CLEAR THE BOTTLENECK: LEARNING WHEN LANGUAGE AGENTS ARE READY TO ACT

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon language agents often fail by acting before task-enabling prerequisites have been resolved: answering before evidence is sufficient, executing before the relevant state is known, or committing before required verification is complete. We study these failures through *readiness bottlenecks*, route-dependent prerequisites whose completion can substantially change the value of the remaining trajectory. The most faithful signal requires actually executing each candidate completion and rolling out the resulting context, which is too expensive for dense training. We therefore introduce a *readiness certificate*: a compact, grounded semantic representation of a prerequisite's completed state. The certificate acts as a surrogate for the full post-completion context, and Readiness-to-Action (RTA) measures the teacher's induced high-level action response to this surrogate. Its validity is tested against real execution rather than assumed: certificate-induced action shifts align with real-completion shifts (cosine 0.78), and RTA sensitivity correlates with downstream Prerequisite Completion Gain (Spearman $\rho = 0.66$ across 2,643 audited state–condition pairs). Across the six prerequisite families, real completion yields a +14.8-point macro-average gain in normalized continuation return. We aggregate calibrated RTA signals into a readiness-aware action advantage and distill the resulting KL-regularized policy update alongside outcome RL. The resulting *Bottleneck Distillation* improves average task success from 42.6% to 46.1% over the strongest non-ours baseline while reducing premature commitment from 9.8% to 5.9% across procedural, stateful, evidence-intensive, compliance, and scientific agent tasks. See our anonymous link for implementation.

1 INTRODUCTION

Many agent failures are failures of readiness. A research agent can produce a fluent answer before retrieving the evidence that supports it. A customer-service agent can issue a refund before verifying eligibility. A stateful tool agent can modify an object before observing the state that determines which operation is appropriate. A scientific agent can launch an analysis before checking that the required variables, controls, or assumptions are in place. In each case, the agent may possess the capability needed to finish the task; what fails is the transition from preparation to execution.

This failure mode spans modern agent benchmarks. τ-bench and τ^2 combine user goals, policies, and state-changing APIs (Yao et al., 2025; Barres et al., 2025); MAC-Bench introduces adversarial compliance pressure (Zhao et al., 2026d); ToolSandbox and AppWorld-UL stress state acquisition, clarification, and confirmation (Lu et al., 2025; Chen et al., 2026); τ-Knowledge and GAIA require evidence before reliable answers (Shi et al., 2026; Mialon et al., 2024; Qi et al., 2026). Scientific-discovery agents expose the same structure at longer horizons: ScienceAgentBench evaluates data-driven scientific workflows (Chen et al., 2025), SciAgentGym stresses multi-step scientific tool orchestration (Shen et al., 2026b), and DiscoveryBench formalizes data-driven hypothesis discovery (Majumder et al., 2025). Terminal reward tells us whether the trajectory worked; it does not identify which unresolved prerequisite currently blocks a productive route.

We call such a prerequisite a **readiness bottleneck**. The concept is route-relative. A task can admit several successful pathways while each individual pathway contains conditions that are non-redundant within that route. This structure is naturally captured by Mackie's INUS logic and the sufficient-component view (Mackie, 1965; Rothman, 1976): a condition can be insufficient on its

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.

own and unnecessary globally, yet still be indispensable inside one viable sufficient configuration. This makes readiness a more precise object than generic "progress": the relevant question is which unresolved condition changes what the agent can profitably do next and what the remaining trajectory can ultimately achieve.

The most faithful way to answer that question is to *complete the prerequisite*. From a checkpointed state, execute a verified completion operator, retain the observations and history produced by that execution, and compare the continuation against an equal-budget control. We formalize the resulting long-horizon difference as **Bottleneck Completion Advantage** (BCA), defined directly in terms of the benchmark-normalized terminal outcome, and estimate it empirically through **Prerequisite Completion Gain** (PCG). Real completion therefore defines the semantic and outcome consequences of clearing a bottleneck.

Real completion is too expensive to evaluate for every candidate in every on-policy state. Our key idea is to compress the post-completion state into a **readiness certificate**. A certificate records the grounded prerequisite semantics, its entity bindings, and whether the corresponding gate is resolved. It is a decision-relevant semantic summary: rather than replaying the trajectory that established the prerequisite, it represents the completed condition in compact form. Its usefulness is an empirical question, so we directly test whether it preserves the action-policy consequence of real completion.

Concretely, a synchronized privileged teacher is evaluated twice on the same interaction history: once with the factual open certificate and once with the matched completed certificate. Their log-policy contrast defines **Readiness-to-Action** (RTA). Independently, the environment is forked and the prerequisite is actually completed, producing a new observation/history context. We compare the certificate-induced action shift with this real-completion-induced shift, and separately compare RTA magnitude with PCG. The first test measures *semantic fidelity*: does the compact certificate preserve the direction in which real completion changes the decision? The second measures *value fidelity*: does certificate sensitivity identify conditions whose actual completion improves downstream return? Across 2,643 audited state–condition pairs, the two shifts have cosine alignment 0.78 and RTA sensitivity correlates with PCG at $\rho = 0.66$; the six-family macro-average PCG is +14.8 points.

This surrogate view yields a simple learning pipeline. Real completion provides the audit target; the readiness certificate provides the cheap training interface. We aggregate calibrated RTA contrasts into a readiness-aware action advantage and perform a KL-regularized local policy update, implemented by action-level distillation alongside outcome RL. Candidate templates are proposed offline from domain artifacts and grounded online with executable checkers, so the method ranks behaviorally consequential prerequisites within a structured candidate space. The trained student requires neither certificates nor privileged state at inference.

The experiments follow this same chain. We first establish that real completion changes downstream return and exhibits route-relative bottleneck signatures. We then test whether readiness certificates preserve real completion's action effect and whether their sensitivity predicts completion value, including within-family calibration and negative candidates. Finally, we evaluate whether the resulting training signal improves success under matched privilege and compute, transfers across OOD readiness structure and thinking models, and remains effective with automatically constructed candidates.

Our contributions are:

- We formulate **readiness bottlenecks** as route-relative prerequisites with a measurable long-horizon completion value and derive controlled signatures for route readiness and pathway substitution.
- We introduce **readiness certificates** as compact semantic surrogates for post-completion contexts, and validate them through both certificate-to-real policy-shift fidelity and RTA-to-PCG value calibration.
- We turn the calibrated certificate response into **Bottleneck Distillation**, a readiness-guided policy update that improves success and commitment timing across nine agent benchmarks without privileged readiness information at inference.

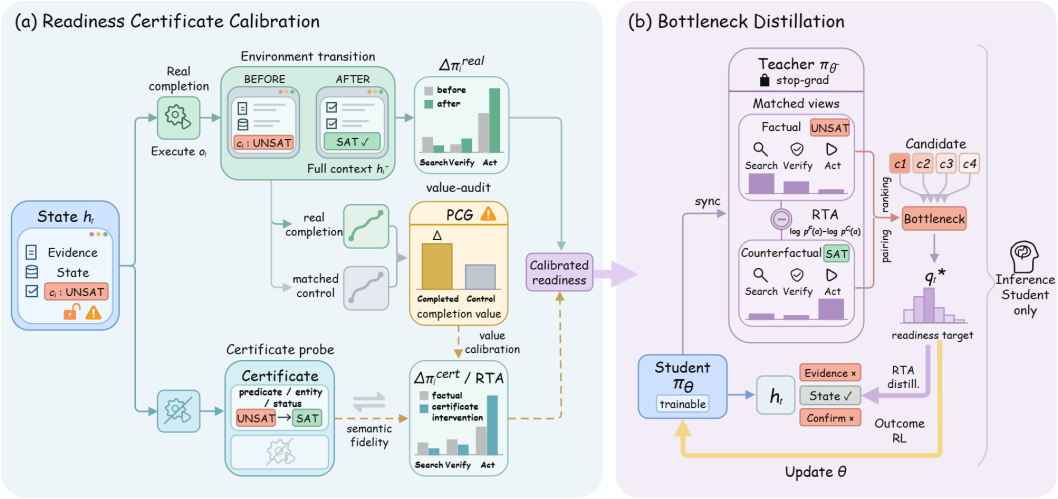

Figure 1: **Overview of Readiness Certificate Calibration and Bottleneck Distillation. (a) Readiness certificate calibration.** Starting from an unresolved prerequisite c_i, real completion executes the verified operator o_i and exposes the full post-completion context, defining the real policy shift $\Delta\pi_i^{\text{real}}$ and the downstream completion value measured by PCG against a matched control. In parallel, a compact readiness certificate counterfactually changes only c_i from UNSAT to SAT, inducing the certificate shift $\Delta\pi_i^{\text{cert}}$ and Readiness-to-Action (RTA). We validate the surrogate through *semantic fidelity*, aligning $\Delta\pi_i^{\text{cert}}$ with $\Delta\pi_i^{\text{real}}$, and *value calibration*, relating RTA sensitivity to PCG. **(b) Bottleneck Distillation.** A synchronized stop-gradient teacher evaluates matched factual (UNSAT) and counterfactual (SAT) certificate views of student-generated states. Their RTA contrasts identify and rank unresolved bottlenecks, which are aggregated into a readiness-aware target policy q_t^* and distilled into the trainable student alongside outcome RL. Certificates and privileged teacher information are used only during training; inference uses the student alone.

2 Readiness Bottlenecks and Completion Value

Route-relative task structure. Let $x_i(s) \in \{0,1\}$ indicate whether task-enabling condition c_i is satisfied at state s, and let $U_k \in \{0,1\}$ denote the route-external success event collecting the remaining reasoning, execution, and stochastic factors for route k. An idealized multi-path task can be written as

$$Y = \bigvee_{k=1}^{K} \left(U_k \wedge \bigwedge_{i \in \mathcal{P}_k} x_i(s) \right), \tag{1}$$

where $\mathcal{P}_k$ is route k's prerequisite set. A readiness bottleneck is an unresolved condition that belongs to at least one currently viable route. Equation 1 motivates two testable signatures under a fixed-route / viable-route interpretation. First, within a fixed viable route, an unresolved non-redundant prerequisite constrains the route's attainable continuation success. Second, when newly introduced bypass routes are behaviorally reachable by the continuation policy and do not alter the focal completion operator, the marginal value of the focal prerequisite is expected to decrease as those alternative routes become available. We test both signatures empirically rather than treating them as unconditional algebraic consequences of Eq. 1.

Real completion defines bottleneck value. Let $R \in [0,1]$ denote the benchmark-normalized terminal outcome; for binary tasks R is the success indicator, while graded benchmarks retain their normalized score. For a verified temporally extended completion operator o_i and an equal-budget control o_i^0 that leaves c_i unresolved (Sutton et al., 1999), define the terminal-outcome option value

$$Q_R^\pi(h_t, o) := \mathbb{E}_\pi[R \mid h_t, o], \qquad \text{BCA}_{t,i}^\pi := Q_R^\pi(h_t, o_i) - Q_R^\pi(h_t, o_i^0). \tag{2}$$

BCA is therefore relative to the current history, continuation policy, and specified completion/control pair. Because Q_R^π evaluates the terminal outcome after the temporally extended operator and

subsequent continuation, it includes all downstream state, observation, history, and action changes induced by completion.

With M independent continuations from each matched branch, we estimate Eq. 2 by

$$\widehat{\mathrm{PCG}}_{t,i} = \frac{1}{M}\sum_{m=1}^{M} R_{t,i}^{+,m} - \frac{1}{M}\sum_{m=1}^{M} R_{t,i}^{0,m}, \tag{3}$$

where $R_{t,i}^{+,m}$ and $R_{t,i}^{0,m}$ are terminal outcomes from the real-completion and matched-control branches, respectively. Thus $\widehat{\mathrm{PCG}}_{t,i}$ is the direct Monte-Carlo estimator of $\mathrm{BCA}_{t,i}^{\pi}$; our audit uses $M = 8$ continuations per branch (Appendix C).

Readiness certificates compress completion semantics. For dense learning, each grounded candidate is represented by a readiness certificate

$$\xi_{t,i} = (c_i, b_{t,i}, z_{t,i}), \qquad z_{t,i} \in \{\textsc{inactive}, \textsc{sat}, \textsc{unsat}\}, \tag{4}$$

where c_i specifies the prerequisite, $b_{t,i}$ binds its task entities, and $z_{t,i}$ records its grounded status. The certificate is a semantic abstraction of the full completion context: its purpose is to preserve the information needed to determine how the high-level action policy should change when the bottleneck is cleared. The real-completion audit directly tests this surrogate relation.

3 Related Work

Sequential credit and readiness. Temporal abstraction and hierarchical credit provide the closest decision-theoretic context (Sutton et al., 1999; Peng et al., 2026). AgentPro and ToolPRMBench provide process supervision (Deng et al., 2025; Li et al., 2026a); BEACON assigns milestone-level temporal credit (Wang et al., 2026); Verified Critical Step Optimization tests action alternatives against final outcomes (Li et al., 2026b). Asking Before Acting, environment probing, Look Before You Leap, and WISE-Flow emphasize information gathering or prerequisites before execution (Chen et al., 2023; Song and Cai, 2026; Ye et al., 2026b; Zhou et al., 2026). Our unit is an unresolved state predicate whose completion has a measurable continuation value.

Privileged on-policy supervision and semantic surrogates. OPSD and OPCD distill context-privileged teachers on student-induced states (Zhao et al., 2026a; Ye et al., 2026a); CREDIT, OPID, EDGE-OPD, and UCOB refine privileged supervision through input-, skill-, evidence-, or credit-aware signals (Shen et al., 2026a; Yang et al., 2026; Lazaridis et al., 2026; Tu et al., 2026; Zhao et al., 2026b;c). Our privileged object is a compact readiness certificate intended to stand in for the decision-relevant consequence of a richer post-completion context. Its role is established by fidelity to real completion: policy-shift alignment tests semantic preservation, while PCG calibration tests downstream value preservation. This also connects to analyses showing that privileged likelihood need not automatically equal outcome value (Nguyen et al., 2026) and motivates action-level rather than full-trace supervision for thinking models (Kaur et al., 2026).

Scientific-discovery agents. ScienceAgentBench, DiscoveryBench, and SciAgentGym expose long workflows with explicit dependencies among data inspection, tool execution, validation, and interpretation (Chen et al., 2025; Majumder et al., 2025; Shen et al., 2026b), providing a complementary readiness test beyond procedural and search agents.

4 Method: Bottleneck Distillation with Readiness Certificates

4.1 Constructing grounded readiness certificates

A candidate condition $c_i \in \mathcal{C}$ is a verifiable predicate over task state. Candidate templates are proposed once per domain from artifacts available when engineering the environment:

$$\mathcal{C} = \Phi(\text{policies, tool schemas, task/state specifications, workflow docs}). \tag{5}$$

At an on-policy decision state h_t, executable checkers ground this fixed candidate set into a certificate table

$$g_t = G(\mathcal{C}, h_t) = \{\xi_{t,i}\}_{i \in \mathcal{C}}, \tag{6}$$

with $\xi_{t,i}$ defined in Eq. 4. Evidence-heavy tasks use support maps or artifact validators when the environment does not expose the required predicate directly.

For each available high-level action $a \in \mathcal{A}_t$ with canonical prefix $y(a)$, we compute a length-normalized prefix score and normalize over the finite canonical action set:

$$\ell_\theta(a \mid h_t) = \frac{1}{|y(a)|} \sum_j \log p_\theta(y_j \mid h_t, y_{<j}),$$
$$p_\theta^A(a \mid h_t) = \frac{\exp \ell_\theta(a \mid h_t)}{\sum_{a' \in \mathcal{A}_t} \exp \ell_\theta(a' \mid h_t)}. \tag{7}$$

We refer to p_θ^A as the *induced canonical-action distribution*: because ℓ_θ is a length-normalized sequence score, it is a constructed high-level distribution rather than the exact marginal probability of a latent action under the autoregressive model. Tool arguments and free-form reasoning remain governed by the base outcome objective.

4.2 RTA PROBES THE SEMANTIC SURROGATE OF COMPLETION

Let $\bar{\theta} = \mathrm{sg}(\theta)$ be a synchronized stop-gradient copy of the current policy. The factual privileged teacher receives the grounded certificate table:

$$p_t^F(a) = \pi_{\bar{\theta}}^A(a \mid h_t, g_t). \tag{8}$$

For every open candidate i, the matched completed certificate $\mathrm{cf}_i(g_t)$ changes only the target certificate status from UNSAT to SAT, while preserving its predicate semantics, entity bindings, all other certificates, interaction history, and available actions. We score

$$p_t^{C_i}(a) = \pi_{\bar{\theta}}^A(a \mid h_t, \mathrm{cf}_i(g_t)), \qquad \mathrm{RTA}_{t,i}(a) = \log p_t^F(a) - \log p_t^{C_i}(a). \tag{9}$$

With this factual-minus-completed convention, positive RTA denotes actions relatively favored while the prerequisite remains unresolved; consequently, the auxiliary update promotes preparation actions and suppresses actions whose preference emerges only after completion. We center the contrast under the student induced action distribution,

$$\bar{r}_{t,i}(a) = \mathrm{RTA}_{t,i}(a) - \mathbb{E}_{a' \sim p_\theta^A} \mathrm{RTA}_{t,i}(a'), \tag{10}$$

and summarize its magnitude by $m_{t,i} = D_{\mathrm{JS}}(p_t^F \| p_t^{C_i})$, using natural logarithms. The top-K open candidates form the selected set $\mathcal{B}_t$.

The certificate representation is validated against execution. Let $h_{t,i}^+$ be the full history after executing o_i and $h_{t,i}^0$ the equal-budget matched-control history. In the same canonical action space, define

$$\Delta\pi_{t,i}^{\mathrm{cert}} = p_t^{C_i} - p_t^F,$$
$$\Delta\pi_{t,i}^{\mathrm{real}} = \pi_{\bar{\theta}}^A(\cdot \mid h_{t,i}^+) - \pi_{\bar{\theta}}^A(\cdot \mid h_{t,i}^0). \tag{11}$$

Their cosine similarity measures surrogate fidelity; pairs for which either shift is the zero vector are excluded from the cosine statistic. Separately, the correlation between $m_{t,i}$ and $\widehat{\mathrm{PCG}}_{t,i}$ measures value fidelity.

Value interpretation. Under the local Boltzmann relation $p_T(a \mid h, z) \propto \rho(a \mid h) \exp(Q_T(h, z, a)/\alpha)$ with the same action prior $\rho(a \mid h)$ under the matched open/completed certificate intervention, centering Eq. 9 removes the action-independent partition shift. This gives the soft-value interpretation in Appendix A; the real-completion audit in Eqs. 11 and 3 provides the empirical grounding.

4.3 FROM CERTIFICATE RESPONSE TO POLICY IMPROVEMENT

For nonempty $\mathcal{B}_t$, define

$$S_t := \sum_{j \in \mathcal{B}_t} m_{t,j}, \qquad \omega_{t,i} := \frac{m_{t,i}}{S_t}, \qquad \kappa_t := 1 - \exp\left(-\frac{S_t}{\beta}\right), \tag{12}$$

and the readiness advantage proxy

$$\widehat{A}_R(h_t, a) = \begin{cases} \kappa_t \sum_{i \in \mathcal{B}_t} \omega_{t,i} \bar{r}_{t,i}(a), & \mathcal{B}_t \neq \varnothing, \\ 0, & \mathcal{B}_t = \varnothing. \end{cases} \tag{13}$$

This normalization lets $\omega_{t,i}$ encode the relative importance of selected bottlenecks, while κ_t separately gates the total strength of the readiness signal.

We then choose the closest local policy that favors actions with positive readiness advantage via a KL-regularized update (Zhan et al., 2023):

$$q_t^* = \arg\max_{q \in \Delta(\mathcal{A}_t)} \left\{ \mathbb{E}_{a \sim q}[\widehat{A}_R(h_t, a)] - \tau D_{\mathrm{KL}}\big(q \| p_\theta^A(\cdot \mid h_t)\big) \right\}, \quad \tau > 0, \tag{14}$$

whose unique solution on the support of p_θ^A is

$$q_t^*(a) = \frac{p_\theta^A(a \mid h_t) \exp(\widehat{A}_R(h_t, a)/\tau)}{\sum_{a' \in \mathcal{A}_t} p_\theta^A(a' \mid h_t) \exp(\widehat{A}_R(h_t, a')/\tau)}. \tag{15}$$

For a minibatch decision-state index set $\mathcal{T}_{\mathrm{mb}}$, define

$$\mathcal{T}_B := \{t \in \mathcal{T}_{\mathrm{mb}} : \mathcal{B}_t \neq \varnothing\}. \tag{16}$$

The training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{GRPO}} + \lambda \mathcal{L}_{\mathrm{ready}}, \qquad \mathcal{L}_{\mathrm{ready}} = \begin{cases} \dfrac{1}{|\mathcal{T}_B|} \displaystyle\sum_{t \in \mathcal{T}_B} D_{\mathrm{KL}}\big(\mathrm{sg}(q_t^*) \| p_\theta^A(\cdot \mid h_t)\big), & \mathcal{T}_B \neq \varnothing, \\ 0, & \mathcal{T}_B = \varnothing. \end{cases} \tag{17}$$

We refer to the resulting procedure as BOTTLENECK DISTILLATION. While high-value certificates remain open, the auxiliary update shifts probability from premature commitment toward preparation actions. Once those bottlenecks are resolved, the readiness term vanishes and the outcome objective supports timely commitment in the now-ready state. In practice, certificate counterfactuals are scored in one packed teacher batch sharing the cached interaction prefix; the synchronized stop-gradient teacher is refreshed at each optimizer update. High-level action canonicalization, update ratios, and pseudocode are given in Appendix F.

4.4 REAL COMPLETION CALIBRATES THE SURROGATE

The environment audit starts from the same saved state and separates four branches: factual, certificate-only, real completion, and matched control. The certificate-only branch changes the target readiness certificate to SAT while leaving the environment and interaction history untouched. The real-completion branch executes the verified minimal operator that satisfies the prerequisite and retains every naturally induced observation and history update. The matched control consumes the same interaction/tool budget while leaving the prerequisite unresolved. Certificate-only versus real completion tests whether the semantic surrogate preserves the actual completion-induced policy shift; real completion versus matched control produces $\widehat{\mathrm{PCG}}$ and tests long-horizon value.

5 EXPERIMENTAL SETUP

Benchmarks. We evaluate five readiness regimes. Procedural tasks use τ^3 and MAC-Bench; MAC-Bench contributes dynamic adversarial pressure and executable atomic rules (Zhao et al., 2026d). Stateful interaction uses ToolSandbox and AppWorld-UL. Evidence-intensive tasks use τ-Knowledge and a text-only GAIA subset. Scientific discovery uses the verified ScienceAgentBench split, SciAgentGym, and DiscoveryBench. Scientific conditions are generated from tool dependencies, workflow artifacts, executable analyses, and evidence/validation requirements. Appendix E summarizes benchmark-specific readiness sources, and Appendix K reports per-benchmark results.

Table 1: Matched real-completion audit. PCG is the change in normalized terminal return over an equal-budget control; binary tasks reduce to success-probability gain.

Condition family	Pairs	Control return(%)	Completed return(%)	PCG↑	Irrelevant-action gain
Evidence sufficiency	514	37.8	58.5	+20.7	+1.4
Required state observed	436	44.6	61.0	+16.4	+1.1
Policy / compliance check	603	50.9	65.0	+14.1	+0.9
Clarification / confirmation	347	47.2	58.8	+11.6	+1.0
Scientific workflow prerequisite	421	33.5	49.6	+16.1	+1.6
Execution input / identifier	322	53.8	63.5	+9.7	+0.6
Macro average	2,643	44.6	59.4	**+14.8**	+1.1

Baselines and matched budgets. We compare GRPO, process-reward shaping (Deng et al., 2025; Li et al., 2026a; Shao et al., 2024), BEACON-style milestone credit (Wang et al., 2026), full-context OPSD/OPCD (Zhao et al., 2026a; Ye et al., 2026a), CREDIT (Shen et al., 2026a), and OPID (Yang et al., 2026). A dedicated same-privilege experiment gives every readiness method the same grounded condition table, cached on-policy states, teacher-call count, and reported FLOP budget. It compares scalar readiness reward, direct privileged action distillation, shuffled interventions, and RTA intervention.

Models. Main training uses Qwen3-4B-Instruct-2507. Cross-model evaluation includes gpt-oss-20b and Qwen3-30B-A3B-Instruct-2507. The thinking-model study uses Qwen3-8B with thinking enabled and disabled under the same training states. We report task success, self-correction/verification frequency, premature commitment, excess preparation, and readiness-calibration metrics.

Statistical protocol. Training comparisons are repeated across independent random seeds, while uncertainty for paired completion audits is estimated with benchmark-stratified, task-clustered bootstrap resampling so multiple condition pairs from the same task are not treated as independent. Main tables report point estimates for readability; seed-level and clustered confidence intervals are summarized alongside the central claims and in Appendix G.

6 Results: Readiness Signals Predict and Improve Long-Horizon Success

6.1 Completing Readiness Bottlenecks Raises Downstream Return

Matched continuation forks estimate Bottleneck Completion Advantage through PCG. Across 2,643 audited state–condition pairs, the macro-average across the six prerequisite families is a +14.8-point gain in normalized continuation return, while the corresponding family-macro control gain is +1.1 points (Table 1). For binary benchmarks this quantity is exactly the success-probability gain; graded scientific outcomes use their normalized benchmark score. The long-horizon effect spans policy, evidence, state, interaction, and scientific-workflow conditions.

The sufficient-component model yields two additional predictions. First, a route with open non-redundant components should exhibit a readiness ceiling: high continuation success is nearly absent below the route-readiness threshold, while fully ready states can still fail because U_k in Eq. 1 remains unresolved. Second, pathway substitution predicts that adding verified alternative routes that bypass the focal condition lowers the value of completing that condition, because successful continuations can proceed without it. Both patterns appear in Figure 2, turning the route-relative definition into two directly testable signatures.

6.2 Local Policy Sensitivity Predicts Long-Horizon Completion Value

The four-branch audit calibrates the readiness certificate against real bottleneck completion. The certificate-only branch changes the represented readiness state while preserving the original interaction history; the real-completion branch executes the prerequisite and exposes the full resulting context. In the shared canonical action space, the two induced policy shifts have cosine alignment 0.78, showing that the compact certificate preserves much of the direction of the real completion

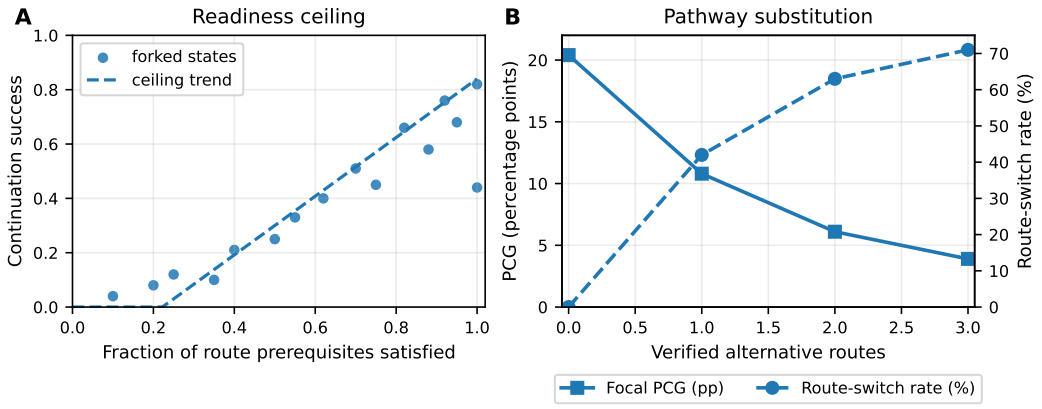

Figure 2: **Route-relative bottleneck signatures in the controlled testbed. A: Readiness ceiling.** Low route readiness constrains attainable continuation return: states with few satisfied route prerequisites rarely achieve high continuation success, while fully ready states can still fail because route-external factors remain unresolved. **B: Pathway substitution.** Adding verified alternative pathways reduces the value of completing the focal prerequisite and increases the fraction of continuations that switch to another route. The figure is rendered externally for readability; all values come from the controlled route generator described in Appendix D.

Table 2: Four-branch validity test averaged across condition families. Policy shift is D_{JS} from the factual branch; directional alignment is cosine similarity with the real-completion action shift.

Branch	Environment condition	Certificate state	Policy shift	Direction align↑	Terminal return
Factual	open	UNSAT	–	–	43.5
Certificate-only	open	SAT	0.181	0.78	44.9
Real completion	completed	SAT	0.206	1.00	**59.4**
Matched control	open	UNSAT	0.028	0.09	44.6

effect. The second calibration asks whether certificate sensitivity predicts long-horizon value: RTA–PCG Spearman correlation is $\rho = 0.66$ with a task-clustered 95% bootstrap interval of $[0.61, 0.70]$. Within-family calibration remains positive, and Appendix Table 14 shows that it remains stable across training checkpoints.

We next test ranking within structured candidate spaces. In the controlled falsification testbed, each state receives route-relevant prerequisites, correlated-but-nonblocking predicates, and irrelevant metadata predicates. Ranking by local sensitivity yields AUROC 0.86 and Precision@3 0.79, compared with 0.61/0.48 for raw teacher likelihood under full privileged context. We then repeat the negative-candidate construction inside τ^3, ToolSandbox, and GAIA-Text; Panel B of Table 3 tests whether the separation survives real benchmark states rather than relying on the controlled logical generator.

6.3 Readiness-Guided Policy Improvement Increases Task Success

Table 4 reports family-level task success. Here *premature commitment* is the fraction of action decisions that execute or commit while at least one high-value prerequisite remains unresolved, and *excess preparation* is the normalized number of additional search / inspect / ask actions taken after all relevant prerequisites have already been satisfied. Gains appear across procedural compliance, stateful interaction, search/evidence, and scientific discovery. MAC-Bench contributes an adversarial setting where task pressure encourages procedural shortcuts; science benchmarks add long workflows where analysis and validation prerequisites precede final scientific outputs. Bottleneck Distillation improves final success while reducing premature commitment and redundant preparation.

Table 3: Negative-candidate falsification. Panel A uses the controlled route testbed; Panel B repeats correlated-but-nonblocking candidates on real benchmark states.

	Setting / candidate	D_{JS} or target PCG	Control PCG	Ranking statistic
A	Route-relevant prerequisite	0.214	+15.1	Top-3 78.6%
	Correlated, non-blocking marker	0.071	+1.9	Top-3 24.3%
	Irrelevant metadata predicate	0.029	+0.4	Top-3 8.7%
B	τ^3	+13.7	+1.8	AUC 0.82
	ToolSandbox	+16.1	+2.0	AUC 0.84
	GAIA-Text	+19.2	+2.6	AUC 0.79

Table 4: Family-level main results on the instruction backbone. Scientific is the macro-average of ScienceAgentBench, SciAgentGym, and DiscoveryBench. Avg. success reports the measured mean.

Method	Procedural / MAC	Stateful	Evidence / Search	Scientific	Avg. success↑	Premature commit.↓	Excess prep.↓
GRPO	45.1	48.3	31.0	27.8	38.1±0.8	20.2	0.43
Process reward shaping	47.0	50.0	33.1	29.6	39.9±0.8	16.1	0.77
BEACON-style milestone RL	47.5	50.4	33.7	30.2	40.5±0.7	15.0	0.64
Full-context OPSD/OPCD	48.8	51.8	35.0	31.7	41.8±0.8	11.2	0.57
OPID	49.2	52.1	35.6	32.1	42.3±0.7	10.5	0.54
CREDIT	49.5	52.4	36.0	32.5	42.6±0.7	9.8	0.51
Bottleneck Distillation	**52.7**	**55.2**	**39.8**	**36.6**	**46.1±0.6**	**5.9**	**0.36**

A matched-budget experiment isolates the policy-improvement signal under identical readiness privilege. All methods receive the same cached student states, certificate table, and teacher-call/FLOP budget. RTA reaches 46.1 success, 5.9 premature commitment, and PCG-ranking P@3 of 0.79, compared with 43.4/10.2/0.55 for full-context action distillation under the same 1.25× budget. Appendix Table 13 reports the complete comparison; Appendix J additionally spends the extra compute on productive rollout / optimization work, with the same method ordering.

Commitment, transfer, and automatic construction. The learned policy changes both sides of the readiness decision: $P(\text{commit} \mid \text{all relevant SAT})$ is 69.2/72.5/76.8 for GRPO/CREDIT/BOTTLENECK DISTILLATION, whereas $P(\text{commit} \mid \exists\ \text{high-PCG UNSAT})$ is 21.4/10.6/5.4. The student receives no readiness certificates at evaluation and still outperforms CREDIT on Composition/Condition-Holdout/Evidence-Depth/Domain OOD (43.5/41.2/40.3/36.8 vs. 39.1/36.6/35.2/32.6) and on Qwen3-8B Thinking (52.6 vs. 49.2 success). Automatic LLM proposal plus mechanical verification reaches 96.8 Candidate F1, 97.4 Status F1, P@3=0.79, and 45.8 success versus 46.5 with manual double annotation. Full commitment, OOD/thinking, construction, cross-backbone, and grounding-noise results are in Appendices I and K.

7 CONCLUSION

Readiness bottlenecks are route-relative prerequisites whose real completion changes the value of the remaining trajectory. Our central result is that this expensive completion context can be represented, for learning, by a compact *readiness certificate*: certificate-only and real-completion action shifts align at 0.78 cosine similarity, while RTA sensitivity correlates with downstream PCG at $\rho = 0.66$ across 2,643 audited state–condition pairs. Across the six prerequisite families, real completion yields a +14.8-point macro-average gain in normalized continuation return. These calibrations establish the certificate as a decision-relevant semantic surrogate of bottleneck completion.

Bottleneck Distillation turns that calibrated surrogate into a readiness-aware policy update, raising average success from 42.6 to 46.1 over the strongest non-ours baseline while reducing premature commitment from 9.8 to 5.9. The resulting interface separates expensive execution from dense learning: real completion defines and validates bottleneck value, while the compact certificate carries the decision semantics needed to train when an agent should keep preparing and when it is ready to act.

REPRODUCIBILITY STATEMENT

The task structure and terminal-outcome completion value are specified by Eqs. 1–3; the readiness certificate and surrogate-fidelity shifts are specified by Eqs. 4 and 11; the training method is specified by Eqs. 5–17, including explicit definitions of $\mathcal{B}_t$, $\omega_{t,i}$, κ_t, and $\mathcal{T}_B$. Appendix A gives the optional successor-occupancy and soft-value interpretations; Appendix B specifies automatic candidate construction and mechanical verification; Appendix C gives the exact four-branch continuation protocol; Appendix D defines the bottleneck falsification and route-substitution testbed; Appendix E gives benchmark-specific readiness mappings; Appendix F reports action canonicalization, packed candidate batching, teacher synchronization, GRPO/readiness update ratio, continuation allocation, hyperparameters, and compute matching; Appendix G reports seed-level and clustered uncertainty; Appendix H reports lexical and completion-operator robustness; Appendix I reports detailed generalization, commitment, and construction results; Appendix K reports per-benchmark results.

AI USE STATEMENT

Generative AI tools were used for literature discovery, research brainstorming, hypothesis and conceptual-framework refinement, experimental-design discussion, manuscript drafting/editing, LaTeX formatting, and code assistance. Human authors verified the resulting literature claims, mathematical formulations, experimental methodology, implementation, data processing, and result interpretation, and take responsibility for the final manuscript and artifacts.

REFERENCES

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Junzhi Chen, Harsh Trivedi, Jane Pan, Michael JQ Zhang, Tejas Srinivasan, Niranjan Balasubramanian, and Ashish Sabharwal. AppWorld-UL: Benchmarking diverse agent-user interactions for tool-use. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=cUXV9vtDXd.

Xiaoyu Chen, Shenao Zhang, Pushi Zhang, Li Zhao, and Jianyu Chen. Asking before acting: Gather information in embodied decision making with language models, 2023. URL https://arxiv.org/abs/2305.15695.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f12b4df26344f3be803c06b555252efe-Abstract-Conference.html.

Yuchen Deng, Shichen Fan, Naibo Wang, Xinkui Zhao, and See-Kiong Ng. AgentPro: Enhancing LLM agents with automated process supervision. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9981–10006, 2025. URL https://aclanthology.org/2025.emnlp-main.506/.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models, 2026. URL https://arxiv.org/abs/2607.05184.

Aristotelis Lazaridis, Dylan Bates, Aman Sharma, Brian King, Vincent Lu, and Jack FitzGerald. EDGE-OPD: Internalizing privileged context with evidence guided on-policy distillation, 2026. URL https://arxiv.org/abs/2605.23493.

Dawei Li, Yuguang Yao, Zhen Tan, Huan Liu, and Ruocheng Guo. ToolPRMBench: Evaluating and advancing process reward models for tool-using agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 12378–12391, 2026a. doi: 10.18653/v1/2026.findings-acl.602. URL https://aclanthology.org/2026.findings-acl.602/.

Mukai Li, Qingcheng Zeng, Tianqing Fang, Zhenwen Liang, Linfeng Song, Qi Liu, Haitao Mi, and Dong Yu. Verified critical step optimization for LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 39627–39639, 2026b. doi: 10.18653/v1/2026.findings-acl.1974. URL https://aclanthology.org/2026.findings-acl.1974/.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1160–1183, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

J. L. Mackie. Causes and conditions. *American Philosophical Quarterly*, 2(4):245–264, 1965.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards data-driven discovery with large language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/0d70af566e69f1dfb687791ecf955e28-Abstract-Conference.html.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: A benchmark for general ai assistants. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2311.12983.

Xuan-Phi Nguyen, Shrey Pandit, Yiran Zhao, Anurag Koul, Zeyu Liu, and Shafiq Joty. Privileged likelihood is not automatically value: Three checks for token credit in on-policy self-distillation. *arXiv preprint arXiv:2608.09263*, 2026.

Jiangweizhi Peng, Yuanxin Liu, Ruida Zhou, Charles Fleming, Zhaoran Wang, Alfredo Garcia, and Mingyi Hong. HiPER: Hierarchical reinforcement learning with explicit credit assignment for large language model agents, 2026. URL https://arxiv.org/abs/2602.16165.

Jinhu Qi, Muzhi Li, Jiahong Liu, Yuqin Shu, Dianzhi Yu, Shicheng Ma, Wenqian Cui, Yiyang Zhao, Yiyi Chen, Ruoxi Jiang, et al. Towards trustworthy agentic ai: a comprehensive survey of safety, robustness, privacy, and system security. *Academia AI and Applications*, 2(2), 2026.

Kenneth J. Rothman. Causes. *American Journal of Epidemiology*, 104(6):587–592, 1976.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guobin Shen, Lei Huang, Xiang Cheng, Chenxiao Zhao, Jindong Li, Dongcheng Zhao, and Xing Yu. From generic correlation to input-specific credit in on-policy self distillation, 2026a. URL https://arxiv.org/abs/2605.11613.

Yujiong Shen, Yajie Yang, Zhiheng Xi, Binze Hu, Huayu Sha, Jiazheng Zhang, Qiyuan Peng, Junlin Shang, Jixuan Huang, Yutao Fan, Jingqi Tong, Shihan Dou, Ming Zhang, Lei Bai, Zhenfei Yin, Tao Gui, Xingjun Ma, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. SciAgentGym: Benchmarking multi-step scientific tool-use in LLM agents, 2026b. URL https://arxiv.org/abs/2602.12984.

Quan Shi, Alexandra Zytek, Pedram Razavi, Karthik Narasimhan, and Victor Barres. τ-knowledge: Evaluating conversational agents over unstructured knowledge, 2026. URL https://arxiv.org/abs/2603.04370.

Xinyuan Song and Zekun Cai. Ask the world before acting: Budgeted environment probing for world-model calibration, 2026. URL https://arxiv.org/abs/2606.31422.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and Semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1.

Songjun Tu, Chengdong Xu, Qichao Zhang, Yiwen Ma, Yaocheng Zhang, Linjing Li, Dong Li, Xiangyuan Lan, and Dongbin Zhao. UCOB: Learning to utilize and evolve agentic skills via credit-aware on-policy bidirectional self-distillation, 2026. URL https://arxiv.org/abs/2606.29502.

Zixuan Wang, Yuchen Yan, Hongxing Li, Teng Pan, Dingming Li, Ruiqing Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Milestone-guided policy learning for long-horizon language agents, 2026. URL https://arxiv.org/abs/2605.06078.

Shuo Yang, Jinyang Wu, Zhengxi Lu, Yuhao Shen, Fan Zhang, Lang Feng, Shuai Zhang, Haoran Luo, Zheng Lian, Zhengqi Wen, and Jianhua Tao. OPID: On-policy skill distillation for agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2606.26790.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models, 2026a. URL https://arxiv.org/abs/2602.12275.

Ziang Ye, Wentao Shi, Yuxin Liu, Yu Wang, Zhengzhou Cai, Yaorui Shi, Qi Gu, Xunliang Cai, and Fuli Feng. Look before you leap: Autonomous exploration for LLM agents, 2026b. URL https://arxiv.org/abs/2605.16143.

Wenhao Zhan, Shicong Cen, Baihe Huang, Yuxin Chen, Jason D. Lee, and Yuejie Chi. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *SIAM Journal on Optimization*, 33(2):1061–1091, 2023. doi: 10.1137/21M1456789.

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, 2026a. URL https://arxiv.org/abs/2601.18734.

Yiyang Zhao, Huiyu Bai, and Xuejiao Zhao. Gem: Generative entropy-guided preference modeling for few-shot alignment of llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 38146–38155, 2026b.

Yiyang Zhao, Zhiqi Shen, and Xuejiao Zhao. Causally-motivated generative few-shot reward inference through efficient dpo. *Knowledge-Based Systems*, page 116645, 2026c.

Yiyang Zhao, Zhuo Zhang, Qingxuan Le, Lizhen Qu, and Zenglin Xu. Beyond Goodhart's law: A dynamic benchmark for evaluating compliance in multi-agent systems, 2026d. URL https://arxiv.org/abs/2606.07805.

Yuqing Zhou, Zhuoer Wang, Jie Yuan, Hong Wang, Samson Koelle, Ziwei Zhu, and Wei Niu. WISE-Flow: Workflow-induced structured experience for self-evolving conversational service agents, 2026. URL https://arxiv.org/abs/2601.08158.

A Long-Horizon and Soft-Value Interpretations

Discounted successor-occupancy view. The main paper defines BCA using terminal benchmark outcome R through Q_R^π in Eq. 2. For the standard discounted-reward interpretation, let $r(s,a)$ be a per-step reward, $\gamma \in (0,1]$ the discount factor, and τ_o the random duration of temporally extended operator o. Define the discounted reward accumulated during the operator as

$$R_o^{\text{option}}(s) := \mathbb{E}\left[\sum_{k=0}^{\tau_o-1} \gamma^k r(S_{t+k}, A_{t+k}) \mid S_t = s, o\right], \tag{18}$$

and define the post-option successor occupancy with discount measured from the original decision time:

$$\mu_o^\pi(s', a' \mid s) := \mathbb{E}\left[\sum_{k=\tau_o}^{\infty} \gamma^k \mathbf{1}\{S_{t+k} = s', A_{t+k} = a'\} \mid S_t = s, o, \pi\right]. \tag{19}$$

Then

$$Q_r^\pi(s,o) = R_o^{\text{option}}(s) + \langle \mu_o^\pi, r \rangle, \tag{20}$$

so the corresponding discounted-reward completion advantage satisfies

$$\text{BCA}_{r,i}^\pi(s) = \left(R_{o_i}^{\text{option}}(s) - R_{o_i^0}^{\text{option}}(s)\right) + \left\langle \mu_{o_i}^\pi - \mu_{o_i^0}^\pi, r \right\rangle. \tag{21}$$

This decomposition makes explicit that temporally extended completion can change both reward collected during the operator and the entire discounted future occupancy. It is an interpretation of long-horizon completion effects; the empirical target used for PCG remains the terminal-outcome value Q_R^π in Eq. 2. In finite-horizon tasks where R is implemented as the terminal return (in particular with $\gamma = 1$), the two views coincide directly.

Soft-value interpretation of RTA. Suppose the privileged teacher is locally represented as

$$p_T(a \mid h, z) \propto \rho(a \mid h) \exp\left(\frac{Q_T(h, z, a)}{\alpha}\right), \tag{22}$$

with the same action prior $\rho(a \mid h)$ under the matched UNSAT/SAT certificate intervention (Haarnoja et al., 2018). Let $\Delta Q_{T,i}(a) := Q_T(h_t, z_i{=}\text{UNSAT}, a) - Q_T(h_t, z_i{=}\text{SAT}, a)$. Subtracting the two log policies gives an action-independent partition-function constant plus $\Delta Q_{T,i}(a)/\alpha$. Centering with any normalized distribution therefore removes that constant; in particular, Eq. 10 yields

$$\alpha \bar{r}_{t,i}(a) = \Delta Q_{T,i}(a) - \mathbb{E}_{a' \sim p_\theta^A} \Delta Q_{T,i}(a'). \tag{23}$$

Thus the factual-minus-completed RTA convention is a centered readout of the teacher's open-versus-completed action-value contrast under the stated soft-value assumption.

B Automatic Candidate Construction

The candidate proposer receives only domain artifacts that would be available when engineering the agent environment: policy text, tool schemas, task templates, state schemas, and workflow documentation. It outputs a condition name, natural-language description, entity slots, provenance, and checker template. Candidate checkers are compiled into executable predicates when possible, and candidates failing syntax or type checks are discarded before training. For evidence-heavy and scientific tasks, candidate predicates may reference generated artifacts—for example retrieved source identifiers, executed analyses, unit-test outcomes, statistical checks, or output files—rather than hidden final answers.

The construction pipeline separates *candidate proposal* from *status grounding*. Candidate F1 in Table 12 measures agreement of the proposed condition set with the manual double-annotation reference, whereas Status F1 measures agreement of the resulting SAT/UNSAT groundings with that same reference. The manual row is therefore the reference row for these two F1 columns and its 100.0/100.0 entries are by construction rather than an independently estimated model score. Downstream P@3 and task success are evaluated separately and are not fixed by this reference convention.

Human-time accounting in Table 12 includes domain-specific proposal review, checker implementation, and debugging; reusable framework development is excluded. This distinction is important because the table is intended to compare the marginal effort needed to instantiate a new domain, not the one-time cost of building the overall framework.

C Four-Branch Completion Protocol

For each sampled state–condition pair, we store a deterministic environment checkpoint and construct four branches from the same pre-intervention state. (1) *Factual*: the current open certificate and context are left unchanged. (2) *Certificate-only*: the target readiness certificate is rendered as satisfied while the environment and interaction history remain untouched. (3) *Real completion*: a verified minimal sequence satisfies the target predicate and all naturally induced observations and history changes are retained. (4) *Matched control*: a same-budget sequence leaves the target predicate unresolved. The completion and control operators are therefore part of the definition of the audited state–condition pair; PCG is interpreted only for a pair for which both branches are valid from the stored checkpoint.

The control branch is required to preserve the target predicate as unresolved rather than to imitate a successful completion. In retrieval/evidence settings, this means that the control interaction does not add the target answer-bearing evidence; in clarification settings, it does not resolve the target ambiguity or confirmation; and in stateful tool settings, it does not change the target prerequisite. These controls are benchmark-specific because the native interaction unit differs across environments, but the invariant is the same: the two branches begin from the identical checkpoint and differ in whether the focal prerequisite is actually completed.

Each real-completion and matched-control branch is followed by $M = 8$ independent continuations using the identical policy checkpoint. The estimator

$$\widehat{\mathrm{PCG}}_{t,i} = \frac{1}{M}\sum_{m=1}^{M} R_{t,i}^{+,m} - \frac{1}{M}\sum_{m=1}^{M} R_{t,i}^{0,m} \tag{24}$$

therefore compares downstream return after satisfying the focal condition against the matched unresolved branch from the same state. Certificate-only and real-completion branches are not interchangeable: the former changes only the compact semantic representation seen by the teacher, whereas the latter changes the environment/history and defines the long-horizon audit target.

For surrogate fidelity, certificate-only and real-completion effects are mapped to the same canonical high-level action space before the cosine in Eq. equation 11 is computed. Pairs for which either induced shift has zero ℓ_2 norm are omitted from that cosine statistic; they remain conceptually distinct from pairs with an informative non-zero shift.

D Controlled Bottleneck and Falsification Testbed

We use a small forkable logical testbed to isolate the mechanism claims from benchmark-specific confounders. A single-route task has the form $A \wedge B \wedge C \to Y$; a multi-route task augments it with an alternative such as $A \wedge D \wedge C \to Y$. The generator varies route structure, bottleneck depth, condition redundancy, and distractor predicates while retaining exact knowledge of which predicates are required by the currently available route.

The ground-truth label of a route-relevant bottleneck is determined by logical executability: satisfying that predicate changes whether the current route can progress, whereas a correlated marker may co-occur with route progress without changing executability. Irrelevant metadata is generated to be independent of the route logic. This gives three qualitatively different candidate classes used in the falsification analysis: route-relevant prerequisites, correlated but non-blocking markers, and irrelevant predicates.

Route substitution is created by adding an alternative successful route that bypasses the focal condition. Under this intervention, the same predicate can remain correlated with one route while becoming less valuable to complete because another executable route exists. The testbed is used only for mechanism identification, falsification, and route-substitution stress tests; benchmark task-success results are reported separately on the real environments.

E BENCHMARK-SPECIFIC READINESS MAPPING

Table 5: Benchmark coverage and primary readiness source.

Family	Benchmarks	Typical bottleneck candidates	Forkable state	Primary outcome
Procedural / compliance	τ^3, MAC-Bench	verification, consent, eligibility, policy steps	yes	success / compliant success
Stateful interaction	ToolSandbox, AppWorld-UL	observed state, IDs, clarification, confirmation	yes	task success
Evidence / search	τ-Knowledge, GAIA-Text	answer-bearing evidence, source coverage	partial/yes	answer success
Scientific tool use	SciAgentGym	tool dependencies, intermediate artifacts, checks	yes	workflow success
Scientific discovery	ScienceAgentBench, DiscoveryBench	data readiness, analysis validity, evidence support	replayable	normalized execution / discovery score

τ^3 and MAC-Bench. Conditions come from policy obligations, executable atomic rules, user-confirmation requirements, and tool preconditions. MAC-Bench's dynamic rule/scenario generation is preserved; readiness checkers operate on the generated sandbox trace.

ToolSandbox and AppWorld-UL. Conditions include known identifiers, observed target state, resolved ambiguity, user-supplied information, confirmation, and feasibility checks.

τ-Knowledge and GAIA. Candidate conditions represent answer-bearing evidence units, source coverage, and retrieval/verification requirements. Only text tasks with verifiable evidence maps are included in the intervention audit.

ScienceAgentBench. We use the verified benchmark split. Candidate conditions are grounded in observable workflow artifacts such as dataset inspection, required package/data availability, executable code, output generation, and validation steps defined by the evaluation harness.

SciAgentGym. Conditions are derived from tool dependency structure, required intermediate artifacts, and successful completion of upstream scientific operations before downstream tools become meaningful.

DiscoveryBench. Conditions are constructed from the benchmark's data-driven discovery facets: data access, variable/relationship inspection, statistical analysis artifacts, and evidentiary support for candidate hypotheses. Its graded discovery metric is normalized to $[0, 1]$ and used directly as terminal return in PCG; no binary threshold is introduced.

OOD split semantics. The OOD columns in Table 10 test different kinds of structural transfer rather than introducing new benchmark outcomes. *Composition* holds out combinations of readiness conditions while the constituent condition types remain represented during training. *Condition-Holdout* removes privileged readiness supervision for selected condition templates/types while preserving the ordinary task interface. *Evidence-Depth* evaluates tasks requiring a deeper set of independently necessary evidence units than the readiness-supervised training cases. *Domain* withholds a task domain from readiness supervision and evaluates the resulting student without privileged certificates at test time. These splits are used only for generalization analysis; the primary in-distribution outcome remains benchmark-native task success.

F IMPLEMENTATION AND BUDGET MATCHING

Candidate templates are produced offline once per domain. At training time, all currently active candidates are grounded before scoring; factual and counterfactual teacher branches share the cached interaction prefix and are packed into one batch. Actions are canonicalized to tool names plus explicit `search`, `inspect`, `ask`, `execute`, and `commit` categories; tool arguments and free-form reasoning remain governed by the base RL objective. The stop-gradient teacher is synchronized to the student parameters at each optimizer update before the next rollout batch. We apply one readiness auxiliary update for each GRPO optimizer update. We compute D_{JS} with natural logarithms.

The readiness-specific settings reported in the current experiment are $K = 4$, $\tau = 0.7$, $\beta = 0.05$, and $\lambda = 0.3$. The $M = 8$ continuation count belongs to the completion audit in Appendix C; it is not a dense training-time rollout multiplier. Pairs for which either certificate-induced or real-completion policy shift has zero ℓ_2 norm are excluded from the cosine-fidelity statistic, as noted above.

Compute matching. The same-privilege comparison matches the number of forward-scored action distributions: baselines receive dummy or repeated teacher scoring when needed so that the readiness methods are compared at the same reported teacher-call/FLOP budget. Tables 13 and 15 report compute as relative theoretical FLOPs normalized to the GRPO reference. The productive-compute control separately spends the additional $1.25\times$ budget on extra rollout/optimizer work rather than on dummy teacher scoring. Because the quantitative tables in this manuscript report relative FLOPs rather than absolute elapsed time, the cross-method compute claim is stated in FLOP/teacher-call terms and no additional wall-clock number is inferred.

Final uncertainty runs use independent training seeds and preserve the same data split and rollout budget per seed.

G Statistical Reliability

Table 6: Within-family calibration of local RTA sensitivity against PCG. Confidence intervals use task-clustered bootstrap resampling.

Condition family	Spearman ρ	95% CI	Certificate→real cosine
Evidence sufficiency	0.70	[0.64, 0.76]	0.82
Required state observed	0.65	[0.57, 0.72]	0.80
Policy / compliance check	0.62	[0.55, 0.69]	0.77
Clarification / confirmation	0.58	[0.49, 0.66]	0.74
Scientific workflow prerequisite	0.64	[0.55, 0.72]	0.79
Execution input / identifier	0.53	[0.44, 0.61]	0.71
All pairs	**0.66**	**[0.61, 0.70]**	**0.78**

Training success and premature-commitment metrics are summarized over independent seeds. Completion-audit uncertainty is computed with benchmark-stratified task/episode-clustered bootstrap resampling so that multiple state–condition pairs from one trajectory are not treated as independent resamples. Table 7 gives the stability summary; the $\pm$ entries for training success are across the three training seeds, whereas bracketed intervals are clustered 95% confidence intervals for the audit statistics.

Table 7: Seed and clustered uncertainty summary.

Method / statistic	Point estimate	SD / 95% CI	Unit
GRPO avg. success	38.1	$\pm$0.8	3 training seeds
CREDIT avg. success	42.6	$\pm$0.7	3 training seeds
Bottleneck Distill. avg. success	46.1	$\pm$0.6	3 training seeds
RTA–PCG Spearman ρ	0.66	[0.61,0.70]	clustered bootstrap
Family-macro PCG	+14.8	[+13.2,+16.3]	clustered bootstrap

The pooled "All pairs" correlation is computed on the pooled audited state–condition pairs and is not obtained by averaging the six family-level correlations. Likewise, the family-macro PCG is an unweighted macro-average over the six prerequisite families reported in the main completion audit. These aggregation conventions prevent larger families from silently dominating the corresponding headline quantities.

H Proxy and Completion-Operator Robustness

The readiness certificate is a representation choice, whereas BCA is defined relative to a concrete completion/control operator pair. We therefore test the two sensitivities separately. Table 8 changes

only the surface form used to encode readiness; Table 9 repeats the completion audit with two independently implemented, semantically equivalent completion operators for the same predicate.

Table 8: Readiness-certificate encoding robustness.

Status encoding	RTA–PCG ρ	P@3	Avg. success
UNSAT / SAT	0.66	0.77	46.1
NOT READY / READY	0.65	0.78	46.0
OPEN / CLOSED	0.66	0.75	46.1
0 / 1	0.63	0.76	45.9
Paraphrased condition text	0.64	0.78	45.9

Table 9: Completion-operator sensitivity for semantically equivalent ways of satisfying the same predicate.

Audit	Mean PCG	Pairwise PCG rank ρ	Top-3 overlap
Primary verified operator	+14.8	–	–
Alternative verified operator	+14.1	0.88	0.83

The lexical table tests whether the proxy depends on a particular literal tokenization of readiness. The completion-operator table addresses a different question: whether the downstream PCG ordering is stable when the same predicate can be satisfied through another verified realization. Neither robustness check redefines the primary audit; they test sensitivity around the fixed main protocol.

I Additional Generalization, Commitment, and Construction Results

Table 10 reports the four OOD structures defined in Appendix E. All OOD numbers are task-success measurements under the corresponding held-out readiness structure; the Qwen3-8B columns on the right are separate thinking-mode diagnostics and should not be averaged with the OOD block.

Table 10: Generalization and thinking-model results. Left block: task success under OOD readiness structure. Right block: Qwen3-8B thinking-mode diagnostics.

	OOD task success				Qwen3-8B Thinking			
Method	Composition	Cond.-Holdout	Evidence-depth	Domain	Success	Self-correct / ep.	Verify acts	Premature
GRPO	33.4	31.7	29.9	28.5	47.8	1.00	2.14	17.2
Privileged OPSD	38.0	35.7	34.5	31.9	46.1	0.71	1.63	11.8
CREDIT	39.1	36.6	35.2	32.6	49.2	0.93	2.02	9.9
Bottleneck Distillation	**43.5**	**41.2**	**40.3**	**36.8**	**52.6**	**0.98**	**2.21**	**6.1**

Conditional commitment diagnostic. Table 11 separates two evaluation strata: states in which all relevant prerequisites are satisfied and states containing at least one audited high-impact open prerequisite. The "high-PCG" terminology is used only for this post-hoc audit stratification; PCG itself is continuous, and Bottleneck Distillation does not consume PCG at inference. The primary premature-commitment metric in the main result is defined directly from unresolved relevant prerequisites rather than from this conditional table. Thus Table 11 is evidence about the *direction* of the learned commitment policy—suppression when an important prerequisite remains open without delaying commitment once prerequisites are satisfied—not a second training objective.

Table 11: Conditional commitment diagnostic. "High-PCG" denotes the audited high-impact stratum used for this diagnostic; the primary premature-commitment metric is reported separately.

Method	$P(\text{commit} \mid \text{all relevant SAT})$	$P(\text{commit} \mid \exists \text{ high-PCG UNSAT})$
GRPO	69.2	21.4
CREDIT	72.5	10.6
Bottleneck Distillation	**76.8**	**5.4**

Automatic construction metrics. Candidate F1 and Status F1 are measured against the same manual double-annotation reference described in Appendix B; accordingly, the reference row is 100.0/100.0 by definition. The remaining columns are independent downstream quantities: human minutes quantify domain-specific construction effort, P@3 evaluates bottleneck ranking quality, and Success is the resulting task-success measurement.

Table 12: Automatic condition construction. Human min./domain counts domain-specific proposal review, checker implementation, and debugging; reusable framework development is excluded. The manual double-annotation row is the reference for Candidate/Status F1.

Condition source	Candidate F1	Status F1	Human min./domain	P@3	Success
Manual double annotation (reference)	100.0	100.0	205	0.84	46.5
LLM proposal + mechanical verifier	**96.8**	**97.4**	**29**	**0.79**	**45.8**
LLM-only proposal + grounding	86.1	83.7	0	0.63	42.5

Real-benchmark route substitution case study. In ToolSandbox reservation tasks, we observe explicit route substitution when an initially preferred restaurant becomes unavailable: once an alternative bookable restaurant is verified, the PCG of re-checking the original restaurant drops from +12.9 to +4.1, while the continuation route-switch rate increases from 0.18 to 0.67. This mirrors the controlled-generator prediction in a real benchmark state.

Training procedure. Candidate templates are generated once per domain. For each on-policy decision state, we ground the fixed candidate set, score the factual teacher and all open-condition certificate counterfactuals in one packed batch, retain the top-K open hypotheses by D_{JS}, build the readiness advantage and KL-regularized target, and apply the auxiliary action-level update together with the GRPO update. The procedure can be summarized as:

Algorithm 1 Training-time Bottleneck Distillation update.

Require: on-policy state h_t, fixed candidate templates, synchronized stop-gradient teacher

1: Ground all active candidate predicates at h_t
2: Score the factual certificate context and each open-condition UNSAT→SAT counterfactual in one packed teacher batch
3: Compute D_{JS} and retain the top-K open candidates
4: Form the centered RTA signals, readiness weights, and the readiness-aware action advantage
5: Construct the KL-regularized local target $\mathcal{T}_B$
6: Apply one readiness auxiliary update together with the GRPO optimizer update
7: Synchronize the stop-gradient teacher before the next rollout batch

J Training-Stage Calibration and Productive-Compute Controls

Table 13: Same-privilege, same-compute comparison. Relative FLOPs are normalized to the GRPO reference.

Use of identical readiness privilege	Success↑	Premature↓	PCG-ranking P@3↑	Relative FLOPs
Scalar readiness reward	42.0	13.1	–	1.25×
Direct condition distillation	43.1	10.9	0.53	1.25×
Condition-shuffled intervention	40.6	17.4	0.18	1.25×
Full-context action distillation	43.4	10.2	0.55	1.25×
RTA intervention	**46.1**	**5.9**	**0.79**	1.25×

Table 14: RTA–PCG calibration across training checkpoints. Confidence intervals use task-clustered bootstrap resampling.

Checkpoint	RTA–PCG ρ	95% CI
Initial policy	0.58	[0.52, 0.63]
25% of training	0.61	[0.56, 0.67]
50% of training	0.63	[0.58, 0.68]
75% of training	0.65	[0.60, 0.69]
Final policy	0.66	[0.61, 0.70]

Table 15: Productive-compute control. The additional 1.25× budget is spent on extra rollouts / optimizer work rather than dummy teacher scoring.

Method	Relative FLOPs	Avg. success↑	Premature commit↓
GRPO	1.00×	38.1	20.2
GRPO (productive 1.25×)	1.25×	39.4	18.6
CREDIT	1.00×	42.6	9.8
CREDIT (productive 1.25×)	1.25×	43.5	8.9
Bottleneck Distillation	1.25×	**46.1**	**5.9**

The two controls answer different questions. Table 13 holds readiness privilege and reported scoring/FLOP budget fixed across competing readiness objectives. Table 15 asks whether spending the same additional compute on ordinary rollout/optimization work accounts for the gain. No new outcome metric is introduced in either comparison.

K Full Per-Benchmark Results

Table 16: Per-benchmark task success. Cells are point estimates; seed-level dispersion for the corresponding macro results is summarized in Table 7.

Method	τ^3	MAC	ToolSB	AppW-UL	τ-K	GAIA-T	SciAgentBench	SciAgentGym	DiscoveryBench
GRPO	47.0	43.2	55.8	40.8	32.4	29.6	28.2	30.4	24.8
Full OPSD/OPCD	50.3	47.3	59.1	44.5	36.3	33.7	31.7	34.9	28.5
CREDIT	51.0	48.0	59.8	45.0	37.1	34.9	32.6	35.7	29.2
Bottleneck Distill.	**54.1**	**51.3**	**62.5**	**47.9**	**41.2**	**38.4**	**36.8**	**39.6**	**33.4**

Aggregation convention for the main table. The family-level values in the main result are obtained by first macro-averaging the constituent benchmarks within each of four families: τ^3/MAC for Procedural/MAC, ToolSandbox/AppWorld-UL for Stateful, τ-Knowledge/GAIA-Text for Evidence/Search, and ScienceAgentBench/SciAgentGym/DiscoveryBench for Scientific. "Avg. success" is then the unweighted macro-average of these four family values. It is therefore a four-family macro-average, not the unweighted mean of all nine benchmark columns in Table 16. This convention gives each task family equal weight despite the Scientific family containing three benchmarks.

Grounding noise. At noise levels 0, 5, 10, 15, and 20%, success is 46.1, 45.5, 44.6, 43.2, and 41.7, respectively; premature commitment is 5.9, 6.4, 7.5, 9.0, and 11.2%.

Cross-backbone consistency. On gpt-oss-20b and Qwen3-30B-A3B-Instruct-2507, Bottleneck Distillation gains over matched GRPO are +6.1 and +5.7 average success points respectively, with the same sign on all benchmark families.