LEARNING TO SOLVE AND EXECUTE: TRAJECTORY REFINEMENT FOR HARNESS-AWARE AGENTIC FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-model agents solve difficult real world tasks under tool execution environments and harness constraints. Supervised fine-tuning, usually called SFT, typically uses demonstrations from stronger teachers, yet these may not cover the states encountered by students. Revising student trajectories offers a way to address this mismatch, but changing an action can invalidate subsequent observations. We introduce Repair SFT, which constructs offline supervision by asking a black-box teacher to correct causal errors in student trajectories while retaining valid behavior. Revised tool calls are executed in the environment to regenerate the affected continuation, keeping actions and observations consistent. Hybrid SFT selects between repaired trajectories and direct teacher demonstrations using training-task outcomes and agent roles. We evaluate both methods with Qwen2.5-14B and Qwen3.5-9B under a shared memory harness. On the Qwen3.5 experiment, Repair SFT reduces step-budget exhaustion from 44.1% to 14.3%, while Hybrid SFT raises MuSiQue F1 from 20.24% to 49.59%. On Qwen2.5, Hybrid SFT improves MuSiQue and Qasper F1 over Direct SFT and achieves the highest IIRC F1 among the three SFT variants. These findings support student-conditioned trajectory repair and selective supervision for agentic SFT.

1 INTRODUCTION

Agentic supervised fine-tuning (Agentic SFT) trains language models on trajectories of reasoning, tool calls, observations, and answers (Yao et al., 2023; Chen et al., 2023; Zeng et al., 2024). A common approach, which we call *Direct SFT*, uses demonstrations from a stronger teacher. Students must learn task solving alongside execution protocols and stopping conditions. These demands can diverge: format following and reasoning exhibit different learning dynamics (Chen et al., 2024), and agent benchmarks expose failures in multi-step execution (Liu et al., 2023; Yao et al., 2024). Answers must be produced within the interaction protocol and budget, while normal termination does not ensure correctness. Evaluating trajectory supervision therefore requires both task outcomes and execution behavior.

Teacher demonstrations reflect the teacher decisions, whereas deployment contexts depend on the student actions. This distribution mismatch can compound errors (Ross & Bagnell, 2010; Ross et al., 2011). Supervision on learner-induced states, as in DAgger and on-policy distillation, addresses this problem (Ross et al., 2011; Agarwal et al., 2024). Agentic correction has an additional constraint: changing a tool call can invalidate every subsequent observation. A revised trajectory must keep actions and evidence consistent.

We introduce *Repair SFT*, which asks a black-box teacher to revise student attempts by correcting the earliest causal error while retaining valid behavior. Revised actions are executed in the environment to regenerate their consequences. The resulting offline targets require neither teacher probabilities nor a shared tokenizer. *Hybrid SFT* selects between repairs and direct teacher demonstrations using training-task outcomes and agent roles. We test whether student-conditioned revisions improve supervision and whether selecting from both sources further improves the policy.

We compare these sources in a fixed *memory harness* that separates an append-only history from editable working context. Motivated by limitations of long-context models (Liu et al., 2024), its Step

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.

Tree Agent uses step summaries to guide four tools for message processing, hiding, restoration, and rollback with environment replay. The students are Qwen2.5-14B and Qwen3.5-9B, with matched training instance counts and optimizer schedules within each model. Hybrid SFT also changes which source is selected, how roles are balanced, and how many tokens are supervised, so its comparison measures these factors jointly (Sections 2.3 and 3.2).

We evaluate retrieval-based QA, WebShop, ALFWorld, and ScienceWorld, with IIRC and BFCL testing unseen task families. Repair SFT achieves the highest recorded completion on both students and reduces Qwen3.5 step-budget exhaustion from 44.1% to 14.3%. Hybrid SFT leads the trained variants in MuSiQue and Qasper F1 and WebShop reward on both students. Relative to Direct SFT, it raises MuSiQue/Qasper F1 from 20.24%/9.48% to 0.4959/0.2580 on Qwen3.5 and from 28.99%/17.33% to 36.36%/22.96% on Qwen2.5. Its QA advantage extends to Qwen2.5 IIRC transfer, reaching 20.97% F1 versus Direct SFT 14.35%. On Qwen3.5, both answer coverage and F1 on answered questions improve. Together, these findings demonstrate the value of student-conditioned corrections for execution and selective supervision for task performance.

We make three contributions:

- **Student-conditioned trajectory supervision.** Repair SFT constructs offline targets through black-box teacher revisions and real environment execution. Hybrid SFT selects complementary targets from direct demonstrations and repairs by task outcome and agent role.
- **A memory harness for context management.** A Step Tree Agent uses four tools to manage working context and roll back interactions through replay, guided by step summaries while preserving the original history.
- **Evaluation across two students and diverse tasks.** Matched comparisons on QA and interactive benchmarks distinguish improvements in task performance from changes in answer coverage and stopping behavior, documenting the execution gains of Repair SFT and the QA and WebShop improvements of Hybrid SFT over Direct SFT.

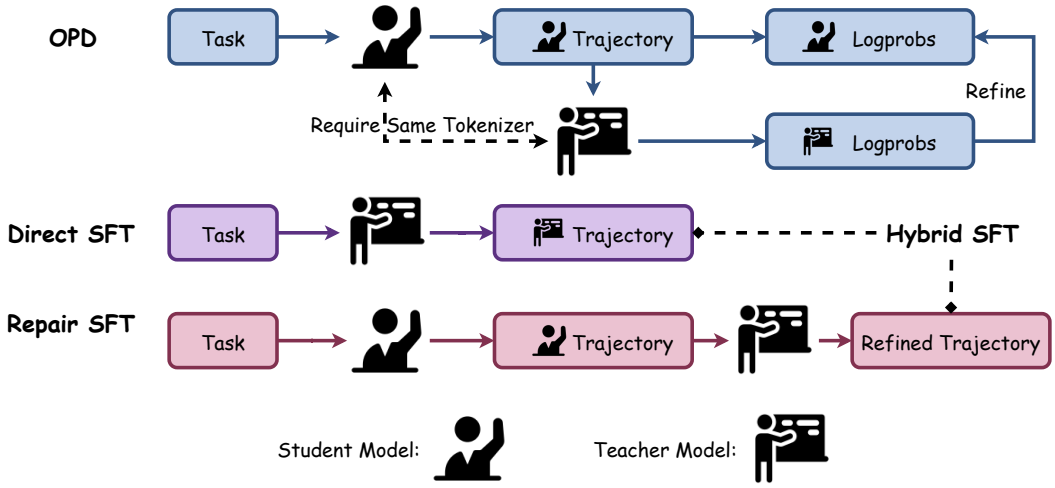

Figure 1: Supervision sources. Direct SFT uses teacher trajectories; Repair SFT uses teacher revisions of student attempts with real tool execution; Hybrid SFT selects from both. White-box OPD matches teacher token distributions on student rollouts; our SFT variants require only generated responses.

2 METHOD

We vary how training trajectories are constructed while keeping the SFT objective fixed. Direct SFT supplies teacher demonstrations, Repair SFT revises student attempts, and Hybrid SFT selects

between them. Construction occurs before optimization, so the resulting records can be inspected and retokenized independently of the teacher.

2.1 AGENTIC SFT AND SUPERVISION SOURCES

A trajectory is $\tau = ((h_t, a_t, o_t))_{t=0}^{T}$: the agent generates an assistant response a_t from visible context h_t, and tool execution returns observation o_t ($\emptyset$ when no tool is called). The context need not contain the full environment state. For each training example (x_i, y_i), x_i contains the preceding context and role prompt, and y_i is the target assistant span. All variants use the same masked SFT objective (Wei et al., 2022; Ouyang et al., 2022):

$$\mathcal{L}(\theta) = -\sum_{i} \sum_{t \in \mathcal{I}(y_i)} \log p_\theta(y_{i,t} \mid x_i, y_{i,<t}),$$

where $\mathcal{I}(y_i)$ indexes target assistant tokens; input context and tool observations are excluded from the loss.

Direct SFT uses trajectories executed by a stronger teacher from the task specification and tool schemas, without a student draft (Chen et al., 2023; Zeng et al., 2024). **Repair SFT** instead conditions the teacher on student attempts, and **Hybrid SFT** selects from both sources (Figure 1). Direct SFT contexts follow teacher decisions; Repair SFT starts from states encountered by the student, where corrections can address its particular strategy or execution errors. Editing an action can invalidate later observations, so useful supervision requires regenerating the dependent continuation rather than only rewriting the recorded text. These methods construct offline response targets and do not depend on a particular memory architecture. Unlike the white-box OPD path illustrated in the figure, they require no teacher probabilities or aligned vocabularies; black-box OPD formulations also exist (Ye et al., 2025).

2.2 REPAIR SFT: STUDENT-CONDITIONED TRAJECTORY CORRECTION

For each source task, we collect a student draft and give a black-box teacher the task, tool interfaces, and draft. The teacher constructs a corrected trajectory through three steps.

Locate and revise the causal error. The teacher is instructed to correct the earliest causal error and its dependent continuation while retaining valid reasoning, tool choices, and action order. Already-correct drafts receive only restrained clarification or compression. These instructions limit edit scope without guaranteeing a byte-identical prefix. The aim is to retain the useful decisions of students while changing the decisions that led the attempt astray. A fully rewritten demonstration could solve the task but discard this connection to the student's behavior. Retention is a construction principle, however, rather than a guarantee that the repaired trajectory remains on the original state distribution of student.

Regenerate the consequences. When a tool call changes, the environment is reset and the retained prefix replayed as needed. The revised continuation uses actual tool responses; teacher-written substitute observations are rejected. For example, replacing a search query requires executing it and grounding later reasoning in the newly returned documents. Pairing a revised query with the original search results would train the student on evidence that the revised action never produced. Replay prevents this inconsistency; task-outcome checks separately assess whether the executable revision solves the task.

Verify and extract targets. Candidates are checked for execution integrity, task outcome, and unnecessary rewriting. Failures can trigger retries with generic verification feedback. Reference answers, labels, and numerical scores remain hidden from the repair teacher, although a correctness indicator may guide edit scope. Accepted assistant responses become SFT targets; drafts and repair guidance are excluded from the student runtime inputs.

Targets use the native chat template, tokenizer, and tool-call format. Training then requires no teacher calls or environment interaction. The drafts are frozen before SFT, and revised continuations may reach new states. Thus Repair SFT provides student-conditioned offline supervision, without

DAgger iterative dataset aggregation or continuous sampling from the evolving student (Ross et al., 2011).

2.3 Hybrid SFT: selecting complementary supervision

Repairs preserve useful student behavior, but need not supply the strongest solution to every task. Direct SFT trajectories can introduce strategies absent from a draft, while repairs show how to correct the student own attempt. Hybrid SFT retains access to both forms of supervision through target selection.

Hybrid SFT selects between Direct SFT and Repair SFT candidates using the outcome metric for the corresponding training task. Scores are compared within tasks, never across benchmarks. Role-specific source rules are detailed in Section 3.1 and Appendix F.

The selected records undergo a separate SFT run with the same objective; Hybrid SFT combines supervision rather than model weights. Its comparison therefore measures source access, selection, and role balancing jointly. Without a random-mixture control matched for supervised token budget, gains cannot be attributed to source mixing alone.

3 Experiments

We compare Direct SFT, Repair SFT, and Hybrid SFT with the original instruction model (Raw) under a fixed memory harness, matching architecture, training instance counts, and optimizer schedules within each student.

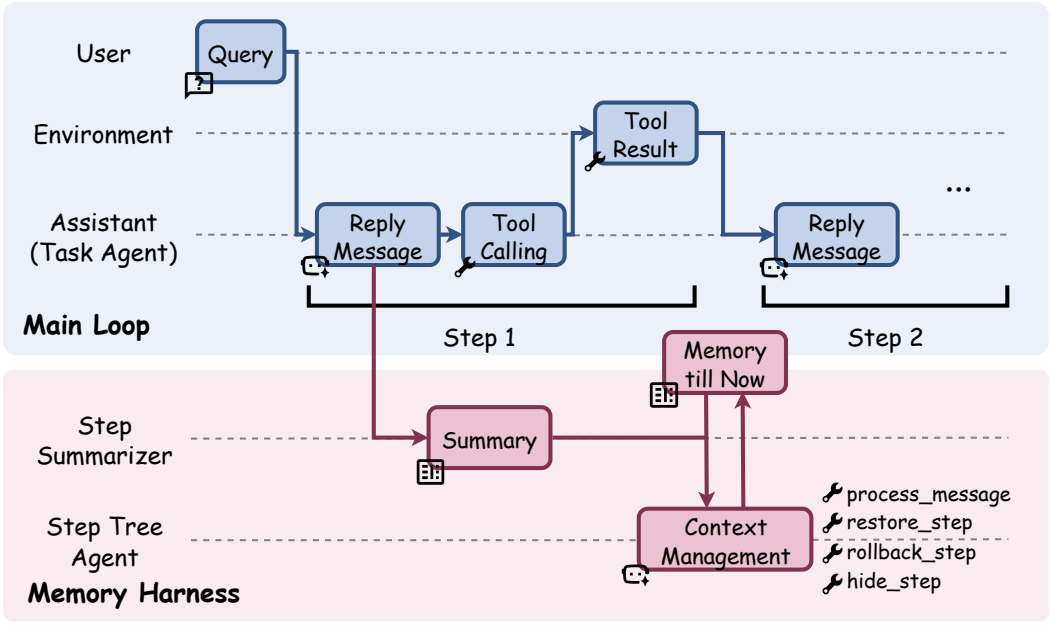

Figure 2: Memory harness. The Task Agent executes tasks, the Step Summarizer summarizes turns, the Step Tree Agent manages context through four tools, and the Message Processor transforms selected content. Original messages remain in the canonical history.

3.1 Memory harness

Our memory harness separates canonical history from editable working context (Figure 2). Its four roles share the student model with separate prompts and contexts: the Task Agent executes task tools, the Step Summarizer summarizes turns, the Step Tree Agent manages context, and the Message Processor transforms content. Guided by step summaries, the Step Tree Agent can call four context-management tools: `process_message` invokes the Message Processor to rewrite a message active

view; `hide_step` removes an older step from active context; `restore_step` restores a step's raw content; and `rollback_step` rewinds the active branch and environment through reset and replay, committing only if replay succeeds. These operations preserve canonical history. Repair SFT also applies beyond this architecture; here we compare supervision sources within the same harness.

For Hybrid SFT, Task Agent targets use the higher-scoring trajectory; ties prefer fewer task-tool calls, then Repair SFT for WebShop and Direct SFT otherwise. Step Summarizer targets follow that trajectory actions. Message Processor targets come from Repair SFT, while Step Tree Agent targets combine both sources, balancing memory actions and no-change decisions (Appendix F).

3.2 Training data and configuration

We fine-tune Qwen2.5-14B-Instruct (Yang et al., 2025) and Qwen3.5-9B (Qwen Team, 2026) on 2,000 source tasks (1,900 training, 100 validation). Half come from HotpotQA, 2WikiMultiHopQA, and Qasper (Yang et al., 2018; Ho et al., 2020; Dasigi et al., 2021); the rest come from ALFWorld, WebShop, and ScienceWorld. GPT-5.6 supplies Direct SFT trajectories and repairs. Each student has its own draft and repair archive with matched task identities and splits.

Each variant uses 30,396 role-conditioned instances, one epoch of full-parameter BF16 SFT with DeepSpeed ZeRO-3 (Rajbhandari et al., 2020), learning rate 10^{-5}, global batch size 32, and 950 optimizer steps. Context limits are 32,768 tokens for Qwen2.5 and 49,152 for Qwen3.5. Supervised token counts and target diversity differ. In addition, 100 Direct SFT source trajectories received hidden answers or verified actions as teacher assistance; this guidance was not serialized into student inputs. These cases are distinct from the 100-task validation split. Source provenance, role composition, supervised token budgets, and remaining training settings are in Appendices E and F.

3.3 Evaluation datasets

Both students cover MuSiQue dev (2,417 items) (Trivedi et al., 2022), Qasper dev (1,005) (Dasigi et al., 2021), WebShop test (500) (Yao et al., 2022), ALFWorld valid-seen/unseen (140/134) (Shridhar et al., 2021), and ScienceWorld test (1,819) (Wang et al., 2022). Aggregate execution statistics use the original run suites: 4,196 non-ScienceWorld items for Qwen3.5 and all 6,015 for Qwen2.5. These aggregates support comparisons within each student; their different task compositions preclude direct comparisons between students. Task scores are never pooled across benchmarks. MuSiQue is absent from the training families; the other benchmarks use held-out items. Qasper training and evaluation identifiers were checked for overlap.

The quick eval set contains 100 items each from MuSiQue, Qasper, WebShop, and ScienceWorld, plus 50 from each ALFWorld split. Its 400 non-ScienceWorld items overlap the complete suite, so it is not an independent replication. Additional Qwen2.5 transfer evaluations use IIRC dev (1,301 questions, 430 passages) (Ferguson et al., 2020), with ten linked articles missing from the local corpus, and 1,000 selected BFCL multi-turn tasks (200 per category) scored by the official checker (Patil et al., 2025).

3.4 Evaluation protocol

Primary task outcomes are full-split MuSiQue and Qasper F1. We also report QA exact match, WebShop reward and success, ScienceWorld environment points, ALFWorld success, and official BFCL accuracy. Task scores use planned denominators. QA without a completed answer scores zero; interactive tasks retain saved outcomes, including negative ScienceWorld penalties. Qasper evidence F1 is unavailable because runs emit no evidence paragraphs. Planned-denominator QA F1 is the product of answer coverage and mean F1 on answered items with completed status. Reporting both terms distinguishes missing answers from low-scoring answers. Conditional F1 alone can favor a system that answers only a small subset of questions. Recorded completion C denotes runner status, not task success, and can include budget exhaustion. Budget exits are identified by `terminationReason=max_steps`, independently of completion status; assistant-ended episodes use `assistant_without_tool_call`. Definitions and status semantics are in Appendices F and B.

Evaluations use temperature 0, top-$p = 1$, matched prompts and tool schemas, and Task Agent budgets of 12 steps for QA, 40 for WebShop, and 50 for ScienceWorld and ALFWorld. BFCL allows 20 steps per user turn and 2,048 output tokens per generation. Memory calls consume no additional Task Agent steps. One step comprises a Task Agent generation and any resulting tool execution; rollback does not replenish the budget. Thus matched step limits do not equalize total model calls or tokens and do not establish cost efficiency. Each condition uses one checkpoint and retained terminal records, including resumes and retries. Qwen3.5 SFT comparisons report 95% intervals from 10,000 paired bootstrap resamples, clustering Qasper by paper. These intervals condition on retained outcomes and exclude training-seed and retry variation; other comparisons are descriptive. Historical execution limits and consolidation rules appear in Appendix E.

Table 1: Complete-split results for the three trained variants. QA F1, WebShop reward, ALFWorld success, C, and Budget are scaled by 100; ScienceWorld uses signed environment points and Steps is a mean count. Bold/underline mark the best/second-best distinct values within each model, with ties retained; rankings are descriptive.

	In-domain				OOD	Execution		
Source	Qasper F1↑	SciWorld Score↑	WebShop Reward↑	ALF Success↑	MuSiQue F1↑	C↑	Steps↓	Budget↓
Qwen2.5-14B								
Direct SFT	17.33	**78.95**	65.55	**95.99**	<u>28.99</u>	83.18	<u>13.15</u>	<u>21.38</u>
Repair SFT	<u>20.55</u>	63.45	<u>66.84</u>	<u>88.69</u>	12.34	**98.35**	**11.33**	**10.32**
Hybrid SFT	**22.96**	<u>73.05</u>	**67.48**	**95.99**	**36.36**	<u>84.12</u>	13.33	22.96
Qwen3.5-9B								
Direct SFT	9.48	**79.18**	<u>69.83</u>	**95.99**	20.24	45.40	9.73	44.11
Repair SFT	<u>21.63</u>	32.87	67.90	88.32	<u>48.22</u>	**86.03**	<u>6.50</u>	**14.28**
Hybrid SFT	**25.80**	<u>60.77</u>	**72.35**	<u>95.26</u>	**49.59**	<u>83.29</u>	**6.20**	<u>16.73</u>

C is recorded completion, not success. ALF pools 274 games. Qwen2.5 execution uses 6,015 items; Qwen3.5 uses 4,196, excluding ScienceWorld. Steps omit failed records that stored no count.

4 RESULTS

4.1 TASK PERFORMANCE ACROSS BOTH STUDENTS

Table 1 aligns both students across the same tasks. Hybrid SFT has the highest QA F1 among the SFT variants: relative to Direct SFT, MuSiQue/Qasper F1 rises from 20.24%/9.48% to 49.59%/25.80% on Qwen3.5, and from 28.99%/17.33% to 36.36%/22.96% on Qwen2.5. Repair SFT also improves Qwen3.5 QA, but Qwen2.5 Repair SFT has the lowest MuSiQue F1 (12.34%). Raw remains strongest on MuSiQue for both students (51.42%/43.10%), while Hybrid SFT exceeds Raw on Qasper.

Hybrid SFT also achieves the highest WebShop reward among the trained variants on both students. Direct SFT leads WebShop success and ScienceWorld, as well as pooled ALFWorld success on Qwen3.5 and ties Hybrid SFT on Qwen2.5. All SFT variants improve interactive scores over Raw. ScienceWorld retains signed terminal penalties, which produce negative Raw means (Appendices C and D).

For Qwen3.5, Hybrid SFT–Direct SFT differences are 0.2935 MuSiQue F1 (95% CI [0.2746, 0.3125]) and 0.1633 Qasper F1 ([0.1479, 0.1795]). The closer Hybrid SFT–Repair SFT MuSiQue difference, 0.0137 ([-0.0045, 0.0315]), and Hybrid SFT–Direct SFT WebShop reward difference, 0.0252 ([-0.0020, 0.0547]), do not establish an advantage under the conditional bootstrap protocol. Figure 3 places Qwen3.5 MuSiQue F1 alongside suite-wide completion; these summarize different item sets and task weights.

4.2 ANSWER COVERAGE AND TERMINATION

Repair SFT yields the highest recorded completion for both students. On Qwen3.5, Repair SFT and Hybrid SFT reduce Direct SFT budget-exit rate from 44.1% to 14.3% and 16.7%, and mean steps

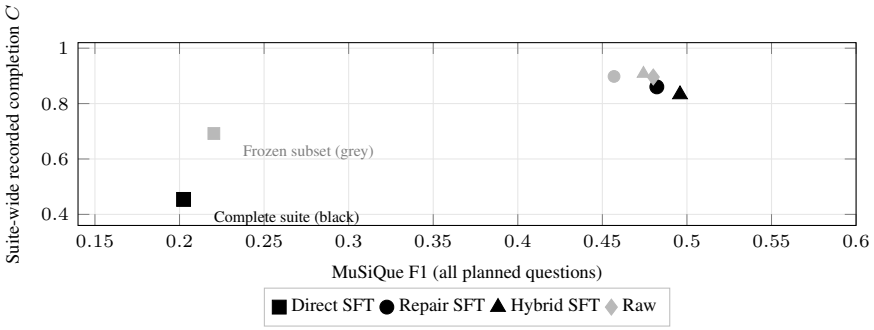

Figure 3: MuSiQue F1 and suite-wide recorded completion for Qwen3.5-9B. Shapes identify SFT variants; black markers show complete-split results and grey markers the frozen subset, including Raw. The horizontal axis uses 2,417 MuSiQue questions for black markers and 100 for grey markers. Each point pairs two aggregate metrics; completion is not QA coverage or environment success.

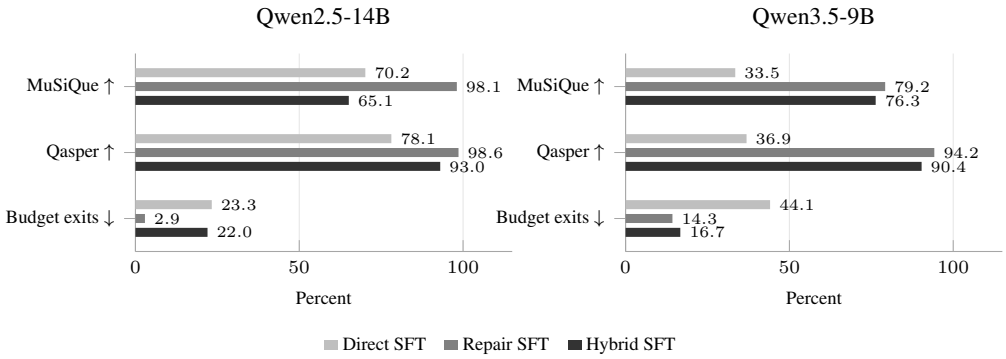

Figure 4: QA answer coverage and budget exits, grouped by student. Coverage is higher-is-better; budget exits are lower-is-better. Each group orders Direct SFT, Repair SFT, and Hybrid SFT from top to bottom. Budget rates use the same 4,196 non-ScienceWorld items for both students. Labels are rounded to one decimal; exact coverage and conditional F1 appear in Appendix A.

from 9.73 to 6.50 and 6.20. Qwen2.5 Repair SFT reaches 98.35% recorded completion and reduces budget exits from 21.4% to 10.3%, while Hybrid SFT remains close to Direct SFT (Table 1).

Figure 4 shows QA coverage and budget exits on the 4,196-item non-ScienceWorld suite. Qwen3.5 Direct SFT answers only 33.47% of MuSiQue and 36.92% of Qasper; Repair SFT raises coverage to 79.23%/94.23%, and Hybrid SFT to 76.33%/90.35%. On Qasper, Direct SFT's higher conditional F1 than Repair SFT does not offset its lower coverage. Qwen2.5 Repair SFT answers nearly every QA item, but its low MuSiQue F1 shows that coverage alone is insufficient. Hybrid SFT has the highest conditional F1 among SFT variants on both datasets and students, although differing answered sets prevent a direct comparison of underlying competence. In particular, Qwen2.5 Hybrid SFT answers fewer MuSiQue questions than Direct SFT (65.12% versus 70.17%) yet attains higher conditional F1 (0.5583 versus 0.4131). Its planned-F1 gain thus differs from the Qwen3.5 pattern, where both coverage and conditional F1 increase.

Assistant-ended episodes measure stopping behavior, not correctness. Runner status also obscures some budget exits: every ALFWorld record is marked completed (Appendix B). Infrastructure failures contribute to missing answers, but assigning perfect F1 to all failed Qwen3.5 Direct SFT QA records still leaves Hybrid SFT ahead on both datasets (Appendix G). These results identify coverage and termination differences without isolating their causes.

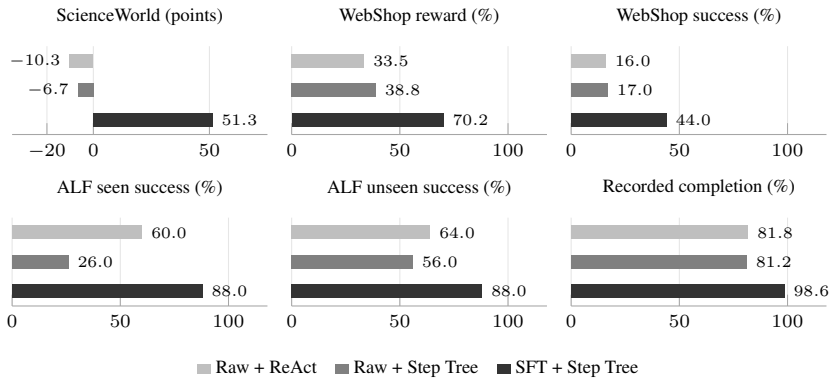

Figure 5: Harness and training comparison for Qwen2.5-14B on the quick eval set. Each panel orders Raw + ReAct, Raw + Step Tree, and SFT + Step Tree from top to bottom. ScienceWorld uses signed environment points; other panels show reward, success, or recorded completion on a 0–100 scale. The Raw systems change only the controller; SFT also changes weights. Labels are rounded; exact values appear in Appendix A.

4.3 Harness and training

At fixed Qwen2.5 Raw weights, our memory harness (labeled Step Tree in Figure 5) improves ScienceWorld and WebShop over ReAct but reduces ALFWorld success on the frozen 500-item suite. Adding trajectory SFT within this harness improves all interactive scores, reaching 51.30 ScienceWorld points and 493/500 recorded completions. The Raw rows isolate the controller change; comparison with the SFT row evaluates the trained system and does not establish that the same training gains would occur under ReAct.

4.4 Transfer to IIRC and BFCL

On Qwen2.5 IIRC with official DROP rescoring (Dua et al., 2019), Hybrid SFT reaches 20.97% F1, versus Direct SFT 14.35% and Repair SFT 10.51% (Table 2). Repair SFT completes the most questions (1,277/1,301), compared with Hybrid SFT 743 and Direct SFT 679. Thus Hybrid SFT QA advantage extends to this local-corpus transfer setting, and Repair SFT provides the highest recorded completion. Results are conditional on the shared corpus, which lacks ten linked articles.

Table 2: Qwen2.5-14B on 1,301 IIRC questions. EM/F1 use official DROP rescoring; unfinished answers score zero. Bold/underline: best/second-best (lowest/second-lowest for maximum-step).

Method	EM	F1	Completed	Max-step
Direct SFT	<u>0.0592</u>	<u>0.1435</u>	679	622
Repair SFT	0.0300	0.1051	**1,277**	**24**
Hybrid SFT	**0.1414**	**0.2097**	<u>743</u>	<u>558</u>

Denominator is 1,301. The local corpus lacks ten linked articles.

On the 1,000 selected BFCL multi-turn tasks, Repair SFT leads official checker accuracy at 6.4%, followed by Hybrid SFT at 5.3% and Direct SFT at 4.0% (Table 7 in Appendix). Repair SFT improves accuracy by 2.4 percentage points over Direct SFT on this selection. Absolute accuracy remains low, leaving substantial room to improve multi-turn function calling.

Interpreting the differences across students. The coverage decomposition identifies distinct failure patterns. Qwen2.5 Repair SFT reaches 98.10% MuSiQue coverage but only 12.58% conditional F1; missing answers therefore cannot explain its low planned F1. Qwen3.5 Direct SFT instead has low coverage despite a conditional F1 of 60.48%. Because answered sets differ, these observations do not isolate competence on a common subset.

What the source comparison establishes. Hybrid SFT QA gains support the selection recipe, which jointly changes task-level selection, role sources, and supervised token counts. A random mixture matched for role proportions and token budget would test selection beyond source access; multiple seeds would assess training variability. The current bootstrap addresses retained evaluation outcomes rather than these training questions.

Execution behavior and practical use. Fewer budget exits do not ensure correct answers, and fewer Task Agent steps do not establish lower inference cost: summarization and memory maintenance add model calls within the outer step budget. Measuring total tokens, latency, and task success under matched resource budgets would clarify the practical trade-offs of the reported execution gains.

5 Related Work

Agentic supervised fine-tuning. FireAct studies teacher-generated agent trajectories, AgentTuning combines agent interactions with general instruction data, and ToolLLM constructs instruction–API solution pairs (Chen et al., 2023; Zeng et al., 2024; Qin et al., 2024). Agent-FLAN separates format learning from agent reasoning in data design (Chen et al., 2024); Lumos trains modular planning and grounding components (Yin et al., 2024). We study target construction under a fixed harness: Repair SFT turns student attempts into executable corrections, and Hybrid SFT selects demonstrations and repairs by outcome and role.

Data distillation and student-generated contexts. Sequence-level distillation trains students on teacher-generated outputs (Kim & Rush, 2016), while Self-Instruct generates and filters instruction-tuning examples (Wang et al., 2023). Distribution-based methods use teacher probabilities: GKD supervises student-generated sequences, and MiniLLM optimizes reverse KL divergence through on-policy learning (Agarwal et al., 2024; Gu et al., 2024). Black-box on-policy distillation can instead train a discriminator on teacher and student responses (Ye et al., 2025). DAgger supervises learner-induced states through iterative expert labeling and aggregation (Ross et al., 2011). Repair SFT connects student-conditioned supervision with offline data distillation by learning black-box revisions of frozen attempts without teacher probabilities or online teacher calls. Executing revised actions keeps distilled trajectories grounded in consistent observations.

Trajectory refinement and data selection. Self-Refine revises outputs through self-feedback; Reflexion stores verbal feedback for later attempts, both without weight updates (Madaan et al., 2023; Shinn et al., 2023). Repair SFT converts corrections into training targets for subsequent student execution. LIMA studies curated instruction examples, while LESS selects data through gradient-based influence estimates (Zhou et al., 2023; Xia et al., 2024). Hybrid SFT selects Direct SFT or Repair SFT candidates using training-task outcomes and role-specific source rules.

6 Conclusion

We introduced Repair SFT and Hybrid SFT for student-conditioned trajectory supervision. Repair SFT converts student attempts into offline targets through black-box teacher corrections and environment execution, preserving consistency between revised actions and observations. Hybrid SFT selects repaired trajectories and direct demonstrations by task outcome and agent role. Our memory harness coordinates execution and context management, with a Step Tree Agent using four tools while preserving canonical history.

Across Qwen2.5-14B and Qwen3.5-9B, Repair SFT achieves the highest recorded completion among trained variants and reduces Qwen3.5 budget exhaustion from 44.1% to 14.3%. Hybrid SFT leads trained variants in MuSiQue and Qasper F1 and WebShop reward on both students, and extends its QA advantage to IIRC transfer. These findings show that student-conditioned corrections and selective teacher supervision improve agent execution and task performance. Executable trajectories built from generated teacher responses bring student-aware data distillation to standard offline SFT with black-box teachers, making trajectory repair and selection practical components of agentic training.

AI USE STATEMENT

GPT-5.6 generated Direct SFT trajectories and student-conditioned repairs, including the assisted Direct SFT subset described in Section 3.2. All student trajectories were executed against the stated tools and environments; generated observations were not substituted for tool outputs. A generative-AI assistant was also used for manuscript organization and language editing. The authors reviewed the resulting training records, analysis, and manuscript, and benchmark-native scoring was not replaced by model-based judgment.

ETHICS STATEMENT

The study did not recruit human participants or collect private user data. It uses public research benchmarks and model-generated interaction trajectories. Because improved tool use can increase both useful automation and misuse, any release of checkpoints or trajectories should retain benchmark licenses, document the accessible tools, and avoid packaging credentials or private tool state.

REPRODUCIBILITY STATEMENT

The experiment records include frozen split identifiers, configuration files, checkpoint hashes, and scoring outputs. Subject to benchmark and model licenses, the anonymized submission package will include derived training-data construction scripts, evaluation manifests, and the reported aggregate scores.

DATA AVAILABILITY STATEMENT

Evaluation datasets are drawn from the public benchmarks cited in Section 3. Derived trajectories and run manifests will be distributed with the anonymized supplementary material where their upstream licenses permit redistribution; otherwise, scripts and content hashes will be provided.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. FireAct: Toward Language Agent Fine-tuning. *arXiv preprint arXiv:2310.05915*, 2023. doi: 10.48550/arXiv.2310.05915. URL https://arxiv.org/abs/2310.05915.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. Agent-FLAN: Designing Data and Methods of Effective Agent Tuning for Large Language Models. *arXiv preprint arXiv:2403.12881*, 2024. doi: 10.48550/arXiv.2403.12881. URL https://arxiv.org/abs/2403.12881.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4599–4610, 2021. doi: 10.18653/v1/2021.naacl-main.365.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1383–1392, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1128. URL https://aclanthology.org/P18-1128/.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs.

In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2368–2378, 2019. doi: 10.18653/v1/N19-1246.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. IIRC: A dataset of incomplete information reading comprehension questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 1137–1147, 2020. doi: 10.18653/v1/2020.emnlp-main.86.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as Agents. *arXiv preprint arXiv:2308.03688*, 2023. doi: 10.48550/arXiv.2308.03688. URL https://arxiv.org/abs/2308.03688.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. *arXiv preprint arXiv:2303.17651*, 2023. doi: 10.48550/arXiv.2303.17651. URL https://arxiv.org/abs/2303.17651.

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 *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley Function Calling Leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Qwen Team. Qwen3.5-9B model card. Hugging Face, 2026. URL https://huggingface.co/Qwen/Qwen3.5-9B. Accessed 2026-09-14.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020. doi: 10.1109/SC41405.2020.00024.

Stéphane Ross and J. Andrew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pp. 661–668. PMLR, 2010. URL https://proceedings.mlr.press/v9/ross10a.html.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. *arXiv preprint arXiv:2303.11366*, 2023. doi: 10.48550/arXiv.2303.11366. URL https://arxiv.org/abs/2303.11366.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=q5QaTQcUbS.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475.

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting Influential Data for Targeted Instruction Tuning. *arXiv preprint arXiv:2402.04333*, 2024. doi: 10.48550/arXiv.2402.04333. URL https://arxiv.org/abs/2402.04333.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. doi: 10.48550/arXiv.2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757, 2022. doi: 10.52202/068431-1508.

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. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024. doi: 10.48550/arXiv.2406.12045. URL https://arxiv.org/abs/2406.12045.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025. doi: 10.48550/arXiv.2511.10643. URL https://arxiv.org/abs/2511.10643.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. Agent lumos: Unified and modular training for open-source language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12380–12403. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.670.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. AgentTuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024. doi: 10.18653/v1/2024.findings-acl.181.

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, pp. 55006–55021, 2023.

A Exact values for the result figures

Tables 3 and 4 list the values plotted in Figures 4 and 5.

Table 3: Values plotted in Figure 4. Conditional F1 is the mean over answered items. Bold/underline: best/second-best within each model; for budget exit, lowest/second-lowest.

		Conditional F1		Coverage and termination			
		F1 | answered		Coverage		Assistant	Budget
Model	Source	MuSiQue	Qasper	MuSiQue	Qasper	ended	exit
Qwen3.5-9B	Direct SFT	0.6048	0.2567	0.3347	0.3692	28.1%	44.1%
	Repair SFT	0.6086	0.2295	**0.7923**	**0.9423**	**68.3%**	**14.3%**
	Hybrid SFT	**0.6497**	**0.2856**	0.7633	0.9035	65.6%	16.7%
Qwen2.5-14B	Direct SFT	0.4131	0.2219	0.7017	0.7811	59.1%	23.3%
	Repair SFT	0.1258	0.2084	**0.9810**	**0.9861**	**80.3%**	**2.9%**
	Hybrid SFT	**0.5583**	**0.2468**	0.6512	0.9303	59.8%	22.0%

Table 4: Values plotted in Figure 5. Raw rows change only the controller. ScienceWorld uses environment points; WebShop reports reward/success, ALFWorld reports success rates, and Completed is a count. Bold/underline mark the highest/second-highest distinct values.

System	ScienceWorld	WebShop R./S.	ALF seen	ALF unseen	Completed
Raw + ReAct	-10.25	0.3345/0.16	0.60	0.64	409/500
Raw + Step Tree	-6.66	0.3883/0.17	0.26	0.56	406/500
SFT + Step Tree	**51.30**	**0.7017/0.44**	**0.88**	**0.88**	**493/500**

Table 5: Terminal records on the 4,196-item Qwen3.5 suite. Completed, Max-step, and Failed are mutually exclusive status counts. Both is a subset of Completed whose termination reason was budget exhaustion; total budget exits equal Max-step plus Both.

Benchmark	Source	Completed	Max-step	Failed	Both
MuSiQue dev (2,417)	Direct SFT	809	1233	375	0
	Repair SFT	1915	485	17	0
	Hybrid SFT	1845	564	8	0
Qasper dev (1,005)	Direct SFT	371	558	76	0
	Repair SFT	947	58	0	0
	Hybrid SFT	908	93	4	0
WebShop test (500)	Direct SFT	451	49	0	0
	Repair SFT	474	26	0	0
	Hybrid SFT	468	32	0	0
ALFWorld seen (140)	Direct SFT	140	0	0	8
	Repair SFT	140	0	0	17
	Hybrid SFT	140	0	0	12
ALFWorld unseen (134)	Direct SFT	134	0	0	3
	Repair SFT	134	0	0	13
	Hybrid SFT	134	0	0	1
All (4,196)	Direct SFT	1905	1840	451	11
	Repair SFT	3610	569	17	30
	Hybrid SFT	3495	689	12	13

Table 6: Qwen2.5-14B complete splits under Step Tree. QA uses planned denominators. ScienceWorld is in environment points; other scores are rates. Bold/underline: best/second-best.

Split	Metric	N	Raw	Direct SFT	Repair SFT	Hybrid SFT
MuSiQue	EM/F1	2,417	**0.3074/0.4310**	0.1427/0.2899	0.0116/0.1234	<u>0.2404</u>/<u>0.3636</u>
Qasper	F1	1,005	<u>0.2216</u>	0.1733	0.2055	**0.2296**
ScienceWorld	Score	1,819	−28.94	**78.95**	63.45	<u>73.05</u>
WebShop	R/S	500	0.3892/0.168	0.6555/**0.428**	<u>0.6684</u>/0.414	**0.6748**/<u>0.418</u>
ALF seen	Success	140	0.5143	<u>0.9214</u>	0.8643	**0.9357**
ALF unseen	Success	134	0.5821	**1.0000**	0.9104	<u>0.9851</u>
Completion C	Rate	6,015	<u>0.9413</u>	0.8318	**0.9835**	0.8412

B TERMINAL-STATUS BREAKDOWN

Section 3.4 distinguishes recorded completion C from successful termination. Table 5 gives the underlying counts on the complete Qwen3.5 splits. Direct SFT has more maximum-step statuses (1,840) than failed statuses (451), so failed requests alone cannot account for its low C. The ALFWorld runner also writes `completed` for all 274 items under each reported SFT source while some of those episodes ended on the step budget, so C alone does not reveal all budget exits. For Direct SFT, 1,840 maximum-step records plus 11 completed records with budget exhaustion give 1,851/4,196, or 44.1%.

C QWEN2.5 COMPLETE-SPLIT COMPARISON

Table 6 reports the four completed Qwen2.5-14B complete-split evaluations. Raw is the original Qwen2.5-14B-Instruct checkpoint: its 6,015 terminal records come from servers whose logs identify that checkpoint and were rescored offline with the same planned-denominator metrics as Direct SFT and Hybrid SFT. Repair SFT is the completed Repair SFT checkpoint run, scored on the same 6,015 core items. Every planned item has a terminal record; this record coverage does not imply answer coverage or task success. The table retains the planned denominators used in the main comparison.

Table 7: Qwen2.5-14B on 1,000 BFCL v3 multi-turn tasks. Accuracy uses the official checker; failed records count as incorrect.

Variant	Correct	N	Official accuracy	Failed
Direct SFT	40	1,000	0.040	8
Repair SFT	64	1,000	0.064	38
Hybrid SFT	53	1,000	0.053	20

Table 8: Qwen3.5-9B on all 1,819 ScienceWorld test episodes. Scores keep signed environment points, including negative terminations.

Variant	ScienceWorld score	Negative episodes	Failed
Raw	−31.90	1,074	5
Direct SFT	79.18	17	0
Repair SFT	32.87	325	9
Hybrid SFT	60.77	175	0

The BFCL evaluation covers 1,000 selected multi-turn tasks across five categories (200 each): base, composite, long-context, missing-function, and missing-parameter. Official multi-turn checker accuracy, with failed records retained in the denominator, is shown in Table 7.

D Complete ScienceWorld evaluation

The Qwen3.5 runs also cover the complete 1,819-item ScienceWorld test split. Scores use the benchmark's signed environment-point convention, including negative terminal penalties for all variants; the results are shown in Table 8. The installed environment API returns 100 times its internal score and terminates on a negative score. We retain the resulting -100 values without clipping; Raw has 1,074 such episodes, Direct SFT 17, Repair SFT 325, and Hybrid SFT 175. Recorded completion is a runner status and can include episodes ending in a negative score.

E Construction and execution records

Student draft provenance. The Qwen2.5 draft archive identifies its collector as Qwen2.5-14B-Instruct; the Qwen3.5 archive identifies Qwen3.5-9B. These archives precede the reported SFT runs. Qwen3.5 collection uses its native chat template with thinking disabled. Records that retain decoding metadata specify temperature 0.6 and top-$p = 0.95$, but not every record retains those fields. The Qwen2.5 source archive contains 3,365 records before selection of the frozen tasks and includes targeted recollection of invalid role/tool-scope traces. The Qwen3.5 archive contains 2,000 task records; its repair projection requires accepted repairs and preserves the frozen task correspondence and split keys. These archives establish separate student sources, but do not provide a uniform per-task attempt count or a collector checkpoint hash for every draft.

Direct SFT source retention. The Direct SFT source archive retains 1,925 trajectories with completed status and 75 with maximum-step status (55 WebShop and 20 ScienceWorld). Retention preserves the frozen task set rather than filtering exclusively for successful teacher demonstrations. The 100 assisted cases described in Section 3.2 are a separate property from both completion status and the train/validation assignment.

Resume and consolidation rules. The final coverage audit found one terminal record for every planned item in the nine added runs (26,502 records across 24 dataset slices), with no duplicate or unexpected IDs and no unreadable JSON records. The final resume filled only missing or unfinished items and retained existing terminal outcomes, including failed and maximum-step records. The archived result manager normally skips completed items. Maximum-step records are also skipped when its accept-maximum-steps flag is enabled. A skip-existing flag instead retains all terminal records, except failed records when the retry-failed flag is enabled. Missing or running records remain eligible. Reruns replace the per-item result file rather than preserving every attempt there.

The Qwen3.5 complete-suite launcher defaults to two passes, but later resumes mean this is not a verified total attempt count per item. The scoring audit prefers an explicitly consolidated shard when available, checks item IDs against their file paths, and rejects unresolved duplicate IDs. It does not select the highest-scoring record. Bootstrap estimates therefore condition on retained outcomes rather than reconstructing first-attempt rates.

Generation and timeout limits. Saved complete-suite records confirm maximum Task Agent steps of 12 for MuSiQue, Qasper, and IIRC, 40 for WebShop, and 50 for ALFWorld; the frozen and complete ScienceWorld launchers use 50. The archived IIRC launcher explicitly caps Task Agent outputs at 2,048 tokens. The archived Qwen3.5 launchers do not explicitly set that cap, and the available records do not establish one common output limit or timeout across all historical runs. Current runtime defaults are not used to infer those missing historical settings. Infrastructure failures remain distinguished from model stopping behavior.

F TRAINING AND METRIC DETAILS

Tables 9, 10, and 11 give the source composition, role selection policy, and supervised token budgets. Hybrid SFT selection compares candidates for the same task. The 100 assisted Direct SFT trajectories are distinct from the 100-task validation split; hidden answers or verified actions were supplied to the teacher but not serialized into student inputs. Repair SFT guidance covers all four roles, so changed interactions can regenerate supporting role calls.

Table 9: Source tasks before splitting and role targets in the training split.

Source family	Tasks
HotpotQA, 2WikiMultiHopQA, Qasper	1,000
ALFWorld / WebShop / ScienceWorld	431 / 336 / 233
Total (training / validation)	2,000 (1,900 / 100)
Training role	**Instances per variant**
Task Agent	15,198
Step Tree Agent	7,599
Message Processor	4,559
Step Summarizer	3,040
Total	30,396

Training configuration. We fine-tune Qwen2.5-14B-Instruct (Yang et al., 2025) and Qwen3.5-9B (Qwen Team, 2026). Each model uses its official chat template and native tool-call representation. Qwen2.5 uses unified assistant content; Qwen3.5 keeps its model-native separation of reasoning content and tool-call syntax. All variants use full-parameter BF16 SFT for one epoch with DeepSpeed ZeRO-3 (Rajbhandari et al., 2020), learning rate 10^{-5}, micro-batch size 1, global batch size 32, and 950 optimizer steps. Context limits are 32,768 tokens for Qwen2.5 and 49,152 for Qwen3.5. We report one checkpoint per condition. Validation token accuracy is a training-health diagnostic only, because validation target distributions differ across trajectory sources.

Coverage and stopping behavior. For N planned QA items, let A contain the items with a completed record and an answer prediction. Answer coverage is $|A|/N$, and conditional F1 is the mean over A. Planned F1 equals coverage times conditional F1, with unanswered items scored zero. The answered sets may differ across models. Recorded completion C is the fraction of planned items with `status=completed`, independently of the stopping reason. Budget exhaustion is the fraction with `terminationReason=max_steps`, including records marked completed. The assistant-ended statistic in the results counts `assistant_without_tool_call`: it measures an assistant ending without a tool call, not answer correctness or all forms of successful environment termination. Environment-triggered stops are recorded separately as `external_stop`. Appendix B details the runner-specific status semantics. Aggregate execution rates weight planned items equally; the 500-item suite includes 50 items per ALFWorld split and 100 per other benchmark. Task scores are not pooled.

Table 10: Role-specific target selection for Hybrid SFT. Outcome comparisons use candidates for the same training task.

Role	Source selection
Task Agent	Higher task score; ties use fewer task-tool calls, then Repair SFT for WebShop and Direct SFT otherwise.
Step Summarizer	Follows the trajectory selected for the Task Agent.
Message Processor	Repair SFT records.
Step Tree Agent	Both sources, with balancing of memory actions and no-change decisions.

Table 11: Supervised target tokens in millions, from checkpoint-linked tokenization manifests. Counts exclude input-only context.

Student	Direct SFT	Repair SFT	Hybrid SFT
Qwen2.5-14B	3.663	2.490	2.351
Qwen3.5-9B	3.879	2.450	2.259

Decoding and step budgets. Evaluations use temperature 0, top-$p = 1$, native tool-call parsers, identical task IDs, and matching prompt/tool-schema versions across variants. Maximum Task Agent steps are 12 for MuSiQue, Qasper, and IIRC; 40 for WebShop; and 50 for ScienceWorld and ALFWorld. One step is one Task Agent generation and its tool execution, when present. Summarization, memory control, and message processing occur within this outer loop and do not each consume an additional Task Agent step. Rollback does not replenish the budget. These limits do not equalize total model calls or tokens.

Scoring and statistical scope. QA is rescored against the original annotations. ScienceWorld and WebShop retain saved finite scores or rewards even for interrupted episodes; ALFWorld uses the saved success outcome. Task scores are not pooled across benchmarks. Bootstrap resampling uses seed 20260915, with 10,000 paired resamples and Qasper clustering by source paper. Reported intervals apply to the earlier Qwen3.5 SFT comparison; the added Raw, ScienceWorld, Qwen2.5 complete-split, and BFCL comparisons are descriptive. Intervals do not capture training-seed variation, retry selection, shared MuSiQue components, or subset selection. They are not adjusted for multiple comparisons and do not establish a ranking across all tasks (Dror et al., 2018). Qasper contributes 68 source tasks and is treated as in-domain; its training and evaluation identifiers were checked for overlap.

G Sensitivity to failed QA records

Direct SFT has 451 failed records in the complete Qwen3.5 suite, of which 426 are header timeouts. Its non-completed statuses also include 1,840 maximum-step records; an additional 11 ALFWorld records are marked completed despite budget exhaustion. Assigning F1 of 1.0 to every failed Direct SFT QA record still leaves Hybrid SFT ahead by 0.1383 MuSiQue F1 and 0.0877 Qasper F1. This is an accounting upper bound on those failed items, not a prediction of rerun outcomes, and it does not remove retry selection or explain the mechanism behind budget exhaustion.