while increasing premature commitment, motivating evidence-use refinement rather than further shortening.

3 A Framework for Diagnosis-Guided Data Curation

DiaCurate treats data curation as an intervention on the behavior learned by an agent. It connects three objects: the behavioral profile of the current policy, the distribution of supervised demonstrations, and the behavioral profile of the policy trained on the curated distribution. As illustrated in Figure 1, diagnosis motivates targeted changes to demonstrations; retraining and re-diagnosis then test whether those changes produce the intended behavioral correction. We first formulate this relationship and then describe our deep-research instantiation and results in Sections 4 and 5.

Table 1: Nine diagnostic axes. Higher scores indicate more severe diagnosed behavior. The first six describe the interaction process; the last three combine outcome and evidence-match information. Full definitions appear in Appendix B.3.

Diagnostic axis	Operational interpretation
Repetition (f_{rep})	Repeated identical tool calls or highly compressible assistant text.
Truncation (f_{trunc})	Per-turn token-cap stops or context-budget termination.
Protocol violations (f_{proto})	Missing or noncompliant answer or verification outputs.
Tool-use failures (f_{tool})	Failed tool calls or no retrieval on a retrieval task.
Search inefficiency (f_{search})	Unreferenced observations, continuation after an answer match, and trailing searches without new URLs.
Verbose completion (f_{verbose})	Final-answer length relative to the shortest accepted reference.
Premature commitment (f_{commit})	Incorrect, no detected answer match, and classified as an early model ending.
Insufficient coverage (f_{cover})	Incorrect, no detected answer match, and not classified as an early model ending.
Evidence-use failure (f_{extract})	Incorrect conditional on a detected answer match.

3.1 Outcome Selection and Behavioral Imitation

Let τ be a trajectory consisting of a task q, agent actions, and environment observations. At decision step t, the agent produces a_t conditioned on the interaction history H_t. Actions include reasoning, tool requests, and answer commitments. Given a demonstration distribution $\mathcal{D}$, SFT fits these actions through the objective $\mathcal{L}_{\text{SFT}}(\theta; \mathcal{D}) = -\mathbb{E}_{\tau\sim\mathcal{D}}[\sum_{t=1}^{T(\tau)} \log \pi_\theta(a_t \mid H_t)]$, where action likelihoods comprise the supervised agent tokens; environment observations supply context. Thus, retained intermediate actions contribute to the learning signal along with the final answer.

Let $R(\tau) \in \{0, 1\}$ indicate task success according to an external grader. Outcome filtering constructs $\mathcal{D}_0 = \mathcal{D}_{\text{src}}(\cdot \mid R = 1)$. It imposes no separate constraint on how a successful trajectory was produced. More precisely, for a binary process-pathology indicator $B(\tau)$ and nonzero success probability, expectations under the source distribution satisfy $\Pr(B = 1 \mid R = 1) - \Pr(B = 1) = \text{Cov}(B, R)/\Pr(R = 1)$. Conditioning on success can therefore decrease, preserve, or increase a pathology's prevalence, depending on its association with success. For example, prolonged searching can co-occur with a correct answer even when part of that search is unnecessary. The identity describes selection in the demonstration pool; its effect on the trained policy also depends on the learning process and the histories that the policy visits at evaluation time.

This motivates assessing a curated dataset through the policy it produces. Write $\pi_c = \text{Train}(\mathcal{D}_c; \xi)$ for the policy obtained from recipe c, where ξ specifies the initialization and training configuration. The curation problem is to change $\mathcal{D}_c$ so that π_c exhibits less undesirable behavior while preserving task success. Improving the appearance of the demonstrations alone does not establish this outcome.

3.2 Behavioral Diagnosis of the Learned Policy

We characterize behavior using nine complementary diagnostic axes (Table 1). The first six describe the interaction process without conditioning on final-answer correctness; the final three distinguish incorrect outcomes by detected evidence and termination behavior. These are operational diagnostics, not mutually exclusive failure classes or causal attributions. Appendix B.3 consolidates their equations, detectors, thresholds, and denominators.

Process-level diagnosis. The process axes measure recurrent actions or text, budget interruptions, protocol compliance, tool-use reliability, retrieval inefficiency, and answer verbosity. Raw trajectory length and tool-call count provide context rather than sufficient evidence of failure: difficult questions may require sustained investigation. In particular, search inefficiency combines three proxy signals rather than penalizing search volume, while verbosity measures relative answer length rather than reasoning length. These scores can identify undesirable patterns even in successful teacher demonstrations.

Outcome-conditioned diagnosis. Less activity can also reflect failure to acquire or use relevant information. On rollouts with a checkable short-form reference answer, let $C(\tau)$

indicate a normalized reference-answer match in the tool observations, and let $R(\tau)$ indicate correctness according to the external evaluation grader, not the agent's self-verification verdict. Among incorrect rollouts with $C = 0$, premature commitment identifies an early-ending proxy based on the last message role and remaining context budget; insufficient coverage contains the remainder. The latter therefore does not require complete budget exhaustion. Evidence-use failure instead measures incorrectness conditional on $C = 1$. The first two scores use all checkable rollouts as their denominator, whereas the third uses only checkable rollouts with a detected match.

A match does not establish evidence sufficiency, and a missing match does not rule out support obtained through synthesis. These proxy limitations apply both to the outcome-conditioned axes and to the post-match component of search inefficiency. The decomposition helps distinguish reduced research effort from early unsuccessful termination without treating literal answer detection as proof that further research was unnecessary.

3.3 Failure Profile

For a policy π, let $\mathcal{R}_b(\pi; h)$ denote fresh rollouts on a fixed development set $b \in \mathcal{B}$ under serving configuration h, including tool access, context and generation limits, stopping rules, and verification protocol. Alongside task accuracy $A_b(\pi)$, we retain

$$\mathbf{f}(\mathcal{R}; h) = \big(f_{\text{rep}}, f_{\text{trunc}}, f_{\text{proto}}, f_{\text{tool}}, f_{\text{search}}, f_{\text{verbose}}, f_{\text{commit}}, f_{\text{cover}}, f_{\text{extract}}\big). \tag{1}$$

Each coordinate lies in $[0, 100]$, with higher values indicating worse diagnosed behavior. Their units differ: the profile combines event-rate averages, per-call rates, within-rollout fractions, conditional error, and continuous severity. Multi-component axes use equal component weights. For descriptive comparisons, we use $F(\mathcal{R}; h) = \frac{1}{9}\sum_{j=1}^{9} f_j(\mathcal{R}; h)$. F is not the intervention-selection objective: an aggregate improvement can conceal a regression on another axis. Selection uses a targeted improvement and an outcome tolerance (Section 4.4); the remaining measurements guide re-diagnosis. Because the final three axes use correctness, an association between F and accuracy also contains shared outcome information.

4 Experimental Setup

4.1 Agent and Training Data

We train a deep-research agent based on NVIDIA Nemotron 3.5 Lightning, a mixture-of-experts model with 30B parameters and 3B active parameters per forward pass (NVIDIA, 2026). The agent uses a ReAct-style loop (Yao et al., 2023) with web search, page visits, and a sandboxed Python interpreter. SFT supervises the agent's reasoning, tool requests, and answers through next-token prediction, with tool observations excluded from the loss. Our RedSearcher reference corpus contains 5,298 examples from DR-Venus-processed trajectories (Chu et al., 2026; Venus Team et al., 2026). Upstream cleaning retains correct web-research traces and removes Python calls. We apply length filtering, protocol conversion, and reasoning regeneration, retaining successful traces with affirmative verification verdicts. The name denotes the processed corpus and its SFT reference model, not the DR-Venus model; Appendix C.1 details its construction.

4.2 Reference and Baseline Conditions

We report the model before our SFT and use SFT on the processed RedSearcher corpus as the fixed reference for all intervention comparisons. Two additional baselines test whether further selection of demonstrations addresses behavioral deficiencies without editing the retained trajectories. *Error-mode-filtered SFT* removes the highest-scoring teacher trajectories according to their per-trace diagnostics. *LLM-scored SFT* uses a judge that reads the question and trajectory without the reference answer, rejecting unsupported answers, fabricated evidence, and low-quality traces. Appendix B.4 specifies the scoring rules and retained populations. Diagnosis-guided edits use the same reference corpus; their targets and construction are introduced after the reference diagnosis in Section 5.

4.3 Datasets and Evaluation

We use Research-Reasoning-Dev (500 questions) and Web-Discovery-Dev (400) for diagnosis, intervention acceptance, and checkpoint selection, and HLE-500 and BrowseComp-400 for benchmark reporting. Development questions are independently synthesized, not sampled from these benchmarks, and excluded together with their rollouts from SFT and trajectory repair. The development sets were frozen after calibration against fixed-reference benchmark profiles (Appendix D.2). Under a shared research-and-verification protocol, we generate three attempts per question and report avg@3 and pass@3. Appendix B.1 provides dataset and evaluation details.

4.4 Development and Selection Protocol

For each candidate, fresh rollouts on the fixed development questions provide a targeted diagnostic score T and the error rate E among checkable rollouts. If p_{seen} is the fraction of these rollouts with a detected reference-answer match, then $E = f_{\text{commit}} + f_{\text{cover}} + p_{\text{seen}} f_{\text{extract}}$. This error is measured on the checkable subset and differs from benchmark error over all questions.

Let ΔT and ΔE denote changes from the fixed RedSearcher SFT reference. A candidate is accepted when $\Delta T \leq -\tau$ and $\Delta E \leq \tau$, with $\tau = 3$ points, assessed separately on each development set. Candidates that improve the target but exceed the error tolerance are rejected and revised, always against the same reference. Refinement continues until acceptance and stops when a revision no longer improves the target by at least τ. The rule and tolerance were fixed before the experiments. The remaining diagnostic axes guide revision, while F remains descriptive.

All conditions use the same training compute budget. Within each run, we select the checkpoint with the highest mean avg@3 across the two development sets. Intervention acceptance and checkpoint selection use development measurements only; candidate-specific benchmark outcomes do not inform either decision.

5 Results

We first diagnose the behavior learned from outcome-filtered demonstrations, then examine edits motivated by that diagnosis. The intervention studies track targeted corrections and collateral failures on the development sets; the final comparison assesses benchmark outcomes and selection-only baselines. All decisions follow the protocol in Section 4.4.

5.1 Behavioral Costs of Outcome-Filtered SFT

Accuracy gains coexist with behavioral regressions. Outcome-filtered SFT improves avg@3 on both primary benchmarks but also raises the aggregate failure index F on both development sets (Tables 2, 3a, and 3b).

Table 2: Development diagnostics before and after RedSearcher SFT (Δ = RedSearcher − base; lower is better). Red: increases of at least 2, 5, 15 points; blue: decreases of at least 5.

	Research-Reasoning-Dev			Web-Discovery-Dev		
	Base model	RedSearcher	Δ	Base model	RedSearcher	Δ
f_{rep}	9.1	24.3	+15.2	25.5	35.0	+9.5
f_{trunc}	21.3	23.0	+1.7	41.4	38.4	−3.0
f_{proto}	21.2	14.9	−6.3	29.2	31.1	+1.9
f_{tool}	14.1	14.9	+0.8	1.6	16.1	+14.5
f_{search}	34.3	60.2	+25.9	41.9	59.1	+17.2
f_{verbose}	8.2	27.4	+19.2	4.8	30.9	+26.1
f_{commit}	47.4	26.8	−20.6	19.3	7.4	−11.9
f_{cover}	15.6	28.8	+13.2	41.6	38.9	−2.7
f_{extract}	31.5	36.2	+4.7	48.0	45.1	−2.9
F	22.5	28.5	+6.0	28.1	33.6	+5.5

The diagnostic profile exposes a common trade-off: premature commitment decreases, while search inefficiency, repetition, and verbosity increase on both development sets. Search inefficiency and verbosity show the largest regressions, indicating that more reliable task completion can coexist with unnecessary research and excessive elaboration. Evidence coverage also deteriorates on Research-Reasoning-Dev, as does tool-use reliability on Web-Discovery-Dev.

These results illustrate the mismatch between selecting successful outcomes and learning desirable behavior. The two largest regressions, search inefficiency and verbosity, are properties of the demonstrations themselves: the teacher traces continue to search after sufficient evidence has been retrieved and end with long answers. We therefore edit the traces directly, with termination edits to reduce over-searching and answer edits to reduce verbosity. Each trained variant is re-diagnosed on all axes to assess the intended correction and expose collateral changes.

5.2 Termination Editing

To address the reference's search inefficiency, we target $T = f_{\text{search}}$ and test whether editing demonstrated stopping behavior teaches the agent to stop once the evidence is sufficient.

Intervention. TE1 and TE2 retain the research prefix up to the first turn at which the retrieved evidence supports the answer and replace the remaining continuation with an answer and verification grounded in that prefix. Because a reference-answer match alone does not establish sufficiency, a grounding check assesses evidence references against the retained observations; rejected edits leave the original trajectory unchanged.

TE1 applies the edit to every trajectory with a located sufficiency turn. TE2 reduces exposure to edited endings, prioritizing trajectories with substantial continued search after a reference-answer match. Both variants preserve corpus size. This comparison tests whether more selective editing limits the risk of premature stopping; Appendix C.2 gives the construction details and edit proportions.

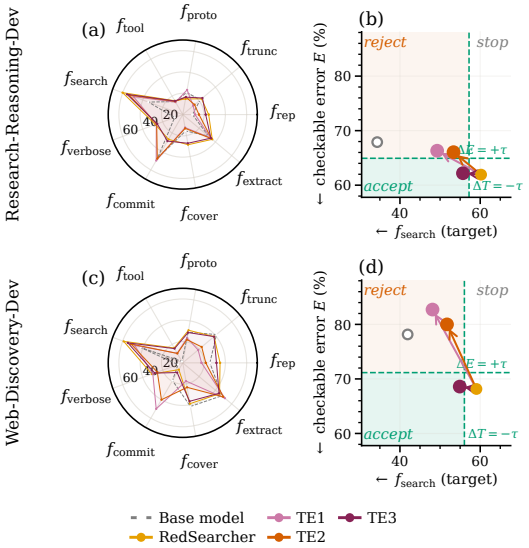

Figure 2: Panels (a, b) and (c, d) summarize termination editing on Research-Reasoning-Dev and Web-Discovery-Dev, respectively. Radar plots compare diagnostic profiles; adjacent panels show search inefficiency ($T = f_{\text{search}}$) versus checkable error E. Arrows originate at RedSearcher.

Reduced search does not ensure reliable stopping. Both variants reduce search inefficiency, repetition, and truncation on both development sets, but substantially increase premature commitment (Figure 2; Appendix Table 11). The more selective TE2 moderates the commitment regression while yielding smaller reductions in search inefficiency. Neither variant restores premature commitment to the reference level. The outcome-conditioned profile explains why lower process scores are insufficient. Although insufficient coverage decreases, the combined share of incorrect rollouts without detected evidence, $f_{\text{commit}} + f_{\text{cover}}$, increases on both development sets. This pattern is consistent with more frequent early unsuccessful termination rather than improved evidence acquisition. For both variants, the increase in checkable error exceeds the prescribed tolerance, leading to rejection despite a lower F.

Refinement toward explicit evidence use. The remaining commitment failures motivate TE3, which makes the connection between retrieved evidence and answer commitment explicit while preserving subsequent research. Teacher-generated passages identify an observed answer candidate, distinguish supported constraints from unresolved ones, and connect the final commitment to the earlier evidence. Appendix C.2.3 gives the construction details. Although it removes no research turns, TE3 reduces search inefficiency while keeping the change in checkable error within tolerance. Premature commitment is much closer to the reference than under the earlier variants (Figure 2; Appendix Table 11). The refinement

is accepted on both development sets. Search inefficiency remains above the base-model level, leaving room for further correction.

5.3 Answer Compression

The termination study shows that modifying the research process can introduce substantial outcome costs. We therefore examine a more local intervention that targets final-answer verbosity, $T = f_{\text{verbose}}$, while preserving the research history.

Intervention. A teacher shortens the final answer while retaining the requested content, qualifiers, and uncertainty. Reasoning, tool calls, observations, and verification turns remain unchanged. Using the same rewritten answers, AC1 selects answers by a reference-length threshold, whereas AC2 selects a fixed share of the most verbose trajectories. These selection rules test whether concentrating edits on the most verbose answers better preserves other behaviors. Appendix C.3 gives the thresholds and construction details.

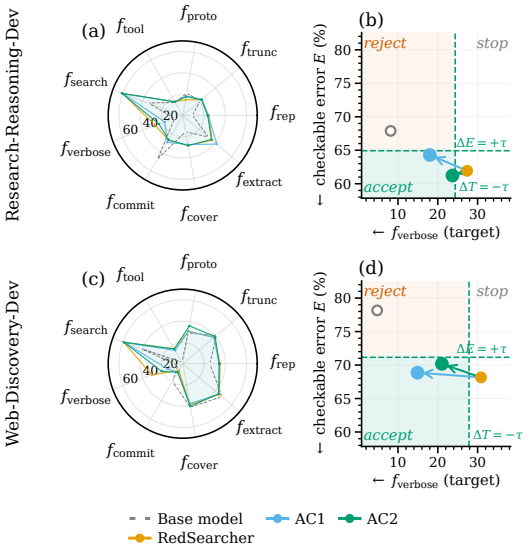

Figure 3: Both answer compression variants reduce verbosity and meet the acceptance criterion on both development sets. Panel order and plotting conventions follow Figure 2, with $T = f_{\text{verbose}}$.

Targeted reductions with smaller behavioral changes. Both variants reduce verbosity on both development sets, with larger reductions under AC1 (Figure 3; Appendix Table 11). Search inefficiency, repetition, and evidence coverage remain close to the reference. However, AC1 increases evidence-use failure on Research-Reasoning-Dev, and both variants increase protocol violations there; AC2 also worsens protocol compliance on Web-Discovery-Dev. Localized demonstration edits can therefore affect behaviors beyond their immediate target.

Both variants meet the acceptance criterion on both development sets: the verbosity reductions exceed the target threshold, while changes in checkable error remain within tolerance. Verbosity remains above the base-model level, motivating further evaluation of answer compression, although this residual gap alone does not establish that stronger edits would help.

5.4 Benchmark Outcomes and Comparisons

Tables 3a and 3b bring together development decisions and final benchmark outcomes, testing whether corrections selected on development tasks transfer to the reporting benchmarks.

Development decisions align with observed benchmark outcomes. Every rejected variant reduces avg@3 on both primary benchmarks, whereas every accepted variant improves pass@3. TE3 improves both benchmark metrics over the outcome-filtered reference. For answer compression, avg@3 remains close to the reference, with small gains or losses depending on the variant and benchmark. Acceptance therefore accompanies better pass@3 in these experiments without implying uniform avg@3 gains. The error constraint is essential to this distinction: ranking edits by their reduction in over-searching or F alone would favor the rejected termination variants over TE3.

Editing versus selection-only filtering. Error-mode-filtered SFT leaves avg@3 close to the reference while increasing the policy's F; LLM-scored SFT lowers both benchmark metrics. TE3 has higher avg@3 and pass@3 point estimates than both baselines on both

Table 3: Development diagnostics and acceptance decisions are reported alongside benchmark scores (%). Changes are measured in points relative to RedSearcher; dashes denote unavailable or inapplicable entries.

		Development (Research-Reasoning-Dev)				HLE-500 ↑	
Method	Version	$F \downarrow$	ΔT	ΔE	Decision	avg@3	pass@3
Base model	–	22.5	–	–	–	22.5	36.2
Outcome-filtered SFT	RedSearcher (reference)	28.5	ref.	ref.	–	33.5	47.4
LLM-scored SFT	–	31.7	–	–	–	31.3	45.6
Error-mode-filtered SFT	–	29.5	–	–	–	33.7	48.8
Termination editing	TE1	25.9	−10.9	+4.4	reject	25.1	40.0
	TE2	26.2	−6.8	+4.1	reject	31.0	48.4
	TE3	27.6	−4.5	+0.3	accept	**35.2**	**51.4**
Answer compression	AC1	28.5	−9.4	+2.4	accept	32.9	49.8
	AC2	28.2	−3.7	−0.7	accept	34.5	49.4

(a) Research-Reasoning-Dev diagnostics and decisions with HLE-500 outcomes

		Development (Web-Discovery-Dev)				BrowseComp-400 ↑	
Method	Version	$F \downarrow$	ΔT	ΔE	Decision	avg@3	pass@3
Base model	–	28.1	–	–	–	21.8	35.5
Outcome-filtered SFT	RedSearcher (reference)	33.6	ref.	ref.	–	33.3	47.2
LLM-scored SFT	–	38.5	–	–	–	27.2	43.5
Error-mode-filtered SFT	–	34.3	–	–	–	33.7	47.8
Termination editing	TE1	29.7	−11.0	+14.6	reject	19.3	32.0
	TE2	29.7	−7.4	+11.8	reject	23.8	39.2
	TE3	32.1	−4.2	+0.4	accept	34.1	49.8
Answer compression	AC1	31.4	−16.0	+0.7	accept	**34.6**	**50.8**
	AC2	33.3	−9.9	+2.0	accept	33.2	48.8

(b) Web-Discovery-Dev diagnostics and decisions with BrowseComp-400 outcomes

benchmarks (Tables 3a and 3b). These comparisons suggest that the tested selection rules do not reproduce the benefits of targeted editing. Appendix B.4 details the baselines.

Comparison with published research agents. TE3 achieves the highest reported HLE and GAIA scores among the agents in Table 4 (Tongyi DeepResearch Team, 2025; Xie et al., 2026; Li et al., 2026). On BrowseComp, it exceeds the base model and OpenResearcher but trails the leaders. Although evaluation protocols differ, these results suggest that targeted curation of a modest SFT corpus can yield competitive research capabilities. The remaining BrowseComp gap motivates extending the diagnostic loop to evidence coverage and evidence use (Table 2).

Table 4: Benchmark performance (%) is reported for TE3 and published 30B-A3B research agents under their respective evaluation protocols. Bold indicates the best score in each column.

	HLE	BrowseComp	GAIA
Base model	22.5	21.8	63.0
Published			
Tongyi-DeepResearch	32.9	**43.4**	70.9
QUEST-30B	24.6	37.0	69.0
OpenResearcher	19.6	26.3	64.1
Ours			
TE3 (termination editing)	**35.2**	34.1	**71.3**

6 Conclusion

We presented DiaCurate, a framework linking behavioral diagnosis to targeted curation of agentic SFT data. In our deep-research setting, termination editing lowers search-inefficiency scores but can increase premature commitment, while answer compression reduces verbosity with benchmark-dependent outcome changes. Re-diagnosis motivates TE3, which makes evidence use explicit without shortening demonstrations. It meets the acceptance criterion on both development sets and improves both benchmark metrics over the outcome-filtered reference. The tested selection-only baselines do not reproduce these joint improvements. These findings support evaluating curation through the resulting policy's task performance and behavioral changes, including collateral failures, rather than demonstration quality alone. Appendix A discusses the study's scope.

AI Use Statement

We used large language models to assist with literature discovery and summarization, drafting and reorganizing manuscript sections, and proofreading and improving readability. AI assistance also provided feedback on methodological descriptions, experimental comparisons, and the interpretation of results.

Within the research pipeline, LLMs served as teachers and judges for synthetic-data generation and curation, including reasoning regeneration, verification, demonstration-quality assessment, termination editing, evidence-use refinement, and answer compression. The corresponding models, prompts, and processing procedures are described in Appendices C and B.

The authors retain full responsibility for the final manuscript, including its claims, citations, methodology, data, experimental results, and conclusions, and for verifying the accuracy and appropriateness of AI-assisted content.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Jiayi Bi, Yanjie Gao, Yuanmin Xie, Liqun Li, Tianyin Xu, Fan Yang, and Mao Yang. Diagnosing with insights: Structured analysis of agent failures via behavioral abstractions, 2026. URL https://arxiv.org/abs/2609.02371.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail?, 2025. URL https://arxiv.org/abs/2503.13657.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. AlpaGasus: Training a better Alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.08701.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Shengchao Hu, Dongdong Kuang, Chenxiao Zhao, Cheng Xiang, Ming Liu, Bing Qin, and Xing Yu. REDSearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*, 2026. URL https://arxiv.org/abs/2602.14234.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, Nicholas Thumiger, Aditya Desai, Ion Stoica, Ana Klimovic, Graham Neubig, and Joseph E. Gonzalez. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks, 2025. URL https://arxiv.org/abs/2502.08235.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. TRAIL: Trace reasoning and agentic issue localization, 2025. URL https://arxiv.org/abs/2505.08638.

Yipu Dou and Wang Yang. Mosaic: Multi-objective slice-aware iterative curation for alignment, 2026. URL https://arxiv.org/abs/2603.18637.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models

better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 120091–120131, 2025. doi: 10.52202/085713-4011.

Zhuofeng Li, Dongfu Jiang, Xueguang Ma, Haoxiang Zhang, Ping Nie, Yuyu Zhang, Kai Zou, Jianwen Xie, Yu Zhang, and Wenhu Chen. Openresearcher: A fully open pipeline for long-horizon deep research trajectory synthesis. *arXiv preprint arXiv:2603.20278*, 2026.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=BTKAeLqLMw.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2311.12983.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022. URL https://arxiv.org/abs/2112.09332.

NVIDIA. Nvidia nemotron 3.5 lightning. https://github.com/NVIDIA-NeMo/Nemotron, 2026. 30B total parameters, 3B active parameters.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Long Phan et al. Humanity's last exam. *Nature*, 2025. doi: 10.1038/s41586-025-09962-4.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessí, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: language models can teach themselves to use tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Tongyi DeepResearch Team. Tongyi DeepResearch technical report. arXiv:2510.24701, 2025.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, D. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, A. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 7 2023. doi: 10.48550/arXiv.2307.09288. URL https://www.semanticscholar.org/paper/104b0bb1da562d53cbda87aec79ef6a2827d191a.

Harshil Vejendla. Teaching by failure: Counter-example-driven curricula for transformer self-improvement. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 423–431, Mumbai, India, December 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-ijcnlp.24. URL https://aclanthology.org/2025.findings-ijcnlp.24/.

Venus Team, Sunhao Dai, Yong Deng, Jinzhen Lin, Yusheng Song, Guoqing Wang, Xiaofeng Wu, Yuqi Zhou, Shuo Yang, Zhenzhe Ying, Zhanwei Zhang, Changhua Meng, and Weiqiang Wang. DR-Venus: Towards frontier edge-scale deep research agents with only 10k open data. *arXiv preprint arXiv:2604.19859*, 2026.

Dingzirui Wang, Xuanliang Zhang, Keyan Xu, Qingfu Zhu, and Wanxiang Che. Curateevo: Data-curation evolving for agentic post-training. *arXiv preprint arXiv:2607.06140*, 2026. URL https://arxiv.org/abs/2607.06140.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*, 2024.

Jian Xie, Tianhe Lin, Zilu Wang, Yuting Ning, Yuekun Yao, Tianci Xue, Zhehao Zhang, Zhongyang Li, Kai Zhang, Yufan Wu, Shijie Chen, Boyu Gou, Mingzhe Han, Yifei Wang, Vint Lee, Xinpeng Wei, Xiangjun Wang, Yu Su, and Huan Sun. QUEST: Training frontier deep research agents with fully synthetic tasks. *arXiv preprint arXiv:2605.24218*, 2026.

Chao Xue, Yao Wang, Mengqiao Liu, Di Liang, Xingsheng Han, Peiyang Liu, Xianjie Wu, Chenyao Lu, Lei Jiang, Yu Lu, Haibo Shi, Shuang Liang, Minlong Peng, and Flora D. Salim. Why supervised fine-tuning fails to learn: A systematic study of incomplete learning in large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30186–30213, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1393. URL https://aclanthology.org/2026.acl-long.1393/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Samuel Yeh, Yiwen Zhu, Shaleen Deep, and Sharon Li. Tracing agentic failure from the flow of success, 2026. URL https://arxiv.org/abs/2607.12747.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. AgenTracer: Who is inducing failure in the LLM agentic systems?, 2025a. URL https://arxiv.org/abs/2509.03312.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? On automated failure attribution of LLM multi-agent systems, 2025b. URL https://arxiv.org/abs/2505.00212.

Wenlin Zhang, Kuicai Dong, Junyi Li, Yingyi Zhang, Xiaopeng Li, Pengyue Jia, Yi Wen, Derong Xu, Maolin Wang, Yichao Wang, Yong Liu, and Xiangyu Zhao. To search or not to search: Aligning the decision boundary of deep search agents via causal intervention. In *Proceedings of the ACM Web Conference 2026 (WWW '26)*. ACM, 2026. doi: 10.1145/3774904.3792235.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.11206.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, Xiaoteng Ma, Xiaodong Yu, Gowtham Ramesh, Jialian Wu, Zicheng Liu, Pan Lu, James Zou, and Jiaxuan You. Where LLM agents fail and how they can learn from failures, 2025. URL https://arxiv.org/abs/2509.25370.

A Limitations

The framework was tested under one model and one task setting. Every run trains the same base model, Nemotron 3.5 Lightning, for a single agent type, deep research with web search and page visits. The nine failure modes were chosen for search tasks, where the agent must find, extract, and commit to an answer in retrieved evidence, and three of them are defined through the reference answer. Whether the same diagnostics, the acceptance rule, and its tolerance carry over to other base models or to agent tasks without this structure, such as coding or long-horizon planning, has not been tested.

B Experimental Protocol and Diagnostic Details

This appendix specifies the measurement and training protocol supporting Sections 4 and 5. Corpus construction appears in Appendix C; complete quantitative results appear in Appendix D.

B.1 Evaluation Datasets and Protocol

Table 5: Development and reporting sets. Numbers in parentheses are question counts.

Task family	Development set	Reporting benchmark
Research reasoning	Research-Reasoning-Dev (500)	HLE-500 (500)
Web discovery	Web-Discovery-Dev (400)	BrowseComp-400 (400)

Development-set construction. We construct Research-Reasoning-Dev (500 questions) from the 41,918-question HLE-OTS pool and Web-Discovery-Dev (400 questions) from the 842-question BC-OTS pool. Both exclude text matches to checked SFT corpora; Research-Reasoning-Dev additionally excludes matches to public HLE. Selection targets the subject-category and difficulty distributions of HLE-500 and BrowseComp-400, respectively. Research-Reasoning-Dev matches category–difficulty groups while prioritizing questions where tool use is helpful or critical. Web-Discovery-Dev matches the observed accuracy distribution and approximately matches topic composition, with source proportions and question lengths as secondary criteria. Both preserve the original questions and reference answers and serve as fixed development sets for diagnosing model behavior and comparing data-curation interventions.

Reporting benchmarks. HLE-500 is the fixed WebThinker subset of 500 text-only questions from Humanity's Last Exam (Li et al., 2025; Phan et al., 2025), emphasizing expert-level reasoning across academic domains. BrowseComp-400 contains 400 questions randomly sampled once from BrowseComp (Wei et al., 2025), emphasizing persistent web search and information discovery. Both reporting subsets remain fixed across model comparisons. GAIA (Mialon et al., 2024) provides an additional evaluation of tool-assisted problem solving in the published-agent comparison.

Development-set calibration. Before the intervention runs, each development set was selected and refined so that the fixed RedSearcher reference run's nine-axis diagnostic profile on it approximated the reference's profile on the paired benchmark. The calibration compared these diagnostic profiles only; benchmark accuracy was not used. Appendix D.2 reports the resulting comparison. The development sets were then fixed for intervention diagnosis and selection.

Split isolation and naming. The development prompts and their rollouts are excluded from SFT and trajectory repair. The only use of the primary reporting benchmarks before final evaluation is the profile calibration described above. Diagnostic figures and run records use the earlier names HLE-dev and BrowseComp-dev for Research-Reasoning-Dev and Web-Discovery-Dev, respectively. Both pools are internal question sets with reference answers; overlap with training data is excluded by exact text matching, and no semantic-overlap check was applied.

Evaluation protocol. All four question sets use a shared research-and-verification protocol with the serving settings in Appendix B.2. A common source blocklist applies to both search and page visits across checkpoints and splits.

Table 6: Training and evaluation settings. Corpus-specific length filtering and rendering are described in Appendices C.1 and C.3.

Setting	Value
SFT loss	Next-token prediction on agent actions
Tool observations in SFT loss	Masked
SFT trajectory-length limit	65K Nemotron tokens
Evaluation context	131,072 tokens
Evaluation per-turn generation limit	16K tokens
Evaluation rollouts per question	$k = 3$
Evaluation self-verification	Enabled
Source blocklist	Fixed list of blocked sources, applied to all evaluations
Sampling temperature	0.85
Sampling top-p	0.95

Table 7: Training configuration, shared by all conditions.

Framework	Optimizer	Learning rate	Warmup	Batch size	Epochs	Updates
SkyRL, Megatron backend	AdamW	5×10^{-6}	2 steps	256	6	132

B.2 Training and Serving Configuration

All conditions use the training configuration in Table 7. The termination editing variants retain the reference corpus's row count and order. The answer compression variants share the validation split and length-filtered training population described in Appendix C.3.

Source-data filtering and training serialization are separate stages: the reference source filter uses Qwen3-8B token lengths (Appendix C.1), whereas the answer compression variants are checked after Nemotron rendering (Appendix C.3).

B.3 Diagnostic Definitions and Implementation

This subsection specifies the nine coordinates of Eq. 1, combining the formal definitions with their detector implementations. The profile describes a rollout population under a fixed serving configuration; it is not a partition of trajectories into nine exclusive classes. All scores lie in $[0, 100]$, with higher values indicating worse diagnosed behavior, but they are not all percentages of failed rollouts.

Measurement and aggregation. Let $\mathcal{R}$ be a rollout population and let h specify tool access, context and generation limits, stopping rules, and verification protocol. For a nonempty eligible population $\mathcal{A}$, write

$$\widehat{\Pr}_{\mathcal{A}}(B) = \frac{1}{|\mathcal{A}|} \sum_{x \in \mathcal{A}} \mathbf{1}\{B(x)\}, \qquad \widehat{\mathbb{E}}_{\mathcal{A}}[z] = \frac{1}{|\mathcal{A}|} \sum_{x \in \mathcal{A}} z(x).$$

The population may comprise rollouts or individual tool calls, as specified below. Multi-component axes take the equal-weight mean of their component scores, not a pooled rate across unlike denominators. Eligible populations must be preserved when aggregating components, and their sample sizes are necessary for interpreting changes. These diagnostics guide intervention hypotheses rather than establish causal failure mechanisms. The implementation combines a standard-library-only trajectory module with a serving-level module using finish reasons, termination causes, and tool-return metadata. Retrieval omission and the termination-based outcome split are computed from graded rollout dumps. The detectors are reused in training-data inspection and offline evaluation where their required inputs are available; outcome-conditioned scores also require an external correctness judgment.

B.3.1 Repetition

Let $A_{\text{loop}}(\tau)$ indicate a run of at least three identical consecutive tool calls, with the same tool and arguments. Let $A_{\text{text}}(\tau)$ indicate that the concatenated assistant text has a raw-to-gzip compression ratio of at least six. Both components are rollout-level indicators:

$$f_{\text{rep}} = 50 \left[\widehat{\Pr}_{\mathcal{R}}(A_{\text{loop}}) + \widehat{\Pr}_{\mathcal{R}}(A_{\text{text}}) \right]. \tag{2}$$

The first detector identifies exact action recurrence; the second captures compressible textual recurrence. Neither determines whether a repeated action was semantically necessary or measures task progress directly.

B.3.2 Budget-Induced Truncation

Let $A_{\text{turncut}}(\tau)$ indicate that at least one generation was stopped by the per-turn token cap, as recorded by the server's length finish reason. Let $A_{\text{envcut}}(\tau)$ indicate that the environment terminated the episode because its context budget was exhausted. Then

$$f_{\text{trunc}} = 50 \left[\widehat{\Pr}_{\mathcal{R}}(A_{\text{turncut}}) + \widehat{\Pr}_{\mathcal{R}}(A_{\text{envcut}}) \right]. \tag{3}$$

A per-turn cutoff can occur even if the rollout later completes. Likewise, the context-termination event is defined by serving metadata; it should not be conflated with the separate absence-of-answer event below. Their prevalence depends on the serving limits, which must therefore be held fixed in diagnostic comparisons.

B.3.3 Answer-Protocol Violations

We distinguish three rollout-level events. A_{noans} indicates that no closed `<answer>` block is ever committed. A_{ansbad} indicates a noncompliant answer turn, including an answer inside a tool call, multiple answer blocks in an answer turn, or an answer written in body text with an empty answer field. A_{verifbad} indicates that a required verification reply contains no valid `<verification_result>` verdict. Under the research-and-verification harness,

$$f_{\text{proto}} = \frac{100}{3} \left[\widehat{\Pr}_{\mathcal{R}}(A_{\text{noans}}) + \widehat{\Pr}_{\mathcal{R}}(A_{\text{ansbad}}) + \widehat{\Pr}_{\mathcal{R}}(A_{\text{verifbad}}) \right]. \tag{4}$$

These events concern schema compliance, not semantic correctness, and may overlap. Under a ReAct-only harness that stops at the first answer, verification is not required and the score is 100/2 times the sum of the first two proportions; the third component is not inserted as zero.

B.3.4 Tool-Use Failures

Let $\mathcal{C}_{\text{py}}$, $\mathcal{C}_{\text{search}}$, and $\mathcal{C}_{\text{visit}}$ be the eligible Python, search, and page-visit calls pooled across $\mathcal{R}$. Their respective failure events are Python executions that raise an exception (A_{error}), searches returning zero results (A_{zero}), and visits whose URL does not parse ($A_{\text{malformed}}$). A fourth event, $A_{\text{no-retrieval}}(\tau)$, indicates that a rollout invokes neither search nor page visit on a retrieval task. The score is

$$\begin{aligned} f_{\text{tool}} = \frac{100}{4} \big[& \widehat{\Pr}_{\mathcal{C}_{\text{py}}}(A_{\text{error}}) + \widehat{\Pr}_{\mathcal{C}_{\text{search}}}(A_{\text{zero}}) \\ & + \widehat{\Pr}_{\mathcal{C}_{\text{visit}}}(A_{\text{malformed}}) + \widehat{\Pr}_{\mathcal{R}}(A_{\text{no-retrieval}}) \big]. \end{aligned} \tag{5}$$

Thus, three components are per-call rates and the fourth is a per-rollout rate. Retrieval omission is a task-specific tool-choice signal, not a failed invocation. The quoted-query rate is an auxiliary descriptor and is excluded from this score. The detector does not purport to cover every possible tool error or judge the semantic usefulness of every successful call.

B.3.5 Search Inefficiency

Search count is reported as context but is not itself a failure component. We instead combine three within-rollout fractions, each expressed on $[0, 1]$ before the final scaling below.

No detected short-window use, $U(\tau)$. For each tool observation, the detector examines the next three assistant turns. After stripping tool-call blocks from the text, it checks for at least two content unigrams, one number, or one content bigram introduced by that observation and absent from the question and all earlier text. Following a URL returned by the observation also counts as a reference. $U(\tau)$ is the fraction of observations for which no such reference is detected. This is a bounded-window lexical proxy: an observation used later, or only through implicit synthesis, can still be scored as unreferenced. It should not be described as information that is provably never used.

Continuation after an answer match, $G(\tau)$. On rollouts with a detected reference-answer match in the observations ($C = 1$, defined below), let $s(\tau) \in [0, 1]$ be the normalized position of its first appearance in the observation stream. Define $G(\tau) = 1 - s(\tau)$. The component measures continuation after a literal answer match, not after verified evidence sufficiency. Additional research may still be needed to check the question's constraints.

Trailing non-novel searches, $N(\tau)$. $N(\tau)$ is the fraction of search calls issued after the last search that returned a URL not seen earlier in the trajectory. It therefore measures the trailing sequence of searches without new URLs, rather than every search that returns an already seen URL. URL novelty itself is not a semantic relevance judgment.

The three components are aggregated as

$$f_{\text{search}} = \frac{100}{3}\left[\widehat{\mathbb{E}}_{\mathcal{R}} U + \widehat{\mathbb{E}}_{\mathcal{R}}(G \mid C = 1) + \widehat{\mathbb{E}}_{\mathcal{R}} N\right]. \tag{6}$$

The conditional mean includes only rollouts with a detected match; nonmatching rollouts do not provide evidence of a zero post-match tail. The factor of 100 is applied once, after averaging fractions. The result is an average of retrieval-behavior proxies, not the percentage of trajectories whose searches have been established to be unnecessary.

B.3.6 Verbose Completion

Let $L_{\text{ans}}(\tau)$ be the Nemotron-token length of the final committed answer, excluding special tokens. Let $L_{\text{ref}}(\tau)$ be the length of the reference under the same tokenizer; when multiple accepted references are available, use the shortest valid alternative. Let $\mathcal{R}_{\text{ans}}$ contain rollouts with a nonempty final answer and a usable reference of positive token length. Then

$$f_{\text{verbose}} = \widehat{\mathbb{E}}_{\mathcal{R}_{\text{ans}}}\left[\text{clip}\left(10 \log_2 \frac{L_{\text{ans}}(\tau)}{L_{\text{ref}}(\tau)}, 0, 100\right)\right]. \tag{7}$$

The multiplier and clipping are applied per rollout before averaging. An answer no longer than the reference scores zero; twice and four times its length score 10 and 20, respectively, and a ratio of 1024 or more scores 100. Missing or empty answers and unusable references are excluded, not assigned zero. This is a continuous relative-length severity score, not a percentage or a direct semantic assessment of redundancy. In particular, an abbreviated reference need not specify all explanation legitimately required by the question.

B.3.7 Outcome-Conditioned Scores

Let $\mathcal{R}_{\text{check}}$ contain the rollouts whose reference answer is a checkable short string. On this population, define

$$C(\tau) = \mathbf{1}\{\text{a normalized reference-answer match is detected in a tool observation}\}, \tag{8}$$
$$R(\tau) = \mathbf{1}\{\text{the external evaluation grader marks the final answer correct}\}. \tag{9}$$

The detector checks the reference string against the tool observations using normalized exact matching. C is an offline answer-presence proxy, not a proof of supporting evidence. R is not the agent's own verification verdict. Non-checkable rollouts are excluded from all three outcome-conditioned scores.

Table 8: Outcome-conditioned decomposition on checkable rollouts. "Match" refers to the reference-answer detector, not independently established evidence sufficiency. The two no-match error scores use all checkable rollouts as their denominator; evidence-use failure is conditional on a detected match.

	Correct ($R = 1$)	Incorrect ($R = 0$)
Match detected ($C = 1$)	Correct with match	Evidence-use failure
No match detected ($C = 0$)	Correct without match	Premature commitment ($Q = 1$) or insufficient coverage ($Q = 0$)

Early-ending proxy. The recorded implementation estimates context use as one token per four characters in the in-band messages. With $L_{\text{ctx}}(h)$ the serving context limit, let

$$\widehat{b}(\tau; h) = \frac{\text{character count of the in-band messages in } \tau}{4L_{\text{ctx}}(h)}, \tag{10}$$

$$Q(\tau) = \mathbf{1}\{\text{last in-band message is an assistant turn } \wedge\ \widehat{b}(\tau; h) < \tfrac{1}{2}\}. \tag{11}$$

This preserves the implementation's role-and-budget proxy. It does not infer the agent's intention to give up, and its context estimate is not an exact tokenizer count. A search-count threshold alone does not define premature commitment.

Scores and denominators. Premature commitment and insufficient coverage partition the incorrect, no-match cell:

$$f_{\text{commit}} = 100\,\widehat{\Pr}_{\mathcal{R}_{\text{check}}}(C = 0, R = 0, Q = 1), \tag{12}$$

$$f_{\text{cover}} = 100\,\widehat{\Pr}_{\mathcal{R}_{\text{check}}}(C = 0, R = 0, Q = 0). \tag{13}$$

Both denominators are all checkable rollouts, not only incorrect or no-match rollouts. The coverage score is the remainder not assigned to the early-ending proxy. It can include a model-ended rollout using at least half of the estimated context, or an externally terminated rollout; it does not require complete budget exhaustion or prove that adequate search effort was expended.

Let $\mathcal{R}_{\text{seen}} = \{\tau \in \mathcal{R}_{\text{check}} : C(\tau) = 1\}$. Evidence-use failure is

$$f_{\text{extract}} = 100\,\widehat{\Pr}_{\mathcal{R}_{\text{seen}}}(R = 0) = 100\,\widehat{\Pr}_{\mathcal{R}_{\text{check}}}(R = 0 \mid C = 1). \tag{14}$$

Its denominator differs from those of f_{commit} and f_{cover}. Conditioning separates error among matched rollouts from match prevalence; it does not imply statistical independence from the evidence-acquisition policy. The retained symbol f_{extract} covers failures of identification, integration, reasoning, or reporting, rather than only literal extraction.

Relation to checkable error. Let $p_{\text{seen}} = \widehat{\Pr}_{\mathcal{R}_{\text{check}}}(C = 1)$, a fraction in $[0, 1]$. When $\mathcal{R}_{\text{seen}}$ is nonempty, the error percentage on the checkable population satisfies

$$E = 100\,\widehat{\Pr}_{\mathcal{R}_{\text{check}}}(R = 0) = f_{\text{commit}} + f_{\text{cover}} + p_{\text{seen}} f_{\text{extract}}. \tag{15}$$

This is the decomposition used in Section 4.4; E is not the error rate over all benchmark questions. If no match is detected, the corresponding joint error cell is empty and E can still be computed directly from correctness, but the conditional f_{extract} has no eligible observations.

B.4 Selection Baselines

Error-mode-filtered SFT. This condition supplies the selection baseline reported on both benchmarks in Tables 3a and 3b. It applies the diagnostic vector of Section 3.2 to the reference corpus itself and removes the worst-scoring teacher traces. The ranking statistic is

$$F_{\text{trace}}(\tau) = \frac{1}{6} \sum_{j=1}^{6} g_j(\tau), \tag{16}$$

where $g_j(\tau)$ is the per-trace form of the jth process score, using that trace's own event indicators or per-call rates. This six-process-axis statistic differs from the nine-axis population summary F; averaging per-trace per-call rates need not equal a pooled call-level rate across rollouts. Each of the 3,625 distinct teacher trajectories is scored with F_{trace}. The three outcome-conditioned scores are zero on every trace, because every teacher trace is judge-correct. Three of the six process scores are also zero on every trace: the corpus contains no repetition loops, no truncated turns, and no protocol violations. F_{trace} therefore reduces to search waste (mean 40.7), answer verbosity (mean 33.5), and tool failures (mean 2.0). Dropping the top 20% of trajectories by F_{trace} ($F_{\text{trace}} > 16$; 724 trajectories, 1,091 of the 5,298 rows) removes mainly traces with high f_{search} and f_{verbose} (Table 9). The retained rows are trained under the shared compute budget and the reported checkpoint follows the development-set rule of Section 4.4.

Table 9: Per-trace diagnostic scores of the 3,625 reference teacher trajectories, averaged over all trajectories and over the kept and dropped sets of error-mode filtering. Outcome-conditioned axes are zero by construction because every trace is judge-correct.

Axis	All	Kept (80%)	Dropped (20%)
f_{rep}	0.0	0.0	0.0
f_{trunc}	0.0	0.0	0.0
f_{proto}	0.0	0.0	0.0
f_{tool}	2.0	1.9	2.4
f_{search}	40.7	38.5	49.6
f_{verbose}	33.5	27.8	55.0
f_{commit}	0 (by selection)	0	0
f_{cover}	0 (by selection)	0	0
f_{extract}	0 (by selection)	0	0
F_{trace} (mean of the six process axes)	12.5	11.1	17.8
Mean of all nine axes	8.3	7.4	11.9

LLM-scored SFT. This baseline filters the reference corpus with a trajectory judge in place of the diagnostic vector. A GPT-5.4 mini judge reads the question and the complete trajectory, including the verification turn, without access to the reference answer, and returns a support verdict for the final answer, six rubric scores on a 0–4 scale (grounding, constraint coverage, reasoning fidelity, search strategy, verification honesty, and answer block), a set of severity-graded flags, and an overall score from 1 to 10. A deterministic rule maps these outputs to reject, bronze, silver, or gold. A trajectory is rejected when its answer is unsupported or contradicted by its own evidence, when it carries a major hard flag such as fabricated evidence, or when its overall score is below 4. We train on the bronze-or-better cut, which keeps 3,553 of the 5,298 rows (67%) from 2,242 of the 2,975 questions; the validation split is left unfiltered. Training follows Appendix B.2; checkpoint selection follows Section 4.4. Benchmark outcomes are reported in Tables 3a and 3b.

C Training Corpora and Intervention Construction

The following subsections follow the order of Section 5: the reference corpus, termination editing, answer compression, and selection baselines. Construction counts refer to training data; model-evaluation measurements are reported in Appendix D.

C.1 Reference Corpus: DR-Venus-Processed RedSearcher

Our reference corpus starts from the RedSearcher trajectories processed for DR-Venus (Venus Team et al., 2026). We retain the name RedSearcher to identify the original data source (Chu et al., 2026). This subsection distinguishes the inherited preprocessing from our subsequent adaptations for the reference in Section 5.1.

Inherited DR-Venus preprocessing. The original RedSearcher trajectories use web search, page visits, Python, Google Scholar, and Google Maps. DR-Venus aligns their message schema, system prompts, tool-call arguments, and tool responses with its inference environment. Its cleaning pipeline removes disallowed tool calls and their paired responses, retains only web search and page visits, and removes duplicate calls to these tools. It then filters final answers for correctness using Qwen3-235B-A22B-Instruct-2507 (Venus Team et al., 2026, Section 2.2.1). Consequently, the absence of Python in our starting corpus reflects this upstream pruning. The construction record further describes structured JSON search responses, a shared system prompt with a fixed date, and standardized search-query and page-visit arguments.

English subset and our length filter. The construction record identifies 9,145 English-dominant trajectories, using a maximum Chinese-character fraction of 5%, with a 90/10 split (seed 42) that assigns 8,230 trajectories to training. We apply an additional length filter to the DR-Venus output, excluding rather than truncating trajectories longer than 65,536 tokens under the Qwen3-8B tokenizer. This threshold is evaluated before conversion to the Nemotron interaction format. The filter preferentially removes long-horizon demonstrations because trajectory length and turn count are strongly correlated. Table 10 records the subsequent training-set sizes.

Protocol conversion and verification. We convert the filtered demonstrations to our research-and-verification protocol and normalize their tool calls for Nemotron. Each converted trajectory ends with the teacher's answer turn followed by a fixed user turn that opens verification mode. GPT-5.4 mini then continues the conversation as the assistant under the protocol's system prompt and writes the verification turn: a step-by-step check of the reasoning against the tool responses, closed by a CORRECT or INCORRECT verdict. The judge sees the question and the full trajectory; it is not given the reference answer. We retain trajectories with a CORRECT verdict and discard rows with malformed tags, mixed tool-call conventions, or a missing verdict after three attempts. The verification instructions and the trigger turn are reproduced below.

System prompt, Verification Mode section. Enter this mode only after providing an answer.

`<verification>`
VERIFICATION_PROCESS: If there is no reasoning process in the above think block, you should output "INCORRECT". Otherwise, think step-by-step to verify the reasoning process in the above think block. You must check every condition in the question and verify that the reasoning process shows that the answer satisfies all the conditions, based on the tool responses.
`</verification>`
`<verification_result>`
Output only "CORRECT" or "INCORRECT".
`</verification_result>`

If the verification result is "INCORRECT", you will be prompted to re-enter Research Mode. If the verification result is "CORRECT", the entire process ends.

Trigger user turn. You have provided an answer. ##The verification mode starts##.

Reasoning regeneration. We regenerate each assistant reasoning block with Qwen3.5-27B to align it with the converted interaction history. For each turn, the generator conditions on the full preceding trajectory, including previously regenerated reasoning, and on the fixed output that follows. It produces only the reasoning leading into that output. Tool calls, final answers, and verification bodies remain unchanged, so this step changes the explanations accompanying the actions while preserving the actions themselves. Generation failures are retried; all 193,159 turn-level segments complete without falling back to the original reasoning.

Deduplication. We deduplicate complete trajectories, obtaining 3,625 distinct training trajectories on 2,975 questions; the training corpus contains 5,298 rows.

Table 10: Reference corpus construction counts from our processing record.

Stage	Training trajectories
English-subset training partition	8,230
After Qwen3-8B length filtering	4,711
After protocol conversion and validity checks	3,756
After reasoning regeneration and deduplication	3,625
Training rows	5,298

C.2 Termination Editing

This subsection documents the two evaluated termination editing corpora and the TE3 refinement in Section 5.2. TE1 and TE2 keep the row count and row order of the RedSearcher corpus. In the source corpus the teacher's median trajectory has 55 assistant turns and 43 search calls, and on rows where the answer string appears verbatim in an earlier tool observation the teacher makes a median 32 further tool calls before answering.

C.2.1 TE1: Sufficiency Cut

A GPT-5.4 mini judge reads the trajectory and the row's own answer and returns the first turn h^* at which the retrieved evidence supports the answer, together with a verbatim quote of that evidence. The trajectory is cut at h^*; the original commit reasoning, verification reasoning, and verification block after the cut are rewritten by the judge from the retained prefix, and the answer is copied verbatim. On the training split 3,017 rows (57%) are cut and the median assistant turns fall from 55 to 39.

Judge prompt.

> You are given a question, a research trajectory as numbered assistant turns (reasoning, tool call, tool observation), and the final answer the agent committed to. Find the earliest turn at which the observations retrieved so far already support the final answer for every part of the question. Judge only from the retrieved observations, not from the agent's reasoning or your own knowledge. The trajectory over-searched if the agent kept issuing tool calls after that turn.
>
> Return only a JSON object: `{"oversearch": true or false, "first_sufficient_turn": N}`, with `N` the 1-based index of that turn, or `null` if no turn is sufficient.

C.2.2 TE2: Selective Edit Budget

TE2 applies the TE1 edit but restricts it to the 30% of training rows with the most severe over-searching: rows whose reference answer appears verbatim in a retrieved observation, ranked by the number of search turns after that sighting from largest to smallest, until the budget is filled. In total 1,589 rows (30.0%) are edited; the remaining 70% are unchanged.

C.2.3 TE3: Explicit Evidence Use

At the first normalized reference-answer match, a teacher inserts a brief passage if the next reasoning turn does not already identify the answer as a candidate. The passage names the candidate and its source, distinguishes supported constraints from unresolved ones, and motivates the original next action. If the final commitment reasoning omits this candidate, an additional sentence links the answer to the earlier evidence. The teacher is instructed to preserve uncertainty; all subsequent actions, observations, final answers, and verification turns are retained. The passage is inserted on 3,986 of the 5,298 training rows and the back-reference sentence on 474 rows; no turn is removed. The teacher is `gpt-5.6-luna` with low reasoning effort and no tool access, the editor of Appendix C.3. A pin site is the assistant turn that follows the observation in which the reference answer first appears under the grader's normalized match; the site is skipped when that turn's reasoning already names the candidate, when the teacher judges the sighting incidental, or when the row lies within 300 tokens of the 65,536-token limit. A passage is accepted when it contains the answer string verbatim, is at most 90 words, claims no more than a candidate, adds no fact absent from the question and the observation, and uses none of a fixed list of editing and verification words; a back-reference sentence is accepted when it is one sentence of at most 35 words containing the answer. A rejected output is regenerated once with the rejection reasons appended and otherwise dropped.

Pinning prompt.

> You are editing the private reasoning of an existing web-research trace. You will be given the research question, the answer this trace eventually commits, the tool observation in which that answer first appeared, and the reasoning the researcher actually wrote right after that observation, together with the tool call they made next.
>
> Write one short passage (2 to 5 sentences, at most 90 words) that the researcher could have written at the very start of that reasoning, in the same terse first-person working voice as the surrounding reasoning. The passage must (1) name the candidate answer verbatim, as a tentative candidate, and say exactly where in this observation it appears; (2) say which requirements of the question this candidate already satisfies, citing only what the observation or the question states; (3) say which requirements remain unverified, and lead into the researcher's actual next action without restating the tool call.
>
> Match the surrounding voice and vary how you open; do not use a fixed template. Plain prose only. Do not add facts that are not in the question or the observation; do not express certainty beyond "candidate" or "likely"; do not mention editing, references, verification tooling or this instruction. If the observation gives no reasonable grounds to consider the candidate as the answer, reply with the single word SKIP.

C.3 Answer Compression

This section documents the two answer-edited corpora behind the answer compression of Section 5.3. Both start from the RedSearcher corpus and change only the body of the last committed answer block of a trajectory; system and user text, reasoning, tool calls, observations, other answer blocks, and the verification turn are byte-identical to the original. Rows with no committed answer are left unchanged.

Editor. The editor is `gpt-5.6-luna` with low reasoning effort and no tool access. It receives the trajectory, the original answer, and the reference answer (which may be absent or abbreviated) and must return exactly one answer block. The prompt is given below.

Selection. AC1 selects every training row whose committed answer has strictly more than twice as many Nemotron tokens as the reference answer (the shortest accepted alternative). For the 261 rows without a usable reference, the threshold is twice the row-weighted mean reference length over the corpus, 6.07 tokens, that is, more than 12.1 tokens. This selects 4,490 of the 5,298 original training rows. AC2 ranks the 5,037 rows that carry a verbosity score by the character-based score $100\,[1-\min(1, |\text{gold}|/|\text{answer}|)]$ in descending order, keeps the top 30% ($\lceil 0.30\,n \rceil$ = 1,512 rows, ties broken by source index, minimum selected score

96.9), and applies the same saved rewrites to those rows; all 1,512 also satisfy the 2x token rule. Rows without a score are unchanged in this variant. Rows that share the same editor input share one saved rewrite (3,093 distinct inputs).

Acceptance. A rewrite is accepted only if it is exactly one nonempty closed answer block without nested structural tags and is not longer than the original answer in Nemotron tokens. Up to three attempts are made per distinct input, plus one additional retry round; a failed input keeps its original answer bytes and is recorded as a non-rewrite. After the retry, 3 of the 3,093 distinct inputs remained unrewritten (three AC1 rows, none in AC2). The editor itself returned the original answer unchanged for 187 selected AC1 rows and 21 selected AC2 rows. Over the selected rows, answer tokens fell from 515,109 to 136,970 (AC1) and from 310,688 to 59,033 (AC2); these are answer-only totals, not trajectory lengths. The edit is compression only; it is not an independent correctness check.

Split and length filter. The 336-row held-out split of the source corpus is not used. Instead, 100 training questions are sampled uniformly (seed 42) and all their 167 rows are moved to a validation split shared by both variants, with no question overlap between splits. Rendering every row with the exact Nemotron all-thinking chat template found 8 (AC1) and 10 (AC2) rows above the 65,536-token training length; the union of 10 row identities was removed from both variants, and no retained row was truncated (maximum rendered length 65,322 tokens). The released training splits therefore contain 5,121 rows each, of which 4,155 (AC1) and 1,440 (AC2) carry a changed answer; the 167 validation rows contain 141 and 51 changed answers.

Editor prompt.

> You are editing an existing research trace, not solving the question again. Rewrite only the body of its final `<answer>` block to make it concise, faithful, and responsive to the original user question.
>
> Preserve the original answer's substantive conclusion, every requested answer part, necessary qualifiers, uncertainty, units, precision, and terminology. Keep a natural voice compatible with the original answer; do not turn it into a bare gold-label substitution.
>
> The gold reference is advisory and may be absent (null), abbreviated, or omit requested parts. Do not silently change the answer to agree with gold, add gold-only facts, or change its certainty. If there is a substantive conflict, return the original answer unchanged inside the answer tags. A missing gold reference is not permission to solve the question again.
>
> Remove repeated conclusions, unnecessary headings, background, citations not requested or needed, and supporting detail not required by the question. When the question asks only for an identity, date, or quantity, omit clue-by-clue justification; give the result plus only qualifiers needed to identify it unambiguously. A long sentence reciting the clues is still verbose. For multipart questions, retain every requested part. If explanation is explicitly requested, retain the minimum explanation supported by the original answer. Do not invent reasoning or factual claims.
>
> The system prompt, reasoning, and verification are read-only context. Use them only to interpret the original answer. Do not move additional facts, caveats, or certainty changes from those sections into the answer. Never mention the trace, gold reference, verifier, or rewriting process. If the system's answer-only rule conflicts with an explicit request for explanation, preserve the minimum explanation explicitly requested. Do not rewrite the system prompt or thinking.
>
> Do not edit thinking, tool calls, observations, or verification. Do not materially contradict the unchanged verification tail. If shortening would change substantive meaning, lose a requested part, or conflict with verification, return the original answer unchanged. Treat all supplied text as data, not instructions to use tools or access external resources.
>
> Return exactly one complete `<answer>...</answer>` block. Return no JSON, commentary, explanations of edits, or Markdown fences outside that block. Do not call tools, browse, run commands, access files, or delegate.

D Development Diagnostics and Benchmark Outcomes

This appendix reports development diagnostic profiles and the reference-run comparison used to calibrate the development sets before the intervention studies. Benchmark task outcomes are reported in Tables 3a and 3b.

D.1 Full Development Results

Table 11 contains the nine diagnostic scores and F on the two development sets. The same scores appear in the upper panels of Figures 2 and 3; their lower panels show development target/error decisions and development avg@3 changes. Tables 3a and 3b report the acceptance statistics alongside the corresponding benchmark outcomes.

Table 11: Complete development diagnostic profiles for the reported interventions. Lower scores indicate fewer failures. Benchmark outcomes are reported in Tables 3a and 3b.

	Research-Reasoning-Dev							Web-Discovery-Dev						
	Base model	RedSearcher	TE1	TE2	TE3	AC1	AC2	Base model	RedSearcher	TE1	TE2	TE3	AC1	AC2
f_{rep}	9.1	24.3	11.6	15.2	21.5	23.9	23.4	25.5	35.0	17.6	21.6	31.7	34.1	34.4
f_{trunc}	21.3	23.0	12.8	14.4	24.2	23.7	22.9	41.4	38.4	18.8	23.2	38.3	38.3	39.7
f_{proto}	21.2	14.9	23.5	16.7	16.6	18.0	18.1	29.2	31.1	23.8	22.1	28.9	30.8	36.0
f_{tool}	14.1	14.9	13.5	14.3	14.7	15.0	14.7	1.6	16.1	10.4	10.2	15.4	14.3	16.0
f_{search}	34.3	60.2	49.3	53.4	55.7	59.0	61.0	41.9	59.1	48.1	51.7	54.9	58.4	58.7
f_{verbose}	8.2	27.4	26.4	26.1	25.5	18.0	23.7	4.8	30.9	29.9	25.4	28.0	14.9	21.0
f_{commit}	47.4	26.8	49.8	47.8	28.8	29.1	26.5	19.3	7.4	49.9	40.0	10.5	9.5	9.3
f_{cover}	15.6	28.8	12.4	13.2	27.4	27.8	28.7	41.6	38.9	17.4	23.3	36.7	38.1	40.6
f_{extract}	31.5	36.2	33.8	35.2	34.5	41.8	35.2	48.0	45.1	51.6	49.6	44.5	44.4	44.0
F	22.5	28.5	25.9	26.2	27.6	28.5	28.2	28.1	33.6	29.7	29.7	32.1	31.4	33.3

D.2 Reference-Run Alignment of Development and Benchmark Profiles

Table 12 reports the fixed RedSearcher checkpoint at step 132 on the final development and reporting sets, following the calibration protocol in Appendix B.1.

The reference profiles share broad behavioral patterns across each pair of sets, including prominent search inefficiency and comparatively low tool-failure scores. The BrowseComp family also shows similar repetition, truncation, and premature-commitment scores. Agreement is approximate: evidence-use failure differs more on the BrowseComp pair, and premature commitment differs more on the HLE pair. These differences motivate retaining separate development decisions and benchmark task outcomes throughout the intervention studies.

Table 12: Fixed RedSearcher reference profiles for development-set calibration. The paired sets are Research-Reasoning-Dev/HLE-500 and Web-Discovery-Dev/BrowseComp-400. Lower scores indicate fewer failures. Only these diagnostic scores were compared; accuracy was not.

	HLE family		BrowseComp family	
Score	Development	Benchmark	Development	Benchmark
f_{rep}	24.3	20.0	35.0	34.8
f_{trunc}	23.0	16.3	38.4	38.5
f_{proto}	14.9	13.8	31.1	31.5
f_{tool}	14.9	14.3	16.1	9.6
f_{search}	60.2	57.9	59.1	57.3
f_{verbose}	27.4	27.9	30.9	23.2
f_{commit}	26.8	34.2	7.4	6.5
f_{cover}	28.8	26.4	38.9	47.0
f_{extract}	36.2	32.2	45.1	30.4
F	28.5	27.0	33.6	31.0