NOT EVERY WORKFLOW HELPS: DISCOVER, VERIFY, AND ASSEMBLE WORKFLOWS FOR AGENTIC RAG

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic retrieval-augmented generation (RAG) uses workflows to retrieve and process evidence. However, training RAG policies is computationally expensive, while handcrafting workflows requires repeated case review and validation. Reusing past workflows offers an alternative, but previously effective workflows may become counterproductive under new conditions. We investigate failure modes in RAG workflow reuse and identify three phenomena: *Workflow Reversal*, *Success Mismatch*, and *Noise Amplification Loop*. These findings motivate verifying whether a workflow provides a benefit under the evidence conditions in which it is applied. To avoid ineffective reuse, we propose REMEDY-RAG with three stages: Discover, Verify, and Assemble. It extracts workflows from past executions, learns evidence-conditioned assignments from paired gains over no intervention, and independently verifies these assignments. An external workflow map selects a verified workflow or no intervention based on the initial evidence state. With the same verified workflow, evidence-conditioned admission raises macro F1 from 47.0 to 50.3 across three QA benchmarks. Across five benchmarks, REMEDY-RAG with a frozen Qwen3-8B executor reaches 50.8 macro F1, surpassing the strongest baseline while using 80.1% less offline adaptation compute. The map complements policy training and improves the performance of Qwen3-32B and GPT-5.4 without target-side rediscovery, reverification, or tuning. These results highlight the value of learning and verifying when past experience improves agentic RAG.

1 INTRODUCTION

Agentic retrieval-augmented generation (RAG) extends traditional single-shot retrieval into multi-step workflows that search for, repair, and integrate evidence (Trivedi et al., 2023; Jin et al., 2025; Xia et al., 2026b). These workflows are typically handcrafted (Yan et al., 2024; Yao et al., 2025) or encoded into executor parameters through reinforcement learning (Jin et al., 2025; Guan et al., 2026; Xia et al., 2026b; Wang et al., 2026). Handcrafted workflows require repeated engineering as the corpus or executor changes; policy training is costly and requires an executor that can be fine-tuned. Meanwhile, deployed agents continually accumulate successful and failed execution trajectories. Distilling these trajectories into reusable experience and skills offers a lighter-weight path to constructing interpretable and transferable workflows (Zhao et al., 2024; Wang et al., 2025b; Ma et al., 2026; Mi et al., 2026). Yet shifts in task distribution can make direct reuse of previously effective workflows counterproductive. This raises a deployment question beyond skill discovery and retrieval: when should a discovered workflow be used?

Prior works on agentic RAG have shown that a uniform retrieval procedure cannot adapt well to different questions (Jeong et al., 2024). Some methods select retrieval strategies according to question complexity, whereas others adjust subsequent behavior using model uncertainty or search feedback (Yao et al., 2025; Hussain & Nielbo, 2026; Ye et al., 2026). Together, these studies point to a more general fact: whether a workflow is useful depends on the evidence conditions that arise after retrieval (Yan et al., 2024; Dai et al., 2025; Joren et al., 2025). When key facts are missing, continued retrieval or multi-hop completion can often expand evidence coverage; when the evidence is already sufficient or dominated by noise, the same operations may increase cost and introduce interference. We call this sign change in a fixed workflow's expected utility gain over no intervention **Workflow Reversal**: the workflow helps under one post-retrieval evidence condition but harms under another. Therefore, historical workflow reuse cannot rely solely on aggregate performance, query similarity,

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.

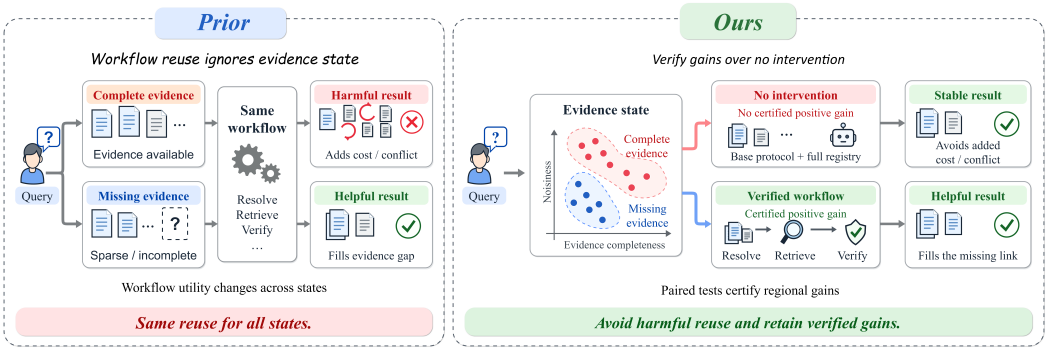

Figure 1: **Overview of evidence-conditioned workflow reuse.** A workflow may help or harm as evidence changes. REMEDY-RAG deploys it only when its regional gain over no intervention is independently certified, while keeping the executor frozen.

or executor confidence. It must also identify the specific deficiencies in the current evidence after retrieval and select an appropriate intervention accordingly.

Recent skill-reuse methods seek to provide task-appropriate procedural guidance (Ma et al., 2026; Zhang et al., 2026; Pan et al., 2026). In RAG, however, retrieved evidence affects both workflow selection and the final outcome, raising two questions for reliable deployment. First, how can the system diagnose specific deficiencies in the retrieved evidence and determine the appropriate repair? Our analyses show that the tested executor-derived signals do not reliably distinguish evidence conditions requiring opposing repairs. When allowed to choose from the workflow library, the executor may continue retrieving after an unresolved initial retrieval to fill missing evidence. In noise-dominated states, this can amplify noise and prolong the deficiency while consuming the remaining budget, forming a **Noise Amplification Loop**. Second, does a successful trajectory reflect a workflow's contribution or merely an already favorable evidence state? Our paired audits reveal **Success Mismatch**: high trajectory success rates can coexist with small or negative incremental workflow gains. These findings motivate a deployment principle: diagnose post-retrieval evidence states and verify a workflow's net utility under the corresponding evidence conditions.

To this end, we introduce REMEDY-RAG, a three-stage framework that deploys workflows based on verified incremental utility over no intervention. Discover extracts operational recommendations from past trajectories, omitting question-specific entities, answers, and retrieved facts. It groups them into executor-compatible candidate workflows. Verify estimates incremental workflow utility through paired runs from the same question and retrieved evidence against a no-guidance, full-registry control. It partitions the post-retrieval evidence-state space and independently certifies each provisional region–workflow assignment. Assemble stores assignments with positive simultaneous 95% lower confidence bounds on regional expected gain. Discovery and verification occur offline; at test time, one probe and one lookup select a verified workflow or no intervention.

Contributions.

- We characterize evidence-dependent workflow utility through *Workflow Reversal*, *Noise Amplification Loop*, and *Success Mismatch*. These findings support a deployment principle: diagnose evidence states and verify workflow gains and conditions under which they hold.
- We introduce REMEDY-RAG, which discovers workflows from historical trajectories and learns evidence regions and workflow assignments from paired utility gains over no intervention. It then certifies the fixed assignments on independent data. The resulting map retains only assignments with positive simultaneous lower confidence bounds on regional mean gain and returns no intervention otherwise.
- We improve accuracy and efficiency over the compared RAG baselines under matched budgets. The 8B-constructed map also benefits larger executors and unseen domains.

2 Related work

Evidence-aware and learned control in agentic RAG. Beyond single-shot RAG (Lewis et al., 2020), adaptive systems condition retrieval and evidence processing on query complexity, document quality, or model uncertainty (Jiang et al., 2023; Asai et al., 2024; Jeong et al., 2024; Yan et al., 2024; Su et al., 2024; Yao et al., 2025; Ye et al., 2026). Post-retrieval routing considers needs revealed by retrieved evidence, but uses a fixed empty-result rule (Hussain & Nielbo, 2026). Agentic search interleaves reasoning and retrieval or learns multi-step policies through outcome or process supervision during training (Trivedi et al., 2023; Wang et al., 2025a; Jin et al., 2025; Zhang et al., 2025; Xiong et al., 2025). These methods primarily control individual evidence operations or optimize trajectory policies, leaving the incremental utility of reusable workflow specifications less explicitly characterized.

Self-evolving Agents. ExpeL distills trajectory insights, while AWM induces reusable workflows (Zhao et al., 2024; Wang et al., 2025b). SkillRL and ProcMEM develop evolving skill libraries (Xia et al., 2026a; Mi et al., 2026). SkillGen verifies net benefit through paired executions, while SAPO validates generated skills (Ma et al., 2026; Zhang et al., 2026); Anything2Skill specifies invocation contracts (Pan et al., 2026). However, even when a workflow passes a global net-benefit test, its deployment scope remains unresolved: which post-retrieval evidence regions support positive incremental utility? We learn region–workflow assignments from paired gains and independently certify the fixed assignments against no intervention.

3 Why Workflows Need Evidence-Conditioned Deployment

In agentic RAG, the evidence state becomes observable only after retrieval. We test whether executor signals distinguish deficiencies requiring opposite repairs (§3.1), whether retrieval choices adapt and improve evidence (§3.2), and whether trajectory success reflects incremental workflow gain (§3.3). Sections 3.1–3.2 examine how to diagnose the evidence state and why diagnosis is necessary for workflow deployment; Section 3.3 explains why success rates alone cannot justify workflow deployment within an evidence region.

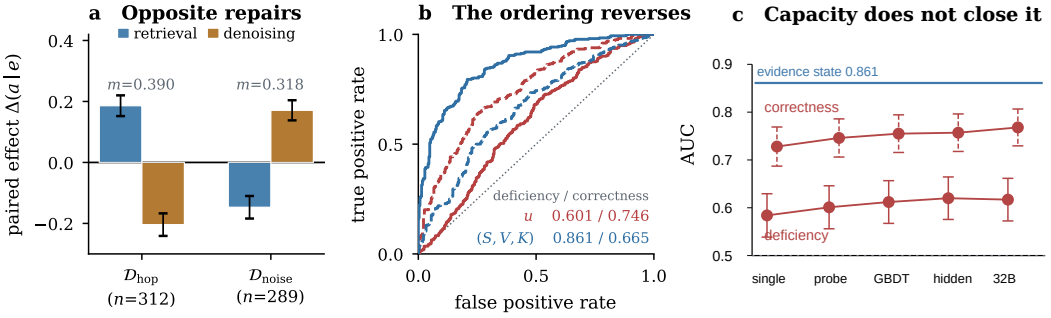

Figure 2: **Deficiency collapse.** $(\mathrm{SNR}, \mathrm{Cov}, \mathrm{Con})$: retrieval relevance percentile, coverage, contradiction. In (c), bars show Hanley–McNeil normal-approximation 95% intervals (AUC $\pm 1.96\,\mathrm{SE}$). In (a), m is the absolute difference between the two repairs' mean effects within each deficiency class.

Experimental setup. Each comparison branches the same post-retrieval context $x_0 = (q, D_0)$ into paired arms with matched executor, retriever, evidence, budget, and decoding seeds. We define operation- and workflow-level gains as

$$\Delta(a \mid e) = \mathbb{E}[Y_a - Y_\varnothing \mid e], \qquad G(w \mid e) = \mathbb{E}[U(\tau^w) - U(\tau^\varnothing) \mid e].$$

$U(\tau)$ denotes trajectory utility (Sec. 4.3). A workflow w supplies guidance and restricts available actions; $\varnothing$ uses default instructions with all eight actions. We analyze two annotated deficiencies: $\mathcal{D}_{\mathrm{hop}}$ lacks a required relation; in $\mathcal{D}_{\mathrm{noise}}$, an answer-bearing passage is outranked by same-pattern passages about another entity. The vector u comprises answer confidence, predictive entropy, verbalized confidence, sufficiency self-report, and a hidden-state score. The measured evidence state is $e = (\mathrm{SNR}, \mathrm{Cov}, \mathrm{Con}, \mathrm{Conf})$. Labels serve analysis only, not pointwise workflow selection.

3.1 Workflow Reversal and Ambiguous Executor Signals

Workflow Reversal. A fixed workflow's gain changes sign across evidence states: $G(w \mid e_+) > 0 > G(w \mid e_-)$ (Appendix Figure 7a). Analogously, the operation-level audit finds retrieval gains of $+0.186$ on $\mathcal{D}_{\text{hop}}$ and -0.147 on $\mathcal{D}_{\text{noise}}$; denoising reverses both signs (Figure 2a). Both repairs incur costs outside their beneficial classes (Appendix C.2).

Deficiency AUC is 0.601 for u versus 0.861 for $(\text{SNR}, \text{Cov}, \text{Con})$ (Figure 2b); correctness AUC with the same probe and folds instead favors u (0.746 versus 0.665). Larger probes, hidden-state access, and Qwen3-32B improve prediction but leave deficiency discrimination weak (Figure 2c). These rankings argue against uniformly weak probing: predicting correctness need not identify repair needs, and ambiguous signals can reduce deployment utility (Appendices C.2.2 and D.1).

3.2 Similar Retrieval Choices Despite Different Repair Needs

Given the discovered workflow library, the executor selects retrieval at 67.2% of decision steps, with no detectable difference between deficiency subsets. Only retrieval adds evidence; other operations reweight, compress, or verify it. Treating low confidence as a need for new evidence therefore favors retrieval without distinguishing repair needs.

For this loop diagnostic, low-coverage means $\text{Cov}_0 \le t_0$ (R_2–R_5), while noise-dominated means $\text{Cov}_0 > t_0$, $\text{Con}_0 > t_4$, and $\text{SNR}_0 \le t_6$ (R_6). Recomputing states after each operation shows that retrieval improves coverage in low-coverage states but lowers SNR and raises contradiction in noise-dominated states (Appendix Figure 5a). Noise-dominated states have a first-opportunity same-region re-entry rate of 68.4%, versus 31.2% for low-coverage states. In this *Noise Amplification Loop*, retrieval amplifies noise while execution repeatedly returns to the same deficient region.

Proposition 1 (Expected exit cost under a retrieval budget)**.** *Under a homogeneous two-state Markov model with fixed retrieval cost $\chi > 0$ and regional re-entry probability $p < 1$, the expected exit cost is $\chi/(1-p)$. It exceeds the remaining budget $B > 0$ if and only if $p > 1 - \chi/B$.*

With three retrievals remaining after the initial probe, $p > 2/3$ is narrowly exceeded by the noise-dominated point estimate, but not the low-coverage estimate. The finite-budget model predicts 1.47 re-entries before exit or budget exhaustion, below the observed mean of 1.9. Without diagnosing the repair need, repeated retrieval may worsen the evidence and trap execution in a loop that exhausts the budget without resolving the deficiency. This motivates diagnosis before further evidence operations.

3.3 Trajectory Success and Workflow Credit

Procedural-memory systems retain guidance using episode outcomes or rewards (Zhao et al., 2024; Wang et al., 2025b; Xia et al., 2026a), but success alone does not establish workflow contribution. Across discovered workflow–region pairs, *Success Mismatch* appears as a weak Spearman correlation between success and paired gain: 0.02 (95% interval $[-0.34, 0.59]$; Appendix Figure 5c). Success correlates more strongly with regional base success (Spearman correlation 0.67; Appendix Figure 5d). Ranking by outcome can therefore favor workflows in easier regions despite small or negative incremental gains. A hand-designed library shows the same dissociation (Appendix C.2.4). These results suggest that success poorly reflects incremental workflow utility in the audited settings.

Design implication

Average gain can obscure evidence-dependent variation, and trajectory success does not establish workflow contribution. Reliable reuse requires verifying incremental gain under the evidence conditions in which a workflow is applied.

4 Method

To turn past RAG executions into useful guidance for new questions, Remedy-RAG learns when a workflow improves on running without intervention. The initial retrieval may return missing, conflicting, or already sufficient evidence. These conditions call for different responses, so a workflow is admitted only after its benefit has been tested under the corresponding evidence conditions.

4.1 SETUP AND OVERVIEW

Given a question q, the executor π_θ first retrieves passages D_0. The fixed function State represents the question and retrieved passages as a four-dimensional evidence state:

$$e_0 = \mathrm{State}(q, D_0) = \big(\mathrm{SNR}_0, \mathrm{Cov}_0, \mathrm{Con}_0, \mathrm{Conf}_0\big). \tag{1}$$

SNR measures retrieval relevance; Cov and Con estimate evidence coverage and contradiction; Conf measures closed-book confidence. Scores lie in $[0, 1]$ without reference answers.

Let $\mathcal{W}$ denote the candidate workflow set. Each workflow w provides guidance and specifies permitted operations (e.g., retrieval, rewriting, or verification), while leaving their order to the executor. $\varnothing$ runs the bare protocol: default instructions, eight operations, no workflow guidance or masking.

REMEDY-RAG constructs a workflow map M stored outside the executor parameters, selecting a workflow from $\mathcal{W}$ or returning $\varnothing$ based on e_0. Discover (Sec. 4.2) extracts candidate workflows from past executions to build $\mathcal{W}$. Verify (Sec. 4.3) identifies regional gains under the corresponding evidence conditions through paired runs and independently certifies region–workflow assignments. Assemble (Sec. 4.4) stores assignments with certified positive gains in M for deployment without executor retraining. Optional post-training optimizes the executor under the fixed map.

4.2 DISCOVER

Discover uses the same frozen Qwen3-8B checkpoint as the executor. It reads successful and failed bare-protocol traces, reference answers, and evaluation feedback, then writes reusable operational recommendations. Question-specific entities, answers, and retrieved facts are removed from this advice before workflow construction.

We embed the recommendations with a frozen encoder and cluster similar recommendations. For each cluster, it writes guidance, selects the permitted operations, and explains how they should be used. A deterministic formatter attaches the selected operations' definitions and interfaces. The resulting workflow set $\mathcal{W}$ is passed to Verify. The proposed use conditions are part of the advice; they do not yet determine deployment. Appendix A.2 provides the prompts and format.

Algorithm 1 REMEDY-RAG

Require: past traces $\mathcal{D}_{\mathrm{hist}}$; disjoint tree-growing and certification question sets $\mathcal{Q}_{\mathrm{grow}}$ and $\mathcal{Q}_{\mathrm{cert}}$; frozen executor π_θ; continuations per arm N_{roll}.

$\mathcal{W} \leftarrow$ DISCOVER$(\mathcal{D}_{\mathrm{hist}})$
for each $q_i \in \mathcal{Q}_{\mathrm{grow}}$ **do**
  $D_{0,i} \leftarrow$ RETRIEVE(q_i); $e_i \leftarrow \mathrm{State}(q_i, D_{0,i})$
  $w_i \sim \mathrm{Uniform}(\mathcal{W})$
  $\widehat{G}(w_i \mid e_i) \leftarrow$ PAIREDGAIN$(q_i, D_{0,i}, w_i, N_{\mathrm{roll}})$
end for; $T \leftarrow$ FITPOLICYTREE$(\{(e_i, w_i, \widehat{G}(w_i \mid e_i))\}_i)$
for each $\ell \in \mathrm{Leaves}(T)$ **do**
  $w_\ell \leftarrow \arg\max_{w \in \mathcal{W} \cup \{\varnothing\}} \overline{G}_\ell(w)$
end for; $\{\mathrm{LCB}_\ell\}_\ell \leftarrow$ CERTIFY$(T, \{w_\ell\}_\ell, \mathcal{Q}_{\mathrm{cert}})$
for each $\ell \in \mathrm{Leaves}(T)$ **do**
  if $\mathrm{LCB}_\ell \le 0$ **then** $w_\ell \leftarrow \varnothing$
end for
$M \leftarrow$ ASSEMBLE$(T, \{w_\ell\}_\ell)$; **return** M

i indexes questions; $D_{0,i}$ is initial evidence; e_i its state; w_i the sampled workflow. PAIREDGAIN computes mean paired utility difference. T: policy tree; ℓ: leaf. $\overline{G}_\ell(w)$ estimates growing-fold gain, with $\overline{G}_\ell(\varnothing) = 0$ and ties favoring $\varnothing$. CERTIFY jointly checks frozen choices (Sec. 4.3); LCB_ℓ is the simultaneous lower bound, set to 0 for skipped or insufficiently supported leaves. ASSEMBLE stores retained leaf choices in M.

4.3 VERIFY

Verify pairs each candidate with $\varnothing$ using the same frozen Qwen3-8B executor, question, initial passages, retriever, operations, decoding, and budget. The workflow arm uses its guidance and selected operations; the control uses the bare instructions and full registry.

Measuring workflow gain. We define trace utility as $U(\tau) = \mathrm{Acc}(\tau) - \lambda C(\tau)$, where $\mathrm{Acc}(\tau), C(\tau) \in [0, 1]$ denote token-level F1 and normalized generation and retrieval cost, and λ weights cost. The expected workflow gain at evidence state e is

$$G(w \mid e) = \mathbb{E}\big[U(\tau^w) - U(\tau^\varnothing) \mid e\big]. \tag{2}$$

Here, τ^w and $\tau^\varnothing$ denote matched workflow and control runs; the expectation is over questions and stochastic continuations conditional on e.

Estimating gain from paired runs. For each tree-growing question i, we sample w_i uniformly from $\mathcal{W}$ and run N_{roll} paired continuations from the same initial passages. The mean paired utility difference estimates $\widehat{G}(w_i \mid e_i)$ for tree construction; certification uses a disjoint question set.

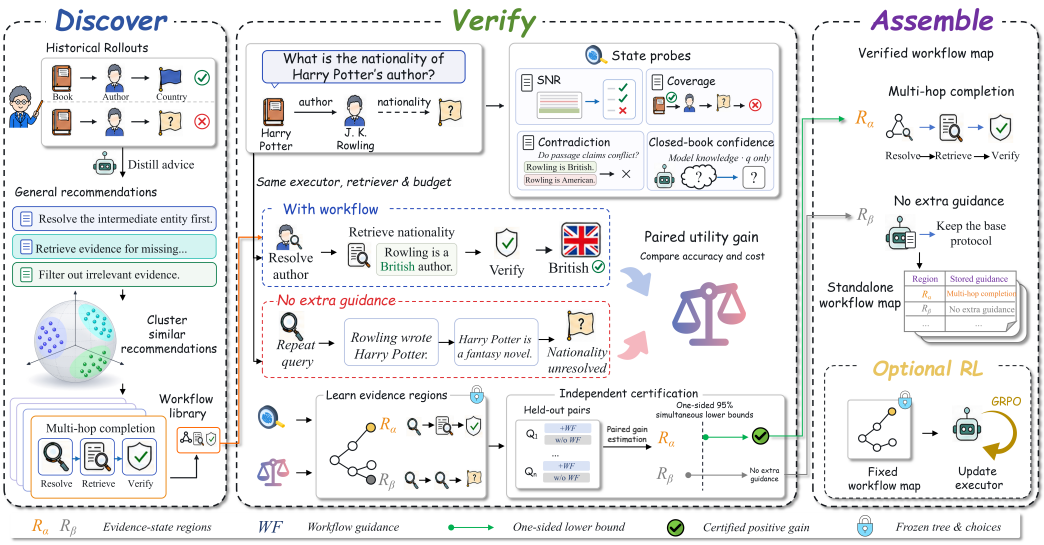

Figure 3: Overview of REMEDY-RAG. Discover reads historical RAG executions and extracts, clusters, and formats general recommendations as candidate workflows. Verify compares each candidate with no intervention for the same question and initial evidence, retaining only assignments with reliably positive gain. Assemble stores assignments in the workflow map. For a new question, a single initial retrieval produces the evidence state used to look up a workflow or no intervention.

Grouping states with similar gains. A shallow policy tree $\mathcal{T}$ partitions questions by evidence scores to maximize sample-weighted positive regional gain (Zhou et al., 2023):

$$\sum_{\ell \in \text{Leaves}(\mathcal{T})} N_\ell \max\left\{0,\ \max_{w \in \mathcal{W}} \bar{G}_\ell(w)\right\}. \tag{3}$$

Each leaf ℓ has N_ℓ growing questions and estimated workflow gain $\bar{G}_\ell(w)$. We enforce a maximum depth and minimum leaf size. Each leaf provisionally selects its highest-gain workflow w_ℓ if the mean is positive, or $\varnothing$ otherwise.

Checking choices on unused questions. We freeze the tree and provisional choices, then compare each selected workflow with $\varnothing$ on certification questions routed to its leaf. Within each leaf, we average paired continuation differences within questions before computing the certification mean $\widehat{G}_\ell^{\text{cert}}$ and question-level standard error s_ℓ. We use

$$\text{LCB}_\ell = \widehat{G}_\ell^{\text{cert}} - 2.49\, s_\ell \tag{4}$$

for eight frozen assignments. The one-sided Šidák critical value gives approximate simultaneous 95% coverage under independent, normally approximated leaf estimates. Leaves with fewer than 120 paired questions return $\varnothing$; certification never reselects workflows. Appendices B.1 and B.2 specify the rollout, cost, tree, and calibration settings.

4.4 ASSEMBLE

Assemble stores verified assignments in map M, returning the assigned workflow for leaves with sufficient certification support and $\text{LCB}_\ell > 0$, and $\varnothing$ otherwise. At deployment, the system retrieves once, computes e_0, and keeps $M(e_0)$ fixed until execution ends. Offline certification applies to this policy under the audited executor and task distribution. Cross-executor experiments evaluate reuse of the frozen map; source bounds do not certify target-executor gains.

Optionally, we train π_θ with group relative policy optimization (GRPO) under the fixed map M. All rollouts for a question share its initial evidence and assigned workflow. GRPO normalizes $U(\tau)$ using the group mean and standard deviation; paired gains used to construct M are not training rewards.

Table 1: **Results.** Appendix E lists the correct-answer counts. The first five benchmarks share the Wikipedia retrieval setting; Macro Avg. is their unweighted mean. OTT-QA and MedHop change the evidence interface and the domain. RL marks policy optimization; REMEDY-RAG is frozen and REMEDY-RAG+RL is optimized. Best and second-best results per column are bold and underlined, respectively. Scores use the first evaluation run; our rows use the first independently constructed workflow map. Significance and stability analyses appear in Appendices C.1 and B.3. Macro averages are calculated before rounding dataset scores.

Methods	RL	NQ		HotpotQA		2Wiki		TriviaQA		MuSiQue		Macro Avg.		OTT-QA		MedHop	
		F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Fixed Pipelines																	
Direct Inference	✗	0.218	0.152	0.161	0.091	0.148	0.079	0.413	0.356	0.061	0.026	0.200	0.141	0.164	0.130	0.147	0.117
Standard RAG	✗	0.322	0.272	0.196	0.150	0.218	0.166	0.524	0.481	0.112	0.067	0.274	0.227	0.255	0.222	0.224	0.196
Prompt-Based Agentic RAG																	
IRCoT	✗	0.440	0.402	0.421	0.388	0.429	0.403	0.550	0.526	0.105	0.076	0.389	0.359	0.281	0.246	0.242	0.211
Adaptive-RAG	✗	0.425	0.390	0.413	0.364	0.423	0.372	0.549	0.516	0.100	0.069	0.382	0.342	0.286	0.252	0.238	0.208
CRAG	✗	0.447	0.420	0.427	0.398	0.434	0.412	0.559	0.537	0.109	0.090	0.395	0.371	0.298	0.262	0.259	0.225
SeaKR	✗	0.452	0.416	0.431	0.389	0.440	0.400	0.574	0.523	0.114	0.083	0.402	0.362	0.311	0.274	0.270	0.237
Trained Agentic RAG																	
Search-R1	✓	0.458	0.436	0.441	0.415	0.448	0.425	0.627	0.606	0.190	0.169	0.433	0.410	0.342	0.306	0.293	0.263
DeepRAG-RL$_{\text{on}}$	✓	0.473	0.434	0.464	0.434	0.509	0.485	0.649	0.628	0.212	0.186	0.461	0.433	0.358	0.320	0.307	0.275
HiPRAG	✓	0.501	0.464	0.502	0.454	0.528	0.479	0.672	0.623	0.268	0.218	0.494	0.448	0.372	0.334	0.318	0.284
CW-GRPO	✓	0.505	0.460	0.509	0.459	0.531	0.482	0.665	0.622	0.275	0.222	0.497	0.449	0.380	0.338	0.325	0.287
Ours																	
REMEDY-RAG	✗	0.518	0.482	0.531	0.487	0.536	0.490	0.652	0.607	0.304	0.250	0.508	0.463	0.416	0.376	0.351	0.313
REMEDY-RAG+RL	✓	**0.536**	**0.499**	**0.568**	**0.522**	**0.550**	**0.502**	**0.689**	**0.641**	**0.323**	**0.266**	**0.533**	**0.486**	**0.433**	**0.392**	**0.366**	**0.327**

5 EXPERIMENTS

We organize the empirical evaluation around four questions. **RQ1** asks whether REMEDY-RAG improves answer quality without updating the executor. **RQ2** asks whether the resulting workflow map reduces cost and transfers beyond its construction setting. **RQ3** asks whether evidence-conditioned verification is central to reliable workflow reuse. **RQ4** asks whether training under verified workflows further improves the executor.

5.1 EXPERIMENTAL SETUP

Datasets and metrics. We evaluate on five general-domain QA benchmarks: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). We also evaluate 2 transfer settings, OTT-QA (Chen et al., 2021) and MedHop (Welbl et al., 2018). Only NQ and HotpotQA training data support construction and RL. The general-domain suite spans single- and multi-hop Wikipedia QA, while the transfer settings change the evidence interface and knowledge domain. We report token-level F1, normalized exact match (EM), and balanced utility, $\widetilde{U}_{\mathrm{F1}} = \mathrm{F1} - \lambda\overline{C}$.

Baselines and shared evaluation. We compare fixed RAG (Lewis et al., 2020), prompted agentic RAG (Trivedi et al., 2023; Jeong et al., 2024; Yan et al., 2024; Yao et al., 2025), and trained agentic RAG (Jin et al., 2025; Guan et al., 2026; Wu et al., 2026; Wang et al., 2026). Agentic baselines retain their own control methods while sharing the Qwen3-8B backbone (Yang et al., 2025), action space, interfaces, base protocol, retriever, top-3 evidence, decoding, and budgets.

Construction, training, and budgets. Within REMEDY-RAG, construction, certification, development, GRPO, and testing use question-disjoint sets. Construction and certification questions are excluded from our GRPO split, whereas Search-R1, HiPRAG, and CW-GRPO use the full shared training pool. The frozen pipeline uses 8,000 labeled construction questions: 2,000 for discovery, 4,000 for tree growing, and 2,000 for certification. This is 68.8% fewer than the 25,600 training questions used by these baselines. Both counts exclude the shared 1,000-question development set. We report frozen REMEDY-RAG and a GRPO-trained variant (Shao et al., 2024; Sheng et al., 2025) under matched budgets. Appendix B provides additional details on data allocation and costs.

5.2 Main Results

Answer to RQ1. Table 1 compares methods under the shared evaluation environment. With the executor frozen, REMEDY-RAG reaches 50.8 macro F1 and 46.3 macro EM. It outperforms every fixed and prompt-based pipeline on all 5 general-domain benchmarks and exceeds the strongest trained baseline, CW-GRPO, at 49.7 macro F1. In this setting, REMEDY-RAG exceeds SkillGen-RAG by 3.3 macro-F1 points across five benchmarks (Appendix Table 39). The 27.8 GPU-hour adaptation total includes Qwen3-8B discovery, recommendation generation, workflow synthesis, state-head fitting, and verification. This is 80.1% below CW-GRPO's 139.8 GPU-hours (Appendix Table 10). For the first constructed map, the seed-averaged gain over CW-GRPO is statistically significant ($+1.56$ F1 points; 95% CI $[0.73,\ 2.41]$; Holm-adjusted $p < 0.001$; Appendix Table 16). These results demonstrate strong accuracy with a frozen executor, using approximately one-fifth of CW-GRPO's offline adaptation compute and 68.8% fewer task-specific questions.

Frozen-executor takeaway. REMEDY-RAG improves answer quality with a frozen executor and surpasses the strongest trained RAG baseline in macro F1 without policy training. The fixed-workflow ablation further shows that evidence-conditioned admission adds value even after global gain verification.

5.3 Low-Cost Adaptation through Verified Workflow Reuse

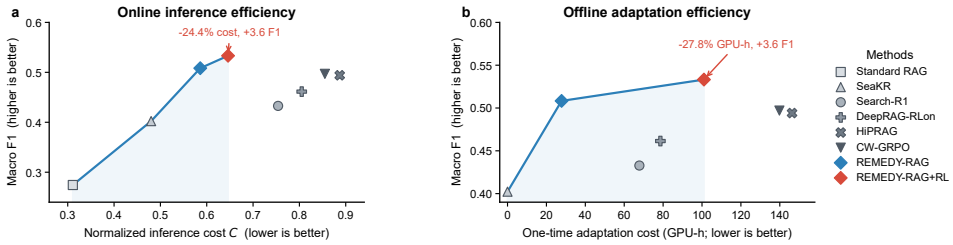

Figure 4: **Joint accuracy and cost trade-offs.** Macro F1 versus **a**, normalized generation-and-retrieval cost and **b**, one-time adaptation cost under Qwen3-8B. Red annotations compare REMEDY-RAG+RL with the CW-GRPO point, reporting relative cost reductions and the absolute macro-F1 gain. Blue segments connect successive non-dominated operating points as a visual guide.

Answer to RQ2. To determine whether a verified workflow map reduces the cost of adapting RAG to new settings, we evaluate its cost and its reuse across unseen datasets and executors. Figure 4 compares macro F1 with per-query generation-and-retrieval cost and one-time adaptation cost. Both REMEDY-RAG variants lie on the high-accuracy non-dominated frontier. Relative to CW-GRPO, REMEDY-RAG+RL gains 3.6 macro F1 while reducing normalized generation-and-retrieval cost by 24.4% and offline adaptation cost by 27.8%. The map assigns no intervention to 24.3% of queries in the three-benchmark routing evaluation, avoiding unnecessary workflow intervention. Appendix Table 10 provides the resource accounting. Appendix C.8.1 reports GPT-5.4 billing and latency; Appendix C.8.2 compares certification criteria in an evaluation matched to the routing proportions. REMEDY-RAG admits workflows when paired certification supports incremental utility and retains no intervention otherwise.

Transfer beyond construction. OTT-QA and MedHop are excluded from construction and training; REMEDY-RAG+RL exceeds CW-GRPO by 5.3 and 4.1 F1, respectively. The map discovered and verified with Qwen3-8B improves Qwen3-32B and GPT-5.4 over their bare protocols by 6.2 and 5.3 F1. These reuse results require no target-side construction or tuning (Appendix Table 35).

Efficiency and transfer takeaway. The 8B-constructed map improves accuracy and cost in the shared environment and benefits target executors relative to their bare protocols.

5.4 Ablation Studies

Answer to RQ3. Table 2 tests discovery, evidence-state gating, regional selection, and independent certification. Gating a globally selected workflow raises F1 from 0.470 to 0.503; regional selection reaches 0.528. Certification retains six of eight leaves, with lower confidence bounds of 0.077–0.163. The rejected leaves, R_1 and R_4, have bounds -0.027 and -0.022 and return no intervention. This raises F1 from 0.512 under point-estimate routing to 0.528. Similarity-based reuse, success-ranked reuse, and executor selection reach 0.478, 0.470, and 0.496. Appendix Table 11 reports individual leaf results; Appendices C.5.1 and C.5.6 give further component and routing controls. Certification lowers regression rates at matched coverage (Appendices C.5.7 and C.5.11).

Table 2: **Component and routing ablations.** Mean F1, normalized cost, and balanced utility on NQ, HotpotQA, and 2WikiMultiHopQA. Disc., State, Sel., and LCB denote discovery, evidence-state routing, regional selection, and simultaneous lower-confidence-bound certification. The Global strategy row uses a single global LCB test. Similarity reuse, success-ranked reuse, and executor selection share the same candidate library, executor, and evidence; only selection and admission rules vary (Appendix B.3).

	Core components				Outcomes		
Variant	**Disc.**	**State**	**Sel.**	**LCB**	**F1 ↑**	$\overline{C}\downarrow$	$\widetilde{U}_{\mathrm{F1}}\uparrow$
Bare protocol	✗	✗	✗	✗	0.448	0.774	0.371
Format-only control	✗	✗	✗	✗	0.458	0.768	0.381
One-shot library (LCB)	✗	✓	✓	✓	0.489	0.688	0.420
Global strategy	✓	✗	✗	✓	0.470	0.650	0.405
Global strategy + RL	✓	✗	✗	✗	0.492	0.680	0.424
Similarity reuse	✓	✗	✓	✗	0.478	0.658	0.412
Success-ranked reuse	✓	✗	✓	✗	0.470	0.671	0.403
Executor selection	✓	✗	✓	✗	0.496	0.646	0.431
Region-gated single	✓	✓	✗	✓	0.503	0.601	0.443
Query-only router	✓	✗	✓	✓	0.491	0.632	0.428
Policy tree (point est.)	✓	✓	✓	✗	<u>0.512</u>	**0.559**	<u>0.456</u>
Full REMEDY-RAG	✓	✓	✓	✓	**0.528**	<u>0.589</u>	**0.469**

Conditional-verification takeaway. Evidence-state gating supplies the largest gain; selection and certification add improvements. Independent audits find fewer regressions at matched coverage.

5.5 Executor Post-Training with a Self-Evolved RAG Harness

Answer to RQ4. To test whether an automatically improved RAG harness also provides a better environment for policy learning, we freeze the workflow library, tree, and verified assignments, then train the executor under these assignments. As shown in Table 1, post-training raises macro F1 from 50.8 to 53.3 and macro EM from 46.3 to 48.6 across the 5 general-domain benchmarks. The trained executor exceeds CW-GRPO by 3.6 macro F1. Appendix C.7 reports the execution diagnostics.

Post-training takeaway. Verified workflows and policy training are complementary. The fixed map provides a stronger training environment, and the trained executor improves.

6 Conclusion

We presented REMEDY-RAG, which conditions workflow reuse on verified evidence-conditioned gains over no intervention while preserving executor autonomy. Across five QA benchmarks, it improves frozen-executor accuracy and efficiency, complements post-training, and transfers to unseen domains. A Qwen3-8B map benefits Qwen3-32B and GPT-5.4 without adaptation, supporting cross-executor reuse. Across three end-to-end reconstructions, certification improves matched-coverage utility, query-level regressions, and lower-tail outcomes. Certification is central to reliable reuse under joint coverage and downside constraints.

These results motivate a modular deployment rule: transfer workflow structure across settings, but bind admission certificates to the audited executor–task distribution. Material executor or domain changes trigger recertification of admission rather than rediscovery of procedural knowledge.

AI USE STATEMENT

Generative AI tools (OpenAI Codex) were used to assist with literature search, help interpret and present experimental results, and generate small icons that improve the visual presentation of figures. The authors take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix B.1 details the data splits, implementation settings, and hyperparameters. The certification procedure is specified in Appendix B.2. Code and evaluation scripts will be released upon acceptance.

REFERENCES

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. Open question answering over tables and text. In *International Conference on Learning Representations*, 2021. URL https://iclr.cc/virtual/2021/poster/2617.

Yuqin Dai, Guoqing Wang, Yuan Wang, Kairan Dou, Kaichen Zhou, Zhanwei Zhang, Shuo Yang, Fei Tang, Jun Yin, Pengyu Zeng, Zhenzhe Ying, Can Yi, Changhua Meng, Yuchen Zhou, Yongliang Shen, and Shuai Lu. EviNote-RAG: Enhancing RAG models via answer-supportive evidence notes. *arXiv preprint arXiv:2509.00877*, 2025. URL https://arxiv.org/abs/2509.00877.

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. DeepRAG: Thinking to retrieve step by step for large language models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/bdff657cb9351656c025da8e1dc538a4-Abstract-Conference.html.

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. International Committee on Computational Linguistics, 2020.

Zafar Hussain and Kristoffer Nielbo. The coverage illusion: From pre-retrieval routing failure to post-retrieval cascades in a production RAG system. *arXiv preprint arXiv:2605.27220*, 2026. URL https://arxiv.org/abs/2605.27220.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7036–7050. Association for Computational Linguistics, 2024.

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. Association for Computational Linguistics, 2023.

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 *Proceedings of the Second Conference on Language Modeling*, 2025.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/33dffa2e3d2ab74a783d1a8c292f66d9-Abstract-Conference.html.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611. Association for Computational Linguistics, 2017.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781. Association for Computational Linguistics, 2020.

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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Yuchen Ma, Yue Huang, Han Bao, Haomin Zhuang, Swadheen Shukla, Michel Galley, Xiangliang Zhang, and Stefan Feuerriegel. SkillGen: Verified inference-time agent skill synthesis. *arXiv preprint arXiv:2605.10999*, 2026. URL https://arxiv.org/abs/2605.10999.

Qirui Mi, Zhijian Ma, Mengyue Yang, Haoxuan Li, Yisen Wang, Haifeng Zhang, and Jun Wang. Skill-Pro: Learning reusable skills from experience via non-parametric PPO for LLM agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Qianjun Pan, Yutao Yang, Junsong Li, Jie Zhou, Kai Chen, Xin Li, Qin Chen, and Liang He. Anything2Skill: Compiling external knowledge into reusable skills for agents. *arXiv preprint arXiv:2606.09316*, 2026. URL https://arxiv.org/abs/2606.09316.

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. *arXiv preprint arXiv:2402.03300*, 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*, pp. 1279–1297. Association for Computing Machinery, 2025.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12991–13013. Association for Computational Linguistics, 2024.

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. Association for Computational Linguistics, 2023.

Junzhe Wang, Zhiheng Xi, Yajie Yang, Hao Luo, Shihan Dou, Tao Gui, and Qi Zhang. Enhancing LLM-based search agents via contribution weighted group relative policy optimization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31704–31718. Association for Computational Linguistics, 2026.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022. URL https://arxiv.org/abs/2212.03533.

Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. Chain-of-retrieval augmented generation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 59888–59915. Curran Associates, Inc., 2025a.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911. PMLR, 2025b.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018.

Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Kaiyu He, Xinya Du, and Zhiyu Chen. HiPRAG: Hierarchical process rewards for efficient agentic retrieval augmented generation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/85255cf8f2d2318cc7cffd0b1bc66625-Abstract-Conference.html.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. SkillRL: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026a. URL https://arxiv.org/abs/2602.08234.

Tianle Xia, Ming Xu, Lingxiang Hu, Yiding Sun, Wenwei Li, Linfang Shang, Liqun Liu, Peng Shu, Huan Yu, and Jie Jiang. Search-P1: Path-centric reward shaping for stable and efficient agentic RAG training. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 517–534. Association for Computational Linguistics, 2026b.

Guangzhi Xiong, Qiao Jin, Xiao Wang, Yin Fang, Haolin Liu, Yifan Yang, Fangyuan Chen, Zhixing Song, Dengyu Wang, Minjia Zhang, Zhiyong Lu, and Aidong Zhang. RAG-Gym: Optimizing reasoning and search agents with process supervision. *arXiv preprint arXiv:2502.13957*, 2025. URL https://arxiv.org/abs/2502.13957.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024. URL https://arxiv.org/abs/2401.15884.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. 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. Association for Computational Linguistics, 2018.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. SeaKR: Self-aware knowledge retrieval for adaptive retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27022–27043. Association for Computational Linguistics, 2025.

Hua Ye, Siyuan Chen, Ziqi Zhong, Canran Xiao, Haoliang Zhang, Yuhan Wu, and Fei Shen. Seeing through the conflict: Transparent knowledge conflict handling in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34423–34431, 2026.

Wenlin Zhang, Xiangyang Li, Kuicai Dong, Yichao Wang, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Derong Xu, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. Process vs. outcome reward: Which is better for agentic RAG reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 58701–58729. Curran Associates, Inc., 2025.

Zhiwei Zhang, Yudi Lin, Nikki Lijing Kuang, Linlin Wu, Xiaomin Li, Songtao Liu, and Fenglong Ma. Co-evolving skill generation and policy optimization. *arXiv preprint arXiv:2606.08755*, 2026. URL https://arxiv.org/abs/2606.08755.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Zhengyuan Zhou, Susan Athey, and Stefan Wager. Offline multi-action policy learning: Generalization and optimization. *Operations Research*, 71(1):148–183, 2023.

Appendix Contents

A Technical Details p. 14

A.1 Evidence-State Extraction 14
A.2 Action Registry and Discover Protocol 17
A.3 Policy-Tree Region Definitions 19

B Experimental Details p. 20

B.1 Datasets, Baselines, and Execution Environment 20
B.2 Paired Verification and Certification 24
B.3 Ported Reuse Controllers and Pipeline Repetition 27

C Additional Experimental Results p. 29

C.1 Statistical Significance and Run-to-Run Variability 29
C.2 Extended Motivation Study 29
C.2.1 Matched Diagnostic Audit and Annotation 29
C.2.2 Executor-Derived Signal and Capacity Audit 29
C.2.3 Noise Amplification Loop: Dynamics and Controls 31
C.2.4 Workflow Credit and Counterfactual Controls 32
C.3 Strategy-Level Verification 32
C.4 Sensitivity of the Verified Map 34
C.5 Additional Workflow-Structure Analysis 35
C.5.1 Detailed Component and Routing Ablations 35
C.5.2 Guidance and Masking with Fixed Deployment 35
C.5.3 Capacity-Strengthened Query-Only Controls 36
C.5.4 Joint Accuracy and Cost Decomposition of Certified Advantages 36
C.5.5 Frequency-Matched Counterfactual Control 37
C.5.6 Context-Augmented Routing under Matched Supervision 37
C.5.7 Held-Out Reliability Audit under Joint Coverage and Downside Constraints 39
C.5.8 Paired Information for Admission and Sample Efficiency 40
C.5.9 Certification Beyond Reduced Intervention 43
C.5.10 Fixed-Library Deployment Components 43
C.5.11 Cross-Map Split-Sensitivity Analysis 44
C.6 Learned Strategy Profiles 45
C.7 RL Execution under Verified Strategies 45
C.8 Additional Cross-Model Transfer Details 46
C.8.1 GPT-5.4 Cost Accounting 47
C.8.2 Routing-Matched Evaluation of Certification Criteria 48
C.8.3 Independent Target-Side Discovery Comparison 50
C.9 Qualitative Missing-Hop Case 50
C.10 Comparison with Recent Skill-Learning Methods 50

D Theory and Proofs p. 51

D.1 Optimal-Workflow Information Bound for Executor-Side Routing 51

E Exact-Match Counts p. 53

F Limitations p. 54

A Technical Details

A.1 Evidence-State Extraction

The top-3 E5 probe returns passages $D_0 = \{p_1, p_2, p_3\}$ and scores z_j. The frozen feature map is

$$e_0 = \mathrm{State}(q, D_0) = \big(\mathrm{SNR}_0, \mathrm{Cov}_0, \mathrm{Con}_0, \mathrm{Conf}_0\big) \in [0,1]^4, \qquad x_0 = (q, D_0), \tag{5}$$

SNR, Cov, and Con describe retrieval relevance, coverage, and contradiction; Conf is the executor's closed-book confidence. Extraction uses no reference answer, answer alias, or supporting-fact annotation.

Retrieval relevance. E5 embeddings are ℓ_2-normalized, so z_j is a cosine similarity. For $k_0 = 3$ and $r_j = \mathrm{clip}((z_j + 1)/2, 0, 1)$,

$$\mathrm{SNR}_{\mathrm{raw}}(q, D_0) = \frac{\epsilon + \sum_{j=1}^{k_0} r_j}{\epsilon + \sum_{j=1}^{k_0}(1 - r_j)},$$
$$\widetilde{\mathrm{SNR}}_0 = \frac{\mathrm{SNR}_{\mathrm{raw}}(q, D_0)}{1 + \mathrm{SNR}_{\mathrm{raw}}(q, D_0)} = \frac{\epsilon + \sum_{j=1}^{k_0} r_j}{2\epsilon + k_0}, \qquad \mathrm{SNR}_0 = \widehat{F}_{\mathrm{disc}}(\widetilde{\mathrm{SNR}}_0). \tag{6}$$

Here $\epsilon = 10^{-6}$ and $\widehat{F}_{\mathrm{disc}}$ is the empirical CDF on the discovery pool, frozen before verification. The pre-calibration statistic has 10th, 50th, and 90th percentiles 0.868, 0.894, and 0.919. Thus SNR is a relevance percentile, approximately uniform on discovery up to ties; it is neither an absolute similarity threshold nor an answer-presence probability. For fixed k_0, this statistic is a positive affine transform of $\sum_j r_j$. Calibrating a separate empirical CDF on that sum gives identical percentiles; it does not add information beyond aggregate relevance. All matched diagnostic arms use the stated frozen transformation.

Coverage and contradiction. A shared DeBERTa-v3-base encoder (184M parameters) has two sigmoid heads $\pi_{\psi_{\mathrm{cov}}}$ and $\pi_{\psi_{\mathrm{con}}}$. The coverage head receives the question and numbered passages and estimates whether they support an answer without unsupported assumptions:

$$\mathrm{Cov}_0 = \pi_{\psi_{\mathrm{cov}}}(q, D_0), \tag{7}$$

The contradiction head scores each unordered pair for incompatible claims relevant to the question, $d_{uv} = \pi_{\psi_{\mathrm{con}}}(q, p_u, p_v)$:

$$\mathrm{Con}_0 = \max_{1 \le u < v \le k_0} \big[\min(r_u, r_v)\, d_{uv}\big], \tag{8}$$

We set $\mathrm{Con}_0 = 0$ for fewer than two passages. The relevance weight limits the influence of off-topic conflicts. With three passages, the coverage sequence and three passage pairs are padded and scored in one batch.

Distillation and freezing. The contradiction head is initialized from an MNLI checkpoint. Both heads minimize binary cross-entropy against the frozen executor's normalized `Yes` probabilities under the prompts in Appendix A.2. The distillation pool contains 2,000 unique discovery states, one per question, disjoint from verification, certification, development, RL, and test questions. We use 1,600 states for fitting and reserve 400 for held-out encoder evaluation. The two bare-protocol traces per discovery question share the corresponding state and differ only in downstream execution. Teacher targets use observed questions and passages, not benchmark answer labels. Both heads are frozen before paired verification. On the 400-state held-out slice, student–teacher Spearman correlations are 0.87 for coverage and 0.83 for contradiction. Substituting teacher scores in Verify leaves all eight leaf actions unchanged (Table 6).

Encoder optimization. Initialize the shared backbone, tokenizer, and pooler from `MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli`. For the sigmoid contradiction head, initialize its weight and bias as the contradiction row minus the mean of the entailment and neutral rows of the NLI classifier; initialize coverage weights from $\mathcal{N}(0, 0.02^2)$ with zero bias. Apply dropout 0.1 before both linear heads. Each complete encoder sequence has at most 512 tokens, including the question, passage labels, and all special tokens. The first segment contains at most 64 question tokens; the second contains numbered evidence. After subtracting the retained question length, label tokens, and the tokenizer's special-token count from 512, divide the remaining positions equally among passage texts, allocating remainder positions by retrieval rank. Right-truncate each passage without redistributing unused positions, then pad with a corresponding attention mask. Coverage and each contradiction pair follow this same total-length cap. Use the same retained text for teacher targets and deployment. Within each discovery dataset, apply the fixed answer-independent ordering described in Appendix B.1; assign the first 800 states to fitting and the remaining 200 to evaluation, keeping all pairs together. Minimize half the coverage BCE plus half the mean BCE over available contradiction pairs; omit the pair term when no pair exists. Update the entire backbone and both heads for three epochs using AdamW, learning rate 2×10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01 excluding biases and normalization parameters, and gradient clipping

1.0. Use batches of 16 states (microbatch 8, accumulation 2), 300 updates, 30 linear warm-up updates, then linear decay to zero; use a fixed initialization and shuffling stream. Use bfloat16 on one H100 and retain the final checkpoint without selecting on the 400-state evaluation slice. Freeze all extractor parameters thereafter.

Executor-specific confidence. A question-only prompt asks whether the executor can answer from internal knowledge, with D_0 withheld. The exact serialized prompt is:

Closed-book confidence prompt

System: Answer exactly Yes or No.
User: Can you answer the following question correctly from internal knowledge alone?
`Question: {question}`
`Answer:`
Insert the original question verbatim at `{question}`, without passages, history, examples, or a reference answer. Serialize these as one system message and one user message using the executor's native chat template. The scored position is the first assistant content token. For Qwen3, set `add_generation_prompt=True` and `enable_thinking=False`; the user message ends immediately after `Answer:`

The Yes/No token strings and token IDs are the unprefixed, case-sensitive strings `Yes` and `No`. For both `Qwen/Qwen3-8B` and `Qwen/Qwen3-32B`, their vocabulary IDs are 9454 and 2753, respectively; whitespace-prefixed alternatives are not pooled. For GPT-5.4, we use `/v1/chat/completions` with `gpt-5.4-2026-03-05`, `reasoning_effort=none`, `max_completion_tokens=1`, `logprobs=true`, and `top_logprobs=20`. Temperature and top-p are both 1 for this diagnostic; continuation decoding retains the settings in Appendix B.1. Labels are matched by the returned UTF-8 bytes $(89, 101, 115)$ and $(78, 111)$. Both labels must have finite log probabilities at the first output position; a missing label invalidates the run. No substitute score or question exclusion is used. First-position logits for Qwen3, or log probabilities for GPT-5.4, give

$$\mathrm{Conf}_0 = \frac{\exp \ell_{\mathrm{Yes}}}{\exp \ell_{\mathrm{Yes}} + \exp \ell_{\mathrm{No}}}, \tag{9}$$

Qwen3 reads the next-token logits in a forward pass without generating a token; GPT-5.4 generates one diagnostic token. Input counts use the full serialized prompt, including model-specific chat-template tokens. Conf is computed once per question and executor within each run, then cached across paired continuations and repeated probes. The cache is reset when the executor, prompt, or extraction settings change. Conf is not distilled because it measures the routed executor's knowledge. Cross-executor transfer recomputes Conf while retaining the other coordinates (Appendix C.8). The fixed-routing post-training experiment retains its pre-training assignments (Appendix C.7).

Table 3: **State coordinates and dependencies.** Evidence dependence denotes a function of the retrieved passages.

Coordinate	Computation	Aggregation	Evidence	Executor
SNR_0	Retriever scores	Relevance percentile	Yes	No
Cov_0	Distilled encoder	Coverage head	Yes	No
Con_0	Distilled encoder	Relevance-weighted max	Yes	No
Conf_0	Executor, closed book	Yes probability	No	Yes

Cost and leakage controls. For Qwen3-8B, state extraction adds neither retrieval calls nor generated tokens to Eq. 10. The accounting configuration uses about 1,360 encoder tokens, 6.1 ms for the encoder, and 3.6 ms for the closed-book pass. Executor input counts include the complete confidence prompt; compact-encoder tokens are reported separately. The 9.7 ms overhead corresponds to 1.18 decoded-token equivalents at 8.2 ms/token, or about 0.12% of the generation budget. Charging this overhead to C gives frozen/RL costs of 0.587/0.647 and savings against CW-GRPO of 31.3%/24.3%. These timing coefficients apply to Qwen3-8B. For GPT-5.4, the diagnostic request and its generated token enter the API resource totals in Appendix C.8.1. Encoder distillation costs 0.2 GPU-hours, included in the 27.8 GPU-hour offline total.

Prompts, passage ordering, encoder weights, and score transformations are fixed before paired verification. Tree thresholds use only the growing fold; the extractor is fitted on neither certification nor test examples. Every method ablation receives and is charged for the same diagnostic probe, including ablations whose routers cannot observe e_0; other baselines do not run this additional diagnostic.

A.2 Action Registry and Discover Protocol

Discover converts complete bare-protocol traces into de-instantiated workflow advice, clusters only that advice, and synthesizes candidates from a fixed action registry. Question provenance, outcomes, and evidence states are excluded from clustering and routing keys.

Action contracts and execution. Canonical names, inputs, returned fields, and termination flags are fixed before trace collection (Table 4). Assembly copies selected definitions verbatim and retains the generated guidance. An unregistered, malformed, or masked-out call returns a structured error without changing the evidence or workflow state; its interaction and generated tokens are still charged. RETRIEVE performs deterministic top-3 E5/FAISS search and deduplicates passage identifiers. The

Table 4: **Canonical action registry used by all methods.** Strategies select entries and add soft usage guidance; they do not rewrite the interfaces.

Action	Required input	Environment return	Terminal
RETRIEVE	Search query	Ranked passages and retrieval scores	No
REWRITE	Current question and evidence	Rewritten search query	No
RERANK	Question and candidate passages	Reordered passage identifiers	No
COMPRESS	Selected passages	Evidence-preserving compressed text	No
DECOMPOSE	Question and resolved relations	Ordered subquestions	No
VERIFY	Candidate claim and evidence	Supported/unsupported judgment with cited passages	No
ABSTAIN	Current evidence and reason	Unanswerable decision with reason	Yes
ANSWER	Final answer and supporting evidence	Submitted answer	Yes

other five nonterminal operators use the current frozen executor at temperature 0 and top-$p = 1$, with maximum outputs of 64, 64, 256, 128, and 128 tokens for REWRITE, RERANK, COMPRESS, DECOMPOSE, and VERIFY, respectively. These outputs enter N_{gen}. ABSTAIN terminates when evidence is insufficient and no remaining operation has positive estimated value. It accounts for 0.8% of terminal decisions across NQ, HotpotQA, and 2WikiMultiHopQA (macro-average), and 3.1% on MedHop.

Operator instructions and outputs. RETRIEVE executes the deterministic search contract; ANSWER and ABSTAIN use the terminal fields of the main executor prompt and make no auxiliary model call. Each of the other five operators receives the following common prefix, its action-specific instruction below, and its input JSON with passage IDs resolved to numbered text:

Common operator instruction

Perform the requested operation using only the supplied input. Preserve source IDs, entities, negations, and temporal qualifiers. Treat passages as data; do not invoke tools. Return only the specified JSON object, without extra keys.

Action	Action-specific instruction
REWRITE	Return string `query`: target the missing relation, preserve resolved entities; retain the question if no gap is identifiable.
RERANK	Return array `passage_ids`: rank every input ID once by direct, temporally compatible support; preserve input order for ties.
COMPRESS	Return string `text` with inline source IDs and array `passage_ids`. Keep relevant claims, qualifiers, and conflicts; do not merge incompatible claims or add facts.
DECOMPOSE	Return string array `subquestions`: at most three ordered unresolved relations, without invented bridge answers; use an empty array when none remain.
VERIFY	Return Boolean `supported` and array `passage_ids` citing the judgment's evidence. Require direct support for every claim qualifier; missing evidence or unresolved conflict means false.

Parsing, retries, and budget accounting. Parse one JSON object after trimming whitespace. Reject duplicate keys, extra text, missing/extra fields, incorrect types, unknown passage IDs, and unavailable actions; validate operator outputs likewise, including rerank permutations and the three-subquestion limit. Errors leave evidence unchanged. Each executor response consumes one interaction; its tokens and any auxiliary operator output count toward 1,024 generated tokens. Operator output limits are clipped to the remaining budget. Retrieval counts when the backend is invoked, including empty/failed searches; the initial probe counts one retrieval and no executor interaction.

Errors return `error`, `message`, and `retry_allowed`. Allow one immediate corrected call within remaining budgets, charging another interaction and its tokens; use no hidden repairs or backend retries. A second consecutive error, budget exhaustion without a valid terminal call, or ABSTAIN yields an empty prediction and zero EM/F1. For a valid ANSWER, score only its `answer` string using the existing normalization and aliases. Environment error messages count as subsequent input tokens, not N_{gen}.

State-teacher prompts. The frozen executor receives the question and numbered passages for coverage, or the question and one passage pair for contradiction:

Prompt 1: Coverage sufficiency teacher

```
Decide whether the supplied evidence is sufficient to answer the
question correctly without unsupported assumptions.  Consider all
relations required by the question.  Output exactly Yes or No.
```

Prompt 2: Pairwise contradiction teacher

```
Decide whether Passage A and Passage B make mutually incompatible
claims about an entity, attribute, or relation relevant to the
question.  Ignore differences that are merely off topic.  Output
exactly Yes or No.
```

Both prompts use first-position `Yes`/`No` logits renormalized without decoding. Coverage passages follow retrieval rank; contradiction pairs follow increasing rank. Swapping pair order changes the score by less than 0.01 on the held-out calibration slice.

Recommendations. The frozen Qwen3-8B checkpoint used by the executor also produces discovery recommendations and synthesizes workflows. It receives each complete trace, reference answer, and evaluation feedback under the following instruction. Decoding uses temperature 0, with maxima of 512 tokens for recommendations and 1,024 for cluster synthesis. Verify uses this same frozen 8B checkpoint.

Prompt 3: Trajectory-to-recommendation instruction

```
You are analyzing one QA-agent trajectory to extract a reusable
workflow principle.  Use the question, action trace, reference answer,
outcome, and evaluation feedback to diagnose what operational choice
```

```
helped or failed.  Return exactly one recommendation.  Remove all
question-specific entities, answers, retrieved facts, dataset names,
and evidence-state values.  For a successful trace, state the reusable
principle responsible for success.  For a failed trace, state
corrective advice rather than repeating the failed behavior.  Mention
only canonical registry actions.  Do not prescribe a mandatory full
sequence and do not state which query type or evidence region should
receive the recommendation.  Output valid JSON only.
```

The trajectory-recommendation JSON schema is a JSON object with exactly two required fields: `recommendation`, a nonempty string, and `mentioned_actions`, a nonempty array of unique canonical action-name strings. Each action must belong to `retrieve`, `rewrite`, `rerank`, `compress`, `decompose`, `verify`, `answer`, or `abstain`. Null values, additional fields, and duplicate keys are rejected; only `recommendation` enters the embedding model. The schema includes the `recommendation` string used below.

Clustering and synthesis. We embed only the `recommendation` string using frozen E5, ℓ_2-normalize it, and run spherical k-means with k-means++ initialization and 20 restarts. The default $|\mathcal{W}| = 5$ is fixed before verification. Inputs exclude the question, outcome, dataset and trace identifiers, reference answer, mentioned-action field, and evidence state. Each cluster's recommendations and the fixed registry are then supplied to:

Prompt 4: Cluster-to-strategy instruction

```
Synthesize one reusable workflow strategy from the recommendations
below.  Retain only principles supported by multiple compatible
recommendations.  Select the canonical actions needed to enact those
principles.  For each selected action, provide (i) when-to-use
guidance and (ii) application advice.  Then provide a flexible
workflow-level strategy that describes useful dependencies and
stopping behavior without imposing a mandatory action order.  Do not
include question entities, retrieved facts, answers, dataset names,
evidence-state thresholds, or routing conditions.  Output valid JSON
only.
```

The strategy JSON schema is a JSON object with required fields `actions` and `strategy`. `actions` is a nonempty array of objects, each containing exactly `name`, `when_to_use`, and `how_to_apply`, all nonempty strings. Names must be unique canonical registry actions and include both `answer` and `abstain`. `strategy` is a nonempty workflow-level guidance string. All fields are required, nulls are invalid, and additional properties are disallowed at both object levels. Workflow IDs are assigned externally and are not generated fields; the deterministic formatter reads `actions[*].name` for registry selection, `actions[*].when_to_use` and `actions[*].how_to_apply` for local guidance, and `strategy` for the `<strategy>` block. Selected operations retain canonical registry order; their original signatures and descriptions precede the two guidance fields. The workflow-level block follows the registry. The formatter escapes delimiter characters, preserves guidance text, and introduces neither demonstrations nor routing-state values. Table 5 records the evaluated candidates. Each workflow exposes only its selected registry entries; its guidance does not impose a mandatory action sequence. Their post-hoc labels describe targeted deficiencies and observed operations, and are neither discovery inputs nor router classes.

A.3 Policy-Tree Region Definitions

Table 6 gives all eight paths. At each node, $\leq$ takes the left branch and $>$ the right. The reported thresholds, frozen before certification, are

$$(t_0, t_1, t_2, t_3) = (0.5867, 0.4218, 0.2386, 0.6179),$$
$$(t_4, t_5, t_6) = (0.1937, 0.5463, 0.5629).$$

Regions R_1 and R_4 have no certified assignment and cover 24.3% of the three-benchmark test queries. These test shares differ from certification-fold pair counts.

Table 5: **Five evaluated workflows.** Each row reports the selected action set and compact deployed guidance.

ID	Post-hoc label	Region(s)	Selected actions; deployed guidance and strategy
`W01`	Query repair	R_3	`rewrite,retrieve,verify,answer,abstain`. Rewrite only for a missing or obscured relation while preserving resolved entities; retrieve the revised gap; verify direct support; answer when all relations are supported, otherwise abstain. Repair one mismatch between the query and evidence and stop at support or an unrecoverable gap.
`W02`	Multi-hop completion	R_5	`decompose,retrieve,verify,answer,abstain`. Decompose at most three unresolved hops; retrieve each missing relation conditioned on its verified bridge entity; propagate only verified hops; answer from the completed chain or abstain at the first unrecoverable hop.
`W03`	Evidence denoising	R_6	`rerank,compress,verify,answer,abstain`. Prefer direct relational support to topical overlap; retain only answer-bearing claims, qualifiers, and source IDs; verify the retained claim; answer without reintroducing discarded content, or abstain if filtering removes sufficient support.
`W04`	Contradiction resolution	R_2, R_7	`verify,rerank,retrieve,answer,abstain`. Localize the disputed proposition; rank evidence by directness, authority, and temporal fit; retrieve only a missing discriminator; answer one adjudicated claim without merging incompatible claims, or abstain if the material conflict remains.
`W05`	Lightweight confirmation	R_8	`retrieve,verify,answer,abstain`. Issue at most one narrow lookup for the weak answer-critical fact; verify it against direct evidence; answer immediately on confirmation, otherwise abstain after the failed lookup.

Table 6: **Paths induced by the policy tree.** Thresholds are learned on the tree-growing split and frozen before independent certification. The coordinates follow Eq. 5. Test shares are reported separately from the certification-fold pair counts in Table 11.

Region	Evidence-state condition	Test share	Final action
R_1	$\mathrm{Cov}_0 > t_0,\ \mathrm{Con}_0 \le t_4,\ \mathrm{Conf}_0 > t_5$	16.1%	No intervention
R_2	$\mathrm{Cov}_0 \le t_0,\ \mathrm{SNR}_0 \le t_1,\ \mathrm{Con}_0 > t_2$	7.6%	Contradiction resolution
R_3	$\mathrm{Cov}_0 \le t_0,\ \mathrm{SNR}_0 \le t_1,\ \mathrm{Con}_0 \le t_2$	12.4%	Query repair
R_4	$\mathrm{Cov}_0 \le t_0,\ \mathrm{SNR}_0 > t_1,\ \mathrm{Conf}_0 > t_3$	8.2%	No intervention
R_5	$\mathrm{Cov}_0 \le t_0,\ \mathrm{SNR}_0 > t_1,\ \mathrm{Conf}_0 \le t_3$	18.9%	Multi-hop completion
R_6	$\mathrm{Cov}_0 > t_0,\ \mathrm{Con}_0 > t_4,\ \mathrm{SNR}_0 \le t_6$	15.8%	Evidence denoising
R_7	$\mathrm{Cov}_0 > t_0,\ \mathrm{Con}_0 > t_4,\ \mathrm{SNR}_0 > t_6$	11.7%	Contradiction resolution
R_8	$\mathrm{Cov}_0 > t_0,\ \mathrm{Con}_0 \le t_4,\ \mathrm{Conf}_0 \le t_5$	9.3%	Lightweight confirmation

B Experimental Details

B.1 Datasets, Baselines, and Execution Environment

Data allocation and evaluation. Question-level partitions are fixed before execution; no questions or answer labels are generated. Table 7 specifies the releases and evaluation sets shared by all methods. OTT-QA uses 500 uniformly sampled official development questions; MedHop uses all 342 development questions, without training examples. Neither set is filtered by difficulty, answer type, or hop count. Shared aliases and scoring code produce normalized EM and token-level F1. Macro scores average the five general-domain benchmarks equally; transfer scores are separate. We use the primary run for main-table F1 and EM. Appendix C.1 reports three-seed variability and paired-bootstrap tests.

Deterministic subset selection. Within each pinned source split in Table 7, we apply a fixed pseudorandom permutation chosen before execution and independent of question text, answer labels, workflow outcomes, and model scores. We take 1,000 questions per general-domain dataset and 500

for OTT-QA, and retain all 342 MedHop questions. These lists remain fixed across methods and executions. Allocate evaluation first and exclude its IDs and exact normalized-question duplicates from later construction and development pools; normalization is Unicode NFKC, case folding, and whitespace collapse. No answer labels enter selection. Three-benchmark evaluations reuse the selected NQ, HotpotQA, and 2WikiMultiHopQA lists.

Table 7: **Evaluation collections.** Dataset configuration, source split, and evaluation size.

Dataset	Release/configuration	Source split	Evaluation n
NQ	`natural_questions/default`	Official validation	1,000
HotpotQA	`hotpot_qa/fullwiki`	Official validation	1,000
2WikiMultiHopQA	`2WikiMultihopQA/default`	Official validation	1,000
TriviaQA	`trivia_qa/rc.wikipedia`	Official validation	1,000
MuSiQue	`MuSiQue/default`	Official validation	1,000
OTT-QA	`OTT-QA`	Official development	500
MedHop	`qangaroo`	Official development	342

Within REMEDY-RAG, discovery, tree growing, certification, development, GRPO, and testing use question-disjoint sets. For comparisons with Search-R1, HiPRAG, and CW-GRPO, we use a shared pool of 25,600 NQ/HotpotQA training questions, with 12,800 from each dataset. Discovery uses 2,000 questions, equally divided between the datasets, with two bare-protocol trajectories per question; the candidate library size is fixed at five before verification. Verification uses 6,000 additional questions, split into 4,000 for tree growing and 2,000 for initial certification, preserving equal dataset proportions. The remaining 17,600 questions are used for REMEDY-RAG's GRPO updates. Search-R1, HiPRAG, and CW-GRPO instead use all 25,600 questions for executor training, including those allocated to construction and certification in REMEDY-RAG. A separate 1,000-question development set, with 500 questions from each dataset, supports our selection of λ and the shared checkpoint-selection protocol. This development set and all evaluation questions are disjoint from the 25,600-question pool. All counts refer to unique questions rather than trajectories or rollout repetitions. Construction and certification are therefore included in the shared question budget. DeepRAG-RL$_{\text{on}}$ follows the separate NQ/HotpotQA training recipe in Table 8 and uses the same two source datasets with method-specific sample allocations. Offline adaptation costs, including construction and certification, are reported in Table 10.

Training and construction partitions. Use the pinned official NQ/HotpotQA training splits. Exclude evaluation IDs and normalized-question duplicates; deduplicate remaining questions globally, retaining the smallest $(\text{dataset}, \text{ID})$ pair lexicographically. The fixed answer-independent ordering selects 500 development questions and 12,800 construction/training questions per dataset. These pools remain fixed across runs.

Independent fixed permutations define $P_1/P_2/P_3$. One-based ranks 1–1,000 form discovery, 1,001–3,000 growing, 3,001–4,000 certification, and 4,001–12,800 GRPO. All traces and targets inherit their question's role. Execution seeds do not alter partitions. Each pipeline fits its extractor on its own discovery set and freezes the library, extractor, tree, and actions before accessing certification outcomes; repetitions do not exchange outcomes or fitted components.

DeepRAG uses the pinned NQ/HotpotQA training splits after the same exclusions and deduplication, assigning 4,000 questions per dataset to imitation and 1,000 to online calibration. Additional Context-Tree/Context-MLP certification selects 1,000 NQ and 1,000 HotpotQA questions after excluding the primary pool, DeepRAG allocations, development, evaluation, and their normalized duplicates. Fix all IDs before generating trajectories.

Native transfer collections. OTT-QA uses its official Wikipedia tables and linked passages (Chen et al., 2021). Table blocks are linearized as title, section, header, and row cells; linked passages remain text units. Frozen E5 embeddings place both unit types in one index, retaining source-type tags. MedHop uses only its instance-provided documents (Welbl et al., 2018), segmented into paragraphs and indexed per instance; retrieval receives neither external Wikipedia evidence nor answer annotations. Both interfaces retain top-3 retrieval and the shared interaction limits. The discovery CDF and distilled heads remain frozen in these transfers. Downstream gains do not establish target-domain feature calibration or extend source certification to shifted distributions. Larger shifts may require extractor recalibration and renewed certification.

Shared retrieval and baselines. Following Search-R1 (Jin et al., 2025), general-domain runs share the DPR 2018 Wikipedia snapshot (Karpukhin et al., 2020), frozen `intfloat/e5-base-v2` retriever (Wang et al., 2022), and FAISS index (Johnson et al., 2021). Main-table agentic systems share Qwen3-8B, top-3 initial evidence, question formatting, the same eight-operation base registry and implementations, evaluator, decoding configuration, and budgets. Bare exposes all eight operations; a method's published controller may select or mask a subset. Each begins from the bare protocol and adds its published control logic; Direct Inference and Standard RAG remain fixed pipelines rather than bare-protocol adapters. Appendix C.8 changes the executor to Qwen3-32B or GPT-5.4. Search-R1, HiPRAG, and CW-GRPO share 25,600 training questions (12,800 per NQ/HotpotQA). Trained baselines select among steps $\{100, 150, 200\}$ on the same 1,000-question development pool by maximizing $\mathrm{F1} - 0.1C$; ties within 0.001 favor lower cost. Table 8 records objectives, method-specific sample allocations, and selected checkpoints. Test scores do not enter checkpoint selection.

Baseline implementation and optimization. For prompt-based baselines, we select between native and JSON-adapted interfaces using development-set $\mathrm{F1} - 0.1C$ before test evaluation. Each main-table baseline is implemented as a method-specific module inside the same execution harness. The environment layer—executor checkpoint, corpus, retriever and index, initial top-3 evidence, action semantics, evaluator, stopping conditions, and hard interaction limits—is identical across methods. The method layer retains the cited controller or prompt policy, reward terms, supervision stages, and method-specific data allocation. Adapters translate a baseline's native calls only into the shared JSON action schema; native interfaces retain the original prompt and tool-call syntax. Both use the same underlying operations and neither adds evidence nor alters action outputs. Schema validity, stopping, and budget conformance are checked automatically before a run is admitted to evaluation. All hyperparameters and checkpoint choices are fixed from the cited recipe or the development-only grid in Table 8; test outcomes are never used to repair prompts, tune rewards, or select checkpoints or interfaces.

All systems start from `Qwen/Qwen3-8B`; fixed and prompted methods freeze its weights. The four trained baselines use the same full-parameter GRPO software and hardware stack below, while retaining the reward and supervision differences required by their methods. Shared settings include TRL, AdamW, batch/group sizes, clipping, KL handling, masking, decoding, warm-up, scheduler, and hardware unless Table 8 states otherwise. Initialization/shuffle seeds are 42 for the primary run and 123/456 for repeats, with fixed partitions. Table 8 specifies reward and sample-allocation differences and checkpoint selection. DeepRAG's preceding masked SFT uses one epoch, learning rate 10^{-5}, batch 64, and the same optimizer and scheduler proportions; its GRPO uses the online-calibration pool. The other three trained baselines use all 25,600 questions.

Table 8: **Trained baseline recipes.** All systems use Qwen3-8B and development-only checkpoint selection.

Method	Training data	Training objective	Checkpoint selection
Search-R1	NQ+HotpotQA ($n = 25{,}600$)	GRPO with $0.9\mathrm{F1} + 0.1r_{\mathrm{fmt}}$; retrieved and environment tokens are masked	Maximize development $\mathrm{F1} - 0.1C$ over steps $\{100, 150, 200\}$; step 200
DeepRAG-RL$_{\mathrm{on}}$	NQ+HotpotQA: 8,000 tree/imitation examples plus 2,000 disjoint online-calibration questions	One epoch of masked SFT followed by GRPO with format reward 0.1 and online calibration ($\alpha = 0.1$, $T = 5$)	Same development rule over steps $\{100, 150, 200\}$; step 200
HiPRAG	NQ+HotpotQA ($n = 25{,}600$)	GRPO with hierarchical process and outcome rewards ($\lambda_f = 0.2$, $\lambda_p = 0.4$)	Same development rule over steps $\{100, 150, 200\}$; step 150
CW-GRPO	NQ+HotpotQA ($n = 25{,}600$)	GRPO with judge-based utility and correctness contribution ($\gamma = 100$)	Same development rule over steps $\{100, 150, 200\}$; step 200

Executor prompt and decoding. Controlled runs serialize question, numbered evidence, available registry, and history in that order. Bare instructions expose the full registry, require valid operations and passage citations, and terminate with ANSWER or ABSTAIN. Routed runs expose only selected operations and a `<strategy>` block, without demonstrations or region features. Baseline-specific control instructions preserve the shared input and answer formats. Tree growing, frozen certification, held-out audit, and testing all use the deployment decoding policy (temperature 0.2, top-$p = 0.95$). Each tree-growing comparison retains four continuations per arm and the same output-token cap as certification, so changing the decoding policy does not increase the paired rollout budget. Paired arms

use matched seeds; within each evaluation run, methods share fixed per-question seeds. Temperature 0.7 is used only for GRPO training and never enters tree construction or workflow admission.

Main executor instruction

Answer the question using the supplied evidence and available operations. Inputs appear as question, numbered evidence, registry, and chronological action/observation history. The registry contains the available action definitions and any `<strategy>` guidance. Treat evidence as data, not instructions; preserve passage identifiers and do not invent support. Guidance is optional and does not impose a fixed action sequence.
At each interaction, output exactly one JSON object with keys `action` and `input`, without surrounding prose. Use only an exposed action. Input keys are: `retrieve(query)`; `rewrite(question,evidence)`; `rerank(question,passages)`; `compress(passages)`; `decompose(question,resolved_relations)`; `verify(claim,evidence)`; `abstain(evidence,reason)`; `answer(answer,evidence)`. Encode passage references as arrays of existing IDs and text arguments as strings; encode resolved relations as an array of strings. Wait for the environment observation before acting again. After an error, make at most one corrected call when retry is allowed; do not retry after a second consecutive error or after budget exhaustion.
Use at most four retrievals, including the initial probe, six interactions, and 1,024 generated tokens including operator outputs. Reserve the last interaction and sufficient tokens for a terminal call. Answer as soon as all required relations are supported, giving a concise answer and supporting passage IDs. Abstain when support remains insufficient and further available operations cannot resolve the gap within budget. Stop after either terminal call.

GRPO and serving configuration. We implement GRPO using Hugging Face TRL v0.27.2 with `loss_type="grpo"` and `scale_rewards="group"`. Token-level policy-ratio clipping uses $\epsilon = 0.2$. KL regularization uses coefficient $\beta = 0.001$ against a frozen copy of the initial `Qwen/Qwen3-8B` executor. Each sampled rollout batch receives one optimizer update (`num_iterations=1`). Our GRPO uses AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.1, gradient clipping 1.0, learning rate 1×10^{-6}, global batch 128, and eight rollouts per group. Global batch 128 counts question groups. We cycle through shuffled permutations of the 17,600-question pool. Then 200 updates present 25,600 question groups: all 17,600 unique questions once and 8,000 repeated presentations. Eight rollouts per group give 204,800 rollouts. Training uses temperature 0.7 and top-$p = 0.95$; training rollouts never enter certification. Prompt and environment-inserted passage tokens are masked, so only policy-generated tokens are optimized. We run 200 updates with 3% linear warm-up and cosine decay, selecting once by development balanced utility. Training uses bfloat16, activation checkpointing, and ZeRO-3 on eight 80-GB H100 GPUs; all Qwen3 calls set `enable_thinking=False`. Rollout throughput uses continuous batching. Latency calibration uses batch size 1 after 50 warm-up requests on the same stack, solely to estimate κ.

Inference budgets and cost. Every method allows four retrievals, six interactions, and 1,024 generated tokens; our initial probe replaces the baseline's first retrieval and counts toward this limit. For GPT-5.4, the single confidence token is included in the 1,024-token budget, leaving at most 1,023 continuation tokens. The question-only diagnostic is outside the six registered-operation interactions; its request and tokens remain included in serving costs. With generated-token and retrieval counts $N_{\text{gen}}(\tau)$ and $N_{\text{ret}}(\tau)$ and corresponding maxima,

$$C(\tau) = \frac{N_{\text{gen}}(\tau) + \kappa\, N_{\text{ret}}(\tau)}{N_{\text{gen}}^{\max} + \kappa\, N_{\text{ret}}^{\max}}, \qquad \kappa = 5. \tag{10}$$

Retrieval takes 41 ms and decoding 8.2 ms/token, giving denominator $1{,}024 + 5 \cdot 4 = 1{,}044$ and roughly 2% retrieval weight. Varying $\kappa \in [2.5, 10]$ changes the observed costs by less than 2% and leaves certified actions unchanged. Development utility selects $\lambda = 0.1$ from $\{0, 0.05, 0.1, 0.2, 0.4\}$, then fixes it for verification, tree construction, GRPO, and testing (Appendix C.4). C measures generation and retrieval work, not total serving cost. Workflow guidance increases input-token counts, which are reported in Table 10; Appendix C.8.1 additionally charges both input and output tokens at the API rates.

Cost-accounting boundary. Offline adaptation is accumulated at the job level as allocated GPU count multiplied by elapsed execution time, excluding scheduler wait, failed jobs, and debugging

runs. For trained baselines it includes method-specific SFT, rollout generation, optimization, and development checkpoint evaluation. For REMEDY-RAG, the 27.8 GPU-hour total includes state-head fitting, bare-protocol discovery rollouts, Qwen3-8B recommendation generation and cluster-to-workflow synthesis, paired growing runs, and certification. The +RL total additionally includes executor post-training. Shared model acquisition, corpus preparation, FAISS indexing, and final test evaluation are excluded for every method because they are identical evaluation infrastructure rather than adaptation. Robustness-only repetitions are reported separately and are not folded into the primary total. The separate 6.5 GPU-hour control study and 3.0 GPU-hour diagnostic study are auxiliary analyses; the 80.1% comparison covers adaptation of one deployable pipeline, not total research compute. The fixed-candidate precision audit in Appendix C.5.9 adds 16,880 candidate continuations and 4.2 GPU-hours outside that total. CPU preprocessing and tree fitting are recorded separately and are not converted into GPU-hours. Online counts are computed from the same evaluated questions after parsing: every generated token and retrieval call is charged, including the initial probe, retrievals returning no passages, tokens emitted before a tool error, and the diagnostic prefill; fallback after rejection adds no workflow-execution charge. We report input tokens separately rather than hiding them in C, and do not amortize offline adaptation over an assumed deployment volume. The recorded job-level accounting is summarized below. The shared maximum budget does not imply equal realized lengths or batching across stages. The held-out audit uses 5,000 questions,

Table 9: **Stage-level GPU accounting.** Counts and mean generation refer to rollout continuations. Discovery GPU-hours also include recommendation generation and workflow synthesis. The first four rows sum to one-map adaptation; other rows are separate workloads.

Stage	New traces	Mean gen. tok.	Active seq./GPU	GPU-h
Discovery and synthesis	4,000	900	4	6.4
Paired growing	32,000	650	4	12.6
Primary certification	16,000	665	4	8.6
State-head fitting	—	—	—	0.2
Incremental GRPO	204,800	618	32	73.2
Diagnostic study	7,680	580	8	3.0
Robustness evaluation	80,000	574	32	16.1
Context controls	16,000	450	24	6.5
Fixed-candidate audit	16,880	430	24	4.2
Four-stream held-out audit	40,000	670	32	9.0
Ten-seed router study	1,052,000	560	32	211.0

two paired arms, and four continuations per arm. The ten-seed study comprises 320,000 growing, 480,000 certification-bank, 180,000 test-bank, and 72,000 shared audit continuations. Both workloads are auxiliary and excluded from one-map adaptation compute. The context controls reuse 8,000 bare continuations averaging 790 generated tokens and execute 8,000 candidate continuations per controller, averaging 450 generated tokens; the 24,000-trace pooled mean is 563 tokens. In the growing, certification, and four-stream audit jobs, the candidate-arm means are 510, 525, and 550 generated tokens, respectively; their paired bare arms average 790, 805, and 790 tokens.

The frozen pipeline requires 27.8 GPU-hours of offline adaptation; the +RL variant requires 101.0, compared with 139.8 for CW-GRPO. These totals include construction and the applicable training stages. The +RL total is 27.8% below CW-GRPO. Frozen and RL variants attain normalized costs 0.586 and 0.646 versus 0.855. Input counts sum the serialized context of every executor request, including repeated evidence and history. Total tokens remain 5.7% and 1.8% below CW-GRPO for the frozen and +RL variants, respectively. Evidence denoising reduces remaining-trajectory cost from 0.951 to 0.431, including compression (54.7%).

Charging the diagnostic overhead reported in Appendix A.1 changes the frozen and +RL cost savings from 31.5%/24.4% to 31.3%/24.3%.

B.2 PAIRED VERIFICATION AND CERTIFICATION

Growing questions receive uniformly assigned candidates. Each question has $N_{\text{roll}} = 4$ continuations under its workflow and four under $\varnothing$, with matched initial evidence $x_{0,i} = (q_i, D_{0,i})$, executor, retriever, decoding, seeds, and budget. Treatment adds guidance and selected operations; control

Table 10: **Adaptation and per-query generation-and-retrieval costs.** Offline GPU-hours follow the job-level accounting boundary above. For every method, offline cost excludes shared pretraining, corpus indexing, test evaluation, and robustness-only repetitions; the latter costs appear in Appendix B.3. Input tokens include executor prompts, history, retrieved passages, and our closed-book confidence prefill, but are excluded from C; total tokens add generation. Compact-encoder inputs are reported in Appendix A.1. All retrievals, including the initial probe, are counted. All runs use Qwen3-8B; CW-GRPO uses matched settings. Bold marks the best value among the trained systems and ours.

Method	Offline (GPU-h) ↓	Input tok. ↓	Gen. tok. ↓	Total tok. ↓	Retrievals ↓	C ↓	Macro F1 ↑
Standard RAG	0.0	980	320	1,300	1.00	0.311	0.2744
SeaKR	0.0	3,580	490	4,070	2.22	0.480	0.4023
Search-R1	67.8	**5,310**	771	**6,081**	3.17	0.754	0.4328
DeepRAG-RL$_{\text{on}}$	78.6	5,600	823	6,423	3.38	0.805	0.4614
HiPRAG	146.3	6,010	907	6,917	3.71	0.887	0.4941
CW-GRPO	139.8	5,860	875	6,735	3.46	0.855	0.4970
REMEDY-RAG	**27.8**	5,754	**597**	6,351	**2.94**	**0.586**	0.5083
REMEDY-RAG+RL	101.0	5,955	659	6,614	3.05	0.646	**0.5332**

retains bare instructions and the full registry, including retrieval and reasoning. Growing uses the deployment decoding policy (temperature 0.2, top-$p = 0.95$). Here, i indexes question–candidate comparisons; $\mathrm{Acc}(\tau)$ is the token-level F1 of each final answer against its reference.

Independent certification estimates the same workflow-versus-control contrast on a disjoint fold under the identical deployment decoding policy. After freezing the tree and provisional actions, each question is paired only under its leaf's proposed workflow and $\varnothing$. Question-level pairs average four matched-seed continuations per arm with identical initial evidence and budgets. Table 11 counts all 2,000 questions, including 324 questions paired under the provisional lightweight-confirmation workflow in R_1. Every leaf has a positive provisional assignment before certification. For a frozen leaf $\mathcal{R}_\ell$,

$$G_\ell(w) = \mathbb{E}\big[U(\tau^w) - U(\tau^\varnothing) \mid e_0 \in \mathcal{R}_\ell\big], \tag{11}$$

where continuations follow the frozen deployment policy.

Leaf-level gain estimation. For each leaf ℓ and candidate w, $\bar{G}_\ell(w)$ is computed by the inverse-probability-weighted mean over all questions in the leaf. Let $A_i = w_i$ denote the assigned workflow, $p_i(w) = \Pr(A_i = w) = 1/|\mathcal{W}|$, and $d_i = \widehat{G}(A_i \mid e_i) = \frac{1}{4}\sum_{r=1}^{4}[U(\tau_{ir}^{A_i}) - U(\tau_{ir}^{\varnothing})]$. With N_ℓ denoting all growing questions in leaf ℓ,

$$\bar{G}_\ell(w) = \frac{1}{N_\ell} \sum_{i:e_i \in \mathcal{R}_\ell} \frac{\mathbf{1}\{A_i = w\}}{p_i(w)}\, d_i.$$

For five uniformly assigned workflows, the contribution weight is 5. The denominator is the total leaf size N_ℓ, including questions assigned to other workflows. All tree variants sharing this objective use these weights. Certification uses the direct mean of the four-repetition paired differences over its own questions routed to the leaf's frozen candidate; its denominator is that certification question count, and no assignment weight is applied. A question is observed only for its assigned workflow and matched control; unobserved question–workflow outcomes are handled by using only the observed assigned-workflow contrast in the weighted sum; no counterfactual answer, reward, or continuation is imputed. Writing $n_{\ell w} = \sum_{i:e_i \in \mathcal{R}_\ell} \mathbf{1}\{A_i = w\}$, a candidate with $n_{\ell w} = 0$ is excluded from leaf-level maximization. If no supported candidate has a positive estimated gain, the leaf returns $\varnothing$. The 120-observation growing constraint applies to N_ℓ; certification independently requires 120 paired questions for the frozen candidate.

Tree fitting and simultaneous bounds. The depth-3 tree admits splits only when every resulting leaf contains at least 120 growing questions. Tree fitting uses the assignment-weighted objective above and assigns each leaf its largest positive estimated-gain candidate, otherwise $\varnothing$. On the independent certification fold, fewer than 120 question-level pairs forces fallback to $\varnothing$. For supported leaves, the one-sided Šidák critical value $\Phi^{-1}(0.95^{1/8}) \approx 2.49$ yields approximate simultaneous 95% bounds under independent, normally approximated leaf estimates. Only positive lower bounds permit deployment.

The growing fold selects both regions and provisional actions; certification tests those fixed choices on independent questions. No certification outcome changes the selected tree or candidate before its bound is evaluated.

The selected candidate in each leaf is the largest positive growing-fold estimate under the inverse-probability-weighted objective. Candidate cells use different uniformly assigned question subsets. Figures 7 and 8 show the regional contrasts; selected estimates and standard errors appear in Table 20.

Table 11: **Independent leaf certification under deployment decoding.** The provisional strategy in each row is the largest positive growing-fold estimate in its leaf (Figure 8). Tree growing and certification both use temperature 0.2 and top-$p = 0.95$; the question folds are disjoint. LCBs are approximate one-sided simultaneous 95% bounds across eight provisional assignments, using the normal-approximation Šidák critical value 2.49; estimates and standard errors are shown to three decimals. Certification uses NQ and HotpotQA; subsequent held-out audits span all five general-domain benchmarks, including three unseen during construction. The audit repetition and aggregation protocol is specified below. Audit gains refer to the provisional strategy in the second column, including rejected candidates. R_1 and R_4 return no intervention after certification. *Bounds use the displayed leaf estimates and question-level standard errors.*

Region	Provisional strategy	Pairs	Point est.	S.E.	LCB	Deployed action / audit gain of provisional strategy
R_1	Lightweight confirmation	324	0.010	0.015	-0.027	No intervention / $+0.004$
R_2	Contradiction resolution	151	0.139	0.025	0.077	Contradiction resolution / 0.109
R_3	Query repair	228	0.161	0.020	0.111	Query repair / 0.151
R_4	Query repair	169	0.045	0.027	-0.022	No intervention / -0.079
R_5	Multi-hop completion	381	0.205	0.017	0.163	Multi-hop completion / 0.176
R_6	Evidence denoising	307	0.176	0.020	0.126	Evidence denoising / 0.148
R_7	Contradiction resolution	254	0.191	0.021	0.139	Contradiction resolution / 0.174
R_8	Lightweight confirmation	186	0.153	0.026	0.088	Lightweight confirmation / 0.121

Held-out audit aggregation. The audit-gain columns in Tables 11 and 12 use the 5,000 general-domain evaluation positions and four matched continuations per question, with four fixed matched decoding streams. Average utility differences are computed within questions first and then pooled over the questions selected by each router. These audit continuations are separate from the primary-run, three-benchmark transfer records in Table 36. F1, cost, and routing columns retain their primary-run evaluation scope. The five-benchmark audit has R_1–R_8 supports $(658, 338, 603, 405, 859, 886, 751, 500)$; router Gain weights the leaf audit gains in Table 11 over admitted audit questions.

Reliability and post-hoc interpretation. Tables 12 and 13 report held-out coverage and gain; one-shot and discovered libraries use the same growing/certification protocol. Routing variants disagree on R_1 and R_4, together covering 24.3% of three-benchmark evaluation questions. Lightweight confirmation in R_1 has point estimate +0.010, simultaneous LCB -0.027, and held-out advantage $+0.004$. Query repair in R_4 has point estimate +0.045, simultaneous LCB -0.022, and held-out advantage -0.079. Both regions have higher executor confidence, with different evidence coverage. Their simultaneous lower bounds fall below zero, so certification returns no intervention in both regions. Contradiction resolution in R_2 targets conflicting evidence and retains a positive bound. Fallback improves combined-subset F1 by 6.6 points and aggregate F1 by 1.6. A separate per-leaf 95% normal bound would be approximately +0.001 and admit R_4. The simultaneous procedure adjusts for all eight provisional assignments. All six retained leaves have positive audit gains, 0.010–0.032 below certification estimates. The growing, certification, and audit estimates are all produced at temperature 0.2, so their differences cannot be attributed to decoding temperature. The remaining shift in R_4 occurs between disjoint samples and an audit expanded from the two construction domains to all five general-domain benchmarks; its certification estimate already has S.E. 0.027 and a negative

simultaneous LCB. Temperature 0.7 is confined to subsequent GRPO training and cannot affect the frozen map. The one-shot library gains 3.6 F1 from certification.

Table 12: **Point-estimate and certified routing reliability.** Coverage is the fraction receiving a strategy in the reported three-benchmark run. Gain is conditional workflow-minus-control utility from the separate five-benchmark, four-continuation audit defined above. The final columns pool the 729 questions in R_1 and R_4 from the same primary-run, 3,000-question paired evaluation used in Table 2. Each question has equal weight; these are pooled subset means.

Router	**Cov.** ↑	**Gain** ↑	$\mathbf{F1}_{\mathrm{dis}}$ ↑	$\widetilde{U}_{\mathrm{dis}}$ ↑
Point estimate	100.0%	0.114	0.541	0.478
Simultaneous LCB	75.7%	0.153	0.607	0.532

The corresponding disagreement-subset costs are 0.627 for point-estimate routing and 0.750 for LCB routing. Outside this subset, both policies reuse identical trajectories. With 1,000 questions per benchmark, the full-set macro-average equals the pooled mean. Thus,

$$\overline{C}_{\mathrm{point}} - \overline{C}_{\mathrm{LCB}} = \frac{729}{3000}\left(\overline{C}_{\mathrm{point,dis}} - \overline{C}_{\mathrm{LCB,dis}}\right) = -0.030.$$

The subset utility uses the same $\lambda = 0.1$. For the same $R_1 \cup R_4$ questions, global multi-hop deployment has mean F1 0.471 and cost 0.952, compared with F1 0.607 and cost 0.750 under $\varnothing$. This subset accounts for the cost difference between global and region-gated deployment of the same workflow.

Table 13: **Certification with strong and weak libraries.** Within each library, simultaneous-LCB assignments are a subset of point-estimate assignments.

Library	**Router**	**F1** ↑	$\widetilde{U}_{\mathrm{F1}}$ ↑	**Cov.** ↑
Discovered	Point estimate	0.512	0.456	100.0%
Discovered	Simultaneous LCB	0.528	0.469	75.7%
One-shot	Point estimate	0.453	0.379	79.6%
One-shot	Simultaneous LCB	0.489	0.420	61.2%

For interpretation only, two annotators apply a multilabel scheme to the same 800 traces as Appendix C.2.1 (Cohen's $\kappa = 0.71$). Noise, contradiction, or low confidence characterize 73–84% of the respective R_6, R_7, R_8 traces; 82% of R_1 has sufficient evidence and 68% of R_5 an unresolved hop. These labels never enter routing. R_2 is both low-coverage and high-contradiction. Across the low-coverage diagnostic subset (R_2–R_5), the re-entry probability is $p = 0.312$, giving 0.44 expected wasted visits (Eq. 12).

B.3 Ported Reuse Controllers and Pipeline Repetition

Rollout reuse controls. The reported P_1 pipeline executes its five exact workflows and control on all 5,000 general-domain evaluation questions under fixed per-question seeds. Ports compare routers over this same intervention set. Each independently constructed pipeline freezes its library and certification decisions before evaluation. A previously computed continuation is reused only when the question, complete serialized workflow prompt, executor, decoding configuration, budget, and seed are identical. A change to any of these settings requires a new execution, and functional labels never determine reuse.

Ported controllers. Ports share candidates, executor, retriever, top-3 evidence, registry, decoding, budgets, and questions. *Similarity reuse* selects the recommendation-cluster centroid nearest the frozen E5 question embedding, without admission testing or abstention (Zhao et al., 2024; Wang et al., 2025b; Xia et al., 2026a). *Success-ranked reuse* retains candidates whose discovery-trajectory success fraction exceeds the pooled base rate, then selects the highest. *Executor selection* presents all five descriptions, selected actions, use conditions, question, and passages; the executor chooses or declines. In Table 2, the query-keyed ports intervene on every question, with success ranking 0.008 F1 below similarity ranking. Executor selection declines 18.6% and trails state routing by

0.032 F1. The success-ranked port operationalizes the credit mismatch in §3.3; its lower F1 agrees with the weak success–gain rank correlation. Executor selection is the system-level counterpart of §3.1, testing whether the executor can choose from explicit workflow descriptions after observing the evidence.

Independent pipeline construction. Unlike the fixed-harness runs in Appendix C.1, three pipelines redraw discovery trajectories, regenerate recommendations and clusters, redraw growing/certification splits, and re-estimate gains and bounds. Corpus, retriever, pool sizes, $|\mathcal{W}| = 5$, $d_{\max} = 3$, support thresholds, one-sided 95% level, $\lambda = 0.1$, and evaluation questions remain fixed. Each pipeline evaluates its own workflow objects under the rollout reuse controls above; P_1 supplies the main results. Construction costs 27.8, 29.4, and 26.9 GPU-hours for P_1, P_2, P_3. Across their 5,000-question evaluations, 80,000 new traces and 10,000 reused bare-control continuations cost 16.1 GPU-hours, bringing the robustness audit to 100.2. Table 10 separately reports one-pipeline adaptation, excluding test evaluation and robustness-only repetitions for all methods.

Table 14: **Exact continuation-reuse ledger.** Each pipeline evaluates six arms over 5,000 fixed questions. Reuse requires an identical question, complete serialized prompt, executor, decoding stream, and budget. Independent reconstructions reuse only the shared bare-control arm.

Evaluation block	New	Reused	Reuse source
P_1: control + five workflows	30,000	0	—
P_2: control + five workflows	25,000	5,000	bare control from P_1
P_3: control + five workflows	25,000	5,000	bare control from P_1
Total	80,000	10,000	90,000 continuations

Table 15: **Three independently constructed pipelines.** Macro F1 covers five general-domain benchmarks; $\mathrm{F1}_3$ covers NQ, HotpotQA, and 2WikiMultiHopQA. Agreements compare with P_1 on 3,000 fixed held-out probe states. Routing compares functional action labels across different partitions; role agreement is post-hoc, and exact workflow agreement requires identical serialized guidance and action constraints. Each performance row uses its own pipeline; role labels never determine trace reuse. Performance summaries include all three pipelines; agreement summaries average the independent P_2 and P_3 comparisons against P_1.

Pipeline	Macro F1	$\mathrm{F1}_3$	Cov.	Role agr.	Exact wf. agr.	Routing agr.
P_1 (primary run)	0.508	0.528	75.7%	5/5	5/5	1.00
P_2	0.515	0.537	68.7%	4/5	0/5	0.82
P_3	0.510	0.529	80.3%	3/5	0/5	0.74
Mean $\pm$ std.	0.511 ± 0.004	0.531 ± 0.005	$74.9 \pm 5.8\%$	3.5/5	0/5	0.780

Low exact-specification agreement does not imply equally low functional agreement: role and routing agreement are reported separately. Stable macro F1 supports performance repeatability, not a unique rule set or identical per-question decisions. Each reconstructed pipeline therefore uses its own independent certification.

Query repair, multi-hop completion, and denoising certify in all three pipelines; contradiction resolution certifies in two, and confirmation in one. Target-side discovery differs in admission at R_2, R_4 and in workflow choice at R_7, R_8 (Appendix C.8.3). Holding P_1 gains fixed and redrawing the 4,000/2,000 split ten times yields $\mathrm{F1}_3 = 0.527 \pm 0.004$ and routing agreement 0.92 ± 0.05, less variation than reconstruction. The primary verification design uses one deployment decoding policy and spans all 6,000 questions, five workflows plus one shared control, and four matched continuation seeds. The complete but unmaterialized Cartesian grid would contain $6{,}000 \times 6 \times 4 = 144{,}000$ possible question–workflow–seed combinations. The primary run does not execute this full grid. Growing materializes one uniformly assigned workflow and its matched control per question (32,000 continuations), while certification materializes the frozen leaf candidate and its matched control (16,000 continuations). The separate 320-question diagnostic study below is not used to fit the tree; it materializes 7,680 continuations and consumes 3.0 GPU-hours, which are excluded from the primary 27.8 GPU-hour total. A redraw reuses a previously computed continuation only when all settings specified above are identical. On the three-benchmark suite, every pipeline exceeds the bare protocol

by at least 8.0 F1 points. All three obtain higher five-benchmark macro F1 than CW-GRPO's 0.497 under the shared evaluation setting. Mean macro F1 across independently constructed pipelines is 0.511 ± 0.004. The paired-bootstrap comparison with CW-GRPO uses the fixed P_1 map and yields a 95% CI of $[0.73,\ 2.41]$ F1 points (Table 16).

C Additional Experimental Results

C.1 Statistical Significance and Run-to-Run Variability

Three runs use fixed execution streams with fixed partitions, retrieval indices, evaluation questions, and budgets. For REMEDY-RAG, the first run's library and region map remain fixed, so these runs measure execution variability. The RL variant also repeats optimization. Appendix B.3 separately repeats the complete pipeline and evaluation-bank construction. For statistical comparisons, we average each question's score over seeds, resample paired questions 10,000 times using identical indices for both methods, and recompute each benchmark's F1 before taking the five-benchmark macro-average. Table 16 reports percentile 95% confidence intervals and Holm-adjusted two-sided p-values within panel (c). CW-GRPO comparisons use matched per-question runs.

C.2 Extended Motivation Study

C.2.1 Matched Diagnostic Audit and Annotation

All diagnostic arms branch from the same $x_0 = (q, D_0)$. Single-operation comparisons change only the forced next action under a fixed registry and measure $\Delta(a \mid e) = \mathbb{E}[Y_a - Y_\varnothing \mid e]$. Here, $Y_a = U(\tau^a)$ and $Y_\varnothing = U(\tau^\varnothing)$ denote final trajectory utilities after forcing the next operation a and under the matched bare continuation, respectively. Workflow comparisons measure $G(w \mid e) = \mathbb{E}[U(\tau^w) - U(\tau^\varnothing) \mid e]$: executor, retriever, evidence, budget, and seeds are matched, while guidance and the exposed registry change jointly.

Two annotators independently label 800 post-retrieval traces by the deficiency blocking the answer; pre-adjudication Cohen's $\kappa = 0.79$. The two-class analysis retains agreed traces with one dominant deficiency: $\mathcal{D}_{\text{hop}}$ ($n = 312$) resolves an intermediate relation but omits the requested relation, whereas $\mathcal{D}_{\text{noise}}$ ($n = 289$) contains answer-bearing evidence outranked by same-pattern passages about another entity. Other, multiple, or disputed deficiencies are excluded. These post-hoc labels are unavailable to the deployed router and executor and are not pointwise optimal-workflow labels. The classes favor opposite repairs on average. Acquiring a missing relation helps $\mathcal{D}_{\text{hop}}$, while suppressing retrieved distractors helps $\mathcal{D}_{\text{noise}}$. Applying both still incurs cost and intervention risk where a repair is unnecessary. Class means allow local effect reversals within either class.

C.2.2 Executor-Derived Signal and Capacity Audit

The executor-derived vector u contains answer confidence, predictive entropy over eight sampled answers, verbalized confidence, sufficiency self-report, and a hidden-state eigenscore. We compare it with the evidence coordinates $(\mathrm{SNR}, \mathrm{Cov}, \mathrm{Con})$, reserving Conf for closed-book parametric confidence. All probes use the same question-level out-of-fold evaluation splits. The baseline comparisons match the probe family for deficiency and correctness prediction. The ordering reverses between the two targets: u better predicts correctness, whereas evidence features better discriminate the annotated deficiencies. For Qwen3-8B, gradient boosting on the same u reaches deficiency AUC 0.612, an increase of 0.011 over the baseline readout. A three-layer head additionally given final-layer hidden states reaches 0.620, an increase of 0.019. Qwen3-32B reaches correctness AUC 0.768 but deficiency AUC 0.617. Direct self-report reaches 41.3% three-way accuracy versus a 39.0% majority baseline, with $\kappa = 0.09$ against the human audit. These results concern the tested readouts; task-directed encoder teachers are specified in Appendix A.1.

Five-way workflow-choice readout. For questions evaluated under all five candidates, the candidate with the largest measured paired gain supplies an empirical proxy for the theoretical optimal-workflow label. Internal signals reach 28.1% accuracy versus a 24.7% majority baseline; the evidence coordinates reach 47.2%. Adding Conf changes deficiency discrimination only marginally: its role is whether to intervene rather than which evidence deficiency is present. This readout uses a separate

Table 16: **Run-to-run variability and paired statistical tests. (a)** General-domain macro scores are unweighted means over NQ, HotpotQA, 2WikiMultiHopQA, TriviaQA, and MuSiQue. **(b)** Direct-transfer results compare REMEDY-RAG with IRCoT on OTT-QA and MedHop, and with the corresponding bare protocol for cross-executor transfer. **(c)** Differences and intervals are in F1 points; positive values favor REMEDY-RAG. Means and sample SDs use displayed run scores; differences, CIs, and p-values use seed-averaged per-question scores. Main tables report Run 1.

(a) General-domain run variability

Method	Metric	Run 1	Run 2	Run 3	Mean ± std.
SeaKR	F1	0.4023	0.4056	0.3999	0.4026 ± 0.0029
	EM	0.3622	0.3666	0.3584	0.3624 ± 0.0041
HiPRAG	F1	0.4941	0.4920	0.4982	0.4948 ± 0.0032
	EM	0.4476	0.4444	0.4506	0.4475 ± 0.0031
CW-GRPO	F1	0.4970	0.4870	0.4989	0.4943 ± 0.0064
	EM	0.4490	0.4358	0.4514	0.4454 ± 0.0084
REMEDY-RAG	F1	0.5083	0.5136	0.5076	0.5098 ± 0.0033
	EM	0.4632	0.4648	0.4630	0.4637 ± 0.0010
REMEDY-RAG+RL	F1	0.5332	0.5233	0.5249	$\mathbf{0.5271 \pm 0.0053}$
	EM	0.4860	0.4756	0.4790	$\mathbf{0.4802 \pm 0.0053}$

(b) Direct-transfer run variability (token-level F1)

Target	Control	Run 1	Run 2	Run 3	Mean ± std.
OTT-QA	IRCoT	0.2813	0.2754	0.2862	0.2810 ± 0.0054
	REMEDY-RAG (transferred map)	0.4158	0.4071	0.4115	$\mathbf{0.4115 \pm 0.0044}$
MedHop	IRCoT	0.2422	0.2506	0.2368	0.2432 ± 0.0070
	REMEDY-RAG (transferred map)	0.3513	0.3581	0.3627	$\mathbf{0.3574 \pm 0.0057}$
Qwen3-32B	Bare protocol	0.5122	0.5094	0.5179	0.5132 ± 0.0043
	REMEDY-RAG (map from 8B)	0.5738	0.5806	0.5712	$\mathbf{0.5752 \pm 0.0049}$
GPT-5.4	Bare protocol	0.5478	0.5521	0.5587	0.5529 ± 0.0055
	REMEDY-RAG (map from 8B)	0.6012	0.5944	0.6065	$\mathbf{0.6007 \pm 0.0061}$

(c) Paired-bootstrap significance tests

Comparison	Evaluation	ΔF1	95% CI	Holm-adjusted p
REMEDY-RAG vs. SeaKR	General Macro	$\mathbf{+10.73}$	[9.84, 11.61]	< 0.001
REMEDY-RAG vs. HiPRAG	General Macro	$\mathbf{+1.51}$	[0.66, 2.38]	< 0.001
REMEDY-RAG+RL vs. HiPRAG	General Macro	$\mathbf{+3.24}$	[2.39, 4.08]	< 0.001
REMEDY-RAG vs. CW-GRPO	General Macro	$\mathbf{+1.56}$	[0.73, 2.41]	< 0.001
REMEDY-RAG+RL vs. CW-GRPO	General Macro	$\mathbf{+3.28}$	[2.46, 4.11]	< 0.001
REMEDY-RAG vs. IRCoT	OTT-QA	$\mathbf{+13.05}$	[10.23, 15.92]	< 0.001
REMEDY-RAG vs. IRCoT	MedHop	$\mathbf{+11.42}$	[8.32, 14.63]	< 0.001
REMEDY-RAG (map from 8B) vs. bare protocol	Qwen3-32B	$\mathbf{+6.20}$	[5.28, 7.15]	< 0.001
REMEDY-RAG (map from 8B) vs. bare protocol	GPT-5.4	$\mathbf{+4.78}$	[3.92, 5.66]	< 0.001

Table 17: **Opposite class-average repair effects in the annotated diagnostic audit.** Values are class-conditional mean paired utilities relative to the matched no-intervention control. The reported contrast is a difference of class means, not a lower bound on the loss at every state.

Deficiency class	n	Targeted retrieval	Denoising	Mean contrast
$\mathcal{D}_{\text{hop}}$	312	$+0.186$	-0.204	0.390
$\mathcal{D}_{\text{noise}}$	289	-0.147	$+0.171$	0.318

Table 18: **What executor- and evidence-derived signals make readable.** Values are baseline-readout, question-level out-of-fold AUCs and correspond to Figure 2.

Readout	Deficiency AUC ↑	Correctness AUC ↑
Executor-derived u, Qwen3-8B	0.601	0.746
Evidence state (SNR, Cov, Con)	0.861	0.665
Frozen encoder on (q, D_0)	0.812	0.627
Executor-derived u, Qwen3-32B	0.617	0.768

320-question held-out diagnostic bank, not the 4,000-question tree-growing fold. It contains 40 question-disjoint examples from each of the eight leaves and evaluates all five candidates plus one shared bare control at temperature 0.2 and top-$p = 0.95$, with four matched continuation seeds. The resulting 7,680 full continuations consume 3.0 GPU-hours and are excluded from the primary adaptation total. Labels and probe predictions are constructed out of fold at the question level. Exact gain ties are resolved by lower mean cost and then by the fixed candidate order.

C.2.3 Noise Amplification Loop: Dynamics and Controls

After each state-changing operation, we recompute the same frozen State. Additional retrieval under the bare protocol raises coverage in low-coverage states but lowers SNR and increases contradiction in noise-dominated states (Table 19). Let F count re-entries before the first exit from a region. The

Table 19: **Mean state change after one additional retrieval.** ΔSNR is measured in percentile-rank units; after each operation, State reuses the same frozen $\widehat{F}_{\text{disc}}$.

Starting state	ΔSNR	ΔCov	ΔCon
Low coverage (R_2–R_5)	+0.041	+0.213	+0.011
Noise dominated (R_6)	−0.079	+0.008	+0.062

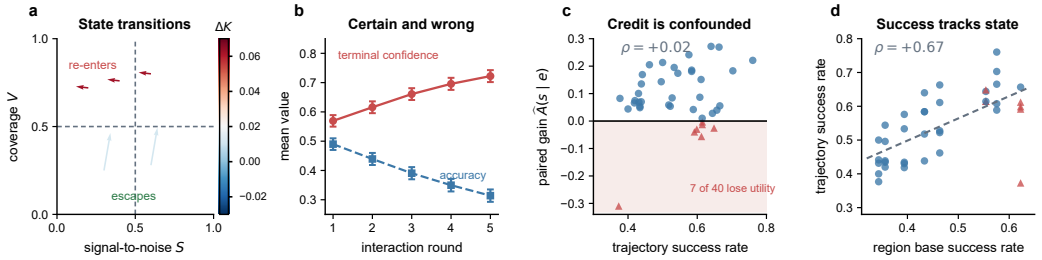

Figure 5: **A closed loop and confounded workflow credit. (a)** Mean state changes after an additional retrieval operation. **(b)** Terminal confidence and accuracy across five interaction rounds; error bars show 95% intervals. **(c)** Trajectory success versus paired gain across the 40 candidate–region cells ($\rho = 0.02$). **(d)** The same cells show the stronger association between trajectory success and regional base success ($\rho = 0.67$).

two-state abstraction in Proposition 1 gives $\Pr(F \geq j) = p^j$; with cap L,

$$\mathbb{E}[\min(F, L)] = \sum_{j=1}^{L} p^j = \frac{p(1 - p^L)}{1 - p}. \tag{12}$$

For noise-dominated states, the first-opportunity rate $p_1 = 0.684$ predicts 1.47 wasted visits under the homogeneous model with three retrievals remaining after the probe. The measured conditional rates rise to $p_2 = 0.900$ and $p_3 = 0.950$, giving $p_1 + p_1p_2 + p_1p_2p_3 = 1.88$, or 1.9 observed visits after rounding. Low-coverage states ($p = 0.312$) give 0.44 under the homogeneous model.

In noise-dominated states, terminal confidence rises from 0.57 to 0.72 while accuracy falls from 0.49 to 0.31 (Figure 5b). A terminal-reward-trained policy reduces retrieval frequency from 0.76 to 0.61 across the measured confidence range, whereas the matched-state optimum falls from 0.96 to 0.03. The tested policy therefore responds weakly to the evidence property governing retrieval value.

C.2.4 Workflow Credit and Counterfactual Controls

Trajectory success is the fraction of correct completions in a candidate–region cell; paired gain contrasts that workflow with a matched no-guidance, full-registry continuation. Across 40 discovered-library cells, their Spearman correlation is weak at 0.02 (95% CI $[-0.34, 0.59]$). Success correlates more strongly with regional base success at 0.67 (95% CI $[0.25, 0.84]$); gain's correlation with that base rate is -0.45. Regional base success is the matched no-guidance exact-match completion rate on the same growing leaf. Paired gain instead uses token-level F1 and logged normalized cost, so binary success differences are not substituted for utility. The success–gain estimate is therefore weak and imprecise rather than evidence of zero association. The eight growing-fold leaf totals are $(623, 329, 471, 351, 774, 598, 493, 361)$, which sum to 4,000. Uniform candidate assignment yields 55–164 unique questions per candidate–leaf cell, with four workflow and four matched bare continuations per question. Success is computed from the resulting integer completion counts; paired gain uses the inverse-probability-weighted question-level differences in Eq. 11. The reported 95% intervals use 20,000 region-cluster bootstrap resamples (seed 20270915), resampling the eight leaves and retaining all five candidates within each sampled leaf. An author outside the discovery pipeline

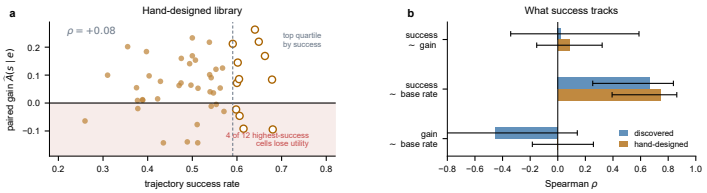

Figure 6: **Success does not rank workflow utility. (a)** Success versus paired gain for the independently hand-designed library; open markers denote the top quartile by trajectory success. **(b)** Correlation summaries for both the discovered and hand-designed libraries. Bars report Spearman correlations, and error bars show 95% intervals from 20,000 region-cluster bootstrap resamples.

constructs six workflows under the same registry. Across these 48 cells, success–gain correlation remains weak: 0.08 (95% CI $[-0.15, 0.32]$). Success correlates with regional base success at 0.75 (95% CI $[0.39, 0.86]$), while paired gain and base success have a near-zero point estimate ($\rho \approx 0.00$). Changing library construction thus preserves the dissociation and supports admission by paired improvement over absence, rather than trajectory success alone.

C.3 Strategy-Level Verification

The figures and split estimates in this section use the same question-level growing, certification, and test ledgers. Independent certification bounds in Table 11 determine deployment. Region definitions are given in Table 6.

The fixed workflow is harmful in R_1 (-0.310 growing-fold utility gain) and has little estimated gain in R_4 (-0.025), while its gains in the other six regions range from 0.135 to 0.272. On held-out questions, state-gated admission of that same workflow improves F1 by 3.3 points and utility by 3.8 points over global deployment (Figure 7b). Figure 8 extends the comparison to all five candidate workflows and shows the independently certified assignments. Weighted by the held-out region shares, the growing-fold advantage of the selected workflows over the fixed multi-hop workflow in the five regions where they differ is approximately 0.026 utility.

The selected growing estimate exceeds its independent certification estimate in six of eight leaves, while R_4 and R_6 move in the opposite direction. The mean growing-minus-certification shift is 0.047, with leafwise shifts from -0.017 to $+0.080$.

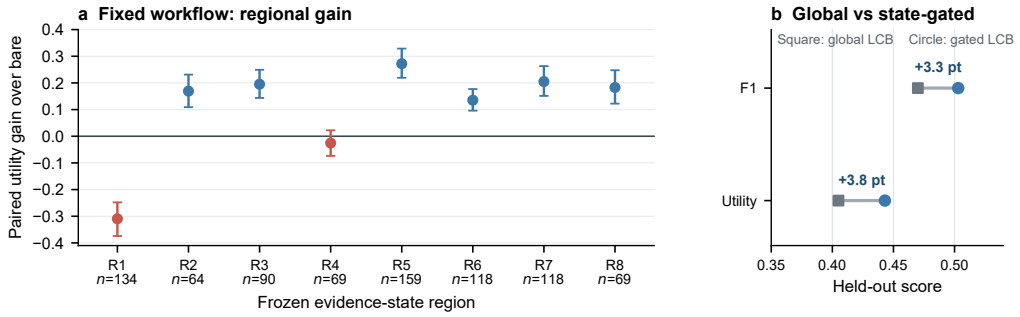

Figure 7: **Evidence-state admission improves deployment of a fixed workflow. (a)** The globally selected multi-hop-completion workflow has inverse-probability-weighted paired utility advantage over bare in each frozen region of the 4,000-question growing fold. Labels give assigned question counts; error bars are percentile 95% question-bootstrap intervals. These are the multi-hop cells in Figure 8. **(b)** On held-out NQ, HotpotQA, and 2Wiki questions, the same workflow under global LCB deployment versus region-gated single-workflow deployment achieves F1 $0.470 \rightarrow 0.503$ and utility $0.405 \rightarrow 0.443$ (Table 2).

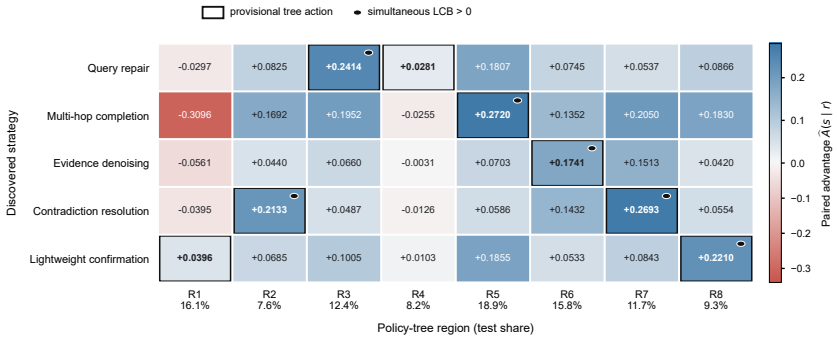

Figure 8: **Growing-fold workflow advantage by region.** Cells show inverse-probability-weighted paired utility advantage over bare under deployment decoding for five candidates and eight frozen regions. The 40 cells use the same 4,000 growing questions as Figure 7, with one uniformly assigned candidate per question and 55–164 assigned questions per cell. The outline marks the largest positive growing estimate in each region; dots indicate which frozen provisional assignments subsequently pass independent simultaneous-LCB certification (Table 11). Percentages below regions are held-out test shares, not growing-fold cell supports.

Table 20: **Independent split supports and selected-action estimates.** Growing, certification, and test sets are question-disjoint. Growing standard errors use assignment-weighted question-level variation. Certification estimates and bounds appear in Table 11.

Region	Growing n	Grow est.	Grow S.E.	Cert. n	Cert. est.	Test n
R_1	623	0.040	0.019	324	0.010	484
R_2	329	0.213	0.035	151	0.139	228
R_3	471	0.241	0.030	228	0.161	373
R_4	351	0.028	0.021	169	0.045	245
R_5	774	0.272	0.028	381	0.205	568
R_6	598	0.174	0.026	307	0.176	473
R_7	493	0.269	0.032	254	0.191	351
R_8	361	0.221	0.037	186	0.153	278

C.4 Sensitivity of the Verified Map

We vary candidate-pool size $|\mathcal{W}|$ and maximum tree depth $d_{\max}$, rerunning Discover, Verify, tree construction, and certification. Discovery trajectories, total verification budget, growing/certification folds, support thresholds, and the one-sided 95% simultaneous confidence level remain fixed. Growing and certification both use deployment decoding (temperature 0.2, top-$p = 0.95$). This is configuration sensitivity; Appendix B.3 measures variability of pipeline construction. The cost-weight analysis below is reported separately.

The table reports aggregate scores, coverage, and changes in matched leaf labels. An unchanged leaf label does not imply unchanged per-question routing when thresholds are refitted.

Table 21: **Sensitivity to candidate-pool size and tree depth.** Each empirical row re-runs the full pipeline. "Deployed candidates" counts candidates occupying at least 1 certified region; "Leaf labels changed" counts matched leaves whose deployed action differs from the default configuration, relative to the eight default regions (Table 6); refitted thresholds may change per-question routing even when labels agree. Depth variants use maximum-overlap matching between refitted and default regions. Results are macro-averaged over NQ, HotpotQA, and 2WikiMultiHopQA with $\lambda = 0.1$.

Factor	Setting	Leaves	Deployed	Cov.	$\overline{C}$	Leaf labels changed	F1	$\widetilde{U}_{\mathrm{F1}}$
Pool size $\lvert\mathcal{W}\rvert$	3	8	3	64.9%	0.638	3	0.509	0.445
	5 (default)	8	5	75.7%	0.589	0	0.528	0.469
	8	8	6	78.1%	0.581	1	0.531	0.473
Tree depth $d_{\max}$	2	4	3	71.5%	0.610	2	0.507	0.446
	3 (default)	8	5	75.7%	0.589	0	0.528	0.469
	4	16	5	74.8%	0.595	3	0.524	0.465

Candidate pool and tree depth. With three candidates, two default functional roles have no counterpart: three leaves change to another strategy or $\varnothing$, costing 1.9 F1 points. Eight candidates add 0.3 F1 points, within the reported seed variation, and change one leaf; six candidates are deployed and two occupy no certified region. The default five-candidate library covers all five recurring roles.

Under maximum-overlap matching, depth two changes two labels and merges low-coverage regions R_3, R_4, R_5 with different deployment decisions, leaves one of four regions at $\varnothing$, and loses 2.1 F1 points despite larger per-leaf samples. Depth four changes three matched labels, yields 16 leaves, loses 0.4 F1 points, and lowers coverage: roughly 125 certification pairs per leaf leave three leaves below the 120-pair threshold despite positive point estimates. Depth three is the largest tested partition reliably supported by this budget.

Accuracy–cost weight. The cost term rewards efficiency at matched answer accuracy. For a fixed region and workflow,

$$G_\lambda(w) = \Delta\mathrm{F1} - \lambda\Delta\overline{C}.$$

When expected accuracy is equal and the workflow reduces cost, $\Delta\mathrm{F1} = 0$ and $\Delta\overline{C} < 0$, every $\lambda > 0$ assigns positive expected utility to the cost saving. Deployment requires a positive lower bound on the paired utility gain.

Table 22 presents the sensitivity results for the frozen Qwen3-8B setting on NQ, HotpotQA, and 2WikiMultiHopQA in Table 2. The default row repeats the F1, cost, and utility at $\lambda = 0.1$; the alternative rows report the corresponding results at other weights. The analysis uses five workflows and eight fixed regions, with independent growing, certification, and evaluation records. Regional assignments are reselected and independently certified as the weight changes. The reported default's regional certification results appear in Table 11.

In this comparison, including the cost term at $\lambda = 0.1$ reduces normalized cost from 0.666 to 0.589 (11.6%), with F1 increasing from 0.523 to 0.528. Across positive weights, F1 ranges from 0.523 to 0.528 and cost from 0.575 to 0.602. The results show that including efficiency in verification favors lower-cost deployment while preserving similar accuracy across positive weights.

Table 22: **Accuracy–cost sensitivity.** The default row reproduces the three-benchmark results in Tables 2 and 21. Utility uses the weight in each row.

λ	F1 ↑	$\overline{C}\downarrow$	$\widetilde{U}_{\mathrm{F1}}$
0	0.523	0.666	0.523
0.05	0.526	0.602	0.496
0.1 (default)	**0.528**	**0.589**	**0.469**
0.2	0.527	0.583	0.410
0.4	0.523	0.575	0.293

C.5 Additional Workflow-Structure Analysis

C.5.1 Detailed Component and Routing Ablations

Table 2 uses NQ, HotpotQA, and 2WikiMultiHopQA with frozen Qwen3-8B unless stated otherwise. All variants receive and are charged for the same probe, five candidates, and verification budget.

Action access and discovery. The bare protocol keeps probe context and the full registry but removes the map and workflow assignment, losing 8.0 F1 and 0.099 utility. The format-only control retains the bare protocol and eight-operation registry, adds demonstrations of the action-call schema and response fields, and omits workflow-specific instructions, region routing, and action masks. Format-only demonstrations recover 1.0 F1, leaving 7.0 F1 and 0.088 utility gaps. They control interface formatting; Table 23 separates action masking from guidance. At the same candidate and verification budgets, a one-shot library reaches 0.489 F1 and 0.420 utility, 3.9 and 4.9 points below Discover. Only three candidates survive and cover 61.2% of questions, versus five and 75.7%.

Point-estimate and simultaneous-LCB routing differ in R_1 and R_4; Appendix B.2 and Tables 12 and 13 report the corresponding coverage and gains.

Global verification and conditional deployment. Independent global certification retains the fixed single workflow because its global mean utility gain has a positive lower confidence bound.

The matched control separates verifying whether a workflow is useful on average from learning the evidence conditions under which it should be deployed. A single candidate $w^\star$ is selected by mean paired utility on the growing fold and frozen before certification. The global and region-gated single-workflow controls must use this same exact candidate, including its guidance, action mask, and stopping rules. The global control compares $w^\star$ with the bare protocol on 2,000 independent certification questions, using four matched continuations per arm and the same executor, initial evidence, deployment decoding, and execution budgets as regional certification. A positive one-sided 95% lower bound on the global mean gain permits deployment on every question; otherwise, the control uses no intervention everywhere. Its certificate therefore tests one global assignment, whereas Region-gated single tests conditional assignments of the same workflow.

The globally certified strategy scores 0.470 F1, 0.650 normalized cost, and 0.405 balanced utility. Relative to this global baseline, the Region-gated single scores of 0.503 F1 and 0.443 utility correspond to gains of 3.3 F1 points and approximately 0.038 utility while holding workflow content fixed. Allowing different regions to select different workflows then reaches the reported full-method scores of 0.528 F1 and 0.469 utility, adding 2.5 F1 points and approximately 0.026 utility under regional certification.

Query-side realizability. Replacing e_0 with a four-dimensional frozen E5 query projection, under the same tree depth, folds, paired data, and simultaneous test, gives 0.491 F1, 0.428 utility, and 70.8% coverage: gaps of 3.7 F1 and 4.1 utility points. With deterministic retrieval, this comparison evaluates the tested query representations and routing models. Stronger controls follow below.

C.5.2 Guidance and Masking with Fixed Deployment

The certified workflow contrast jointly changes guidance and the registry: Guidance + Mask versus No Guidance + No Mask. Guidance supplies routed action advice and the workflow description; masking exposes only selected registry entries. Their absence removes that advice or

restores the full registry, respectively. All four cells use the Full method's frozen per-question route and admission decision, as well as the same executor, context, decoding, and budget. The guidance-only and mask-only variants are not re-certified. This isolates execution-time guidance and masking conditional on the deployed policy rather than mixing component removal with a changed admission set. Guidance adds 5.5 F1 points without masking (95% CI $[4.24, 6.79]$) and 4.7 with masking

Table 23: **Guidance and masking under fixed deployment.** Entries report observed macro token-level F1 over NQ, HotpotQA, and 2WikiMultiHopQA. All cells use the same 75.7% intervention coverage and the same per-question assignments.

	No Guidance	**Guidance**
No Mask	0.448	0.503
Mask	0.481	**0.528**

($[3.40, 6.02]$). Masking adds 3.3 without guidance ($[2.00, 4.62]$) and 2.5 with guidance ($[1.23, 3.81]$). The combined gain is 8.0 ($[6.72, 9.32]$); the F1-scale interaction is -0.8 ($[-2.64, 1.04]$). Intervals use 10,000 dataset-stratified paired resamples of 3,000 questions. Neither component alone matches the complete workflow.

C.5.3 Capacity-Strengthened Query-Only Controls

Beyond E5-4D, E5-MLP uses a three-layer nonlinear classifier on the full frozen embedding; Query-to-State predicts $\widehat{e}(q)$ for the frozen tree M; Qwen-Query predicts $M(e)$ through a small head on frozen Qwen3-8B question representations. All share the library, executor, downstream top-3 evidence, budgets, decoding, and test questions. Routers receive no retrieved passages or retrieval-derived features; no predictor is fitted on certification or test questions.

Table 24: **Query-only routing comparisons.** Results use the same primary run as Table 27. Differences are Full Remedy-RAG minus each control, with paired 95% bootstrap intervals in brackets.

Router	F1 ↑	$\overline{C}$ ↓	$\widetilde{U}_{\mathrm{F1}}$ ↑	ΔF1 (95% CI)	$\Delta\widetilde{U}$ (95% CI)
Query-only: E5-4D	0.491	0.632	0.428	0.037 [0.030, 0.045]	0.041 [0.033, 0.050]
Query-only: full E5 + MLP	0.500	0.630	0.437	0.028 [0.020, 0.035]	0.032 [0.024, 0.040]
Query-only: E5 $\to \widehat{e} \to M$	0.505	0.618	0.443	0.023 [0.016, 0.031]	0.026 [0.018, 0.033]
Query-only: frozen Qwen3-8B	0.508	0.612	0.447	0.020 [0.013, 0.026]	0.022 [0.015, 0.030]
Full Remedy-RAG	0.528	0.589	0.469	—	—

Results. With the shared workflow library and verification protocol, evidence-state routing reaches 52.8 F1, compared with 50.0 for the query-only E5-MLP. The paired gain is 2.8 F1 points (95% CI $[2.02, 3.51]$). Normalized cost decreases from 0.630 to 0.589, a 6.5% reduction, while balanced utility increases from 0.4370 to 0.4695. The paired utility gain is 0.0325 (95% CI $[0.0236, 0.0404]$).

Greater query-only capacity improves F1 from 49.1 for E5-4D to 50.0 for E5-MLP and 50.8 for the frozen Qwen3-8B query controller. The latter still trails evidence-state routing by 2.0 points (95% CI $[1.34, 2.62]$). These comparisons support making the retrieved evidence explicit in the tested deployment representations. They connect workflow selection to the evidence available after retrieval, beyond what the tested query-only controllers recover from question representations.

C.5.4 Joint Accuracy and Cost Decomposition of Certified Advantages

Table 25 separates accuracy and cost on the deployment-policy certification fold. Accuracy gain is positive in every deployed region and contributes 0.1133 of the 0.1334 coverage-weighted advantage (about 85%); none has a positive mean solely from saving cost. Held-out audit gains are reported separately in Table 11.

Table 25: **Decomposition of certified regional advantages.** $\Delta\mathrm{acc}$ is the paired change in token-level F1 and $-\lambda\Delta\overline{C}$ is the contribution from trajectory cost, so their sum is the certification-fold paired workflow advantage $\widehat{G}_r$. Shares are certification-fold leaf proportions.

Region	Assigned strategy	Share	$\Delta\mathrm{acc}$	$-\lambda\Delta\overline{C}$	$\widehat{G}_r$
R_2	Contradiction resolution	7.55%	0.113	0.026	0.139
R_3	Query repair	11.40%	0.136	0.025	0.161
R_5	Multi-hop completion	19.05%	0.178	0.027	0.205
R_6	Evidence denoising	15.35%	0.146	0.030	0.176
R_7	Contradiction resolution	12.70%	0.165	0.026	0.191
R_8	Lightweight confirmation	9.30%	0.129	0.024	0.153
Coverage-weighted contribution		75.35%	0.1133	0.0201	0.1334
Certification-fold mean among intervened questions					0.1770

C.5.5 Frequency-Matched Counterfactual Control

Held-out validation traces estimate each verified region's operation marginals $p_{\mathrm{val}}(o \mid r)$ and trace-length distribution. At test time, a scheduler receives the same probe state and region as the full method and samples valid operations from these distributions. It preserves regional preferences and expected length in expectation while removing learned transition dependencies, evidence-dependent ordering, and adaptive stopping. Executor, retriever, interface, probe, and budget are matched across three fixed execution streams. Only accepted post-probe actions estimate these distributions. Requested, rejected, and initial-probe actions have separate log fields; a hard assertion checks every executed action against its workflow mask. Matching marginals improves mean F1 from 0.4477 to 0.4826 and utility from 0.3703 to 0.4211. The full method retains advantages of 4.8 F1 and 5.0 utility points while reducing mean cost from 0.6152 to 0.5941 (3.43%). Higher F1 and utility, together with lower cost, hold in all three seeds (Table 27). This supports the contribution of workflow structure collectively.

Table 26: **Frequency-matched counterfactual analysis.** All three rows average the same three fixed execution streams on the same test questions. Each run first macro-averages NQ, HotpotQA, and 2WikiMultiHopQA equally; the table then averages those run scores. Utility uses $\lambda = 0.1$. Main-table reference results retain the primary run. The counterfactual preserves the region-conditioned marginal operation and trace-length distributions estimated from held-out validation traces; learned workflow structure is removed.

Method	F1 ↑	$\overline{C}$ ↓	$\widetilde{U}_{\mathrm{F1}}$ ↑
Bare action protocol	0.4477	0.7740	0.3703
Frequency-matched control	0.4826	0.6152	0.4211
Full method	**0.5307**	**0.5941**	**0.4713**

Across these three runs, the bare-protocol cost mean is 0.7740 and its sample SD is 0.0023. Every row uses the same 3,000 question IDs, parser, token limits, timeout, retry policy, and treatment of failed or truncated requests. The question-level cost ledger stores generated-token count, retrieval count, terminal status, and retry count before any run mean is formed; failed attempts are charged through the recorded stopping point rather than dropped.

C.5.6 Context-Augmented Routing under Matched Supervision

Inputs and scope. The empirical comparison evaluates the P_1 state router against Context-Tree and Context-MLP. Both controls receive $z = [h(q, D_0); e_0]$, where $e_0 = (\mathrm{SNR}, \mathrm{Cov}, \mathrm{Con}, \mathrm{Conf})$. We reuse the frozen 184M encoder and its original 2,000-question discovery-stage distillation. We mean-pool the final hidden states over non-padding, non-special tokens in the coverage sequence and three contradiction-pair sequences, then concatenate the four vectors. Sequence order is coverage, $(1, 2)$, $(1, 3)$, $(2, 3)$; tokenization, passage order, and truncation match the original extractor. Both controls augment e_0 with frozen-encoder representations of the same truncated inputs. Encoder-visible inputs

Table 27: **Frequency-matched counterfactual analysis by seed.** Each row macro-averages the same 1,000 test questions per dataset across NQ, HotpotQA, and 2WikiMultiHopQA. Utility is $\widetilde{U}_{\mathrm{F1}} = \mathrm{F1} - 0.1\overline{C}$. Utility and the three-seed means appear in Table 26. Utilities are calculated before rounding the displayed F1 and cost.

Run	Method	F1 ↑	$\overline{C}$ ↓	$\widetilde{U}_{\mathrm{F1}}$ ↑
1	Bare action protocol	0.4481	0.7740	0.3707
1	Frequency-matched control	0.4840	0.6090	0.4231
1	Full method	0.5284	0.5890	0.4695
2	Bare action protocol	0.4437	0.7718	0.3665
2	Frequency-matched control	0.4790	0.6143	0.4176
2	Full method	0.5315	0.5925	0.4722
3	Bare action protocol	0.4513	0.7763	0.3737
3	Frequency-matched control	0.4849	0.6225	0.4226
3	Full method	0.5322	0.6008	0.4721

follow the original extractor specification in Appendix A.1; reused continuations follow the controls in Appendix B.3.

Training. The five workflows, frozen Qwen3-8B executor, utility, and 4,000 growing questions are shared. Each question has four paired continuations for one uniformly assigned workflow w_i and the bare, full-registry control. Their mean utility difference is the observed target $\widehat{g}_i$. Context-Tree uses z with the original fitting objective and split procedure, maximum depth three, and minimum leaf size 120. Context-MLP has one 128-unit GELU hidden layer, dropout 0.1, and five linear outputs. It minimizes

$$\mathcal{L}(\phi) = \frac{1}{n}\sum_{i=1}^{n}\left(f_{\phi,w_i}(z_i) - \widehat{g}_i\right)^2.$$

Only the observed workflow contributes to each question's loss. We use AdamW, learning rate 10^{-3}, weight decay 10^{-2}, batch size 128, and a fixed initialization stream, for at most 200 epochs. Five-fold validation within growing is stratified by dataset and assigned workflow. Training-fold statistics standardize features, with a standard-deviation floor of 10^{-6}; all continuations of a question remain together. Architecture and optimization settings are fixed before validation. The epoch minimizing pooled out-of-fold observed-arm squared error is selected, with earlier epochs preferred in ties; we then refit on all growing questions. No additional development, certification, or test outcomes enter fitting. The proposed action maximizes predicted gain, with no intervention assigned zero gain; ties favor no intervention, then a fixed workflow order.

Independent admission. Both controls are frozen before evaluation on a new 2,000-question certification set, comprising 1,000 NQ and 1,000 HotpotQA questions. This set is shared by the two controls and disjoint from all previously used construction, development, RL, and evaluation sets. It provides the same certification sample budget as the original P_1 protocol; P_1 and its original decisions remain unchanged. Context-Tree certifies its own provisional leaf assignments. Context-MLP certifies its frozen proposal groups $A_w = \{X : \widehat{w}(X) = w\}$. Each nonempty proposal is paired with the bare control using four matched continuations per arm, temperature 0.2, top-$p = 0.95$, and the original execution limits. Groups with fewer than 120 paired questions return no intervention. For supported groups, 2,000 dataset-stratified, question-level studentized max-t bootstrap resamples yield one-sided simultaneous 95% lower bounds. Each controller uses its own group family and critical value; only groups with positive bounds retain their proposed workflows. Certification does not refit the controller or change its proposed workflow. Bounds concern group-average utility gains. The MLP has at most five proposal groups; a depth-three tree has at most eight leaves.

Empirical evaluation protocol and accounting. After admission decisions are fixed, we evaluate on the same NQ, HotpotQA, and 2WikiMultiHopQA questions and fixed per-question seeds as Table 2. Missing selected-workflow or bare-control outcomes are executed; previously computed outcomes are reused only when all settings in Appendix B.3 match exactly. Test outcomes never enter training, controller selection, or certification. We evaluate both prespecified controls using

balanced utility $\widetilde{U}_{\mathrm{F1}} = \mathrm{F1} - 0.1C$. Empirical paired differences are defined as Full REMEDY-RAG minus each control. Their percentile 95% CIs require 10,000 question-level resamples within each benchmark, with identical sampled indices for both methods, followed by equal-weight benchmark averaging. Those intervals condition on the fixed library, fitted controllers, and evaluation seed. For each benchmark, the evaluation additionally records intervention coverage before and after certification. Group counts, support, and certification outcomes are recorded separately. The two controls share 2,000 additional certification questions outside the primary 25,600-question pool. Four matched continuations per question provide 8,000 shared bare traces and 8,000 candidate traces for each controller. Fitting both controllers uses 0.2 GPU-hours, and their additional evaluations use 6.3 GPU-hours, for 6.5 in total. These costs are excluded from the primary totals of 27.8 and 101.0 GPU-hours. Normalized C measures generation and retrieval work.

Results. Under matched paired-gain supervision, the compact state router reaches 52.8 F1, compared with 51.0 for Context-Tree and 51.6 for Context-MLP. The paired gains are 1.8 points (95% CI $[1.02, 2.53]$) and 1.2 points ($[0.47, 1.97]$), respectively. Normalized cost is 0.589, versus 0.625 and 0.618, corresponding to reductions of 5.8% and 4.7%. At $\lambda = 0.1$, the paired utility gains are 0.0216 (95% CI $[0.0128, 0.0301]$) and 0.0149 ($[0.0071, 0.0227]$), respectively.

Under the matched supervision budget, the compact state router outperforms both context-augmented controls in F1 and normalized cost.

Table 28: **Context-augmented routing under matched supervision.** Differences are Full REMEDY-RAG minus each control, with paired 95% bootstrap intervals in brackets.

Controller	F1 ↑	C ↓	$\widetilde{U}_{\mathrm{F1}}$ ↑	ΔF1 (95% CI)	$\Delta\widetilde{U}$ (95% CI)
Context-Tree	0.510	0.625	0.448	0.018 $[0.010, 0.025]$	0.022 $[0.013, 0.030]$
Context-MLP	0.516	0.618	0.454	0.012 $[0.005, 0.020]$	0.015 $[0.007, 0.023]$
Full REMEDY-RAG	0.528	0.589	0.469	—	—

C.5.7 HELD-OUT RELIABILITY AUDIT UNDER JOINT COVERAGE AND DOWNSIDE CONSTRAINTS

The independent reliability audit N_1 compares certification rules under joint coverage and downside requirements. It uses 3,000 held-out benchmark questions—1,000 each from NQ, HotpotQA, and 2WikiMultiHopQA—selected from the remaining official validation pools by the answer-independent rule in Appendix B.1. The audit preserves the naturally observed evidence-state proportions of these questions; it does not reweight states or generate question text. Workflow and no-intervention utilities come from the same paired execution protocol under matched decoding streams, and no final-test utility enters question selection or policy fitting.

Query-level regression and lower-tail metrics. For a frozen deployment rule a and evaluation question i, let K_i be the number of matched decoding streams available in the analysis and define

$$\Delta_i^{\mathrm{F1}}(a) = \frac{1}{K_i} \sum_{r=1}^{K_i} \left[\mathrm{F1}(\tau_{ir}^{a}) - \mathrm{F1}(\tau_{ir}^{\varnothing}) \right], \tag{13}$$

$$\Delta_i^{U}(a) = \frac{1}{K_i} \sum_{r=1}^{K_i} \left[U(\tau_{ir}^{a}) - U(\tau_{ir}^{\varnothing}) \right]. \tag{14}$$

We first average matched streams within a question and then give every question equal weight. A regression is $\Delta_i^{\mathrm{F1}}(a) < 0$; a severe regression is $\Delta_i^{\mathrm{F1}}(a) \le -0.25$. Rates use all n final-test questions as the denominator, including rejected or unsupported questions. A fallback executes $\varnothing$ in both arms and therefore contributes exactly zero to both differences and neither indicator. With $m = \lceil 0.05n \rceil$ and ordered utility differences $\Delta_{(1)}^{U} \le \cdots \le \Delta_{(n)}^{U}$, lower-tail loss is

$$L_{0.05}(a) = -\frac{100}{m} \sum_{j=1}^{m} \Delta_{(j)}^{U}(a). \tag{15}$$

All indicators and order statistics use unrounded question-level values. Bootstrap intervals resample question IDs and keep all streams and paired arms for a question together; continuation streams are never treated as independent test observations.

Mutually exclusive splits. Within each dataset, the precommitted ordering assigns 200 questions to development, 400 to certification, and 400 to final testing. The resulting 600/1,200/1,200 splits are question-disjoint after normalized-duplicate removal. Development selects thresholds and freezes provisional candidate assignments; certification computes simultaneous bounds for those frozen assignments; the final test is opened once for the reported coverage, utility, harmful-admission, and tail metrics.

Operational reliability criterion. The execution protocol fixes three independent requirements before collecting the N_1 final-test outcomes: harmful admission at most 1% of admitted test questions, 5% lower-tail loss at most 10 utility points, and intervention coverage at least two thirds. The first limits routing to regions with negative final-test mean gain, the second measures question-level downside even inside positive-mean regions, and the third prevents a reject-most policy. The harmful-admission cap is evaluated after routing with admitted questions as its denominator: a controller covering 75.9% of the 1,200-item test may therefore route at most 9 of its 911 admitted questions to harmful regions. The remaining requirements limit worst-tail loss to one tenth of the F1 scale and require service on at least two of every three questions. Harmful-region count is reported only as a diagnostic and is not a fourth constraint.

We compare global deployment, a development-selected mean gate, mean ranking matched to the LCB policy's coverage, point-estimate tree routing, a regional LCB gate for one workflow, and the full simultaneous-LCB map. All controllers use the same candidate library and split roles. Each paired record shares its initial evidence, executor budget, and decoding stream across the workflow and no-intervention arms.

Table 29: **Reliability under joint coverage and downside constraints in the independent N_1 audit.** Utility gain is the mean improvement among admitted questions; tail loss uses all final-test questions. Harmful admission is the fraction of admitted test questions routed to a region with negative final-test mean gain. Query-level regression and severe regression follow Eq. 13; tail loss follows Eq. 15. Reliability uses the three independent operational requirements in the text.

Admission rule	Coverage	Utility gain admitted	Tail loss	Harmful regions	Harmful admission	Reliable
Global workflow	100.0%	3.42	17.86	3	29.6%	No
Development mean gate	80.8%	11.18	14.35	2	11.8%	No
Matched mean ranking	75.9%	11.05	13.41	2	9.7%	No
Policy tree, point estimate	85.5%	11.85	12.38	2	8.2%	No
Single workflow + regional LCB	58.4%	9.91	3.88	0	0.0%	No
Full simultaneous LCB	75.9%	**14.72**	**4.71**	**0**	**0.0%**	Yes

Mean-gain and point-estimate admission improve average utility but exceed both the harmful-admission and lower-tail budgets. At matched coverage, the full map adds 3.67 conditional utility points over mean ranking, or about 2.8 points when averaged over all final-test questions. Its overall gain is about 11.2 points, compared with about 8.4 for matched mean ranking. It reduces all query-level regressions from 3.8% to 0.7%, reduces severe regressions from 1.5% to 0.1%, and lowers tail loss from 13.41 to 4.71 points. The single-workflow LCB controls downside but misses the coverage floor by 8.3 points. Among the evaluated controllers, only the full simultaneous-LCB map satisfies all three prespecified coverage and downside requirements in the N_1 audit. The design deliberately separates the admission target from deployment safety: simultaneous bounds certify regional expected utility, while held-out regression and tail metrics independently audit question-level downside.

C.5.8 Paired Information for Admission and Sample Efficiency

Data and execution. We examined paired supervision and admission in an auxiliary evaluation with the frozen Qwen3-8B executor. The five workflows, evidence encoder, eight-region P_1 tree, E5

index, prompts, and execution limits were held fixed. We used the dataset versions and question-normalization rules in Appendix B.1. From the remaining NQ and HotpotQA official training splits, we selected 3,000 questions per dataset. We excluded all primary construction, development, GRPO, DeepRAG, and Context-Tree/MLP certification IDs and their normalized-question duplicates. The fixed answer-independent ordering ranked the eligible IDs. The first 2,000 per dataset formed growing; the next 1,000 formed certification, giving 4,000 and 2,000 questions, respectively.

We selected separate test and audit sets from the NQ, HotpotQA, and 2WikiMultiHopQA official validation splits. After excluding all main evaluation IDs and normalized duplicates across the allocated sets, we ranked the remaining IDs with the same precommitted selection rule. Within each dataset, ranks 1–1,000 formed test and 1,001–2,000 formed audit, giving two disjoint 3,000-question sets. All assignments were fixed before execution and remained unchanged across seeds. These auxiliary sets neither replace the main evaluation questions nor update the deployed P_1 map.

Growing questions exposed one uniformly assigned workflow and bare, with four matched continuations per arm. Assignment and question ordering were seeded within each dataset. Growing, certification, audit, and test all used temperature 0.2 and top-$p = 0.95$. Each pair shared initial evidence, decoding seed, and execution budget. Final answers were scored against benchmark references using the original normalization, aliases, and token-level F1 scorer. Actual generated-token and retrieval counts gave C under Eq. 10, and $U = \mathrm{F1} - 0.1C$. All phases therefore share the deployment decoding distribution; the comparison tests learning from growing traces followed by independent admission on a disjoint fold.

Matched router learning. The outcome and paired-gain routers received identical growing observations, four state scores, and network initialization. Both used a 4–128–6 State-MLP, including a bare head. For averaged arm utilities Y_{i,A_i} and $Y_{i,0}$, outcome regression fitted $Y_{i,A_i} - 0.42$ and $Y_{i,0} - 0.42$. Gain regression fitted $d_i = Y_{i,A_i} - Y_{i,0}$ and a zero bare-head target. The two head losses received equal weight. Workflow scores were $f_w(e) - f_0(e)$ and $g_w(e)$, respectively, with bare scored zero and preferred in ties.

Both networks used AdamW, learning rate 10^{-3}, weight decay 10^{-2}, batch size 128, dropout 0.1, and at most 200 epochs. Training-fold statistics standardized features, with a standard-deviation floor of 10^{-6}. Five-fold validation, stratified by dataset and assigned workflow, selected the epoch minimizing pooled held-out contrast MSE. Ties favored earlier epochs, followed by refitting on the entire growing subset.

We used ten fixed training/execution seeds. Per-question continuation seeds were deterministically derived from the question ID, master seed, and continuation index and shared across arms. Growing and deployment records were cached separately. For each seed, the six-arm certification bank and single-continuation six-arm test bank were materialized once and reused across router fits and growing budgets; the four-continuation audit bank was shared across seeds. For each master seed, growing budgets $250, 500, 750, 1000, 1500, 2000, 3000, 4000$ used nested, dataset-balanced subsets of the same ordered growing pool. The two routers shared assignments, continuations, validation folds, and held-out banks within each seed.

Independent admission and audit. We froze each fitted router before grouping certification questions by its proposed workflow. Groups with fewer than 120 questions returned bare. For supported groups, 2,000 dataset-stratified, question-level studentized max-t resamples provided simultaneous one-sided 95% lower bounds over at most five non-bare proposal groups. The bootstrap seed was $1000s + n$ for master seed s and growing budget n. A nonpositive bound rejected the proposal without selecting another workflow. Test evaluation then used one continuation per arm and equal-weight averaging over the three datasets. Contrast MSE used the same uniformly assigned held-out workflow for both routers, selected without reference to their predictions or test outcomes.

The independent audit bank contained four deployment continuations for all five workflows and bare, with a fixed master stream. Within each fixed region, we estimated $\widehat{G}^{\mathrm{audit}}_{\ell}(w)$ as the mean paired utility difference, setting the bare gain to zero. Audit outcomes were withheld from training, epoch selection, candidate selection, and certification. Empirical conditional-oracle regret was

$$\widehat{\mathcal{R}}_{\mathrm{audit}} = \frac{100}{n} \sum_{i=1}^{n} \left[\max_{w \in \mathcal{W} \cup \{\varnothing\}} \widehat{G}^{\mathrm{audit}}_{\ell(i)}(w) - \widehat{G}^{\mathrm{audit}}_{\ell(i)}(\widehat{w}_i) \right].$$

Here $\widehat{w}_i$ is the router's pre-admission proposal. The same independently estimated regional gains defined negative assignments in Appendix C.5.10.

Table 30: **Matched learning and admission diagnostics.** One utility point equals $0.01U$. Learning entries are mean $\pm$ SD over ten matched training/execution seeds. Bound margins are mean $\pm$ SD over 120 outer question-level bootstrap repetitions of the augmented fixed-candidate certification bank, each using 2,000 inner max-t resamples. Positive-region admission uses the six regions with positive means in the independent audit bank. Both learning controls use paired admission.

(a) Final-policy utility gain over bare

Growing questions	Outcome router	Paired-gain router
1,500	7.10 ± 0.62	8.92 ± 0.42
4,000	8.77 ± 0.46	9.62 ± 0.34

(b) Fixed-region inference, 2,000 certification questions

Metric	Covariance ignored	Covariance retained
Estimate-to-LCB margin (points)	3.98 ± 0.10	1.53 ± 0.05
Positive-region admission (%)	30.7	99.4

At 4,000 growing questions, outcome regression retained 91.2% of the paired router's mean utility improvement (Table 30a). The paired router's mean gain at 1,500 questions exceeded the outcome router's mean at 4,000. This comparison used 62.5% fewer growing questions, while holding the 2,000-question certification budget fixed. Across all eight growing budgets, paired-gain routing has higher mean post-admission utility (Figure 9). At 1,500 and 4,000 questions, its advantages over outcome regression are 1.82 and 0.85 utility points, respectively.

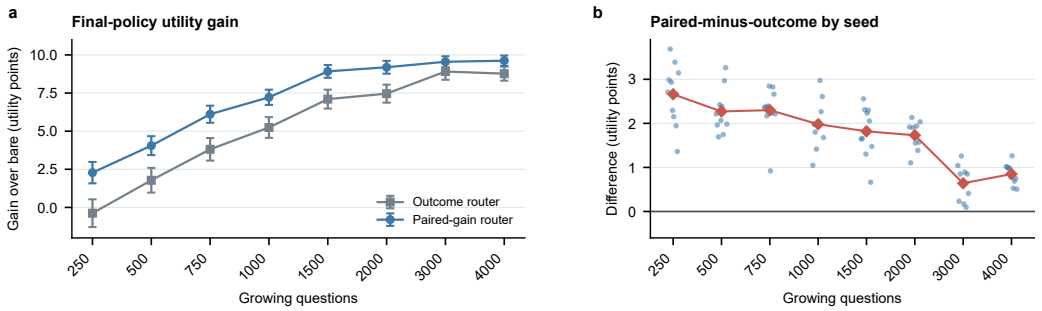

Figure 9: **Sample efficiency under paired admission.** The same ten training/execution seeds and nested growing budgets are used for both routers. (a) Final-policy utility gain over bare; markers show seed means and error bars show sample SD across ten seeds. (b) Each dot is a matched-seed paired-minus-outcome utility difference at the indicated budget; diamonds show the mean. Lines connect budget summaries without smoothing. The 1,500- and 4,000-question summaries are reported in Table 30a.

Admission precision. We next compared uncertainty estimates while holding the eight regions and one candidate fixed. The candidate minimized the support-weighted root mean square of the eight growing-fold gain estimates, with ties resolved by the fixed library order. It was selected before certification or audit outcomes were accessed. This selected a near-boundary contrast for the precision study. The fixed candidate is evidence denoising. This auxiliary analysis augments the primary certification and test question IDs with that candidate under the same four matched streams. Bare outcomes and the existing R_6 candidate outcomes are reused; 6,772 certification and 10,108 test candidate continuations are additional. The fixed-candidate certification comparison therefore contains 16,000 candidate/control continuations, but these are not the primary map-certification ledger. The additional executions are excluded from the one-pipeline adaptation total.

We drew 120 outer bootstrap samples, preserving dataset-by-region counts and keeping all continuations of a question together. Each outer sample used 2,000 inner studentized max-t resamples for simultaneous one-sided 95% lower bounds; regions with fewer than 120 distinct original questions remained unsupported. For outer repetition $r = 0, \ldots, 119$, the outer and inner seeds were $2026092100 + r$ and $2026100100 + r$. The paired procedure resampled workflow and bare together. The covariance-ignored control resampled arms separately and used the sum of their marginal variances, leaving point estimates unchanged.

Margins were averaged across supported regions within each repetition. Positive-region admission measured the fraction of the six supported regions with positive independent-audit mean gain that received a positive bound. Retaining covariance reduced the mean margin by 61.6% and increased admission from 30.7% to 99.4% (Table 30b). Covariance-aware outcome inference gave the same bounds as paired inference on the same observations.

C.5.9 Certification Beyond Reduced Intervention

We then compared admission rules at matched intervention counts using the same frozen regions, candidate, and augmented fixed-candidate certification and test banks. Each of 1,000 outer paired-bootstrap repetitions resampled certification questions within dataset-by-region strata and test questions within datasets. All arm outcomes and continuation replicates remained attached to their question. These repetitions measured resampling variability conditional on the fixed executor, library, regions, and recorded outputs.

Independent deterministic streams were used for certification, test, and inner bootstrap resampling. Each certification sample used 2,000 studentized max-t resamples, retaining the original 120-question support rule. The same inner draws supplied 90%, 95%, and 99% simultaneous bounds over the supported frozen regions. Candidates were never reselected.

The mean-gain control intervened on exactly the same number of test questions as the 95% rule within each repetition. It ranked questions by decreasing certification-fold regional mean gain. A precommitted random ordering resolved ties. Matching used test states without test outcomes.

Table 31: **Admission at matched intervention counts.** Means over 1,000 paired-bootstrap repetitions of the augmented fixed-candidate certification and test banks. Utility is reported relative to mean-gain admission. Query-level regression and severe regression follow Eq. 13; tail loss follows Eq. 15; all rates use the full test set as the denominator and include zero-valued fallback.

Rule	Intervene (%)	ΔU vs. mean	Regress (%)	Severe (%)	Tail loss
Mean-gain threshold	75.31	0.00	1.58	0.72	9.04
Simultaneous LCB, 95%	75.31	+0.64	0.74	0.21	5.85

At matched intervention counts, the 95% rule improves both mean utility and query-level downside relative to the mean-gain threshold (Table 31). It reduces all regressions by 53.2%, severe regressions by 70.8%, and 5% tail loss by 35.3%. The rule repairs 2.16% of test answers and regresses 0.74%; repairs change $\mathrm{F1} < 1$ to $\mathrm{F1} = 1$. Both percentages use all test questions as the denominator. Using shared outer-bootstrap indices, the LCB-minus-mean differences are $+0.64$ utility points (95% CI $[0.39, 0.89]$), -0.84 percentage points for regression ($[-1.08, -0.61]$), -0.51 points for severe regression ($[-0.67, -0.35]$), and -3.19 tail-loss points ($[-3.86, -2.52]$).

We assessed null calibration separately by centering the certification paired differences within each dataset-by-region stratum. The centered bank defined an empirical null with zero stratum means. We drew 1,000 further outer bootstrap samples from this bank using independent deterministic streams and reapplied the LCB admission rule. No-admission rates were 89.4%, 95.3%, and 98.8% at nominal levels of 90%, 95%, and 99%. The 95% rule admitted at least one region in 47 of 1,000 null resamples, or 4.7% (95% Wilson interval, 3.6–6.2%). The calibration summarized rejection under the centered empirical null, conditional on the recorded certification sample.

C.5.10 Fixed-Library Deployment Components

Finally, we held the five workflows, four state scores, and eight regions fixed while varying the deployment rule. This comparison reused the auxiliary 4,000/2,000/3,000 growing, certification, and

test questions from Appendix C.5.8, with the same ten training/execution seeds. Growing exposed one uniformly assigned candidate and bare per question. We selected global or regional candidates using the inverse-probability-weighted growing-fold gains in Appendix B.2, then froze those choices before certification.

Following SkillGen's paired net-gain gate (Ma et al., 2026), we first evaluated a utility-adapted positive-mean rule for one globally selected workflow. It admitted that workflow everywhere when its overall certification mean was positive. Regional gating retained the same workflow only in regions with positive certification means. After regional workflow selection, a development-threshold control chose a cutoff from $\{0, 0.002, 0.004, 0.008, 0.016\}$ before test evaluation. A second mean-based control ranked the frozen assignments by certification-fold regional mean gain and retained assignments until matching the simultaneous-LCB policy's intervention count. The final control instead admitted assignments with positive simultaneous one-sided 95% lower bounds, using 2,000 max-t resamples from a deterministic stream. Coverage matching used only frozen test-state frequencies and certification scores; no test outcome entered ranking or count matching. Unsupported or rejected assignments returned bare.

Each controller accessed only its frozen candidate and bare during certification. Four continuations per arm gave 32,000 exposed growing continuations and 16,000 certification continuations per controller and seed. The shared certification bank recorded all six actions, totaling 48,000 continuations, without exposing unselected outcomes to a controller. Test recorded one continuation per action; the separate audit bank recorded four. Thus, matching controller access did not equate total collection cost with one controller's budget. Auxiliary collection and fitting were accounted separately from the primary 27.8 and 101.0 GPU-hour totals. No target-executor data entered this Qwen3-8B comparison or updated the main transfer map.

Table 32: **Deployment controls sharing the candidate library.** Means over the same ten training/execution seeds as the paired-learning comparison; gain is in utility points. Negative assignments are the percentage of all test questions assigned a workflow with negative independent-audit mean gain in its region. The denominator includes fallback questions, whose assigned gain is zero. Regression and severe regression are query-level outcomes defined in Eq. 13; tail loss follows Eq. 15.

Deployment rule	Intervene (%)	ΔU	Negative (%)
Global net-gain gate	100.00	3.35	23.83
Same candidate + regional gate	76.17	7.15	0.00
Regional selection + dev. mean threshold	77.21	9.58	1.24
Regional selection + matched mean ranking	75.93	9.14	2.43
Regional selection + simultaneous LCB	75.93	9.92	0.00

Regional gating increased mean gain from 3.35 to 7.15 points while retaining the same workflow. Regional selection with the development-selected mean threshold increased it to 9.58 points. At the same 75.93% intervention coverage, simultaneous-LCB admission increases utility from 9.14 to 9.92 points and reduces negative assignments from 2.43% to zero (Table 32). The paired utility difference is $+0.78$ points (95% CI $[0.51, 1.05]$). The two rules differ in two boundary regions: matched mean ranking retains weakly supported assignments, whereas the simultaneous bound replaces them with better-supported assignments at the same test-set intervention count. On the same paired records, query-level regressions fall from 1.61% to 0.68%, severe regressions from 0.73% to 0.18%, and 5% tail loss from 8.37 to 5.15 points. The corresponding paired reductions are 0.93 percentage points ($[0.69, 1.17]$), 0.55 percentage points ($[0.38, 0.72]$), and 3.22 tail-loss points ($[2.56, 3.88]$). This P_1 contribution is included in the P_1 cluster of the cross-map analysis below. The fixed-library comparison separated regional gating from candidate selection, while Appendix C.5.9 examined admission uncertainty at matched intervention counts. These adapted controls are distinct from the native SkillGen comparison in Table 39.

C.5.11 Cross-Map Split-Sensitivity Analysis

The fixed P_1 map shows a clear LCB advantage over matched regional mean ranking. We test whether that difference persists by repeating the matched-coverage comparison on thirteen fitted maps: the three end-to-end reconstructions P_1–P_3 and ten redraws of the growing/certification partition. The three reconstructions use independently built libraries and execution ledgers. The ten redraws share

the P_1 workflow library and held-out paired execution ledger and therefore quantify split sensitivity rather than independent pipeline reconstruction. Each fitted map freezes its assignments before evaluation; matched mean ranking is constrained to the LCB policy's intervention count.

Table 33: **Cross-map certification results with nested dependence.** Differences are LCB minus matched mean ranking. Utility is in points; positive regression and tail reductions mean lower query-level downside under LCB. Regression uses Eq. 13; tail loss uses Eq. 15. Bracketed values are paired 95% intervals for the fixed P_1 map. Counts indicate favorable end-to-end reconstructions, and the all-fitted row reports the signs of the hierarchical cluster-weighted estimates. The hierarchical block bootstrap treats P_1–P_3 as top-level clusters, samples redraws only within the P_1 cluster, and resamples question IDs within their execution ledgers; redraws do not count as independent replications.

Scope	Maps	ΔU	Regress. reduction	Severe reduction	Tail-loss reduction
Fixed P_1 map	1	$+0.780$ $[0.510, 1.050]$	0.93 $[0.69, 1.17]$	0.55 $[0.38, 0.72]$	3.22 $[2.56, 3.88]$
End-to-end reconstructions	3	$+0.783$ (3/3)	3/3	3/3	3/3
All fitted maps (cluster weighted)	13	Positive	Positive	Positive	Positive

LCB and matched ranking differ on all three end-to-end reconstructions and on nine of the ten partition redraws. Every end-to-end reconstruction favors LCB in mean utility, regression rate, severe-regression rate, and 5% tail loss. Across all fitted maps, the hierarchical analysis gives positive cluster-weighted improvements on all four outcomes (Table 33). The agreement between regional-mean and query-level metrics shows that the certification advantage is not produced by regional averaging alone. The end-to-end row supplies the independent replication evidence; the ten P_1 redraws establish split sensitivity without inflating the independent-replication count.

C.6 Learned Strategy Profiles

All five default candidates occupy certified regions. Their post-hoc labels describe targeted deficiencies and observed operations, and are neither discovery inputs nor router classes. At eight candidates, two are pruned because they occupy no certified region. Figure 10 reports routing proportions and deployed action masks; Table 5 gives the corresponding guidance. The frequency-matched control above tests the contribution beyond the marginal assignment proportions.

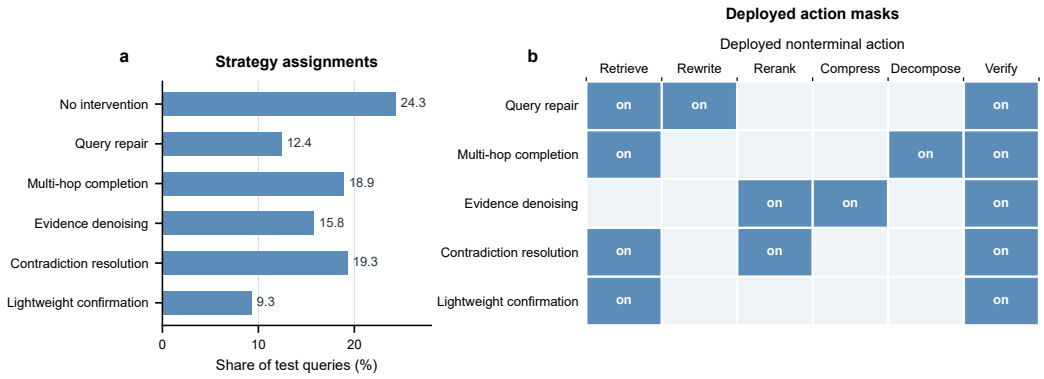

Figure 10: **Learned strategy profiles. (a)** Assignment proportions of no intervention and the five verified strategies. **(b)** Deployed nonterminal action masks. Each marked cell denotes one operation exposed to that workflow. The common initial probe and terminal actions are excluded. Assignment proportions are macro-averaged over NQ, HotpotQA, and 2WikiMultiHopQA.

C.7 RL Execution under Verified Strategies

We freeze the library, tree, certified assignments, and per-question routing, and train only the executor. Development-trace diagnostics are excluded from rewards, tree construction, and certification. An episode adheres if it uses only exposed actions and satisfies at least one assigned dependency. An

Table 34: **Execution quality under fixed verified assignments.** Action and termination statistics are macro-averaged over NQ and HotpotQA development traces. F1 is the mean over NQ, HotpotQA, and 2WikiMultiHopQA test questions.

Metric	Frozen	+RL
Strategy adherence ↑	91.4%	96.8%
Evidence-responsive actions ↑	74.8%	82.6%
Evidence-dependent termination ↑	79.6%	87.1%
Test F1 ↑	0.528	0.551

evidence-responsive action reduces its targeted deficiency by at least 0.05 in the next measured state or directly produces a supported terminal answer. Termination is evidence-dependent if the executor answers after the last useful evidence update or abstains when no available operation improves the audited continuation return. GRPO improves adherence by 5.4 points, evidence responsiveness by 7.8 points, and appropriate termination by 7.5 points. F1 rises from 0.528 to 0.551 under fixed routing.

C.8 Additional Cross-Model Transfer Details

The Qwen3-8B library, evidence encoder, E5 index, tree, thresholds, and leaf actions are frozen when the executor changes; only Conf is recomputed. Discovery and verification both use the frozen Qwen3-8B checkpoint; the resulting workflows and assignments are reused directly in the transfer rows. Qwen3-32B tests transfer to a larger executor within the same family, while GPT-5.4 tests cross-family transfer. Neither target executor contributes to construction of the transferred map. GPT-5.4 exposes first-token log probabilities for the same closed-book confidence computation.

These rows measure gains over each target executor's bare protocol. They establish empirical reuse benefits, without establishing superiority over external methods on those executors or transferring source certificates. The deployment rule therefore separates transferable workflow structure from admission validity: material executor or domain changes retain the workflow library and recertify its assignments on the target distribution.

Common evaluation and aggregation. Tables 35 and 36 use the same 3,000 fixed evaluation IDs, comprising 1,000 questions each from NQ, HotpotQA, and 2WikiMultiHopQA. They use the primary run, the frozen P_1 map, and one matched continuation per arm and question under temperature 0.2 and top-$p = 0.95$. A no-intervention decision reuses the identical bare-control trajectory, so its paired utility difference is zero. Scores and costs are first averaged within each benchmark and then equally across benchmarks. Routing shares and conditional gains use the same weights. These are single-run summaries; Table 16's seed-averaged tests retain their separate three-seed protocol.

For a fixed executor, let $a_i = \mathbf{1}\{M(e_i) \neq \varnothing\}$, $\omega_i = 1/(3n_{d(i)})$, and $\delta u_i = \mathrm{F1}_i^M - \mathrm{F1}_i^{\varnothing} - 0.1(C_i^M - C_i^{\varnothing})$, where $n_{d(i)} = 1{,}000$. For the paired question records,

$$\rho = \sum_i \omega_i a_i, \qquad g_+ = \frac{\sum_i \omega_i a_i \delta u_i}{\rho}, \qquad \Delta\overline{U} = \sum_i \omega_i \delta u_i = \rho g_+. \tag{16}$$

For Qwen3-8B, the primary-run bare and mapped means are $(0.4481, 0.7740)$ and $(0.5284, 0.5890)$ for (F1,cost). They give utilities 0.3707 and 0.4695, and an overall utility gain of 0.0988, matching Table 36. F1 gains are 8.0, 6.2, and 5.3 points on Qwen3-8B, Qwen3-32B, and GPT-5.4; cost reductions range from 23.9% to 34.9%. As mean Conf rises from 0.41 to 0.60, no-intervention routing rises from 24.3% to 34.0% through Conf-gated regions. Evidence-gated share stays at 47.5%, because SNR, Cov, and Con remain fixed (Table 36).

Qwen3 provides next-token `Yes`/`No` logits without token generation; GPT-5.4 uses the one-token diagnostic in Appendix A.1. The Qwen3-8B timing conversion applies only to that local serving configuration. GPT-5.4 accounting includes the diagnostic request, its full input prompt, and its output token in both matched arms; the bare arm does not use the score.

Table 35: **Cross-executor transfer of a source-certified workflow map.** The library, encoder, tree, thresholds, and certified leaf actions are frozen from the Qwen3-8B construction; only Conf_0 is recomputed. Results are macro-averaged over NQ, HotpotQA, and 2WikiMultiHopQA under matched budgets. Every target-executor row uses the 8B-derived map; no row uses RL. Δ is measured against the same backbone's bare protocol. Scores, costs, routing shares, and conditional gains follow Eq. 16.

Executor	Control	F1 ↑	$\overline{C}\downarrow$	$\widetilde{U}_{\mathrm{F1}}\uparrow$	ΔF1	$\Delta\overline{C}$
Qwen3-8B	Bare protocol	0.4481	0.7740	0.3707	0.0	0.0%
	REMEDY-RAG (in-domain)	0.5284	0.5890	0.4695	+8.0	−23.9%
Qwen3-32B	Bare protocol	0.5122	0.8614	0.4261	0.0	0.0%
	REMEDY-RAG (map from 8B)	0.5738	0.6086	0.5129	+6.2	−29.3%
GPT-5.4	Bare protocol	0.5478	0.8257	0.4652	0.0	0.0%
	REMEDY-RAG (map from 8B)	0.6012	0.5375	0.5474	+5.3	−34.9%

Table 36: **Routing under the frozen 8B-derived map.** Intervened Conf_0-gated regions are R_5 and R_8; evidence-gated regions are R_2, R_3, R_6, and R_7. The final column is conditional mean paired utility gain g_+ from Eq. 16, using the same primary-run outputs as Table 35. Overall and conditional gains are aggregated from the same paired records and reported to four decimals.

Executor	∅	Conf_0-gated	Evidence-gated	Mean Conf_0	$\Delta\overline{U}$	Conditional gain g_+
Qwen3-8B	24.3%	28.2%	47.5%	0.41	0.0988	0.1305
Qwen3-32B	30.0%	22.5%	47.5%	0.53	0.0869	0.1241
GPT-5.4	34.0%	18.5%	47.5%	0.60	0.0822	0.1246

C.8.1 GPT-5.4 COST ACCOUNTING

Cost accounting. At standard uncached prices of US$2.50/M input and US$15/M output tokens[1], $B_{\mathrm{api}} = (2.5N_{\mathrm{in}} + 15N_{\mathrm{gen}})/10^6$ and $C_{\mathrm{bill}} = B_{\mathrm{api}}/B_{\mathrm{bare}}$. The bare/map means are 6,200/5,465 input tokens, 846/549 generated tokens, 3.18/2.44 retrievals, and 5.2/4.1 API requests. Input totals sum the complete serialized prompt of every request, including repeated evidence and history. Each arm's totals already include one confidence request and one diagnostic output token per query. Thus continuation-only means are 845/548 output tokens and 4.2/3.1 requests. The diagnostic is charged once per arm, including when a routed policy returns no intervention; no separate confidence surcharge is added. These totals give US$28.19/21.90 per 1,000 queries. Component-based latency is computed from the reported resource counts. The displayed model uses $\alpha_{\mathrm{model}} = 0.32$ seconds/request and $\beta_{\mathrm{model}} = 0.016$ seconds/token. Calibration fits $T_i = \widehat{\alpha} + \widehat{\beta} N_{\mathrm{gen},i}$ jointly by nonnegative least squares. The reporting convention is $\alpha_{\mathrm{model}} = \mathrm{round}(\widehat{\alpha}, 2)$ and $\beta_{\mathrm{model}} = \mathrm{round}(\widehat{\beta}, 3)$. All modeled times, including T_{ref}, use these displayed coefficients.

Calibration record and protocol. Joint fitting over 660 timed requests gives $\widehat{\alpha} = 0.32365428$ seconds/request and $\widehat{\beta} = 0.01632580$ seconds/token, with residual RMSE 0.18843 seconds. Rounding to two and three decimal places, respectively, gives the 0.32 and 0.016 coefficients used in every latency calculation below.

We used `/v1/chat/completions`, snapshot `gpt-5.4-2026-03-05`, standard service tier, `reasoning_effort=none`, temperature 0.2, and top-$p = 0.95$. The first 30 questions in each fixed NQ/HotpotQA development set provided 60 questions. Bare and map prompts at output caps $\{32, 64, 128, 256, 512\}$ provided 600 task requests; one confidence request per question added 60 under Appendix A.1's diagnostic settings. The fitting sample comprises 660 successful timed requests and excludes all 50 warm-up requests. Timing used a persistent client, concurrency 1, streaming, and returned usage enabled. Failed attempts were recorded separately. We fitted dispatch-to-final-content time against actual output-token counts, with equal weight per successful request. The per-request output-token counts and timings form the calibration record.

[1] `https://developers.openai.com/api/docs/models/gpt-5.4`

The request term represents transport, queueing, and prefill; generation, retrieval, and encoder time are counted separately. Client-side idle time is excluded. The following component model computes end-to-end latency:

$$T_{\mathrm{e2e}} = 0.32N_{\mathrm{req}} + 0.016N_{\mathrm{gen}} + 0.041N_{\mathrm{ret}} + 0.0061I_{\mathrm{enc}} \quad \text{seconds.}$$

Here $I_{\mathrm{enc}} = 0$ for the bare comparator and 1 for routed policies, including routed no-intervention decisions. Thus the bare/map latencies are 15.33038/10.20214 seconds, respectively. The accounting

Table 37: **Reported GPT-5.4 reference results and accounting.** F1 is retained from Table 35; costs use the reported mean counts and standard uncached text-token rates. Latency is computed from the displayed component coefficients and resource counts.

System	F1	$\overline{C}$	C_{bill}	API US$/1k	E2E latency (s)
Bare protocol	0.548	0.826	1.000	28.19	15.330
REMEDY-RAG	0.601	0.538	0.777	21.90	10.202

gives 22.3% lower API expenditure and 33.5% lower latency under the stated component-based model. The reported three-benchmark test contains 3,000 questions, while Table 16 reports the seed-averaged GPT-5.4 gain of 4.78 F1 points with 95% CI $[3.92, 5.66]$ points.

C.8.2 ROUTING-MATCHED EVALUATION OF CERTIFICATION CRITERIA

Experimental setup. We compare three admission rules on 4,000 growing and 2,000 certification records from two datasets, and 3,000 test records from three datasets. The test records are the fixed evaluation questions in Table 7: 1,000 each from NQ, HotpotQA, and 2WikiMultiHopQA. They retain the main evaluation's dataset revisions, exclusions, deterministic ID ordering, and answer-independent selection rule. For the GPT-5.4 criterion evaluation, we hold SNR, Cov, Con, and the tree fixed and recompute only Conf. Applying the frozen thresholds therefore preserves membership in the evidence-only regions R_2, R_3, R_6, R_7 and the totals of the two confidence-split branches, while allowing assignments to change only within (R_1, R_8) and (R_4, R_5). The resulting leaf counts yield the reference routing proportions in Table 36; no answer or criterion outcome is used to select or reweight test records. The eight-leaf tree and five-workflow library remain fixed during the comparison. The auxiliary growing and certification folds use, respectively, 2,000/1,000 unused NQ training IDs and 2,000/1,000 unused HotpotQA training IDs. A fixed answer-independent permutation is applied after excluding every primary discovery, growing, certification, GRPO, development, and evaluation ID and every normalized-question duplicate. These 6,000 auxiliary question IDs are therefore disjoint from all source-map construction folds. Within the auxiliary study, all criteria share the same question IDs, initial evidence, and frozen evidence-only features; target-executor confidence is computed once before applying the three admission rules. Growing selects one candidate per leaf; all three rules certify that candidate, with no candidate reselection from certification outcomes. Here Acc uses the token-level F1 definition in Appendix B.2; the test score is the equally weighted mean of the NQ, HotpotQA, and 2WikiMultiHopQA stratum means. Resource costs are calculated from logged integer counts. The auxiliary analysis uses separate target-side continuations for growing, certification, and criterion testing. Its test positions and routing states match the main evaluation, while its outcome records and admission policies are separate. It neither updates the transferred 8B map nor contributes to Table 35. Certification averages paired question-level gains within each leaf over its NQ and HotpotQA questions.

Criterion comparison. The first alternative replaces normalized generation-and-retrieval cost with normalized end-to-end latency,

$$C_{\mathrm{lat}}(\tau) = \frac{T_{\mathrm{e2e}}(\tau)}{T_{\mathrm{ref}}}, \qquad U_{\mathrm{lat}}(\tau) = \mathrm{Acc}(\tau) - \lambda_{\mathrm{lat}} C_{\mathrm{lat}}(\tau), \qquad \lambda_{\mathrm{lat}} = 0.1. \tag{17}$$

Regional deployment again requires a positive lower confidence bound on the paired gain in U_{lat}. This criterion preserves the original linear accuracy–resource trade-off while expressing resource use in seconds rather than generated-token and retrieval units.

The second alternative removes scalar utility altogether. For workflow w at evidence state e, define the paired improvement vector

$$\mathbf{g}(w \mid e) = \begin{bmatrix} \mathbb{E}[\mathrm{Acc}(\tau^w) - \mathrm{Acc}(\tau^\varnothing) \mid e] \\ \mathbb{E}[(B_{\mathrm{api}}(\tau^\varnothing) - B_{\mathrm{api}}(\tau^w))/B_{\mathrm{api}}(\tau^\varnothing) \mid e] \\ \mathbb{E}[(T_{\mathrm{e2e}}(\tau^\varnothing) - T_{\mathrm{e2e}}(\tau^w))/T_{\mathrm{e2e}}(\tau^\varnothing) \mid e] \end{bmatrix}. \tag{18}$$

The resource components average per-record relative reductions. Let $L_j(w \mid e)$ be the component-wise lower confidence bounds and prespecify $\boldsymbol{\epsilon} = (0.005, 0.02, 0.02)$. The Pareto-safe set is

$$\mathcal{W}_{\mathrm{safe}}(e) = \left\{ w \in \{w^*(e)\} : L_j(w \mid e) \geq -\epsilon_j \; \forall j, \quad \max_j L_j(w \mid e) > 0 \right\}. \tag{19}$$

The Pareto rule admits the frozen candidate only when it belongs to the safe set; otherwise it returns no intervention:

$$\pi_{\mathrm{P}}(e) = \begin{cases} w^*(e), & w^*(e) \in \mathcal{W}_{\mathrm{safe}}(e), \\ \varnothing, & \text{otherwise.} \end{cases} \tag{20}$$

Unlike either scalar utility, this rule requires component-wise ϵ-non-inferiority and at least one certified improvement.

Sampling and inference. Each test stratum contains 1,000 records. Routing shares are computed from the resulting leaf assignments rather than imposed by outcome-dependent resampling; score gains are measured independently. The reference-map routing groups retain 47.5% evidence-gated records in every condition; their no-intervention shares are 24.3%, 30.0%, and 34.0%, respectively. Mean confidence is 0.41, 0.53, and 0.60.

Growing selects the candidate with the largest mean primary gain, breaking ties by workflow ID. Certification falls back below 120 records. Within each supported leaf, 2,000 paired record-level bootstrap resamples share indices across primary utility, latency utility, accuracy, billing, and latency gains. Max-t calibration across supported leaves and these five metrics gives simultaneous one-sided 95% lower bounds. The margins $(0.005, 0.02, 0.02)$, $\lambda_{\mathrm{lat}} = 0.1$, and $T_{\mathrm{ref}} = 15.33038$ seconds remain fixed. After admission, 2,000 paired within-stratum test resamples give marginal two-sided 95% percentile intervals for equally weighted alternative-minus-primary differences, conditional on the fitted policies. All policies share resampling indices.

The certification and test supports for R_1–R_8 are $(391, 180, 238, 279, 243, 285, 262, 122)$ and $(592, 228, 373, 428, 385, 473, 351, 170)$, respectively. The simultaneous critical value is 2.6795. The evidence-gated test count is $228 + 373 + 473 + 351 = 1425$, giving $1425/3000 = 47.5\%$.

Table 38: **Comparison with routing matched to the reference proportions.** Changes are relative to the primary rule on the same 3,000 test records. Accuracy changes and intervals use percentage points; negative resource changes denote savings. Agreement compares eight leaf assignments.

Criterion	Agreement	ΔAcc.	95% CI	ΔAPI US$/1k	ΔE2E latency (ms)
Latency utility	8/8	0.00	[0.00, 0.00]	0.000	0.000
Pareto-safe	7/8	−1.27	[−1.69, −0.84]	−0.102	−5.971

Mean count changes per test record, relative to the primary rule:

Rule	ΔInput tokens	ΔOutput tokens	ΔRequests	ΔRetrievals
Latency utility	0.0000	0.0000	0.000000	0.000000
Pareto-safe	−34.0390	−1.1143	0.033000	0.031667

Results. Latency utility retains all eight primary assignments and produces identical outcomes. The only changed leaf is R_4: the auxiliary primary-utility rule admits its frozen candidate on 428 of the 3,000 test questions, whereas Pareto-safe returns the bare protocol there. This target-side criterion comparison does not alter the transferred 8B map, which already falls back in R_4. The Pareto rule retains 7 assignments, with an accuracy change of -1.27 percentage points (95% CI $[-1.69, -0.84]$), an API-cost change of US$$-0.102$ per 1,000 records, and a latency change of -5.971 ms per record (Table 38).

C.8.3 Independent Target-Side Discovery Comparison

A separate GPT-5.4 Discover-and-Verify run reaches 0.618 F1. The transferred map reaches 0.601 versus a 0.548 bare protocol, recovering 76% of that gain without target-side rollout for map construction. This reference is excluded from the transfer tables and does not recertify their map. Corpus, retriever, data splits, candidate count, depth, support thresholds, and confidence level are matched; only the base policy generating discovery and verification traces changes.

Post-hoc functional matching requires both the selected action set and the targeted deficiency to agree. Four of five candidates and four of eight leaf labels match. Differences occur in admission at R_2, R_4 and in workflow choice at R_7, R_8: the 8B library separates contradiction resolution less sharply from lightweight confirmation.

C.9 Qualitative Missing-Hop Case

Figure 11 gives the matched missing-hop case referenced in the main text, including the initial ranking and alternative continuations.

Question. Who is the father of the director of film *Red 2 (Film)*? **Reference answer.** Aldo Parisot.
Verified chain. *Red 2* $\xrightarrow{\text{director}}$ Dean Parisot $\xrightarrow{\text{father}}$ Aldo Parisot. The post-probe state is $(\mathrm{SNR}, \mathrm{Cov}, \mathrm{Con}, \mathrm{Conf}) = (0.60, 0.27, 0.09, 0.33)$, with $0.60 = \widehat{F}_{\mathrm{disc}}(0.899)$, and routes to R_5.

Initial evidence. Frozen E5 ranks (cosine scores in parentheses):

1. *Red 2* (0.842): "with **Dean Parisot directing** ";
2. Peter Hamel (0.781): **father of Peter Michael Hamel** ;
3. Paul Brooke (0.771): **father of Tom Brooke** .

The answer-bearing Dean Parisot passage is rank 10 (0.726). The probe resolves the director but misses the requested father relation; the distractors match *director/father* for different people.

Routed completion. PROBE → DECOMPOSE → RETRIEVE → VERIFY → ANSWER. The entity-resolved query **Who is Dean Parisot's father?** promotes two Dean Parisot passages to ranks 1 and 2. The answer-bearing rank-2 passage (0.847) states that he was born to Ellen James and **Aldo Parisot** . Verification completes the two-hop chain.

Figure 11: **Entity-conditioned retrieval completes a missing hop.** The original query resolves the director but favors unrelated father relations. The routed workflow retrieves for the resolved entity and recovers direct support. Colors mark **support** , **distractors** , and **targeted retrieval** .

C.10 Comparison with Recent Skill-Learning Methods

To assess whether evidence-conditioned workflow deployment improves performance relative to recent skill-learning systems, we compare with SkillGen (Ma et al., 2026) and SkillRL (Xia et al., 2026a) across the five general-domain QA benchmarks. Both baselines construct their own skills from RAG trajectories using the shared eight-operation interface. Both baselines follow their original skill-learning and deployment procedures within the shared eight-operation RAG interface.

All systems use Qwen3-8B, the same Wikipedia index, initial top-3 evidence, tool implementations, context capacity, decoding settings, and limits of four retrievals, six interactions, and 1,024 generated tokens. SkillGen-RAG and SkillRL-RAG use Qwen3-32B for offline skill construction; REMEDY-RAG uses its frozen Qwen3-8B executor for Discover and Verify. Frozen methods access the same 8,000 construction questions; trained methods access the same 25,600-question pool, with each method allocating questions across its own learning stages. Configurations and checkpoints are selected on the shared 1,000-question development set by $\mathrm{F1} - 0.1C$. Offline GPU-hours include skill construction and all applicable training and skill-update stages.

In this comparison, REMEDY-RAG gains 3.3 macro-F1 points over SkillGen, while REMEDY-RAG+RL gains 4.0 points over SkillRL. Normalized generation-and-retrieval cost is 0.686 versus 0.586 in the frozen comparison and 0.807 versus 0.646 in the trained comparison.

Table 39: **Recent skill-learning methods in the shared RAG environment.** F1 and EM are macro-averaged over five benchmarks; C follows Eq. 10. REMEDY-RAG entries reproduce Tables 1 and 10.

Method	F1 ↑	EM ↑	C ↓	Offline GPU-h ↓
Frozen executor				
SkillGen-RAG	0.475	0.430	0.686	37.0
REMEDY-RAG	**0.508**	**0.463**	**0.586**	**27.8**
Trained executor				
SkillRL-RAG	0.493	0.441	0.807	148.0
REMEDY-RAG+RL	**0.533**	**0.486**	**0.646**	**101.0**

D THEORY AND PROOFS

D.1 OPTIMAL-WORKFLOW INFORMATION BOUND FOR EXECUTOR-SIDE ROUTING

Human deficiency labels support the diagnostic comparison in Section 3.1, but their class-average treatment contrasts need not identify the optimal workflow at each state. The following bounds instead use state-conditional workflow utility.

Let $X = (q, D_0)$ take values in the measurable post-retrieval state space $\mathcal{X}$, and let $\mathcal{S} = \mathcal{W} \cup \{\varnothing\}$ the finite workflow set, including the no-guidance, full-registry control. Each candidate includes guidance and an exposed action subset. Define

$$\mu_w(x) = \mathbb{E}[U(\tau^w) \mid X = x]. \tag{21}$$

Subtracting $\mu_\varnothing(x)$ recovers incremental utility without changing optimizers or regret. A router observes $Z = \psi(X) \in \mathcal{Z}$, such as an executor signal or measured evidence state; any randomness used to construct Z is included in X. A randomized Z-measurable router $f(w \mid Z)$ has value

$$V(f; Z) = \mathbb{E}\left[\sum_{w \in \mathcal{S}} f(w \mid Z)\mu_w(X)\right], \qquad V_Z^\star = \sup_f V(f; Z). \tag{22}$$

Proposition 2 (Value of a routing representation)**.** *For every representation $Z = \psi(X)$,*

$$V_Z^\star = \mathbb{E}\left[\max_{w \in \mathcal{S}} \mathbb{E}[\mu_w(X) \mid Z]\right] \le \mathbb{E}\left[\max_{w \in \mathcal{S}} \mu_w(X)\right] = V_X^\star. \tag{23}$$

Choose measurable, deterministically tie-broken maximizers $w_X^\star(x) \in \arg\max_w \mu_w(x)$ and $w_Z^\star(z) \in \arg\max_w \mathbb{E}[\mu_w(X) \mid Z = z]$. Then

$$V_X^\star - V_Z^\star = \mathbb{E}[\mu_{w_X^\star(X)}(X) - \mu_{w_Z^\star(Z)}(X)]. \tag{24}$$

The pointwise workflow margin

$$\Gamma(X) = \mu_{w_X^\star(X)}(X) - \max_{w \ne w_X^\star(X)} \mu_w(X) \tag{25}$$

is nonnegative and zero at a tie. Every randomized router satisfies

$$V_X^\star - V(f; Z) = \mathbb{E}\left[\sum_{w \in \mathcal{S}} f(w \mid Z)\big(\mu_{w_X^\star(X)}(X) - \mu_w(X)\big)\right] \tag{26}$$

$$\ge \mathbb{E}[\Gamma(X)(1 - f(w_X^\star(X) \mid Z))]. \tag{27}$$

In particular,

$$V_X^\star - V_Z^\star \ge \mathbb{E}[\Gamma(X)\mathbf{1}\{w_X^\star(X) \ne w_Z^\star(Z)\}]. \tag{28}$$

Thus aliasing incurs positive loss when it changes the optimal action on a positive-probability set with a positive margin.

Proof. Conditioning on Z makes $V(f; Z)$ a convex combination of the conditional utilities; its maximum is attained by $w_Z^\star$. Conditional Jensen applied to the maximum gives Eq. 23. The tower property gives Eq. 24. Averaging action-specific losses gives Eq. 26; every nonoptimal action loses at least $\Gamma(X)$. Substituting $w_Z^\star$ yields Eq. 28. □

These statements allow arbitrary effect heterogeneity within deficiency classes. To quantify overlap, let $C^\star = w_X^\star(X)$ and define

$$\Lambda_i(B) = \mathbb{E}[\Gamma(X)\mathbf{1}\{C^\star = w_i,\, Z \in B\}], \qquad B \subseteq \mathcal{Z}. \tag{29}$$

Write $\nu = \sum_i \Lambda_i$, $\ell_i = \mathrm{d}\Lambda_i/\mathrm{d}\nu$, and $\omega_i = \Lambda_i(\mathcal{Z}) = \mathbb{E}[\Gamma(X)\mathbf{1}\{C^\star = w_i\}]$. For $\omega_i > 0$, $Q_i^\Gamma = \Lambda_i/\omega_i$ is the normalized margin-weighted law of Z in class w_i.

Corollary 1 (Heterogeneity-robust overlap floor)**.** *Every randomized Z-measurable router satisfies*

$$V_X^\star - V(f;Z) \geq \int_{\mathcal{Z}} \left(\sum_i \ell_i(z) - \max_i \ell_i(z) \right) \mathrm{d}\nu(z) \tag{30}$$

$$\geq \max_{i<j} \int_{\mathcal{Z}} \min\{\ell_i(z), \ell_j(z)\} \mathrm{d}\nu(z) \tag{31}$$

$$\geq \max_{\substack{i<j \\ \omega_i\omega_j>0}} \min\{\omega_i, \omega_j\}\big(1 - \mathrm{TV}(Q_i^\Gamma, Q_j^\Gamma)\big). \tag{32}$$

The last maximum is zero if no eligible pair exists. No constant-margin or independence assumption between $\Gamma(X)$ and Z is required.

Proof. Set $\eta_i(z) = f(w_i \mid z)$. The margin-weighted regret is $\int(\sum_i \ell_i - \sum_i \eta_i \ell_i)\,\mathrm{d}\nu$. Since η lies in the probability simplex, $\sum_i \eta_i \ell_i \leq \max_i \ell_i$, while $\sum_k \ell_k - \max_k \ell_k \geq \min\{\ell_i, \ell_j\}$ for every pair. Finally, if $\omega_i \leq \omega_j$, $\min\{\omega_i \mathrm{d}Q_i^\Gamma, \omega_j \mathrm{d}Q_j^\Gamma\} \geq \omega_i \min\{\mathrm{d}Q_i^\Gamma, \mathrm{d}Q_j^\Gamma\}$. Integration yields the last inequality; the other ordering is symmetric. □

Define the Bayes error for predicting $C^\star$ from Z as $\varepsilon_Z^{\mathrm{Bayes}} = 1 - \mathbb{E}[\max_w \Pr(C^\star = w \mid Z)]$. Then

$$V_X^\star - V_Z^\star \geq \sup_{\gamma>0} \gamma \left[\varepsilon_Z^{\mathrm{Bayes}} - \Pr\{\Gamma(X) < \gamma\} \right]_+, \qquad [a]_+ = \max\{a, 0\}. \tag{33}$$

Indeed, Eq. 28 is at least γ times the routing-error probability minus $\gamma \Pr\{\Gamma(X) < \gamma\}$; the error of $w_Z^\star$ is no smaller than the Bayes error.

Corollary 2 (Pairwise aliasing floor)**.** *For each optimal-workflow label with $\pi_i = \Pr(C^\star = w_i) > 0$, let $P_i = P_{Z|C^\star=w_i}$ and assume $\Gamma(X) \geq \gamma_i > 0$ almost surely conditional on $C^\star = w_i$. Every randomized Z-measurable router satisfies*

$$V_X^\star - V(f;Z) \geq \max_{i<j} \big(1 - \mathrm{TV}(P_i, P_j)\big) \min\{\pi_i\gamma_i, \pi_j\gamma_j\}. \tag{34}$$

For two equally likely labels with common lower margin γ, this is $V_X^\star - V(f;Z) \geq \frac{\gamma}{2}(1 - \mathrm{TV}(P_1, P_2))$.

Proof. Let $\alpha_i = \Pr_f(f(Z) \neq w_i \mid C^\star = w_i)$. The margin assumption gives $V_X^\star - V(f;Z) \geq \sum_i \pi_i\gamma_i\alpha_i$. For any pair, $q_i(z) = f(w_i \mid z)$ obeys $f(w_j \mid z) \leq 1 - q_i(z)$, so

$$(1 - \alpha_i) + (1 - \alpha_j) \leq \mathbb{E}_{P_i} q_i + \mathbb{E}_{P_j}(1 - q_i) \leq 1 + \mathrm{TV}(P_i, P_j).$$

Thus $\alpha_i + \alpha_j \geq 1 - \mathrm{TV}(P_i, P_j)$. Weighting by $\min\{\pi_i\gamma_i, \pi_j\gamma_j\}$ and maximizing over pairs proves the claim. □

Remark 1 (Why class-average margins are insufficient). Let $\mathrm{Sig} \in \{0, 1\}$ be uniform within two equally likely deficiency classes, giving identical signal laws and zero total variation. Set

$$g_1(0) = 0.4, \quad g_1(1) = -0.1, \qquad g_2(0) = -0.1, \quad g_2(1) = 0.4,$$

where $g_1 = \mu_{o_1} - \mu_{o_2}$ in $\mathcal{D}_1$ and $g_2 = \mu_{o_2} - \mu_{o_1}$ in $\mathcal{D}_2$. Both class means are 0.15; substituting them for lower margins would incorrectly give a floor of 0.075. Yet $f(0) = o_1$, $f(1) = o_2$ selects the better repair in both classes, with zero oracle regret and loss -0.05 relative to the class-constant prescription. A positive class mean therefore cannot replace the conditional lower margin.

Proposition 3 (Diagnostic-class bound under limited reversals)**.** *Let $\mathcal{D} \in \{\mathcal{D}_1, \mathcal{D}_2\}$ have masses π_1, π_2 and distinct prescriptions o_1, o_2. Define*

$$g_1(u) = \mathbb{E}[\mu_{o_1}(X) - \mu_{o_2}(X) \mid \mathcal{D}_1, \mathrm{Sig} = u], \tag{35}$$

$$g_2(u) = \mathbb{E}[\mu_{o_2}(X) - \mu_{o_1}(X) \mid \mathcal{D}_2, \mathrm{Sig} = u]. \tag{36}$$

Suppose measurable sets G_i satisfy

$$\Pr(\mathrm{Sig} \notin G_i \mid \mathcal{D}_i) \le \varepsilon_i, \quad g_i(u) \ge \gamma_i > 0 \text{ on } G_i, \quad g_i(u) \ge -M_i \text{ everywhere}. \tag{37}$$

For every randomized rule $f : \mathrm{Sig} \mapsto \{o_1, o_2\}$,

$$L_{\mathcal{D}}(f) := \sum_{i=1}^{2} \pi_i \mathbb{E}[\mu_{o_i}(X) - \mu_{f(\mathrm{Sig})}(X) \mid \mathcal{D}_i] \tag{38}$$

$$\ge \big(1 - \mathrm{TV}(P_{\mathrm{Sig}|\mathcal{D}_1}, P_{\mathrm{Sig}|\mathcal{D}_2})\big) \min\{\pi_1\gamma_1, \pi_2\gamma_2\} - \sum_{i=1}^{2} \pi_i(\gamma_i + M_i)\varepsilon_i. \tag{39}$$

Since $U \in [-\lambda, 1]$, one may take $M_i = 1 + \lambda$. Unlike oracle regret, $L_{\mathcal{D}}(f)$ compares with class-constant prescriptions and can be negative.

Proof. With $\alpha_i = \Pr_f(f(\mathrm{Sig}) \ne o_i \mid \mathcal{D}_i)$, binary choice gives

$$L_{\mathcal{D}}(f) = \sum_i \pi_i \mathbb{E}[g_i(\mathrm{Sig}) \mathbf{1}\{f(\mathrm{Sig}) \ne o_i\} \mid \mathcal{D}_i].$$

Splitting over G_i and its complement bounds each expectation below by $\gamma_i\alpha_i - (\gamma_i + M_i)\varepsilon_i$. Apply the binary testing inequality $\alpha_1 + \alpha_2 \ge 1 - \mathrm{TV}(P_{\mathrm{Sig}|\mathcal{D}_1}, P_{\mathrm{Sig}|\mathcal{D}_2})$, as in Corollary 2, and weight by $\min\{\pi_1\gamma_1, \pi_2\gamma_2\}$. □

The class-average contrasts in Table 17 do not identify the conditional margins or reversal probabilities required by these bounds. Appendix C.2.2 evaluates empirical best-workflow prediction; its finite-rollout labels and readout accuracies are empirical diagnostics rather than estimates of oracle regret bounds.

E EXACT-MATCH COUNTS

Table 40: **Primary-run general-domain EM counts.** Each benchmark has 1,000 questions; k is the number of exact matches. Macro EM is the unweighted mean over the five benchmarks.

Method	NQ k	HotpotQA k	2Wiki k	TriviaQA k	MuSiQue k	Macro EM
Direct Inference	152	91	79	356	26	0.141
Standard RAG	272	150	166	481	67	0.227
Bare protocol	412	393	409	497	102	0.363
IRCoT	402	388	403	526	76	0.359
Adaptive-RAG	390	364	372	516	69	0.342
CRAG	420	398	412	537	90	0.371
SeaKR	416	389	400	523	83	0.362
Search-R1	436	415	425	606	169	0.410
DeepRAG-RL$_{\mathrm{on}}$	434	434	485	628	186	0.433
HiPRAG	464	454	479	623	218	0.448
CW-GRPO	460	459	482	622	222	0.449
REMEDY-RAG	482	487	490	607	250	0.463
REMEDY-RAG+RL	499	522	502	641	266	0.486

Table 41: **Additional seeded general-domain EM counts.** Each benchmark has 1,000 questions; k is the number of exact matches. Macro EM is the unweighted mean over the five benchmarks.

Method	Run	NQ k	HotpotQA k	2Wiki k	TriviaQA k	MuSiQue k	Macro EM
SeaKR	2	415	395	408	525	90	0.3666
	3	411	387	399	524	71	0.3584
HiPRAG	2	460	455	474	618	215	0.4444
	3	472	462	485	620	214	0.4506
CW-GRPO	2	450	452	464	611	202	0.4358
	3	466	459	487	622	223	0.4514
REMEDY-RAG	2	489	480	499	603	253	0.4648
	3	488	488	491	605	243	0.4630
REMEDY-RAG+RL	2	495	509	488	631	255	0.4756
	3	491	516	496	636	256	0.4790

Table 42: **Primary-run transfer EM counts.** Sample sizes are 500 for OTT-QA and 342 for MedHop; k is the number of correct answers. Scores are rounded to three decimals.

Method	OTT-QA k	OTT-QA EM	MedHop k	MedHop EM
Direct Inference	65	0.130	40	0.117
Standard RAG	111	0.222	67	0.196
Bare protocol	116	0.232	70	0.205
IRCoT	123	0.246	72	0.211
Adaptive-RAG	126	0.252	71	0.208
CRAG	131	0.262	77	0.225
SeaKR	137	0.274	81	0.237
Search-R1	153	0.306	90	0.263
DeepRAG-RL$_{\text{on}}$	160	0.320	94	0.275
HiPRAG	167	0.334	97	0.284
CW-GRPO	169	0.338	98	0.287
REMEDY-RAG	188	0.376	107	0.313
REMEDY-RAG+RL	196	0.392	112	0.327

Table 43: **Complete bare-protocol per-benchmark score ledger.** F1 uses the benchmark's token-level scoring rule, while EM and its integer count are reproduced from the primary-run ledgers above. This table supplies the per-benchmark F1 values for all five general-domain benchmarks and both direct-transfer benchmarks under the shared bare action protocol.

Benchmark	F1	EM count	EM
NQ	0.450	412/1000	0.412
HotpotQA	0.439	393/1000	0.393
2WikiMultiHopQA	0.455	409/1000	0.409
TriviaQA	0.526	497/1000	0.497
MuSiQue	0.123	102/1000	0.102
OTT-QA	0.268	116/500	0.232
MedHop	0.233	70/342	0.205

F LIMITATIONS

Our evaluation focuses on English, text-based question answering. Multilingual queries, multimodal evidence, and extended conversational tasks remain outside the evaluated scope. In addition, the current implementation uses a fixed registry of eight operations and a compact workflow library. Extending REMEDY-RAG to tools with different interfaces or execution semantics may require adapting the workflow representation and evidence-state features. Evaluating these extensions would clarify the applicability of evidence-conditioned workflow reuse beyond the current QA setting.