HELICES: ADAPTIVE EXPANSION OF CAPABILITY FRONTIERS FOR AGENT SELF-EVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-evolution of large language model (LLM)-based agents aims to continually improve their capabilities by discovering and internalizing effective behaviors from past interactions. Yet existing approaches often separate these processes: experience extraction does not consistently adapt to what the agent still needs to learn. We argue that experience extraction and internalization should be coupled around a dynamic capability frontier. Accordingly, we introduce HELICES, a method for adaptive capability-frontier expansion that couples experience extraction and internalization as the policy evolves. HELICES probes this frontier with self-generated natural-language hints, using gains over unguided execution to identify effective behaviors not yet mastered. HELICES jointly optimizes hint generation and utilization through shared parameters, and then internalizes independently verified gains by fully supervising actions and selectively weighting reasoning tokens based on hint attribution. Together, adaptive probing and verified internalization continually expand the capability frontier for independent task solving. Across three agent benchmarks, HELICES surpasses strong baselines by up to 7.2 percentage points and generalizes to unseen tasks without hints.

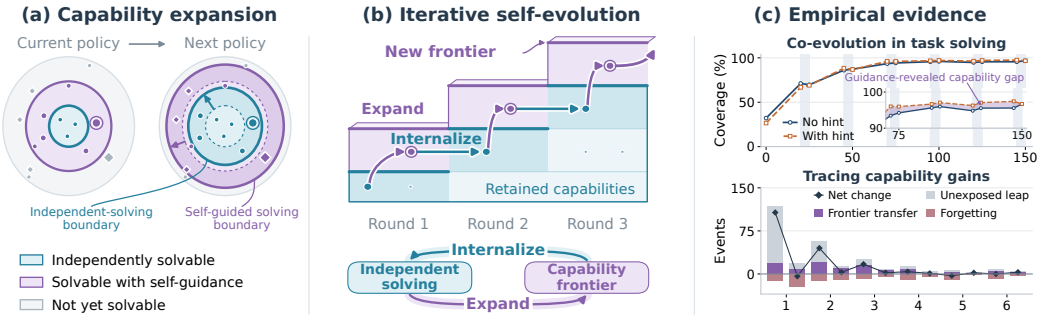


Figure 1: **Self-evolution through capability expansion.** (a) Self-guidance reveals effective behaviors beyond independent task solving. (b) Internalizing these behaviors strengthens the policy, enabling experience extraction at a new capability frontier. (c) Guided and independent capabilities improve together. Independent gains extend beyond previously guidance-reachable tasks.

1 INTRODUCTION

Large language models (LLMs) increasingly enable agents to tackle complex tasks through planning and tool use in real-world environments (Yao et al., 2023; Zhou et al., 2024). During task execution, agents accumulate experience from their actions, environmental feedback, and task outcomes, which can inform the refinement of their task-solving strategies. Accordingly, agent self-evolution aims to turn this accumulated experience into sustained capability gains (Yu et al., 2025; Zhou et al., 2025b). This involves discovering effective behaviors that the agent has not yet mastered and consolidating them into reliable capabilities for independent task solving (Acikgoz et al., 2026; Hou et al., 2026).

Existing work on agent self-evolution explores two directions: *experience extraction* and *experience internalization*. Experience-extraction methods distill successful and failed trajectories into

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.

reusable guidance for subsequent tasks (Shinn et al., 2023; Wang et al., 2025), with some methods organizing and refining this guidance as feedback accumulates (Fan et al., 2025; Li et al., 2025; Liu et al., 2025; Zhang et al., 2026a). Experience-internalization methods use interaction experience to improve policies, reinforcing effective behaviors from successful trajectories and correcting errors using feedback from failed attempts (Wu et al., 2026a; Wang et al., 2026b). Extraction provides guidance in context; internalization consolidates behavior through parameter updates.

However, existing approaches often pursue experience extraction and internalization separately. In particular, capability gains from internalization do not consistently improve the ability of the agent to extract useful experience. As the policy improves, guidance for already-mastered behaviors may offer limited additional learning value, while previously inaccessible behaviors can become opportunities for further learning. A continual self-evolution loop should link these processes in both directions: internalization improves experience extraction, and the resulting experience supports further capability growth at the policy's evolving frontier (Ge et al., 2025; Wu et al., 2026a).

To establish this loop, we formulate agent self-evolution as the *adaptive expansion of a capability frontier*: as the limits of independent task solving move, so do the behaviors most valuable to learn next. Effective guidance should enable the model to solve tasks that it cannot yet solve reliably on its own (Feng et al., 2025; Su et al., 2026). Once the strategies used to solve these tasks have been internalized as independent capabilities, the stronger policy can in turn expose new learning needs and generate more valuable training experience (Zhou et al., 2025b; Acikgoz et al., 2026). Motivated by this principle, we propose **HELICES**, an agent self-evolution method for adaptive capability-frontier expansion. HELICES derives hints as concise conditional rules from source trajectories and provides them as context for fresh task executions. They serve as adaptive probes of the agent's evolving capability frontier, extending their role beyond temporary assistance. Successful interventions reveal behaviors that the unguided policy cannot yet reliably produce. Once these behaviors are verified and internalized, the resulting capability gains guide the next round of hint generation. HELICES therefore establishes a self-evolution cycle that couples experience extraction with internalization to expand the capability frontier of the agent (Figure 1).

Specifically, HELICES extracts concise hints from interaction trajectories and environmental feedback. Each hint specifies an observable condition and guides a fresh execution of the same task. Using gains over unguided execution and task success as the respective rewards, HELICES jointly optimizes hint generation and utilization through shared parameters. As the policy evolves, hint generation also improves, using fresh experience to adapt to what the agent has yet to master. Once joint learning stabilizes, HELICES selects trajectories with independently verified gains over unguided execution. To internalize these gains, we introduce a *selective internalization mechanism guided by hint attribution*. With hints removed from training inputs, it fully supervises actions while selectively weighting hint-supported reasoning to strengthen independent task solving.

The co-evolution of hint generation and utilization builds on the observation that reflecting on a trajectory with environmental feedback can reveal useful corrections even when direct execution fails (Shinn et al., 2023; Dou et al., 2024). In Section 3, we formalize this process using dynamic unguided and hint-reachable capability regions, defined by the tasks reliably solvable without and with hints, respectively. Our analysis decomposes changes in reliable-task coverage into acquisition and forgetting, and derives sufficient conditions for internalization to expand the unguided capability region while accounting for retention without assuming monotonic capability growth.

Across three agent benchmarks and two model scales, HELICES consistently outperforms competing methods, transfers to unseen tasks, and broadens reliable task coverage over repeated cycles without test-time hints, demonstrating cumulative, independently deployable capability gains.

Our main contributions are:

- We identify a missing feedback loop between experience extraction and internalization that hinders continual agent self-evolution, and develop a theoretical formulation of agent capability expansion based on dynamic capability regions and changes in reliable task coverage.
- We propose HELICES for adaptive capability-frontier expansion. It jointly optimizes hint generation and utilization through shared parameters, and introduces a selective internalization mechanism guided by hint attribution to consolidate verified gains into the unguided policy.

- Experiments across agent benchmarks show that HELICES achieves state-of-the-art performance and consistent gains on held-out tasks. These gains persist under unguided execution, supporting its effectiveness in internalizing experience to improve independent task solving.

2 Related Work

Experience extraction for agent self-evolution. Experience-extraction methods convert interaction experience into reusable guidance. Reflexion and ExpeL extract verbal reflections and transferable insights, respectively (Shinn et al., 2023; Zhao et al., 2024). MetaReflection learns instructions from past reflections (Gupta et al., 2024), while feedback-free Meta-Reflection stores insights in a retrievable codebook (Wang et al., 2025). Agent-Pro refines policy-level beliefs (Zhang et al., 2024), and MARS combines principle-based and procedural reflection into instructions (Hou et al., 2026). Experience is organized into structured manuals by AutoManual (Chen et al., 2024), transferable skills by Skill Set Optimization (Nottingham et al., 2024), and hierarchical workflows by MetaFlows (Fan et al., 2025). To adapt experience generation and access, DORA dynamically optimizes reflection prompts (Li et al., 2025), CER supports contextual replay (Liu et al., 2025), and ExpSeek introduces self-triggered experience seeking (Zhang et al., 2026a). These approaches enable adaptation through external guidance, whose reliability, transferability, and availability underpin their effectiveness. External experience updates alone do not consolidate benefits into the unguided policy.

Experience internalization for agent self-evolution. Experience-internalization methods convert interaction experience into parameter updates, strengthening the policy's ability to reliably reproduce effective behaviors. Early work uses tool-use and general agent trajectories for supervised adaptation (Yang et al., 2023; Zeng et al., 2024; Song et al., 2024). Subsequent methods construct learning signals through reflection, search, and failure analysis (Dou et al., 2024; Yu et al., 2025; Ge et al., 2025), while reinforcement learning incorporates corrective feedback and skills into policy improvement (Kumar et al., 2025; Feng et al., 2025; Wang et al., 2026b; Wu et al., 2026a). Adaptive self-improvement also considers how training experience should change with the learner (Zhou et al., 2025b; Acikgoz et al., 2026; Zhang et al., 2026b). Two closely related approaches make the role of guidance explicit. HiPO reuses prefixes from rare on-policy successes as hints to provide a contrastive learning signal (Qiyuan et al., 2026). AgentFrontier synthesizes training data around tasks that the model cannot solve independently but can master with guidance (Chen et al., 2026). HELICES connects experience extraction and internalization by coupling adaptive hint generation with the selective transfer of independently verified behavioral gains into the unguided policy.

3 Formulation of Dynamic Capability Expansion

We develop a task-level account of capability expansion, using successful-trajectory projection under unguided conditioning as an idealized model of internalization. Under the explicit assumptions in Appendix B, we relate this idealized model to the selective supervision and GRPO updates in HELICES and derive sufficient conditions for capability expansion.

3.1 Dynamic capability frontiers

We formulate agent self-evolution around dynamic capability frontiers, coupling experience extraction with capability internalization. This task-level view tracks how policy updates change the set of reliably solvable tasks, distinguishing frontier expansion from improvements in average performance.

For tasks $x \sim \mathcal{D}$, let π_{θ_k} be the solver at learning round k and $\tau_k^0(\omega)$ its unguided trajectory. Here ω captures environment initialization and rollout randomness. Using the success indicator $J_x(\tau) \in \{0, 1\}$, define task-level success probability and the overall average unguided success rate:

$$v_k^0(x) = \mathbb{E}_\omega[J_x(\tau_k^0(\omega))], \qquad S_k = \mathbb{E}_{x\sim\mathcal{D}}[v_k^0(x)]. \tag{1}$$

For a prespecified reliability threshold $\alpha \in (0, 1)$, the unguided capability region at round k is

$$\mathcal{C}_k^0(\alpha) = \{x : v_k^0(x) \geq \alpha\}. \tag{2}$$

The threshold separates this region from the remaining tasks, defining the *capability frontier*, or capability horizon. Under a fixed task distribution, decoding protocol, and execution budget, policy updates can move tasks across this frontier. Higher average success S_k need not expand this region.

3.2 Adaptive frontier probing

We use hints as adaptive probes to expose learning opportunities at the capability frontier. A generator produces a hint from an unguided source trajectory τ_k^{src} and its environmental feedback f_k^{src}:

$$h \sim q_{\phi_k}(\cdot \mid x, \tau_k^{\mathrm{src}}, f_k^{\mathrm{src}}). \tag{3}$$

The hint provides a concise decision rule grounded in the source trajectory, without supplying a complete solution. Its value changes with the current solver and its evolving interaction experience.

Let $Q_k(h \mid x)$ denote the hint distribution induced by sampling source trajectories from the round-k solver and applying q_{ϕ_k}. Using fresh randomness independent of the source trajectories, define

$$v_k^h(x) = \mathbb{E}_{h\sim Q_k(\cdot|x),\,\omega}[J_x(\tau_k^h(\omega))], \qquad \mathcal{C}_k^h(\alpha) = \{x : v_k^h(x) \geq \alpha\}. \tag{4}$$

Here τ_k^h uses the same solver with h; reachability is protocol-dependent, and hints need not help.

The *hint-exposed region*

$$\mathcal{B}_k(\alpha) = \mathcal{C}_k^h(\alpha) \setminus \mathcal{C}_k^0(\alpha) \tag{5}$$

contains tasks reliably solved with hints but not without them. For matched rollouts, let $\Delta_k(h; x, \omega) = J_x(\tau_k^h(\omega)) - J_x(\tau_k^0(\omega))$ with mean under fresh evaluation, $\delta_k(h; x) = \mathbb{E}_\omega[\Delta_k(h; x, \omega)]$. As $\mathbb{E}_{h\sim Q_k}[\delta_k(h; x)] = v_k^h(x) - v_k^0(x)$, exposure requires average hint gains to bridge the gap to α for unmastered tasks. Trajectory gains inform frontier probing. For fixed h,

$$\delta_k(h; x) \leq 1 - v_k^0(x). \tag{6}$$

Higher unguided reliability limits potential success gains from hints. Adaptive probing couples these opportunities with fresh source trajectories and joint updates of hint generation and utilization.

3.3 The capability ratchet

We organize co-evolution as a *capability ratchet*: acquired and retained behaviors advance the starting point for subsequent experience extraction. Hints expose useful behaviors for policy learning; updated solver trajectories reshape the evidence for hint generation, and the evolving solver tests the resulting guidance. This feedback connects ongoing experience extraction to the agent's changing capabilities.

Let $M_k = \mathbb{P}_{x\sim\mathcal{D}}[x \in \mathcal{C}_k^0(\alpha)]$ denote reliable-task coverage. Partition acquisition into tasks inside the previously hint-exposed region, with probability mass ρ_k^{tr}, and outside it, with mass ρ_k^{leap}. Let ρ_k^{forget} be the mass of previously mastered tasks lost after updating. At a fixed reliability threshold,

$$M_{k+1} - M_k = \rho_k^{\mathrm{tr}} + \rho_k^{\mathrm{leap}} - \rho_k^{\mathrm{forget}}. \tag{7}$$

This balance links frontier transfer, acquisition beyond prior hint exposure, and retention to net expansion. Growth requires acquisition to outweigh forgetting. These quantities support the empirical task-level analysis in Section 5.3; their formal set definitions are provided in Appendix A.4.

After gains from joint learning stabilize at checkpoint K, verified internalization targets effective behaviors that still depend on hints. For each task with accepted successful targets, let $\mu_K^+(\tau \mid x)$ be their fixed distribution and $P_\theta^0(\tau \mid x)$ the unguided trajectory distribution under the same environment protocol. The unguided success probability $v_\theta^0(x) = \mathbb{P}_{\tau\sim P_\theta^0}[J_x(\tau) = 1]$ satisfies

$$v_\theta^0(x) \geq \exp\left[-D_{\mathrm{KL}}\left(\mu_K^+(\cdot \mid x) \,\|\, P_\theta^0(\cdot \mid x)\right)\right]. \tag{8}$$

Divergence at most $-\log\alpha$ suffices for an exposed task to enter the unguided region. This idealized condition links successful-target fitting to frontier transfer. HELICES learns verified trajectories with full action and selective reasoning supervision; fresh unguided evaluation assesses acquisition. Its selective loss differs from full-trajectory divergence (Appendix A.2). Appendix B provides conditional convergence and net expansion bounds. Efficiency gains are assessed separately.

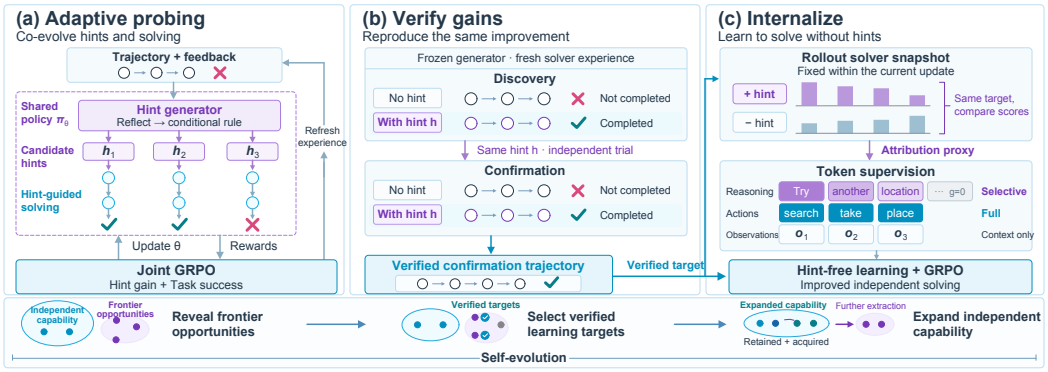


Figure 2: **HELICES overview.** (a) Joint learning of hint generation and task solving. (b) Independent confirmation of gains from the same hint. (c) Hint-free internalization with full action and selective reasoning supervision, plus GRPO. Regions, bars, and token weights are schematic.

4 HELICES: Adaptive Probing and Verified Internalization

HELICES couples adaptive probing with verified internalization to expand the agent's independent capabilities (Figure 2). During joint learning, a shared policy learns to generate useful hints from current experience and use them for task solving (a). Updated solver trajectories guide subsequent hint generation, aligning experience generation with the evolving policy. Verified internalization consolidates hint-dependent gains: independent confirmation selects reproducible improvements (b), and full action and selective reasoning supervision transfer them into hint-free execution (c). The resulting gains support further experience extraction, closing the self-evolution loop.

4.1 Adaptive Probing of the Capability Frontier

HELICES uses hints to discover effective behaviors the current policy does not yet reliably produce. These opportunities lie in the hint-exposed region $\mathcal{B}_k(\alpha)$ defined in Section 3: tasks reliably solvable with hints but not yet without them. Trajectory-level improvements provide probing signals.

A hint is a concise conditional decision rule grounded in a source trajectory and its feedback. The generator abstracts the observed difficulty into an observable applicability condition and useful response, retaining decision-relevant information but omitting incidental details and a complete solution. Re-execution tests whether the solver can apply the rule to activate effective behavior.

During joint learning, θ_k denotes shared solver and generator parameters at update k, so $\phi_k = \theta_k$ in Section 3. For task x_i, HELICES samples an unguided source trajectory and G hint candidates conditioned on that trajectory and its feedback; each protocol-valid candidate is followed by a fresh execution of the same task:

$$\tau_i^0 \sim \pi_{\theta_k}^0(\cdot \mid x_i), \qquad h_{ij} \sim q_{\theta_k}(\cdot \mid x_i, \tau_i^0, f_i), \qquad \tau_{ij}^h \sim \pi_{\theta_k}^h(\cdot \mid x_i, h_{ij}), \tag{9}$$

where f_i is the source feedback and $j = 1, \ldots, G$. Candidates from the same source explore alternative decision rules whose executions test whether the solver can use them effectively. Protocol-invalid generator outputs bypass solver execution; their treatment is detailed in Appendix C.1.

4.2 Co-evolution of Hint Generation and Utilization

Executions in Equation 9 provide feedback for learning both hint generation and utilization. Define the observed success improvement and role-specific advantages for the generator and solver as

$$d_{ij} = J_{x_i}(\tau_{ij}^h) - J_{x_i}(\tau_i^0), \qquad A_{ij}^q = \mathcal{N}_j(r_{ij}^{\text{probe}}), \qquad A_{ij}^h = \mathcal{N}_j\big(J_{x_i}(\tau_{ij}^h)\big), \tag{10}$$

where $\mathcal{N}_j$ centers and scales rewards within the candidates for one source. The probing reward r_{ij}^{probe} is driven by d_{ij}. When execution cost admits a meaningful efficiency comparison, it can additionally reward cost reductions conditional on both executions succeeding; this term is omitted for fixed-cost tasks. The two roles are optimized together through the shared-parameter objective

$$\mathcal{L}_{\text{joint}} = \lambda_q \mathcal{L}_{\text{GRPO}}^q + \lambda_h \mathcal{L}_{\text{GRPO}}^h + \mathcal{R}_{\text{joint}}, \tag{11}$$

where the GRPO terms use the corresponding role contexts and advantages, and $\mathcal{R}_{\text{joint}}$ contains role-specific regularization. The generator learns to produce useful interventions, while the solver learns successful execution. Successful guided trajectories become policy-learning experience within the evolving shared model, providing a route for these behaviors to improve unguided execution.

These updates also refresh the evidence for subsequent probing. New unguided trajectories expose the solver's current difficulties; the generator proposes hints from this evidence, and re-execution tests their usefulness to the updated solver. This feedback keeps probing aligned with the evolving policy. The common source-success baseline cancels under group centering, so normalization compares same-source hints; adaptation comes from this feedback rather than explicit task prioritization.

4.3 Verification of residual hint gains

Joint learning may leave useful behaviors hint-dependent. At each internalization transition, HELICES freezes a generator copy to read fresh solver experience and continues updating the solver from the same joint checkpoint. Only independently confirmed gains become hint-free targets.

Matched unguided and hinted discovery trajectories identify a candidate improvement, primarily failure without the hint and success with it. Where efficiency is meaningful, successful executions meeting a prespecified cost-reduction threshold can also qualify. To test whether the improvement reproduces beyond discovery, independent confirmation evaluates the same hint on a fresh realization of the task, against that realization's own unguided baseline. Let c_i^{disc} and c_i^{conf} denote the qualifying improvement in the two comparisons, with $\varnothing$ indicating none. Acceptance requires

$$a_i = \mathbb{I}\big[c_i^{\text{disc}} = c_i^{\text{conf}} \neq \varnothing\big]. \tag{12}$$

For each accepted candidate, the improved *confirmation* trajectory y_i^+ becomes a target paired with its original task context and history, with the hint removed. This transfers a reproduced improvement under unguided conditioning; fresh unguided evaluation measures task-level capability acquisition.

4.4 Hint-attribution-guided internalization

Verification selects trajectories; attribution determines how their reasoning is learned. Uniform distillation would reinforce incidental reasoning, while action-only supervision could omit useful decision context. HELICES fully supervises actions and selectively weights hint-supported reasoning.

Let $\bar{\pi}_k$ be the solver snapshot used to collect targets. For policy-generated token y_{it}^+, $\mathcal{H}_{it}^0$ includes preceding target tokens and observations with all explicit hint conditioning removed. Define

$$\Delta_{it} = \log \bar{\pi}_k(y_{it}^+ \mid x_i, \mathcal{H}_{it}^0, h_i) - \log \bar{\pi}_k(y_{it}^+ \mid x_i, \mathcal{H}_{it}^0). \tag{13}$$

For a fixed target token and history, the likelihood contrast measures hint support as an attribution proxy. Each round refreshes the snapshot; both scores and the gate remain fixed during the update.

Let $\mathcal{T}_i^{\text{act}}$ and $\mathcal{T}_i^{\text{rea}}$ partition policy-generated tokens into actions and reasoning. The gate is

$$g_{it} = \begin{cases} 1, & t \in \mathcal{T}_i^{\text{act}}, \\ \max\{2\sigma(\gamma(\Delta_{it} - m)) - 1, 0\}, & t \in \mathcal{T}_i^{\text{rea}}, \end{cases} \tag{14}$$

where σ is the sigmoid, $m > 0$ the margin, and $\gamma > 0$ controls sharpness. Set $b_{it} = 1$ for actions and lower base weights for reasoning, supervised only if $\Delta_{it} > m$. Observations are context only.

For a nonempty accepted set $\mathcal{A}$, the internalization objective under hint-free conditioning is

$$\mathcal{L}_{\text{int}} = -\frac{1}{\sum_{i\in\mathcal{A}} w_i} \sum_{i\in\mathcal{A}} w_i \operatorname*{Mean}_{t\in\mathcal{T}_i^{\text{act}}\cup\mathcal{T}_i^{\text{rea}}} \big[b_{it} g_{it} \log \pi_\theta(y_{it}^+ \mid x_i, \mathcal{H}_{it}^0)\big], \tag{15}$$

where w_i gives success gains higher weight when efficiency refinements are retained. The solver combines the independently normalized objectives of unguided GRPO and verified internalization:

$$\mathcal{L}_{\text{solver}} = \mathcal{L}_{\text{GRPO}}^0 + \beta_0 \mathcal{L}_{\text{KL}}^0 + \lambda_{\text{int}} \mathcal{L}_{\text{int}}. \tag{16}$$

Here β_0 controls source KL regularization and λ_{int} the relative weight of internalization against unguided GRPO. The selective loss differs from the full-trajectory KL projection in Section 3 (Appendix A.2). Full implementation settings and training details are provided in Appendices C.1–D.1.

Table 1: ALFWorld success rate (SR) and Search-QA exact match (EM), in %, grouped by Qwen2.5 scale. Bold marks per-scale best results (ties included); see Appendix D.3 for provenance details.

	ALFWorld							Single-hop QA				Multi-hop QA				
Method	Pick	Look	Clean	Heat	Cool	Pick2	**SR**	NQ	Triv	Pop	**Avg.**	Hotp	2Wk	MuS	Bam	**Avg.**
Qwen2.5-3B family																
Vanilla	25.7	23.1	3.7	18.8	12.0	0.0	13.6	12.3	30.6	5.6	16.1	16.0	19.2	4.5	18.2	16.6
Skill-Prompt	51.4	69.2	48.1	0.0	4.0	8.3	30.7	23.7	46.2	30.6	35.8	24.4	22.1	7.3	9.1	21.2
Few-Shot (text)	42.9	46.2	29.6	43.8	8.0	4.2	27.9	11.9	30.9	20.2	23.3	13.7	18.4	4.5	27.3	15.4
PPO	82.9	76.9	88.9	75.0	60.0	45.8	72.1	34.0	54.5	37.8	43.8	32.4	31.9	10.5	27.3	29.7
GRPO	91.4	84.6	70.4	87.5	80.0	66.7	80.0	39.1	**60.6**	41.1	48.4	37.4	34.6	15.5	27.3	33.4
EvolveR	77.1	23.1	48.1	43.8	24.0	20.8	43.6	**43.5**	58.4	43.4	49.2	37.3	**38.1**	13.6	36.4	35.2
Skill-GRPO	88.6	69.2	59.3	68.8	40.0	29.2	60.0	**43.5**	58.8	43.0	49.2	36.8	32.2	11.8	18.2	31.4
SkillRL	91.4	**100.0**	81.5	87.5	80.0	70.8	84.3	38.7	57.6	40.3	46.8	33.6	31.1	13.2	54.5	30.1
AgentOCR	91.4	84.6	77.8	75.0	76.0	70.8	80.0	38.7	56.5	41.7	47.1	33.6	30.7	14.5	27.3	29.9
Skill0	94.3	76.9	**100.0**	87.5	80.0	75.0	87.1	39.9	57.6	42.3	47.9	35.1	33.7	13.2	**63.6**	32.1
Evolving-RL	54.3	**100.0**	**100.0**	**100.0**	**88.0**	54.2	78.6	36.0	57.4	41.5	47.0	35.8	37.3	13.6	36.4	34.3
HELICES	**97.1**	92.3	**100.0**	87.5	**88.0**	**95.8**	**94.3**	43.1	59.8	**45.7**	**50.8**	**37.6**	37.4	**18.6**	54.5	**35.5**
Qwen2.5-7B family																
Vanilla	34.3	23.1	18.5	6.3	4.0	4.2	16.4	11.5	35.6	1.2	15.8	16.3	22.2	5.0	18.2	18.4
Skill-Prompt	51.4	46.2	33.3	6.3	4.0	0.0	25.0	30.8	52.1	32.7	40.0	32.7	27.9	12.7	**63.6**	28.0
Few-Shot (text)	74.3	61.5	66.7	25.0	20.0	8.3	45.0	12.3	36.8	24.5	27.8	17.7	18.2	6.4	27.3	16.8
PPO	91.4	61.5	92.6	87.5	80.0	70.8	82.9	39.1	61.0	39.7	47.9	37.0	41.4	14.5	36.4	37.0
GRPO	91.4	84.6	96.3	81.3	64.0	58.3	80.0	45.1	63.7	44.0	51.8	43.5	43.2	16.8	36.4	40.4
EvolveR	65.7	30.8	48.1	12.5	32.0	33.3	41.4	43.5	63.4	44.6	51.8	38.2	42.0	15.5	54.5	38.0
Skill-GRPO	88.6	69.2	66.7	62.5	56.0	75.0	71.4	45.1	63.7	45.7	52.6	43.1	43.3	19.5	27.3	40.6
SkillRL	97.1	69.2	88.9	87.5	96.0	87.5	90.0	45.8	61.2	45.9	51.8	43.8	39.9	17.3	45.5	38.8
AgentOCR	94.3	**100.0**	77.8	75.0	72.0	70.8	81.4	43.1	61.0	45.4	51.2	40.9	38.3	15.9	36.4	36.7
Skill0	**100.0**	84.6	96.3	81.3	84.0	79.2	89.3	42.7	61.1	45.3	51.1	40.0	38.3	16.4	**63.6**	36.6
SAPO	**100.0**	76.9	96.3	93.8	84.0	87.5	91.4	**48.2**	62.8	**46.7**	53.1	45.0	**45.2**	18.2	45.5	42.2
Evolving-RL	94.3	**100.0**	96.3	**100.0**	**100.0**	62.5	91.4	42.3	64.7	45.0	52.3	37.6	35.5	14.1	**63.6**	34.0
HELICES	**100.0**	**100.0**	**100.0**	**100.0**	92.0	**91.7**	**97.1**	47.4	**64.9**	46.6	**53.8**	**45.3**	44.8	**21.8**	**63.6**	**42.6**

5 EXPERIMENTS

We evaluate hint-free performance across three agent benchmarks, then examine how joint training, verified internalization, and repeated cycles contribute to independent capability acquisition.

5.1 EVALUATION SETUP AND MAIN RESULTS

Benchmarks and metrics. ALFWorld (Shridhar et al., 2021) measures success across six household-task families. Search-R1 (Jin et al., 2025) reports exact match on three single-hop and four multi-hop QA datasets. Mind2Web (Deng et al., 2023) measures action accuracy (AA) and full-task success (SR) on cross-task, cross-website, and cross-domain splits (Fan et al., 2026).

Baselines. We compare prompting (Vanilla, Skill-Prompt (Wu et al., 2026b), Few-Shot (Lu et al., 2026b)), and policy optimization (PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), Skill-GRPO (Wu et al., 2026b)). Experience-based baselines include EvolveR (Wu et al., 2025), SkillRL (Xia et al., 2026), Skill0 (Lu et al., 2026b), SAPO (Zhang et al., 2026c), Evolving-RL (Fan et al., 2026), AgentOCR (Feng et al., 2026), and the Mind2Web memory methods Reasoning-Bank (Ouyang et al., 2025) and Memento (Zhou et al., 2025a) for experience reuse.

Training and evaluation. We use Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct with 120 joint-training updates followed by 30 verified-internalization updates, matching the benchmark-specific baselines' policy-update Solver-trajectory budgets (Wu et al., 2026b; Zhang et al., 2026c; Lu et al., 2026b; Fan et al., 2026). Appendix D.1 separates this matched update budget from the additional environment interactions used for gain discovery and independent confirmation. Final Solver evaluation is greedy and hint-free; protocols, ablations, and cycle analyses appear in Appendices D–G.

Experience becomes independent capability. HELICES leads the reported ALFWorld and QA aggregate scores at both scales without test-time hints (Table 1), with its largest margins over the strongest baselines on ALFWorld: 7.2/5.7 percentage points (pp) for 3B/7B. Consistent gains across household interaction and retrieval-based reasoning show that internalized experience supports independent solving across environments with different actions and feedback.

Table 2: Mind2Web action accuracy (AA) and full-task success (SR), in %, on cross-task, cross-website, and cross-domain splits. Bold marks the best results per model scale, including ties.

	Cross-task		Cross-website		Cross-domain		Avg.	
Method	AA	SR	AA	SR	AA	SR	**AA**	**SR**
Qwen2.5-3B-Instruct								
Base Model	10.98	0.00	6.84	**0.00**	7.82	0.11	8.28	0.07
ReasoningBank	7.98	0.00	7.85	**0.00**	6.49	0.11	6.95	0.07
Memento	11.34	0.00	4.32	**0.00**	8.85	0.33	8.72	0.22
GRPO	19.63	0.40	17.67	**0.00**	14.56	0.99	15.92	0.75
Evolving-RL	19.62	**0.79**	**19.63**	**0.00**	13.58	0.44	15.51	0.45
HELICES	**21.17**	0.40	17.84	**0.00**	**17.13**	**1.10**	**17.98**	**0.82**
Qwen2.5-7B-Instruct								
Base Model	12.57	0.40	7.95	0.00	7.91	0.44	8.79	0.37
ReasoningBank	11.84	0.00	17.76	0.56	12.09	0.66	12.79	0.52
Memento	15.27	0.40	15.46	0.56	13.13	0.88	13.84	0.75
GRPO	28.79	**1.59**	24.61	0.00	20.84	1.64	22.83	1.41
Evolving-RL	35.32	1.59	35.15	**3.39**	24.67	1.21	28.05	1.57
HELICES	**35.33**	**1.98**	**38.32**	2.82	**27.83**	**2.08**	**30.62**	**2.16**

Generalization to new websites and domains. On Mind2Web, HELICES achieves the best average action accuracy and task success at both scales, including 7B gains on cross-website and cross-domain splits (Table 2), supporting hint-free transfer of decision rules beyond their original context.

5.2 Joint learning and verified internalization

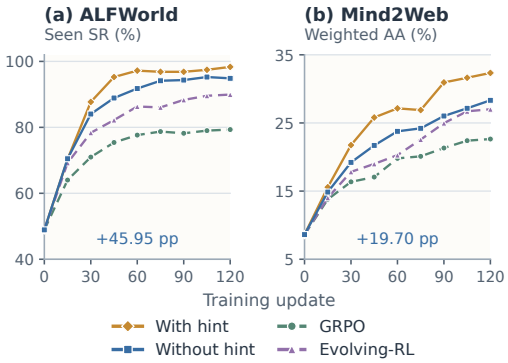


Figure 3: First-stage joint-training curves.

Method	120	135	150	Δ
ALFWorld Seen SR (%)				
HELICES	94.86	96.23	97.14	+2.28
GRPO	79.33	80.33	80.30	+0.97
Evolving-RL	89.94	90.94	91.20	+1.26
Mind2Web weighted AA (%)				
HELICES	28.32	29.77	30.62	+2.30
GRPO	22.64	22.84	22.83	+0.19
Evolving-RL	27.01	27.34	28.05	+1.04

Columns show training updates; Δ: 150 minus 120 (pp).

Table 3: Second-stage hint-free means over three continuation seeds.

Learning dynamics. Guided and unguided curves rise together, linking hint use to independent improvement (Figure 3); their remaining gap marks headroom as joint ALFWorld gains slow. From a stronger starting point, later updates still yield larger unguided gains than either continuation baseline (Table 3), supporting verified internalization of behaviors exposed during joint learning.

Table 4: ALFWorld ablations: task-weighted, hint-free success (%), mean ± SD over three continuation-training seeds. Variants share a joint-training checkpoint and trajectory budget.

	Qwen2.5-3B-Instruct		Qwen2.5-7B-Instruct	
Training variant	Seen SR	Unseen SR	Seen SR	Unseen SR
Continue GRPO	88.57 ± 0.82	78.36 ± 1.00	94.13 ± 0.55	93.78 ± 0.75
GRPO + success distillation	92.14 ± 0.82	76.87 ± 1.51	94.60 ± 0.69	91.54 ± 1.08
HELICES w/o independent confirmation	91.43 ± 0.63	82.84 ± 0.75	95.71 ± 0.48	94.03 ± 0.43
HELICES w/o reasoning gate	93.57 ± 0.63	84.33 ± 1.00	95.79 ± 0.50	95.52 ± 0.50
HELICES	94.29 ± 0.41	84.58 ± 0.66	97.14 ± 0.24	96.27 ± 0.50

Verified internalization improves generalization. Table 4 distinguishes how successful trajectories affect transfer: ordinary success distillation improves Seen but reduces Unseen success relative

to continued GRPO, whereas HELICES improves both from the same checkpoint and budget. Internalization therefore depends on selecting reproducible gains and controlling the learning signal, not merely collecting successful trajectories.

Reliable targets and focused supervision. Confirmation yields larger Unseen gains at both scales, and gating improves mean performance over ungated supervision. These results support reproducible targets, selective reinforcement of hint-supported reasoning, and full action supervision.

5.3 Capability Expansion Across Repeated Cycles

We alternate six ALFWorld cycles of 20 joint and 5 internalization updates. A task is empirically mastered when all three evaluations succeed; coverage measures the proportion of mastered tasks.

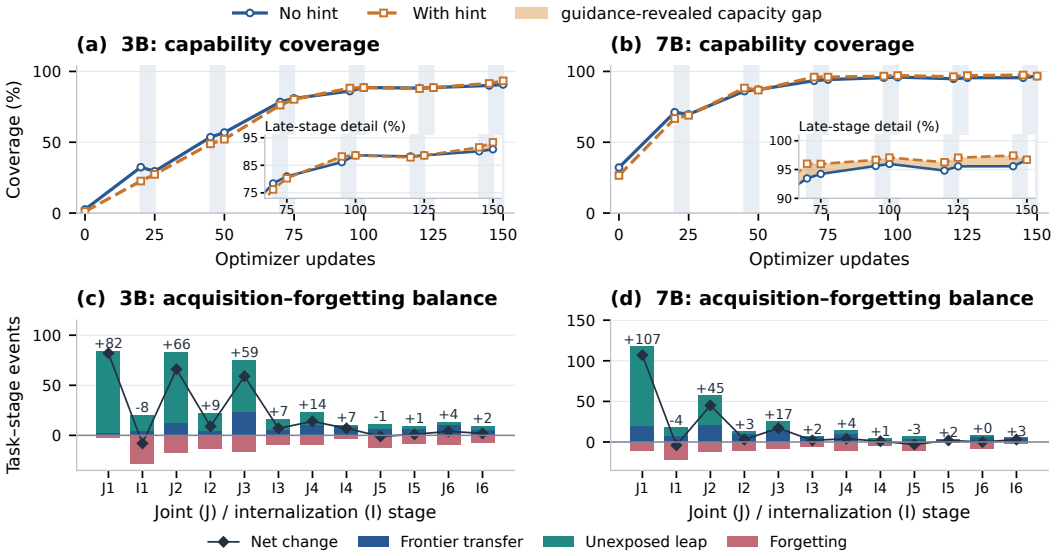


Figure 4: **Repeated cycles expand independent capability.** Top: Seen/Unseen-balanced 3/3 coverage; blue bands mark internalization and orange insets show the positive guided–unguided coverage gap. Bottom: task counts of frontier transfer, unexposed leap, forgetting, and net gains (diamonds).

Repeated cycles expand independent capability. Figure 4 tracks the expansion of reliably solved tasks and connects the results to Section 3. Across six cycles, joint learning and internalization both contribute positive cumulative net acquisition after forgetting. Each extraction–consolidation cycle thus builds on prior capabilities and expands the frontier while preserving gains from earlier stages.

Hint exposure predicts subsequent acquisition. Previously hint-only tasks have more than twice the subsequent acquisition rate of other unmastered tasks during internalization at both scales, with the same ordering during joint learning. Hint exposure therefore identifies productive opportunities for independent acquisition along the evolving capability frontier in later stages.

Retention supports cumulative capability growth. Most tasks acquired during joint learning remain mastered after the next internalization, with more retained at the final checkpoint. Together with net acquisition, this supports a capability ratchet in which learned behaviors persist for further experience extraction and capability growth throughout later rounds of adaptive self-evolution.

6 Conclusion

We introduced HELICES, a self-evolution method that couples experience extraction and internalization around a dynamic capability frontier. A shared model learns to generate and use hints as its capabilities evolve, while independent confirmation and hint attribution transfer verified gains through full action and selective reasoning supervision. Across three agent benchmarks and two model scales, HELICES achieves leading aggregate performance and generalizes to unseen tasks and environments without test-time hints. Repeated cycles further expand reliable task coverage, converting adaptively extracted and verified experience into cumulative independent capability.

AI USE STATEMENT

Generative AI tools were used only to assist with manuscript writing and language polishing, and with the creation and refinement of scientific figures. All AI-assisted text and visual materials were carefully reviewed by the authors for accuracy and consistency with the underlying methods and results. Generative AI tools were not used for any other part of the research, including research ideation, theoretical or conceptual development, mathematical claims or proofs, methodology or experiment design, method implementation, data generation, cleaning, processing, or analysis, or result interpretation. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human participants, personally identifiable or sensitive data, private user information, or deployment in real-world systems. All experiments are conducted on established public research benchmarks for methodological evaluation. We therefore identify no ethical issues specific to this study.

REPRODUCIBILITY STATEMENT

This work is reproducible from the complete methodological, training, and evaluation specifications provided in the paper and appendices. Appendices A and B give the formal definitions, diagnostics, and theoretical analysis. Appendix C specifies the hint representation, optimization objectives, independent-confirmation procedure, and target-weighting rules. Appendix D documents training schedules, sampling and environment budgets, hyperparameters, benchmark splits and metrics, evaluation manifests, inference settings, and baseline provenance. Appendices D.5 and D.7 report independent full-pipeline multi-seed results, while Appendices F and G provide controlled cross-evaluations, end-to-end controls, repeated-cycle schedules, and complete capability-frontier analyses. Together, these materials provide the objectives, procedures, hyperparameters, data splits, task manifests, evaluation rules, random-seed repetitions, and result-provenance information needed to reproduce HELICES training, evaluation, main results, ablations, and capability-frontier analyses.

REFERENCES

Emre Can Acikgoz, Cheng Qian, Heng Ji, Dilek Hakkani-Tür, and Gokhan Tur. TT-SI: Self-improving LLM agents with test-time training. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. doi: 10.18653/v1/2026.findings-acl.462. URL https://aclanthology.org/2026.findings-acl.462/.

Minghao Chen, Yihang Li, Yanting Yang, Shiyu Yu, Binbin Lin, and Xiaofei He. AutoManual: Constructing instruction manuals by LLM agents via interactive environmental learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0019.

Xuanzhong Chen, Zile Qiao, Guoxin Chen, Liangcai Su, Zhen Zhang, Xinyu Wang, Pengjun Xie, Fei Huang, Jingren Zhou, Yong Jiang, and Ting Chen. Expanding the capability frontier of LLM agents with ZPD-guided data synthesis. In *International Conference on Learning Representations*, pp. 114473–114505, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ba79fb5c4fe70050752f20c90c5f07ca-Paper-Conference.pdf.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.06070.

Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. Re-ReST: Reflection-reinforced self-training for language agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15394–15411, 2024. doi: 10.18653/v1/2024.emnlp-main.861.

Shengda Fan, Xin Cong, Zhong Zhang, Yuepeng Fu, Yesai Wu, Hao Wang, Xinyu Zhang, Enrui Hu, and Yankai Lin. Generalizing experience for language agents with hierarchical MetaFlows. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2149.

Zhiyuan Fan, Wenwei Jin, Feng Zhang, Bin Li, Yihong Dong, Yao Hu, and Jiawei Li. Evolving-RL: End-to-end optimization of experience-driven self-evolving capability within agents. *arXiv preprint arXiv:2605.10663*, 2026. URL https://arxiv.org/abs/2605.10663.

Lang Feng, Fuchao Yang, Feng Chen, Xin Cheng, Haiyang Xu, Zhenglin Wan, Ming Yan, and Bo An. AgentOCR: Reimagining agent history via optical self-compression. *arXiv preprint arXiv:2601.04786*, 2026. URL https://arxiv.org/abs/2601.04786.

Xueyang Feng, Bo Lan, Quanyu Dai, Lei Wang, Jiakai Tang, Xu Chen, Zhenhua Dong, and Ji-Rong Wen. Improving retrospective language agents via joint policy gradient optimization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 112–141, 2025. doi: 10.18653/v1/2025.naacl-long.6.

Yubin Ge, Salvatore Romeo, Jason Cai, Monica Sunkara, and Yi Zhang. SAMULE: Self-learning agents enhanced by multi-level reflection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16591–16610, 2025. doi: 10.18653/v1/2025.emnlp-main.839.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. URL https://arxiv.org/abs/1309.5549.

Priyanshu Gupta, Shashank Kirtania, Ananya Singha, Sumit Gulwani, Arjun Radhakrishna, Gustavo Soares, and Sherry Shi. Meta-reflection: Learning instructions for language agents using past reflections. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8369–8385, 2024. doi: 10.18653/v1/2024.emnlp-main.477.

Xinmeng Hou, Bohao Qu, Wuqi Wang, Peiliang Gong, Qing Guo, and Yang Liu. Learn like humans: Use meta-cognitive reflection for efficient self-improvement. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28802–28824, 2026. doi: 10.18653/v1/2026.acl-long.1329.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025. URL https://arxiv.org/abs/2503.09516.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak–Łojasiewicz condition. *arXiv preprint arXiv:1608.04636*, 2016. URL https://arxiv.org/abs/1608.04636.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=CjwERcAU7w.

Kun Li, Tingzhang Zhao, Wei Zhou, and Songlin Hu. DORA: Dynamic optimization prompt for continuous reflection of LLM-based agent. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 7546–7557, 2025. URL https://aclanthology.org/2025.coling-main.504/.

Yitao Liu, Chenglei Si, Karthik R. Narasimhan, and Shunyu Yao. Contextual experience replay for self-improvement of language agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14179–14198, 2025. doi: 10.18653/v1/2025.acl-long.694.

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

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. SKILL0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*, 2026b. URL https://arxiv.org/abs/2604.02268.

Kolby Nottingham, Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Sameer Singh, Peter Clark, and Roy Fox. Skill set optimization: Reinforcing language model behavior via transferable skills. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38409–38425. PMLR, 2024. URL https://proceedings.mlr.press/v235/nottingham24a.html.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*, 2025. URL https://arxiv.org/abs/2509.25140.

Deng Qiyuan, Kehai Chen, Min Zhang, and Zhongwen Xu. HiPO: Self-hint policy optimization for RLVR. In *International Conference on Learning Representations*, pp. 7100–7117, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0c0936a505992ce0d45169cd595b9755-Paper-Conference.pdf.

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

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0377.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Yifan Song, Weimin Xiong, Xiutian Zhao, Dawei Zhu, Wenhao Wu, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. AgentBank: Towards generalized LLM agents via fine-tuning on 50000+ interaction trajectories. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2124–2141, 2024. doi: 10.18653/v1/2024.findings-emnlp.116.

Junhao Su, Yuanliang Wan, Junwei Yang, Hengyu Shi, Tianyang Han, Yurui Qiu, and Junfeng Luo. Failure makes the agent stronger: Enhancing accuracy through structured reflection for reliable tool interactions. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12712–12734, 2026. doi: 10.18653/v1/2026.findings-acl.618.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-SD: Skill-conditioned self-distillation for multi-turn LLM agents. *arXiv preprint arXiv:2604.10674*, 2026a. URL https://arxiv.org/abs/2604.10674.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. Reinforcement learning for self-improving

agent with skill library. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1529–1550, 2026b. doi: 10.18653/v1/2026.acl-long.69.

Yaoke Wang, Yun Zhu, Xintong Bao, Wenqiao Zhang, Suyang Dai, Kehan Chen, Wenqiang Li, Gang Huang, Siliang Tang, and Yueting Zhuang. Meta-reflection: A feedback-free reflection learning framework. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3958–3976, 2025. doi: 10.18653/v1/2025.acl-long.201.

Haiyan Wu, Chenchen Wang, Chaoqun Sun, Chengxiong Lu, Yan-Hong Chen, Zhiqiang Zhang, and Xiaoqing Feng. Reflective RAG: Self-evaluation driven strategy optimization in agentic retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13252–13267, 2026a. doi: 10.18653/v1/2026.findings-acl.648.

Jinyang Wu, Shuo Yang, Zhengxi Lu, Fan Zhang, Yuhao Shen, Lang Feng, Haoran Luo, Zheng Lian, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. SEED: Self-evolving on-policy distillation for agentic reinforcement learning. *arXiv preprint arXiv:2607.14777*, 2026b. URL https://arxiv.org/abs/2607.14777.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, and Botian Shi. EvolveR: Self-evolving LLM agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025. URL https://arxiv.org/abs/2510.16079.

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*, 2026. URL https://arxiv.org/abs/2602.08234.

Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. GPT4Tools: Teaching large language model to use tools via self-instruction. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3149.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Xiao Yu, Baolin Peng, Vineeth Vajipey, Hao Cheng, Michel Galley, Jianfeng Gao, and Zhou Yu. ExACT: Teaching AI agents to explore with reflective-MCTS and exploratory learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a3b893ba1de12f76020b03f7ae8e1afd-Abstract-Conference.html.

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

Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. Agent-Pro: Learning to evolve via policy-level reflection and optimization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5348–5375, 2024. doi: 10.18653/v1/2024.acl-long.292.

Wenyuan Zhang, Xinghua Zhang, Haiyang Yu, Shuaiyi Nie, Bingli Wu, Juwei Yue, Tingwen Liu, and Yongbin Li. ExpSeek: Self-triggered experience seeking for web agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 29255–29283, 2026a. doi: 10.18653/v1/2026.findings-acl.1462.

Yunjian Zhang, Sudong Wang, Yang Li, Peiran Xu, Conghao Zhou, Xiaoyue Ma, Jianing Li, and Yao Zhu. Resource-efficient reinforcement for reasoning large language models via dynamic one-shot policy refinement. *arXiv preprint arXiv:2602.00815*, 2026b. doi: 10.48550/arXiv.2602.00815. URL https://arxiv.org/abs/2602.00815.

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*, 2026c. 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. doi: 10.1609/aaai.v38i17.29936.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026. URL https://arxiv.org/abs/2601.18734.

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and Jun Wang. Memento: Fine-tuning LLM agents without fine-tuning LLMs. *arXiv preprint arXiv:2508.16153v2*, 2025a. URL https://arxiv.org/abs/2508.16153v2.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

Yifei Zhou, Sergey Levine, Jason E. Weston, Xian Li, and Sainbayar Sukhbaatar. Self-challenging language model agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-3801.

A Formal Details and Capability Diagnostics

A.1 Source-conditioned hints and independent evaluation

Let $z = (\tau^{\mathrm{src}}, f^{\mathrm{src}})$ denote a source trajectory and its feedback, with $z \sim P_k^{\mathrm{src}}(\cdot \mid x)$. The induced hint distribution used in Section 3 is

$$Q_k(h \mid x) = \mathbb{E}_{z \sim P_k^{\mathrm{src}}(\cdot|x)}[q_{\phi_k}(h \mid x, z)]. \tag{17}$$

The definition of v_k^h averages over this source-generation process and independent evaluation randomness. It therefore differs from evaluating a selected hint only on the same random realization that produced its source trajectory.

The hint-reachable region is relative to the source-feedback, hint-generation, and execution protocol, including its information and interaction budget; it is not an intrinsic upper bound on the solver. No inclusion $\mathcal{C}_k^0 \subseteq \mathcal{C}_k^h$ is assumed, since hints may be ineffective or harmful. The role-specific parameters θ_k and ϕ_k may be coupled; HELICES shares them during joint learning.

Discovery can reuse the source randomness that informed hint selection, making its observed gain potentially optimistic. Independent confirmation evaluates the same hint on a held-out realization whose unguided trajectory was not used to generate that hint. The regions in Section 3 describe learning goals; verified internalization uses confirmed trajectory-level improvements as practical candidate signals.

For a fixed task–hint pair, let $\omega_1, \ldots, \omega_M$ be identically distributed, independent evaluation draws, independent of the source used to generate the hint. Each draw couples guided and unguided rollouts while preserving their prescribed marginal evaluation distributions. Define

$$\delta_k(h; x) = \mathbb{E}_\omega[\Delta_k(h; x, \omega)], \qquad \widehat{\delta}_{k,M}(h; x) = \frac{1}{M} \sum_{m=1}^{M} \Delta_k(h; x, \omega_m). \tag{18}$$

Conditional on the fixed task and hint, the estimator is unbiased. Writing $J^h = J_x(\tau_k^h)$ and $J^0 = J_x(\tau_k^0)$, independence across evaluation pairs gives

$$\mathrm{Var}(\widehat{\delta}_{k,M}) = \frac{1}{M} \left[\mathrm{Var}(J^h) + \mathrm{Var}(J^0) - 2\,\mathrm{Cov}(J^h, J^0)\right]. \tag{19}$$

Compared with independent guided and unguided sampling with the same marginals, pairing reduces variance when the covariance is positive. Reusing a seed alone does not guarantee that condition. These statements apply to fresh evaluation of a fixed hint, not automatically to the discovery gain used to select it or to estimates filtered by successful confirmation. An audit of accepted hints requires further evaluation draws independent of that selection.

Fresh-seed confirmation audit. We freeze all task–hint pairs before the audit and evaluate each pair on three additional paired seeds that are disjoint from source generation, discovery, and the confirmation used for target selection. Guided and unguided executions under each seed share the prescribed task initialization and differ only in hint conditioning. The discovery-qualified set contains all candidates satisfying the discovery improvement criterion; the HELICES-confirmed and confirmation-rejected sets partition it according to the held-out confirmation decision.

Table 5 shows that confirmation selects hints whose gains persist more consistently under fresh evaluation. Relative to rejected candidates, confirmed candidates increase the paired gain from 7.8 to 20.4 points for 3B and from 4.5 to 16.1 points for 7B. Same-class reproduction rises from 41.4% to 58.3% and from 43.9% to 66.1%, respectively, while the corresponding harm rates fall from 17.5% to 6.5% and from 13.6% to 4.8%. Among confirmed candidates, 3B reproduces the original improvement class in 126 of 216 audit executions, with 127 guided successes; 7B reproduces it in 111 of 168 executions, with 112 guided successes.

For the same marginal evaluation laws, $\delta_k(h; x) = \mathbb{E}_\omega[J_x(\tau_k^h(\omega))] - v_k^0(x)$. Since the first term is at most one, Equation 6 follows. Increasing unguided reliability reduces this upper bound, but does not imply that every hint's actual gain decreases across updates. The bound motivates adapting probes to the current solver; it does not guarantee that optimization discovers a useful new hint.

Table 5: **Fresh-seed audit of hint-gain reproducibility on ALFWorld.** Each fixed task–hint pair is evaluated on three new paired seeds excluded from generation, discovery, and target-selection confirmation. SR_0 and SR_h pool all $N \times M$ unguided and guided executions. Δ is the mean guided–unguided difference across the three audit seeds, with sample SD. Reprod. is the percentage of all $N \times M$ audit pairs that reproduce the original F2S or efficiency improvement criterion; it is not conditioned on guided success. Harm is the fraction of pairs with $J^h < J^0$. Confirmed and rejected candidates partition the discovery-qualified set. SR, Reprod., and Harm entries are percentages.

Model	Candidate set	N	M	$N \times M$	SR_0	SR_h	Δ (pp)	Reprod.	Harm
3B	Discovery-qualified	192	3	576	42.0	54.5	$+12.5 \pm 2.3$	47.7	13.4
3B	**HELICES-confirmed**	72	3	216	38.4	58.8	$\mathbf{+20.4 \pm 1.6}$	**58.3**	**6.5**
3B	Confirmation-rejected	120	3	360	44.2	51.9	$+7.8 \pm 3.2$	41.4	17.5
7B	Discovery-qualified	144	3	432	55.1	64.1	$+9.0 \pm 2.1$	52.5	10.2
7B	**HELICES-confirmed**	56	3	168	50.6	66.7	$\mathbf{+16.1 \pm 1.8}$	**66.1**	**4.8**
7B	Confirmation-rejected	88	3	264	58.0	62.5	$+4.5 \pm 2.3$	43.9	13.6

A.2 Successful-target projection and unguided capability acquisition

Fix a task x with a nonempty set of accepted successful targets, and hold the target distribution $\mu_k^+(\cdot \mid x)$ fixed. The index k identifies the target-generating checkpoint; take $k = K$ for the stabilization checkpoint in Section 3. All distributions below are over complete trajectories of the same task and environment protocol. Let $A_x = \{\tau : J_x(\tau) = 1\}$, so that $\mu_k^+(A_x \mid x) = 1$, and define $v_\theta^0(x) = P_\theta^0(A_x \mid x)$. The idealized hint-free projection objective is

$$\mathcal{L}_{\mathrm{proj}}(\theta; x) = D_{\mathrm{KL}}\big(\mu_k^+(\cdot \mid x) \,\|\, P_\theta^0(\cdot \mid x)\big). \tag{20}$$

Assume finite divergence, using natural logarithms. This implies $v_\theta^0(x) > 0$. On the support of the target distribution,

$$P_\theta^0(\tau \mid x) = v_\theta^0(x) P_\theta^0(\tau \mid x, A_x). \tag{21}$$

Substituting this factorization into the divergence gives

$$\mathcal{L}_{\mathrm{proj}}(\theta; x) = D_{\mathrm{KL}}\big(\mu_k^+(\cdot \mid x) \,\|\, P_\theta^0(\cdot \mid x, A_x)\big) - \log v_\theta^0(x) \tag{22}$$

$$\geq -\log v_\theta^0(x). \tag{23}$$

Exponentiation yields Equation 8, with $k = K$. If the divergence is at most $-\log \alpha$, then $v_{k+1}^0(x) \geq \alpha$ for updated parameters $\theta = \theta_{k+1}$, giving the sufficient acquisition condition

$$x \in \mathcal{B}_k(\alpha), \quad \mathcal{L}_{\mathrm{proj}}(\theta_{k+1}; x) \leq -\log \alpha \quad \Longrightarrow \quad x \in \mathcal{B}_k(\alpha) \cap \mathcal{C}_{k+1}^0(\alpha). \tag{24}$$

The condition is sufficient rather than necessary: a solver can succeed through trajectories different from the selected targets.

For discrete finite-horizon trajectories, the unguided rollout law factorizes as

$$P_\theta^0(\tau \mid x) = K_x(\tau) \prod_{t \in \mathcal{T}^{\mathrm{pol}}(\tau)} \pi_\theta(y_t \mid x, \mathcal{H}_t), \tag{25}$$

where K_x contains the environment initialization and transition probabilities, $\mathcal{T}^{\mathrm{pol}}$ indexes all policy-generated tokens, and $\mathcal{H}_t$ is the preceding interaction history with the hint excluded. For fixed targets and environment dynamics, under the integrability conditions required for the following expectations,

$$\mathcal{L}_{\mathrm{proj}}(\theta; x) = \mathbb{E}_{\tau \sim \mu_k^+}\left[-\sum_{t \in \mathcal{T}^{\mathrm{pol}}(\tau)} \log \pi_\theta(y_t \mid x, \mathcal{H}_t) \right] + C_{k,x}, \tag{26}$$

where $C_{k,x}$ does not depend on θ. Thus, full-trajectory likelihood fitting minimizes the idealized divergence, although the numerical likelihood loss is not itself that divergence.

The bound describes a conditional distribution-level relationship, not a convergence or generalization guarantee for empirical training. In particular, successful confirmation defines the target support but does not imply a small projection divergence. HELICES uses sample weights, token normalization, selective reasoning supervision, and additional reinforcement-learning terms, so its actual loss

is not identical to Equation 20. A decrease in that surrogate cannot be substituted into the success bound; capability acquisition is evaluated through fresh unguided rollouts. The relationship explains why fitting successful behavior under a hint-free distribution can support capability acquisition without asserting that the selective objective necessarily achieves the sufficient condition.

A.3 Controlled comparisons of guided and unguided performance

Define aggregate guided performance and its signed advantage over unguided execution as

$$H_k = \mathbb{E}_x[v_k^h(x)], \qquad G_k = H_k - S_k. \tag{27}$$

The gap is a difference of mean success rates. Gains and harms across tasks can cancel, so it does not measure the size of $\mathcal{B}_k$.

To compare solvers under a fixed reference probe, retain the round-k source distribution and generator, hence Q_k, and define

$$H_{k+1|k} = \mathbb{E}_{x,\, h\sim Q_k(\cdot|x),\, \omega}[J_x(\tau_{k+1}^h(\omega))]. \tag{28}$$

A finite audit can implement this comparison by caching the same task–source–hint tuples and evaluating both solvers with the same distribution of fresh seeds, hint placement, decoding settings, and execution budgets. A frozen generator supplied with newly sampled solver trajectories defines a different comparison, because its induced hint distribution can change.

For a prespecified tolerance $\varepsilon_H \geq 0$, the conditions

$$S_{k+1} > S_k, \qquad H_{k+1|k} \geq H_k - \varepsilon_H \tag{29}$$

indicate improved average unguided success with limited loss of guided performance under the reference probe. They neither imply contraction of the guided–unguided gap nor establish an increase in the reliable-task region. For example, at $\alpha = 0.8$, changing two equally weighted task success probabilities from $(0.9, 0.1)$ to $(0.79, 0.79)$ improves their average while removing the only task previously above the threshold. Establishing where capability is acquired requires task-level measurements.

A.4 Capability ratchet diagnostics

Let the mass of the unguided capability region be

$$M_k = \mathbb{P}_x[x \in \mathcal{C}_k^0(\alpha)]. \tag{30}$$

Using the same task distribution and threshold across rounds, define the newly acquired and forgotten task sets:

$$\mathcal{A}_k = \mathcal{C}_{k+1}^0(\alpha) \setminus \mathcal{C}_k^0(\alpha), \qquad \mathcal{F}_k = \mathcal{C}_k^0(\alpha) \setminus \mathcal{C}_{k+1}^0(\alpha). \tag{31}$$

The three probability masses in Equation 7 are

$$\rho_k^{\mathrm{tr}} = \mathbb{P}_x[x \in \mathcal{B}_k(\alpha) \cap \mathcal{C}_{k+1}^0(\alpha)], \tag{32}$$

$$\rho_k^{\mathrm{leap}} = \mathbb{P}_x\left[x \in \mathcal{C}_{k+1}^0(\alpha) \setminus (\mathcal{C}_k^0(\alpha) \cup \mathcal{B}_k(\alpha))\right], \tag{33}$$

$$\rho_k^{\mathrm{forget}} = \mathbb{P}_x[x \in \mathcal{C}_k^0(\alpha) \setminus \mathcal{C}_{k+1}^0(\alpha)]. \tag{34}$$

These quantities describe acquisition within the previously exposed region, acquisition outside it, and loss of previously reliable tasks. The set $\mathcal{A}_k$ partitions into its intersection with $\mathcal{B}_k$ and its complement, so $\mathbb{P}_x[x \in \mathcal{A}_k] = \rho_k^{\mathrm{tr}} + \rho_k^{\mathrm{leap}}$, while $\mathbb{P}_x[x \in \mathcal{F}_k] = \rho_k^{\mathrm{forget}}$. Subtracting the forgotten mass recovers Equation 7:

$$M_{k+1} - M_k = \rho_k^{\mathrm{tr}} + \rho_k^{\mathrm{leap}} - \rho_k^{\mathrm{forget}}. \tag{35}$$

This is a descriptive decomposition, not a guarantee of improvement or a causal attribution of acquisition to hint-based training. It permits forgetting and does not assume monotonic capability growth. When the denominators are nonzero, the corresponding transfer and retention rates are $\rho_k^{\mathrm{tr}}/\mathbb{P}_x[x \in \mathcal{B}_k(\alpha)]$ and $1 - \rho_k^{\mathrm{forget}}/M_k$, respectively.

These population quantities define diagnostics rather than results established by a few successful rollouts. Estimation requires repeated task-level evaluations, a prespecified threshold, and uncertainty reporting near that threshold. Efficiency improvements should be assessed separately under a stated success and task-matching protocol; they need not change membership in a success-based capability region. The alternating ALFWorld analysis in Section 5.3 and Appendix G estimates this task-level decomposition using repeated evaluations and an explicit empirical membership rule.

B Conditional Convergence and Capability Expansion

This section develops the successful-target projection condition in Section 3 into convergence bounds for a fixed-target internalization block. Selective supervision enters through an explicit update-discrepancy term. We then bound acquisition and forgetting in Equation 7 to obtain a sufficient condition for net capability expansion. The optimization arguments use the smooth stochastic descent analysis of Ghadimi & Lan (2013) and the Polyak–Łojasiewicz condition studied by Karimi et al. (2016).

B.1 Fixed-target objective and selective-update discrepancy

Fix a task distribution ν and a target distribution $\mu^+(\cdot \mid x)$ supported on successful trajectories, so that $\mu^+(\{\tau : J_x(\tau) = 1\} \mid x) = 1$ for ν-almost every task. Both remain fixed throughout the block. All trajectory distributions use the same environment protocol and finite execution budget. Write

$$F(\theta) = \mathbb{E}_{x\sim\nu}\big[D_{\mathrm{KL}}\big(\mu^+(\cdot \mid x) \,\|\, P_\theta^0(\cdot \mid x)\big)\big], \qquad F_* = \inf_\theta F(\theta) \geq 0. \tag{36}$$

Assume finite divergence on the iterates under consideration and conditions permitting differentiation under the expectation. For a target trajectory τ, let $\mathcal{T}(\tau)$ index its policy-generated tokens, and define the token score $s_j(\theta; x, \tau) = \nabla_\theta \log \pi_\theta(y_j \mid x, \mathcal{H}_j^0)$. By the trajectory factorization in Appendix A.2,

$$\nabla F(\theta) = -\mathbb{E}_{x\sim\nu,\,\tau\sim\mu^+}\left[\sum_{j\in\mathcal{T}(\tau)} s_j(\theta; x, \tau)\right]. \tag{37}$$

Selective supervision instead assigns fixed, nonnegative coefficients $a_j(x, \tau)$ to these terms. These coefficients can include trajectory weights, token normalization, base weights, attribution gates, and auxiliary-loss scaling. Let

$$F_a(\theta) = -\mathbb{E}_{x\sim\nu,\,\tau\sim\mu^+}\left[\sum_{j\in\mathcal{T}(\tau)} a_j(x, \tau) \log \pi_\theta(y_j \mid x, \mathcal{H}_j^0)\right]. \tag{38}$$

Proposition 1 (Discrepancy induced by selective supervision). Suppose $\|s_j(\theta; x, \tau)\| \leq G$ on the target support. If the conditional mean update direction is $\nabla F_a(\theta) + u(\theta)$, where $\|u(\theta)\| \leq R$, then

$$\big\|\nabla F_a(\theta) + u(\theta) - \nabla F(\theta)\big\| \leq G\,\mathbb{E}_{x\sim\nu,\,\tau\sim\mu^+}\left[\sum_{j\in\mathcal{T}(\tau)} |a_j(x, \tau) - 1|\right] + R. \tag{39}$$

Proof. Subtract Equation 37 from the gradient of Equation 38. The difference is $-\mathbb{E}[\sum_j (a_j - 1)s_j] + u$. The triangle inequality, followed by the score bound, gives Equation 39. ∎

Thus, the effect of selective weighting can be retained as a gradient-discrepancy term instead of identifying the selective objective with full-trajectory KL projection. The following results allow any update direction satisfying the stated discrepancy bound.

B.2 Convergence of inexact internalization updates

Consider stochastic first-order updates

$$\theta_{t+1} = \theta_t - \eta g_t, \qquad t = 0, \ldots, T-1, \tag{40}$$

where t indexes optimization steps within the fixed-target block. Let $\mathcal{H}_t^{\mathrm{opt}}$ contain the block's fixed targets and all optimization randomness preceding step t, and write $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{H}_t^{\mathrm{opt}}]$. Assume:

1. F has an L-Lipschitz gradient on a domain containing every update segment, with $L > 0$.
2. $\mathbb{E}_t[g_t] = \nabla F(\theta_t) + b_t$, where $\|b_t\| \leq \varepsilon_g$ almost surely.
3. $\mathbb{E}_t\|g_t - \mathbb{E}_t[g_t]\|^2 \leq \sigma^2$.

All norms are Euclidean. The initial point and target distributions are held fixed when taking expectations over optimization randomness.

Theorem 1 (Convergence to a stationary neighborhood). For $0 < \eta \le 1/L$, the iterates in Equation 40 satisfy

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\theta_t)\|^2 \le \frac{2(F(\theta_0)-F_*)}{\eta T} + \varepsilon_g^2 + L\eta\sigma^2. \tag{41}$$

In particular, with $\eta = \min\{1/L, T^{-1/2}\}$, the average expected squared gradient norm is at most $\varepsilon_g^2 + O(T^{-1/2})$. For unbiased deterministic projection updates, $\varepsilon_g = \sigma = 0$, a constant admissible step size gives an $O(T^{-1})$ bound.

Proof. Let $d_t = \nabla F(\theta_t)$ and $\bar{g}_t = d_t + b_t$. Smoothness and the conditional variance bound imply

$$\begin{aligned}\mathbb{E}_t F(\theta_{t+1}) &\le F(\theta_t) - \eta\langle d_t, \bar{g}_t\rangle + \frac{L\eta^2}{2}\big(\|\bar{g}_t\|^2 + \sigma^2\big)\\ &\le F(\theta_t) - \frac{\eta}{2}\|d_t\|^2 + \frac{\eta}{2}\|b_t\|^2 + \frac{L\eta^2\sigma^2}{2}.\end{aligned} \tag{42}$$

For the second inequality, use $L\eta \le 1$ and $-\langle d, d+b\rangle + \frac{1}{2}\|d+b\|^2 = -\frac{1}{2}\|d\|^2 + \frac{1}{2}\|b\|^2$. Take total expectations, sum over t, and use $\mathbb{E}F(\theta_T) \ge F_*$. Dividing by $\eta T/2$ proves the bound. ∎

The stationarity result does not require convexity. A quantitative rate for the projection risk itself follows under an additional gradient-dominance condition.

Theorem 2 (Projection convergence and reliable-task acquisition). In addition to the assumptions of Theorem 1, suppose the Polyak–Łojasiewicz inequality

$$\frac{1}{2}\|\nabla F(\theta)\|^2 \ge \mu\big(F(\theta) - F_*\big), \qquad \mu > 0, \tag{43}$$

holds throughout the iterate domain. Choose $0 < \eta < \min\{1/L, 1/\mu\}$, and define

$$q = 1 - \mu\eta, \qquad D_0 = F(\theta_0) - F_*, \qquad R_\eta = \frac{\varepsilon_g^2 + L\eta\sigma^2}{2\mu}. \tag{44}$$

Then

$$\mathbb{E}F(\theta_t) \le F_* + q^t D_0 + (1-q^t)R_\eta. \tag{45}$$

For the reliability threshold $\alpha \in (0,1)$, let $c_\alpha = -\log\alpha > 0$ and $U_t = \mathbb{P}_{x\sim\nu}[v_{\theta_t}^0(x) < \alpha]$. The fraction of tasks below the reliability threshold satisfies

$$\mathbb{E}U_t \le \min\left\{1, \frac{F_* + q^t D_0 + (1-q^t)R_\eta}{c_\alpha}\right\}. \tag{46}$$

Proof. Apply Equation 43 to Equation 42, subtract F_*, and take expectations:

$$\mathbb{E}[F(\theta_{t+1}) - F_*] \le q\,\mathbb{E}[F(\theta_t) - F_*] + \frac{\eta}{2}(\varepsilon_g^2 + L\eta\sigma^2). \tag{47}$$

Unrolling this recursion yields Equation 45. For each fixed θ, write $D_x(\theta) = D_{\mathrm{KL}}(\mu^+(\cdot \mid x)\|P_\theta^0(\cdot \mid x))$. The successful-target bound in Equation 8 gives $v_\theta^0(x) \ge \exp[-D_x(\theta)]$. Consequently,

$$\{x : v_\theta^0(x) < \alpha\} \subseteq \{x : D_x(\theta) > c_\alpha\}, \qquad \mathbb{P}_{x\sim\nu}[v_\theta^0(x) < \alpha] \le \frac{F(\theta)}{c_\alpha}, \tag{48}$$

where the last step is Markov's inequality. Taking expectations and substituting Equation 45 proves the claim. ∎

Corollary 1 (Finite-step acquisition). For a desired residual fraction $\epsilon \in (0,1)$, suppose $r = \epsilon c_\alpha - F_* - R_\eta > 0$. A sufficient number of updates for $\mathbb{E}U_t \le \epsilon$ is

$$t \ge \frac{1}{\mu\eta}\log\left(\max\left\{1, \frac{D_0}{r}\right\}\right). \tag{49}$$

Proof. Use $q^t \le e^{-\mu\eta t}$ and $(1-q^t)R_\eta \le R_\eta$ in Equation 46. The stated condition makes the numerator at most ϵc_α. ∎

These bounds distinguish approximation error F_*, update discrepancy ε_g, and stochastic noise σ. In the realizable, exact, deterministic case, $F_* = \varepsilon_g = \sigma = 0$, the projection risk and the upper bound on the unreliable-task fraction decay geometrically. With constant-step stochastic or inexact updates, Equation 45 instead gives a residual neighborhood. The task-coverage conclusion concerns the fixed distribution ν used in the block.

B.3 NET FRONTIER EXPANSION WITH RETENTION CONTROL

We now return to the original task distribution $\mathcal{D}$. Fix a round k and its capability regions. Select a measurable region $\mathcal{E}_k \subseteq \mathcal{B}_k(\alpha)$ with mass $b_k = \mathbb{P}_{x\sim\mathcal{D}}[x \in \mathcal{E}_k] > 0$, and successful-target distributions $\mu_k^+(\cdot \mid x)$ on this region. For the updated policy, define

$$F_{\mathcal{E},k+1} = \mathbb{E}_{x\sim\mathcal{D}(\cdot|\mathcal{E}_k)}\left[D_{\mathrm{KL}}\Big(\mu_k^+(\cdot \mid x)\|P^0_{\theta_{k+1}}(\cdot \mid x)\Big)\right]. \tag{50}$$

For $0 < \gamma < 1-\alpha$, let

$$m_k(\gamma) = \mathbb{P}_{x\sim\mathcal{D}}[\alpha \le v_k^0(x) < \alpha+\gamma],$$
$$K_k^{\mathrm{ret}} = \int_{\mathcal{C}_k^0(\alpha)} D_{\mathrm{KL}}\Big(P^0_{\theta_k}(\cdot \mid x)\|P^0_{\theta_{k+1}}(\cdot \mid x)\Big)\, d\mathcal{D}(x). \tag{51}$$

Here $m_k(\gamma)$ measures previously reliable tasks close to the threshold, and K_k^{ret} measures trajectory-distribution drift on the old capability region. Assume both KL risks are finite. Let $\mathcal{A}_k$ and $\mathcal{F}_k$ be the acquisition and forgetting sets in Equation 31. Their masses are $\rho_k^{\mathrm{tr}} + \rho_k^{\mathrm{leap}}$ and ρ_k^{forget}, respectively.

Theorem 3 (Acquisition–retention bound). Let $[z]_+ = \max\{z, 0\}$ and $c_\alpha = -\log\alpha$. Under the fixed task distribution, reliability threshold, and evaluation protocol,

$$M_{k+1} - M_k \ge b_k\left[1 - \frac{F_{\mathcal{E},k+1}}{c_\alpha}\right]_+ - m_k(\gamma) - \frac{K_k^{\mathrm{ret}}}{2\gamma^2}. \tag{52}$$

Thus, a sufficient condition for strict net expansion is

$$b_k\left[1 - \frac{F_{\mathcal{E},k+1}}{c_\alpha}\right]_+ > m_k(\gamma) + \frac{K_k^{\mathrm{ret}}}{2\gamma^2}. \tag{53}$$

Proof. Apply Equation 48 with $\nu = \mathcal{D}(\cdot \mid \mathcal{E}_k)$. Because $\mathcal{E}_k$ is disjoint from $\mathcal{C}_k^0(\alpha)$, every task in $\mathcal{E}_k \cap \mathcal{C}_{k+1}^0(\alpha)$ is newly acquired. Therefore,

$$\mathbb{P}_x[x \in \mathcal{A}_k] \ge b_k\left[1 - \frac{F_{\mathcal{E},k+1}}{c_\alpha}\right]_+. \tag{54}$$

For retention, fix x and abbreviate $p = v_k^0(x)$, $q = v_{k+1}^0(x)$. Partitioning trajectories into successful and unsuccessful outcomes and applying the log-sum inequality gives

$$D_{\mathrm{KL}}(P^0_{\theta_k}\|P^0_{\theta_{k+1}}) \ge \mathrm{kl}(p\|q) \ge 2(p-q)^2, \tag{55}$$

where all distributions are conditional on x, logarithms are natural, and kl is Bernoulli KL divergence. For completeness, when $q \in (0,1)$, the function $p \mapsto \mathrm{kl}(p\|q)$ and its first derivative vanish at $p = q$, while its second derivative is $1/[p(1-p)] \ge 4$. Taylor's inequality gives the second bound; boundary cases follow by limits.

A forgotten task either has $\alpha \le p < \alpha+\gamma$, or has $p \ge \alpha+\gamma$ and $q < \alpha$. In the latter case, $p - q > \gamma$, so Equation 55 implies a trajectory KL divergence greater than $2\gamma^2$. Integrating over the old capability region yields

$$\mathbb{P}_x[x \in \mathcal{F}_k] \le m_k(\gamma) + \frac{K_k^{\mathrm{ret}}}{2\gamma^2}. \tag{56}$$

Subtract this upper bound from Equation 54 using the exact acquisition–forgetting balance in Equation 7. ∎

Connecting optimization to frontier expansion. If the fixed-target block uses $\nu = \mathcal{D}(\cdot \mid \mathcal{E}_k)$ and satisfies Theorem 2, set $\overline{F}_t = F_* + q^t D_0 + (1-q^t)R_\eta$. Taking expectations in Theorem 3 and using convexity of $z \mapsto [1 - z/c_\alpha]_+$ gives

$$\mathbb{E}[M_{k+1} - M_k] \ge b_k\left[1 - \frac{\overline{F}_t}{c_\alpha}\right]_+ - m_k(\gamma) - \frac{\mathbb{E}K_k^{\mathrm{ret}}}{2\gamma^2}, \tag{57}$$

where θ_{k+1} denotes the output after t inner updates. The resulting condition links the amount of exposed learning opportunity, the accuracy of internalization, and the stability of previously reliable behavior. It applies round by round, with targets and the selected region fixed within each analyzed block.

Relation to the implementation. The analysis gives conditional sufficient conditions linking successful-target fitting and retention to capability expansion. Theorems 1–2 analyze a fixed-target objective under the stated optimization assumptions, with the Polyak–Łojasiewicz condition used in Theorem 2. HELICES holds targets and detached attribution weights fixed within each learner update, refreshes them across rounds, and combines selective supervision, GRPO, and regularization using AdamW. Extending the convergence rates to this evolving training process requires control of target drift and the effective optimizer-update discrepancy. Proposition 1 quantifies the departure of selective supervision from full-trajectory projection; the discrepancy and expansion bounds have not been numerically calibrated for the reported runs. The gate's practical contribution is assessed by the component ablations in Appendices E and D.7, while fresh unguided evaluations measure acquisition and retention.

C HELICES Implementation and Training Protocol

C.1 Hint representation and joint optimization

The ALFWorld implementation uses one JSON rule with `evidence` and `advice` fields. Evidence grounds the rule in the complete source trajectory and feedback (success, termination status, and step count). Advice begins with an observable applicability condition and abstracts instance identifiers into task roles. The prompt disallows complete routes and fixed object–location priors. Only the advice enters the solver's system prompt before a new episode and remains visible throughout it. These are generation constraints; they do not establish that every sampled hint is transferable. No nearest-neighbor retrieval or persistent hint library is used in HELICES.

The success-conditioned efficiency term is optional: it requires a meaningful execution-cost measure and is omitted when that measure is fixed or unavailable. In the ALFWorld configuration below, environment steps provide this measure. For a valid candidate, let J^0, J^h be source and hinted success, L^0, L^h their step counts, $T_{\max}$ the step budget, and z^0, z^h their truncation indicators. The implemented generator reward is

$$r^{\text{probe}} = (J^h - J^0) + 0.2 J^0 J^h \,\text{clip}\left(\frac{L^0 - L^h}{T_{\max}}, -1, 1\right) - 0.1\mathbb{I}[z^0 = 0, z^h = 1]. \tag{58}$$

Protocol-invalid hints receive a pre-normalization raw generator reward of -1 and do not trigger a solver rollout or a replacement hint-generation attempt. Their generator samples remain in the candidate group used for reward standardization; zero-token failed or aborted samples contribute no model-generated tokens to the loss. The solver reward for a valid candidate is proportional to hinted task success. Generator and solver rewards are standardized separately over the candidates belonging to one source, using population standard deviation and 10^{-6} in the denominator, then multiplied by their respective role weights. The common source-success offset cancels in the success component after centering; the efficiency and truncation terms can still distinguish candidates. Thus, this normalization ranks interventions within each source and is not a task-level frontier-sampling rule. Source and hinted executions use independent sampling streams during joint learning; same-seed pairing is introduced for verified internalization.

Each role uses the clipped surrogate

$$\mathcal{L}^r_{\text{GRPO}} = -\mathbb{E}\left[\text{Mean}_{t\in\mathcal{T}^r} \min(\rho^r_t A^r, \text{clip}(\rho^r_t, 1-\epsilon_-, 1+\epsilon_+)A^r)\right], \tag{59}$$

where ρ^r_t compares current and rollout-time probabilities in the same role context, and only model-generated tokens contribute. Regularization is $\sum_r [\beta_r \mathcal{L}^r_{\text{KL}} - \eta_r \mathcal{H}^r]$. The original source trajectories are not included in the joint policy-gradient batch.

The KL term uses the token-wise low-variance estimator. Let $d_t = \log \pi_\theta(a_t \mid x_t) - \log \pi_{\text{ref}}(a_t \mid x_t)$. The implementation computes

$$\widehat{D}_{\text{KL},t} = \text{clip}(\exp(-d_t) - 1 + d_t, -10, 10),$$

then takes a sample-wise token mean under the loss mask before applying the role-specific coefficient. No additional importance-ratio correction is used. During joint training, π_{ref} is the frozen original base model. For single-switch internalization, it is the frozen joint endpoint that also initializes the internalization actor. In the alternating runs, each joint block retains the original base model

as its reference, whereas each internalization block uses its immediately preceding joint endpoint. The reference remains fixed within a block. It is distinct from the rollout-time old policy used in the PPO importance ratio.

C.2 Matched discovery, independent confirmation, and target weighting

At the transition to internalization, the jointly trained checkpoint supplies both a frozen generator copy and the initial trainable solver. The generator continues to read newly collected solver trajectories; its output distribution is therefore not fixed. A fixed-probe capability audit instead requires the cached task–source–hint protocol in Appendix A.

Each internalization round samples a batch of tasks and a group of unguided trajectories per task, using the benchmark-specific values in Table 6. Discovery evaluates a hint under the sampling seed of its source trajectory. For a group of G trajectories, candidate j is confirmed using the seed and unguided outcome of partner $(j+1) \bmod G$. The partner trajectory is held out from the generation of that hint.

For the ALFWorld improvement labels c_i^{disc} and c_i^{conf} in Equation 12, F2S requires failure without the hint and success with it. The optional efficiency case, denoted Eff in this implementation, requires success on both sides, at least two fewer environment steps, and a relative reduction of at least 15%. All other pairs return $\varnothing$. Both discovery and confirmation must satisfy the same class. When efficiency is disabled, only failure-to-success pairs qualify. Candidates failing this check do not enter the auxiliary learner batch. A small number of accepted targets may additionally be removed to align the batch with data parallelism; the implementation removes efficiency cases before F2S cases where possible.

The accepted confirmation response is kept intact and paired with its partner's original no-hint prompt. Environment observations remain conditioning history with zero loss weight. Actions inside `<action>` spans receive base weight 1 and bypass attribution gating; other model-generated tokens receive base weight 0.25. Teacher and student attribution scores are obtained by rescoring the same target with the current rollout solver under hinted and unhinted prompts. Both scores and the gate are detached during optimization; this scoring solver is distinct from the frozen generator.

Let N_s be the source count and $B = N_s + |\mathcal{A}|$ the learner batch size. To implement Equation 15 under a full-batch average, source policy-gradient and KL coefficients are multiplied by B/N_s, and auxiliary target coefficients are

$$c_i = \frac{B w_i}{\sum_{j \in \mathcal{A}} w_j}, \qquad w_i = \begin{cases} 4, & i \in \mathrm{F2S}, \\ 1, & i \in \mathrm{Eff}. \end{cases} \tag{60}$$

The F2S-to-efficiency weight ratio is per sample, with no additional multiplication by shortening magnitude. Token means use all model-generated positions, including those assigned zero gate weight; they are not renormalized over retained tokens. Auxiliary targets contribute no policy-gradient, KL, or entropy term. Unguided source trajectories receive binary success rewards, grouped by task for GRPO. An empty accepted set contributes zero auxiliary loss.

Equation 16 combines unguided GRPO, source KL regularization, and verified internalization with coefficient λ_{int}.

C.3 Interpretation of the attribution contrast

For a fixed unhinted history $\mathcal{H}^0$, write $p_h(y) = \bar{\pi}_k(y \mid x, \mathcal{H}^0, h)$ and $p_0(y) = \bar{\pi}_k(y \mid x, \mathcal{H}^0)$. Then

$$\mathbb{E}_{y \sim p_h}\left[\log \frac{p_h(y)}{p_0(y)}\right] = D_{\mathrm{KL}}(p_h \| p_0) \geq 0. \tag{61}$$

This identity describes the conditional distributions before selection; it is not a guarantee about token averages in accepted successful trajectories. Positive likelihood support need not imply correct or necessary reasoning, and whole-trajectory success does not establish that every action was optimal. The gate is a selective supervision rule whose effect must be assessed empirically. Likewise, its weighted loss is not the full-trajectory divergence in Appendix A.2.

D EXPERIMENTAL DETAILS

D.1 TRAINING SCHEDULE AND SAMPLING BUDGET

We use Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct as the backbones. Table 6 summarizes the training schedule shared by both scales. A batch consists of B tasks, with G Solver trajectories sampled per task; the per-round Solver trajectory budget is therefore BG. We follow the task-specific baseline configurations (Wu et al., 2026b; Zhang et al., 2026c; Lu et al., 2026b; Fan et al., 2026) to match the number of policy-update Solver trajectories. We use this quantity, rather than update count alone, because methods package different numbers of trajectories into an update.

Table 6: HELICES training schedule. Batch size counts tasks; group size counts Solver trajectories per task. A round denotes one training update.

Configuration	ALFWorld	Search	Mind2Web
Task batch size B	16	128	16
Trajectories per task G	8	8	32
Solver trajectories per round BG	128	1,024	512
Joint-training rounds	120	120	120
Internalization rounds	30	30	30
Total training rounds	150	150	150
Total policy-update Solver trajectories	19,200	153,600	76,800

Table 7: Reported policy-update Solver-trajectory budgets for closely related agent-training methods. Totals multiply the reported number of updates, task batch size, and Solver rollouts per task. A dash means that the source reports epochs rather than an unambiguous number of rollout updates, so we do not infer a total. Hardware is reported for context, not normalized across GPU types.

Benchmark	Method	Backbone	Updates	Tasks	Solver/task	Total	Hardware
ALFWorld	SEED (Wu et al., 2026b)	Qwen2.5-3B/7B	150	16	8	19,200	8×A800
	SAPO (Zhang et al., 2026c)	Qwen2.5-7B	150 ep.	16	8	—	8×H200
	Skill0 (Lu et al., 2026b)	Qwen2.5-VL-3B/7B	≤ 180	16	8	$\leq 23,040$	4×H800
	Evolving-RL (Fan et al., 2026)	Qwen2.5-7B	75	16	8×4	38,400	8×H800
	HELICES	Qwen2.5-3B/7B	150	16	8	**19,200**	8 GPUs
Search-QA	SEED (Wu et al., 2026b)	Qwen2.5-3B/7B	150	128	8	153,600	8×A800
	SAPO (Zhang et al., 2026c)	Qwen2.5-7B	1 ep.	512	8	—	8×H200
	HELICES	Qwen2.5-3B/7B	150	128	8	**153,600**	8 GPUs
Mind2Web	Evolving-RL (Fan et al., 2026)	Qwen2.5-7B	150	16	8×4	76,800	8×H800
	HELICES	Qwen2.5-3B/7B	150	16	32	**76,800**	8 GPUs

Table 7 uses the most comparable quantity available across papers: Solver trajectories that enter policy optimization. For Evolving-RL, each of the $N = 8$ candidate skills is evaluated on $K = 4$ retrieved tasks, so its per-update Solver budget is BNK; its 75-step ALFWorld run therefore uses twice the HELICES policy-update trajectory total, whereas its 150-step Mind2Web run matches HELICES exactly. SEED uses exactly the HELICES trajectory total on ALFWorld and Search-QA. SAPO reports the same ALFWorld batch and group sizes but describes its duration as 150 epochs; without an explicit update count, we leave its total blank. Skill0 reports at most 180 steps, so its ALFWorld total is an upper bound. The table does not equate wall-clock cost across different backbones, modalities, hardware, sequence lengths, or auxiliary procedures.

Budget interpretation. These comparisons show that HELICES operates within the standard policy-training budget of recent agent-RL methods rather than obtaining its gains by substantially increasing the number of Solver trajectories used for policy updates. Its update-trajectory budget is identical to SEED on ALFWorld and Search-QA and to Evolving-RL on Mind2Web, while using half as many such trajectories as Evolving-RL on ALFWorld. These cross-paper comparisons concern policy-update sampling. HELICES additionally performs hint generation, discovery, and confirmation, whose environment-interaction upper bounds are disclosed separately in Table 8. Our direct ALFWorld comparisons in Appendix D.6 supplement this accounting with total environment episodes and measured token and timing costs on matched hardware.

Table 8: HELICES environment-interaction accounting. "Update" counts Solver trajectories used in the policy objective. "Source" counts the one additional unguided source episode per task during joint training. "Verify" is a conservative upper bound of two additional guided executions per internalization candidate (discovery and confirmation). Invalid hints and candidates that do not require confirmation reduce the realized total.

Environment episodes	ALFWorld	Search	Mind2Web
Policy-update Solver trajectories	19,200	153,600	76,800
Joint-stage source episodes	1,920	15,360	1,920
Internalization discovery/confirmation (upper bound)	7,680	61,440	30,720
Total environment episodes (upper bound)	**28,800**	**230,400**	**109,440**

Rounds 1–120 jointly optimize hint generation and utilization through shared parameters. Within joint training, the shared model updated at round r is used to collect the unguided source trajectories, generate hints, and execute hinted rollouts at round $r + 1$. Capability improvements made by the Solver therefore immediately change the evidence and model used for the next round of hint generation; this feedback operates inside every joint block, not only across joint–internalization stage boundaries. At the stage transition, we freeze a copy of the hint generator and initialize the internalization Solver from the joint model. For the main experiments, the optimizer state is reset at this transition. The alternating ALFWorld experiments preserve optimizer moments, step counters, and FP32 master parameters across stage transitions (Appendix G). Rounds 121–150 combine unguided policy optimization with supervision on verified improved trajectories. Discovery and independent confirmation select the targets; hints are removed from the Solver's training inputs, with full action supervision and attribution-weighted reasoning supervision. For all main experiments, internalization starts from the joint checkpoint after update 120, and the main benchmark results use the final Solver checkpoint after update 150. The trajectory budget above specifies Solver training sampling; hint generation itself is a language-model generation rather than an environment episode. Table 8 separately accounts for the source, discovery, and confirmation interactions used to construct and verify those hints; their procedure is detailed in Appendix C.2.

Table 9: Shared training hyperparameters for Qwen2.5-3B/7B on ALFWorld, Search-R1, and Mind2Web, also used in the alternating ALFWorld experiments.

Parameter	Joint training	Internalization
Learning rate (constant per stage)	10^{-6}	2×10^{-7}
Solver role weight	1.0	1.0
Generator role weight	0.05	Frozen
Solver KL coefficient	0.01	0.01
Generator KL coefficient	0.01	—
Internalization coefficient λ_{int}	—	0.005
Solver temperature	1.0	1.0
Generator temperature	1.0	0.7
Top-p (both roles)	0.9	0.9
Top-k (both roles)	-1 (disabled)	-1 (disabled)
GRPO clipping (ϵ_-, ϵ_+)	$(0.10, 0.15)$	$(0.10, 0.15)$
Solver entropy coefficient	0	0
Generator entropy coefficient	-0.012	—
Maximum gradient norm	1.0	1.0
Reasoning gate margin m	—	0.02
Reasoning gate sharpness γ	—	5.0
Action / reasoning base weights	—	$1.0 \,/\, 0.25$
F2S / efficiency target weights	—	$4 \,/\, 1$

AdamW uses Megatron's CPU-offload implementation, configured as `adam` in the command-line interface. Weight decay is zero for biases and one-dimensional parameters. Each rollout batch is processed once and produces one optimizer step. The learner uses the entire batch without a separate PPO mini-batch partition.

Table 10: Shared optimizer and update settings for both model scales on all three benchmarks and for the alternating ALFWorld experiments. B and G follow Table 6; K counts accepted auxiliary targets retained for the update.

Parameter	Joint training	Internalization
Optimizer	AdamW	AdamW
Adam (β_1, β_2)	$(0.9, 0.98)$	$(0.9, 0.98)$
Optimizer epsilon	10^{-8}	10^{-8}
Weight decay	0.1	0.1
Warmup (optimizer steps)	0	0
Learning-rate schedule	Constant	Constant
Passes over each rollout batch	1	1
Optimizer steps per rollout batch	1	1
Configured global sequence batch	$2BG$	BG
Actual global sequence batch	$2BG$	$BG + K$
Configured micro-batch size	1	1
Token limit per GPU	20,480	20,480
Training GPUs	8	8
Data / tensor parallel size	$4 \,/\, 2$	$4 \,/\, 2$
Pipeline / context parallel size	$1 \,/\, 1$	$1 \,/\, 1$

The joint learner batch contains BG Generator outputs and BG Solver trajectories. Its total size is 256 sequences for ALFWorld, 2,048 for Search-R1, and 1,024 for Mind2Web. The internalization learner batch contains BG unguided source trajectories and K accepted auxiliary targets, giving $128 + K$, $1{,}024 + K$, and $512 + K$ sequences, respectively. Both dynamic micro-batching and dynamic global batching are enabled. The micro-batch configuration is 1, while runtime micro-batches are formed dynamically under a 20,480-token budget per GPU. Gradients accumulate over all micro-batches before the optimizer step.

Sampling and entropy regularization. Table 9 gives the settings shared across the three benchmarks and both model scales. The Solver temperature is 1.0 for unguided source and hinted executions during joint training, and for source, discovery, and confirmation executions during internalization. Both roles use top-p 0.9 with top-k truncation disabled. The Generator is frozen during internalization and generates hints at temperature 0.7. Joint GRPO uses the same clipping parameters for both roles; internalization applies them to the Solver. Under the loss convention $\mathcal{L} - \eta_r \mathcal{H}^r$, the joint Generator coefficient $\eta_q = -0.012$ contributes $+0.012\mathcal{H}^q$ to the minimized loss, implementing an entropy penalty. The Solver entropy coefficient is zero in both stages.

Reasoning gates and target weights. During internalization, Equation 14 uses $m = 0.02$ and $\gamma = 5.0$, giving the reasoning-token gate

$$g_{it} = \max\{2\sigma(5(\Delta_{it} - 0.02)) - 1,\, 0\}\,.$$

The reasoning base weight 0.25 is applied before gating, so its token coefficient is $0.25 g_{it}$. Action tokens have base weight 1 and gate 1. Each accepted F2S trajectory receives relative weight 4, and each accepted efficiency trajectory receives relative weight 1, followed by normalization over accepted targets as in Appendix C.2. For a nonempty accepted set with n_{F2S} F2S and n_{Eff} efficiency targets, their respective total normalized weights are $4n_{\mathrm{F2S}}/(4n_{\mathrm{F2S}}+n_{\mathrm{Eff}})$ and $n_{\mathrm{Eff}}/(4n_{\mathrm{F2S}}+n_{\mathrm{Eff}})$.

D.2 Benchmarks and evaluation metrics

Data splits and sampling. Table 11 lists the training pools and actual evaluation sizes used by HELICES at both model scales. No separate validation set is allocated for these experiments. Joint-training data are shuffled with seed 42.

ALFWorld retains the full training pool across all six task families. HELICES evaluates all Seen and Unseen tasks in three passes, giving 420 Seen and 402 Unseen episodes per checkpoint. The ALFWorld baseline results in Table 1 use one evaluation pass per checkpoint on the 140 Seen tasks; the separate multi-seed and ablation studies follow their respective repetition protocols. Mind2Web

Table 11: HELICES training splits and actual evaluation sizes. Counts denote unique tasks or questions. The Mind2Web rows share one training pool of 1,009 tasks, and the ALFWorld rows share one pool of 3,553.

Benchmark / dataset	Training split (count)	Evaluation split	Count
ALFWorld Seen	`train` (3,553)	`valid_seen`	140
ALFWorld Unseen	Same training pool	`valid_unseen`	134
Natural Questions	`train` (79,168)	`test` subset	253
TriviaQA	Not used (0)	`test` subset	794
PopQA	Not used (0)	`test` subset	1,001
HotpotQA	`train` (90,447)	`dev` subset	673
2WikiMultiHopQA	Not used (0)	`dev` subset	1,144
MuSiQue	Not used (0)	`dev` subset	220
Bamboogle	Not used (0)	`test` subset	11
Mind2Web Cross-task	`train` (1,009)	`test_task`	252
Mind2Web Cross-website	Same training pool	`test_website`	177
Mind2Web Cross-domain	Same training pool	`test_domain`	912

retains the official splits and evaluates all 1,341 test tasks. Neither benchmark subsamples its evaluation tasks.

For Search-R1, single-hop training uses Natural Questions and multi-hop training uses HotpotQA. Within each evaluation group, we sample without replacement using seed 42, stratified by dataset in proportion to its public evaluation-set size. Each group contains 2,048 fixed questions: 253/794/1,001 for NQ/TriviaQA/PopQA and 673/1,144/220/11 for HotpotQA/2WikiMultiHopQA/MuSiQue/Bamboogle. Bamboogle therefore contributes 11 questions from its 125-question test split. The remaining five QA datasets evaluate cross-dataset generalization.

The main experiments use a prespecified 150-update training schedule and do not perform periodic benchmark evaluation during training. Their reported results are obtained only from the prespecified final checkpoint after update 150; benchmark feedback is not used for early stopping, checkpoint selection, or hyperparameter decisions. The periodic measurements shown in the stage comparisons and learning curves are produced by separate, post hoc diagnostic evaluations of saved intermediate checkpoints after training has completed. Those diagnostics use fixed evaluation sets solely to make checkpoints comparable and do not affect the main experimental results or the choice of the reported checkpoint. Further details are given in Appendix D.8.

D.3 Main-table result provenance

Table 1 groups methods by the Qwen2.5 3B or 7B model family and does not imply that every row uses the same input modality. In particular, the reported Skill0 (Lu et al., 2026b) and AgentOCR (Feng et al., 2026) results use the Qwen2.5-VL-3B-Instruct and Qwen2.5-VL-7B-Instruct backbones at the corresponding parameter scales. HELICES uses the text-only Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct backbones. Thus, the Skill0 and AgentOCR rows are size-family references rather than identical-backbone comparisons.

Evaluation source. To maximize comparability, we prioritize, in order of availability, checkpoint evaluation on our fixed manifests, complete end-to-end reproduction, and results reported in the original paper. Every checkpoint evaluation and complete reproduction uses the same task manifest, benchmark evaluator, and aggregation rule as HELICES; decoding and execution budgets are also matched wherever the method interface permits. We do not construct a benchmark-group result by selecting per-dataset maxima from multiple source configurations. Results that cannot be rerun are retained from one cited configuration and identified explicitly below. In Table 1, the Vanilla, Skill-Prompt, Few-Shot (text), and SkillRL rows are our evaluations of the corresponding checkpoints. For AgentOCR, Skill0, and SAPO, the Search-R1 cells are likewise obtained by our checkpoint evaluation. We reproduce the complete training pipeline for PPO, GRPO, EvolveR, Skill-GRPO, and Evolving-RL across their reported Table 1 results, and for Skill0 on ALFWorld. The non-Search

AgentOCR and SAPO results and all remaining baseline rows are taken intact from the cited original papers rather than assembled from different papers or configurations. Except for the Qwen2.5-7B Mind2Web result described below, each HELICES main-table row is from one prespecified complete end-to-end training pipeline, not an average of the three continuation seeds reported in Table 3. For every Search-R1 row, the three single-hop datasets are evaluated from one common single-hop checkpoint, and the four multi-hop datasets are evaluated from one common multi-hop checkpoint. Table 2 is produced entirely by our complete reproduction pipeline for all listed configurations. Its Qwen2.5-7B split cells report means from three independent full-pipeline training seeds for GRPO, Evolving-RL, and HELICES. The corresponding dispersion, per-seed weighted summaries, and the minor rounding distinction between aggregation orders appear in Appendix D.7.

ALFWorld. ALFWorld (Shridhar et al., 2021) comprises Pick, Look, Clean, Heat, Cool, and Pick2 tasks. Task success is the fraction of episodes that complete the specified goal. We evaluate HELICES on 140 seen and 134 unseen tasks; Table 1 presents the seen task-family scores and their task-weighted summaries. Seen task counts n_c in Pick/Look/Clean/Heat/Cool/Pick2 order are 35/13/27/16/25/24. Baseline results use one evaluation pass: each category score is $100s_c/n_c$, and the SR column is $100\sum_c s_c/140$, where s_c counts successful episodes. HELICES pools three evaluation passes: its category score is $100s_c/(3n_c)$ and its overall SR is $100\sum_c s_c/420$, with s_c counting successes across all three passes. Category percentages and SR are rounded to one decimal place. All checkpoint evaluations and complete reproductions use this same 140-task manifest, so their category and aggregate comparisons are task-aligned. For rows retained from original papers, the fixed category counts standardize our aggregation but cannot by themselves establish identical task instances, checkpoints, or repetition protocols in the source evaluation. The separate stage comparisons and controlled-ablation results compute sample-pooled success, $100\sum_c s_c/\sum_c n_c$, from their own evaluation counts. Seen and Unseen are evaluated separately; Unseen category counts are 24/18/31/23/21/17. Their checkpoints and repetition protocols remain as specified in the corresponding subsections. Each episode permits up to 50 environment steps, with at most 512 generated tokens per step.

Search. The Search-R1 suite (Jin et al., 2025) includes Natural Questions, TriviaQA, and PopQA for single-hop QA, and HotpotQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle for multi-hop QA. EM is the fraction of answers matching a reference after the evaluator's answer normalization. Group summaries use the evaluation-size weighting defined below. The agent can issue up to four search rounds, retrieving the top three passages per query. Generation is limited to 512 tokens per round and 1,536 tokens in total, with up to 2,200 characters per retrieved passage.

Mind2Web. Mind2Web (Deng et al., 2023) contains 252 cross-task, 177 cross-website, and 912 cross-domain test tasks. Following the sequential evaluator of Fan et al. (2026), an episode terminates on an incorrect action or task completion. AA is the number of correct actions divided by the number of executed actions; SR is the fraction of tasks completed in full. Each split-level SR is rounded directly from its integer number of successful tasks on the stated split manifest. Thus, AA measures accuracy along the executed trajectory. We report AA and SR separately for each split, together with a task-count-weighted summary across the three splits. The context limit is 16,384 tokens, with at most 4,096 generated tokens per step and DOM depth 20. The 3B Evolving-RL result uses the final checkpoint at update 150 with skills enabled during inference.

Evaluation-manifest-weighted summaries. For Search-R1 and Mind2Web groups with displayed percentage scores r_i, we report $\mathrm{Avg.} = \sum_i n_i r_i / \sum_i n_i$, where n_i is the number of examples in the evaluation manifest actually used for that component. ALFWorld instead pools integer episode successes over the complete 140-task Seen manifest, with family counts 35/13/27/16/25/24: one pass for main-table baselines and three passes for HELICES. Mind2Web uses the complete test manifests, with 252/177/912 tasks for the cross-task, cross-website, and cross-domain splits. Search-R1 instead uses the fixed random samples described above: its single-hop manifest contains 253/794/1,001 examples from NQ/TriviaQA/PopQA, and its multi-hop manifest contains 673/1,144/220/11 examples from HotpotQA/2WikiMultiHopQA/MuSiQue/Bamboogle. Each Search group therefore contains 2,048 evaluated examples. Search-R1 and Mind2Web aggregate values are recomputed from the displayed rounded component scores; Table 1 reports one decimal

place and Table 2 reports two. Mind2Web's split-weighted AA is distinct from pooling correct and executed actions across splits.

Inference. HELICES is evaluated without hints, with one trajectory per task per evaluation pass and greedy decoding (temperature 0 and top-p 1). HELICES uses three evaluation passes on ALFWorld; the main-table ALFWorld baselines use one pass. Failed and truncated episodes remain in the task-success denominator. Guided evaluations use the same task sets; generating hints and replaying episodes can incur additional inference cost.

D.4 Comparison with more self-distillation methods

SEED (Wu et al., 2026b) is closely related to HELICES in its use of training-time guidance to improve a policy evaluated without skills. We include SEED and three related baselines from its evaluation: GRPO+OPSD (Zhao et al., 2026), Skill-SD (Wang et al., 2026a), and SDAR (Lu et al., 2026a). SEED uses GLM-5.2 for offline hindsight-skill annotation to initialize trajectory analysis through supervised fine-tuning. During reinforcement learning, its analyzer is synchronized with the current policy; inference requires neither the external model nor skill prompts. We present this comparison separately to make the initialization supervision and evaluation settings explicit.

Table 12 reports the original task-family and dataset scores from Table 1 of Wu et al. (2026b), including its evaluations of GRPO+OPSD, Skill-SD, and SDAR. HELICES results are taken from Table 1. Results follow their respective evaluation protocols.

HELICES achieves strong performance without external-LLM supervision during post-training, using self-generated hints and environmental feedback. At 7B, it matches or exceeds the reported results of GRPO+OPSD, Skill-SD, and SDAR in all six ALFWorld task families, and those of SEED in four of six ALFWorld families and four of seven Search-QA datasets. These results highlight the effectiveness of self-generated guidance and verified internalization.

D.5 Full-pipeline multi-seed robustness on ALFWorld and Search-QA

We repeat the complete training pipelines of GRPO, Evolving-RL, and HELICES with three independent training seeds (42, 84, and 126) at both model scales. Each run starts from the corresponding base checkpoint and includes the full prespecified training schedule; seed labels are fixed in advance and are not used for model or checkpoint selection. Evaluations use the same fixed manifests, greedy hint-free decoding, and aggregation rules as the main experiments. Entries below are means $\pm$ sample standard deviations across the three complete runs. HELICES-minus-Evolving-RL differences are computed within matched seeds before aggregation. Because there are only three training seeds, these comparisons are descriptive; we report exact counts and paired seed wins rather than relying on asymptotic significance tests.

ALFWorld. Each checkpoint is evaluated over three passes on 140 Seen and 134 Unseen tasks, giving 420 and 402 episodes per training seed. Table 13 shows stable HELICES gains at both scales. HELICES exceeds Evolving-RL in all three matched seeds on both splits, with mean paired gains of 15.71/4.23 points on Seen/Unseen for 3B and 5.71/3.23 points for 7B.

Search-QA. Each complete run is evaluated on the fixed 2,048-question single-hop and 2,048-question multi-hop manifests. Table 15 again shows a HELICES win in every matched seed. The paired gains over Evolving-RL are 3.86/1.27 points on single-hop/multi-hop QA for 3B and 1.46/8.59 points for 7B. The smaller 3B multi-hop and 7B single-hop margins remain consistent in direction across all three complete runs.

D.6 Environment budgets and measured training cost on ALFWorld

We compare HELICES and Evolving-RL using Qwen2.5-7B-Instruct on ALFWorld. Table 17 reports success rates over three independent complete training runs. Each final checkpoint is evaluated over three passes on the 140 Seen and 134 Unseen tasks, following Appendix D.5. These repetitions are distinct from the fixed-checkpoint continuation study in Appendix E.

Table 12: Comparison with SEED and related self-distillation methods (%). Baseline results are taken from Table 1 of SEED (arXiv:2607.14777v1), including its evaluations of GRPO+OPSD, Skill-SD, and SDAR; HELICES results are from Table 1. We report original task-family and dataset scores under the respective evaluation protocols.

(a) ALFWorld Seen

Method	Pick	Look	Clean	Heat	Cool	Pick2
Qwen2.5-3B-Instruct						
GRPO+OPSD	100.0	82.4	85.7	75.0	70.0	60.0
Skill-SD	88.2	50.0	96.2	52.4	65.0	57.9
SDAR	97.1	62.5	100.0	61.9	75.0	84.2
SEED	100.0	100.0	100.0	100.0	70.6	80.0
HELICES	97.1	92.3	100.0	87.5	88.0	95.8
Qwen2.5-7B-Instruct						
GRPO+OPSD	91.4	61.5	100.0	87.5	76.5	52.2
Skill-SD	93.9	93.8	90.9	100.0	69.2	68.4
SDAR	94.7	75.0	100.0	86.7	68.2	78.9
SEED	100.0	100.0	96.3	80.0	100.0	100.0
HELICES	100.0	100.0	100.0	100.0	92.0	91.7

(b) Search-QA

Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bam.
Qwen2.5-3B-Instruct							
GRPO+OPSD	44.9	61.2	45.2	40.4	38.5	16.0	66.1
Skill-SD	44.4	60.4	44.0	39.5	40.4	15.4	64.9
SDAR	44.8	58.1	44.3	38.6	36.2	15.7	66.1
SEED	44.3	61.9	47.2	41.0	42.0	16.9	67.7
HELICES	43.1	59.8	45.7	37.6	37.4	18.6	54.5
Qwen2.5-7B-Instruct							
GRPO+OPSD	47.3	64.5	46.9	43.8	39.3	18.0	69.4
Skill-SD	47.1	64.5	47.8	44.2	42.1	20.2	69.0
SDAR	46.3	63.5	48.2	43.8	48.4	19.6	73.0
SEED	47.0	64.9	47.8	45.2	45.3	20.1	70.2
HELICES	47.4	64.9	46.6	45.3	44.8	21.8	63.6

Table 13: ALFWorld hint-free success rate (%) over three independent complete training runs. Differences are matched-seed HELICES-minus-Evolving-RL values; the final column reports Seen/Unseen seed wins.

Scale	Method	Seen SR	Unseen SR	ΔSeen	ΔUnseen	Seed wins
3B	GRPO	80.00 ± 1.19	78.11 ± 1.99	—	—	—
3B	Evolving-RL	78.57 ± 1.09	81.26 ± 2.75	0	0	—
3B	**HELICES**	$\mathbf{94.29 \pm 1.80}$	$\mathbf{85.49 \pm 3.31}$	$\mathbf{+15.71 \pm 1.65}$	$\mathbf{+4.23 \pm 0.66}$	**3/3; 3/3**
7B	GRPO	80.00 ± 0.95	78.19 ± 1.37	—	—	—
7B	Evolving-RL	91.43 ± 1.19	93.03 ± 1.74	0	0	—
7B	**HELICES**	$\mathbf{97.14 \pm 0.71}$	$\mathbf{96.27 \pm 1.00}$	$\mathbf{+5.71 \pm 0.48}$	$\mathbf{+3.23 \pm 0.75}$	**3/3; 3/3**

Table 14: Seed-level audit for Table 13. Each cell reports Seen successes out of 420 / Unseen successes out of 402.

Scale	Method	Seed 42	Seed 84	Seed 126
3B	GRPO	341/420; 314/402	331/420; 322/402	336/420; 306/402
3B	Evolving-RL	335/420; 337/402	326/420; 315/402	329/420; 328/402
3B	**HELICES**	**397/420; 355/402**	**388/420; 329/402**	**403/420; 347/402**
7B	GRPO	336/420; 320/402	340/420; 309/402	332/420; 314/402
7B	Evolving-RL	389/420; 381/402	379/420; 367/402	384/420; 374/402
7B	**HELICES**	**411/420; 391/402**	**405/420; 383/402**	**408/420; 387/402**

Table 15: Search-QA hint-free exact match (%) over three independent complete training runs. Differences are matched-seed HELICES-minus-Evolving-RL values; the final column reports single-hop/multi-hop seed wins.

Scale	Method	Single-hop EM	Multi-hop EM	ΔSingle	ΔMulti	Seed wins
3B	GRPO	48.39 ± 0.82	33.45 ± 1.03	—	—	—
3B	Evolving-RL	46.97 ± 0.58	34.28 ± 0.64	0	0	—
3B	**HELICES**	$\mathbf{50.83 \pm 0.94}$	$\mathbf{35.55 \pm 1.18}$	$\mathbf{+3.86 \pm 1.50}$	$\mathbf{+1.27 \pm 0.78}$	**3/3; 3/3**
7B	GRPO	51.76 ± 0.82	40.43 ± 1.03	—	—	—
7B	Evolving-RL	52.29 ± 0.75	34.03 ± 0.85	0	0	—
7B	**HELICES**	$\mathbf{53.76 \pm 1.02}$	$\mathbf{42.63 \pm 1.40}$	$\mathbf{+1.46 \pm 0.32}$	$\mathbf{+8.59 \pm 1.11}$	**3/3; 3/3**

Table 16: Seed-level audit for Table 15. Each cell reports single-hop / multi-hop exact-match successes out of 2,048.

Scale	Method	Seed 42	Seed 84	Seed 126
3B	GRPO	1,004/2,048; 693/2,048	972/2,048; 701/2,048	997/2,048; 661/2,048
3B	Evolving-RL	959/2,048; 716/2,048	975/2,048; 690/2,048	952/2,048; 700/2,048
3B	**HELICES**	**1,050/2,048; 742/2,048**	**1,019/2,048; 700/2,048**	**1,054/2,048; 742/2,048**
7B	GRPO	1,063/2,048; 844/2,048	1,075/2,048; 804/2,048	1,042/2,048; 836/2,048
7B	Evolving-RL	1,084/2,048; 717/2,048	1,054/2,048; 685/2,048	1,075/2,048; 689/2,048
7B	**HELICES**	**1,115/2,048; 890/2,048**	**1,077/2,048; 840/2,048**	**1,111/2,048; 889/2,048**

Table 17: Full-pipeline ALFWorld comparison at different environment budgets. SR is mean $\pm$ sample SD over three independent training runs, in %. Episodes include source collection and auxiliary environment executions; the HELICES entry is an upper bound, with a logged single-run total of 27,600 reported in Table 18.

Configuration	Updates	Policy-update trajectories	Environment episodes	Seen SR	Unseen SR
Evolving-RL, 55 updates	55	28,160	29,040	90.63 ± 1.07	91.79 ± 1.00
Evolving-RL, full	75	38,400	39,600	91.43 ± 1.19	93.03 ± 1.74
HELICES, full	150	19,200	$\leq 28,800$	$\mathbf{97.14 \pm 0.71}$	$\mathbf{96.27 \pm 1.00}$

For Evolving-RL, each update uses 512 policy-update trajectories and 16 additional source episodes, giving $55 \times 528 = 29,040$ and $75 \times 528 = 39,600$ episodes. Its 55-update configuration therefore provides a near-matched interaction comparison: it uses 0.8% more episodes than the HELICES upper bound and 5.2% more than the logged HELICES run. HELICES achieves higher mean Seen and Unseen success at both Evolving-RL budgets. Against the 55-update configuration, the differences in displayed means are 6.51 and 4.48 percentage points, respectively.

Table 18: Measured full-training cost for one logged seed per configuration on eight GPUs of the same model. M denotes million tokens. Timing includes the complete HELICES pipeline, including hint generation, discovery, confirmation, attribution rescoring, and policy optimization. The attribution column counts additional attribution rescoring; zero for Evolving-RL does not exclude its other auxiliary computation.

Configuration	Episodes	Environment actions	Input/output tokens	Attribution tokens	GPU-hours	Wall-clock (h)
Evolving-RL, 55 updates	29,040	464,640	360M / 30M	0	176	22
Evolving-RL, 75 updates	39,600	633,600	490M / 41M	0	240	30
HELICES, full	27,600	386,400	340M / 28M	18M	**160**	**20**

Table 18 records single-seed log measurements, separately from the three-seed performance summaries. On the matched hardware, HELICES uses 160 GPU-hours, compared with 176 and 240 for the two Evolving-RL configurations, reductions of 9.1% and 33.3% in these logged runs. The measured HELICES cost includes its verification and rescoring overhead. Together, the interaction and cost comparisons show that its performance advantage in this experiment is achieved without a larger environment-episode budget or greater measured training time.

D.7 MIND2WEB MULTI-SEED ROBUSTNESS AND INTERNALIZATION ABLATIONS

Independent full-pipeline repetitions. We repeat the complete Qwen2.5-7B-Instruct training pipeline with three independent training seeds (42, 84, and 126). Each final checkpoint is evaluated without hints on the complete Mind2Web manifests described in Appendix D.2. Table 19 reports the mean and sample standard deviation across these runs. For HELICES, the prespecified comparator is our reproduced Evolving-RL pipeline, which uses the same backbone, task manifests, training-update budget, and evaluation rule. HELICES improves weighted AA by 2.57 percentage points over this comparator and has the higher weighted AA in all three matched-seed comparisons. The consistency across complete runs complements the continuation-seed study below: it includes variation from both joint training and verified internalization rather than varying only the final 30 updates.

Table 19: Full-pipeline Mind2Web results for Qwen2.5-7B-Instruct, in %. Entries are mean ± sample SD over three independent training seeds. Weighted summaries are computed within each seed from unrounded split scores and then averaged; they can therefore differ slightly from recomputing the summary from the displayed split means. All evaluations are hint-free.

Method	Cross-task	Cross-website	Cross-domain	Weighted avg.
Action accuracy (AA)				
GRPO	28.79 ± 0.94	24.61 ± 0.87	20.84 ± 0.71	22.83 ± 0.78
Evolving-RL	35.32 ± 0.89	35.15 ± 0.65	24.67 ± 0.50	28.05 ± 0.57
HELICES	$\mathbf{35.33 \pm 1.06}$	$\mathbf{38.32 \pm 1.32}$	$\mathbf{27.83 \pm 1.08}$	$\mathbf{30.62 \pm 1.10}$
Full-task success (SR)				
GRPO	1.59 ± 0.40	0.00 ± 0.00	1.64 ± 0.22	1.42 ± 0.22
Evolving-RL	1.59 ± 0.40	$\mathbf{3.39 \pm 0.56}$	1.21 ± 0.11	1.57 ± 0.22
HELICES	$\mathbf{1.98 \pm 0.40}$	2.82 ± 0.56	$\mathbf{2.08 \pm 0.22}$	$\mathbf{2.16 \pm 0.30}$

Table 20: Seed-level audit for the weighted Mind2Web summaries in Table 19. Each entry is weighted AA / weighted SR (%).

Method	Seed 42	Seed 84	Seed 126
GRPO	22.02 / 1.19	22.90 / 1.42	23.57 / 1.64
Evolving-RL	27.41 / 1.34	28.25 / 1.57	28.50 / 1.79
HELICES	**29.46 / 1.86**	**30.77 / 2.16**	**31.65 / 2.46**

Verified-internalization ablations. Table 21 isolates the second stage using three continuation seeds from the same update-120 checkpoint. All variants receive the same continuation-update budget. Success distillation uses successful trajectories without the HELICES verification and selective reasoning mechanisms; the remaining ablations remove one HELICES component at a time. HELICES produces the largest gain in both weighted AA and weighted SR, and its cross-domain advantage shows that the improvement is not confined to the in-domain split.

Table 21: Mind2Web verified-internalization ablations for Qwen2.5-7B-Instruct, in %. Update-150 entries are mean ± sample SD over three continuation-training seeds. AA@120 is the fixed reference value 28.32, not a per-seed random baseline. For each continuation seed, ΔAA subtracts this same constant from AA@150; its sample SD therefore equals that of AA@150. Reference-estimation uncertainty is not propagated into this descriptive gain statistic. ΔSR compares update 150 with the shared update-120 checkpoint.

Training variant	Fixed AA@120	Avg. AA@150	ΔAA	Domain AA@150	Avg. SR@150	ΔSR
Continue GRPO	28.32	28.55 ± 0.84	$+0.23 \pm 0.84$	25.80 ± 1.35	1.49 ± 0.27	$+0.07$
GRPO + success distillation	28.32	29.05 ± 0.98	$+0.73 \pm 0.98$	26.20 ± 1.30	1.57 ± 0.20	$+0.15$
HELICES w/o confirmation	28.32	29.45 ± 1.00	$+1.13 \pm 1.00$	26.65 ± 1.13	1.72 ± 0.26	$+0.30$
HELICES w/o reasoning gate	28.32	30.05 ± 1.20	$+1.73 \pm 1.20$	27.20 ± 1.05	1.94 ± 0.34	$+0.52$
HELICES	28.32	$\mathbf{30.62 \pm 1.10}$	$\mathbf{+2.30 \pm 1.10}$	$\mathbf{27.83 \pm 1.08}$	$\mathbf{2.16 \pm 0.30}$	$\mathbf{+0.75}$

Mechanism diagnostics. Table 22 summarizes the accepted learning targets in the same continuation runs. Removing confirmation admits substantially more targets but reduces the downstream AA and SR gains in Table 21, consistent with confirmation filtering out less reproducible improvements. Removing the reasoning gate leaves the accepted-target count and composition close to full HELICES, but supervises every reasoning token and yields a smaller gain. Thus, the two components act at different levels: confirmation controls which trajectories enter the learner, whereas the reasoning gate controls which parts of accepted trajectories receive reasoning supervision. These are descriptive mechanism diagnostics rather than causal estimates for individual tokens or targets.

Table 22: Target-composition and supervision diagnostics for the Mind2Web internalization ablations. F2S denotes failure-to-success targets; the two target-type percentages sum to 100%. Reasoning-token retention is measured after gating, while action tokens always receive full supervision.

Variant	Targets/update	F2S	Efficiency	Reasoning retained	Actions
w/o confirmation	27.8 ± 3.6	$65.0 \pm 3.2\%$	$35.0 \pm 3.2\%$	$40.1 \pm 4.5\%$	Full
w/o reasoning gate	18.6 ± 2.4	$72.6 \pm 2.8\%$	$27.4 \pm 2.8\%$	100%	Full
HELICES	$\mathbf{18.2 \pm 2.5}$	$\mathbf{73.1 \pm 2.6\%}$	$\mathbf{26.9 \pm 2.6\%}$	$\mathbf{42.7 \pm 4.1\%}$	**Full**

D.8 Post-hoc stage-comparison protocol

Figure 5 reports ALFWorld Seen success, NQ single-hop EM, HotpotQA multi-hop EM, and Mind2Web Cross-task AA. After each training run has completed, saved checkpoints at 15-update intervals are evaluated in a separate post hoc diagnostic experiment to reconstruct learning progress. These retrospective evaluations are not available to the training procedure and are not used to select the main checkpoint, which is fixed in advance at update 150. ALFWorld uses the 7B Seen evaluations, matching the ALFWorld protocol in Figure 3 and Table 3. Mind2Web measures AA on 252 Cross-task examples; its final score is 35.33%, as reported in Table 2. The displayed first and joint QA evaluations use 253 NQ and 673 HotpotQA questions; final QA endpoints follow Table 1. Within each benchmark, the figure compares the displayed first-stage, joint-training, and internalized checkpoints, with guided and unguided execution shown at the joint checkpoint.

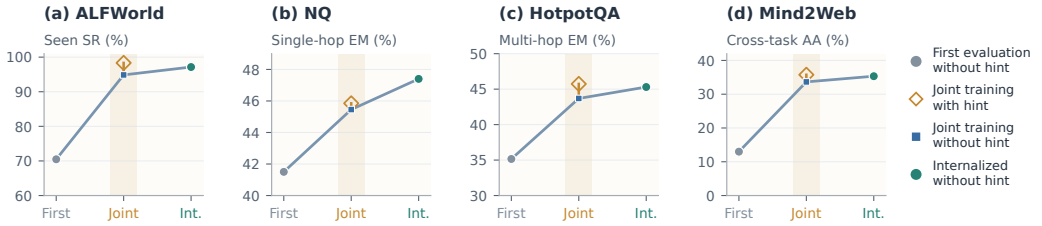


Figure 5: Qwen2.5-7B stage comparison on held-out tasks. ALFWorld reports Seen success; QA and Mind2Web final endpoints follow the main-table evaluation protocol.

Guided performance provides a target for independent learning. Across the four evaluations in Figure 5, final unguided performance approaches or exceeds the displayed joint-stage guided results. This gives the guided reference a learning role: it indicates a performance level that subsequent policy updates can make accessible without hints. This progression is consistent with transferring hint-supported behavior into independent capability through probing and internalization.

Internalization starting checkpoint. All main experiments initialize internalization from joint checkpoint `iter119`, corresponding to 120 completed training updates under zero-based checkpoint numbering. This includes NQ, HotpotQA, and Mind2Web. The Solver then completes 30 internalization updates, and final results are reported after 150 total updates. At this joint checkpoint, NQ achieves 45.45% EM without hints and 45.85% with hints; HotpotQA achieves 43.69% and 45.73%, respectively. Mind2Web Cross-task AA is 33.64% without hints and 35.76% with hints.

Final results. The final hint-free endpoints are the HELICES results in Table 1: 47.4% on NQ and 45.3% on HotpotQA. The final Mind2Web Cross-task AA is 35.33%, as reported in Table 2. Solid

connectors link the displayed unguided results across stages. First and Int. denote the first diagnostic evaluation and the prespecified final internalized checkpoint, respectively; panel scales differ.

D.9 Post-hoc learning-curve protocol

Figure 3 presents ALFWorld and Mind2Web learning curves for first-stage joint training through update 120. These curves are reconstructed post hoc from saved checkpoints in the separate diagnostic evaluation described in Appendix D.8; no curve point is used for checkpoint selection, early stopping, or modification of the main runs. ALFWorld reports overall success on Seen tasks; Mind2Web reports weighted action accuracy across evaluation splits under the protocol of Table 2 and Appendix D.3. Points are unsmoothed; blue annotations show unguided gains from update 0 to update 120 in percentage points. GRPO and Evolving-RL are shown as dashed curves.

Table 3 reports a separate three-seed continuation study, with update 120 as the stage boundary and updates 135 and 150 as second-stage checkpoints. The three branches share the same fixed joint-training checkpoint, and each table entry is their mean; these are not independent repetitions of the first 120 updates. They are distinct from the full-pipeline Mind2Web repetitions in Table 19, which also vary the first-stage run. HELICES performs verified internalization during this interval; the baseline rows report their continued training over the same updates. Unguided HELICES gains another 2.28/2.30 percentage points on ALFWorld/Mind2Web. At update 150, it exceeds Evolving-RL by 5.94/2.57 points and GRPO by 16.84/7.79 points.

Figure 6 separately shows the first 120 joint-training updates of the ALFWorld Solver-weight-1.0 and Mind2Web 7B runs. Faint traces are individual training batches, with nine-step centered moving averages and shorter windows at the boundaries. The ALFWorld run follows Table 6: each batch contains 16 tasks, with one source trajectory and eight downstream Solver trajectories per task, totaling 16 source and 128 downstream trajectories. The Mind2Web run uses 16 source trajectories and an unguided solver fallback when a generated hint is invalid. The training curves report batch-level metrics, while Figure 3 reports held-out evaluation results.

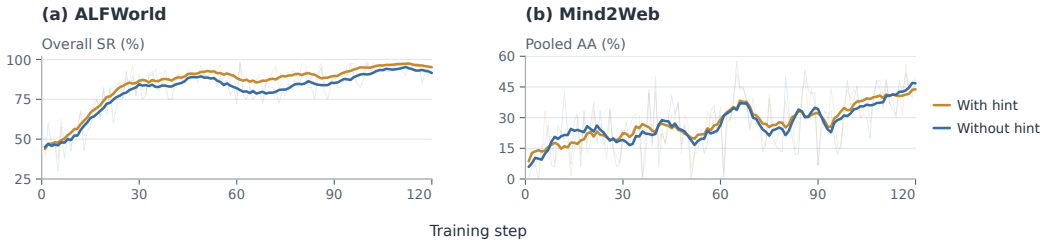


Figure 6: Training trajectories during joint training on ALFWorld and Mind2Web. Solid curves are nine-step centered moving averages; faint traces show the original batch-level observations.

E Ablation Results and Evaluation Protocol

Table 4 reports the completed ALFWorld ablations for Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct; this appendix records the corresponding evaluation protocol and controls. Within each model size, all reported continuation branches start from the same checkpoint obtained after 120 joint hint-training updates on ALFWorld for the corresponding model size. For both 3B and 7B, every variant uses the same three independent continuation-training seeds: 42, 84, and 126. All variants use the same training-trajectory budget. Hyperparameters follow the initial experimental settings and are identical for the 3B and 7B models. For each continuation-training seed, the final checkpoint is evaluated over three passes on the same 140 Seen and 134 Unseen tasks, with one hint-free greedy rollout per task per pass. The three evaluation passes are executed independently with identical sampling parameters; environment random states may differ across passes. For each continuation-training run, task success pools successful and evaluated episodes across the three passes within each split: 420 Seen episodes and 402 Unseen episodes. It is computed from episode counts rather than the mean of category rates. The reported success rates are mean $\pm$ sample standard deviation across the three continuation-training runs, with the variance normalized by $3-1$; SD

is expressed in percentage points and measures variation across training seeds, not across evaluation passes. These repetitions are conditional on the shared joint checkpoint within each model size, rather than independent repetitions of the entire joint-training and continuation pipeline. The five variants are Continue GRPO, GRPO + success distillation, HELICES w/o independent confirmation, HELICES w/o reasoning gate, and HELICES.

Shared controls. All continuation branches retain the same unguided GRPO objective. Auxiliary targets always receive full action supervision; observations receive no loss. "HELICES w/o reasoning gate" removes the reasoning gate while retaining the same reasoning base weight. Success distillation uses successful unguided rollouts without hint-based selection.

Expanded GRPO continuation budget. We extend the Qwen2.5-7B fixed-checkpoint continuation comparison by allowing GRPO 90 additional updates, while HELICES retains 30 verified-internalization updates. Both use 128 policy-update trajectories per update. The expanded GRPO branch therefore receives 11,520 additional environment episodes, matching the HELICES continuation-stage upper bound of 3,840 unguided episodes plus at most 7,680 discovery and confirmation executions. All branches start from the same update-120 joint checkpoint and follow the continuation-seed and evaluation protocol above. The shared first-stage cost is excluded from the incremental budgets in Table 23.

Table 23: ALFWorld continuation with an expanded GRPO interaction budget. SR is mean ± sample SD over three continuation-training seeds, in %. Budgets count only the continuation stage; the HELICES environment count includes a conservative upper bound for discovery and confirmation.

Configuration	Additional updates	Policy-update trajectories	Environment episodes	Seen SR	Unseen SR
Continue GRPO	30	3,840	3,840	94.13 ± 0.55	93.78 ± 0.75
Continue GRPO, expanded	90	11,520	11,520	95.00 ± 0.71	94.53 ± 1.00
HELICES internalization	30	3,840	$\leq 11,520$	$\mathbf{97.14 \pm 0.24}$	$\mathbf{96.27 \pm 0.50}$

Expanding GRPO's continuation budget improves its mean Seen and Unseen success, while HELICES retains advantages of 2.14 and 1.74 percentage points, respectively. This comparison supports the value of verified internalization beyond allocating its interaction budget to additional unguided GRPO updates.

E.1 Attribution-specificity controls for reasoning supervision

We test whether attribution-based token selection improves learning beyond sparsifying or down-weighting reasoning supervision. This separate ALFWorld mechanism study uses shared, independently confirmed auxiliary targets and frozen attribution scores, whereas the main HELICES procedure refreshes both during training.

Shared targets and training conditions. At each model scale, all branches start from the same update-120 joint checkpoint and complete 30 continuation updates. They share target text, token IDs, environment histories, hint-free inputs, F2S/efficiency labels and sample weights, and action, reasoning, and valid generation masks. A common fixed scoring Solver precomputes attribution scores and gate weights. Target order is matched within each of five paired continuation seeds. Loss normalization, the internalization coefficient, and all other training settings are shared. Each branch updates its own Solver and collects its own on-policy GRPO trajectories; only the reasoning-weighting rule is deliberately varied.

Matched supervision rules. For one target trajectory, let $g = (g_1, \ldots, g_R)$ be the original gate vector at its R valid reasoning positions, and let $S = \sum_{t=1}^{R} g_t$. We compare the following rules:

- **A: Attribution gate** uses the original vector g.
- **B: Shuffled gate** randomly permutes g within the same trajectory's valid reasoning positions, including all zero weights. It preserves the number of nonzero weights, their sum, and their distribution, while changing their assignment to tokens.

- **C: Equal-total-weight uniform supervision** assigns S/R to every valid reasoning position, preserving the total gate weight separately for each trajectory.
- **D: Ungated reasoning supervision** assigns gate weight 1 to every valid reasoning position.
- **E: Action-only supervision** assigns gate weight 0 to every reasoning position.

The reasoning base coefficient 0.25 multiplies these gate weights in every group. Action tokens always receive weight 1, and environment observations receive no supervision. All groups retain the same token-mean denominator over model-generated positions, including positions with zero gate weight. In particular, B matches both sparsity and the weight distribution of A, whereas C matches its total reasoning-supervision weight without selecting individual reasoning tokens.

Table 24: **Reasoning-gate specificity on ALFWorld.** Hint-free success rate (%), mean ± sample SD over five paired continuation-training seeds. All variants use shared fixed auxiliary targets and frozen attribution scores. Bold marks the highest mean in each column.

	Qwen2.5-3B-Instruct		Qwen2.5-7B-Instruct	
Reasoning supervision	Seen SR	Unseen SR	Seen SR	Unseen SR
D: Ungated	93.52 ± 0.91	83.63 ± 1.63	96.14 ± 0.68	95.32 ± 1.13
E: Action-only	92.90 ± 0.94	82.64 ± 1.34	95.71 ± 0.75	94.88 ± 1.09
C: Equal-total-weight uniform	93.57 ± 0.87	83.28 ± 1.30	96.43 ± 0.73	95.27 ± 0.91
B: Shuffled gate	93.48 ± 0.96	83.03 ± 1.52	96.33 ± 0.62	95.17 ± 1.03
A: Attribution gate	$\mathbf{94.00 \pm 1.11}$	$\mathbf{84.33 \pm 1.59}$	$\mathbf{96.81 \pm 0.73}$	$\mathbf{95.92 \pm 1.09}$

Evaluation and paired uncertainty. Each final checkpoint is evaluated without hints using greedy decoding over three passes on 140 Seen and 134 Unseen tasks. Within each training seed, we pool successes across the three passes, giving 420 Seen and 402 Unseen episodes. Table 24 reports the mean and sample standard deviation across five continuation-training seeds. For an Unseen comparison between A and a control V, let $k_{A,i}$ and $k_{V,i}$ be their pooled success counts for paired seed i. We compute

$$d_i = 100\frac{k_{A,i} - k_{V,i}}{402}, \qquad \bar{d} = \frac{1}{5}\sum_{i=1}^{5} d_i, \qquad s_d = \sqrt{\frac{1}{4}\sum_{i=1}^{5}(d_i - \bar{d})^2}. \tag{62}$$

Differences are in percentage points. The two-sided 95% paired t interval is $\bar{d} \pm t_{0.975,4}s_d/\sqrt{5}$, with $t_{0.975,4} \simeq 2.776$. These are pointwise intervals for individual comparisons; the independent statistical units are the five training-seed pairs. Positive-seed counts record $d_i > 0$. Differences are computed before rounding, rather than by subtracting displayed rounded means. The uncertainty describes continuation training from the shared joint checkpoint with the fixed auxiliary targets.

Table 25: **Paired Unseen gains from attribution gating.** Mean differences and two-sided 95% paired t intervals are in percentage points. Positive seeds count strictly positive differences out of five.

	Qwen2.5-3B-Instruct			Qwen2.5-7B-Instruct		
Comparison	Mean gain	95% CI	Positive seeds	Mean gain	95% CI	Positive seeds
A − B	$+1.29$	$[+0.56, +2.03]$	5/5	$+0.75$	$[+0.26, +1.23]$	5/5
A − C	$+1.04$	$[+0.31, +1.78]$	5/5	$+0.65$	$[+0.13, +1.16]$	4/5
A − E	$+1.69$	$[+0.73, +2.65]$	5/5	$+1.04$	$[+0.79, +1.30]$	5/5

Results. Attribution gating gives the highest mean Seen and Unseen success at both scales (Table 24). Its Unseen gains over shuffled gating are 1.29/0.75 points for 3B/7B, positive in all five seeds; gains over equal-total-weight uniform supervision are 1.04/0.65 points, positive in five/four seeds. The corresponding pointwise 95% intervals exclude zero (Table 25). These controls support attribution-guided supervision beyond random sparsification or uniform downweighting. Gains of 1.69/1.04 points over action-only training also support retaining selected reasoning alongside full action supervision under this fixed-target protocol.

F FIXED-PROBE CROSS-EVALUATION AND END-TO-END CONTROLS

This section separates changes in hint generation, hint utilization, source experience, and training schedule on ALFWorld. The fixed-probe experiments use the Qwen2.5-3B checkpoints at the joint endpoint (Pre, update 70) and the following internalization endpoint (Post, update 75) of the same alternating cycle. Because HELICES uses a shared model, either checkpoint can be used in the Generator or Solver role. The end-to-end controls are separate complete training runs from the base checkpoint.

F.1 FIXED-SOURCE, CACHED-HINT GENERATOR–SOLVER CROSSING

For each of the 140 Seen and 134 Unseen tasks, we cache one source trajectory and its feedback from the Pre checkpoint. Each Generator then samples four hints from the same cached sources. The resulting 1,096 task–hint pairs are frozen before Solver evaluation, and exactly the same hint text is reused for the Pre and Post Solvers. Each task–hint pair is evaluated on three fresh paired seeds disjoint from source generation and training, giving 1,680 Seen and 1,608 Unseen slots per cell. Guided and unguided executions in a slot share the task initialization and evaluation seed.

The primary analysis is intention-to-treat (ITT). A protocol-invalid hint does not trigger guided execution and is assigned guided success zero; it therefore remains in the 3,288-slot denominator and contributes zero environment steps. Valid rate is instead computed over the 1,096 generated hints. Unless otherwise stated, success, gain, F2S, and harm rates average the Seen and Unseen split rates equally. F2S denotes unguided failure followed by guided success, harm denotes unguided success followed by guided failure, and their difference equals the signed guided–unguided gain within each split. Reproduction is conditional on discovery-qualified candidates: it is the fraction that satisfy the same original F2S or efficiency criterion under fresh evaluation. "Any-of-4 exposed" is the fraction of tasks that fail the unguided 3/3 mastery rule but for which at least one of the four cached hints achieves 3/3 guided success. This diagnostic depends on the four-probe budget and is not directly comparable to single-probe coverage.

Table 26: **Fixed-source, cached-hint 2×2 cross-evaluation.** Primary ITT results; rates are Seen/Unseen-balanced percentages. Gain, F2S, and harm are in percentage points. Reprod. is conditional on discovery-qualified candidates, whereas valid rate and Any-of-4 exposed use all generated hints and all tasks, respectively.

Generator	Solver	SR_0	SR_h	Gain	F2S	Harm	Valid	Reprod.	Any-of-4 exposed
Pre	Pre	80.74	82.91	+2.16	5.06	2.89	93.98	57.00	11.31
Post	Pre	80.74	84.49	+3.75	5.88	2.14	95.99	64.48	14.60
Pre	Post	83.17	86.70	+3.54	6.40	2.86	93.98	62.50	13.87
Post	Post	83.17	89.78	+6.62	8.66	2.04	95.99	71.30	18.61

Holding the Pre Solver fixed, moving from the Pre to Post Generator state increases the ITT gain by 1.58 points. Holding the Pre Generator fixed, moving to the Post Solver increases it by 1.37 points. The corresponding effects at the Post Solver and Post Generator are 3.08 and 2.87 points, respectively, yielding a positive Generator–Solver interaction of 1.50 points. The Post Generator was not directly optimized with a Generator objective during internalization; this comparison therefore measures the effect of reusing the post-internalization shared parameter state in the Generator role, rather than direct Generator training during that stage.

Table 27: **Valid-executed diagnostic and interaction cost.** Valid-only results condition on a Generator-dependent subset and are diagnostic rather than primary comparisons. Steps/attempt divides by all 3,288 ITT slots; steps/executed divides by the number of valid guided executions.

G/S	Valid hints	Executed	SR_0	SR_h	Gain	Steps	Steps/attempt	Steps/executed	Exposed
Pre/Pre	1,030/1,096	3,090	84.46	88.22	+3.76	75,120	22.85	24.31	31/274
Post/Pre	1,052/1,096	3,156	83.17	88.02	+4.85	73,120	22.24	23.17	40/274
Pre/Post	1,030/1,096	3,090	86.81	92.26	+5.44	71,080	21.62	23.00	38/274
Post/Post	1,052/1,096	3,156	85.47	93.54	+8.07	68,080	20.71	21.57	51/274

Table 28: **Exact fixed-probe outcome counts.** Seen and Unseen denominators are 1,680 and 1,608 ITT slots. Valid-subset denominators depend only on the Generator state.

G/S	ITT SR_0 (S; U)	Valid SR_0 (S; U)	SR_h (S; U)	F2S (S; U)	Harm (S; U)
Pre/Pre	1,384; 1,272	1,359/1,578; 1,252/1,512	1,440; 1,288	101; 66	45; 50
Post/Pre	1,384; 1,272	1,368/1,614; 1,258/1,542	1,465; 1,315	112; 82	31; 39
Pre/Post	1,432; 1,304	1,404/1,578; 1,280/1,512	1,485; 1,367	101; 109	48; 46
Post/Post	1,432; 1,304	1,413/1,614; 1,286/1,542	1,530; 1,423	131; 153	33; 34

Table 29: **Exact fixed-probe Generator and diagnostic counts.** Valid-hint counts use 560 Seen and 536 Unseen task–hint pairs. Reproduction uses its discovery-qualified denominator.

G/S	Valid hints (S; U)	Reprod.	Any-of-4 exposed	Steps
Pre/Pre	526/560; 504/536	513/900	31/274	75,120
Post/Pre	538/560; 514/536	648/1,005	40/274	73,120
Pre/Post	526/560; 504/536	600/960	38/274	71,080
Post/Post	538/560; 514/536	770/1,080	51/274	68,080

F.1.1 Source- and task-stratified diagnostics

Tables 30–32 stratify the $G_{\text{post}}/S_{\text{post}}$ cell under the cached Pre-source probe set. They are descriptive diagnostics and do not alter the primary ITT estimand. Rates pool attempted slots within each displayed stratum rather than rebalancing Seen and Unseen. In particular, failed sources leave more baseline headroom than successful sources.

Table 30: **Post/Post results by cached Pre-source outcome.**

Source	Tasks	Slots	SR_0	SR_h	Gain	F2S	Harm	Valid	Exposed	Steps/valid
Failure	170	2,040	78.43	88.73	+10.29	12.50	2.21	96.32	24.12	22.20
Success	104	1,248	91.03	91.59	+0.56	2.32	1.76	95.43	9.62	20.53

Among all attempted slots, F2S accounts for 12.50% after failed sources and 2.32% after successful sources. Relative to the corresponding unguided failure mass, these values recover approximately 58.0% and 25.9% of failures, respectively. The comparison remains descriptive because source outcome is associated with task difficulty and trajectory content.

Table 31: **Post/Post results by ALFWorld task type.**

Type	Tasks	Slots	SR_0	SR_h	Gain	F2S	Harm	Valid	Exposed	Steps/valid
Pick	59	708	86.16	93.22	+7.06	8.47	1.41	96.19	20.34	21.20
Look	31	372	83.33	89.52	+6.18	7.80	1.61	95.97	16.13	21.00
Clean	58	696	83.33	91.24	+7.90	9.63	1.72	96.12	18.97	21.40
Heat	39	468	81.20	90.38	+9.19	11.11	1.92	96.15	20.51	21.90
Cool	46	552	82.43	88.77	+6.34	8.15	1.81	96.20	19.57	21.50
Pick2	41	492	81.50	83.74	+2.24	6.30	4.07	95.12	14.63	22.56

Table 32: **Post/Post results by cached Pre-source termination.**

Termination	Tasks	Slots	SR_0	SR_h	Gain	F2S	Harm	Valid	Exposed	Steps/valid
Success	104	1,248	91.03	91.59	+0.56	2.32	1.76	95.43	9.62	20.53
Step limit	90	1,080	78.70	89.81	+11.11	13.43	2.31	96.67	25.56	22.03
Invalid action/loop	50	600	78.33	89.17	+10.83	12.83	2.00	96.00	24.00	22.05
Other failure	30	360	77.78	84.72	+6.94	9.17	2.22	95.83	20.00	22.97

F.2 Generator parameter state by source state

We next hold the Post Solver fixed and cross the Pre/Post Generator parameter state with Pre/Post source experience. Each cell first generates and caches 1,096 hints, which are then evaluated with the same 3,288-slot ITT protocol. The unguided baseline is common to all four cells: 1,432/1,680 Seen and 1,304/1,608 Unseen successes, or 83.17% under split-balanced reporting.

Table 33: **Generator-state by source-state factorial evaluation.** The Solver is fixed at Post. Reprod. and Any-of-4 exposed follow the conditional and task-level definitions above.

Generator	Source	SR_h	Gain	F2S	Harm	Valid	Reprod.	Exposed	Steps/attempt	Steps/valid
Pre	Pre	86.70	+3.54	6.40	2.86	93.98	62.50	13.87	21.62	23.00
Post	Pre	89.78	+6.62	8.66	2.04	95.99	71.30	18.61	20.71	21.57
Pre	Post	87.71	+4.54	6.85	2.31	94.89	66.67	16.06	21.24	22.38
Post	Post	91.53	+8.36	9.82	1.46	96.72	73.68	21.17	20.07	20.75

At the Pre source state, the Generator parameter-state effect on gain is 3.08 points; at the Post source state it is 3.82 points. Holding the Pre Generator fixed, the source-state effect is 1.00 point, compared with 1.74 points for the Post Generator. The difference-in-differences interaction is therefore 0.74 points. This is a source-conditioned parameter-state comparison: changes in source success, termination, and trajectory content remain possible mediators of the source-state effect.

Table 34: **Exact Generator-by-source factorial counts.** Guided, F2S, and harm entries are Seen; Unseen. Valid-hint denominators are 560 and 536; reproduction uses its discovery-qualified denominator.

Generator	Source	Guided (S; U)	F2S	Harm	Valid hints	Reprod.	Exposed / Steps
Pre	Pre	1,485; 1,367	101; 109	48; 46	526; 504	600/960	38/274 / 71,080
Post	Pre	1,530; 1,423	131; 153	33; 34	538; 514	770/1,080	51/274 / 68,080
Pre	Post	1,502; 1,383	108; 117	38; 38	532; 508	660/990	44/274 / 69,840
Post	Post	1,550; 1,460	142; 180	24; 24	542; 518	840/1,140	58/274 / 66,000

F.3 Complete-pipeline schedule and co-evolution controls

Each run in this subsection starts independently from the base checkpoint with its own training shuffle and rollout randomness; the runs do not share a joint checkpoint. The five variants use the following prespecified mechanisms:

- **Full HELICES**: six cycles of 20 joint and 5 internalization updates; after every transition, the updated shared parameters reassume both roles.
- **Frozen Generator**: cycle 1 is unchanged, after which the Generator snapshot at J1/update 20 is frozen for cycles 2–6 while the Solver continues to update.
- **Separate G/S**: two complete models start from the same base weights. The Generator receives only the Generator objective and the Solver receives the Solver and internalization objectives; parameters are not shared.
- **Static probes after J1**: task–source–hint tuples are cached at J1 and reused in cycles 2–6. The same execution slots are retained for environment-budget matching, but newly collected source evidence does not refresh the probes. This variant was operationally named "no feedback loop"; it does not remove environment observations from the Solver.
- **Single switch**: 120 joint updates followed by 30 consecutive internalization updates, with the entire pipeline rerun independently for every seed.

Each final checkpoint is evaluated over three passes on 140 Seen and 134 Unseen tasks, giving 420 and 402 executions per seed. Means and sample SDs are across the three complete training runs. Pooled counts are reported for auditability but are not treated as 1,260 or 1,206 independent training replicates.

Full HELICES exceeds Frozen Generator by 1.67/3.57 Seen/Unseen points and Static probes after J1 by 2.38/7.05 points on the three-seed means. Relative to Separate G/S, its mean advantage is smaller, 0.87/1.82 points, and the separate models are slightly better in seed 2; this comparison therefore provides only descriptive, non-uniform evidence for parameter sharing. The separate variant also uses two complete parameter sets, so environment-budget matching does not imply matched compute.

The repeated-cycle schedule improves both splits in all three paired runs, with the larger mean difference on Unseen tasks. These three-run summaries describe training-seed variation; with only

Table 35: **Complete-pipeline ALFWorld controls over three independent training seeds.** Each seed entry stacks Seen above Unseen successful executions. Mean ± SD is in percent.

Variant	Seed 1	Seed 2	Seed 3	Seen	Unseen	Pooled Seen	Pooled Unseen
	396/420	400/420	403/420				
Full HELICES	354/402	361/402	367/402	95.16 ± 0.84	89.72 ± 1.62	1,199/1,260	1,082/1,206
	390/420	396/420	392/420				
Frozen Generator	342/402	351/402	346/402	93.49 ± 0.73	86.15 ± 1.12	1,178/1,260	1,039/1,206
	392/420	401/420	395/420				
Separate G/S	348/402	362/402	350/402	94.29 ± 1.09	87.89 ± 1.88	1,188/1,260	1,060/1,206
	386/420	394/420	389/420				
Static probes after J1	327/402	339/402	331/402	92.78 ± 0.96	82.67 ± 1.52	1,169/1,260	997/1,206
	392/420	399/420	394/420				
Single switch	340/402	336/402	343/402	94.05 ± 0.86	84.49 ± 0.87	1,185/1,260	1,019/1,206

Table 36: **Paired Full-minus-Single-switch differences.** Each pair uses the corresponding independent complete-pipeline seed.

Split	Seed 1	Seed 2	Seed 3	Paired mean ± SD
Seen	+4 (+0.95 pp)	+1 (+0.24 pp)	+9 (+2.14 pp)	$+1.11 \pm 0.96$ pp
Unseen	+14 (+3.48 pp)	+25 (+6.22 pp)	+24 (+5.97 pp)	$+5.22 \pm 1.51$ pp

three complete runs, we emphasize effect sizes and directional replication rather than a large-sample significance claim.

G COMPLETE ALFWORLD CAPABILITY FRONTIER RESULTS

G.1 SCHEDULE, MEASUREMENT, AND INTERPRETATION

This complementary ALFWorld experiment alternates six cycles of 20 joint updates and 5 verified-internalization updates, for 150 optimizer updates (120 joint and 30 internalization). It is separate from the single-switch schedule used for the main benchmark comparison. Joint stages cover updates 0–20, 25–45, 50–70, 75–95, 100–120, and 125–145; internalization covers 20–25, 45–50, 70–75, 95–100, 120–125, and 145–150. Evaluation covers the original base models at update 0 and all 12 stage endpoints. Stage summaries include all six joint and six internalization transitions. The update-0 results come from a separate baseline evaluation on the same task set, with the loaded weights verified to match the original base models and no optimizer updates. Task paths align this baseline with the task IDs in the training export. Joint and internalization stages differ in duration and starting capability, so their pooled gains do not identify a causal or per-update efficiency advantage of one stage over the other.

The runs start from the original base models. Within each joint block, the model produced by one update generates the source trajectories and hints for the next update. At every internalization-to-joint transition, the internalized Solver weights become the shared model weights for both hint generation and utilization in the next joint block; training does not resume from the pre-internalization frozen generator. AdamW optimizer moments, step counters, and FP32 master parameters are preserved across stage transitions in both directions. Joint training and internalization use constant learning rates of 10^{-6} and 2×10^{-7}, respectively, with internalization coefficient 0.005. All alternating results reported here come from the September 17, 2026 continuous-Adam rerun with this state-preserving transition protocol. The earlier September 15 parameter-only alternating run is not included. The Generator weight during joint training is 0.05 for both 3B and 7B; the Solver weight is 1. Both stages use rollout batches of 16 tasks. Joint/internalization samples per prompt are 1/8. The KL coefficient is 0.01 for each optimized role. The Generator is frozen during internalization. The Generator temperature is 1.0 during joint training and 0.7 during internalization. Both model scales use the shared sampling, clipping, entropy, and gating settings in Table 9 and the optimizer and update settings in Table 10.

The two model scales each have 140 Seen and 134 Unseen tasks, evaluated three times per checkpoint and hint condition. The complete numeric export and the added baseline provide 7,124 task–checkpoint rows, each containing both branch success counts (42,744 branch-level executions in total). We reconstruct all point estimates from these counts, aligning the exported task IDs within each split across checkpoints. The original rollouts are not re-executed in this analysis. Every success or coverage percentage that combines Seen and Unseen weights the splits equally; integer set sizes pool all 274 tasks. A 3/3 task is called empirically mastered. This is the plug-in implementation of $\alpha = 0.8$, not evidence that the population success probability has been established to exceed 0.8. Three executions give coarse membership estimates, so we also report all nontrivial success-count cutoffs.

For consecutive endpoints, write $P = \widehat{\mathcal{C}}^0_{\mathrm{pre}}$, $Q = \widehat{\mathcal{C}}^0_{\mathrm{post}}$, and $E = \widehat{\mathcal{C}}^h_{\mathrm{pre}} \setminus P$. Acquisition is $Q \setminus P$, forgetting is $P \setminus Q$, and retention is $P \cap Q$. Acquisition partitions into *frontier transfer*, $Q \cap E$, and *unexposed leap*, $Q \setminus (P \cup \widehat{\mathcal{C}}^h_{\mathrm{pre}})$. These are the empirical task sets underlying ρ^{tr}_k and ρ^{leap}_k in Appendix A.4. The orange inset shading in Figure 4 shows the guidance-revealed capacity gap, defined here as the positive part of the guided–unguided coverage difference. Since the two capability regions need not be nested, this difference is the mass of hint-only tasks minus that of unguided-only tasks, rather than the mass of the hint-exposed region $\mathcal{B}_k$ itself. The exact accounting identity is

$$|Q| - |P| = |Q \cap E| + |Q \setminus (P \cup \widehat{\mathcal{C}}^h_{\mathrm{pre}})| - |P \setminus Q|. \tag{63}$$

Acquisition rates divide by the corresponding previously unmastered candidate count; forgetting and retention rates divide by prior mastery. When pooling stages, counts are task–stage events: a task may be acquired more than once. Distinct-task totals are reported separately. This accounting identity does not independently test a causal learning mechanism or the projection bound in the theory.

Uncertainty and scope. We recompute 95% percentile bootstrap intervals with 10,000 replicates and seed 20260918 from the supplied task-level counts. Each replicate resamples 140 Seen and 134 Unseen task IDs within their respective splits; all branches and checkpoints for a sampled task stay together. Thus, repeated appearances of a task are not treated as independent observations. These intervals describe task-resampling variation conditional on the recorded runs and executions. They do not capture training-seed variation or certify population capability membership. The alternating runs use stage-specific evaluation seed handling, and the generator observes evolving source trajectories; adjacent-stage comparisons are not fresh common-seed policy evaluations or the fixed cached-probe comparison of Appendix A. The trajectories document associations across checkpoints; neither the stage schedule nor hint exposure was randomly assigned for this audit.

Interpreting cumulative capability growth. The capability ratchet describes cumulative acquisition with substantial retention: acquired behaviors provide the starting point for subsequent experience extraction. Individual stages can reverse as acquisition and forgetting change, consistent with the balance in Equation 7. At high coverage, fewer unmastered tasks remain, and the smaller late-stage changes are consistent with a performance plateau on the evaluated task set. The positive pooled internalization estimates have intervals spanning zero (Table 52), so their contribution is interpreted alongside the complete-pipeline controls. Comparisons with frozen generators, static probes, and a single-switch schedule in Appendix F.3 provide complementary evidence for the role of adaptive feedback and repeated internalization in cumulative independent capability growth.

Table 37: **Acquisition and transfer in both stages.** Counts pool task–stage events. Hint-exposed tasks were mastered only with hints; fractions are acquired/eligible (%). J/I: joint/internalization, six stages each.

Model	Stage	New	Lost	Net	Hint-exposed	Other unmastered
3B	J	289	65	+224	69/99 (69.7)	220/593 (37.1)
3B	I	86	68	+18	38/106 (35.8)	48/362 (13.3)
7B	J	229	59	+170	76/89 (85.4)	153/254 (60.2)
7B	I	51	44	+7	36/78 (46.2)	15/95 (15.8)

G.2 Complete checkpoint coverage and success

Table 38 reports every checkpoint. The union is the set of tasks mastered under at least one condition; it is not a single executable policy. Guided and unguided sets need not be nested. For example, 3B at update 20 has higher guided mean success but lower guided 3/3 coverage.

Table 38: **All 13 checkpoints per model.** SR and coverage (Cov.) are Seen/Unseen-balanced percentages. $|C^0|$, $|C^h|$, $|U|$, $|B|$, and $|O|$ count no-hint, hinted, union, hint-only, and no-hint-only 3/3 tasks out of 274. The intersection is $|C^0| - |O|$. J/I denote joint/internalization.

Model	End	Step	SR_0	SR_h	$\lvert C^0\rvert$	$\lvert C^h\rvert$	$\lvert U\rvert$	$\lvert B\rvert$	$\lvert O\rvert$	$Cov._0$	$Cov._h$
3B	Base	0	9.71	9.82	7	3	10	3	7	2.56	1.09
3B	J1	20	44.64	46.23	89	62	103	14	41	32.38	22.59
3B	I1	25	41.44	47.16	81	75	106	25	31	29.50	27.31
3B	J2	45	56.28	58.90	147	134	172	25	38	53.59	48.88
3B	I2	50	59.92	60.19	156	143	182	26	39	56.87	52.16
3B	J3	70	80.74	82.91	215	209	235	20	26	78.42	76.21
3B	I3	75	83.17	86.10	222	220	239	17	19	80.96	80.27
3B	J4	95	87.84	90.53	236	242	255	19	13	86.12	88.22
3B	I4	100	89.25	90.80	243	243	257	14	14	88.64	88.59
3B	J5	120	89.69	90.29	242	241	255	13	14	88.22	87.86
3B	I5	125	89.95	90.41	243	243	257	14	14	88.61	88.58
3B	J6	145	91.65	93.72	247	251	262	15	11	90.07	91.51
3B	I6	150	92.52	94.45	249	256	264	15	8	90.80	93.35
7B	Base	0	43.61	41.45	88	73	111	23	38	32.13	26.55
7B	J1	20	78.36	77.88	195	183	219	24	36	71.15	66.76
7B	I1	25	76.99	78.50	191	189	217	26	28	69.69	69.02
7B	J2	45	89.77	92.94	236	242	257	21	15	86.11	88.28
7B	I2	50	91.45	93.18	239	238	254	15	16	87.16	86.82
7B	J3	70	95.52	97.45	256	263	270	14	7	93.46	96.01
7B	I3	75	96.04	97.93	258	263	270	12	7	94.24	95.98
7B	J4	95	96.48	97.54	262	265	268	6	3	95.62	96.67
7B	I4	100	96.95	98.05	263	266	269	6	3	95.98	97.08
7B	J5	120	95.31	97.01	260	264	266	6	2	94.82	96.27
7B	I5	125	95.80	97.66	262	266	269	7	3	95.55	97.06
7B	J6	145	96.92	98.03	262	267	269	7	2	95.57	97.44
7B	I6	150	96.92	98.16	265	265	270	5	5	96.67	96.71

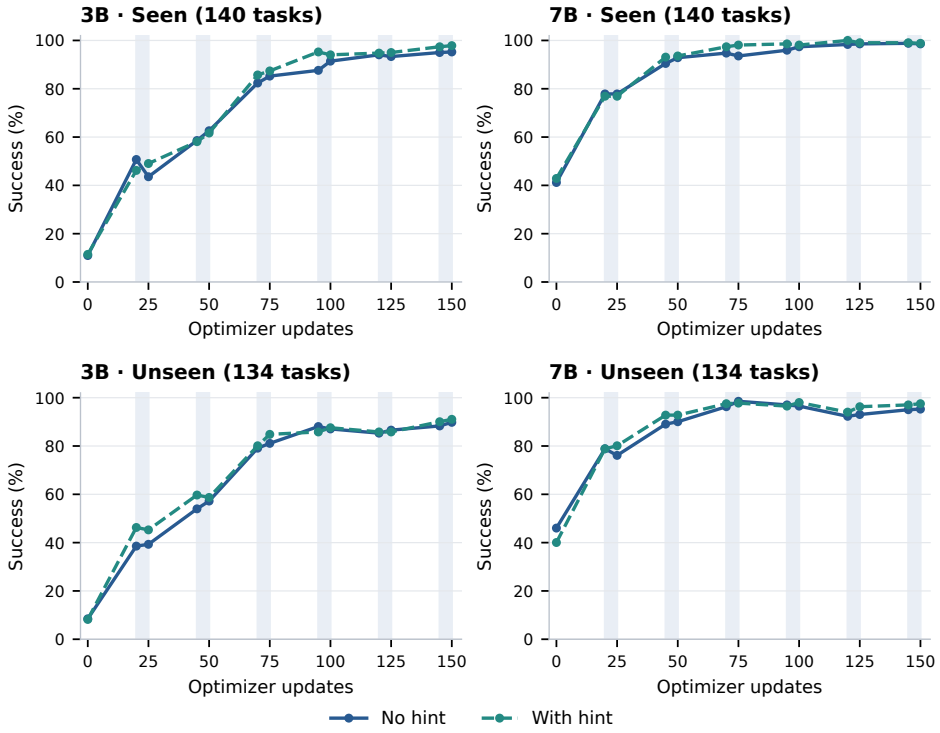


Figure 7: **Complete split-level success trajectories.** Every plotted point averages three executions per task. Pale bands mark the six 5-update internalization stages; intervening stages each contain 20 joint updates. Update 0 evaluates the original base weights.

Table 39: **Exact split-level counts.** Each condition lists successful executions/total executions and mastered tasks (M). Seen has 140 tasks and 420 executions; Unseen has 134 tasks and 402 executions. Divide M by the corresponding task count to recover coverage.

Model	Step	Seen 0	M	Seen h	M	Unseen 0	M	Unseen h	M
3B	0	46/420	3	48/420	2	34/402	4	33/402	1
3B	20	213/420	52	194/420	34	155/402	37	186/402	28
3B	25	183/420	45	206/420	42	158/402	36	182/402	33
3B	45	246/420	79	244/420	70	217/402	68	240/402	64
3B	50	263/420	84	259/420	75	230/402	72	236/402	68
3B	70	346/420	113	360/420	111	318/402	102	322/402	98
3B	75	358/420	117	367/420	114	326/402	105	341/402	106
3B	95	368/420	121	400/420	130	354/402	115	345/402	112
3B	100	384/420	127	395/420	130	350/402	116	352/402	113
3B	120	395/420	130	398/420	129	343/402	112	345/402	112
3B	125	392/420	129	399/420	131	348/402	114	345/402	112
3B	145	399/420	131	409/420	134	355/402	116	362/402	117
3B	150	400/420	132	411/420	136	361/402	117	366/402	120
7B	0	173/420	44	180/420	43	185/402	44	161/402	30
7B	20	327/420	101	323/420	95	317/402	94	317/402	88
7B	25	327/420	99	323/420	94	306/402	92	322/402	95
7B	45	380/420	122	391/420	126	358/402	114	373/402	116
7B	50	390/420	126	393/420	124	362/402	113	373/402	114
7B	70	398/420	129	409/420	133	387/402	127	392/402	130
7B	75	393/420	127	412/420	135	396/402	131	393/402	128
7B	95	403/420	134	414/420	138	390/402	128	388/402	127
7B	100	409/420	135	412/420	136	388/402	128	394/402	130
7B	120	413/420	137	420/420	140	371/402	123	378/402	124
7B	125	414/420	138	416/420	137	374/402	124	387/402	129
7B	145	415/420	137	416/420	137	382/402	125	390/402	130
7B	150	414/420	138	415/420	136	383/402	127	392/402	129

G.3 Every adjacent-stage acquisition and forgetting transition

All following transitions use the strict 3/3 criterion. Acquisitions are not unique tasks across stages. The original-base evaluation supplies the start of the first joint stage.

Table 40: **3B: all stage transitions under both hint conditions.** Branch 0 is no hint; branch 1 is with hint. Before/after, new/lost, and kept are task counts. ΔCov. and ΔSR are domain-balanced percentage-point changes.

Stage	Updates	Branch	Before	After	New	Lost	Kept	Net	ΔCov.	ΔSR
J1	0–20	0	7	89	84	2	5	+82	+29.81	+34.93
J1	0–20	1	3	62	60	1	2	+59	+21.50	+36.41
I1	20–25	0	89	81	20	28	61	-8	-2.87	-3.20
I1	20–25	1	62	75	32	19	43	+13	+4.72	+0.93
J2	25–45	0	81	147	83	17	64	+66	+24.08	+14.84
J2	25–45	1	75	134	80	21	54	+59	+21.57	+11.74
I2	45–50	0	147	156	22	13	134	+9	+3.28	+3.64
I2	45–50	1	134	143	25	16	118	+9	+3.28	+1.29
J3	50–70	0	156	215	75	16	140	+59	+21.55	+20.83
J3	50–70	1	143	209	78	12	131	+66	+24.05	+22.72
I3	70–75	0	215	222	16	9	206	+7	+2.55	+2.42
I3	70–75	1	209	220	24	13	196	+11	+4.06	+3.20
J4	75–95	0	222	236	23	9	213	+14	+5.16	+4.67
J4	75–95	1	220	242	28	6	214	+22	+7.95	+4.43
I4	95–100	0	236	243	10	3	233	+7	+2.52	+1.41
I4	95–100	1	242	243	11	10	232	+1	+0.37	+0.28
J5	100–120	0	243	242	11	12	231	-1	-0.42	+0.44
J5	100–120	1	243	241	15	17	226	-2	-0.73	-0.51
I5	120–125	0	242	243	9	8	234	+1	+0.39	+0.26
I5	120–125	1	241	243	7	5	236	+2	+0.71	+0.12
J6	125–145	0	243	247	13	9	234	+4	+1.46	+1.70
J6	125–145	1	243	251	18	10	233	+8	+2.94	+3.30
I6	145–150	0	247	249	9	7	240	+2	+0.73	+0.87
I6	145–150	1	251	256	9	4	247	+5	+1.83	+0.74

Table 41: **7B: all stage transitions under both hint conditions.** Branch 0 is no hint; branch 1 is with hint. Before/after, new/lost, and kept are task counts. ΔCov. and ΔSR are domain-balanced percentage-point changes.

Stage	Updates	Branch	Before	After	New	Lost	Kept	Net	ΔCov.	ΔSR
J1	0–20	0	88	195	118	11	77	+107	+39.01	+34.75
J1	0–20	1	73	183	116	6	67	+110	+40.21	+36.43
I1	20–25	0	195	191	18	22	173	-4	-1.46	-1.37
I1	20–25	1	183	189	32	26	157	+6	+2.25	+0.62
J2	25–45	0	191	236	57	12	179	+45	+16.42	+12.78
J2	25–45	1	189	242	67	14	175	+53	+19.26	+14.44
I2	45–50	0	236	239	13	10	226	+3	+1.06	+1.69
I2	45–50	1	242	238	16	20	222	-4	-1.46	+0.24
J3	50–70	0	239	256	25	8	231	+17	+6.30	+4.06
J3	50–70	1	238	263	28	3	235	+25	+9.18	+4.27
I3	70–75	0	256	258	7	5	251	+2	+0.78	+0.52
I3	70–75	1	263	263	8	8	255	+0	-0.03	+0.48
J4	75–95	0	258	262	14	10	248	+4	+1.38	+0.44
J4	75–95	1	263	265	7	5	258	+2	+0.70	-0.38
I4	95–100	0	262	263	5	4	258	+1	+0.36	+0.47
I4	95–100	1	265	266	4	3	262	+1	+0.41	+0.51
J5	100–120	0	263	260	7	10	253	-3	-1.15	-1.64
J5	100–120	1	266	264	6	8	258	-2	-0.81	-1.04
I5	120–125	0	260	262	3	1	259	+2	+0.73	+0.49
I5	120–125	1	264	266	5	3	261	+2	+0.79	+0.64
J6	125–145	0	262	262	8	8	254	+0	+0.02	+1.11
J6	125–145	1	266	267	5	4	262	+1	+0.37	+0.37
I6	145–150	0	262	265	5	2	260	+3	+1.10	+0.01
I6	145–150	1	267	265	3	5	262	-2	-0.73	+0.13

G.4 Exposure-conditioned transfer and stage-level uncertainty

Table 42: **Complete exposure-conditioned transition counts.** Entries are acquired/eligible (%). Each row satisfies net = exposed acquisition + other acquisition − forgetting. Eligibility is defined at the beginning of that stage.

Model	Stage	Hint-exposed	Other unmastered	Lost	Net
3B	J1	3/3 (100.0)	81/264 (30.7)	2	+82
3B	I1	5/14 (35.7)	15/171 (8.8)	28	-8
3B	J2	13/25 (52.0)	70/168 (41.7)	17	+66
3B	I2	5/25 (20.0)	17/102 (16.7)	13	+9
3B	J3	24/26 (92.3)	51/92 (55.4)	16	+59
3B	I3	6/20 (30.0)	10/39 (25.6)	9	+7
3B	J4	11/17 (64.7)	12/35 (34.3)	9	+14
3B	I4	9/19 (47.4)	1/19 (5.3)	3	+7
3B	J5	7/14 (50.0)	4/17 (23.5)	12	-1
3B	I5	7/13 (53.8)	2/19 (10.5)	8	+1
3B	J6	11/14 (78.6)	2/17 (11.8)	9	+4
3B	I6	6/15 (40.0)	3/12 (25.0)	7	+2
7B	J1	20/23 (87.0)	98/163 (60.1)	11	+107
7B	I1	8/24 (33.3)	10/55 (18.2)	22	-4
7B	J2	21/26 (80.8)	36/57 (63.2)	12	+45
7B	I2	10/21 (47.6)	3/17 (17.6)	10	+3
7B	J3	15/15 (100.0)	10/20 (50.0)	8	+17
7B	I3	7/14 (50.0)	0/4 (0.0)	5	+2
7B	J4	10/12 (83.3)	4/4 (100.0)	10	+4
7B	I4	3/6 (50.0)	2/6 (33.3)	4	+1
7B	J5	4/6 (66.7)	3/5 (60.0)	10	-3
7B	I5	3/6 (50.0)	0/8 (0.0)	1	+2
7B	J6	6/7 (85.7)	2/5 (40.0)	8	+0
7B	I6	5/7 (71.4)	0/5 (0.0)	2	+3

Table 43: **Unguided stage summaries with uncertainty.** Rates (%) and 95% task-cluster intervals pool the eligible events. Acquisition is conditional on not being mastered; forgetting/retention are conditional on prior mastery. Unique counts distinct tasks ever acquired within that stage type. Pos. counts stages with positive net growth (out of six stages of each type).

Model	Stage	Acquisition [CI]	Forgetting [CI]	Retention [CI]	Unique	Pos.
3B	Joint	41.8 [38.2, 45.5]	6.8 [5.2, 8.5]	93.2 [91.5, 94.8]	229	5
3B	Internal.	18.4 [14.8, 22.0]	5.8 [4.5, 7.2]	94.2 [92.8, 95.5]	73	5
7B	Joint	66.8 [61.5, 72.2]	4.5 [3.4, 5.8]	95.5 [94.2, 96.6]	190	4
7B	Internal.	29.5 [22.5, 37.0]	3.0 [2.1, 3.9]	97.0 [96.1, 97.9]	47	5

G.5 Retention and newly hint-solvable tasks

Table 44: **Retention of capabilities acquired during joint stages 1–6.** Denominators count acquisition events, not unique tasks. Final retention permits intervening losses and reacquisitions; continuous retention requires membership at every checkpoint from acquisition through update 150.

Model	Retention target	Retained/acquired (%)	95% CI
3B	Next internalization	244/289 (84.4)	[80.7, 88.2]
3B	Final update 150	265/289 (91.7)	[87.7, 95.2]
3B	Every later checkpoint	191/289 (66.1)	[61.4, 70.9]
7B	Next internalization	203/229 (88.6)	[84.7, 92.5]
7B	Final update 150	218/229 (95.2)	[91.2, 98.3]
7B	Every later checkpoint	168/229 (73.4)	[68.1, 78.6]

New hinted mastery often coincides with unguided mastery at the same joint endpoint. The next internalization converts additional hint-only tasks into independent mastery.

Table 45: **Per-cycle retention and new hinted mastery.** New = unguided acquisitions; Next/Final/All = retained after the next internalization, at update 150, or continuously. New h = previously unmastered in both branches, then mastered with hints. Of these, Indep. are already unguided at the joint endpoint, Only h remain hint-only, and Transfer become unguided after the next internalization.

Model	Stage	New	Next	Final	All	New h	Indep.	Only h	Transfer
3B	J1	84	57	80	38	56	43	13	5
3B	J2	83	74	75	56	63	47	16	4
3B	J3	75	69	68	59	51	40	11	2
3B	J4	23	22	20	18	18	10	8	4
3B	J5	11	10	10	8	5	4	1	0
3B	J6	13	12	12	12	7	2	5	3
7B	J1	118	103	114	82	90	73	17	5
7B	J2	57	51	55	44	44	33	11	6
7B	J3	25	22	24	18	14	8	6	3
7B	J4	14	13	11	10	3	3	0	0
7B	J5	7	7	7	7	3	3	0	0
7B	J6	8	7	7	7	2	2	0	0

Table 46: **Fate of newly hint-solvable task–stage events.** Indep./Only h partition the new hinted set at the joint endpoint. Transfer refers to Only h. Kept/Lost refer to the already independent subset during the next internalization. Next and Final count all events independently mastered at those endpoints.

Model	New h	Indep.	Only h	Transfer (%)	Kept	Lost	Next	Final
3B	200	146	54	18/54 (33.3)	127	19	145	184
7B	156	122	34	14/34 (41.2)	111	11	125	146

G.6 THRESHOLD SENSITIVITY

Table 47: **Acquisition, forgetting, and exposure enrichment at each cutoff.** J pools six joint stages; I pools six internalization stages. Transfer entries are acquired/eligible (%); Δ is exposed minus other acquisition in percentage points.

Model	Cutoff	Stage	New	Lost	Net	Exposed	Other	Δ
3B	1/3	J	283	88	+195	128/201 (63.7)	155/341 (45.5)	+18.2
3B	2/3	J	289	71	+218	98/139 (70.5)	191/487 (39.2)	+31.3
3B	3/3	J	289	65	+224	69/99 (69.7)	220/593 (37.1)	+32.6
3B	1/3	I	81	70	+11	56/178 (31.5)	25/169 (14.8)	+16.7
3B	2/3	I	78	63	+15	45/150 (30.0)	33/258 (12.8)	+17.2
3B	3/3	I	86	68	+18	38/106 (35.8)	48/362 (13.3)	+22.6
7B	1/3	J	154	35	+119	87/106 (82.1)	67/102 (65.7)	+16.4
7B	2/3	J	186	51	+135	75/93 (80.6)	111/171 (64.9)	+15.7
7B	3/3	J	229	59	+170	76/89 (85.4)	153/254 (60.2)	+25.2
7B	1/3	I	34	38	-4	26/60 (43.3)	8/29 (27.6)	+15.7
7B	2/3	I	46	34	+12	34/68 (50.0)	12/61 (19.7)	+30.3
7B	3/3	I	51	44	+7	36/78 (46.2)	15/95 (15.8)	+30.4

The exposed acquisition rate exceeds the other-unmastered rate for both models, both stage types, and all three cutoffs. The magnitude varies, and the 7B internalization net is negative (-4 events) at the most permissive cutoff. Overall growth therefore does not imply positive change at every stage or every empirical threshold.

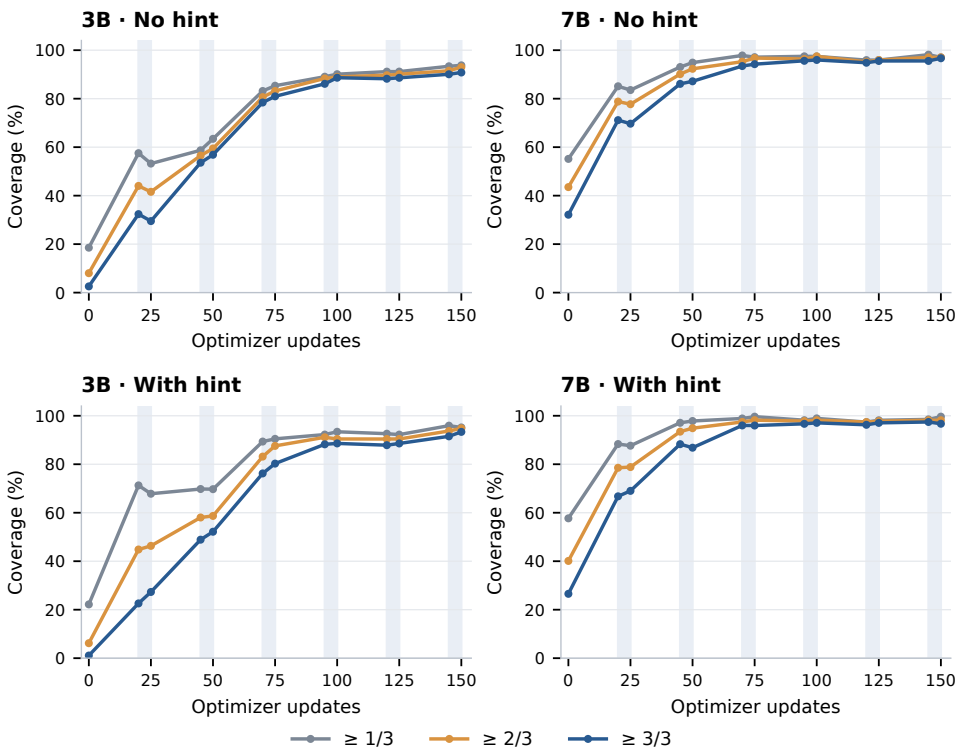


Figure 8: **Every checkpoint under all nontrivial three-execution cutoffs.** Coverage averages Seen and Unseen equally. The three curves correspond to at least one, two, or three successes. Alternative cutoffs are descriptive sensitivity checks, not confidence bounds on population task reliability.

G.7 Split-level and prior-success stratification

Table 48: **Stage summaries separately for Seen and Unseen.** The 3/3 rule and event-counting convention are unchanged. Joint stages 1–6 and all six internalization stages are pooled separately; transfer entries are acquired/eligible (%).

Model	Split	Stage	New	Lost	Net	Exposed	Other
3B	Seen	J	155	34	+121	35/51 (68.6)	120/284 (42.3)
3B	Unseen	J	134	31	+103	34/48 (70.8)	100/309 (32.4)
3B	Seen	I	41	33	+8	17/50 (34.0)	24/164 (14.6)
3B	Unseen	I	45	35	+10	21/56 (37.5)	24/198 (12.1)
7B	Seen	J	115	24	+91	38/45 (84.4)	77/126 (61.1)
7B	Unseen	J	114	35	+79	38/44 (86.4)	76/128 (59.4)
7B	Seen	I	22	19	+3	20/42 (47.6)	2/38 (5.3)
7B	Unseen	I	29	25	+4	16/36 (44.4)	13/57 (22.8)

Table 49: **Joint-stage acquisition stratified by prior unguided successes.** Conditioning on 0, 1, or 2 previous successes partially controls proximity to the 3/3 threshold. Entries are acquired/eligible (%) and task-cluster 95% intervals. Small strata and remaining difficulty differences preclude causal attribution.

Model	Prior	Exposed	95% CI	Other	95% CI
3B	0/3	45/65 (69.2)	[57.6, 80.3]	155/477 (32.5)	[28.5, 36.7]
3B	1/3	11/16 (68.8)	[43.8, 91.7]	31/68 (45.6)	[34.3, 57.0]
3B	2/3	13/18 (72.2)	[50.0, 92.9]	34/48 (70.8)	[58.1, 83.3]
7B	0/3	29/38 (76.3)	[61.8, 89.2]	87/170 (51.2)	[43.2, 59.8]
7B	1/3	15/16 (93.8)	[81.2, 100.0]	31/40 (77.5)	[63.6, 90.0]
7B	2/3	32/35 (91.4)	[81.5, 100.0]	35/44 (79.5)	[66.7, 91.5]

Table 50: **Both-branch aggregate stage statistics.** Branch 0/1 means no hint/with hint. Unique counts distinct acquired tasks; Pos. counts positive-net stages. A/F/R are acquisition/forgetting/retention rates (%). $\Sigma\Delta$Cov. sums domain-balanced coverage changes across the corresponding stage type; durations and starting capability differ.

Model	Stage	Branch	New	Lost	Net	Unique	Pos.	A	F	R	$\Sigma\Delta$Cov.
3B	J	0	289	65	+224	229	5	41.8	6.8	93.2	+81.65
3B	J	1	279	67	+212	225	5	38.9	7.2	92.8	+77.28
3B	I	0	86	68	+18	73	5	18.4	5.8	94.2	+6.59
3B	I	1	108	67	+41	93	6	21.4	5.9	94.1	+14.98
7B	J	0	229	59	+170	190	4	66.8	4.5	95.5	+61.98
7B	J	1	229	40	+189	185	5	65.6	3.1	96.9	+68.92
7B	I	0	51	44	+7	47	5	29.5	3.0	97.0	+2.56
7B	I	1	68	65	+3	62	3	42.5	4.4	95.6	+1.23

G.8 Complete uncertainty summaries

Intervals below are recomputed using the same task-cluster bootstrap as the main table. They condition on this evaluation export and preserve dependencies across checkpoints. They can differ slightly from intervals supplied in the original export because the bootstrap implementation is explicitly specified here.

Table 51: **All adjacent-stage changes with 95% intervals.** Each cell is change [lower, upper] in percentage points. Cov. uses 3/3 membership; SR uses the mean of the three executions. Both average Seen and Unseen equally.

Model	Stage	ΔCov.$_0$	ΔSR$_0$	ΔCov.$_h$	ΔSR$_h$
3B	J1	+29.81 [24.3, 35.3]	+34.93 [30.0, 39.9]	+21.50 [16.7, 26.6]	+36.41 [32.1, 40.7]
3B	I1	-2.87 [-7.9, 2.2]	-3.20 [-7.6, 1.0]	+4.72 [-0.4, 9.8]	+0.93 [-3.5, 5.3]
3B	J2	+24.08 [17.6, 30.6]	+14.84 [8.7, 20.9]	+21.57 [14.9, 28.2]	+11.74 [5.9, 17.6]
3B	I2	+3.28 [-0.8, 7.6]	+3.64 [-0.1, 7.5]	+3.28 [-1.2, 7.7]	+1.29 [-2.4, 4.9]
3B	J3	+21.55 [15.4, 28.0]	+20.83 [14.9, 26.9]	+24.05 [17.9, 30.3]	+22.72 [17.6, 28.0]
3B	I3	+2.55 [-1.1, 6.2]	+2.42 [-0.3, 5.2]	+4.06 [-0.3, 8.4]	+3.20 [0.0, 6.3]
3B	J4	+5.16 [1.2, 9.2]	+4.67 [1.2, 8.3]	+7.95 [3.9, 12.0]	+4.43 [1.5, 7.4]
3B	I4	+2.52 [-0.0, 5.1]	+1.41 [-0.8, 3.7]	+0.37 [-2.9, 3.7]	+0.28 [-2.1, 2.6]
3B	J5	-0.42 [-3.8, 2.9]	+0.44 [-2.8, 3.6]	-0.73 [-4.8, 3.3]	-0.51 [-3.7, 2.7]
3B	I5	+0.39 [-2.5, 3.3]	+0.26 [-1.9, 2.5]	+0.71 [-1.8, 3.3]	+0.12 [-1.6, 2.0]
3B	J6	+1.46 [-1.8, 4.8]	+1.70 [-1.1, 4.5]	+2.94 [-0.7, 6.9]	+3.30 [0.3, 6.4]
3B	I6	+0.73 [-2.2, 3.6]	+0.87 [-1.0, 2.7]	+1.83 [-0.7, 4.4]	+0.74 [-0.7, 2.3]
7B	J1	+39.01 [32.4, 45.9]	+34.75 [29.0, 40.4]	+40.21 [34.0, 46.5]	+36.43 [31.0, 41.7]
7B	I1	-1.46 [-6.2, 3.0]	-1.37 [-5.0, 2.2]	+2.25 [-3.2, 7.7]	+0.62 [-3.3, 4.5]
7B	J2	+16.42 [10.9, 22.2]	+12.78 [8.3, 17.4]	+19.26 [13.3, 25.4]	+14.44 [10.1, 18.9]
7B	I2	+1.06 [-2.3, 4.6]	+1.69 [-1.2, 4.6]	-1.46 [-5.8, 2.9]	+0.24 [-2.0, 2.6]
7B	J3	+6.30 [2.3, 10.3]	+4.06 [1.2, 7.0]	+9.18 [5.5, 13.2]	+4.27 [2.1, 6.5]
7B	I3	+0.78 [-1.8, 3.3]	+0.52 [-1.3, 2.5]	-0.03 [-2.9, 2.9]	+0.48 [-1.1, 2.2]
7B	J4	+1.38 [-1.9, 4.7]	+0.44 [-2.1, 3.1]	+0.70 [-1.8, 3.2]	-0.38 [-2.3, 1.6]
7B	I4	+0.36 [-1.8, 2.5]	+0.47 [-1.2, 2.2]	+0.41 [-1.4, 2.3]	+0.51 [-0.5, 1.5]
7B	J5	-1.15 [-4.1, 1.8]	-1.64 [-4.3, 0.9]	-0.81 [-3.4, 1.8]	-1.04 [-3.1, 0.9]
7B	I5	+0.73 [-0.7, 2.2]	+0.49 [-0.4, 1.6]	+0.79 [-1.1, 2.7]	+0.64 [-0.5, 2.0]
7B	J6	+0.02 [-2.9, 3.0]	+1.11 [-1.2, 3.6]	+0.37 [-1.8, 2.5]	+0.37 [-1.1, 2.0]
7B	I6	+1.10 [-0.7, 3.0]	+0.01 [-1.1, 1.0]	-0.73 [-2.9, 1.4]	+0.13 [-1.0, 1.4]

Table 52: **Aggregate rates and coverage changes for both branches.** The first column gives model, stage type, and branch (0 = no hint; 1 = with hint). Rates and 95% intervals are in %; summed coverage changes are in percentage points.

Model / branch	Acquisition [CI]	Forgetting [CI]	Retention [CI]	$\Sigma\Delta$Cov. [CI]
3B J / 0	41.8 [38.2, 45.5]	6.8 [5.2, 8.5]	93.2 [91.5, 94.8]	+81.65 [72.6, 90.7]
3B J / 1	38.9 [35.8, 42.2]	7.2 [5.7, 8.9]	92.8 [91.1, 94.3]	+77.28 [68.2, 86.4]
3B I / 0	18.4 [14.8, 22.0]	5.8 [4.5, 7.2]	94.2 [92.8, 95.5]	+6.59 [-2.5, 15.5]
3B I / 1	21.4 [17.8, 24.9]	5.9 [4.5, 7.4]	94.1 [92.6, 95.5]	+14.98 [5.8, 24.1]
7B J / 0	66.8 [61.5, 72.2]	4.5 [3.4, 5.8]	95.5 [94.2, 96.6]	+61.98 [53.1, 71.0]
7B J / 1	65.6 [60.3, 71.0]	3.1 [2.1, 4.2]	96.9 [95.8, 97.9]	+68.92 [60.0, 78.2]
7B I / 0	29.5 [22.5, 37.0]	3.0 [2.1, 3.9]	97.0 [96.1, 97.9]	+2.56 [-4.4, 9.2]
7B I / 1	42.5 [35.7, 49.6]	4.4 [3.4, 5.5]	95.6 [94.5, 96.6]	+1.23 [-6.8, 9.0]

Table 53: **Exposed and other candidate acquisition rates with 95% intervals.** Rates are percentages; their difference is in percentage points. Eligibility sets and pooled counts match Table 37.

Model	Stage	Exposed [CI]	Other [CI]	Difference [CI]
3B	Joint	69.7 [60.5, 78.5]	37.1 [33.4, 41.0]	+32.6 [22.8, 41.9]
3B	Internal.	35.8 [27.1, 44.5]	13.3 [9.8, 16.8]	+22.6 [13.2, 31.8]
7B	Joint	85.4 [78.2, 92.2]	60.2 [53.7, 67.0]	+25.2 [15.6, 34.2]
7B	Internal.	46.2 [35.0, 57.5]	15.8 [8.9, 23.8]	+30.4 [17.3, 43.2]

Table 54: **Internalization of newly hint-only events.** These tasks were unmastered under both conditions before a joint stage, attained guided mastery only at its endpoint, and are evaluated for unguided mastery at the next internalization endpoint.

Model	Transferred/eligible (%)	95% CI
3B	18/54 (33.3)	[21.1, 45.7]
7B	14/34 (41.2)	[25.8, 57.1]

G.9 Task-level outcome matrices

Figures 9–10 encode the complete per-task outcomes used in this analysis. Task IDs follow the supplied export: 1–140 are Seen and 141–274 are Unseen. Each cell gives a success count from 0 to 3, with one matrix per model and hint condition; all checkpoints remain in chronological order. The accompanying source export preserves the exact numeric entries, including all 16 supplied tables.

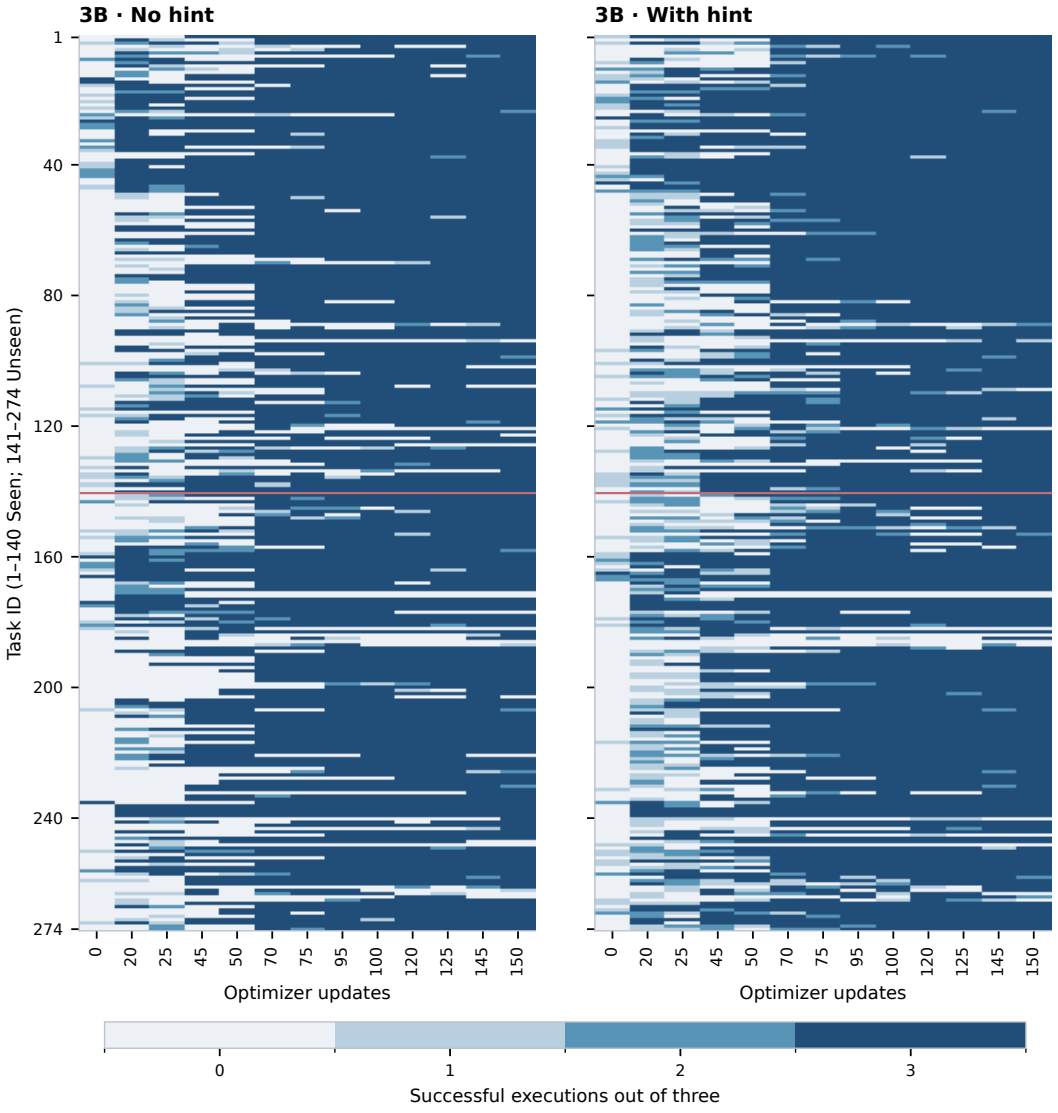


Figure 9: **3B: complete task-level success counts.** Rows preserve task identity across all 13 checkpoints. The horizontal line separates Seen from Unseen. Darkest cells meet the primary 3/3 capability criterion; lighter cells expose instability hidden by aggregate success.

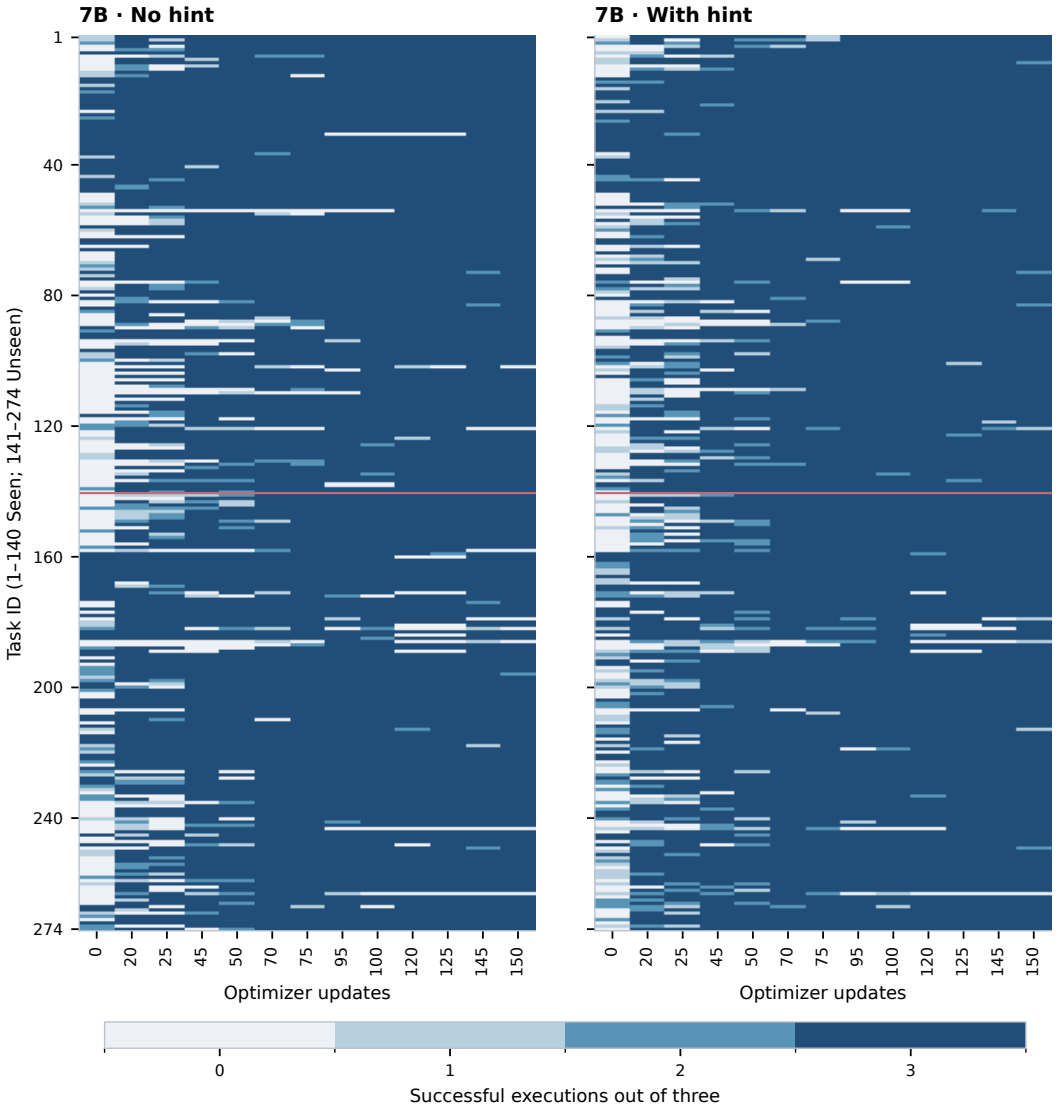


Figure 10: **7B: complete task-level success counts.** Rows preserve task identity across all 13 checkpoints. The horizontal line separates Seen from Unseen. Darkest cells meet the primary 3/3 capability criterion; lighter cells expose instability hidden by aggregate success.

H Discussion of Dynamic Capability Frontiers and Self-Evolution

HELICES connects experience generation, guidance utilization, verified internalization, and the renewal of learning opportunities within a common task-level formulation and training framework. This section explains how these contributions fit together, distinguishes the framework from related approaches, and relates its claims to the theoretical and empirical analyses reported in the paper.

H.1 Learning opportunities evolve with the policy

The learning value of experience depends on the agent's current capabilities. Guidance for a behavior already performed reliably without assistance may offer less additional success-related learning value, whereas guidance that the solver previously could not use may become useful after its capabilities improve. Sustained self-evolution therefore requires experience generation and internalization to adapt to one another. Current policy trajectories and environmental feedback inform hint generation; hinted execution exposes effective behaviors; and internalizing those behaviors changes the policy that produces and uses subsequent experience. Internalization can thus affect both independent task solving and the generation and utilization of future guidance.

Section 3 expresses this relationship through the unguided reliable-task region $\mathcal{C}_k^0(\alpha)$, the region reliably solvable under the current hint protocol $\mathcal{C}_k^h(\alpha)$, and the hint-exposed region

$$\mathcal{B}_k(\alpha) = \mathcal{C}_k^h(\alpha) \setminus \mathcal{C}_k^0(\alpha).$$

Hint reachability depends jointly on the current solver, its source trajectories, and the hint generator. Updating the model can change which guidance it produces, which guidance it can use, and which behaviors that guidance supports. The dynamic frontier therefore describes a concrete learning relationship: changes in experience generation and task solving reshape the behaviors available for further internalization.

This formulation makes the question of *what can be learned next* explicitly dependent on the policy and its interaction experience. The regions are relative to a specified task distribution, reliability threshold, and generation and execution protocol. They need not be nested, and the hint-reachable region is not an intrinsic upper bound on the solver. These qualifications are important when interpreting newly acquired tasks or comparing frontiers across checkpoints.

H.2 Hints as adaptive learning probes

HELICES uses hints to probe for behaviors that the current policy can produce with guidance but has not yet mastered independently. A trajectory-level improvement supplies a candidate learning signal, and reproduction under independent execution supplies evidence for selecting a more reliable target. Joint optimization of hint generation and utilization couples these signals to the evolving shared model: the generator learns from execution feedback to propose usable guidance, while the solver learns to execute successfully under that guidance. New policy trajectories then reveal updated difficulties and supply new evidence for generating hints.

The capability regions organize this process and its evaluation; the implemented algorithm does not estimate complete region membership or use it as an explicit task-sampling curriculum. It uses trajectory-level probing and confirmation as practical signals. In particular, the common source-success baseline cancels under within-source reward centering (Section 4.2). Adaptation comes from refreshed experience, updated generation and utilization, and subsequent verification, rather than from an explicit preference over tasks induced by that baseline. Fresh hint-free evaluation measures whether the resulting updates improve independent capabilities.

H.3 Reliable targets and selective behavioral internalization

HELICES treats target reliability and supervision granularity as connected design questions. Independent confirmation requires the same hint to produce the same class of improvement against a fresh realization's own unguided baseline. This reduces the risk of admitting gains specific to the discovery execution. The confirmation trajectory, rather than the original discovery trajectory,

supplies the accepted behavioral target. Confirmation is a selection rule supported by fresh-seed auditing (Table 5); it does not certify that a hint will help on every future execution.

For accepted trajectories, explicit hints are removed from the training inputs. Action tokens receive full supervision, while reasoning tokens are selectively weighted using the likelihood contrast between hinted and unhinted contexts for the same target and history. The two mechanisms act at different levels: confirmation selects which reproduced improvements enter the learner, and attribution controls how reasoning within those targets contributes to learning. Full action supervision preserves the successful behavioral sequence. Together, these mechanisms specify a path from reproducible execution gains to independent policy learning.

The likelihood contrast is a proxy for hint support, not a certificate of token correctness or causal necessity. Its learning value is tested by the attribution-specificity controls in Appendix E.1, which hold auxiliary targets and scoring fixed while comparing attribution weights with shuffled weights, equal-total-weight uniform supervision, ungated reasoning, and action-only supervision.

H.4 Relation to co-evolution and self-distillation methods

Adaptive experience and joint improvement of generation and utilization also appear in related work. Evolving-RL (Fan et al., 2026) jointly optimizes experience extraction and utilization and reports benefits without test-time experience accumulation. SAPO (Zhang et al., 2026c) uses reward differences between matched rollout groups to assess a candidate skill's marginal utility and train skill generation. These connections motivate locating the HELICES contribution in its specific formulation, verification and internalization mechanisms, and controlled study of their feedback, rather than in co-evolution alone.

SEED (Wu et al., 2026b) is particularly close in its use of a policy-synchronized analyzer and skill-conditioned probability shifts for on-policy self-distillation. Its offline stage uses GLM-5.2 to annotate completed trajectories with hindsight skills, followed by supervised fine-tuning to initialize trajectory analysis. During reinforcement learning, the analyzer is synchronized with the current policy. HELICES requires no external LLM for skill annotation, analyzer initialization, or teacher supervision during post-training. Given a pretrained backbone and task environment, it learns hint generation and utilization from its own trajectories, generated hints, and environmental feedback.

This supervision setting makes the reliability of self-generated guidance part of the learning problem. HELICES optimizes the generator through execution feedback, independently confirms behavioral gains from the same hint, and internalizes accepted confirmation trajectories with full action and selective reasoning supervision. The comparison concerns both the source of supervision and the way it is validated and used. It does not imply that evolving guidance or likelihood contrasts are unique to HELICES. The separate SEED comparison in Appendix D.4 preserves the distinction between the methods' initialization and evaluation protocols.

H.5 Capability acquisition, retention, and theoretical scope

Average success, guided performance, and reliable-task coverage describe different aspects of improvement. A guided–unguided performance gap alone cannot measure internalization progress when both the generator and solver evolve. Independent capabilities may improve while updated hints expose additional opportunities, leaving the gap unchanged or larger. Moreover, the difference between guided and unguided coverage is not generally the mass of the hint-exposed region, because the two regions need not be nested.

The task-level decomposition in Equation 7 therefore separates acquisition within the previously exposed region, acquisition outside that region, and forgetting of previously reliable tasks. Acquisition beyond prior exposure means acquisition beyond the specified earlier probing protocol, not beyond an absolute capability limit. Tracking these components and retention reveals changes hidden by aggregate success rates. It also distinguishes cumulative growth from monotonic improvement at every stage. The empirical 3/3 criterion in Appendix G is a finite-sample diagnostic; it does not establish population-level reliability for each task.

The successful-target projection bound in Equation 8 relates fitting successful behavior under unguided conditioning to independent success probability. Combined with control of policy drift on

the previously reliable region, the acquisition–retention analysis gives conditional sufficient conditions for net expansion (Appendix B). This links the available learning opportunities, the adequacy of internalization, and the stability of prior capabilities.

The theory and implementation have distinct roles. The convergence analysis uses fixed targets and explicit optimization assumptions, including the Polyak–Łojasiewicz condition where required; selective supervision enters through an update-discrepancy term. Actual training refreshes targets and combines selective supervision with GRPO and AdamW. The reported bounds are not numerically calibrated guarantees for those runs. The theory identifies relationships sufficient for expansion, the algorithm constructs and learns verified targets, and unguided evaluation tests acquisition and retention.

H.6 How the mechanism studies support the framework

The mechanism studies examine complementary links in the feedback process. The fixed-source, cached-hint Generator–Solver crossing in Appendix F.1 tests how pre- and post-internalization parameter states affect guidance generation and utilization. The Generator-state by source-state comparison in Appendix F.2 further separates parameter changes from changes in the experience supplied to the generator. Together, these evaluations examine how internalization can influence subsequent experience use and generation, beyond its effect on unguided performance.

The complete-pipeline controls in Appendix F.3 compare repeated cycles with a frozen generator, static probes, separate Generator/Solver parameters, and a single-switch schedule. They test the value of refreshing experience and repeating extraction and internalization. Evidence for parameter sharing is more limited: its advantage over separate models is not uniform across seeds, and the separate models have a different compute footprint. The attribution-specificity study separately tests whether reasoning-token selection contributes beyond sparsification or uniform downweighting.

Finally, task-level tracking finds higher subsequent acquisition rates among previously hint-exposed tasks than among other unmastered tasks (Table 42), supporting exposure as a useful signal of learning opportunity. This association does not itself establish a causal effect of exposure; the controlled comparisons provide complementary evidence. Collectively, these studies connect the dynamic frontier formulation to testable processes of experience generation, verified internalization, capability acquisition, and retention. HELICES thus contributes a coordinated account of how current experience supports learning and how learning changes the experience available next.