LEARNING RETRIEVAL PLANS FROM SIBLING EXECUTION OUTCOMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-hop question answering requires agents to retrieve evidence and use earlier findings to guide later searches and reasoning. Yet different workflows can yield the same correct answer, so answer labels alone do not define a unique target for supervising intermediate actions. To address this problem, we propose SIPPO (Sibling-Informed Planner Preference Optimization), which learns executable workflows through execution-grounded supervision and final-answer rewards. A shared role-conditioned policy generates a directed acyclic graph whose dependencies resolve against node memory, connecting actions to states from earlier execution. We construct execution-grounded supervision by executing teacher-proposed workflows on multi-hop question-answering data, verifying evidence support and answer correctness, and pairing each recorded action with the state available at that step. We instantiate RLOO at workflow granularity (Workflow-RLOO), assigning advantages from final-answer Exact Match (EM) rewards to every generated role action in a workflow. ***Sibling Execution Outcomes*** are the terminal EM rewards for on-policy workflows sampled for the same question. Planner Sibling Contrast (PSC) uses reward differences to compare planner responses from EM-correct and EM-incorrect workflows, without additional intermediate labels. Across four multi-hop benchmarks comprising 22,517 answerable questions, SIPPO achieves the highest macro EM and macro F1 in the main comparison in both backbone settings. Its macro EM/F1 are 0.3362/0.4212 on Qwen2.5-3B and 0.3813/0.4701 on Qwen3-4B. Adding PSC to Workflow-RLOO improves macro EM by 2.72 and 7.01 percentage points, respectively; the paired 95% confidence interval for the 3B gain excludes zero. The 3B gain accompanies fewer generated tokens and more retrieval calls. Frozen-policy diagnostics show that PSC gradients respond to the terminal EM reward labels of same-question workflows.

1 INTRODUCTION

Multi-hop open-domain question answering requires agents to retrieve evidence and use earlier findings to guide later actions (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022). Agentic Retrieval-Augmented Generation (RAG) exposes these decisions through interleaved actions and retrieval (Yao et al., 2022; Trivedi et al., 2023; Jin et al., 2025). Yet multiple workflows can produce the same correct answer, leaving no unique intermediate target. This motivates our central question: *How can we train a structured multi-hop retrieval agent when final answers are verifiable, but there is no unique correct intermediate workflow?*

Dependent queries and node conclusions must match the state produced by earlier retrievals. In Reinforcement Learning (RL), role actions must also remain tied to the execution that produced the answer, although its terminal reward gives the planner coarse feedback. We therefore *learn to execute, learn from outcomes, and refine planner supervision*.

SIPPO (Sibling-Informed Planner Preference Optimization) learns from executable workflows. A shared role-conditioned policy emits a Directed Acyclic Graph (DAG) whose symbolic dependencies feed earlier node conclusions to later calls. For example, `#2#` binds a downstream task to node 2's conclusion. The runtime validates the graph, resolves references against node memory, executes nodes in dependency order, and records each state and response. These *state semantics* connect planning, rewriting, evidence interpretation, reasoning, and synthesis in one trajectory.

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

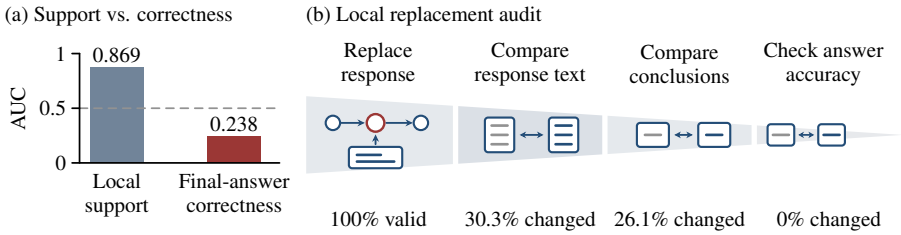


Figure 1: Local signals and final correctness. (a) Reason-premise attribution discriminates local support on controlled cases (AUC 0.869) but poorly ranks final-answer correctness on natural traces (AUC 0.238); dashed line: chance. (b) Local replacement audit: replacing an intermediate response leaves all audited executions valid; 30.3% of replacements change the response text and 26.1% change the conclusions, yet 0% change final-answer correctness.

We use executable workflows for *execution-grounded workflow supervision*: teacher-proposed workflows are executed in the Wiki18 environment, later inputs are populated from resolved dependencies and retrieved evidence, and accepted records yield state–response targets for Supervised Fine-Tuning (SFT). From 4,956 source questions we retain 800 workflows (760 train, 40 validation), yielding 4,081 training actions. SFT teaches the shared policy to use the role interfaces; outcome learning decides which executions to prefer.

Our offline diagnostics show why local process signals require validation before assigning outcome credit (Figure 1). Reason-premise attribution reaches AUC 0.869 on controlled local-support examples but only 0.238 for outcome ranking on natural trajectories. In a validated replacement audit, changing intermediate responses produces no genuine change in final-answer correctness. Thus strong local-support discrimination and intermediate variation do not by themselves provide reliable terminal-outcome credit.

To train dependent role actions without intermediate reward labels, we use RLOO at workflow granularity, which we call Workflow-RLOO. ***Sibling Execution Outcomes*** are the terminal Exact Match (EM) rewards for on-policy workflows sampled for the same question. For each question, we sample four complete workflows, termed sibling executions, and assign each workflow's leave-one-out advantage to every model-generated role token in that workflow. This assignment ties role-action updates to the complete execution. Workflow-RLOO provides the outcome-only control for additional planner supervision. Building on Workflow-RLOO, Planner Sibling Contrast (PSC) compares planner responses from on-policy workflows sampled for the same question but receiving different terminal EM rewards. PSC converts the reward differences into weak pairwise preferences over planner responses, complementing workflow-level outcome learning. Both RL conditions start from the same SFT initialization.

On 22,517 questions under deterministic decoding, Qwen2.5-3B SFT obtains 24.80/33.53 macro EM/F1, Workflow-RLOO 30.90/38.87, and SIPPO 33.62/42.12. The paired macro EM and F1 intervals for adding PSC exclude zero; PSC's gains over Workflow-RLOO are positive on HotpotQA, MuSiQue, and Bamboogle but indistinguishable on 2Wiki. We also report compute and test whether PSC gradients respond to the terminal EM rewards of workflows sampled for the same question.

In summary, our contributions are threefold: (1) an executable workflow representation with dependency-resolved runtime states for execution-grounded supervision; (2) Workflow-RLOO for workflow-level terminal-outcome learning and PSC for same-question planner preferences from sibling execution outcomes, without canonical intermediate labels; and (3) evaluation on 22,517 questions across four multi-hop benchmarks and two backbone settings, with paired statistics, compute accounting, and frozen-policy planner-gradient diagnostics.

2 Related Work

Retrieval, planning, and outcome learning. RAG provides external evidence, and adaptive agents interleave retrieval and reasoning (Lewis et al., 2020; Jiang et al., 2023; Asai et al., 2024; Trivedi et al., 2023; Yao et al., 2022). Search-R1 and later systems learn multi-turn search from answer

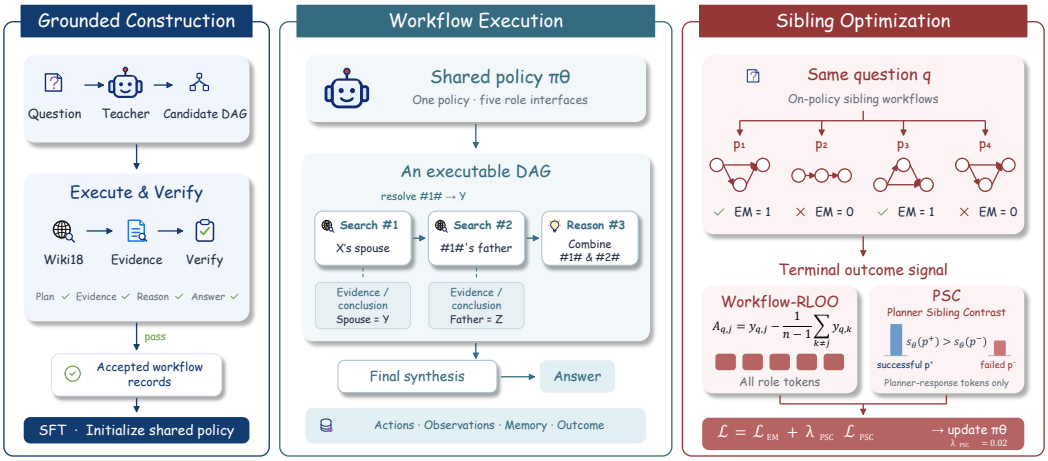


Figure 2: SIPPO overview. Left: execution-grounded Supervised Fine-Tuning (SFT). Middle: dependency-resolved workflow execution. Right: terminal-outcome learning and Planner Sibling Contrast (PSC).

rewards (Jin et al., 2025; Song et al., 2025; Chen et al., 2025). We study outcome learning with a shared multi-role policy and executable state transitions.

Structured workflows and credit assignment. Question decomposition and graph reasoning expose intermediate subproblems (Min et al., 2019; Wolfson et al., 2020; Zhou et al., 2022; Besta et al., 2024); GAP and planner-centric objectives make structure explicit (Wu et al., 2025; Hussain et al., 2026). SIPPO resolves symbolic dependencies through node memory, so predecessor execution materializes each later role's state. O^2-Searcher learns through search interaction (Mei et al., 2025), while shared-prefix, suffix-resampling, and sibling methods derive finer credit from terminal variation (Ji et al., 2026; Zhao et al., 2026; Meng & Chen, 2026). SIPPO instead reuses Workflow-RLOO's complete same-question on-policy workflows and contrasts only their planner responses.

3 METHOD

SIPPO learns multi-hop retrieval through executable workflows that connect planning, retrieval, reasoning, and answer synthesis (Figure 2). We first define these workflows and their execution, then describe execution-grounded supervision of the shared policy. From the same SFT initialization, we compare Workflow-RLOO with Workflow-RLOO plus PSC: both learn from complete execution outcomes, and PSC adds a preference over same-question planner responses throughout outcome learning.

3.1 EXECUTABLE WORKFLOW REPRESENTATION AND EXECUTION

Task and workflow. Given a question q, SIPPO generates a directed acyclic graph $G = (V, E)$. Each node is

$$v_i = \langle i, t_i, x_i \rangle, \qquad t_i \in \{\texttt{search}, \texttt{reason}\}, \tag{1}$$

where i is a unique identifier and x_i is a natural-language task. A symbolic reference `#j#` in x_i induces an edge $(v_j, v_i) \in E$, and the executor preserves every parser-valid generated node.

A complete workflow w contains the generated plan, all role-conditioned model actions, retrieval observations, node conclusions, execution status, and final answer, serving as both an executable program and the outcome-assignment unit. At runtime, $q \to G \to M$: the planner emits G, the executor resolves nodes against memory M, retrieves or reasons, writes conclusions back to M, and lets descendants condition on the updated state. Thus DAG edges have state-transition semantics rather than merely descriptive structure. The workflow record stores the role calls, observations,

memory updates, status, and final answer. SFT extracts state–response targets from this record, while RL assigns outcome credit only to generated response tokens; retrieval outputs and executor bookkeeping are loss-masked (Appendix B). PSC pairs planner responses from workflows for the same question, and the same record supports execution and compute accounting. SIPPO can therefore optimize terminal outcomes while allowing multiple valid topologies and operator sequences. In a logged two-search example, resolving the spouse reference before the second lookup yields an EM-correct final answer (Appendix C).

Shared policy and role interfaces. One shared policy π_θ implements five prompt- and schema-defined interfaces: PlanningAgent emits a `<plan>`; QueryRewriteAgent produces dependency-resolved queries; AnswerGenerationAgent turns passages into node conclusions or requests a rewrite; ReasonerAgent combines node conclusions; and AnswerSummarizationAgent emits the final `<answer>`. Retrieved passages and executor state are observations, and loss applies only to action tokens. PSC computes its auxiliary loss only on the planner-response span. Because all five roles share parameters, this update can affect every role; the role inventory defines available capabilities while leaving operator selection to the policy.

Plan validation and topological execution. The parser checks schema, consecutive identifiers, backward references, and at least one search node. The executor replaces each `#j#` in v_i with the corresponding conclusion $M[j]$. Dependent searches are rewritten into standalone Wiki18 queries; independent searches use their task directly. The top three passages condition AnswerGenerationAgent, which may request one evidence-triggered recovery rewrite. ReasonerAgent receives the resolved task and parent conclusions, and every completed output is written back to M for descendants. The final synthesizer reads q and the completed node table.

The resulting workflow record pairs each role response with the state that preceded it and links all responses to one final answer. These records support both state–response supervision and outcome learning, without identifying a uniquely responsible intermediate action.

3.2 Execution-Grounded Workflow Supervision

Generate actions in executed states. Teacher generation is coupled to the runtime: each action is paired with the state produced by preceding actions and environment observations (resolved conclusions, retrieved passages, or completed node memory). We attempt 4,956 2WikiMultiHopQA training questions with DeepSeek-V4-Pro and execute each candidate in the same top-3 environment. The gold answer is used only to accept or reject completed executions; action targets are the recorded model responses generated from the executed state and may therefore coincide with the gold answer.

Accept grounded, successful workflows. We retain only well-formed plans whose search conclusions are supported by retrieved passages, reason conclusions pass a separate logic verifier, and final answers are node-supported and exact-match correct. This yields 800 workflows (16.14%); splitting by question before expansion gives 760 train and 40 validation workflows without action leakage.

Extract supervision from workflow records. Each action becomes a role-level row with its prompt, execution state, and target response. The 4,081 training rows contain 760 planning, 690 rewrite, 1,687 retrieval-grounded answer, 184 reason, and 760 synthesis actions; the held-out split contains 224 rows (Appendix E). Loss applies only to assistant response tokens. We train the full Qwen2.5-3B policy for two epochs and use the final endpoint; optimization details are reported in Appendix A. We keep this execution-grounded SFT endpoint fixed across the two RL conditions, so the main comparison measures workflow-level outcome learning and the added PSC objective from a shared initialization.

3.3 Workflow-Level Outcome Learning

Once the policy can operate the interfaces, Workflow-RLOO optimizes complete executions while using workflow identifiers to keep every role action connected to the workflow's terminal EM reward. Workflow-RLOO uses the REINFORCE Leave-One-Out (RLOO) estimator (Ahmadian et al., 2024) at workflow granularity. For each training question q, the current policy samples and executes $n = 4$

complete workflows $\{w_{q,1}, \ldots, w_{q,n}\}$ under a shared execution protocol. We call these on-policy, same-question workflows *sibling executions*. *Sibling Execution Outcomes* are the terminal EM rewards for these workflows:

$$y_{q,j} = \mathrm{EM}(\hat{a}_{q,j}, a_q) \in \{0, 1\} \tag{2}$$

where $\hat{a}_{q,j}$ is the final answer from $w_{q,j}$ and a_q is the gold answer. We compute a leave-one-workflow-out advantage

$$A_{q,j} = y_{q,j} - \frac{1}{n-1} \sum_{k \neq j} y_{q,k}. \tag{3}$$

The same $A_{q,j}$ is broadcast to every model-generated token from every role in workflow $w_{q,j}$. This preserves the fact that a plan, its dependent queries, and the final synthesis jointly produced the outcome; they are not independent samples. The actor is updated with a clipped policy objective using these token-level advantages, token-mean loss aggregation, and no learned critic. Workflow-RLOO provides the global outcome-learning component of SIPPO and the control for PSC. The Workflow-RLOO advantage already updates the planner; sibling disagreement also makes a direct planner-response comparison available, which motivates the following refinement.

3.4 Planner Sibling Contrast

PSC complements Workflow-RLOO by ranking planner responses from the same on-policy sibling group. Let P_q^+ and P_q^- contain responses preceding EM-correct and EM-incorrect executions, respectively, under a shared planner context and execution protocol. Let c_q denote the fixed planner context for question q, including the shared system and schema instructions. For a response p, define

$$s_\theta(p) = \frac{1}{|p|} \sum_{t=1}^{|p|} \log \pi_\theta(p_t \mid p_{<t}, c_q). \tag{4}$$

For questions with both outcomes, the pair-averaged loss is

$$\mathcal{L}_{\mathrm{PSC}}(q) = \frac{1}{|P_q^+||P_q^-|} \sum_{p^+ \in P_q^+} \sum_{p^- \in P_q^-} \log\left(1 + e^{-\beta[s_\theta(p^+) - s_\theta(p^-)]}\right). \tag{5}$$

We use $\beta = 1.0$ and average this loss equally over informative questions, setting PSC to zero if none are informative. Uniform-outcome groups also have zero Workflow-RLOO advantages. Length normalization removes the cumulative log-probability scaling with response length. Each pair has the SimPO form with zero target margin (Meng et al., 2024). Because downstream execution also contributes to EM, PSC provides a weak outcome-associated preference over planner responses.

PSC applies its auxiliary loss only to planner-response tokens; other roles receive only the Workflow-RLOO loss (Appendix B). We minimize

$$\mathcal{L} = \mathcal{L}_{\mathrm{EM}} + \lambda_{\mathrm{PSC}} \mathcal{L}_{\mathrm{PSC}}, \qquad \lambda_{\mathrm{PSC}} = 0.02. \tag{6}$$

Here $\mathcal{L}_{\mathrm{EM}}$ denotes the Workflow-RLOO actor objective induced by terminal EM rewards: the EM signal enters through sampled rewards, while optimization uses token-mean aggregation and a policy-ratio clipping range of $[0.8, 1.2]$ (Appendix A).

4 Experiments

4.1 Research Questions

We test answer quality against agentic-RAG systems (RQ1), the gains from workflow-level learning and PSC over execution-grounded SFT (RQ2), changes in execution and compute (RQ3), and whether training preserves valid workflows while PSC responds to sibling outcomes (RQ4).

4.2 Setup

Training data. RL uses 15,000 training questions (40% 2WikiMultiHopQA, 30% HotpotQA, 30% MuSiQue). Both conditions start from the same Qwen2.5-3B SFT policy. Each of 468 iterations samples 32 questions and four workflows per question (128 workflows), with learning rate 5×10^{-7}, temperature 0.8, top-$p = 0.95$, and PSC coefficient 0.02. Initialization, rollout exposure, Workflow-RLOO, retriever, decoding, and evaluation are shared.

Table 1: Main EM comparison on 22,517 answerable questions. All methods use the same Wiki18/E5 top-3 retrieval environment and their reported inference settings. Values are EM (%); Macro is the unweighted mean across datasets. Pale red/blue cells mark the best/second-best result within each backbone. Full EM/F1 results are in Appendix F.

Method	HotpotQA	2Wiki	MuSiQue	Bamboogle	Macro
Qwen2.5-3B					
Inference-time retrieval					
RAG	24.97	21.87	4.52	12.00	15.84
IRCoT	28.59	26.21	5.85	15.20	18.96
RL-trained search agents					
Search-R1$_{\text{GRPO}}$	28.55	26.88	6.18	12.80	18.60
Search-R1$_{\text{PPO}}$	29.02	29.77	8.83	17.60	21.31
ZeroSearch$_{\text{Instr.}}$	28.72	26.22	5.68	12.80	18.36
StepSearch$_{\text{Instr.}}$	18.31	20.40	6.84	13.60	14.79
AutoRefine$_{\text{Base}}$	**42.21**	40.76	16.34	33.60	33.23
SIPPO	34.92	**46.47**	**17.09**	**36.00**	**33.62**
Qwen3-4B					
On-policy distillation					
ReOPD	36.11	31.49	12.73	34.40	28.68
OPD	35.91	31.82	12.98	36.00	29.18
RL-trained search agents					
InfoSeeker$_{\text{ORBIT}}$	**40.96**	40.09	18.04	44.80	35.97
Search-R1$_{\text{ORBIT}}$	36.15	40.26	13.89	**46.40**	34.18
SIPPO	37.41	**50.33**	**19.16**	45.60	**38.13**

Evaluation. The formal suite has 22,517 answerable questions: 12,576 2WikiMultiHopQA, 7,405 HotpotQA, 2,411 MuSiQue after removing six normalized duplicates, and 125 Bamboogle (evaluation only). None overlaps the RL training mixture or 480-question validation set. All full-run comparisons use this manifest and Wiki18/E5 top-3 retrieval, while preserving each baseline's native prompt, controller, and decoding. We report normalized EM and token F1 from each run's evaluator; for the ORBIT InfoSeeker reproduction, JADE's batch scorer uses complete saved prediction/gold pairs. Both SIPPO backbones and both 3B RL conditions use the full suite. Appendix G reports a separate three-seed stochastic evaluation (temperature 0.7, top-$p = 0.95$).

4.3 Main Results

SIPPO leads the main comparison on both backbones. Table 1 reports the main EM comparison; full EM/F1 results are in Table 8. It compares SIPPO with RAG (Lewis et al., 2020), IRCoT (Trivedi et al., 2023), Search-R1 (Jin et al., 2025), ZeroSearch (Sun et al., 2025), StepSearch (Zheng et al., 2025), AutoRefine (Shi et al., 2026), InfoSeeker (Lee et al., 2026), ReOPD/OPD (Liao et al., 2026), and the ORBIT Search-R1 reproduction (Thakur et al., 2026) (RQ1). Following each method's reported prompt, controller, and decoding configuration, all systems are evaluated on the same 22,517-question manifest with the same Wiki18/E5 top-3 retrieval environment. Under this shared evaluation setting, SIPPO achieves the highest macro EM/F1 in both backbone settings: 0.3362/0.4212 on Qwen2.5-3B and 0.3813/0.4701 on Qwen3-4B. SIPPO also achieves the highest 2Wiki scores in both settings. The InfoSeeker row uses the ORBIT authors' Qwen3-4B reproduction of InfoSeeker (Lee et al., 2026; Thakur et al., 2026), evaluated with the ORBIT controller. The ablations below examine the training objectives behind SIPPO's gains over its matched training controls.

4.4 Ablation Studies

Table 2 isolates workflow-level learning and PSC beyond execution-grounded SFT (RQ2). Within each backbone, the RL variants share the SFT policy, collection recipe, DAG representation, and executable interfaces.

Table 2: Answer quality across learning conditions. Red/blue cells with bold/underlined values mark best/second-best within each backbone and metric (higher is better; ties share marks).

Condition	Metric	Benchmarks				Aggregates	
		HotpotQA	2Wiki	MuSiQue	Bamboogle	Macro	Micro
Qwen2.5-3B							
SFT	EM	0.250	0.327	0.127	0.288	0.2480	0.2800
	F1	0.340	0.393	0.201	0.408	0.3353	0.3550
Workflow-RLOO	EM	0.339	0.464	0.146	0.288	0.3090	0.3876
	F1	0.435	**0.519**	0.220	0.381	0.3887	0.4587
SIPPO (Ours)	EM	**0.349**	**0.465**	**0.171**	**0.360**	**0.3362**	**0.3947**
	F1	**0.453**	0.515	**0.251**	**0.466**	**0.4212**	**0.4658**
Qwen3-4B							
SFT	EM	0.280	0.313	0.150	0.376	0.2798	0.2852
	F1	0.368	0.376	0.218	0.519	0.3700	0.3570
Workflow-RLOO	EM	0.317	0.391	0.161	0.376	0.3112	0.3417
	F1	0.419	0.446	0.243	0.504	0.4030	0.4156
SIPPO (Ours)	EM	**0.374**	**0.503**	**0.192**	**0.456**	**0.3813**	**0.4272**
	F1	**0.484**	**0.561**	**0.280**	**0.556**	**0.4701**	**0.5056**

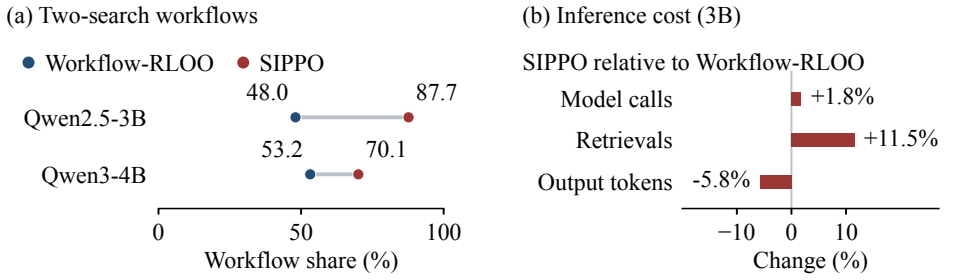


Figure 3: Execution changes with PSC. (a) Two-Search, no-Reason workflow shares. (b) Qwen2.5-3B mean per-question cost changes relative to Workflow-RLOO on 22,517 questions; output tokens exclude input and retrieved text.

Statistics. For Qwen2.5-3B, we pair all 22,517 predictions by unique question–gold keys and use 100,000 paired-bootstrap resamples, stratified for macro averages. Exact McNemar testing of PSC gives $p = 0.0146$ (2,172 wins, 2,013 losses, 18,332 ties); Appendix F gives per-benchmark intervals.

Workflow-level rewards improve beyond interface supervision. On Qwen2.5-3B, Workflow-RLOO raises SFT macro EM/F1 by 6.10/5.34 points. Its paired micro gains are 10.76 EM points (95% CI $[10.12, 11.40]$) and 10.38 F1 points (95% CI $[9.75, 11.01]$), showing gains from terminal outcomes beyond imitation without intermediate reward labels.

Sibling outcomes provide useful additional planner supervision. On Qwen2.5-3B, PSC adds 2.72 macro EM points (95% CI $[1.00, 4.49]$) and 3.25 macro F1 points ($[1.56, 4.97]$) over Workflow-RLOO; both intervals exclude zero (Table 9). Micro EM rises 0.71 points (95% CI $[0.15, 1.27]$), and micro F1 also rises 0.71 points (Table 10). On Qwen3-4B, Workflow-RLOO raises macro EM/F1 from 27.98/37.00 to 31.12/40.30, and SIPPO reaches 38.13/47.01. Thus PSC adds 7.01 macro EM points and 6.72 macro F1 points over Workflow-RLOO under the matched scientific settings in Appendix H.

PSC gains are positive in aggregate but uneven across benchmarks. Figure 4 localizes the Qwen2.5-3B gains. Bamboogle has the largest point gains (7.2 EM / 8.6 F1), followed by MuSiQue

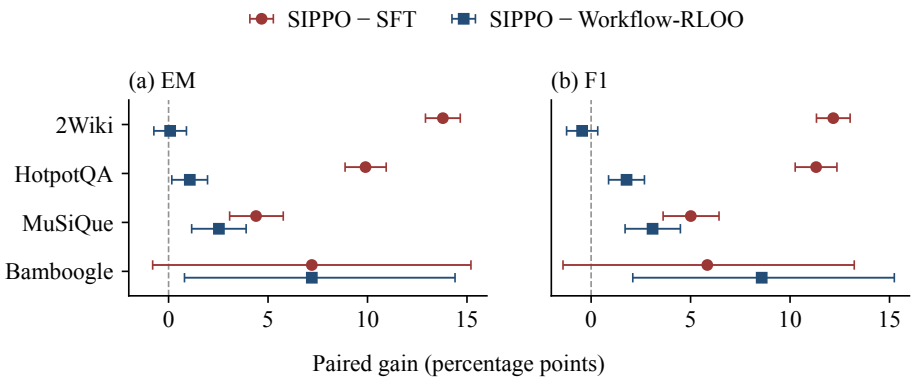


Figure 4: Paired Qwen2.5-3B gains over SFT and Workflow-RLOO. Points show EM/F1 differences on the same questions; bars are paired-bootstrap 95% confidence intervals; dashed line marks zero gain.

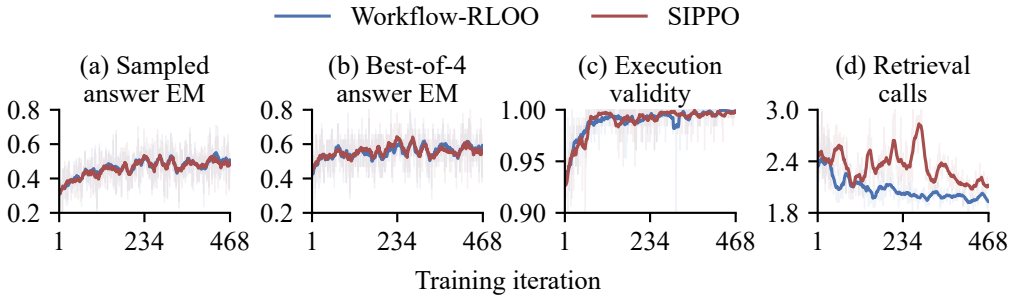


Figure 5: Training diagnostics: (a) sampled-answer EM, (b) best-of-four EM, (c) execution validity, and (d) retrieval calls. Dark/faint curves show seven-iteration means/raw measurements over 468 iterations; the validity axis spans $[0.90, 1.00]$.

(2.5/3.1) and HotpotQA (1.1/1.8), with positive paired intervals; 2Wiki changes by only 0.08 EM and -0.45 F1, and both intervals cross zero. Because macro averaging weights benchmarks equally, Bamboogle contributes 1.80 of the 2.72 macro-EM points from nine net additional correct answers among 125 questions. Across all 22,517 questions, SIPPO has 159 net additional correct answers over Workflow-RLOO.

PSC changes execution structure and inference cost. Figure 3(a) shows that adding PSC raises the share of two-Search, no-Reason workflows from 48.0% to 87.7% on Qwen2.5-3B and from 53.2% to 70.1% on Qwen3-4B (RQ3). This describes planner behavior; the comparison does not isolate topology's effect on accuracy.

Across 22,517 Qwen2.5-3B traces, Workflow-RLOO uses 10.7% fewer model calls, 5.8% fewer output tokens, and 16.8% fewer retrievals than SFT. Adding PSC raises calls by 1.8% and retrievals by 11.5%, while reducing output tokens by 5.8% (Figure 3(b)). Counts include every role response and retrieval; output tokens exclude prompts and retrieved text (Appendix I).

Training preserves executable behavior while improving sampled outcomes. Across 468 iterations, sampled-answer and best-of-four EM improve while workflow validity stays near perfect (Figure 5). Best-of-four EM is an oracle training diagnostic, validity measures execution-rule compliance, and single-answer test accuracy is evaluated separately. PSC uses more retrievals mid-training, with a narrowing late gap.

4.5 Outcome-Sensitive Planner Diagnostics

We freeze the policy and vary outcome labels, question pairing, and length normalization to test PSC's response to sibling outcomes (RQ4).

PSC supplies planner comparisons throughout training. The full SIPPO run has 9,587 planner pairs in 2,906 of 14,976 question groups (19.40%), with signal in all 468 iterations. Against the workflow-EM gradient restricted to planner tokens, PSC has mean weighted norm ratio 0.221, mean cosine 0.695, and positive-cosine rate 0.998 (Appendix A).

Real sibling labels align better than shuffled labels on one current-policy batch. A gradient-only check on 128 fresh workflows freezes the final SIPPO policy, questions, action tokens, and loss while shuffling sibling labels. Real/shuffled labels give weighted PSC gradient norms 0.000516/0.000502 and cosines $+0.399/-0.140$ with the Planner-EM gradient (margin $+0.539$); the two PSC gradients have cosine -0.127. No update occurs. Thus gradient direction responds to sibling labels; endpoint effects come from the trained comparison above.

Figure 6 (Appendix A) gives separate frozen-policy controls. Panel (a) permutes terminal EM labels within question groups: Planner–EM alignment is 0.989 for real labels versus a 500-permutation null mean of -0.275 (one-sided value 0.001996). Panel (b) compares PSC gradients with the Planner–EM gradient: true same-question labels (TL), shuffled labels (SL), other-question pairs (OQ), and no response-length normalization (NLN) yield cosines 0.396, -0.200, 0.165, and 0.368. NLN raises the norm ratio to 6.60. Panel (c) shows the 19.4% mean coverage above. Same-question outcomes affect gradient direction, while length normalization controls magnitude; these controls do not isolate their effects on final accuracy.

Interpretation. These diagnostics delimit what the sibling signal establishes. Because PSC compares plans generated for the same question under the same planner context and execution protocol, its pairwise supervision does not require selecting a canonical intermediate trace. At the same time, an EM-correct workflow provides only a weak preference over its planner response: downstream query rewriting, retrieval, evidence interpretation, reasoning, and final synthesis also contribute to the terminal outcome. Consistent with this distinction, PSC changes execution behavior—on Qwen2.5-3B it raises the two-Search, no-Reason share from 48.0% to 87.7%, uses 11.5% more retrievals, and 5.8% fewer generated tokens—without isolating topology as the cause of the accuracy gain. We therefore interpret the frozen-policy controls as evidence that same-question outcome labels affect planner-gradient direction, rather than as a mechanistic attribution of final-answer improvements to any single planning pattern. PSC therefore refines planner preference from verifiable outcomes without treating any single intermediate action as a ground-truth label.

5 Conclusion

We presented SIPPO, which uses executable workflows as the common object for execution-grounded supervision and terminal-outcome learning. Workflow-RLOO assigns workflow EM advantages to all generated role actions, while PSC turns same-question sibling outcome differences into planner preferences without requiring a canonical intermediate trace. Across the complete 22,517-question suite, SIPPO attains the highest macro EM/F1 in the main comparison for both Qwen2.5-3B (33.62/42.12) and Qwen3-4B (38.13/47.01). Relative to the matched Workflow-RLOO endpoint, PSC adds 2.72/3.25 macro EM/F1 points on 3B and 7.01/6.72 points on 4B. On 3B, the paired macro intervals for both gains exclude zero and the micro-EM gain is 0.71 points (95% CI $[0.15, 1.27]$). The frozen-policy controls further show that PSC's planner-gradient direction responds to the real terminal outcomes of same-question sibling executions. On Qwen2.5-3B, this gain is accompanied by 11.5% more retrieval calls and 5.8% fewer generated tokens. The paired benchmark analysis is positive on HotpotQA, MuSiQue, and Bamboogle, while both 2Wiki intervals cross zero; across all 22,517 questions, SIPPO yields 159 net additional correct answers over Workflow-RLOO. The consistent improvement across both backbones further indicates that the benefit is not specific to a single model setting. These results support executable workflows as a practical interface for learning structured retrieval from final-answer supervision while preserving multiple successful intermediate strategies.

AI USE STATEMENT

DeepSeek-V4-Pro was used to generate candidate teacher workflows for the execution-grounded SFT collection described in Section 3.2 and Appendix E; candidates were executed and filtered by the stated acceptance procedure. Generative AI tools were also used for language editing and LaTeX formatting assistance. The authors take full responsibility for the final manuscript and research artifacts.

REPRODUCIBILITY STATEMENT

The main Qwen2.5-3B RL runs share the same initialization, rollout exposure, training data, optimizer configuration, and evaluation protocol (Appendix A). Losses apply only to recorded assistant responses, with retrieved text and executor bookkeeping masked (Appendix B). Logs retain the actual role inputs, responses, resolved references, and execution state (Appendix D). SFT collection accepts 800 of 4,956 workflows and yields 4,081 training and 224 validation role examples (Appendix E).

Evaluation covers all 22,517 questions; paired analyses use audited unique question–gold keys and 100,000 bootstrap resamples (Appendix F). Compute accounting includes every recorded model response and retrieval call. Relative to Workflow-RLOO, SIPPO uses 11.5% more retrievals and 5.8% fewer generated tokens; token counts exclude prompts and retrieved text (Appendix I).

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, Fan Yang, et al. Research: Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2(3):7, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Mustafa Anis Hussain, Xinle Wu, and Yao Lu. Planner-centric reinforcement learning for deep research with structure-aware reward. *arXiv preprint arXiv:2605.30824*, 2026.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 87362–87388, 2026.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Ka Yiu Lee, Yuxuan Huang, Zhiyuan He, Huichi Zhou, Weilin Luo, Kun Shao, Meng Fang, and Jun Wang. Infoseeker: A scalable hierarchical parallel agent framework for web information seeking. *arXiv preprint arXiv:2604.02971*, 2026.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Baohao Liao, Hanze Dong, Christof Monz, Xinxing Xu, Li Dong, and Furu Wei. Multi-turn on-policy distillation with prefix replay. *arXiv preprint arXiv:2607.04763*, 2026.

Jianbiao Mei, Tao Hu, Daocheng Fu, Licheng Wen, Xuemeng Yang, Rong Wu, Pinlong Cai, Xinyu Cai, Xing Gao, Yu Yang, et al. O^2-searcher: A searching-based agent model for open-domain open-ended question answering. *arXiv preprint arXiv:2505.16582*, 2025.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Zibin Meng and Kani Chen. Craft: Counterfactual credit assignment from free sibling rollouts for self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2606.29476*, 2026.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 6097–6109, 2019.

Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning. *Advances in Neural Information Processing Systems*, 38:155930–155958, 2026.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025.

Nandan Thakur, Zijian Chen, Xueguang Ma, and Jimmy Lin. Orbit: Scalable and verifiable data generation for search agents on a tight budget. *arXiv preprint arXiv:2604.01195*, 2026.

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.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198, 2020.

Jiaqi Wu, Qinlao Zhao, Zefeng Chen, Kai Qin, Yifei Zhao, Xueqian Wang, and Yuhang Yao. Gap: Graph-based agent planning with parallel tool use and reinforcement learning. *arXiv preprint arXiv:2510.25320*, 2025.

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 *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Yubao Zhao, Weiquan Huang, Sudong Wang, Ruochen Zhao, Chen Chen, Yao Shu, and Chengwei Qin. Training multi-turn search agent via contrastive dynamic branch sampling. *arXiv e-prints*, pp. arXiv–2602, 2026.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21816–21841, 2025.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

A TRAINING CONFIGURATION

Table 3: Training and evaluation settings for the controlled Workflow-RLOO/SIPPO comparison.

Configuration	Value
	Model and SFT
Backbone	Qwen2.5-3B, thinking disabled
SFT data	4,081 train / 224 validation role-level examples
SFT optimization	Full-parameter single-process FSDP; AdamW; 2 epochs / 510 updates; batch 16; microbatch 1; max length 2,048; learning rate 10^{-5}
SFT endpoint policy	Final training endpoint; the 224-row validation split is evaluated only at the end and is not used for model selection
	RL and rollout
RL data	15,000 questions: 40% 2Wiki, 30% HotpotQA, 30% MuSiQue
Question batch / workflows per question	32 / 4
Sampled workflows per rollout iteration	128
Advantage estimator	REINFORCE Leave-One-Out, at workflow granularity
Actor loss aggregation	token mean
Actor learning rate	5×10^{-7}
Rollout iterations	468 per condition
Policy surrogate	PPO clipping range $[0.8, 1.2]$; one PPO epoch; entropy coefficient 0; no KL loss or KL reward
Actor optimizer	AdamW; $\beta_1 = 0.9$, $\beta_2 = 0.999$; weight decay 0.01; gradient norm 1.0; constant learning rate
Question-preserving minibatching	Complete four-sibling question groups are preserved; implementation-level microbatching, concurrency, gradient checkpointing, and optimizer offload settings are hardware-specific
Rollout decoding	temperature 0.8; top-$p = 0.95$
PSC objective	$\beta = 1.0$; loss coefficient 0.02; equal-question normalization; coefficient fixed before the formal comparison by a gradient-budget audit targeting a weighted PSC/EM norm ratio in $[0.10, 0.30]$
PSC signal coverage	2,906/14,976 informative question groups (19.40%); 9,587 planner pairs; nonempty signal in all 468 rollout iterations
Shared comparison settings	The runs share initialization, data, 128 workflows per iteration, the Workflow-RLOO objective, learning rate, token-mean aggregation, 468 rollout iterations, rollout decoding, retriever, and formal evaluation.
	Retrieval and evaluation
Retriever / corpus / top-k	E5 / Wiki18 / 3
Evaluation decoding	temperature 0; top-$p = 1$; one rollout per question
Formal evaluation manifest	Complete answerable 2Wiki dev (12,576), HotpotQA dev (7,405), deduplicated MuSiQue dev (2,411), and Bamboogle test (125); 22,517 total; zero normalized-question overlap with RL train or validation
RL rollout node constraint	no numerical prompt bound; runtime truncation disabled
Formal-evaluation node constraint	no numerical node cap in either the planner prompt or runtime

Frozen-policy PSC visualization. Figure 6 collects the frozen-policy permutation, gradient, and informative-group diagnostics discussed in Section 4.5.

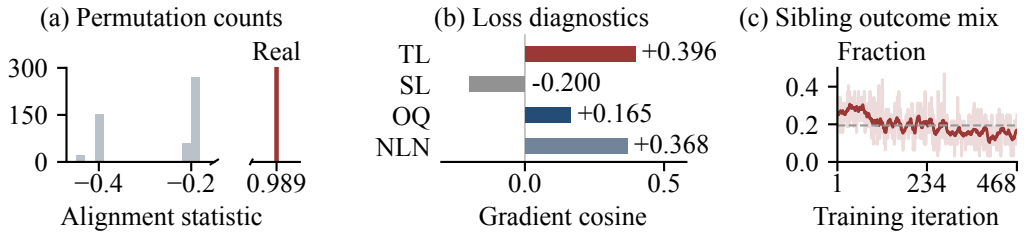


Figure 6: Frozen-policy PSC diagnostics. (a) Outcome permutations; the broken axis omits $(-0.15, 0.95)$. (b) Gradient cosine for TL (true labels), SL (shuffled labels), OQ (other questions), and NLN (no length normalization). (c) Informative-group coverage; dashed line: 19.4% mean.

B Workflow Records and Loss Masks

Table 4 closes the connection between execution and learning. Each model call is stored with the environment state that was actually available when it occurred, preserving the execution order and dependencies between actions. In SFT, the recorded assistant-response span is the target. In RL, the same generated span receives the workflow advantage; the retriever output and executor bookkeeping remain observations and are always masked. This distinction lets the shared policy learn role interfaces without backpropagating through the retrieval system.

PSC across microbatches. At each question-preserving actor minibatch, we recompute planner scores under the current pre-update policy and implement the derivative of Equation 5 as fixed per-token linear coefficients. This preserves the objective's first-order gradient at the current parameters while allowing paired plans to reside in different dynamic microbatches. These coefficients apply only to PlanningAgent response tokens; the other roles receive only the Workflow-RLOO loss.

Table 4: Workflow records and trainable response spans. Context fields are excluded from the loss; PSC applies only to planner responses.

Event	Policy-visible state	Recorded model target / learning signal
Planning	Question and planning schema	`<plan>` DAG; SFT response-token loss, then workflow EM advantage and optional PSC coefficient
Query rewrite	Resolved search task and referenced parent conclusions	Standalone query; SFT response-token loss, then workflow EM advantage
Node answer	Resolved task plus top-3 retrieved passages	Grounded node conclusion or revision request; SFT response-token loss, then workflow EM advantage
Reasoning	Resolved reason task plus parent conclusions	Node conclusion; SFT response-token loss, then workflow EM advantage
Final synthesis	Question and completed node memory	`<answer>` response; SFT response-token loss, then workflow EM advantage

C Worked Executable Workflow

Table 5 shows one successful workflow recorded during the frozen formal evaluation and makes the state transition in Section 3.1 concrete. The planner emits the symbolic dependency before either answer is known. Only after node 1 executes does the runtime replace #1# in node 2 with the stored conclusion, producing the query observed by the retriever.

Table 5: Example of a successful logged workflow, showing dependency resolution from the generated plan to final-answer synthesis.

Record field	Logged content
	Input and plan
Question	Who is Wisigard's father-in-law?
Generated plan	Node 1: search "Who is Wisigard's spouse?"; node 2: search "Who is #1#'s father?"
	Execution
Node 1	Retrieved evidence is interpreted as *Theudebert I*; the executor stores $M[1] =$ Theudebert I.
Node 2	The raw task resolves to "Who is Theudebert I's father?"; retrieved evidence is interpreted as *Theuderic I*.
	Output
Final synthesis	Prediction: *Theuderic I*; final-answer EM $= 1$.

D Prompt and Logged-Interaction Format

Every executable call is retained in the trajectory log rather than recreated from a post-hoc text trace. Concretely, each `raw_trace` records the planner's raw and parsed plan, each role-level `messages` list and corresponding model response, resolved references, rewritten queries, retrieved passages, node conclusions, and final answer. The formal-evaluation artifacts retain these fields, exposing the exact policy-visible state at every action; retrieved passages are recorded as observations and remain loss-masked.

Table 6 summarizes the dynamic fields populated at runtime. These interfaces are fixed components of the executable environment. The matched Qwen2.5-3B formal protocols impose no numerical node cap in either the planner prompt or runtime, as recorded in Table 3.

Table 6: Runtime prompt and response fields for each logged event. Inputs include resolved dependencies and logs retain the actual model responses.

Logged event	Policy-visible user content	Recorded response field
Planning	Original question plus "Generate the execution plan."	One `<plan>` block
Query rewrite	Original goal, draft dependent query, and resolved `#ID#` bindings	Standalone retrieval query
Evidence interpretation	Current resolved sub-question and top-3 retrieved passages	`<conclusion>` or `<rewrite>`
Reasoning	Current logic task after reference resolution	`<conclusion>`
Final synthesis	Original question and executed task/conclusion history	`<answer>`

Compact role prompt templates. The following templates define the shared role interfaces used by the reported formal runs. Braced lower-case fields are filled from the current workflow state. The exact instantiated `messages` lists, including retrieved passages and resolved values, remain in every trajectory record; the templates below expose the fixed system specification and the user-side fields used to instantiate each role call.

PlanningAgent. The planner emits the typed DAG before any node result is available.

```
SYSTEM:
Generate a typed Search/Reason DAG for the question.
```

```
Output exactly one XML plan with this grammar:
<plan sink="#ID#" answer_type="TYPE">
<search id="ID" subject="ENTITY_OR_#ID#" relation="RELATION"
 candidate="OPTIONAL_ANSWER_CANDIDATE">ONE_ATOMIC_RETRIEVAL_QUESTION</search>
<reason id="ID" op="OPERATOR" inputs="#ID#,#ID#"
 return="candidate|value|boolean">CONCRETE_LOGIC_TASK</reason>
</plan>
Contract: IDs start at 1 and increase by 1; references point only to
earlier nodes; Search supplies external facts and Reason performs comparison,
arithmetic, filtering, intersection, merging, or derivation; every node feeds
the single sink; a dependent Search has exactly one prior subject reference;
Reason lists every input and uses an allowed operator; do not omit an external
fact and expect Reason to retrieve it from memory. Replace all uppercase
grammar metavariables with question-specific content. Output no extra text.

USER:
Question: {question}
Generate the execution plan.
```

QueryRewriteAgent. This call is used after a dependent Search has a resolved `#ID#` binding; it produces a standalone query and never an answer.

```
SYSTEM:
Rewrite a dependent search query into a standalone retrieval query.
Replace #ID# placeholders with their provided values. Keep important
constraints from the original goal, including dates, locations, and entity
names. Make the query concise and searchable. Do not answer the query.
Output only the rewritten query, with no tags or explanation.

USER:
Original Goal: {question}
Draft: {raw_query}
# {id_1} = {resolved_value_1}
... # one line per referenced parent
Rewritten query:
```

AnswerGenerationAgent. The searcher either returns a context-supported conclusion or asks for an evidence-triggered rewrite.

```
SYSTEM:
Answer the current sub-question using only the search context.
Output exactly one of:
<conclusion>{SUPPORTED_SHORT_ANSWER}</conclusion>
<rewrite>{QUESTION_SPECIFIC_SEARCH_QUERY}</rewrite>
The uppercase strings are metavariables, not valid outputs. A rewrite must
be a real query containing key entity names. Never output generic phrases,
explanations, both tags, or an unsupported conclusion. Use <conclusion> only
when the answer is directly supported by the context.

USER:
**Current Sub-question**: {query}

**Search Context**:
{top_3_documents}
```

ReasonerAgent. This role uses only the resolved logic task and parent-node conclusions and does not retrieve new facts.

```
SYSTEM:
```

```
Resolve the logic task. Use only the information stated in the task.
Perform the required comparison, computation, extraction, filtering, or
synthesis. Output only
<conclusion>{DERIVED_SHORT_ANSWER}</conclusion>.
Replace the metavariable; an ellipsis or instruction phrase is invalid.

USER:
**Current Logic Task**: {resolved_logic_task}
```

AnswerSummarizationAgent. The final synthesizer sees the original question and the completed node history and emits only the terminal answer.

```
SYSTEM:
Synthesize the resolved sub-questions to answer the original question.
Output only <answer>{SUPPORTED_FINAL_ANSWER}</answer>.
Keep the answer concise; for a name, date, location, number, or yes/no,
output only that value. Use only facts in the resolved sub-questions. Do not
answer from memory when evidence is missing or contradictory. Do not add
explanations. Replace the metavariable; an ellipsis or instruction phrase is
invalid.

USER:
**Original Question**: {question}

**Resolved Sub-questions**:
Task 1 [SEARCH]: {subquestion_1}
Conclusion: {conclusion_1}
...
```

The prompt format specifies both the policy-visible input and the parser-visible output for every role. Figure 7 traces this record from the raw plan through validation and role calls to the final answer; each generated span retains its conditioning state and learning signal.

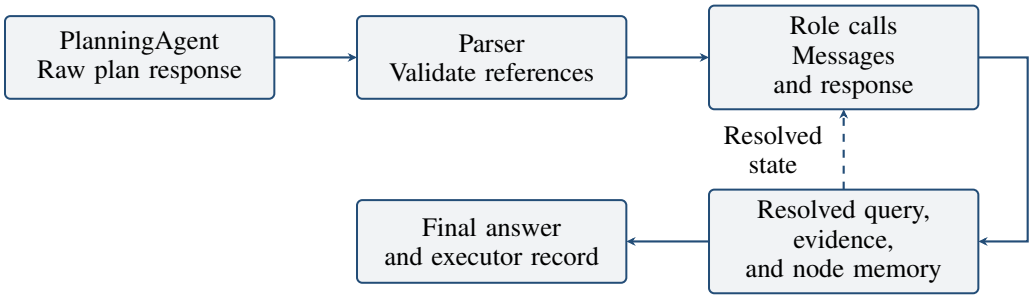


Figure 7: Workflow logging in execution order. Solid arrows trace the recorded sequence; the dashed arrow supplies resolved state to role calls. Environment observations condition responses but receive no training loss.

E Execution-Grounded SFT Collection

Table 7 makes explicit the data pipeline summarized in Section 3.2. The generator and the final agent share the same executable interfaces: a supervised action is recorded only after the preceding agent actions and retrieval observations have produced the action's input state. The gold answer gates acceptance of a completed trajectory and is absent from the policy input. We retain all five role interfaces because the same parameters must perform every role at test time; the uneven row counts follow from the generated graph structure, since a rewrite is needed only for a dependent search.

Table 7: Execution-grounded SFT collection, filtering, and supervision. Counts distinguish accepted workflows, role-level training examples, and held-out validation examples.

Stage	Frozen choice
Generation	
Source questions	4,956 questions from the official 2WikiMultiHopQA training split
Generator	DeepSeek-V4-Pro; one candidate per source question
Execution environment	Wiki18 dense retrieval, top-$k = 3$, full passages
Sampling	Temperature 0.6; top-$p = 0.95$; one candidate per question
Acceptance gates	
Planner gate	Schema, consecutive IDs, backward references, $\geq$1 search node
Evidence gate	Retrieved text states the conclusion and query target
Reason gate	Separate logic verifier marks conclusion supported
Outcome gate	Node-supported final answer exactly matches a gold answer
Accepted data	
Accepted workflows	800 (16.14%); question-level split: 760 train / 40 validation
All accepted role rows	4,305: plan 800; rewrite 731; answer 1,781; reason 193; synthesize 800
Train role rows	4,081: plan 760; rewrite 690; answer 1,687; reason 184; synthesize 760
Validation role rows	224 from the 40 held-out workflows
Audit and supervision	
Frequent rejection modes	Rewrite exhaustion; missing query target; conclusion unsupported by retrieved text; final-answer EM mismatch
Loss mask	Recorded assistant response only; environment and prompt tokens masked

F Evaluation and Statistics

The formal evaluation manifest is frozen before model comparison: all 12,576 answerable 2WikiMultiHopQA development questions, all 7,405 HotpotQA development questions, 2,411 MuSiQue development questions after removing six normalized duplicates, and all 125 Bamboogle test questions. The resulting 22,517 unique questions have zero normalized-question overlap with the 15,000-question RL mixture and the 480-question RL validation set. All three conditions use deterministic decoding, the same Wiki18/E5 top-3 retriever, the same answer normalization, and the same runtime without a numerical node cap. We report each benchmark score and the unweighted macro average. For the Qwen2.5-3B paired comparison, we join by the audited unique question–gold key, use 100,000 paired bootstrap resamples for two-sided 95% intervals, stratify macro resampling by dataset, and apply an exact McNemar test to EM disagreements.

To measure the effect of outcome learning from the shared SFT initialization, we independently re-evaluate the final Qwen2.5-3B SFT policy under the same 22,517-question manifest, PlanningAgent prompt, Wiki18/E5 top-3 retriever, deterministic decoding, and runtime settings. The evaluation covers all 22,517 questions. The SFT policy achieves 0.2480/0.3353 macro EM/F1 and 0.2800/0.3550 micro EM/F1. The PlanningAgent prompt is identical across the SFT and both RL evaluations. The paired-evaluation audit verifies identical question–gold multisets and unique join keys in every dataset. Table 10 reports the paired micro-average gains and McNemar tests based on the corresponding EM disagreements.

External-baseline evaluation provenance. The InfoSeeker row uses the ORBIT authors' reproduction checkpoint evaluated with the ORBIT controller rather than the original InfoSeeker HostAgent benchmark. We rescore its complete saved prediction/gold pairs with `qa_manager.compute_metrics.compute_scores`. Other main-table values are taken from the corresponding archived evaluator outputs.

Table 8: Full EM and F1 comparison corresponding to Table 1. Separate rows report each metric (%) on the complete 22,517-question answerable suite; Macro is the unweighted mean across datasets. Pale red/blue cells mark the best/second-best result within each backbone, metric, and column (higher is better; ties share marks).

Method	Metric	HotpotQA	2Wiki	MuSiQue	Bamboogle	Macro
Qwen2.5-3B						
Inference-time retrieval						
RAG	EM	24.97	21.87	4.52	12.00	15.84
	F1	30.74	24.90	9.97	18.93	21.13
IRCoT	EM	28.59	26.21	5.85	15.20	18.96
	F1	35.38	29.70	12.20	24.04	25.33
RL-trained search agents						
Search-R1$_{\text{GRPO}}$	EM	28.55	26.88	6.18	12.80	18.60
	F1	33.82	29.48	13.70	19.49	24.12
Search-R1$_{\text{PPO}}$	EM	29.02	29.77	8.83	17.60	21.31
	F1	34.27	31.55	16.00	26.46	27.07
ZeroSearch$_{\text{Instr.}}$	EM	28.72	26.22	5.68	12.80	18.36
	F1	34.67	29.11	12.64	25.39	25.45
StepSearch$_{\text{Instr.}}$	EM	18.31	20.40	6.84	13.60	14.79
	F1	25.15	26.20	12.43	18.87	20.66
AutoRefine$_{\text{Base}}$	EM	**42.21**	40.76	16.34	33.60	33.23
	F1	**48.85**	42.63	**25.39**	46.51	40.84
SIPPO	EM	34.92	**46.47**	**17.09**	**36.00**	**33.62**
	F1	45.28	**51.48**	25.07	**46.64**	**42.12**
Qwen3-4B						
On-policy distillation						
ReOPD	EM	36.11	31.49	12.73	34.40	28.68
	F1	41.41	32.63	19.23	45.33	34.65
OPD	EM	35.91	31.82	12.98	36.00	29.18
	F1	41.42	32.82	19.49	45.91	34.91
RL-trained search agents						
InfoSeeker$_{\text{ORBIT}}$	EM	**40.96**	40.09	18.04	44.80	35.97
	F1	**49.48**	44.05	25.55	**56.37**	43.86
Search-R1$_{\text{ORBIT}}$	EM	36.15	40.26	13.89	**46.40**	34.18
	F1	41.42	41.21	21.73	55.79	40.04
SIPPO	EM	37.41	**50.33**	**19.16**	45.60	**38.13**
	F1	48.36	**56.14**	**27.98**	55.57	**47.01**

Table 9: Paired PSC gains over Workflow-RLOO on Qwen2.5-3B, in percentage points. Cells show gains and [95% paired-bootstrap CIs] from 100,000 resamples; macro averages weight benchmarks equally.

Metric	HotpotQA	2Wiki	MuSiQue	Bamboogle	Macro
EM	$+1.07$ $[0.16, 1.96]$	$+0.08$ $[-0.74, 0.90]$	$+2.53$ $[1.16, 3.90]$	$+7.20$ $[0.80, 14.40]$	$+2.72$ $[1.00, 4.49]$
F1	$+1.78$ $[0.88, 2.68]$	-0.45 $[-1.23, 0.33]$	$+3.09$ $[1.71, 4.49]$	$+8.57$ $[2.09, 15.24]$	$+3.25$ $[1.56, 4.97]$

Table 9 reports Qwen2.5-3B per-benchmark and macro PSC increments: both macro intervals exclude zero, while both 2Wiki intervals include zero. These per-benchmark and macro gains and intervals are in percentage points; the micro-average gains in Table 10 use the original 0–1 score scale.

Table 10: Qwen2.5-3B paired micro-average EM/F1 gains on 22,517 questions. Brackets give 95% confidence intervals from 100,000 paired-bootstrap resamples; each contrast uses its stated reference condition.

Comparison	Metric	Gain	Paired 95% CI
Workflow-RLOO − SFT	Micro EM	+0.1076	$[+0.1012, +0.1140]$
	Micro F1	+0.1038	$[+0.0975, +0.1101]$
SIPPO (Ours) − SFT	Micro EM	+0.1147	$[+0.1085, +0.1208]$
	Micro F1	+0.1109	$[+0.1048, +0.1169]$
SIPPO (Ours) − Workflow-RLOO	Micro EM	+0.0071	$[+0.0015, +0.0127]$
	Micro F1	+0.0071	$[+0.0016, +0.0126]$

McNemar p: Workflow-RLOO versus SFT, 5.47×10^{-237}; SIPPO versus SFT, 2.76×10^{-289}; SIPPO versus Workflow-RLOO, 0.0146.

G Repeated Stochastic Evaluation

We additionally repeat the complete 22,517-question evaluation of SIPPO under stochastic decoding with temperature 0.7 and top-$p = 0.95$. For each backbone, we evaluate seeds 42, 43, and 44; each of the six runs covers the full evaluation suite. These sampled runs are separate from the deterministic decoding used for the main comparison. Table 11 reports EM/F1 on the original 0–1 scale. The Mean and Sample SD rows summarize variation across the three sampling seeds.

Table 11: Repeated stochastic evaluation of SIPPO on the full 22,517-question suite. Separate rows report EM and F1 on the 0–1 scale. Decoding uses temperature 0.7 and top-$p = 0.95$; Mean and Sample SD are computed across seeds 42, 43, and 44.

Backbone	Seed	Metric	2Wiki	HotpotQA	MuSiQue	Bamboogle	Macro
Qwen2.5-3B	42	EM	0.4581	0.3464	0.1655	0.3360	0.3265
		F1	0.5083	0.4281	0.2507	0.4495	0.4092
	43	EM	0.4574	0.3434	0.1672	0.3200	0.3220
		F1	0.5081	0.4249	0.2446	0.4492	0.4067
	44	EM	0.4617	0.3400	0.1663	0.3280	0.3240
		F1	0.5086	0.4220	0.2476	0.4476	0.4064
	Mean	EM	0.4590	0.3433	0.1663	0.3280	**0.3242**
		F1	0.5083	0.4250	0.2476	0.4488	**0.4074**
	Sample SD	EM	0.0023	0.0032	0.0008	0.0080	0.0023
		F1	0.0003	0.0031	0.0031	0.0010	0.0015
Qwen3-4B	42	EM	0.5043	0.3745	0.1933	0.4560	0.3820
		F1	0.5613	0.4624	0.2890	0.5520	0.4662
	43	EM	0.5018	0.3739	0.1920	0.4560	0.3810
		F1	0.5595	0.4602	0.2859	0.5568	0.4656
	44	EM	0.5087	0.3765	0.1883	0.4800	0.3884
		F1	0.5661	0.4648	0.2815	0.5637	0.4690
	Mean	EM	0.5049	0.3750	0.1912	0.4640	**0.3838**
		F1	0.5623	0.4624	0.2855	0.5575	**0.4669**
	Sample SD	EM	0.0035	0.0014	0.0026	0.0139	0.0040
		F1	0.0034	0.0023	0.0038	0.0059	0.0018

H QWEN3-4B FULL-ANSWERABLE EVALUATION

Training contract. The Qwen3-4B runs use the same scientific training settings as the Qwen2.5-3B runs in Table 3: the same execution-grounded SFT collection and optimization protocol, 15,000-question RL mixture, 32 question groups with four siblings each, 128 sampled workflows per rollout iteration, 468 rollout iterations, actor learning rate, clipping and optimizer settings, rollout decoding, PSC objective settings, retriever, node constraints, and final-endpoint selection rule. The backbone is changed to Qwen3-4B with thinking disabled; within the 4B comparison, SFT, Workflow-RLOO, and SIPPO otherwise follow the same shared planner/execution protocol and evaluation contract.

We evaluated the Qwen3-4B SFT, Workflow-RLOO, and SIPPO endpoints on the complete 22,517-question answerable suite: 12,576 2WikiMultiHopQA, 7,405 HotpotQA, 2,411 MuSiQue, and 125 Bamboogle questions. All three evaluations use deterministic decoding with top-$k = 3$ and no runtime node cap. Table 12 reports the full-suite EM and F1 scores.

Table 12: Full-suite Qwen3-4B EM/F1 on 22,517 questions. Red/blue cells with bold/underlined values mark best/second-best within each benchmark and metric (higher is better; ties share marks).

Condition	Metric	Benchmarks				Overall
		2Wiki	HotpotQA	MuSiQue	Bamboogle	Macro
		Qwen3-4B				
SFT	EM	0.313	0.280	0.150	<u>0.376</u>	0.280
	F1	0.376	0.368	0.218	<u>0.519</u>	0.370
Workflow-RLOO	EM	<u>0.391</u>	<u>0.317</u>	<u>0.161</u>	<u>0.376</u>	<u>0.311</u>
	F1	<u>0.446</u>	<u>0.419</u>	<u>0.243</u>	0.504	<u>0.403</u>
SIPPO (Ours)	EM	**0.503**	**0.374**	**0.192**	**0.456**	**0.381**
	F1	**0.561**	**0.484**	**0.280**	**0.556**	**0.470**

Relative to Workflow-RLOO, PSC adds 7.01 macro EM points and 6.72 macro F1 points. The Qwen3-4B results extend the evaluation across backbones under one protocol. The cross-backbone comparison does not isolate parameter scale from model-family and training differences.

I INFERENCE COMPUTE ACCOUNTING

Trace-level compute accounting. For each formal question, we count recorded model calls by role, actual retrieval calls, executed Search and Reason nodes, and graph depth. Generated tokens are computed with the exact Qwen2.5 tokenizer over every recorded model response; prompts, retrieved passages, and other environment text are excluded. This definition yields the means in Table 13; the corresponding role and retrieval events are retained in the trace ledger.

Table 13: Mean inference cost per question. Red/blue cells with bold/underlined values mark best/second-best calls, tokens, and retrievals (lower is better; ties share marks). Nodes and depth describe workflow structure.

Condition	Inference compute			Workflow structure	
	Model calls	Output tokens	Retrievals	Executed nodes	Graph depth
SFT	5.147	100.145	2.177	2.251	1.865
Workflow-RLOO	**4.596**	<u>94.327</u>	**1.811**	2.118	1.784
SIPPO (Ours)	<u>4.680</u>	**88.870**	<u>2.020</u>	2.019	1.655

J Offline Counterfactual Signal Audit

This audit evaluates a third estimand: counterfactual replay selection after re-executing affected descendants, rather than natural-trajectory outcome ranking (Figure 1) or the local-support audit, which reports reason-premise attribution AUC 0.8686 on 720 frozen candidate cases (Appendix K). We evaluated whether candidate process signals track the outcome of the action they are intended to describe. The offline audit used 400 logged trajectories, split evenly between 2WikiMultiHopQA and HotpotQA, and held-out counterfactual replay pairs. For each pair, affected descendants were re-executed before comparing terminal answers. Neither RL condition used these candidate signals for training. Direction accuracy measures whether a signal predicts which replay obtains the better terminal outcome; selection gain compares signal-guided replay selection with random selection.

Table 14 shows that the candidate subgraph signal achieved 89.13% direction accuracy with a cluster-bootstrap 95% interval of [80.85, 96.59]. Replay selection guided by the subgraph signal improved EM over random selection by 8.64 percentage points [6.05, 11.44]. The per-dataset gains were also positive: 11.39 points [7.55, 15.54] on 2WikiMultiHopQA and 5.40 points [1.97, 9.00] on HotpotQA. These results motivate executable counterfactuals as a validation tool for process signals before training.

Table 14: Counterfactual replay audit. Direction accuracy measures agreement with outcome ordering; selection gains compare against random selection. Brackets give cluster-bootstrap 95% confidence intervals.

Measure	Estimate	Cluster-bootstrap 95% CI
Direction accuracy	89.13%	[80.85, 96.59]
Selection EM gain over random	+8.64 points	[+6.05, +11.44]
2WikiMultiHopQA selection gain	+11.39 points	[+7.55, +15.54]
HotpotQA selection gain	+5.40 points	[+1.97, +9.00]

K Local Support Attribution Audit

This audit asks whether execution-level attribution signals distinguish locally endorsed from non-endorsed trace events; it does not ask whether those signals rank trajectories by terminal EM. Local support discrimination and terminal-outcome credit are therefore different estimands. We evaluate these signals on 720 frozen candidate cases. We predeclared three acceptance criteria: ranking better than chance, high-confidence precision and recall above threshold, and a mean score gap whose confidence interval excludes zero without a sign reversal. All audited signals passed these criteria. Table 15 reports the AUC, precision, recall, and signal-specific score gaps with their intervals.

Table 15: Local-support audit on 720 cases. Red/blue cells with bold/underlined values mark best/second-best AUC, precision, and recall (higher is better; missing entries excluded). Signal-specific gaps are not ranked.

Signal	AUC	Mean gap	Gap 95% CI	Precision	Recall
Factual confidence	<u>0.9308</u>	2.8993	[2.6529, 3.1553]	<u>0.9272</u>	<u>0.7778</u>
Reason-premise attribution	0.8686	1.9897	[1.7002, 2.3088]	**0.9744**	0.6230
Empty-search attribution	0.9102	1.6194	[1.3856, 1.8529]	–	–
Matched-search attribution	0.9226	1.9094	[1.6398, 2.1751]	–	–
Summary-evidence attribution	**0.9674**	1.5519	[1.3338, 1.7835]	0.9153	**0.9818**

L Paired Performance and Execution Patterns

We further analyze the saved predictions and workflow records from the complete 22,517-question evaluation. Figures 4 and 8 report Qwen2.5-3B paired comparisons; Figures 9 and 10 describe the recorded execution patterns of both backbones.

L.1 Paired Performance Gains

Figure 4 in the main text reports the paired gains. Relative to Workflow-RLOO, SIPPO has positive EM/F1 intervals on HotpotQA, MuSiQue, and Bamboogle, while both 2Wiki intervals include zero. The outcome decomposition in Figure 8 shows 3,924 SIPPO-only successes versus 1,342 SFT-only successes, and 2,172 SIPPO-only successes versus 2,013 Workflow-RLOO-only successes. Exact two-sided McNemar tests give $p = 2.76 \times 10^{-289}$ against SFT and $p = 0.0146$ against Workflow-RLOO.

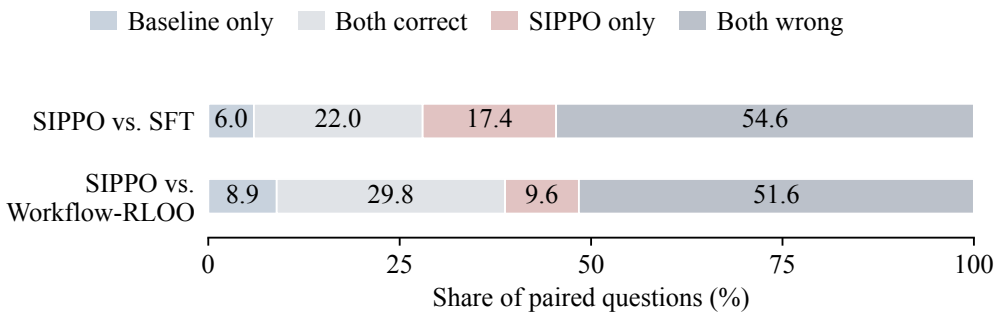


Figure 8: Paired EM outcomes on Qwen2.5-3B. Each bar partitions 22,517 questions into baseline-only correct, both correct, SIPPO-only correct, and both wrong; labels show percentages.

L.2 Execution Patterns Across Backbones

Figure 9 shows a larger S-S share under SIPPO in both backbones: 87.7% for Qwen2.5-3B and 70.1% for Qwen3-4B, compared with 48.0% and 53.2% under Workflow-RLOO. This describes a change in the distribution of executed workflows. Figure 10 further shows that conditional EM varies non-monotonically across complexity buckets. The buckets contain different questions under each policy, so conditional EM characterizes the observed executions rather than the causal effect of adding workflow nodes.

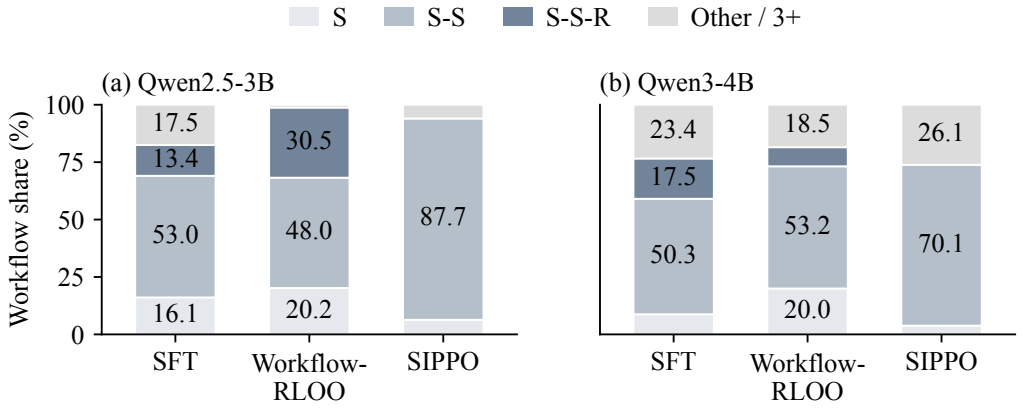


Figure 9: Recorded workflow topologies. S/R denote Search/Reason; Other / 3+ includes all remaining categories and unknown entries. Segments of at least 10% are labeled. Shares describe behavior, not topology quality.

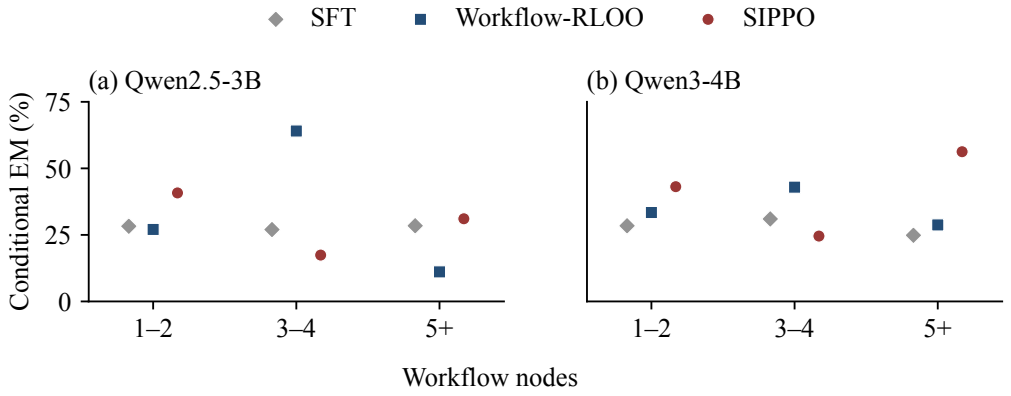


Figure 10: Conditional EM within recorded node-count buckets. Policies select different workflows and questions, so points are descriptive comparisons rather than a continuous or causal complexity trend.