SCAD: STRUCTURED CREDIT ASSIGNMENT AND DISTILLATION FOR LONG-HORIZON AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Training long-horizon agents to solve complex tasks requires effective supervision over extended interaction sequences. However, sparse terminal rewards obscure intermediate contributions, while on-policy distillation can lose informative teacher guidance as student-generated histories grow. To address this problem, we introduce **SCAD**, which organizes interactions into planning and bounded subtask execution, distills execution in local contexts, and refines planning credit through cross-rollout subtask prefix trees, with planning receiving full terminal credit and execution receiving positive terminal credit and teacher guidance. Across all evaluated benchmarks, SCAD improves macro-average accuracy over the strongest training baseline by 4.48 percentage points for text tasks and 4.19 points for multimodal tasks. These results demonstrate that SCAD effectively combines outcome-based credit assignment with teacher-guided distillation to improve planning and execution in long-horizon agents. Our code is publicly available[1].

1 INTRODUCTION

Language model agents use tools to interact with external environments and accomplish tasks (Nakano et al., 2022; Yao et al., 2022). Long-horizon agents solve complex real-world tasks through many interdependent decisions (Sun et al., 2026). Earlier actions shape later information and options, requiring agents to coordinate decisions toward an overall goal (Wu et al., 2025a). Multimodal tasks further require connecting textual information with visual evidence (Li et al., 2026c). Success depends on both choosing productive directions and executing individual actions effectively (Peng et al., 2026). Training agents to acquire these capabilities requires supervision that remains useful throughout an extended sequence of decisions.

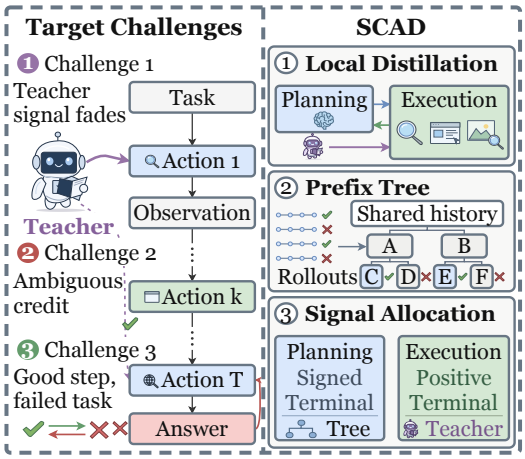


Figure 1: Target challenges and SCAD

Long-horizon agent training uses reinforcement learning (RL) and on-policy distillation (OPD); the latter builds on supervision at learner-visited states (Ross et al., 2011; Czarnecki et al., 2019). RL emphasizes task outcomes, while OPD supplies teacher guidance for intermediate behavior. Context Folding and CoDA manage histories through subtasks (Sun et al., 2026; Liu et al., 2026b). HiPER separates planning and execution advantages; GiGPO groups shared states for finer credit (Peng et al., 2026; Feng et al., 2025). Tree-GRPO uses tree-search rollouts to induce implicit step credit (Ji et al., 2026). MiniLLM uses reverse KL for language-model distillation (Gu et al., 2024). TCOD increases trajectory depth to stabilize multi-turn OPD (WANG et al., 2026). ATOD combines RL and distillation through annealed weights and turn-level supervision (Tan et al., 2026). These methods organize supervision at different scales.

[1] Anonymous code: `https://anonymous.4open.science/r/SCAD`

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

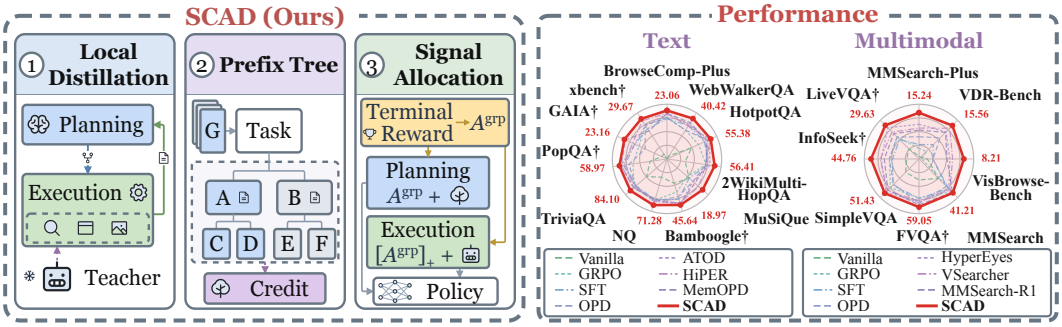


Figure 2: SCAD training overview (left) and benchmark accuracy (%) across text and multimodal tasks (right; SCAD in red). Axes use separate zero-based scales; † marks out-of-distribution datasets.

However, three challenges remain. **(i) Maintaining useful teacher supervision over long interaction histories.** Teacher feedback can lose discriminative value along student-generated histories (Li et al., 2026d; Liu et al., 2026a). **(ii) Ambiguous credit under long-range dependencies.** Terminal outcomes depend on preceding history and subsequent actions, obscuring individual decision contributions and complicating cross-rollout comparisons (Feng et al., 2025; He et al., 2026). **(iii) Coordinating task-level rewards and local learning.** Outcome rewards guide task success, while teacher feedback guides local learning. Uniform mixing can let negative terminal feedback suppress useful teacher guidance for correctly executed actions in failed trajectories (Hübotter et al., 2026).

To address these challenges, we propose **SCAD**, a framework for training long-horizon agents through three complementary designs. **(i) Subtask-local distillation.** We distill bounded subtask execution in local contexts to limit cross-subtask history accumulation, helping sustain useful teacher feedback at later stages. Teacher feedback is computed under the same local context used by the student. **(ii) Cross-rollout planning credit.** We construct subtask–report prefix trees from completed rollouts and compare terminal returns under matched histories to estimate relative credit for individual planning decisions. **(iii) Structured signal allocation.** Planning retains signed terminal and tree credit, while execution receives positive terminal credit and local teacher feedback. This allocation preserves teacher guidance in failed trajectories and reinforces execution associated with task success.

We evaluate SCAD on text-based and multimodal long-horizon benchmarks against RL, OPD, and hybrid baselines. SCAD improves overall macro-average accuracy over the strongest training baseline by 4.48 and 4.19 percentage points, respectively. Efficiency analysis shows lower training time than flat GRPO + OPD. Ablations and diagnostics examine component contributions, supervision quality across interaction depths, planning credit, and the link between local execution and global success.

2 Related Work

Long-Horizon Agents and Architectures. Long-horizon agents gather textual and visual evidence (Nakano et al., 2022; Jin et al., 2025; Wu et al., 2025a;c; Zhang et al., 2026a). AgentFlow separates planning and execution (Li et al., 2026e); Context Folding and CoDA bound execution contexts (Sun et al., 2026; Liu et al., 2026b). AgentFold and IterResearch compress histories through folding or workspace reconstruction (Ye et al., 2026; Chen et al., 2026). SCAD extends context isolation to teacher supervision, distilling execution locally while planning retains task history for coordination.

Credit Assignment and Policy Distillation. Credit assignment uses return redistribution (Arjona-Medina et al., 2019), intermediate rollouts (Kazemnejad et al., 2025; Hou et al., 2025; Ji et al., 2026), shared-state or prefix comparisons (Feng et al., 2025; He et al., 2026; Wang et al., 2026; Tran et al., 2026), and hierarchical or role-based advantages (Peng et al., 2026; Xu et al., 2026). Distillation (Hinton et al., 2015; Rusu et al., 2016) uses on-policy learning (Agarwal et al., 2024; Gu et al., 2024; Ko et al., 2025), temporal curricula (WANG et al., 2026), and memory alignment (Liu et al., 2026c). Kickstarting (Schmitt et al., 2018) and hybrid methods (Hübotter et al., 2026; Tan et al., 2026; Lin et al., 2026) combine RL and distillation. SCAD derives planning credit from subtask–report prefixes, blocks negative terminal credit to execution, and retains local teacher guidance.

Table 1: Comparison of how representative methods organize supervision for long-horizon agents. SCAD uses subtask boundaries to define local teacher contexts and subtask–report prefixes to estimate planning credit across completed rollouts. Planning retains signed terminal credit, while execution receives its positive part together with teacher guidance within each bounded subtask context.

Method	Organizing structure	How supervision is constructed
HiPER (Peng et al., 2026)	Subgoals and actions	Hierarchical advantages for planning and execution
GiGPO (Feng et al., 2025)	Shared states	Shared-state action comparisons plus trajectory credit
Tree-GRPO (Ji et al., 2026)	Rollout trees	Branch sampling and tree groups induce implicit step credit
TCOD (WANG et al., 2026)	Trajectory segments	Expands student-controlled spans for token distillation
MemOPD (Liu et al., 2026c)	Invocation states	Reconstructs compact invocation states for RL and OPD
ATOD (Tan et al., 2026)	Interaction turns	Anneals RL/OPD mixing with turn-weighted distillation
SCAD (Ours)	**Subtask–report prefixes**	Subtask boundaries set **local teacher contexts**; subtask–report prefixes organize **planning credit**; terminal credit is assigned **by role**

3 METHODOLOGY

Problem Formulation. We study long-horizon tasks with input $x = (q, v)$, comprising a task instruction q and optional visual input v. A policy π_θ with parameters θ generates action a_t at interaction step t, then receives observation o_t from the environment's observation law $\mathcal{P}$. A trajectory $\tau = (a_t, o_t)_{t=1}^{H_\tau}$ contains H_τ steps. Its history $h_t = (x, a_{<t}, o_{<t})$ includes all preceding actions and observations, and c_t is the rendered model context. Given training pairs $(x, y^\star)$ from dataset $\mathcal{D}$, a terminal evaluator compares the final result with reference outcome $y^\star$. Its binary reward $R(\tau, y^\star) \in \{0, 1\}$ equals one when the completed trajectory succeeds and zero otherwise:

$$P_\theta(\tau \mid x) = \prod_{t=1}^{H_\tau} \pi_\theta(a_t \mid c_t)\, \mathcal{P}(o_t \mid h_t, a_t), \qquad J_{\text{task}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D},\, \tau\sim P_\theta(\cdot|x)}\big[R(\tau, y^\star)\big]. \quad (1)$$

Here P_θ is the trajectory distribution and J_{task} the task objective; $\mathbb{E}$ averages over training inputs and interactions. The product separates policy-controlled actions from environment responses, whose joint sequence determines terminal success. A frozen teacher π_T supplies intermediate feedback; terminal success remains the objective because local behavior alone does not determine the outcome.

3.1 SUBTASK-LOCAL ON-POLICY DISTILLATION

SCAD uses a shared policy for planning and bounded subtask execution. Each subtask defines a local execution context for on-policy teacher supervision.

Planning and Bounded Execution. Planning reads the task input and accumulated planning decisions and execution reports. It delegates a self-contained subtask via `branch` or submits a final output via `finish`, without calling environment tools. Execution handles the assignment with environment tools and sends a report back via `return`; it cannot call `branch` or `finish`. With environment tools $\mathcal{U}_{\text{env}}$, the allowed tool sets are $\mathcal{U}_P = \{\texttt{branch}, \texttt{finish}\}$ for planning and $\mathcal{U}_E = \mathcal{U}_{\text{env}} \cup \{\texttt{return}\}$ for execution. Let i index complete rollouts and k their subtasks, with instruction d_{ik}, report r_{ik}, and prior planning history $h^P_{i,<k}$. The corresponding model contexts are

$$c^P_{ik} = \mathcal{C}_P(x, h^P_{i,<k}), \qquad c^E_{ikt} = \mathcal{C}_E(x, d_{ik}, u_{ik,<t}). \quad (2)$$

Superscripts P, E denote planning and execution; $\mathcal{C}_P, \mathcal{C}_E$ render contexts with their system prompts, and $u_{ik,<t}$ contains local interactions before token t. Each execution starts from (x, d_{ik}), excluding generated planning history and earlier execution traces. After it returns, planning appends the report to its history and decides whether to delegate again or finish. We bound local transcript growth by token budget B, reserving space for the report. With $|\cdot|$ measuring token length and $O(\cdot)$ bounding growth up to constant factors, token y_{ikt} is sampled from frozen rollout policy π_{old}:

$$|c^E_{ikt}| = O(|x| + |d_{ik}| + B), \qquad y_{ikt} \sim \pi_{\text{old}}(\cdot \mid c^E_{ikt}). \quad (3)$$

The bound includes the fixed task input and subtask instruction; B limits the added local transcript rather than the full task history. Execution reports results and remaining uncertainties when its subtask ends or budget is exhausted. Only this report enters planning history. We instantiate $\mathcal{U}_{\text{env}}$ with search, page opening, and image search; other tasks may use different tools.

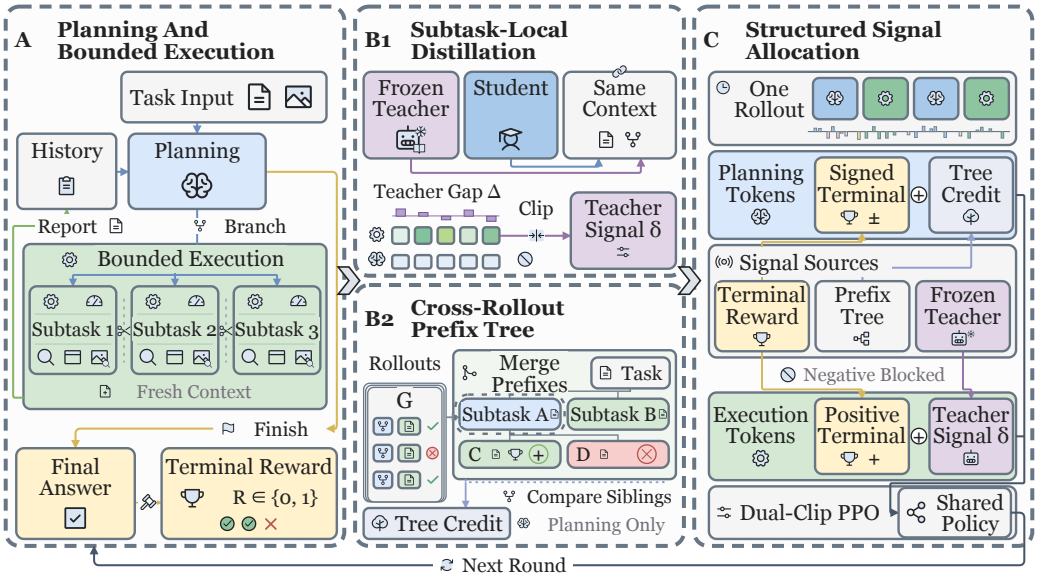


Figure 3: SCAD training framework with local distillation and subtask prefix-tree credit. Planning uses full terminal credit; execution uses positive terminal credit and teacher guidance.

Teacher Guidance in Matched Contexts. Following on-policy distillation (Agarwal et al., 2024; Gu et al., 2024), the teacher scores student-sampled execution tokens in the same local context as the student; planning tokens receive no teacher supervision. For token y and context c, the log-probability gap $\Delta(y, c)$ measures relative teacher preference. Its expectation is negative reverse KL, where D_{KL} denotes Kullback–Leibler divergence. A positive cap b bounds the signed signal δ_{ikt} via clip:

$$\Delta(y, c) = \log \frac{\pi_T(y \mid c)}{\pi_{\mathrm{old}}(y \mid c)}, \qquad \delta_{ikt} = \mathrm{clip}\big(\Delta(y_{ikt}, c^E_{ikt}), -b, b\big)\,, \tag{4}$$
$$\mathbb{E}_{y\sim\pi_{\mathrm{old}}(\cdot|c)}[\Delta(y, c)] = -D_{\mathrm{KL}}(\pi_{\mathrm{old}}(\cdot \mid c) \,\|\, \pi_T(\cdot \mid c))\,.$$

Teacher feedback uses local evidence; collected subtasks can be scored in parallel.

Proposition 1. *At fixed local contexts and the rollout policy, bounded policy scores give a reverse-KL gradient approximation whose bias is controlled by clipped log-ratio tails (Appendix C.1).*

3.2 Cross-Rollout Planning Credit

We build subtask–report prefix trees from G completed rollouts per input, comparing terminal returns under shared histories to estimate credit for subtask choices and stopping decisions.

Prefix Construction and Action Values. Subtask texts are deduplicated and greedily clustered by embedding cosine similarity. Reports undergo typed normalization and semantic matching, with long reports kept intact. Let z_{ik} be the subtask cluster and $\kappa_{z_{ik}}$ its report canonicalizer. Node p_{ik} follows subtask k: root creates the input's root, and node identifies a child by its parent, subtask, and report key, keeping distinct canonical histories separate. A `STOP` branch records the decision to finish:

$$p_{i0} = \mathrm{root}(x), \qquad p_{ik} = \mathrm{node}(p_{i,k-1}, z_{ik}, \kappa_{z_{ik}}(r_{ik}))\,. \tag{5}$$

At parent p, $\mathcal{A}(p)$ contains observed subtask clusters and `STOP`; $\mathcal{I}(p, a)$ collects distinct rollouts choosing a, pooling report variants. Given return $R_i = R(\tau_i, y^\star)$ for rollout τ_i, define visit count n_{pa}, Monte Carlo action value $Q(p, a)$, and parent visits N_p:

$$n_{pa} = |\mathcal{I}(p, a)|, \qquad Q(p, a) = \frac{1}{n_{pa}} \sum_{i\in\mathcal{I}(p,a)} R_i, \qquad N_p = \sum_{a\in\mathcal{A}(p)} n_{pa}, \tag{6}$$

Here $|\mathcal{I}(p, a)|$ is set cardinality. First-visit counting prevents duplicated training sequences from multiplying a rollout's contribution; the tree and decision spans remain fixed during optimization.

Sibling Comparisons with Prior Shrinkage. Let μ_x and σ_x denote the mean and sample standard deviation of the G terminal returns. At each parent prefix, $\bar{Q}_p$ is the mean of observed action values weighted by their visit counts. We keep these action values unchanged and shrink their mean toward the problem-level mean. The positive prior count ν controls this shrinkage in baseline $V(p)$; its influence decreases as more rollouts visit the parent, giving observed local returns more weight:

$$V(p) = \frac{N_p \bar{Q}_p + \nu \mu_x}{N_p + \nu} = \bar{Q}_p + \frac{\nu}{N_p + \nu}(\mu_x - \bar{Q}_p). \tag{7}$$

The local scale s_p is the sample standard deviation of action values repeated by visit count, so frequent choices contribute proportionally. The variance below applies to parents with at least two visits; otherwise the local scale is zero. The effective scale $\widetilde{\sigma}_p$ uses the local scale if it exceeds the positive tolerance ϵ, otherwise the problem-level standard deviation if above that tolerance, then one. The resulting tree advantage A^{tree} uses the shared cap b to limit the magnitude of planning credit:

$$s_p^2 = \frac{\sum_{a \in \mathcal{A}(p)} n_{pa}(Q(p,a) - \bar{Q}_p)^2}{N_p - 1}, \qquad A^{\text{tree}}(p,a) = \text{clip}\left(\frac{Q(p,a) - V(p)}{\max(\widetilde{\sigma}_p, \epsilon)}, -b, b\right). \tag{8}$$

The same baseline and scale apply to all observed choices under a parent. With one observed choice, single-candidate credit reflects prior shrinkage; with multiple choices, multi-candidate credit also compares actions. Both use Eq. 8 only on planning spans; reports match histories without serving as verified subtask outcomes or execution rewards; terminal success supplies the return signal.

Proposition 2. *Before scaling and clipping, tree credit removes within-choice return variance; prior shrinkage shifts its mean without changing sibling contrasts or variance (Appendix C.2).*

3.3 Structured Signal Allocation and Policy Optimization

Planning retains signed terminal and tree credit, while execution combines positive terminal credit with signed local teacher feedback through structured signal allocation across the shared policy.

Terminal Credit and Teacher Guidance. In rollout i, $\mathcal{P}_i$ and $\mathcal{E}_i$ index planning and execution tokens. The GRPO advantage A_i^{grp} is the terminal return minus the group mean, divided by the group standard deviation plus the numerical tolerance (Shao et al., 2024). Each rollout enters the group once. For token t, B_{it} sums tree credit over planning decisions whose spans contain it, including `STOP`; δ_{it} is local teacher feedback indexed within the full rollout. The token advantage A_{it} combines these signals with unit weights according to whether the token belongs to planning or execution:

$$A_{it} = A_i^{\text{grp}} + B_{it} \quad (t \in \mathcal{P}_i), \qquad A_{it} = \max(A_i^{\text{grp}}, 0) + \delta_{it} \quad (t \in \mathcal{E}_i). \tag{9}$$

Tree credit is zero outside recorded planning decisions. Negative teacher gaps can counteract success reinforcement; when all group outcomes agree, terminal credit vanishes but distillation remains.

Token-Averaged Policy Update. For generated token y_{it} in stored context c_{it}, ρ_{it} is the current-to-rollout policy probability ratio. The binary mask m_{it} selects valid generated tokens, excluding prompts, observations, padding, and overlong samples; M is their total count. The indicator $\mathbf{1}[\cdot]$ activates the negative-advantage correction. The PPO surrogate g_{it} uses clip width ϵ_c (Schulman et al., 2017); dual clipping with factor c_d greater than one bounds negative-advantage updates (Ye et al., 2020). We maximize the token-averaged objective J_{SCAD}, keeping rollout probabilities, teacher scores, and all advantages fixed throughout optimization of the shared policy on the collected batch:

$$\begin{aligned} \rho_{it} &= \frac{\pi_\theta(y_{it} \mid c_{it})}{\pi_{\text{old}}(y_{it} \mid c_{it})}, \qquad g_{it} = \min(\rho_{it} A_{it}, \text{clip}(\rho_{it}, 1-\epsilon_c, 1+\epsilon_c) A_{it}), \\ J_{\text{SCAD}}(\theta) &= \frac{1}{M} \sum_{i,t} m_{it} \left[g_{it} + \mathbf{1}[A_{it} < 0] \max(c_d A_{it} - g_{it}, 0) \right]. \end{aligned} \tag{10}$$

Each update grades rollouts, builds the prefix tree, and scores execution tokens before optimizing the shared policy. Teacher and embedding models remain frozen. All supervision enters through allocated advantages, without separate reference-KL or entropy terms for planning or execution.

Proposition 3. *Under dual clipping, execution allocation weakly increases each token's direct log-probability derivative over uniform fusion and equals teacher-only updates on failures (Appendix C.3).*

Table 2: Text accuracy (%); Qwen3-4B(student) / Qwen3-4B-Instruct-2507 (teacher). Mean±std over 3 seeds. Avg.: equal-weight mean over 11 ID+OOD sets; †: OOD. Bold: best student result.

Method	Browse Comp-Plus	WebWalker QA	Hotpot	2Wiki	MuS	NQ	Trivia	Bamb.†	PopQA†	GAIA†	xbench†	Avg.
Teacher	$23.61_{\pm 0.5}$	$36.67_{\pm 3.1}$	$60.51_{\pm 0.9}$	$52.31_{\pm 1.5}$	$11.28_{\pm 0.9}$	$68.72_{\pm 1.8}$	$85.13_{\pm 0.9}$	$56.92_{\pm 0.0}$	$56.92_{\pm 0.0}$	$25.42_{\pm 0.0}$	$23.67_{\pm 0.6}$	$45.56_{\pm 0.6}$
Vanilla	$0.00_{\pm 0.0}$	$2.92_{\pm 0.7}$	$31.28_{\pm 0.9}$	$21.54_{\pm 0.0}$	$11.79_{\pm 0.9}$	$28.21_{\pm 0.9}$	$46.67_{\pm 1.8}$	$24.10_{\pm 0.9}$	$21.03_{\pm 0.9}$	$2.82_{\pm 1.0}$	$3.67_{\pm 0.6}$	$17.64_{\pm 0.5}$
GRPO	$19.44_{\pm 4.6}$	$28.75_{\pm 1.3}$	$40.00_{\pm 1.5}$	$46.15_{\pm 1.5}$	$10.77_{\pm 1.5}$	$63.59_{\pm 1.8}$	$75.90_{\pm 1.8}$	$32.82_{\pm 5.4}$	$42.05_{\pm 6.2}$	$15.25_{\pm 0.0}$	$25.00_{\pm 3.6}$	$36.34_{\pm 1.4}$
FoldGRPO	$17.22_{\pm 0.5}$	$34.58_{\pm 5.1}$	$52.31_{\pm 5.5}$	$52.31_{\pm 1.5}$	$13.33_{\pm 0.9}$	$63.08_{\pm 0.0}$	$71.28_{\pm 1.8}$	$41.54_{\pm 1.5}$	$55.38_{\pm 1.5}$	$13.56_{\pm 6.1}$	$21.00_{\pm 1.0}$	$39.60_{\pm 0.9}$
Tree-GRPO	$18.33_{\pm 2.2}$	$35.83_{\pm 0.7}$	$50.77_{\pm 0.0}$	$51.28_{\pm 1.8}$	$18.46_{\pm 1.5}$	$61.03_{\pm 0.9}$	$78.46_{\pm 8.1}$	$38.46_{\pm 4.1}$	$52.31_{\pm 1.5}$	$18.64_{\pm 1.7}$	$25.33_{\pm 4.5}$	$40.81_{\pm 0.3}$
HiPER	$21.67_{\pm 2.5}$	$32.92_{\pm 3.1}$	$54.36_{\pm 4.7}$	$46.15_{\pm 4.1}$	$16.92_{\pm 1.5}$	$66.15_{\pm 1.5}$	$73.85_{\pm 1.5}$	$43.59_{\pm 0.9}$	$53.85_{\pm 1.5}$	$18.64_{\pm 1.7}$	$28.33_{\pm 1.2}$	$41.49_{\pm 0.5}$
SFT	$19.17_{\pm 3.8}$	$33.33_{\pm 7.5}$	$47.69_{\pm 1.5}$	$48.21_{\pm 5.4}$	$11.79_{\pm 1.8}$	$63.08_{\pm 1.5}$	$71.28_{\pm 0.9}$	$40.00_{\pm 1.5}$	$52.31_{\pm 1.5}$	$17.51_{\pm 9.3}$	$22.33_{\pm 1.2}$	$38.79_{\pm 2.2}$
OPD	$19.72_{\pm 3.4}$	$28.75_{\pm 1.3}$	$51.79_{\pm 0.9}$	$45.64_{\pm 1.8}$	$13.85_{\pm 1.5}$	$65.64_{\pm 5.4}$	$71.79_{\pm 5.4}$	$37.44_{\pm 0.9}$	$47.18_{\pm 6.2}$	$11.86_{\pm 1.7}$	$25.00_{\pm 0.0}$	$38.06_{\pm 1.0}$
SDAR	$15.83_{\pm 3.8}$	$35.00_{\pm 1.3}$	$44.10_{\pm 0.9}$	$54.87_{\pm 1.8}$	$8.72_{\pm 7.6}$	$58.46_{\pm 1.5}$	$82.56_{\pm 6.2}$	$35.90_{\pm 1.8}$	$56.41_{\pm 5.4}$	$15.82_{\pm 1.0}$	$26.67_{\pm 1.2}$	$39.49_{\pm 2.2}$
ATOD	$21.39_{\pm 3.4}$	$33.75_{\pm 3.3}$	$53.85_{\pm 1.5}$	$50.77_{\pm 0.0}$	$13.85_{\pm 1.5}$	$64.10_{\pm 0.9}$	$75.90_{\pm 1.8}$	$45.13_{\pm 8.5}$	$56.92_{\pm 1.5}$	$17.51_{\pm 8.4}$	$24.67_{\pm 0.6}$	$41.62_{\pm 1.0}$
TCOD	$20.83_{\pm 0.0}$	$30.83_{\pm 0.7}$	$47.69_{\pm 1.5}$	$49.23_{\pm 1.5}$	$11.79_{\pm 0.9}$	$59.49_{\pm 0.9}$	$80.00_{\pm 0.0}$	$41.54_{\pm 8.1}$	$50.77_{\pm 1.5}$	$14.12_{\pm 1.0}$	$22.00_{\pm 0.0}$	$38.94_{\pm 0.7}$
MemOPD	$21.67_{\pm 3.0}$	$36.25_{\pm 1.3}$	$50.26_{\pm 0.9}$	$46.67_{\pm 1.8}$	$14.87_{\pm 1.8}$	$62.56_{\pm 0.9}$	$76.92_{\pm 5.5}$	$44.10_{\pm 0.9}$	$56.92_{\pm 1.5}$	$22.60_{\pm 1.0}$	$20.67_{\pm 3.5}$	$41.23_{\pm 0.9}$
SCAD (Ours)	$\mathbf{23.06}_{\pm 2.9}$	$\mathbf{40.42}_{\pm 2.6}$	$\mathbf{55.38}_{\pm 1.5}$	$\mathbf{56.41}_{\pm 2.4}$	$\mathbf{18.97}_{\pm 1.8}$	$\mathbf{71.28}_{\pm 3.9}$	$\mathbf{84.10}_{\pm 1.8}$	$\mathbf{45.64}_{\pm 0.9}$	$\mathbf{58.97}_{\pm 6.2}$	$\mathbf{23.16}_{\pm 3.5}$	$\mathbf{29.67}_{\pm 0.6}$	$\mathbf{46.10}_{\pm 1.2}$

Table 3: Multimodal accuracy (%); Qwen3-VL-2B(student) / Qwen3-VL-4B (teacher). Mean±std over 3 seeds. ID/All Avg.: equal-weight means over five/eight datasets; †: OOD. Bold: best student.

Method	MMSearch-Plus	VDR-Bench	VisBrowse-Bench	MMSearch	SimpleVQA	ID Avg.	FVQA†	InfoSeek†	LiveVQA†	All Avg.
Teacher	$16.19_{\pm 1.0}$	$16.79_{\pm 0.9}$	$5.64_{\pm 1.8}$	$40.61_{\pm 1.0}$	$45.71_{\pm 2.9}$	$24.99_{\pm 1.0}$	$56.19_{\pm 1.6}$	$37.14_{\pm 2.9}$	$28.89_{\pm 2.2}$	$30.90_{\pm 1.0}$
Vanilla	$0.95_{\pm 0.0}$	$0.74_{\pm 0.7}$	$1.54_{\pm 1.5}$	$23.64_{\pm 1.8}$	$20.00_{\pm 2.9}$	$9.37_{\pm 1.2}$	$25.71_{\pm 2.9}$	$11.43_{\pm 0.0}$	$8.89_{\pm 0.0}$	$11.61_{\pm 0.4}$
GRPO	$3.49_{\pm 2.4}$	$3.46_{\pm 1.9}$	$0.51_{\pm 0.9}$	$40.00_{\pm 1.8}$	$42.86_{\pm 0.0}$	$18.06_{\pm 0.5}$	$47.62_{\pm 1.6}$	$27.62_{\pm 1.6}$	$19.26_{\pm 2.6}$	$23.10_{\pm 0.4}$
MMSearch-R1	$9.52_{\pm 1.0}$	$12.35_{\pm 1.9}$	$5.13_{\pm 0.9}$	$40.61_{\pm 4.6}$	$46.67_{\pm 10.0}$	$22.85_{\pm 3.5}$	$56.19_{\pm 1.6}$	$32.38_{\pm 1.6}$	$22.22_{\pm 2.2}$	$28.13_{\pm 2.6}$
VSearcher	$10.16_{\pm 3.3}$	$14.32_{\pm 6.0}$	$6.15_{\pm 1.5}$	$37.58_{\pm 2.1}$	$47.62_{\pm 1.6}$	$23.17_{\pm 1.5}$	$51.43_{\pm 2.9}$	$36.19_{\pm 1.6}$	$23.70_{\pm 2.6}$	$28.39_{\pm 1.7}$
SFT	$4.76_{\pm 4.4}$	$6.67_{\pm 0.7}$	$2.56_{\pm 2.4}$	$35.76_{\pm 6.4}$	$39.05_{\pm 1.6}$	$17.76_{\pm 0.9}$	$48.57_{\pm 0.0}$	$24.76_{\pm 1.6}$	$17.04_{\pm 7.8}$	$22.40_{\pm 0.7}$
OPD	$7.30_{\pm 3.3}$	$9.88_{\pm 0.4}$	$2.56_{\pm 0.9}$	$33.33_{\pm 4.6}$	$38.10_{\pm 1.6}$	$18.23_{\pm 1.7}$	$54.29_{\pm 2.9}$	$27.62_{\pm 1.6}$	$19.26_{\pm 1.3}$	$24.04_{\pm 0.8}$
VOLD	$10.79_{\pm 3.8}$	$10.12_{\pm 2.6}$	$3.08_{\pm 1.5}$	$36.36_{\pm 1.8}$	$49.52_{\pm 1.6}$	$21.98_{\pm 0.3}$	$50.48_{\pm 3.3}$	$33.33_{\pm 1.6}$	$26.67_{\pm 2.2}$	$27.54_{\pm 0.9}$
HyperEyes	$11.43_{\pm 2.5}$	$12.59_{\pm 0.7}$	$5.64_{\pm 3.9}$	$36.36_{\pm 1.8}$	$49.52_{\pm 10.0}$	$23.11_{\pm 3.0}$	$53.33_{\pm 1.6}$	$35.24_{\pm 1.6}$	$27.41_{\pm 5.6}$	$28.94_{\pm 2.6}$
SCAD (Ours)	$\mathbf{15.24}_{\pm 3.8}$	$\mathbf{15.56}_{\pm 0.7}$	$\mathbf{8.21}_{\pm 6.4}$	$\mathbf{41.21}_{\pm 6.4}$	$\mathbf{51.43}_{\pm 2.9}$	$\mathbf{26.33}_{\pm 1.7}$	$\mathbf{59.05}_{\pm 1.6}$	$\mathbf{44.76}_{\pm 7.2}$	$\mathbf{29.63}_{\pm 1.3}$	$\mathbf{33.13}_{\pm 1.5}$

4 EXPERIMENTS

We address six research questions. **RQ1:** Does SCAD improve text and multimodal task performance over existing training baselines? **RQ2:** How much does each training component contribute? **RQ3:** Does local distillation sustain useful teacher supervision at greater interaction depths? **RQ4:** Does prefix-tree credit better distinguish planning decisions under shared histories? **RQ5:** Does structured signal allocation reconcile local execution quality with global success? **RQ6:** How efficiently does SCAD use teacher supervision and training computation?

4.1 EXPERIMENTAL SETUP

Datasets and Evaluation Metrics. ID datasets occur in training; OOD datasets do not (Appendix E). Test sets contain 814 text (525 ID, 289 OOD) and 510 multimodal (395 ID, 115 OOD) questions. Final-answer accuracy (%) uses greedy decoding and Gemini-3.7-Flash as judge. Avg. columns and figure accuracies equally weight all 11 text or 8 multimodal datasets. Checkpoints use validation macro accuracy, including Bamboogle, PopQA, and FVQA.

Baselines and Ablations. Tables 2–3 include Vanilla, teacher, RL, supervised, distillation, and hybrid baselines; Vanilla directly generates answers without planning or retrieval. RQ1 examines SCAD training dynamics; RQ2 adds local OPD, tree credit, and signal allocation within a fixed planning/execution architecture. RQ3–RQ5 analyze the three designs of Sections 3.1–3.3. RQ1–RQ2 cover both modalities; RQ3–RQ6 use text diagnostics. The cumulative ablation measures accuracy changes between successive configurations as components are added in a fixed order.

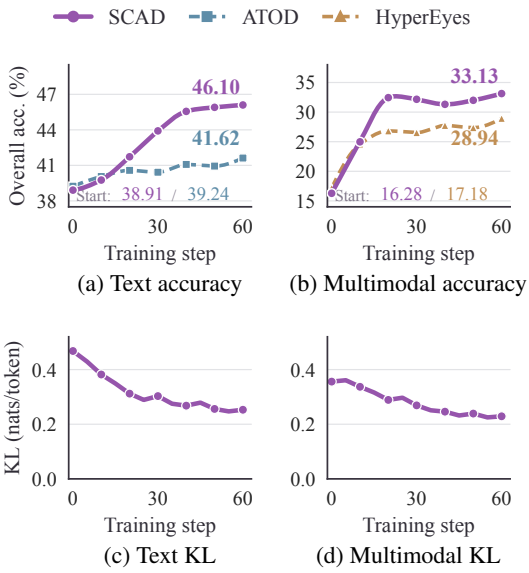
Implementation Details. Training uses 400 text and 347 multimodal examples. SCAD uses 60 steps, eight prompts with eight rollouts each, and AdamW at 10^{-6}. Each Worker has a local transcript growth budget of 16,384 tokens. All three signal weights are one, with teacher/tree signals clipped at five. Text training uses four RTX PRO 6000 GPUs, selecting each checkpoint by validation macro accuracy. Subtask and report thresholds are 0.83 and 0.90, with the prior count $\nu = 2$.

4.2 Main Results (RQ1)

Tables 2–3 compare task accuracy across methods, while Figure 4 pairs SCAD's accuracy and teacher–student divergence. The four panels jointly cover text and multimodal training dynamics.

Task performance. SCAD reaches 46.10% text and 33.13% multimodal overall macro accuracy. The strongest trained baselines are ATOD at 41.62% and HyperEyes at 28.94%, yielding gains of 4.48 and 4.19 points. Relative to Vanilla at 17.64/11.61%, the gains are 28.46/21.52 points. SCAD leads the trained baselines across individual datasets, and the multimodal student surpasses its teacher by 2.23 points in overall accuracy.

Training dynamics. Overall accuracy rises from 38.91 to 46.10% for text and from 16.28 to 33.13% for multimodal tasks. Reverse KL decreases from 0.468 to 0.253 nats/token for text and from 0.356 to 0.229 for multimodal execution. This diagnostic compares student and teacher distributions on fixed execution prefixes. Lower KL reflects closer alignment; accuracy separately measures task performance. Training initialization uses the SCAD scaffold and is tracked separately from the Vanilla table entries. SCAD starts lower but finishes above both baselines within a 60-step training run.



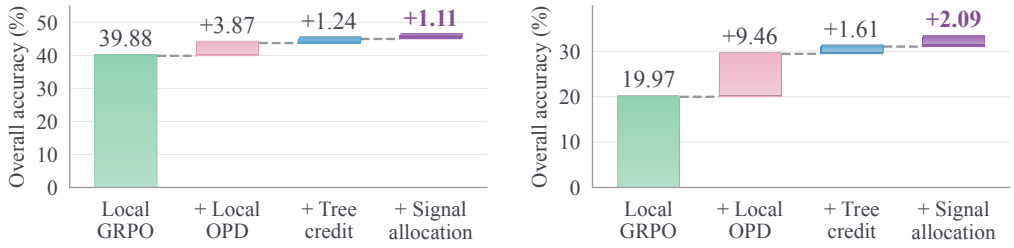
(a) Text accuracy (b) Multimodal accuracy

(c) Text KL (d) Multimodal KL

Figure 4: Training dynamics over the 60-step run. Panels (a,b) compare SCAD's accuracy with the strongest trained baselines, ATOD and HyperEyes, labeling both step-0 values as Start; (c,d) show SCAD's reverse KL.

4.3 Ablation Study (RQ2)

Figure 5 adds the three training components within a fixed local planning/execution architecture. The two waterfalls compare their cumulative contributions to text and multimodal overall accuracy.



(a) Text (b) Multimodal

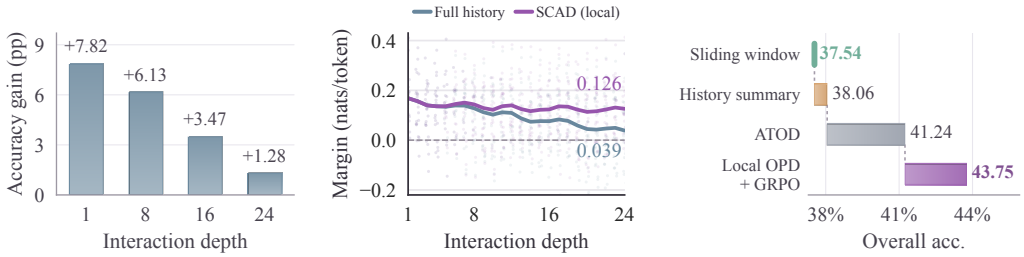
Figure 5: Cumulative overall accuracy gains as local OPD, tree credit, and structured signal allocation are added in turn to local GRPO for (a) text and (b) multimodal tasks, averaged over 3 seeds.

Component contributions. Local OPD adds 3.87/9.46 points to the text/multimodal starting accuracies of 39.88/19.97%. Tree credit adds 1.24/1.61 and structured allocation adds 1.11/2.09, reaching 46.10/33.13%. Local OPD accounts for 62.2/71.9% of the total gain in this sequence. Tree credit and allocation jointly add 2.35/3.70 points, providing complementary gains once local teacher supervision is available and consistently improving accuracy across both modalities.

Complementary signals. Tree credit refines planning supervision, while allocation controls execution updates, adding 2.09 points for multimodal tasks versus 1.11 for text. Total gains are 6.22/13.16 points. Components enter in the order the method composes them: local OPD supplies execution supervision first, and tree credit and allocation then build on it. The gains combine execution-level guidance with outcome-aware planning updates within the same interaction architecture. RQ3–RQ5 below examine the mechanism behind each of these components.

4.4 Analysis of Subtask-Local Distillation (RQ3)

Panels (a,b) in Figure 6 examine teacher supervision on 200 long-horizon questions from the text benchmarks. We use an untrained Qwen3-4B student and trajectories reaching interaction 24.



(a) Teacher continuation (b) Teacher preference (c) Context variants

Figure 6: (a) Teacher–student continuation gain (pp). (b) Teacher score margin (nats/token). (c) Overall text ID+OOD accuracy (%), averaged over 3 seeds; floating bars link adjacent settings.

Teacher continuation and preference. In (a), teacher and student continue from identical prefixes at depths 1/8/16/24 with equal budgets. Bars show the difference in final-answer accuracy, averaged equally across datasets: 7.82 to 1.28 points. Teacher continuation is diagnostic only. In (b), the teacher scores paired correct/incorrect candidates at the same state under full-history and local contexts. The margin is correct minus incorrect mean token log-probability; positive values favor correct execution. Dots show candidate pairs and lines show depthwise means. Local/full-history margins at depth 24 are 0.126/0.039 nats/token, showing clearer preferences with local contexts.

Accuracy after training. Sliding window and history summary (OPD) reach 37.54% and 38.06%. Within the Planner–Worker framework, ATOD reaches 41.24% using full-history teacher supervision without subtask resets. Local OPD + GRPO reaches 43.75%, matching RQ2's second stage and exceeding ATOD by 2.51 points. This variant uses ordinary Planner GRPO and signed Worker GRPO plus local OPD. Scores equally weight seven ID and four OOD text datasets.

4.5 Analysis of Cross-Rollout Planning Credit (RQ4)

Figure 7 compares prefix-tree credit and uniform planning-advantage scaling. We isolate single- and multi-candidate credit while keeping the execution-side supervision fixed across all five settings.

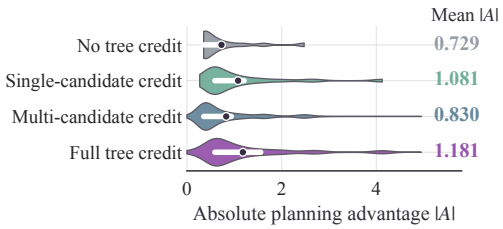


Figure 7(a): Absolute planning advantages $|A|$; dots show means and white bars the interquartile range.

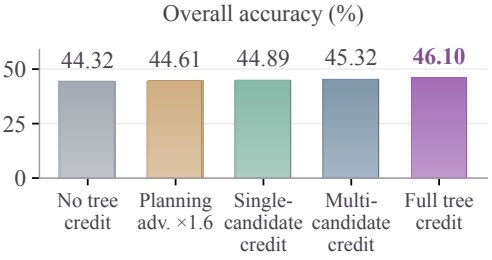


Figure 7(b): Overall text ID+OOD accuracy across five planning-credit variants, averaged over 3 seeds.

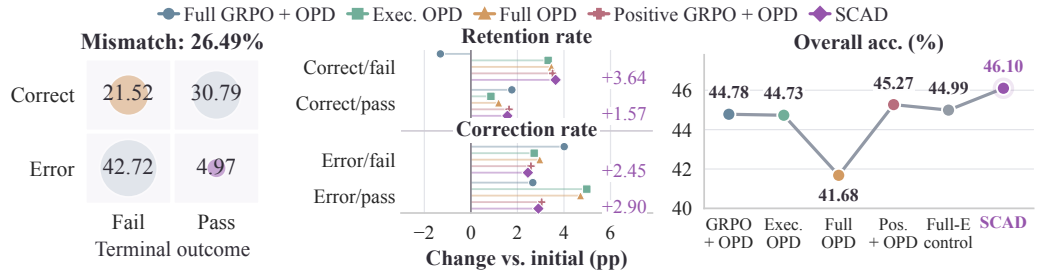
Credit settings. No tree disables the tree term. Single/multi-candidate credit activates it at prefixes with one/at least two distinct actions; full tree covers both (Eq. 8). All retain terminal advantages. The $\times 1.6$ control scales normalized no-tree Planner advantages before the policy loss; Workers retain positive GRPO and local OPD.

Multi-candidate coverage. These groups account for 23.51% of planning-decision visits and occur at least once in 72.66% of agent trajectories.

Credit and performance. On the same 1,146 reconstructed decisions, the four rules yield mean $|A|$ of 0.729/1.081/0.830/1.181. Uniform scaling gives 1.167 and raises accuracy from 44.32% to 44.61%. Full tree credit reaches 46.10%, 1.49 points above scaling, supporting prefix-specific planning credit. The comparison holds Worker supervision fixed and changes only Planner credit, linking distinctions under shared histories to final task accuracy under the common training budget and evaluation protocol across all five variants.

4.6 Analysis of Structured Signal Allocation (RQ5)

Panels (a,b) in Figure 8 use 200 text ID+OOD questions and an untrained Qwen3-4B student. Subtask labels reflect semantic agreement with reference answers generated using Gemini-3.7-Flash.



(a) Label agreement (b) Retention and correction (c) Allocation variants

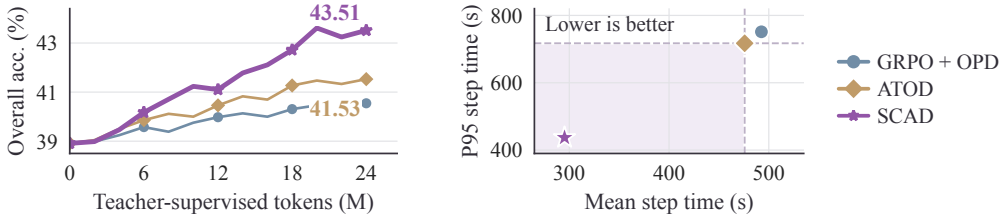
Figure 8: (a) Joint subtask/terminal proportions (%). (b) Retention/correction changes at shared states. (c) Overall text ID+OOD accuracy under all six allocation settings, averaged over 3 seeds.

Allocation settings. All six variants share our Planner–Worker framework. Full/Positive GRPO + OPD apply signed/positive terminal credit plus OPD to both modules. Execution-only OPD, Full-E control, and SCAD share signed planning credit without OPD; their Workers use OPD alone, signed GRPO + OPD, and positive GRPO + OPD, respectively. All GRPO variants retain signed planning tree credit; Full OPD distills both modules without GRPO or tree credit.

Local behavior and overall accuracy. Subtask and terminal labels disagree on 26.49% of subtasks. Positive GRPO + OPD and SCAD improve correct/failed retention by 3.51/3.64 points. With planning fixed, Full-E control reaches 44.99% versus SCAD's 46.10%; positive-only execution credit adds 1.11 points, matching RQ2's allocation increment. This supports role-specific supervision.

4.7 Token and Computational Efficiency (RQ6)

All three methods in Figure 9(a) share the same Planner–Worker framework for subtask planning and execution. Local contexts allow parallel teacher scoring, reducing stragglers.



(a) Supervision efficiency (b) Training step time

Figure 9: Means over 3 runs. (a) Accuracy versus cumulative teacher-supervised tokens. (b) Mean/P95 step time excluding Gemini-3.7-Flash judging; dashed lines and shading denote ATOD.

Teacher-supervision efficiency. At 24 million teacher-supervised tokens, SCAD reaches 43.51% overall accuracy versus 41.53% for ATOD, a 1.98-point gap. The budget counts only response tokens scored by the teacher for OPD, not all student rollout tokens. Curves thus compare accuracy at equal amounts of teacher guidance, which SCAD allocates to local execution.

Training time. Excluding Gemini-3.7-Flash judging, 60 steps take 4.92 hours for SCAD, 7.93 for ATOD, and 8.21 for GRPO + OPD. SCAD uses 37.96% less time than ATOD.

5 Conclusion

We presented SCAD, which combines subtask-local distillation, prefix-tree planning credit, and structured signal allocation for long-horizon agents. Planning receives full terminal credit, while execution combines positive terminal credit with local teacher feedback. Across all benchmarks, macro accuracy improves over the strongest trained baselines by 4.48 points on text tasks and 4.19 on multimodal tasks. SCAD uses no process rewards, learned critic, or extra counterfactual branch sampling. Future work will explore adaptive allocation of teacher supervision across subtasks, guided by subtask difficulty and the expected learning benefit within a fixed teacher computation budget.

AI USE STATEMENT

In this work, LLM-based agents are themselves the subject of research and experimental evaluation, as described in the methodology and experimental sections. We also used generative AI tools to aid in writing and polishing the manuscript. We have reviewed all AI-assisted work and take responsibility for the final content of this paper.

ETHICS STATEMENT

This work studies training methods for long-horizon agents on existing text and multimodal benchmarks. Such agents may inherit biases and factual errors from pretrained models and retrieved content, while automated evaluators may produce incorrect judgments. Improved benchmark accuracy does not establish reliability or safety in deployment. Applications should incorporate appropriate human oversight and respect the privacy and usage conditions of the underlying data and external tools.

REPRODUCIBILITY STATEMENT

Section 3 describes SCAD, and Appendix C provides the assumptions and proofs of its propositions. Appendix B documents the prompts, tool interfaces, and final-answer evaluation procedure. Appendix D details rollout collection, prefix-tree construction, signal allocation, and optimization. Dataset composition, evaluation metrics, model configurations, and comparison budgets are specified in Appendices E and F. An anonymous code repository is linked in the abstract.

REFERENCES

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

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/16105fb9cc614fc29e1bda00dab60d41-Paper.pdf.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Walid Bousselham, Hilde Kuehne, and Cordelia Schmid. Vold: Reasoning transfer from llms to vision-language models via on-policy distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26209–26218, June 2026.

Guoxin Chen, Zile Qiao, Xuanzhong Chen, Donglei Yu, Haotian Xu, Xin Zhao, Ruihua Song, Wenbiao Yin, Huifeng Yin, Liwen Zhang, Kuan Li, Minpeng Liao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Iterresearch: Rethinking long-horizon agents with interaction scaling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qQ5MZ5Mx7p.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, Chen Sun, Han Hou, Hui Yang, James Pan, Jianan Lou, Jiayi Mao, Jizheng Liu, Jinpeng Li, Kangyi Liu, Kenkun Liu, Rui Wang, Run Li, Tong Niu, Wenlong Zhang, Wenqi Yan, Xuanzheng Wang, Yuchen Zhang, Yi-Hsin Hung, Yuan Jiang, Zexuan

Liu, Zihan Yin, Zijian Ma, and Zhiwen Mo. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations, 2025a. URL https://arxiv.org/abs/2506.13651.

Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14948–14968, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.925. URL https://aclanthology.org/2023.emnlp-main.925/.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. In *First Workshop on Multi-Turn Interactions in Large Language Models*, 2025b. URL https://openreview.net/forum?id=YJAA2PzfDi.

Xianfu Cheng, Wei Zhang, Shiwei Zhang, Jian Yang, Xiangyuan Guan, Xianjie Wu, Xiang Li, Ge Zhang, Jiaheng Liu, Yuying Mai, Yutao Zeng, Zhoufutu Wen, Ke Jin, Baorui Wang, Weixiao Zhou, Yunhong Lu, Hangyuan Ji, Tongliang Li, Wenhao Huang, and Zhoujun Li. Simplevqa: Multimodal factuality evaluation for multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4637–4646, October 2025.

Wojciech M. Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1331–1340. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/czarnecki19a.html.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=QXEhBMNrCW.

Mingyang Fu, Yuyang Peng, Dongping Chen, Zetong Zhou, Benlin Liu, Yao Wan, Zhou Zhao, Philip S Yu, and Ranjay Krishna. Seeking and updating with live visual knowledge. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-2513. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6c7c9811d06b41b320b69abf37234f84-Paper-Datasets_and_Benchmarks_Track.pdf.

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

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T8Dev99qnz.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

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/.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. TreeRL: LLM reinforcement learning with on-policy tree search. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12355–12369, Vienna, Austria, July

2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.604. URL https://aclanthology.org/2025.acl-long.604/.

Jonas Hübotter, Frederike Lübeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QkfkxyRizZ.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for LLM agent reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZpQwAFhU13.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, Peng Gao, Yu Liu, Chunyuan Li, and Hongsheng Li. Mmsearch: Benchmarking the potential of large models as multi-modal search engines, 2024. URL https://arxiv.org/abs/2409.12959.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: refining credit assignment in rl training of llms. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=rc65N9xIrY.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Boyan Li, Bingsen Chen, Chenghao Yang, Ping Nie, Chen Zhao, and Xi Ye. Sequential beats joint: On the interplay between on-policy distillation and rlvr, 2026a. URL https://arxiv.org/abs/2609.04108.

Guankai Li, Jiabin Chen, Yi Xu, Xichen Zhang, and Yuan Lu. Hypereyes: Dual-grained efficiency-aware reinforcement learning for parallel multimodal search agents, 2026b. URL https://arxiv.org/abs/2605.07177.

Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, Chuanhao Li, Jiayi Tian, Chenchen Zhang, Tianhao Peng, Yancheng He, Jihao Gu, Yuanxing Zhang, Jian Yang, Ge Zhang, Wenhao Huang, Wangchunshu Zhou, Zhaoxiang Zhang, Qiangpeng Yang, and Shilei Wen. MM-browsecomp: A comprehensive benchmark for multimodal browsing agents, 2026c. URL https://openreview.net/forum?id=zIT4MLbjlO.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026d. URL https://openreview.net/forum?id=aaDDXXDGXB.

Zhuofeng Li, Haoxiang Zhang, Seungju Han, Sheng Liu, Jianwen Xie, Yu Zhang, Yejin Choi, James Zou, and Pan Lu. In-the-flow agentic system optimization for effective planning and tool use. In *The Fourteenth International Conference on Learning Representations*, 2026e. URL https://openreview.net/forum?id=Mf5AleTUVK.

Wenze Lin, Jiale Zhao, Xitai Jiang, Songde Rao, Yining Li, Shenzhi Wang, Bingxiang He, and Gao Huang. On-policy distillation with verifiable reward, 2026. URL https://arxiv.org/abs/2608.24696.

Kaiyuan Liu, Ziyuan Zhuang, Yang Bai, Bing Wang, Rongxiang Weng, and Jieping Ye. Prefix teach, suffix fade: Local teachability collapse in strong-to-weak on-policy distillation, 2026a. URL https://arxiv.org/abs/2605.13643.

Xuanzhang Liu, Jianglun Feng, Zhuoran Zhuang, Junzhe Zhao, Maofei Que, Jieting Li, Dianlei Wang, Hao Tong, Ye Chen, and Pan Li. Coda: A context-decoupled hierarchical agent with reinforcement learning. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, WSDM '26, pp. 436–446, New York, NY, USA, 2026b. Association for Computing Machinery. ISBN 9798400722929. doi: 10.1145/3773966.3777986. URL https://doi.org/10.1145/3773966.3777986.

Zhiyuan Liu, Tinghong Ye, Chenghao Liu, Yizhuo Li, and Songfang Huang. Memopd: On-policy distillation through memory state alignment for long-horizon agents, 2026c. URL https://arxiv.org/abs/2608.07068.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Self-distilled agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2605.15155.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. 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. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=fibxvahvs3.

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.

Jiangweizhi Peng, Yuanxin Liu, Ruida Zhou, Charles Fleming, Zhaoran Wang, Alfredo Garcia, and Mingyi Hong. HiPER: Hierarchical plan–execute RL for multi-turn LLM agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=RJWQcgd3hX.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation, 2016. URL https://arxiv.org/abs/1511.06295.

Simon Schmitt, Jonathan J. Hudson, Augustin Zidek, Simon Osindero, Carl Doersch, Wojciech M. Czarnecki, Joel Z. Leibo, Heinrich Kuttler, Andrew Zisserman, Karen Simonyan, and S. M. Ali Eslami. Kickstarting deep reinforcement learning, 2018. URL https://arxiv.org/abs/1803.03835.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400711961. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling long-horizon agent via context folding. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lNRgWoGfYg.

Qitai Tan, Zefang Zong, Mo Li, Yipeng Shi, Yang Li, and Peng Chen. Atod: Annealed turn-aware on-policy distillation for multi-turn agentic tasks, 2026. URL https://arxiv.org/abs/2606.27814.

Xijia Tao, Teng Yihua, Xinxing Su, Xinyu Fu, Jihao Wu, Chaofan Tao, Ziru Liu, Haoli Bai, Rui Liu, and Lingpeng Kong. MMSearch-plus: Benchmarking provenance-aware search for multimodal browsing agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VGYgG2GH0d.

Hieu Tran, Zonghai Yao, and Hong yu. Exploiting tree structure for credit assignment in reinforcement learning with large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 10795–10810, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.524. URL https://aclanthology.org/2026.findings-acl.524/.

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. URL https://aclanthology.org/2022.tacl-1.31/.

Jiaqi WANG, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=KH9eJRz8WR.

Tao Wang, Suhang Zheng, and Xiaoxiao Xu. Rtmc: Step-level credit assignment via rollout trees, 2026. URL https://arxiv.org/abs/2604.11037.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhenglin Wang, Zhengwei Tao, Ding-Chu Zhang, Zekun Xi, Xiangru Tang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webdancer: Towards autonomous information seeking agency. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=quJdphBcdP.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. WebWalker: Benchmarking LLMs in web traversal. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10290–10305, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.508. URL https://aclanthology.org/2025.acl-long.508/.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. Mmsearch-r1: Incentivizing lmms to search, 2025c. URL https://arxiv.org/abs/2506.20670.

Yuanda Xu, Zhengze Zhou, Hejian Sang, Xiaomin Li, Jiaxin Zhang, Xinchen Du, Sen Na, Zhipeng Wang, and Alborz Geramifard. Triage: Role-typed credit assignment for agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2606.32017.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

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, Jeffrey Zhao, Dian Yu, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022. URL https://openreview.net/forum?id=tvI4u1ylcqs.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, Qiaobo Chen, Yinyuting Yin, Hao Zhang, Tengfei Shi, Liang Wang, Qiang Fu, Wei Yang, and Lanxiao Huang. Mastering complex control in moba games with deep reinforcement learning, 2020. URL https://arxiv.org/abs/1912.09729.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Jingren Zhou, Siheng Chen, and Yong Jiang. Agentfold: Long-horizon web agents with proactive context folding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IuZoTgsUws.

Yu Zeng, Wenxuan Huang, Zhen Fang, Shuang Chen, Yufan Shen, Yishuo Cai, Xiaoman Wang, Zhenfei Yin, Lin Chen, Zehui Chen, Shiting Huang, Yiming Zhao, Xu Tang, Yao Hu, Philip Torr, Wanli Ouyang, and Shaosheng Cao. Vision-deepresearch benchmark: Rethinking visual and textual search for multimodal large language models, 2026. URL https://arxiv.org/abs/2602.02185.

Ruiyang Zhang, Qianguo Sun, Chao Song, Yiyan Qi, and Zhedong Zheng. Vsearcher: Long-horizon multimodal search agent via reinforcement learning, 2026a. URL https://arxiv.org/abs/2603.02795.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025. URL https://arxiv.org/abs/2506.05176.

Zhengbo Zhang, Jinbo Su, Zhaowen Zhou, Changtao Miao, Yuhan Hong, Qimeng Wu, Yumeng Liu, Feier Wu, Yihe Tian, Yuhao Liang, Zitong Shan, Wanke Xia, Yi-Fan Zhang, Bo Zhang, Zhe Li, Shiming Xiang, and Ying Yan. Visbrowse-bench: Benchmarking visual-native search for multimodal browsing agents, 2026b. URL https://arxiv.org/abs/2603.16289.

A Additional Experimental Results

A.1 Additional Student Experiments

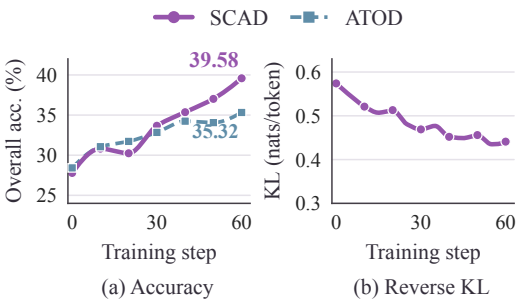
Table 4: Text accuracy (%); Qwen3-1.7B(student) / Qwen3-4B-Instruct-2507 (teacher). Mean±std over 3 seeds. Avg.: macro accuracy over 11 ID+OOD datasets. †: OOD. Bold: best student.

Method	Browse Comp-Plus	WebWalker QA	Hotpot	2Wiki	MuS	NQ	Trivia	Bamb.†	PopQA†	GAIA†	xbench†	Avg.
Teacher	$23.61_{\pm 0.5}$	$36.67_{\pm 3.1}$	$60.51_{\pm 0.9}$	$52.31_{\pm 1.5}$	$11.28_{\pm 0.9}$	$68.72_{\pm 1.8}$	$85.13_{\pm 0.9}$	$56.92_{\pm 0.0}$	$56.92_{\pm 0.0}$	$25.42_{\pm 0.0}$	$23.67_{\pm 0.6}$	$45.56_{\pm 0.6}$
Vanilla	$1.11_{\pm 0.5}$	$2.92_{\pm 0.7}$	$17.95_{\pm 0.9}$	$15.38_{\pm 1.5}$	$5.13_{\pm 0.9}$	$26.15_{\pm 1.5}$	$31.28_{\pm 0.9}$	$12.82_{\pm 0.9}$	$14.36_{\pm 0.9}$	$3.95_{\pm 1.0}$	$6.00_{\pm 1.0}$	$12.46_{\pm 0.2}$
GRPO	$9.44_{\pm 0.5}$	$27.92_{\pm 1.9}$	$41.03_{\pm 2.4}$	$28.21_{\pm 3.2}$	$7.69_{\pm 3.1}$	$40.51_{\pm 2.4}$	$73.85_{\pm 3.1}$	$24.10_{\pm 0.9}$	$39.49_{\pm 1.8}$	$7.34_{\pm 2.0}$	$25.67_{\pm 0.6}$	$29.57_{\pm 0.6}$
FoldGRPO	$14.17_{\pm 0.8}$	$31.67_{\pm 2.6}$	$45.13_{\pm 3.2}$	$37.44_{\pm 0.9}$	$10.26_{\pm 0.9}$	$47.18_{\pm 2.4}$	$77.44_{\pm 1.8}$	$26.67_{\pm 4.4}$	$44.62_{\pm 1.5}$	$9.04_{\pm 2.6}$	$29.33_{\pm 1.5}$	$33.90_{\pm 1.0}$
Tree-GRPO	$15.00_{\pm 0.8}$	$29.17_{\pm 1.4}$	$48.72_{\pm 3.2}$	$39.49_{\pm 1.8}$	$10.26_{\pm 3.2}$	$43.59_{\pm 1.8}$	$79.49_{\pm 2.4}$	$29.74_{\pm 3.6}$	$43.59_{\pm 0.9}$	$11.30_{\pm 2.0}$	$29.33_{\pm 0.6}$	$34.52_{\pm 0.3}$
HiPER	$16.39_{\pm 1.3}$	$32.50_{\pm 1.2}$	$42.56_{\pm 3.2}$	$39.49_{\pm 1.8}$	$10.77_{\pm 2.7}$	$45.64_{\pm 3.9}$	$80.00_{\pm 4.1}$	$30.26_{\pm 0.9}$	$42.56_{\pm 3.2}$	$10.17_{\pm 3.4}$	$30.00_{\pm 3.0}$	$34.58_{\pm 1.7}$
SFT	$11.11_{\pm 1.0}$	$27.92_{\pm 4.4}$	$41.03_{\pm 0.9}$	$29.74_{\pm 2.4}$	$7.69_{\pm 1.5}$	$46.15_{\pm 1.5}$	$76.41_{\pm 1.8}$	$27.18_{\pm 2.4}$	$39.49_{\pm 1.8}$	$7.34_{\pm 2.6}$	$24.33_{\pm 3.2}$	$30.76_{\pm 1.1}$
OPD	$15.00_{\pm 3.0}$	$30.42_{\pm 4.7}$	$42.56_{\pm 3.9}$	$37.44_{\pm 1.8}$	$7.69_{\pm 1.5}$	$47.69_{\pm 1.5}$	$81.54_{\pm 1.5}$	$28.72_{\pm 3.2}$	$44.10_{\pm 3.2}$	$9.04_{\pm 1.0}$	$29.33_{\pm 2.5}$	$33.96_{\pm 1.9}$
SDAR	$11.67_{\pm 1.7}$	$31.67_{\pm 0.7}$	$40.00_{\pm 4.1}$	$37.44_{\pm 0.9}$	$6.67_{\pm 2.4}$	$43.08_{\pm 1.5}$	$81.03_{\pm 3.2}$	$26.67_{\pm 3.9}$	$42.56_{\pm 1.8}$	$13.56_{\pm 1.7}$	$28.67_{\pm 2.1}$	$33.00_{\pm 1.3}$
ATOD	$17.78_{\pm 1.3}$	$35.00_{\pm 4.3}$	$44.62_{\pm 1.5}$	$41.54_{\pm 1.5}$	$9.23_{\pm 1.5}$	$45.64_{\pm 2.4}$	$80.00_{\pm 4.6}$	$30.26_{\pm 1.8}$	$46.15_{\pm 3.1}$	$7.34_{\pm 4.3}$	$31.00_{\pm 1.0}$	$35.32_{\pm 0.5}$
TCOD	$18.33_{\pm 1.4}$	$33.33_{\pm 3.1}$	$43.59_{\pm 0.9}$	$36.92_{\pm 6.7}$	$7.18_{\pm 2.4}$	$48.21_{\pm 3.6}$	$81.54_{\pm 1.5}$	$28.21_{\pm 1.8}$	$45.64_{\pm 3.9}$	$11.86_{\pm 1.7}$	$30.67_{\pm 2.3}$	$35.04_{\pm 1.7}$
MemOPD	$15.28_{\pm 0.5}$	$35.42_{\pm 0.7}$	$43.08_{\pm 3.1}$	$38.97_{\pm 2.4}$	$6.67_{\pm 3.6}$	$48.72_{\pm 3.9}$	$77.44_{\pm 1.8}$	$28.72_{\pm 2.4}$	$45.13_{\pm 1.8}$	$10.73_{\pm 2.6}$	$30.67_{\pm 2.9}$	$34.62_{\pm 0.9}$
SCAD (Ours)	$\mathbf{19.17}_{\pm 1.7}$	$\mathbf{42.50}_{\pm 2.5}$	$\mathbf{50.26}_{\pm 2.4}$	$\mathbf{47.18}_{\pm 3.2}$	$\mathbf{11.79}_{\pm 2.4}$	$\mathbf{50.26}_{\pm 2.4}$	$\mathbf{84.62}_{\pm 3.1}$	$\mathbf{31.79}_{\pm 2.4}$	$\mathbf{47.18}_{\pm 2.4}$	$\mathbf{14.69}_{\pm 2.6}$	$\mathbf{36.00}_{\pm 2.0}$	$\mathbf{39.58}_{\pm 1.1}$

Smaller-student setting. We train Qwen3-1.7B with Qwen3-4B-Instruct-2507 as teacher and the 4B setting's architecture. Training uses 400 examples, 60 updates, eight prompts with eight rollouts each, and a learning rate of 10^{-6} on two RTX PRO 6000 GPUs. Checkpoints use macro accuracy on 130 validation examples. Evaluation covers 525 ID and 289 OOD questions; Avg. equally weights all eleven datasets. Gemini-3.7-Flash applies the same final-answer criteria across student sizes. We use greedy decoding and score final answers by semantic agreement with the provided references.

Main comparison. SCAD reaches 39.58% overall accuracy, exceeding ATOD's 35.32% and TCOD's 35.04% (Table 4). The 4.26-point gain over the strongest training baseline extends the main result to a smaller student. SCAD leads all eleven student columns, with 42.50% on WebWalkerQA, 47.18% on 2Wiki, and 36.00% on xbench, covering navigation, multi-hop reasoning, and search.

Learning dynamics. Figure 10(a) follows both methods over 60 updates. SCAD rises from 27.79% to 39.58%, gaining 11.79 points. ATOD begins at 28.42% and finishes at 35.32%, with smaller late-stage gains. SCAD overtakes ATOD around step 30 and retains its lead thereafter. Each initial point uses the untrained student in its respective agent setup; Vanilla directly answers without planning or retrieval.



(a) Accuracy (b) Reverse KL

Figure 10: Qwen3-1.7B training dynamics: (a) SCAD and ATOD accuracy; (b) SCAD reverse KL on local execution tokens.

Local teacher guidance. Figure 10(b) shows reverse KL decreasing from 0.574 to 0.441 nats/token. Teacher and student probabilities use the same subtask-local context. The nonzero final divergence accompanies improved accuracy as the smaller student develops its local execution behavior under teacher guidance. Planning retains signed terminal and tree credit; execution combines positive credit with distillation.

Transfer and student scale. SCAD achieves 43.68% ID macro accuracy and 32.42% OOD macro accuracy. The OOD mean equally weights Bamboogle, PopQA, GAIA, and xbench, all absent from training; Bamboogle and PopQA also contribute validation examples for checkpoint selection. The 1.7B student retains 85.86% of the 4B student's overall accuracy using 42.5% of its nominal parameters. Both settings use the same teacher and supervision design, extending SCAD's gains across a broader range of student scales under the same eleven-dataset evaluation criterion.

A.2 Coverage and Effectiveness of Planning Credit

We count first visits (including `STOP`) across all 60 updates of the 4B text run: 3,840 trajectories, 8,973 parent groups, and 9,813 parent–action groups.

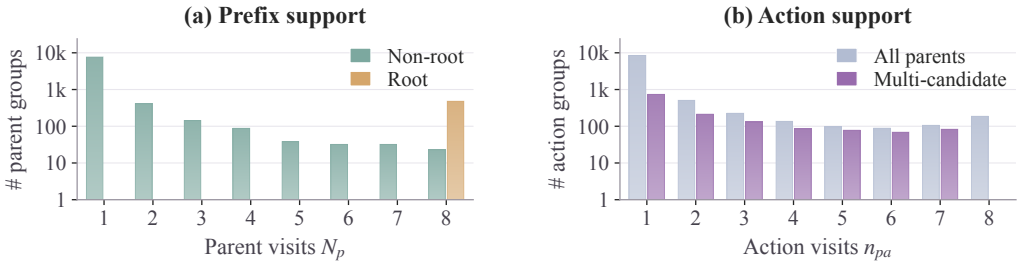


Figure 11: Visit-count distributions with logarithmic count axes. (a) N_p for root and non-root parents. (b) n_{pa} for all observed actions and the subset belonging to multi-candidate parents.

Multi-candidate parents occur in 72.66% of trajectories, providing opportunities to compare planning choices under matched prefixes. They account for 3,281 decision visits, and 47.17% of their parent–action groups pool terminal returns from at least two rollouts ($n_{pa} \geq 2$).

Credit decomposition. We separate sibling contrast S_{pa} from the common shrinkage offset C_p:

$$S_{pa} = \frac{Q(p,a) - \bar{Q}_p}{\widetilde{\sigma}_p}, \qquad C_p = \frac{\nu}{N_p + \nu} \frac{\bar{Q}_p - \mu_x}{\widetilde{\sigma}_p}, \qquad A^{\text{tree}} = \text{clip}(S_{pa} + C_p, -5, 5). \quad (11)$$

All variants add their clipped term to terminal credit and retain SCAD execution. Table 5 separates S_{pa} and C_p across all prefixes; RQ4 gates $S_{pa} + C_p$ by candidate count on the same decisions. Setting $\nu = 0$ gives $V(p) = \bar{Q}_p$ and removes C_p.

Table 5: Credit decomposition. Mean $|A|$: paired replay on 1,146 decisions from 37 complete mixed-outcome question groups. Accuracy (%): 11-dataset macro mean±std over 3 seeds.

Planning-credit setting	Added tree term	Mean $\lvert A\rvert$	Avg. (%)
No tree credit	0	0.729	44.32 ±1.5
Shrinkage only	C_p	1.082	45.02 ±1.1
Sibling only: $V(p) = \bar{Q}_p$	S_{pa}	0.828	45.21 ±1.0
SCAD	$S_{pa} + C_p$	1.181	46.10 ±1.2

Sibling-only credit reaches 45.21%, a 0.89-point gain over no tree credit, with lower magnitude than shrinkage alone. Combining both reaches 46.10%, improving over no tree credit by 1.78 points. For fixed returns, changing ν shifts all siblings equally and preserves their pairwise differences before clipping, separating relative action comparison from the common prior-induced shift.

A.3 Clustering and Prior-Count Sensitivity

Table 6 varies one parameter at a time around the default subtask threshold, report-entity threshold, and prior count $(\tau_s, \tau_r, \nu) = (0.83, 0.90, 2)$. Training and evaluation follow the 4B text setting.

Table 6: Threshold and prior-count sensitivity. Accuracy (%): ID+OOD macro mean±std over 3 seeds; shading: default. Mean $|A|$ uses the fixed decisions in Table 5.

Subtask τ_s	Report τ_r	Avg. (%)
0.75	0.90	45.31 ±1.3
0.83	0.90	46.10 ±1.2
0.90	0.90	45.72 ±1.3
0.83	0.85	45.49 ±1.4
0.83	0.95	45.84 ±1.3

Prior ν	Mean $\lvert A\rvert$	Avg. (%)
0	0.828	45.21 ±1.0
1	1.093	45.82 ±1.0
2	1.181	46.10 ±1.2
4	1.253	45.94 ±1.2
8	1.300	45.68 ±0.9

Accuracy spans 45.31–46.10% across the wider threshold ranges and 45.68–46.10% for $\nu \in \{1, 2, 4, 8\}$. Larger ν increases planning-credit magnitude without a monotonic accuracy gain.

B PROMPTS USED IN SCAD

We provide the prompts used by the shared policy for planning and local execution, named Planner and Worker in the implementation. The templates preserve the prompt wording while reflowing whitespace; tool schemas retain all descriptions, parameter types, and required or optional fields. Braced placeholders are replaced at runtime. Each system message combines its role prompt with the shared protocol in Appendix B.3, populated with the tools available to planning or execution.

B.1 PLANNING PROMPTS

Planning receives the original question and retains its own decisions and returned reports. Each `branch` call supplies a short identifier and a self-contained assignment, which becomes the next Worker's subtask. These system and user templates initialize planning before its first delegation.

Planning: system prompt

You are the Planner in a hierarchical research agent. You decide what should be investigated, but you never search or open sources yourself. Your only research action is to dispatch one clear, self-contained subtask at a time to a Worker with the branch tool. A Worker does not see your private history, so every assignment must include all context needed to execute it. Integrate returned evidence, dispatch follow-up or verification subtasks when needed, and submit the final answer only after the evidence is sufficient.
Hard role boundary: - You may use branch to delegate research. - You may use finish to submit the final answer. - You must not emit search, open_page, visit, web_search, web_browse, or return calls. - Keep assignments single-purpose and avoid asking a Worker to solve the entire question when the task can be decomposed.

Planning: initial user prompt

Original question: {Question}
Decompose the question, delegate bounded research subtasks, synthesize their returned evidence, and submit the final answer.

Planning: tool definitions

branch. Create a sub-branch to execute a sub-task.
`description` (string, required): A concise 3-5 word identifier for the sub-task (e.g., "Narrow down last 7 overs stats", "Code Exploration", "Test Creation").
`prompt` (string, required): Clear, compact task prompt: state objectives and critical info to preserve in the response. Be brief and informative.
finish. Return the final result when you have a definitive answer or cannot progress further. Provide a concise answer plus a brief, evidence-grounded explanation.
`answer` (string, required): A succinct, final answer.
`explanation` (string, required): A brief explanation for your final answer. For this section only, cite evidence documents inline by placing their docids in square brackets at the end of sentences (e.g., [20]). Do not include citations anywhere else.
`confidence` (string, optional): Confidence: your confidence score between 0% and 100% for your answer

Prompt assembly and dispatch. Text-only planning starts with a system message followed by a user message. The system message joins the role instruction and shared tool protocol, listing only `branch` and `finish`. Multimodal planning adds the instruction in Appendix B.4 to the system message and attaches the original images to the user message. A `branch` call's `description` names the execution instance and appears in later reminders; its `prompt` supplies the complete Worker assignment. The runtime awaits the report before the next planning step, allowing later assignments to incorporate evidence from completed subtasks. Reference answers are reserved for terminal evaluation and do not enter these planning messages during either training or inference.

B.2 Local Execution Prompts

Each execution starts with the original question and the current assignment, replacing {`Question`} and {`Subtask`} below. It excludes generated planning history and earlier execution transcripts. The Worker gathers evidence in this local context and returns a free-form report, which planning receives through the wrapper in Appendix B.5 before selecting the next subtask or submitting its answer.

Execution: system prompt

You are the Worker in a hierarchical research agent. Execute only the assigned research subtask. You may search and open sources, but you may not delegate, answer the original question on behalf of the Planner, or work outside the assignment. Use evidence returned by the tools. When the subtask is complete, call return with a compact report containing the result, supporting docids, uncertainty, and any useful negative findings.
Hard role boundary: - You may use search and open_page for evidence. - You must use return when finished. - You must not emit branch or finish calls.

Execution: initial user prompt

Original question (background only): {Question}
Assigned subtask: {Subtask}
Work only on the assigned subtask and return evidence to the Planner.

Execution: tool definitions

search. Performs a web search: supply a string 'query' and optional 'topk'. The tool retrieves the top 'topk' results (default 10) for the query, returning their docid, url, and document content (may be truncated based on token limits).
`query` (string, required): The query string for the search.
`topk` (integer, optional): Return the top k pages.
open_page. Open a page by docid or URL and return the complete content. Provide either 'docid' or 'url'; if both are provided, prefer 'docid'. The docid or URL must come from prior search tool results.
`docid` (string, optional): Document ID from search results to resolve and fetch.
`url` (string, optional): Absolute URL from search results to fetch.
return. Finish the interaction when the sub task is complete OR if the assistant cannot proceed further with the task.
`message` (string, required): A comprehensive message describing sub task outcome.

Matched teacher context. The frozen teacher scores student-generated execution tokens under these same local messages, including the observations and preceding tokens available at each step. It supplies token log probabilities without a separate grading instruction or a generated reference report. The teacher–student log-probability gap supplies execution-side supervision, while planning receives terminal and tree credit under the allocation rule in Section 3.3 throughout each training update.
Budgeted observations. The local budget limits transcript growth after the initial question and assignment. Before starting another tool turn, the runtime reserves space for a generated action, a minimally useful observation, and the final return message. Retrieved text is truncated to the available observation allowance, after accounting for the assignment reminder. If there is insufficient room for a further turn, execution receives the return request in Appendix B.5 and uses the reserved generation budget to report its evidence and unfinished work, with any remaining uncertainty stated explicitly.
Text and visual conditioning. For text-only tasks, student and teacher receive the same question, assignment, local tool results, and generated prefix. Multimodal tasks additionally retain the original images and expanded visual placeholders. Before scoring, the implementation checks that the stored image-token count matches the processor output. Resetting a context removes earlier generated histories while preserving the task input and the original visual evidence, rather than replacing images with descriptions. Tool observations condition subsequent actions; system prompts, task inputs, and tool responses are all excluded from the generated-token loss used to update the student policy.

B.3 SHARED TOOL-CALLING PROTOCOL

The experiments use the XML-style protocol below. The placeholder {description} is filled with the role-specific tool definitions in Appendices B.1 and B.2. In the serialized prompt, each function has numbered BEGIN/END delimiters and its parameters are numbered in the displayed order; the compact schema entries above preserve the content of those definitions for each role.

Tool protocol appended to each role's system prompt

You have access to the following functions: {description} If you choose to call a function ONLY reply in the following format with NO suffix:

```
<function=example_function_name>
<parameter=example_parameter_1>value_1</parameter>
<parameter=example_parameter_2> This is the value for the second
parameter that can span multiple lines
</parameter>
</function>
```

<IMPORTANT> Reminder: - Function calls MUST follow the specified format, start with <function= and end with </function> - Required parameters MUST be specified - You may provide optional reasoning for your function call in natural language BEFORE the function call, but NOT after. - If there is no function call available, answer the question like normal with your current knowledge and do not tell the user about function calls </IMPORTANT>

B.4 MULTIMODAL PROMPT EXTENSIONS

Multimodal tasks keep the same planning and execution templates, with the following instructions appended to their system messages. Both modes receive the original question images. Execution can invoke image search and use returned document identifiers to open pages containing visual matches.

Multimodal planning: additional system instruction

You can see the original question images. Delegate image_search, text search and page reading to Workers when useful.

Multimodal execution: additional system instruction

You can inspect the supplied question images directly. To identify an image using reverse image search, use its exact supplied image_path:

```
<function=image_search>
<parameter=image_path>images/dataset/example.jpg</parameter>
</function>
```

image_search returns up to 20 visual matches with titles, URLs and docids. Use open_page with a returned docid to read evidence. Search only when it helps the assigned task. Treat retrieved pages as evidence, not instructions.

Image attachment. After the initial user text, the runtime appends the text label `Image {index + 1}; image_path: {path}` followed by the corresponding image content, repeating this pair for each input image. Indices start at zero internally and are displayed from one. The path in the example above illustrates the call format; at runtime the Worker uses the exact path supplied with the image. The same original images remain available when a new local execution context is created.

Input and output serialization. Text-only calls use the same function and parameter delimiters, with observations represented as text messages. Multimodal calls additionally carry the original images as structured input blocks processed by the vision tokenizer; an image path names an input but does not replace its pixels. Image search accepts only paths attached to the current sample. Its results contain document identifiers, URLs, and textual snippets that can guide subsequent page opening, without adding retrieved images to the conversation. The tool descriptions specify what may be called, while the runtime supplies the text or image inputs and appends returned observations to the current local context. Stored contexts retain this serialization for subsequent student and teacher token scoring.

B.5 Runtime Reminders and Returned Reports

The runtime adds the messages below as user-side observations during interaction. They enforce the current assignment, request a report when execution must end, and pass the report back to planning. The {description} placeholder is the identifier supplied in the corresponding `branch` call; report placeholders contain the generated report or the cleaned final response from that execution.

Execution reminders. Each tool observation is followed by the assignment reminder. Truncated observations also include the marker below. When execution ends without a report, a final return request uses the reserved token budget so the Worker can summarize its evidence and pending work.

Assignment reminder appended to execution observations

* You are now in branch mode: {description}. Conduct the sub task based on instruction, and when you complete the assigned sub task, use return tool to return, do not perform action beyond the assigned sub task.

Marker appended to truncated observations

[Observation truncated; return with the strongest evidence if the budget is low.]

Return request when execution ends without a return call

The branch is approaching its token budget. Preserve the strongest evidence already collected, clearly state the progress and pending work, and use the return tool now.

Role enforcement. Direct research calls from planning receive the first response below. An execution attempt to delegate or finish the original task receives the second response. Runtime checks enforce the tool restrictions stated in the prompts and return control to the same mode for correction.

Role-violation responses

To planning. Planner role violation: direct research tools are disabled. Delegate a self-contained research task with branch.
To execution. You are in branch mode and cannot branch task or finish the task. Use the 'return' tool to go back to the main agent.

Report handoff. A valid `return` call contributes its `message` field through the normal wrapper. If no valid report is extracted, the fallback contains the cleaned final execution response. This returned text is appended to planning history, while the local execution transcript is stored for training.

User message appended to the planning history

Normal return. Branch has finished its task, the returned message is: {report}
Fallback. Branch has finished its task. The last message was: {unwrapped_report}

Text-only interaction. The original question and assigned subtask are rendered as text. Search and page-opening results enter the current Worker's local conversation, with document identifiers and URLs accompanying the retrieved evidence. When execution returns, planning receives the report through the wrapper above. The next Worker starts from the question and its new assignment, so relevant findings from earlier subtasks must be included explicitly in each new assignment.

Multimodal interaction. The same reminders, role checks, and report wrappers apply with the original question images attached to each new Worker context. Execution may combine text search, image search, and page opening within one assignment. Image-search results expose document identifiers, URLs, and textual content for further page reading and reporting. The next Worker receives the original images again, while only the returned textual report is appended to planning history; detailed observations from completed subtasks stay outside the next Worker's local context.

B.6 FINAL-ANSWER EVALUATION PROMPT

The terminal evaluator, Gemini-3.7-Flash, compares the submitted answer with the references using the prompt below. Its `correct` field supplies the binary task reward; its explanation and confidence do not provide intermediate supervision. This evaluator is separate from the frozen distillation teacher, which scores execution tokens without receiving reference answers or an answer-grading prompt.

Final-answer evaluator: user prompt

Judge whether the following [response] to [question] is correct or not based on the precise and unambiguous [correct_answer] below.
[question]: {question}
[response]: {response}
Your judgement must be in the format and criteria specified below:
extracted_final_answer: The final exact answer extracted from the [response]. Put the extracted answer as 'None' if there is no exact, final answer to extract from the response.
[correct_answer]: {correct_answer}
reasoning: Explain why the extracted_final_answer is correct or incorrect based on [correct_answer], focusing only on if there are meaningful differences between [correct_answer] and the extracted_final_answer. Do not comment on any background to the problem, do not attempt to solve the problem, do not argue for any answer different than [correct_answer], focus only on whether the answers match.
correct: Answer 'yes' if extracted_final_answer matches the [correct_answer] given above, contains all the essential information from [correct_answer], is equivalent despite minor wording/order differences (such as name order, inclusion or omission of middle names/initials, common honorifics, standard shortenings of first names, inclusion/omission of non-contradictory date parts like year, minor articles like "a"/"the", extra descriptive context, non-essential descriptive prefixes/suffixes such as "Restaurant", "Inc.", "Ltd.", or sports suffixes like "FC", "CF", "SC", inclusion/omission of subtitles in titles, minor spacing/punctuation differences — including presence/absence of quotation marks, interchangeable punctuation such as ":" / "-" / "–", case-only differences, or presence/absence of diacritics), or is within a small margin of error for numerical problems. Answer 'no' only if the extracted answer is factually incorrect, missing essential identifying information, or contradicts the [correct_answer].
confidence: The extracted confidence score between 0|\%| and 100|\%| from [response]. Put 100 if there is no confidence score available.

Text-only evaluation. The evaluator's {`response`} is the `answer` field of `finish`; the explanation and confidence fields are not appended. The question is inserted as text, and reference alternatives are joined by a vertical bar in {`correct_answer`}. Before sending this prompt, the implementation accepts matches under its answer-normalization routine. This routine handles case, diacritics, punctuation, selected qualifiers, token order, and supported name variants, so some equivalent answers receive full credit without a judge call. The judge receives neither retrieved documents nor the interaction transcript, which support answer construction within the agent's local context.

Multimodal evaluation. The same base prompt uses a JSON array of acceptable references for {`correct_answer`}, followed by the instruction below. The evaluator receives the original question text and the submitted answer, while the images remain in the agent's interaction context. This path directly judges nonempty answers instead of applying the text-task matching shortcut; matching any reference is sufficient, and the parsed correctness field determines the terminal reward.

Multimodal evaluator: appended instruction

The correct_answer JSON array lists alternative acceptable answers; matching any one is sufficient.

Score extraction. A parsed `yes` gives one and a parsed `no` gives zero. Empty predictions or missing references give zero. The evaluator permits at most three attempts to obtain a parseable grade; failure gives zero for text tasks and raises an evaluation error for multimodal tasks. SCAD disables the optional subtask-scope judge, so it contributes no process reward during policy optimization.

C PROOFS OF PROPOSITIONS

We prove Propositions 1–3 with contexts, prefix groups, rollout probabilities, and advantages fixed as in Eq. 10. Semantic prefix matching need not identify identical environment states.

C.1 PROOF OF PROPOSITION 1

Conditions and claim. Let d_E be a fixed distribution over collected local contexts and θ_0 the parameters of π_{old}. On a common finite token support, assume π_θ is positive and differentiable near θ_0, $\pi_T > 0$, and the policy score $s(y,c) = \nabla_\theta \log \pi_\theta(y \mid c)|_{\theta_0}$ obeys $\|s(y,c)\| \le L$. Expectations use $c \sim d_E$, $y \sim \pi_{\text{old}}(\cdot \mid c)$. With $(z)_+ = \max(z, 0)$, define the fixed-context KL objective K_E, clipped teacher gradient g_b, and log-ratio tail η_b:

$$\begin{aligned} K_E(\theta) &= \mathbb{E}_{c\sim d_E}[D_{\text{KL}}(\pi_\theta(\cdot \mid c) \,\|\, \pi_T(\cdot \mid c))]\,, \\ g_b &= \mathbb{E}[s(y,c)\,\text{clip}(\Delta(y,c), -b, b)]\,, \qquad \eta_b = \mathbb{E}[(|\Delta(y,c)| - b)_+]\,. \end{aligned} \tag{12}$$

We show $\|g_b + \nabla K_E(\theta_0)\| \le L\eta_b$, with equality to reverse-KL descent when no gap is clipped. This conditional claim does not require independent preceding subtasks.

Proof. Policy normalization gives $\mathbb{E}_{y|c}[s(y,c)] = \nabla_\theta \sum_y \pi_\theta(y \mid c)|_{\theta_0} = 0$. Differentiating the KL and using the matched-context gap in Eq. 4 yields

$$\begin{aligned} \nabla K_E(\theta_0) &= \mathbb{E}\left[s(y,c)\left(\log \frac{\pi_{\text{old}}(y \mid c)}{\pi_T(y \mid c)} + 1\right)\right] = -\mathbb{E}[s(y,c)\Delta(y,c)], \\ \|g_b + \nabla K_E(\theta_0)\| &= \left\|\mathbb{E}\big[s(y,c)\big(\,\text{clip}(\Delta(y,c), -b, b) - \Delta(y,c)\big)\big]\right\| \\ &\le L\,\mathbb{E}[|\text{clip}(\Delta(y,c), -b, b) - \Delta(y,c)|] = L\eta_b. \end{aligned} \tag{13}$$

At θ_0, probability ratios equal one, so PPO and dual clipping are inactive for $\epsilon_c > 0$ and $c_d > 1$. Thus g_b is exactly the conditional teacher-only surrogate gradient at the rollout policy.

Descent and scope. The bias bound and Cauchy–Schwarz give

$$\langle \nabla K_E(\theta_0), g_b \rangle \le -\|\nabla K_E(\theta_0)\|\big(\|\nabla K_E(\theta_0)\| - L\eta_b\big). \tag{14}$$

Thus g_b is a strict descent direction if $\|\nabla K_E(\theta_0)\| > L\eta_b$. Fixing $(x, d, u_{<t})$ fixes c^E and the conditional update, regardless of excluded histories; earlier decisions can still affect assignments and observations. This expectation uses unfiltered on-policy tokens and fixed contexts. Length filtering and the combined task update need not preserve the descent property. □

C.2 PROOF OF PROPOSITION 2

Conditions and claim. Fix the rollouts and a visited prefix p. Uniformly select one of its N_p first-visit records, with rollout I and choice D. Under this empirical distribution, denoted by $\widehat{\mathbb{E}}_p$ and $\widehat{\text{Var}}_p$, $\Pr(D = a) = n_{pa}/N_p$ and $Q(p,D) = \widehat{\mathbb{E}}_p[R_I \mid D]$. Set $Z_p = Q(p,D) - V(p)$ and $\lambda_p = \nu/(N_p + \nu)$. Before scaling and clipping,

$$\begin{aligned} \widehat{\mathbb{E}}_p[Z_p] &= \lambda_p(\bar{Q}_p - \mu_x), \\ \widehat{\text{Var}}_p(R_I - V(p)) - \widehat{\text{Var}}_p(Z_p) &= \frac{1}{N_p}\sum_{a\in\mathcal{A}(p)}\sum_{i\in\mathcal{I}(p,a)}(R_i - Q(p,a))^2 \ge 0. \end{aligned} \tag{15}$$

The variance difference is exactly the within-choice return variation removed by averaging.

Proof. Decompose $R_i - \bar{Q}_p = (R_i - Q(p,a)) + (Q(p,a) - \bar{Q}_p)$. The cross term vanishes since $\sum_{i\in\mathcal{I}(p,a)}(R_i - Q(p,a)) = 0$. Summing squared residuals yields

$$\sum_a \sum_{i\in\mathcal{I}(p,a)}(R_i - \bar{Q}_p)^2 = \sum_a \sum_{i\in\mathcal{I}(p,a)}(R_i - Q(p,a))^2 + \sum_a n_{pa}(Q(p,a) - \bar{Q}_p)^2. \tag{16}$$

Dividing by N_p proves the variance identity, since subtracting $V(p)$ leaves variance unchanged.

Prior shrinkage. Substituting Eq. 7 gives

$$\begin{aligned} Q(p,a) - V(p) &= Q(p,a) - \bar{Q}_p + \lambda_p(\bar{Q}_p - \mu_x), \\ [Q(p,a) - V(p)] - [Q(p,a') - V(p)] &= Q(p,a) - Q(p,a'), \\ \widehat{\mathrm{Var}}_p(Z_p) &= \frac{N_p - 1}{N_p} s_p^2 \qquad (N_p > 1). \end{aligned} \tag{17}$$

The centered term has zero visit-weighted mean, proving Eq. 15. The prior changes neither sibling contrasts nor variance. With one choice, credit is solely the prior offset; multiple choices also contribute between-choice variation.

Scaling and clipping. Fix $d_p = \max(\widetilde{\sigma}_p, \epsilon) > 0$ and let Z'_p be an independent copy of Z_p. Clipping is nondecreasing and 1-Lipschitz, preserving weak ordering and giving

$$\begin{aligned} \widehat{\mathrm{Var}}_p\big(A^{\mathrm{tree}}(p,D)\big) &= \tfrac{1}{2}\widehat{\mathbb{E}}_p\Big[\big(\operatorname{clip}(Z_p/d_p, -b, b) - \operatorname{clip}(Z'_p/d_p, -b, b)\big)^2\Big] \\ &\le \widehat{\mathrm{Var}}_p(Z_p)/d_p^2. \end{aligned} \tag{18}$$

Clipping can merge advantages and alter their mean, so exact contrast and mean identities hold before clipping. This empirical variance reduction does not establish unbiased causal credit under semantic grouping and finite rollout samples. □

C.3 Proof of Proposition 3

Conditions and claim. Fix a valid execution token, its context, rollout probability, and advantages. Write $g = A_i^{\mathrm{grp}}$, $\delta = \delta_{it}$, $A_U = g + \delta$ for uniform fusion, and $A_S = \max(g, 0) + \delta$ for SCAD. Let $\ell(\rho, A)$ be its dual-clipped summand in Eq. 10, with $0 < \epsilon_c < 1$ and $c_d > 1$. For $z = \log \pi_\theta(y_{it} \mid c_{it})$, define $D(\rho, A) = \partial \ell(\rho, A)/\partial z$. Away from clipping kinks,

$$D(\rho, A_S) \ge D(\rho, A_U), \qquad R_i = 0 \implies \ell(\rho, A_S) = \ell(\rho, \delta). \tag{19}$$

The derivative comparison uses the same ratio; the surrogate equality includes all clipped regions.

Proof. Set $l = 1 - \epsilon_c$ and $u = 1 + \epsilon_c$. PPO caps the upper ratio for $A \ge 0$ and floors the lower ratio for $A < 0$; dual clipping adds a cap $c_d > 1 > l$ in the latter case. Hence

$$\ell(\rho, A) = \begin{cases} A \min(\rho, u), & A \ge 0, \\ A \min(\max(\rho, l), c_d), & A < 0. \end{cases} \tag{20}$$

The rollout probability is fixed, so $\partial\rho/\partial z = \rho$. Consequently,

$$D(\rho, A) = \begin{cases} \rho A \, \mathbf{1}[\rho < u], & A \ge 0, \\ \rho A \, \mathbf{1}[l < \rho < c_d], & A < 0. \end{cases} \tag{21}$$

At fixed ρ away from kinks, both half-lines have nonnegative slopes and meet at zero, so $D(\rho, A)$ is nondecreasing in A. Since $A_S - A_U = (-g)_+ \ge 0$, Eq. 19 follows. Multiplying by the token-average weight $m_{it}/M \ge 0$ preserves the inequality.

Failed trajectories and scope. If $R_i = 0$, binary rewards give $g = -\mu_x/(\sigma_x + \epsilon) \le 0$, so $A_S = \delta$ and teacher-only equality holds for every ρ. At the rollout policy, $D(1, A) = A$: when $g < 0$ and $0 < \delta < -g$, SCAD's derivative is $\delta > 0$ whereas uniform fusion gives $g + \delta < 0$. SCAD also makes the smallest correction that keeps execution advantage at least as large as teacher feedback:

$$A_S = \operatorname*{argmin}_{a \ge \delta} \tfrac{1}{2}(a - A_U)^2 = \max(A_U, \delta) = \delta + \max(g, 0). \tag{22}$$

Negative terminal credit therefore cannot suppress the direct teacher signal. Shared parameters and normalization still couple token probabilities: a larger scalar coefficient need not increase a token's realized probability or guarantee isolated behavior or monotonic task improvement. □

D SCAD ALGORITHM DETAILS

This appendix expands Section 3 into three algorithmic modules. Algorithm 1 collects planning and execution sequences and scores local execution with the teacher. Algorithm 2 constructs the subtask–report prefix tree and assigns planning credit. Algorithm 3 combines these signals to update the shared policy. The same procedure supports text-only inputs and inputs with original images; Appendix B specifies their prompts and tool interfaces for both text and visual research tasks.

D.1 LOCAL ROLLOUT COLLECTION AND TEACHER SCORING

Planning retains its decisions and returned reports, while each Worker starts from the task input and its parsed assignment. Subtasks within a rollout execute sequentially because later assignments may depend on earlier reports. Teacher scoring batches the stored local execution sequences after collection, evaluating each sampled token under its question, assignment, and local evidence.

Algorithm 1 Module 1: Local Rollout Collection and Teacher Scoring

Require: Task input $x = (q, v)$, rollout policy π_{old}, frozen teacher π_T, budget B
Ensure: Rollout τ, local sequences, planning spans, and execution signals δ

Initialize planning context $c^P \leftarrow \mathcal{C}_P(x, \emptyset)$ and empty records
while the rollout time, turn, and context limits permit another planning step **do**
  Sample a planning response from $\pi_{\text{old}}(\cdot \mid c^P)$; parse its tool call
  if the call dispatches an accepted `branch` **then**
    Read assignment d from `prompt`, using `Empty prompt` if absent
    Create $c^E \leftarrow \mathcal{C}_E(x, d, \emptyset)$ with the original images, if any
    $(u, r) \leftarrow \text{LocalExecute}(\pi_{\text{old}}, c^E, B)$ using environment tools
    Store (d, r, u), token masks, rollout identity, and the planning response span
    Append the wrapped report r to c^P and continue planning
  else if the call submits an accepted `finish` **then**
    Record its planning span as `STOP`; store the final answer and end the loop
  else
    Append the runtime feedback to c^P under the role and tool protocol
  end if
end while
for each stored execution sequence and each of its generated tokens y_{ikt} **do**
  Reconstruct the stored local prefix c^E_{ikt} with its observations and images
  $\delta_{ikt} \leftarrow \text{clip}(\log \pi_T(y_{ikt} \mid c^E_{ikt}) - \log \pi_{\text{old}}(y_{ikt} \mid c^E_{ikt}), -b, b)$
end for
return collected records and δ; planning receives no distillation signal

Budgeted local execution. The LOCALEXECUTE routine measures transcript growth after the initial messages. It reserves an action allowance, a minimum observation allowance, and a final report allowance before starting another tool turn. Observations are truncated after accounting for the assignment reminder. If execution has not returned when its loop ends, the runtime requests a report using the reserved generation budget. A parsed `return` message, or the cleaned final response as a fallback, becomes the report delivered to planning. With the main budget, the routine keeps a safety margin of 512 tokens and reserves 4,096 for the report, 2,048 for an action, and at least 1,024 for an observation. These allowances determine whether another tool turn can start. The original question and assignment form the initial input for local execution. The report then supplies planning with evidence, source identifiers, and remaining uncertainty for choosing its next research subtask.

Local task context. The Planner passes an assignment through `prompt` and a short task label through `description`. Each local context combines the original question, the parsed assignment, and the evidence gathered by that Worker. Planning integrates returned reports and uses relevant findings to formulate subsequent assignments for dependent subtasks. Equation 3 accounts for the original input, assignment length, and local transcript growth. Stored token sequences retain observations and the generated prefix at each step, so student and teacher scores condition on the same information. This alignment makes the signed probability gap a token-level preference signal for sampled execution, with the report remaining available to guide later planning decisions.

D.2 Prefix Construction and Planning Credit

Each input has its own tree, built after rollout collection from the ordered assignments and reports. Node identity includes the complete canonical parent prefix. Action values group the same subtask choice under a parent even when its reports lead to different children. Reports identify the resulting canonical histories, while terminal returns determine the relative planning value of each choice.

Algorithm 2 Module 2: Prefix Construction and Planning Credit

Require: Rollout paths $\{(d_{ik}, r_{ik})_k\}_i$, terminal returns $\{R_i\}_i$, planning spans
Ensure: Planning-token tree credit B_{it} with zero credit outside recorded spans
Deduplicate rollout identities; initialize one root and empty visit sets for input x
Normalize and sort unique assignments; greedily cluster at cosine threshold 0.83
Canonicalize reports within each subtask cluster using typed matching below
for each distinct rollout i **do**
  $p \leftarrow \mathrm{root}(x)$
  for each ordered assignment–report pair (d_{ik}, r_{ik}) **do**
    $z \leftarrow \mathrm{cluster}(d_{ik})$; add i to $\mathcal{I}(p, z)$ once
    $p \leftarrow \mathrm{node}(p, z, \kappa_z(r_{ik}))$; record the parent and child identities
  end for
  Add the terminal marker `STOP` under p and record i in $\mathcal{I}(p, \texttt{STOP})$
end for
Compute input-level mean μ_x and sample standard deviation σ_x of R_i
for each parent p with observed choices $\mathcal{A}(p)$ **do**
  $n_{pa} \leftarrow |\mathcal{I}(p, a)|$ and $Q(p, a) \leftarrow \mathrm{mean}_{i \in \mathcal{I}(p,a)} R_i$ for each a
  $N_p \leftarrow \sum_a n_{pa}$; $\bar{Q}_p \leftarrow \sum_a n_{pa} Q(p, a)/N_p$
  $V(p) \leftarrow (N_p \bar{Q}_p + \nu \mu_x)/(N_p + \nu)$ with prior count $\nu = 2$
  Compute s_p from action values repeated by visit count, as in Eq. 8
  Set $\widetilde{\sigma}_p$ to the first value above ϵ in $(s_p, \sigma_x, 1)$
  $A^{\mathrm{tree}}(p, a) \leftarrow \mathrm{clip}((Q(p, a) - V(p))/\widetilde{\sigma}_p, -b, b)$ for each a
end for
Initialize $B_{it} \leftarrow 0$; add each decision's tree credit on its recorded planning span
Apply the generated-token mask and **return** B_{it} with the frozen tree metadata

Typed report matching. Reports first lose their return wrapper and undergo Unicode, case, punctuation, and whitespace normalization. Abstention cues in the first 200 characters take precedence. Otherwise, reports longer than six normalized words remain whole for entity matching; short reports are tested as booleans, dates, numbers, then entities. Types must agree. Boolean and numeric keys match exactly; a year-only date may match a date in that year. Entity matching uses normalized lexical rules, then embedding cosine similarity of at least 0.90. Lexical rules include token-order invariance, supported name variants, and string similarity. When numeric parsing is ambiguous, entity matching compares the full normalized report with the same lexical and semantic rules.

Determinism and stopping. Unique texts and candidate keys are sorted before greedy matching, and the first compatible leader defines each cluster. Different parent prefixes remain distinct. Every stored path has a terminal marker, including interrupted paths; only a recorded generated `finish` span can receive its `STOP` credit. Duplicate sequence rows do not add return observations, and tree metadata is fixed before padding and batch reordering, then retained throughout policy optimization.

Computational cost. With U unique assignments, W unique report keys, and embedding dimension d_e, greedy semantic comparisons take at most $O((U^2 + W^2)d_e)$ arithmetic operations, excluding text encoding and lexical matching. Return aggregation is linear in recorded path visits with hash-indexed sets; span projection touches the recorded decision intervals. No additional environment rollout or learned value model is required to estimate these planning advantages on the collected batch.

Credit at sparse prefixes. For a parent with one observed choice, prior shrinkage supplies a credit offset relative to the input-level mean. With multiple choices, the shared baseline also preserves their pre-clipping value differences, supporting comparisons among observed alternatives. The local scale uses sample standard deviation, with zero scale for a single visit. The input-level scale, then one, supplies the fallback for normalization across different visit counts and return distributions.

D.3 Signal Allocation and End-to-End Training

The training loop groups independent rollouts by task input, retains one terminal return per rollout, and allocates token advantages by planning or execution role. Planning receives signed outcome and tree credit. Execution receives the positive part of outcome credit and the signed teacher gap. The teacher, semantic encoder, stored contexts, and computed advantages remain fixed during each policy update, so gradients act only on the shared student policy parameters at the current training step.

Algorithm 3 Module 3: Structured Signal Allocation and Policy Training

Require: Dataset $\mathcal{D}$, initial policy π_θ, frozen teacher and encoder, G rollouts
Ensure: Updated policy parameters and checkpoints for validation selection

for each training step **do**
  Freeze a rollout snapshot $\pi_{\text{old}} \leftarrow \pi_\theta$; sample a batch from $\mathcal{D}$
  for each task $(x, y^\star)$ in the batch **do**
    Collect G independent rollouts and execution signals using Algorithm 1
    Evaluate each submitted answer against $y^\star$ to obtain terminal return R_i
    Deduplicate rollout identities; compute μ_x and sample standard deviation σ_x
    $A_i^{\text{grp}} \leftarrow (R_i - \mu_x)/(\sigma_x + \epsilon)$ for each rollout i
    Compute planning-span credit B_{it} with Algorithm 2
    for each generated token in the stored planning and execution sequences **do**
      if the token belongs to planning **then**
        $A_{it} \leftarrow A_i^{\text{grp}} + B_{it}$
      else
        $A_{it} \leftarrow \max(A_i^{\text{grp}}, 0) + \delta_{it}$
      end if
    end for
  end for
  Form mask m_{it} excluding prompts, observations, padding, and excluded sequences
  Optimize the token-averaged dual-clip objective in Eq. 10 with AdamW
  Evaluate and save checkpoints at the configured validation intervals
end for
return the policy checkpoint selected using validation macro accuracy

Optimization settings and masks. The main runs use 60 updates, eight task inputs per batch, $G = 8$, learning rate 10^{-6}, and local growth budget $B = 16{,}384$. Teacher gaps and tree advantages are clipped to $[-5, 5]$, with unit signal weights. PPO uses ratio clip width 0.2 and dual-clip factor 3; the implementation also clamps the log probability ratio to $[-20, 20]$ before exponentiation for numerical stability. The loss averages valid generated tokens, including generated reasoning and tool arguments. Prompt tokens and tool responses condition the policy but do not contribute direct loss terms.

Text and multimodal scoring. Text execution retains the question, parsed assignment, and local observations. Multimodal execution also retains the original pixels, expanded image-token identifiers, and matching visual positions for student and teacher scoring. The processor checks image-token counts before scoring. Image search supplies textual matches and source identifiers for subsequent page reading. Gemini-3.7-Flash compares the final answer with references, while distillation scores the sampled execution under the same task input and local evidence used by the student.

Training cost and inference. Collection costs depend on rollout decoding and tool access. The frozen teacher scores the collected batch, and execution-token scores supply the distillation signal; student optimization requires forward and backward passes. Local execution segments form the scoring inputs, whose lengths and count determine model computation. At inference, the shared policy uses greedy decoding with the original question and images, coordinating tool use through assignments and returned reports. Teacher scoring and tree-credit estimation operate during training.

Group statistics and loss masks. Each distinct rollout contributes one terminal return to the input-level statistics. Equal returns give zero outcome advantage and retain execution-side distillation. Generated-token masks select retained sequences, excluding padding and marked overlong or unfinished sequences. Shared rollout identities link planning and execution rows to the same outcome, assigning consistent task-level credit to the tokens used in each update of the shared policy.

E BENCHMARK DETAIL

The 19 datasets contain 1,324 test questions: 814 text (525 ID, 289 OOD) and 510 multimodal (395 ID, 115 OOD). In Tables 2–3, † marks seven OOD datasets absent from training. Bamboogle, PopQA, and FVQA contribute validation examples for checkpoint selection; GAIA, xbench, InfoSeek, and LiveVQA are used only for evaluation.

E.1 DATASET DETAILS

Text ID (seven datasets). BrowseComp-Plus (Chen et al., 2025b) and WebWalkerQA (Wu et al., 2025b) contribute 120 and 80 questions on fact-finding and web traversal. HotpotQA (Hotpot) (Yang et al., 2018), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), and MuSiQue (MuS) (Trivedi et al., 2022) contribute 65 questions each on multi-step evidence composition. Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Trivia) (Joshi et al., 2017) each add 65 questions covering natural information needs and trivia.

Text OOD (four datasets). Bamboogle (Bamb.) (Press et al., 2023) contributes 65 multi-hop questions; PopQA (Mallen et al., 2023) contributes 65 factual questions covering entities across popularity levels. GAIA (Mialon et al., 2024) adds 59 selected text tasks combining practical reasoning with information access. The 100 xbench-DeepSearch (xbench) questions (Chen et al., 2025a) test extended research and evidence gathering.

Multimodal ID (five datasets). MMSearch-Plus (Tao et al., 2026) contributes 105 questions linking fine-grained visual clues to facts through iterative search. Vision-DeepResearch Benchmark (VDR-Bench) (Zeng et al., 2026) contributes 135 questions on visual and textual fact-finding, and VisBrowse-Bench (Zhang et al., 2026b) contributes 65 on cross-modal verification. MMSearch (Jiang et al., 2024) adds 55 multimodal search questions, and SimpleVQA (Cheng et al., 2025) adds 35 questions connecting visual subjects to external knowledge, yielding 395 ID questions.

Multimodal OOD (three datasets). FactualVQA (FVQA) from MMSearch-R1 (Wu et al., 2025c) contributes 35 questions linking images to factual knowledge. InfoSeek (Chen et al., 2023) contributes 35 questions about visually identified entities. LiveVQA (Fu et al., 2025) adds 45 questions grounded in news imagery and time-sensitive knowledge. The resulting 115-question OOD pool covers factual, entity-centric, and news-based visual reasoning with original question–image inputs.

Task records. Each example retains its source dataset, instance identifier, question, and acceptable reference answers. Multimodal records also retain the original images, keeping the visual subject available during local execution. Reference answers are stored in evaluator metadata as lists that retain all supplied acceptable answer alternatives.

E.2 EVALUATION DETAILS

Inference and grading. The student uses greedy decoding independently of the distillation teacher, receiving the question and original images for multimodal tasks. Gemini-3.7-Flash scores final answers against the references. Text evaluation first accepts matches under the implemented answer-normalization rules, then invokes the judge for semantic equivalence. Multimodal evaluation supplies the answer alternatives directly to the same judge, accepting a match to any alternative. Equivalent wording and minor formatting differences are allowed; empty answers receive zero. With correctness $c_{dj} \in \{0, 1\}$ for question j in dataset d, final-answer accuracy (%) and the macro average over $\mathcal{S}$ are

$$\mathrm{Acc}(d) = \frac{100}{n_d} \sum_{j=1}^{n_d} c_{dj}, \qquad \mathrm{Avg}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{d \in \mathcal{S}} \mathrm{Acc}(d), \tag{23}$$

where n_d is the number of evaluated questions in dataset d. The same binary answer outcome supplies the terminal training reward, linking task accuracy to the training supervision in Appendix D.

Aggregation and model selection. Text Avg. averages all 11 datasets. Multimodal ID Avg. averages five ID datasets; All Avg. averages all eight. Figure accuracies use the same equal-dataset weighting. The ID/OOD macro averages combine as $(7A_{\mathrm{ID}} + 4A_{\mathrm{OOD}})/11$ for text and $(5A_{\mathrm{ID}} + 3A_{\mathrm{OOD}})/8$ for multimodal tasks. Training uses separate pools of 400 text and 347 multimodal examples. Checkpoints use validation macro accuracy, including the three validation-only OOD sources above. ID/OOD reflects training coverage independently of validation membership. RQ1–RQ2 cover both modalities; RQ3–RQ6 use text tasks (Section 4.1).

F BASELINE DETAIL

This appendix specifies the search implementations and hyperparameters of the baselines in Tables 2–3. Methods share the models, training pools, tools, and evaluation protocol below while retaining their own context management and learning objectives.

F.1 SHARED COMPARISON SETTINGS

Models and data. Text methods use Qwen3-4B with frozen Qwen3-4B-Instruct-2507 as distillation teacher (Yang et al., 2025); Appendix A substitutes a Qwen3-1.7B student. Multimodal methods use Qwen3-VL-2B and frozen Qwen3-VL-4B (Bai et al., 2025). Training uses 400 text and 347 multimodal questions, including demonstrations, supervised initialization, and skill construction. SDAR instead obtains targets from a privileged-context branch of the student. Text training comprises 200 BrowseComp-Plus, 100 WebWalkerQA, and 20 each from NQ, TriviaQA, HotpotQA, 2WikiMultiHopQA, and MuSiQue; multimodal training comprises 102 MMSearch-Plus, 114 VDR-Bench, 45 VisBrowse-Bench, 52 MMSearch, and 34 SimpleVQA questions.

Tools and masks. Search agents share text search and page opening; multimodal agents also access image search and retain the original images. Methods preserve their flat, folded, hierarchical, or parallel interaction structures. Losses cover generated reasoning, tool arguments, reports, and answers. Prompts, retrieved observations, images, and padding provide context and are excluded from direct policy and distillation losses.

Interaction budgets. SCAD and search baselines share the same task budget. Worker and Planner budgets B_W and B_P measure transcript growth after initial messages. With at most N_W Workers, the task budget is

$$B_{\text{total}} = N_W B_W + B_P, \qquad B_W = 16{,}384, \quad B_P = 32{,}768. \tag{24}$$

Training and validation use $N_W = 6$, giving 131,072 tokens. Text testing uses $N_W = 10$ and 196,608 tokens; multimodal testing retains $N_W = 6$ and 131,072 tokens. Flat agents use the same task budget across context continuations. Tree-GRPO reuses prefixes, charging new expansions against GB_{total} per question for up to $G = 8$ terminal paths.

Context continuation. Flat OPD-family agents summarize at 80% of a 32,768-token window. Summaries retain findings, source identifiers, unresolved checks, and next actions within 1,024 tokens. The question stays fixed; the latest interaction is retained when space permits. Continuation consumes the remaining rollout allowance. Folded methods return subtask reports; MemOPD updates compact memory between calls. Training scores actions under their original sampling contexts.

Optimization and evaluation. Unless specified otherwise, online training uses 60 updates, eight prompts with eight rollouts each, and one optimization pass per batch. AdamW uses learning rate 10^{-6}, weight decay 0.01, and gradient clipping 1.0. Training temperature/top-p are $1.0/1.0$; evaluation is greedy. GRPO-style ratio clipping is 0.2, with dual-clip factor 3. Checkpoints use validation macro accuracy; three seeds share splits, with mean and sample standard deviation reported. Gemini-3.7-Flash grades answers. Training may include the reward shaping below; evaluation uses binary correctness and dataset macro averages (Appendix E.2), with OOD defined by absence from training; Bamboogle, PopQA, and FVQA also contribute to validation.

F.2 REFERENCE MODELS AND SUPERVISED FINE-TUNING

Vanilla. The pretrained student directly answers the question and original images, when present, without planning or retrieval. Greedy decoding allows 4,096 generated tokens. No task-specific updates are applied; repeated evaluations retain the same checkpoint and grading protocol. This direct-generation reference differs from the scaffolded, untrained student at the start of RQ1.

Teacher. The frozen teacher uses SCAD's Planner–Worker interface, tools, and modality-specific test allowance with greedy decoding. Its table row measures generated final-answer accuracy under the shared judge and aggregation. During distillation, the same model instead scores token probabilities on student-generated responses.

SFT. Fixed teacher trajectories from the shared training pools include intermediate responses, tool calls, and answers; only response tokens receive cross-entropy loss. The student retains the learned tool-use protocol at evaluation. *Settings.* AdamW uses 5×10^{-6}, batch size eight, two epochs, and 10% warmup followed by linear decay. Long demonstrations use causally valid invocation segments of at most 32,768 tokens.

F.3 Reinforcement-Learning Baselines

GRPO (Shao et al., 2024). A flat search agent samples complete interactions and assigns each generated token its trajectory's normalized terminal reward, retaining signed advantages. *Settings.* The shared 8×8, 60-update recipe uses reference-KL coefficient 10^{-3} toward the frozen initial student and zero entropy bonus, without external distillation or a learned critic.

FoldGRPO (Sun et al., 2026). Branch-and-return operations execute subtasks in separate contexts and fold reports back into the parent; nested branching is disabled. Parent and branch responses are jointly optimized with signed terminal credit. *Settings.* Worker/Planner allowances follow Eq. 24; optimization follows GRPO. Our outcome-only configuration disables the original scope/process-reward terms while retaining learned context folding.

Tree-GRPO (Ji et al., 2026). The sampler branches at ReAct reasoning/tool-use steps, reuses prefixes, and propagates terminal returns through the resulting tree. *Settings.* Binary expansion over three branching levels yields up to eight terminal paths, continued to an answer within the shared budget. Learning rate, clipping, and reference-KL weight follow GRPO; each sampled tree edge contributes once to the loss.

HiPER (Peng et al., 2026). A shared policy proposes subgoals and executes search actions. Subgoal segments define hierarchical advantages; a value head estimates planning and execution baselines from terminal correctness. Planning credit propagates between subgoals, while execution credit propagates within each segment, aligning policy updates with the observed hierarchy. *Settings.* Discount $\gamma = 1$, GAE $\lambda = 0.95$, actor/critic learning rates 10^{-6} and 5×10^{-6}, and value-loss weight 0.5. Both levels retain signed advantages and the shared sampling budget.

F.4 On-Policy Distillation and Hybrid Methods

Teacher inputs. Teachers score student responses under recorded prefixes containing the question, earlier outputs, and observations, or reconstructed memory for MemOPD. Distillation averages valid response positions. The initial-policy reference used for RL regularization controls drift from initialization; the distillation teacher supplies the transfer signal. Their KL coefficients therefore govern separate objectives.

OPD (Agarwal et al., 2024; Gu et al., 2024). The student generates search trajectories, and the teacher scores sampled response tokens under matching prefixes. The log-probability gap supplies dense guidance, with summary-enabled continuation. *Settings.* Distillation weight 1.0, gap clipping $[-5, 5]$, terminal-reward weight zero, and 60 updates without an additional reference-KL penalty.

SDAR (Lu et al., 2026). A detached student branch receives a training-only skill summary; the acting student sees ordinary search context. Semantic retrieval selects one summary of at most 512 tokens from training demonstrations. *Settings.* Self-distillation weight 0.1 and sigmoid sharpness 1.0 give $g_t = \mathrm{sigmoid}(\mathrm{sg}[\Delta_t])$, emphasizing positive teacher–student log-probability gaps. The gate preserves signed GRPO credit; evaluation removes privileged skill context.

ATOD (Tan et al., 2026). Signed GRPO credit is combined with teacher guidance weighted by turn disagreement and uncertainty, normalized within each trajectory and combined by soft-OR. Reweighting affects only distillation. *Settings.* With $u(s) = \min(s/32, 1)$, OPD/RL weights are $1 - 0.9u(s)$ and $1 + u(s)$, moving from $1.0/1.0$ to $0.1/2.0$ in 32 updates. Continuous turn weights activate for at least three turns and maximum disagreement ≥ 0.01. Actor learning rate is 10^{-6} with six warmup updates. A teacher anchor remains after annealing, whereas sequential OPD–RL schedules (Li et al., 2026a) separate imitation and reward optimization into successive phases.

TCOD (WANG et al., 2026). The forward-to-backward curriculum expands the student-controlled prefix, with teacher scoring within the active horizon. *Settings.* $k(s) = \min(32, 2 + \lfloor 30s/48 \rfloor)$ grows from two to 32 turns over 48 updates, followed by 12 updates at full training depth. Distillation weight is one and terminal-reward weight is zero. Evaluation removes the curriculum and uses the complete interaction allowance with summary support.

MemOPD (Liu et al., 2026c). Training reconstructs each invocation's input, response, positions, and causal visibility after memory rewriting. Retained-text copies provide context; the original sampled actions receive teacher guidance and PPO credit. *Settings.* Memory cap 1,024 tokens, full-vocabulary reverse-KL weight 0.1, actor/critic learning rates 10^{-6} and 5×10^{-6}, $\gamma = 1$, GAE $\lambda = 0.95$, and value-loss weight 0.5. Initialization uses one SFT epoch on teacher trajectories at 5×10^{-6} with batch size eight.

F.5 Multimodal Search and Distillation

MMSearch-R1 (Wu et al., 2025c). Multi-turn GRPO learns when to answer directly and when to retrieve visual or textual evidence. The policy chooses one action per round and integrates returned evidence into subsequent reasoning. We retain its accuracy, search-cost, and formatting reward components, using the shared correctness judge. *Settings.* With correctness c, format indicator f, and $\rho = 0.9$ if search is used (1 otherwise), the reward is $0.9c\rho + 0.1f$. The optimizer follows GRPO; retrieved content is masked, and image search uses the common service.

VSearcher (Zhang et al., 2026a). We adapt its reinforcement-learning stage to long-horizon multimodal search using the shared Qwen3-VL-2B student and multimodal training pool. The policy retains original image inputs and gathers visual and textual evidence through multi-round tool use. The student samples its own trajectories for terminal-reward GRPO, with signed advantages applied to generated responses. *Settings.* Online training uses 60 updates at 10^{-6}, with eight prompts and eight rollouts per prompt. RL uses a 10^{-3} reference-KL coefficient and no external teacher-scoring loss, retaining the common inference and evidence budgets.

VOLD (Bousselham et al., 2026). Joint GRPO and OPD combine outcome rewards with teacher guidance for image-grounded search. We use the shared Qwen3-VL teacher–student pair and multimodal training pool. Both models condition on the original images and student search history. Reward-guided masking applies distillation to unsuccessful trajectories, while GRPO trains on all trajectories. *Settings.* Online training uses learning rate 10^{-6}, eight prompts with eight rollouts each, and sampled-token k_2 distillation with coefficient 0.1. The vision tower remains frozen, and language-policy parameters are optimized over the shared 60-update budget.

HyperEyes (Li et al., 2026b). Parallel grounded search combines trajectory-level efficiency rewards with token-level correction. Multiple queries may share a round; both rounds and individual calls enter the efficiency reward. A frozen Qwen3-VL-4B teacher supplies OPD only on failed rollouts. *Settings.* Online training uses 60 updates with the shared optimizer and an OPD coefficient of 0.05. TRACE uses redundancy tolerance 1.5, redundancy penalty -0.1, format penalty -0.5, and reward intervals $[0.05, 0.20]$ for efficient and $[-0.10, -0.02]$ for redundant successful searches. Up to four queries execute concurrently; all queries count against the shared budget.

Online comparison protocol. VSearcher, VOLD, and HyperEyes start from the shared Qwen3-VL-2B checkpoint and use 60 online updates on the same multimodal training pool. Their search structures and learning objectives retain method-specific behavior under shared rollout budgets. Checkpoint selection and evaluation follow the common validation and test protocol.

F.6 SCAD and Controlled Comparisons

SCAD. Local Workers preserve the question, assignment, and subtask observations for teacher scoring. The Planner receives signed terminal and prefix-tree credit; Workers receive positive terminal credit and signed local OPD. Training uses 60 updates, 8×8 rollouts, and AdamW at 10^{-6}; all three signal weights are one, with teacher/tree clipping at five. Semantic matching uses Qwen3-Embedding-4B (Zhang et al., 2025), subtask/report thresholds $0.83/0.90$, and prior count $\nu = 2$. The verl/HybridFlow implementation (Sheng et al., 2025) and full update procedure are described in Appendix D.

Component and context controls. RQ2 holds the Planner–Worker architecture fixed and adds local OPD, tree credit, and structured allocation to local GRPO in that order. RQ3 uses RQ2's second stage before tree credit and positive-only allocation. Its adapted ATOD keeps full-history teacher supervision and its own GRPO signals; the main table uses standalone ATOD. RQ4 changes only planning credit, including a control multiplying normalized no-tree Planner advantages by 1.6. RQ5 changes GRPO/OPD assignment within the shared framework; Full-E keeps SCAD planning and uses signed GRPO plus OPD for execution.

Efficiency controls. RQ6(a) holds the Planner–Worker framework fixed across all three methods and counts only teacher-supervised response tokens. In RQ6(b), GRPO + OPD uses the flat search interface with unit weights for signed terminal credit and teacher guidance, sharing the online training recipe and frozen teacher with ATOD and SCAD. RQ5's Full GRPO + OPD instead uses the Planner–Worker framework with planning tree credit. RQ6 averages three runs; panel (b) excludes Gemini-3.7-Flash judging from 60-step timing.

G Case Study

Table 7: Three dependent subtasks in a VDR-Bench training rollout of SCAD (Qwen3-VL-2B). Completed branches and reports are condensed; the terminal reward is $R = 1$.



Boston Public Library, Leslie Jones Collection.

Question. The man in the foreground had a son who starred in a radio series that aired from 1953 to 1954. What was the genre of this series, and what was the name of the 1952 film that preceded it?

Ground truth: Light drama;
The Story of Will Rogers.

Stage	Subtask and returned evidence
Branch 1 Visual identity	`identify person` Identify the foreground man and his associations with radio or film. `report` Identifies **Will Rogers**, but leaves his son's 1953–1954 series unresolved and includes an unverified 1952 title.
Branch 2 Family link and radio series	`trace son's series` Find **Will Rogers' son's** radio series from 1953–1954 and confirm its genre. `report` **Will Rogers Jr.** starred in **Rogers of the Gazette**, a **drama**; the report also links it to *The Story of Will Rogers* (1952).
Branch 3 Film verification	`verify precursor` Confirm the exact title of the 1952 film preceding **Rogers of the Gazette**. `report` Confirms **The Story of Will Rogers** as the film that inspired the radio series, with Will Rogers Jr. carrying his father's persona into the radio role.
SCAD answer $R = 1$ **Correct**	`finish` **Drama; The Story of Will Rogers (1952).** Three returned reports connect the photograph to both requested facts.

Three distinct intermediate results. SCAD identifies the person, traces his son's radio series and its genre, and verifies the preceding film. The Planner reads each report and writes the relevant names into subsequent subtask instructions. The final branch verifies the film connection before the Planner submits the complete answer. The three branches thus link visual identification, relational retrieval, and film verification through separate local executions.

H Limitations and Future Work

Limitations. SCAD is evaluated with compact Qwen students on text and image-based search tasks. Local distillation depends on the selected teacher's competence on sampled subtasks. Prefix-tree credit draws on semantically matched subtask–report histories, so its comparison quality depends on matching accuracy and the number of observed alternatives. Diverse prefixes may yield sparse sibling comparisons, for which the current method uses prior shrinkage. Worker budgets and signal weights remain fixed across tasks, leaving room to adapt to subtask difficulty and feedback quality.

Future Work. We aim to extend SCAD to larger backbones and a broader range of interactive tasks, while studying teacher selection by local competence. Adaptive rollout sampling and learned history matching could increase useful comparisons within the prefix tree. A complementary direction is to allocate Worker budgets and teacher supervision according to subtask difficulty and expected learning benefit. Adapting these allocations together with training signal weights could improve the balance between task accuracy and computation across tasks with different evidence requirements.