CHILL-Harness: Counterfactual Harness Learning for Efficient Reasoning in Long-Horizon Agents

Jiarun Fu[1] Lizhong Ding[1,*] Sida Chen[1] Honglei Xin[1] Chunhui Zhang[1]
Pengqi Li[1] Qiuning Wei[1] Ye Yuan[1] Guoren Wang[1]

[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China

Contact: jrfu@bit.edu.cn

Abstract

Agent harnesses have become the operational infrastructure of modern large language model agents, coordinating context, tools, verification, and execution control to translate latent model capability into reliable long-horizon behavior. However, reliable long-horizon behavior requires harness control to adapt to task demands, execution environments, and evolving execution states, whereas current harnesses predominantly rely on hand-crafted or globally fixed policies; this mismatch manifests as unnecessary computational overhead and, in adverse cases, reduced task success. To address this limitation, we formulate the task of enabling adaptive orchestration in harness systems as a causal learning problem and propose **C**ounterfactual **H**arness **I**ntervention **L**earning for **L**ong-Horizon Agents (**CHILL-Harness**). CHILL-Harness intervenes at the orchestration layer to enable advantage-guided workflow adaptation, thereby improving reasoning and execution efficiency while preserving task performance. Specifically, we develop causal intervention effect learning as the effect-estimation component of CHILL-Harness to estimate intervention-relative workflow advantage from confidence-weighted execution evidence and identify advantageous workflow adaptations. We further introduce advantage-realizing causal orchestration as its realization component to adaptively allocate counterfactual reasoning and realize only workflow adjustments supported by sufficient expected advantage. Finally, we incorporate a success-preserving objective and advantage-margin authorization constraints into CHILL-Harness to promote reliable adaptation. Extensive experiments on heterogeneous long-horizon tasks spanning information seeking, software engineering, and terminal interaction show that CHILL-Harness consistently preserves or improves task success while substantially reducing token consumption and execution time.

Code: https://github.com/csdstar/chill-dev

Introduction

Large language model agents are increasingly deployed on long-horizon tasks Zhou et al. (2024); Xie et al. (2024) that require sustained context management Wang et al. (2025b), external tool interaction Yao et al. (2025), intermediate verification Pan et al. (2025), failure recovery Wang and Liu (2025), and execution control Guo et al. (2026). Agent harnesses have therefore become critical runtime infrastructure for translating latent model capability into reliable behavior Meng et al. (2026); Kapoor et al. (2026). However, effective harness control should adapt to task demands Hu, Wang, and McAuley (2025), execution environments Ba, Li, and Li (2026), and evolving execution states Zhang et al. (2026), whereas existing harnesses remain largely governed by hand-crafted rules, fixed thresholds, or globally configured workflows Li et al. (2026); Marchand et al. (2026). Over long execution horizons, this mismatch can compound redundant reasoning, delayed correction, and unnecessary workflow disruption, increasing execution cost and potentially degrading task success He et al. (2026); Sun et al. (2026).

Existing harness-level approaches to adaptive harness orchestration can be broadly grouped into two categories: (1) task-oriented engineered harnesses such as Terminus-KIRA integrate hand-engineered reasoning and tool-use procedures across long-horizon tasks KRAFTON AI and Ludo Robotics (2026), OpenHands and CodeSweep–SWE-agent provide iterative repository inspection, editing, and verification workflows for software repair Wang et al. (2025a); Yang et al. (2024); Kimi Team (2025), and LemonHarness develops a high-performing execution stack for complex terminal tasks Ren et al. (2026); despite their strong task-level performance, these systems rely primarily on domain-specific or globally configured control logic and provide limited adaptation to the value of an intervention at the current execution state; and (2) orchestration and harness optimization methods improve system-level coordination—AWorld dynamically organizes multi-agent collaboration Yu, Lu, and Chenyi Zhuang (2025), OWL Workforce coordinates a planner, coordinator, and specialized workers through hierarchical orchestration Hu et al. (2025b), and Meta-Harness performs outer-loop search over harness implementations using execution traces and evaluation feedback Lee et al. (2026); however, these methods primarily optimize collaboration structures or aggregate harness configurations rather than determine whether a concrete orchestration intervention causally improves the ongoing workflow. Consequently, existing methods still lack a unified framework that assesses whether a concrete orchestration adjustment improves the current workflow and translates

this assessment into adaptive, success-preserving harness decisions for efficient reasoning.

In this work, we leverage a causal perspective to address this limitation. Causal reasoning is well suited to decisions whose observed outcomes conflate the effect of an action with the context in which it is taken Kuroki and Pearl (2014), since causal interventions can isolate the contribution of the action under a fixed context Rubin (1980); Pearl (2009). It has therefore been increasingly used to distinguish genuine contributions from spurious associations in large-language-model reasoning and autonomous driving Chi et al. (2024); Pourkeshavarz, Zhang, and Rasouli (2024); Tang et al. (2026). Harness orchestration exhibits the same structure: the current workflow provides a factual reference, while admissible alternatives define counterfactual interventions under the same execution context. We therefore formulate the task of enabling adaptive orchestration in harness systems as a causal learning problem. Specifically, we introduce the *causal harness orchestration problem*:

> *How can a harness leverage causal interventions to realize efficient and admissible workflow orchestration while preserving task success?*

To address this problem, we propose **C**ounterfactual **H**arness **I**ntervention **L**earning for **L**ong-Horizon Agents (**CHILL-Harness**), a causal framework for adaptive harness orchestration and efficient long-horizon reasoning. CHILL-Harness intervenes at the orchestration layer to enable advantage-guided workflow adaptation. Specifically, we develop Causal Intervention Effect Learning (CIEL) as its effect-estimation component, which unifies heterogeneous orchestration behaviors and estimates their context-conditioned intervention-relative advantages from confidence-weighted execution evidence. We further introduce Advantage-Realizing Causal Orchestration (ARCO) as its realization component, which routes counterfactual deliberation before candidate generation and authorizes workflow adaptations only when they exhibit sufficient causal–operational advantage. Finally, we incorporate a success-preserving objective and advantage-margin authorization constraints to promote reliable orchestration under effect-estimation uncertainty. Extensive experiments on long-horizon tasks spanning information seeking, software engineering, and terminal interaction show that CHILL-Harness preserves or improves task success while substantially reducing token consumption and execution time. Our contributions are summarized as follows:

- We propose CHILL-Harness, which formulates adaptive orchestration in harness systems as a causal learning problem over factual and admissible alternative workflows, enabling efficient reasoning through intervention.
- We introduce Causal Intervention Effect Learning, which unifies heterogeneous orchestration behaviors and learns their context-conditioned intervention-relative advantages from confidence-weighted execution evidence, enabling selective harness-level intervention.
- We develop Advantage-Realizing Causal Orchestration, which realizes workflow adaptations supported by sufficient estimated advantage, thereby achieving efficient and reliable harness orchestration under success-preserving constraints.

Preliminaries and Related Work

This section introduces the agent-harness setting and causal formulation, followed by a review of adaptive harness orchestration and causal intervention methods.

Preliminaries: Causal Adaptation in Harnesses

An agent harness is the runtime infrastructure that governs how a base language-model agent maintains context, invokes tools, verifies progress, and interacts with its task environment (Meng et al., 2026; Guo et al., 2026). At execution step t, the current context is represented as

$$\chi_t = \left(s_t, c_t^{\text{env}}, g_t\right),$$

where s_t, c_t^{env}, and g_t denote the execution state, environment context, and current task objective, respectively. Given χ_t, the base agent proposes a factual workflow ω_t^0, consisting of the reasoning, tool-use, and execution operations.

Let $\mathcal{A}_t$ denote the admissible workflow set under the current task, environment, safety, and resource constraints, with $\omega_t^0 \in \mathcal{A}_t$, and define $\mathcal{A}_t^{\text{cf}} = \mathcal{A}_t \setminus \{\omega_t^0\}$. The workflow selected for execution therefore satisfies $W_t \in \{\omega_t^0\} \cup \mathcal{A}_t^{\text{cf}}$. Executing $W_t = \omega$ produces downstream task performance U_t under the common continuation and evaluation protocol. Adaptive harness orchestration determines whether to retain ω_t^0 or execute an admissible alternative under χ_t.

This decision naturally induces a potential-outcome formulation (Rubin, 1974; Pearl, 2009). For each $\omega \in \mathcal{A}_t$, let $U_t(\omega)$ denote the performance under $\text{do}(W_t = \omega)$ with χ_t fixed. Under intervention consistency, $U_t = U_t(W_t)$, and the workflow effect relative to the factual workflow is $\Gamma_t(\omega) = \mathbb{E}\left[U_t(\omega) - U_t(\omega_t^0) \mid \chi_t\right], \Gamma_t(\omega_t^0) = 0$.

Related Work: Adaptive Harness Orchestration and Causal Intervention

Existing work on adaptive harness orchestration includes engineered execution systems, automated harness optimization, and component-level adaptation. OpenHands, SWE-agent, and OWL improve execution through specialized interfaces or coordination (Wang et al., 2025a; Yang et al., 2024; Hu et al., 2025b); Automated Design of Agentic Systems, Multi-Agent Architecture Search, and Meta-Harness optimize agent structures, resources, or complete harnesses (Hu, Lu, and Clune, 2025; Zhang et al., 2025; Lee et al., 2026), while Agent Workflow Memory, HiAgent, and OSCAR adapt workflow reuse, context management, and recovery (Wang et al., 2025b; Hu et al., 2025a; Wang and Liu, 2025). Causal methods have also been explored for language-model reasoning and agent decision making. Causal Sufficiency and Necessity refines reasoning chains through counterfactual analysis (Yu

et al., 2025), Counterfactual Planning revises task-level actions using structural causal models (Fu et al., 2026), and Robust Agents Learn Causal World Models exploits causal environment structure for robust control (Richens and Everitt, 2024).

These methods target reasoning steps, agent actions, or environment models, but do not assess heterogeneous harness-level decisions or efficiently realize only workflow-improving adaptations. CHILL-Harness fills this gap by estimating context-conditioned workflow effects and translating them into success-preserving workflow adaptations.

CHILL-Harness: Counterfactual Harness Intervention Learning

CHILL-Harness decomposes causal harness orchestration into two coupled problems.

Definition 1 (Harness Intervention Effect Problem)**.** *The Harness Intervention Effect Problem evaluates whether an admissible workflow alternative improves execution relative to the factual workflow under the same execution context.*

Definition 2 (Harness Intervention Realization Problem)**.** *The Harness Intervention Realization Problem determines whether the estimated benefit of an admissible alternative is sufficient to justify replacing the factual workflow.*

Following the preliminaries, χ_t, ω_t^0, $\mathcal{A}_t$, and $\Gamma_t(\omega)$ denote the execution context, factual workflow, admissible workflow set, and expected gain of ω over ω_t^0, respectively. As illustrated in Figure 1, CHILL-Harness addresses the two problems through **Causal Intervention Effect Learning (CIEL)** and **Advantage-Realizing Causal Orchestration (ARCO)**, under explicit success-preserving workflow authorization constraints.

CIEL: Causal Intervention Effect Learning

CIEL learns workflow-grounded intervention preferences for the Harness Intervention Effect Problem and converts them into factorized deliberation, revision, and completion signals.

Unified Causal Intervention Induction. CIEL abstracts recurring harness adaptations, including additional reasoning, workflow correction, execution recovery, and answer construction, into a unified intervention space (Wang et al., 2025b; Yang et al., 2024; Wang and Liu, 2025). Let

$$\mathcal{R} = \{\textsc{Inspect}, \textsc{Verify}, \textsc{Visual}\}, \quad \mathcal{Q} = \{\textsc{Dedup}, \textsc{NoOpRecovery}\}$$

denote the revision and stabilization modes. The intervention space and its workflow realization are

$$\mathcal{I}_t^{\mathrm{wf}} = \{\textsc{Keep}, \textsc{Deliberate}, \textsc{AnswerSynthesis}\} \cup \{\textsc{Revise}(r) \mid r \in \mathcal{R}\} \cup \{\textsc{Stabilize}(q) \mid q \in \mathcal{Q}\},$$

$$\mathcal{G}_t(\iota) \equiv \mathcal{G}\left(\iota; \chi_t, \omega_t^0\right) \subseteq \mathcal{A}_t, \qquad \mathcal{G}_t(\textsc{Keep}) = \{\omega_t^0\}.$$

Here, $\mathcal{I}_t^{\mathrm{wf}}$ specifies adaptation intent and $\mathcal{G}_t(\iota)$ maps it to admissible workflows. Keep preserves the factual workflow; Deliberate requests additional reasoning; Revise acquires, verifies, or visually inspects evidence; Stabilize suppresses repetition or recovers ineffective execution; and Answer-Synthesis redirects execution toward nonterminal answer construction. Each intervention inherits its causal contribution from its realized workflow (details in Appendix A.2).

CIEL estimates the context-conditioned workflow effect as $\widehat{\Gamma}_\phi(\omega; \chi_t, \omega_t^0) \approx \Gamma_t(\omega)$, with the factual workflow serving as the zero-effect reference. The estimates induce intervention-family preferences that are amortized into online prediction:

$$\widehat{V}_t^{\mathrm{int}}(\iota) = \max_{\omega \in \mathcal{G}_t(\iota)} \widehat{\Gamma}_\phi\left(\omega; \chi_t, \omega_t^0\right), \tag{1a}$$

$$\iota_t^* = \arg\max_{\iota \in \mathcal{I}_t^{\mathrm{wf}}} \widehat{V}_t^{\mathrm{int}}(\iota), \tag{1b}$$

$$\widehat{\iota}_t = \arg\max_{\iota \in \mathcal{I}_t^{\mathrm{wf}}} p_{\theta_u}\left(\iota \mid \chi_t, \omega_t^0\right). \tag{1c}$$

Equations (1a)–(1c) separate workflow-effect aggregation, offline intervention induction, and amortized online prediction. Because $\mathcal{G}_t(\textsc{Keep}) = \{\omega_t^0\}$, conservative tie-breaking selects Keep whenever no alternative family has a positive estimated effect. The workflow-effect estimator learns from checkpointed paired records, while the intervention predictor amortizes the induced offline preference:

$$\mathcal{L}_{\mathrm{eff}} = \mathbb{E}_{\mathcal{D}_{\mathrm{pair}}}\left[w_{t,\omega}\left(\widehat{\Gamma}_\phi\left(\omega; \chi_t, \omega_t^0\right) - \widetilde{\Gamma}_t^{\mathrm{pair}}(\omega)\right)^2\right],$$

$$\mathcal{L}_{\mathrm{int}} = -\log p_{\theta_u}\left(\iota_t^* \mid \chi_t, \omega_t^0\right).$$

Here, $\mathcal{D}_{\mathrm{pair}}$ contains factual–candidate executions restored from the same context, $\widetilde{\Gamma}_t^{\mathrm{pair}}(\omega)$ is their empirical utility difference, and $w_{t,\omega}$ is its reliability weight.

For analysis, define $V_t^{\mathrm{int}}(\iota) = \max_{\omega \in \mathcal{G}_t(\iota)} \Gamma_t(\omega)$, $\iota_t^\dagger = \arg\max_{\iota \in \mathcal{I}_t^{\mathrm{wf}}} V_t^{\mathrm{int}}(\iota)$, and let $C_t = \bigcup_{\iota \in \mathcal{I}_t^{\mathrm{wf}}} \mathcal{G}_t(\iota)$. The workflow-effect error and amortization gap are

$$\varepsilon_t = \sup_{\omega \in C_t} \left|\widehat{\Gamma}_\phi\left(\omega; \chi_t, \omega_t^0\right) - \Gamma_t(\omega)\right|, \quad \rho_t = \widehat{V}_t^{\mathrm{int}}(\iota_t^*) - \widehat{V}_t^{\mathrm{int}}(\widehat{\iota}_t). \tag{2}$$

Then

$$V_t^{\mathrm{int}}(\iota_t^\dagger) - V_t^{\mathrm{int}}(\widehat{\iota}_t) \le 2\varepsilon_t + \rho_t. \tag{3}$$

Thus, effect-estimation and amortization errors jointly control intervention selection regret; Appendix A.3 proves the result, and Appendix A.4 details paired-effect construction.

Factorized Intervention Decision Learning. CIEL factorizes online control into three decisions governing additional deliberation, workflow revision, and task completion.

Counterfactual Deliberation Effect (CDE). CDE addresses harness over-deliberation, where costly planning is repeatedly invoked even when the factual workflow is already sufficient. It predicts the required deliberation route:

$$p_{\theta_p}\left(z_t \mid \chi_t, \omega_t^0, \widehat{\iota}_t\right), \qquad z_t \in \mathcal{Z} = \{\textsc{Skip}, \textsc{Light}, \textsc{Full}\}. \tag{4}$$

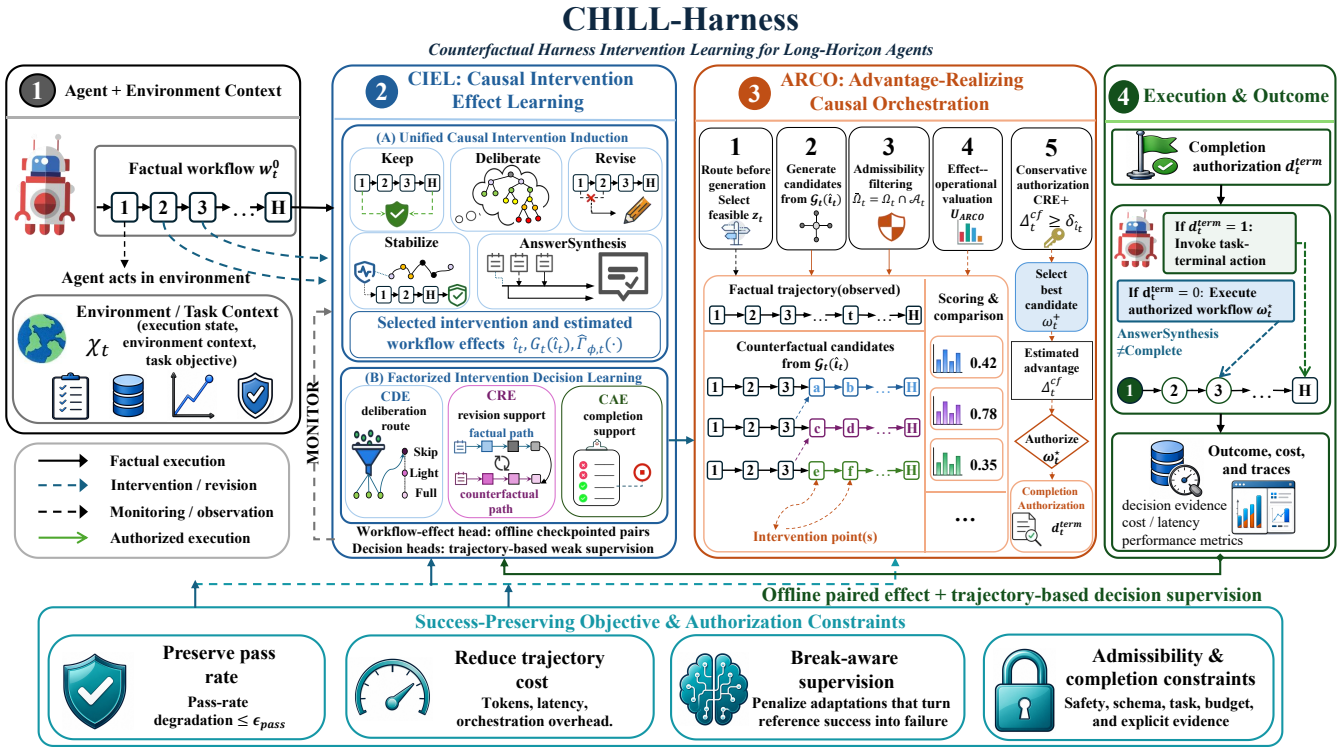

Figure 1: Overview of CHILL-Harness. Given context χ_t and factual workflow ω_t^0, CIEL estimates workflow effects and predicts intervention, deliberation, revision, and completion signals. ARCO realizes them through route-before-generation, admissibility filtering, causal–operational valuation, and conservative authorization. Offline checkpointed paired executions supervise workflow-effect estimation, while trajectory outcomes provide weak supervision for the factorized decision heads.

Skip bypasses candidate generation, Light enables restricted diagnostics, and Full enables broader evaluation. Thus, the intervention family specifies *what* adaptation is useful, whereas CDE determines *how much* computation realizes it.

Counterfactual Revision Effect (CRE). CRE addresses both workflow inertia and harmful replacement: the factual workflow may require correction, while an unsupported alternative may further degrade execution. Given an alternative workflow ω, CRE predicts whether it provides sufficient support for replacing the factual workflow:

$$p_{\theta_c}\left(y_t^{\text{chg}} \mid \chi_t, \omega_t^0, \omega, \widehat{\iota}_t\right), \qquad y_t^{\text{chg}} \in \{\textsc{Keep}, \textsc{Change}\}. \tag{5}$$

Keep retains ω_t^0, whereas Change supports replacement by the evaluated alternative.

Completion Attribution Effect (CAE). CAE predicts completion support to prevent both continued execution after task success and termination triggered solely by futility:

$$p_{\theta_e}\left(b_t \mid \chi_t, o_t, h_t\right), \qquad b_t \in \{0, 1\}. \tag{6}$$

Here, o_t and h_t denote the latest observation and recent execution summary; $b_t = 1$ supports completion, which still requires explicit task evidence.

Logged execution traces provide weak targets z_t^*, $y_t^{\text{chg},*}$, and b_t^* for CDE, CRE, and CAE, respectively. Here, z_t^* denotes the preferred deliberation route, $y_t^{\text{chg},*}$ denotes the preferred revision decision, and b_t^* denotes whether retrospective task evidence supports completion. Efficiency-related decisions receive positive supervision only when task success is preserved. The corresponding losses are

$$\begin{aligned}
\mathcal{L}_{\text{CDE}} &= -\log p_{\theta_p}\left(z_t^* \mid \chi_t, \omega_t^0, \widehat{\iota}_t\right), \\
\mathcal{L}_{\text{CRE}} &= -\log p_{\theta_c}\left(y_t^{\text{chg},*} \mid \chi_t, \omega_t^0, \omega_t^+, \widehat{\iota}_t\right), \\
\mathcal{L}_{\text{CAE}} &= -\log p_{\theta_e}\left(b_t^* \mid \chi_t, o_t, h_t\right).
\end{aligned}$$

Thus, $\mathcal{L}_{\text{CDE}}$ supervises the amount of additional reasoning, $\mathcal{L}_{\text{CRE}}$ supervises factual-versus-candidate workflow selection, and $\mathcal{L}_{\text{CAE}}$ supervises evidence-attributed completion. The complete CIEL objective is

$$\mathcal{L}_{\text{CIEL}} = \mathcal{L}_{\text{eff}} + \mathcal{L}_{\text{int}} + \mathcal{L}_{\text{CDE}} + \mathcal{L}_{\text{CRE}} + \mathcal{L}_{\text{CAE}}. \tag{7}$$

Equation (7) jointly supervises workflow effects, intervention preferences, deliberation, revision, and completion. Detailed weak-target construction is provided in Appendix A.4.

Remark. CIEL separates *what* adaptation is beneficial from *how* it should be realized: workflow-effect learning induces intervention preferences, while CDE, CRE, and CAE factorize deliberation, revision, and completion decisions.

ARCO: Advantage-Realizing Causal Orchestration

ARCO addresses the Harness Intervention Realization Problem through route selection, candidate valuation, conservative revision, and evidence-grounded completion. Let $\widehat{\Gamma}_{\phi,t}(\omega) := \widehat{\Gamma}_{\phi}(\omega; \chi_t, \omega_t^0)$. Then

$$\text{ARCO}\left(\chi_t, \omega_t^0, o_t, h_t, \widehat{\iota}_t, \widehat{\Gamma}_{\phi,t}(\cdot)\right) \longmapsto \left(\omega_t^\star, d_t^{\text{term}}\right).$$

Using the CDE predictor defined in Equation (4), ARCO first selects the provisional deliberation route:

$$\widetilde{z}_t = \arg\max_{z \in \mathcal{Z}} p_{\theta_p}\left(z \mid \chi_t, \omega_t^0, \widehat{\iota}_t\right).$$

Under Skip $\prec$ Light $\prec$ Full, z_t is the highest feasible route not exceeding $\widetilde{z}_t$. Skip retains ω_t^0; otherwise, the selected route exposes $\Omega_t \subseteq \{\omega_t^0\} \cup \mathcal{G}_t(\widehat{\iota}_t), \overline{\Omega}_t = \Omega_t \cap \mathcal{A}_t$, where $\overline{\Omega}_t$ contains the admissible route-exposed workflows. Route feasibility is detailed in Appendix B.1.

Each admissible candidate is valued by combining operational utility and estimated workflow effect:

$$\widehat{U}_{\text{ARCO}}\left(\omega \mid \chi_t, z_t, \widehat{\Gamma}_{\phi,t}\right) = \widehat{U}_{\text{op}}\left(\omega \mid \chi_t, z_t\right) + \eta_\Gamma \widehat{\Gamma}_{\phi,t}(\omega). \tag{8}$$

Here, $\widehat{U}_{\text{op}}$ aggregates progress, cost, risk, information gain, robustness, and safety, while $\eta_\Gamma \geq 0$ weights estimated intervention evidence; Appendix B.2 gives the full construction.

ARCO selects the highest-valued workflow among the admissible route-exposed candidates:

$$\omega_t^+ = \arg\max_{\omega \in \overline{\Omega}_t} \widehat{U}_{\text{ARCO}}\left(\omega \mid \chi_t, z_t, \widehat{\Gamma}_{\phi,t}\right),$$

with advantage over the factual workflow

$$\Delta_t^{\text{cf}} = \widehat{U}_{\text{ARCO}}\left(\omega_t^+ \mid \chi_t, z_t, \widehat{\Gamma}_{\phi,t}\right) - \widehat{U}_{\text{ARCO}}\left(\omega_t^0 \mid \chi_t, z_t, \widehat{\Gamma}_{\phi,t}\right).$$

Instantiating the CRE predictor in Equation (5) with the selected candidate ω_t^+, ARCO obtains

$$y_t^{\text{chg}} = \arg\max_{y \in \{\text{Keep}, \text{Change}\}} p_{\theta_c}\left(y \mid \chi_t, \omega_t^0, \omega_t^+, \widehat{\iota}_t\right).$$

The candidate is executed only when both CRE and its estimated advantage support revision:

$$\omega_t^\star = \begin{cases} \omega_t^+, & y_t^{\text{chg}} = \text{Change} \ \wedge \ \Delta_t^{\text{cf}} \geq \delta_{\widehat{\iota}_t}, \\ \omega_t^0, & \text{otherwise.} \end{cases} \tag{9}$$

In Equation (9), $\delta_{\widehat{\iota}_t} \geq 0$ is an intervention-dependent authorization margin. ARCO executes ω_t^+ only when both the CRE decision and its estimated advantage support replacement; otherwise, it retains ω_t^0 (see Appendix B.3 for details).

Finally, ARCO instantiates the CAE predictor in Equation (6) and combines its output with explicit task evidence:

$$\begin{aligned} \widehat{b}_t &= \arg\max_{b \in \{0,1\}} p_{\theta_e}\left(b \mid \chi_t, o_t, h_t\right), \\ e_t^{\text{comp}} &= \mathbb{I}\left[\mathcal{V}_t\left(\chi_t, o_t, h_t\right) = 1\right], \\ d_t^{\text{term}} &= \widehat{b}_t e_t^{\text{comp}}. \end{aligned}$$

Here, $\mathcal{V}_t$ is the task verifier and $\mathbb{I}[\cdot]$ is the binary indicator.

Remark. ARCO realizes learned effects conservatively: routing limits candidate-generation cost, causal–operational valuation ranks admissible alternatives, and advantage-margin authorization retains the factual workflow when evidence is insufficient.

Success-Preserving Objective and Authorization Constraints

Let τ be a complete execution trajectory, $\pi_{\text{CHILL},\Theta}$ the policy induced by CIEL and ARCO, $\Theta = \{\phi, \theta_u, \theta_p, \theta_c, \theta_e\}$, and π_{ref} the matched reference harness. CHILL-Harness minimizes trajectory cost subject to bounded success degradation:

$$\begin{aligned} \min_{\Theta} \quad & \mathbb{E}_{\tau \sim \pi_{\text{CHILL},\Theta}}\left[\mathcal{J}_{\text{eff}}(\tau)\right] \\ \text{s.t.} \quad & \text{PassRate}\left(\pi_{\text{CHILL},\Theta}\right) \geq \text{PassRate}\left(\pi_{\text{ref}}\right) - \epsilon_{\text{pass}}, \end{aligned}$$

where $\mathcal{J}_{\text{eff}}(\tau)$ aggregates trajectory-level resource cost and $\epsilon_{\text{pass}} \geq 0$ is the allowed pass-rate degradation.

At each step, only admissible route-exposed workflows may be executed, and termination requires both learned and verified completion evidence:

$$\omega_t^\star \in \overline{\Omega}_t = \Omega_t \cap \mathcal{A}_t, \qquad d_t^{\text{term}} = \widehat{b}_t e_t^{\text{comp}}.$$

Because environment execution is not directly differentiable, the constrained objective is realized through outcome-conditioned weak supervision rather than trajectory-level backpropagation. Let $Y_\tau, Y_{\text{ref}} \in \{0, 1\}$ denote the current and reference trajectory outcomes, and let $\lambda_{\text{fail}}, \lambda_{\text{break}} \geq 0$ be their penalty weights. We define

$$\mathcal{L}_{\text{traj}}(\tau) = \mathcal{J}_{\text{eff}}(\tau) + \lambda_{\text{fail}}\left(1 - Y_\tau\right) + \lambda_{\text{break}} \mathbb{I}\left[Y_{\text{ref}} = 1 \wedge Y_\tau = 0\right]. \tag{10}$$

Equation (10) penalizes resource cost, task failure, and success-breaking adaptation, and constructs weak targets for CDE, CRE, and CAE.

Remark. The success-preserving objective and authorization constraints jointly protect task performance: efficiency-related decisions receive positive supervision only when success is preserved, while unsupported workflow revision and premature termination are disallowed.

Algorithm 1 summarizes the inference flow; Appendices C.1–C.4 provide the complete objectives and authorizations.

Experiments

We evaluate whether CHILL-Harness resolves the causal harness orchestration problem by learning beneficial workflow interventions, realizing them efficiently and admissibly, and preserving task success. We examine these requirements through three empirical questions:

- **RQ1: Effectiveness.** Does solving the Harness Intervention Effect Problem in Definition 1 identify adaptations that preserve or improve task success?
- **RQ2: Efficiency.** Does solving the Harness Intervention Realization Problem in Definition 2 reduce reasoning and execution cost without premature failure or termination?

Benchmark	Method	Success	Tokens	Token Gain vs. Min.-Token	Tokens/Solved	Rel. Runtime	Time Gain vs. Min.-Runtime
GAIA	Terminus-KIRA	70.2%	134M	–	2.03M	1.00×	–
	AWorld	69.7%	161M	–	2.45M	1.25×	–
	OWL Workforce	69.1%	174M	–	2.68M	1.30×	–
	CHILL-Harness	**71.3%**	**96M**	**28.4%↓**	**1.43M**	0.534×	**46.6%↓**
SWE-bench Verified	Terminus-KIRA	65.0%	157M	–	2.42M	1.00×	–
	OpenHands	65.4%	165M	–	2.52M	1.10×	–
	CodeSweep–SWE-agent	53.4%	153M	–	2.87M	0.92×	–
	CHILL-Harness	**65.6%**	**133M**	**13.1%↓**	**2.25M**	0.682×	**25.9%↓**
Terminal-Bench 2.0	Terminus-KIRA	49.4%	291M	–	7.11M	1.00×	–
	Meta-Harness	52.6%	340M	–	5.00M	1.25×	–
	LemonHarness	57.4%	380M	–	5.05M	1.40×	–
	CHILL-Harness	**56.1%**	**224M**	**23.1%↓**	**4.87M**	0.974×	**2.6%↓**

Table 1: Effectiveness and efficiency across three long-horizon benchmarks. Token Gain and Time Gain use the eligible non-CHILL baseline with the lowest token consumption and runtime, respectively. Relative runtime is normalized to Terminus-KIRA within each benchmark. Token totals are rounded to the nearest million, whereas Tokens/Solved is reported to two decimal places. CHILL-Harness token totals include all additional model calls for counterfactual deliberation and candidate valuation.

Algorithm 1 CHILL-Harness Overview (Full Procedure in Appendix C.4)

Input: $\chi_t, \omega_t^0, o_t, h_t$; **Output**: $\omega_t^\star, d_t^{\text{term}}$
$(\widehat{\iota}_t, \widetilde{z}_t) \leftarrow \text{CIEL}(\chi_t, \omega_t^0)$
Set z_t to the highest feasible route not exceeding $\widetilde{z}_t$
if $z_t = \textsc{Skip}$ **then**
  $\Omega_t \leftarrow \{\omega_t^0\}$
else
  Generate Ω_t from $\mathcal{G}_t(\widehat{\iota}_t) \cup \{\omega_t^0\}$
end if
$\overline{\Omega}_t \leftarrow \Omega_t \cap \mathcal{A}_t$; select the highest-valued candidate ω_t^+
Authorize $\omega_t^\star$ using CRE and candidate advantage
Authorize d_t^{term} using CAE and completion evidence
if $d_t^{\text{term}} = 1$ **then**
  Invoke the task-terminal action
else
  Execute $\omega_t^\star$
end if
Log execution evidence

- **RQ3: Generalization.** Do the effectiveness and efficiency benefits hold across heterogeneous execution environments?

We conduct baseline comparisons to verify that CHILL-Harness preserves effectiveness while improving efficiency across benchmarks, and perform an Always-Full ablation to establish the contribution of counterfactual orchestration.

Experimental Setup

Benchmarks. We evaluate CHILL-Harness on **GAIA** for information seeking and tool-assisted reasoning (Mialon et al., 2024), **SWE-bench Verified** for repository-level software repair (Jimenez et al., 2024; OpenAI, 2025), and **Terminal-Bench 2.0** for long-horizon terminal interaction (Merrill, Shaw, and Nicholas Carlini, 2026). These benchmarks cover deliberation, revision, stabilization, verification, and completion under distinct execution environments.

Baselines. Each CHILL-Harness evaluation is paired with a reference run using the same model, tools, environment, task, and resource limits. For external comparison, we report Terminus-KIRA on all benchmarks (KRAFTON AI and Ludo Robotics, 2026); AWorld and OWL Workforce on GAIA (Yu, Lu, and Chenyi Zhuang, 2025; Hu et al., 2025b); OpenHands and CodeSweep–SWE-agent on SWE-bench Verified (Wang et al., 2025a; Yang et al., 2024; Kimi Team, 2025); and Meta-Harness and LemonHarness on Terminal-Bench 2.0 (Lee et al., 2026; Ren et al., 2026).

Model Configuration. GAIA uses deepseek-v4-flash, whereas SWE-bench Verified and Terminal-Bench 2.0 use deepseek-v4-pro DeepSeek-AI (2026). CIEL is trained offline on disjoint paired traces and frozen for evaluation; checkpointed replay is disabled at test time. All model calls use temperature 1.0 and at most 150 interaction turns.

Evaluation Metrics. Let m denote a method, N the number of tasks, and $y_i^{(m)} \in \{0, 1\}$ whether m solves task i. Effectiveness is measured by $\text{Success}(m) = \frac{1}{N}\sum_{i=1}^{N} y_i^{(m)} \times 100\%$. Reasoning cost is measured by $\text{Tokens}(m) = \sum_{i=1}^{N}\sum_{k\in\mathcal{K}_m}\left(\text{Tok}_{i,k}^{\text{in}} + \text{Tok}_{i,k}^{\text{out}}\right)$, $\text{Tokens/Solved}(m) = \frac{\text{Tokens}(m)}{\sum_{i=1}^{N} y_i^{(m)}}$, where $\mathcal{K}_m$ includes all model-based harness components. Let $T_i^{(m)}$ be the end-to-end runtime of method m on task i, including inference, orchestration, tools, interaction, and verification:

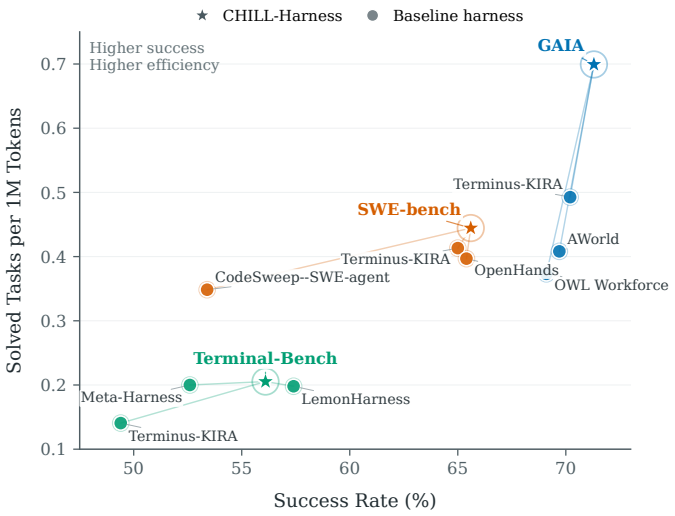

Figure 2: Joint effectiveness–efficiency comparison on GAIA, SWE-bench Verified, and Terminal-Bench 2.0. The horizontal axis reports task success and the vertical axis reports solved tasks per one million tokens; higher values on both axes are preferred. Stars denote CHILL-Harness, circles denote public baseline harnesses, and colors group methods evaluated on the same benchmark.

$\text{RelativeRuntime}(m; b_T) = \frac{\sum_{i=1}^{N} T_i^{(m)}}{\sum_{i=1}^{N} T_i^{(b_T)}}$. For $X \in \{\text{Tokens}, T\}$, resource gain is $\text{Gain}_X(m; b_X) = \left(1 - \frac{X(m)}{X(b_X)}\right) \times 100\%$. Here, b_T and b_X denote the runtime and resource reference baselines, respectively.

Main Results

Table 1 reports effectiveness and resource measurements against baselines, while Figure 2 summarizes their joint success–token efficiency. Controlled matched-reference comparisons assess success preservation, whereas comparisons with public systems assess external competitiveness.

RQ1: Effectiveness. Under matched model, tool, environment, task, and resource settings, CHILL-Harness preserves or improves task success on all three benchmarks. It achieves the highest success rate among the compared systems on GAIA (71.3%) and SWE-bench Verified (65.6%). On Terminal-Bench 2.0, its 56.1% success exceeds Terminus-KIRA and Meta-Harness and remains close to LemonHarness (57.4%). Thus, the efficiency gains do not arise from broadly sacrificing task effectiveness.

RQ2: Efficiency. CHILL-Harness achieves the highest number of solved tasks per one million tokens in each benchmark group. Against the lowest-token public baselines, it reduces token consumption by 28.4%, 13.1%, and 23.1% on GAIA, SWE-bench Verified, and Terminal-Bench 2.0, respectively. Against the fastest eligible baselines, it reduces runtime by 46.6%, 25.9%, and 2.6%.

RQ3: Generalization. CHILL-Harness improves both success and token efficiency over all displayed baselines on GAIA and SWE-bench Verified. On Terminal-Bench 2.0, it achieves the highest token efficiency, while LemonHarness retains a small success advantage, placing both systems on the effectiveness–efficiency frontier. This pattern holds across information seeking, software repair, and terminal interaction.

Remark. Controlled comparisons support success preservation, while public-system comparisons show that CHILL-Harness consistently improves token efficiency and remains competitive in task effectiveness across heterogeneous long-horizon environments.

Ablation Experiment

We evaluate the contribution of CDE and its route-before-generation realization in ARCO. The Always-Full variant disables adaptive deliberation by setting $z_t = \textbf{Full}$ whenever feasible, while retaining CIEL intervention prediction, CRE, and CAE, as well as ARCO candidate generation, admissibility filtering, valuation, and authorization. The comparison therefore isolates the effect of selecting deliberation depth before candidate generation.

Table 2 shows that forced full deliberation causes over-intervention on GAIA, increases cost while reducing success on SWE-bench Verified, and incurs substantial planner overhead with little benefit on Terminal-Bench 2.0. These results validate the contribution of CDE and its route-before-generation realization in ARCO.

Remark. CDE acts before candidate generation: CRE can reject unsupported workflow revisions only after the corresponding candidate-generation cost has been incurred, whereas route-before-generation suppresses low-value deliberation and avoids the associated planner overhead.

Conclusion

We introduced CHILL-Harness, a causal framework that learns whether and how the harness should intervene in the choice between factual and alternative workflows. The empirical results support affirmative answers to our three research questions: CIEL identifies interventions that preserve or improve task effectiveness, ARCO realizes them with lower reasoning and execution cost, and these benefits generalize across heterogeneous long-horizon environments. Experiments on information seeking, software repair, and terminal interaction, together with the Always-Full ablation, support both the overall effectiveness of CHILL-Harness and the necessity of selective counterfactual deliberation. We hope this work encourages the community to move beyond fixed harness engineering toward learned, causal, and success-preserving orchestration, providing a foundation for more efficient and reliable long-horizon agents.

Benchmark	Variant	Success	Tokens	Tokens/Solved	Full Calls / Revision Rate	Always-Full Overhead
GAIA	**CHILL-Harness**	**71.3%**	**96.00M**	**1.43M**	–	–
	Always-Full	27.7%	114.33M	4.40M	2,615 / 76.1%	18.33M tokens 37.73 h
SWE-bench Verified	**CHILL-Harness**	**65.6%**	**133.00M**	**2.25M**	–	–
	Always-Full	53.3%	160.40M	3.34M	3,708 / 3.5%	27.40M tokens 70.95 h
Terminal-Bench 2.0	**CHILL-Harness**	**56.1%**	**224.19M**	**4.87M**	–	–
	Always-Full	56.3%	259.77M	5.65M	4,234 / 3.9%	35.58M tokens 91.98 h

Table 2: Ablation Experiment Results. Full Calls / Revision Rate reports the number of full CDE events and the fraction of those events that authorize workflow revision. Always-Full Overhead reports the additional planner token consumption and cumulative planner latency incurred by Always-Full relative to CHILL-Harness on the same benchmark.

References

Ba, L.; Li, Q.; and Li, S. 2026. Ciber: A comprehensive benchmark for security evaluation of code interpreter agents. *arXiv:2602.19547*.

Chi, H.; Li, H.; Yang, W.; Liu, F.; Lan, L.; Ren, X.; Liu, T.; and Han, B. 2024. Unveiling Causal Reasoning in Large Language Models: Reality or Mirage? *NeurIPS*.

DeepSeek-AI. 2026. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence.

Fu, J.; Ding, L.; Wei, Q.; Guo, Y.; Cheng, Y.; and Zhang, J. 2026. Counterfactual Planning for Generalizable Agents' Actions. *AAAI*.

Guo, J.; Hao, Z.; Wang, C.; Fan, C.; Luo, T.; Li, H.; Gao, Y.; Mei, H.; Peng, J.; Xu, R.; et al. 2026. From Question Answering to Task Completion: A Survey on Agent System and Harness Design. *arXiv:2606.20683*.

He, Z.; Wang, Y.; Zhi, C.; Hu, Y.; Chen, T.-P.; Yin, L.; Chen, Z.; Wu, T. A.; Ouyang, S.; Wang, Z.; et al. 2026. MemoryArena: Benchmarking Agent Memory in Interdependent Multi-Session Agentic Tasks. *arXiv:2602.16313*.

Hu, M.; Chen, T.; Chen, Q.; Mu, Y.; Shao, W.; and Luo, P. 2025a. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. *ACL*.

Hu, M.; Zhou, Y.; Fan, W.; Nie, Y.; Xia, B.; Sun, T.; Ye, Z.; Jin, Z.; Li, Y.; Chen, Q.; Zhang, Z.; Wang, Y.; Ye, Q.; Ghanem, B.; Luo, P.; and Li, G. 2025b. OWL: Optimized Workforce Learning for General Multi-Agent Assistance in Real-World Task Automation. *NeurIPS*.

Hu, S.; Lu, C.; and Clune, J. 2025. Automated design of agentic systems. *ICLR*.

Hu, Y.; Wang, Y.; and McAuley, J. 2025. Evaluating memory in llm agents via incremental multi-turn interactions. *arXiv:2507.05257*.

Jimenez, C. E.; Yang, J.; Wettig, A.; Yao, S.; Pei, K.; Press, O.; and Narasimhan, K. R. 2024. SWE-bench: Can Language Models Resolve Real-world Github Issues? *ICLR*.

Kapoor, S.; Stroebl, B.; Kirgis, P.; Nadgir, N.; Siegel, Z. S.; Wei, B.; Xue, T.; Chen, Z.; Chen, F.; Utpala, S.; Ndzomga, F.; Oruganty, D.; Luskin, S.; Liu, K.; Yu, B.; Arora, A.; Hahm, D.; Trivedi, H.; Sun, H.; Lee, J.; Jin, T.; Mai, Y.; Zhou, Y.; Zhu, Y.; Bommasani, R.; Kang, D.; Song, D.; Henderson, P.; Su, Y.; Liang, P.; and Narayanan, A. 2026. Holistic Agent Leaderboard: The Missing Infrastructure for AI Agent Evaluation. *ICLR*.

Kimi Team. 2025. Kimi K2: Open Agentic Intelligence. *arXiv:2507.20534*.

KRAFTON AI; and Ludo Robotics. 2026. Terminus-KIRA: Boosting Frontier Model Performance on Terminal-Bench with Minimal Harness.

Kuroki, M.; and Pearl, J. 2014. Measurement bias and effect restoration in causal inference. *Biometrika*, 101: 423–437.

Lee, Y.; Nair, R.; Zhang, Q.; Lee, K.; Khattab, O.; and Finn, C. 2026. Meta-Harness: End-to-End Optimization of Model Harnesses. *arXiv:2603.28052*.

Li, J.; Xiao, X.; Zhang, Y.; Liu, C.; Zhao, L.; Liao, X.; Ji, Y.; Wang, J.; Ge, Y.; Xu, W.; Fang, X.; Xu, X.; Zhao, T.; Kim, Y.; Hamm, J.; Wang, T.; and Reddy, C. 2026. Agent Harness Engineering: A Survey.

Marchand, R.; Cathain, A. O.; Wynne, J.; Giavridis, P. M.; Deverett, S.; Wilkinson, J.; Gwartz, J.; and Coppock, H. 2026. Quantifying Frontier LLM Capabilities for Container Sandbox Escape. *ICML*.

Meng, Q.; Wang, Y.; Chen, L.; Wu, W.; Li, Y.; Jiang, W.; Wang, Q.; Lu, C.; Gao, Y.; Wu, Y.; and Hu, Y. 2026. Agent Harness for Large Language Model Agents: A Survey. *Preprints:0.20944/preprints202604.0428.v3*.

Merrill, M. A.; Shaw, A. G.; and Nicholas Carlini, e. 2026. Terminal-Bench: Benchmarking Agents on Hard, Realistic Tasks in Command Line Interfaces. *ICLR*.

Mialon, G.; Fourrier, C.; Swift, C.; Wolf, T.; LeCun, Y.; and Scialom, T. 2024. GAIA: a benchmark for General AI Assistants. *ICLR*.

OpenAI. 2025. Introducing SWE-bench Verified. OpenAI Research.

Pan, J.; Wang, X.; Neubig, G.; Jaitly, N.; Ji, H.; Suhr, A.; and Zhang, Y. 2025. Training software engineering agents and verifiers with swe-gym. *ICML*.

Pearl, J. 2009. *Causality*. Cambridge university press.

Pourkeshavarz, M.; Zhang, J.; and Rasouli, A. 2024. CaDeT: A Causal Disentanglement Approach for Robust Trajectory Prediction in Autonomous Driving. *CVPR*.

Ren, K.; Sun, F.; Liu, J.; Yang, L.; Yin, Z.; Li, J.; Yin, C.; He, M.; Huo, Y.; Liu, J.; Chen, Z.; Huangfu, Y.; Li, R.; Wu, Y.; Su, X.; Xu, Y.; Wu, L.; Zhao, H.; Zhang, L.; Geng, X.; and Fan, J. 2026. LemonHarness Technical Report. *arXiv:2606.24311*.

Richens, J.; and Everitt, T. 2024. Robust Agents Learn Causal World Models. *ICLR*.

Rubin, D. B. 1974. Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies. *Journal of Educational Psychology*, 66: 688–701.

Rubin, D. B. 1980. Randomization Analysis of Experimental Data: The Fisher Randomization Test Comment. *Journal of the American Statistical Association*, 75: 591.

Sun, S.; Song, H.; Huang, L.; Jiang, J.; Le, R.; Lv, Z.; Chen, Z.; Hu, Y.; Luo, W.; Zhao, W. X.; Song, Y.; Xu, H.; Zhang, T.; and Wen, J.-R. 2026. SWE-World: Building Software Engineering Agents in Docker-Free Environments. *arXiv:2602.03419*.

Tang, J.; Zhou, Z.; He, Z.; Zhang, J.; Zhang, K.; and Pu, J. 2026. CausalVAD: De-confounding End-to-End Autonomous Driving via Causal Intervention. *CVPR*.

Wang, X.; Li, B.; Song, Y.; Xu, F. F.; Tang, X.; Zhuge, M.; Pan, J.; Song, Y.; Li, B.; Singh, J.; Tran, H. H.; Li, F.; Ma, R.; Zheng, M.; Qian, B.; Shao, Y.; Muennighoff, N.; Zhang, Y.; Hui, B.; Lin, J.; Brennan, R.; Peng, H.; Ji, H.; and Neubig, G. 2025a. OpenHands: An Open Platform for AI Software Developers as Generalist Agents. *ICLR*.

Wang, X.; and Liu, B. 2025. Oscar: Operating system control via state-aware reasoning and re-planning.

Wang, Z. Z.; Mao, J.; Fried, D.; and Neubig, G. 2025b. Agent workflow memory. *ICML*.

Xie, T.; Zhang, D.; Chen, J.; Li, X.; Zhao, S.; Cao, R.; Hua, T. J.; Cheng, Z.; Shin, D.; Lei, F.; et al. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *NeurIPS*.

Yang, J.; Jimenez, C. E.; Wettig, A.; Lieret, K.; Yao, S.; Narasimhan, K. R.; and Press, O. 2024. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. *NeurIPS*.

Yao, S.; Shinn, N.; Razavi, P.; and Narasimhan, K. 2025. *tau*-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. *ICLR*.

Yu, C.; Lu, S.; and Chenyi Zhuang, e. 2025. AWorld: Orchestrating the Training Recipe for Agentic AI. *arXiv:2508.20404*.

Yu, X.; Wang, Z.; Yang, L.; Li, H.; Liu, A.; Xue, X.; Wang, J.; and Yang, M. 2025. Causal Sufficiency and Necessity Improves Chain-of-Thought Reasoning. *NeurIPS*.

Zhang, G.; Niu, L.; Fang, J.; Wang, K.; Bai, L.; and Wang, X. 2025. Multi-agent architecture search via agentic supernet. *ICML*.

Zhang, Y.; Wang, J.; Ge, Y.; Xu, W.; Hamm, J.; and Reddy, C. K. 2026. Stop comparing LLM agents without disclosing the harness. *arXiv:2605.23950*.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. 2024. Webarena: A realistic web environment for building autonomous agents. *ICLR*.

Appendix A: Problem Formalization and CIEL Foundations

This appendix provides the formal definitions, theoretical grounding, and training implementation of Causal Intervention Effect Learning (CIEL). Appendix A.1 formalizes the two harness intervention problems. Appendix A.2 specifies the structural connection between semantic interventions and executable workflows. Appendix A.3 derives an error-dependent bound for amortized intervention selection. Appendix A.4 establishes paired-replay identification and details the separated supervision of workflow effects and factorized decisions.

A.1 Formalization of the Harness Intervention Problems

At execution step t, let

$$\chi_t = \left(s_t, c_t^{\text{env}}, g_t\right)$$

denote the fixed execution context, where s_t is the execution state, c_t^{env} is the environment context, and g_t is the current task objective.

Let ω_t^0 denote the factual workflow proposed without additional harness adaptation, and let $\mathcal{A}_t$ denote the workflows admissible under the current task, environment, safety, and resource requirements, with

$$\omega_t^0 \in \mathcal{A}_t.$$

The executed workflow is represented by W_t, and its task- and resource-aware execution performance is represented by U_t.

For any $\omega \in \mathcal{A}_t$,

$$\text{do}(W_t = \omega)$$

denotes executing workflow ω while holding χ_t fixed. All potential outcomes are evaluated under the same continuation policy π^{cont}, evaluation horizon, resource-accounting rule, and task evaluator. $U_t(\omega)$ captures downstream task performance, whereas local cost, risk, and execution-quality signals are represented separately by $\widehat{U}_{\text{op}}$.

Definition A.1: Harness Intervention Effect Problem. For an admissible workflow

$$\omega \in \mathcal{A}_t \setminus \{\omega_t^0\},$$

its intervention effect relative to the factual workflow is

$$\Gamma_t(\omega) = \mathbb{E}\left[U_t \mid \text{do}(W_t = \omega), \chi_t\right] - \mathbb{E}\left[U_t \mid \text{do}(W_t = \omega_t^0), \chi_t\right]. \tag{A1}$$

A positive value indicates expected improvement relative to the factual workflow, whereas a negative value indicates expected degradation. The factual workflow is the zero-effect reference:

$$\Gamma_t(\omega_t^0) = 0.$$

The estimand therefore assigns causal value to concrete executable workflows rather than directly to abstract intervention labels.

Definition A.2: Harness Intervention Realization Problem. Let

$$\widehat{\Gamma}_{\phi,t}(\omega) := \widehat{\Gamma}_\phi\left(\omega; \chi_t, \omega_t^0\right)$$

denote the estimated workflow effect, and let $\overline{\Omega}_t \subseteq \mathcal{A}_t$ be the route-exposed admissible workflow set with $\omega_t^0 \in \overline{\Omega}_t$. The highest-valued exposed candidate is

$$\omega_t^+ = \arg\max_{\omega \in \overline{\Omega}_t} \widehat{U}_{\text{ARCO}}\left(\omega \mid \chi_t, z_t, \widehat{\Gamma}_{\phi,t}\right). \tag{A2}$$

The authorized workflow then satisfies

$$\omega_t^\star \in \left\{\omega_t^0, \omega_t^+\right\}. \tag{A3}$$

Candidate selection and execution authorization are distinct: selection identifies the strongest currently exposed candidate, whereas replacement additionally requires admissibility, learned revision support, and a sufficient estimated advantage. The resulting orchestration chain is workflow-effect estimation $\longrightarrow$ route-exposed valuation $\longrightarrow$ execution authorization.

A.2 Workflow-Grounded Intervention Induction

Let

$$I_t \in \mathcal{I}_t^{\text{wf}}$$

denote the semantic harness intervention selected at step t. The intervention families have the following meanings.

- KEEP preserves the factual workflow ω_t^0.
- DELIBERATE allocates additional reasoning without prescribing a particular workflow revision.
- REVISE(INSPECT) acquires execution evidence missing from the current context.
- REVISE(VERIFY) validates an intermediate state, tool result, or candidate answer.
- REVISE(VISUAL) invokes visual processing when image or multimodal evidence is required.
- STABILIZE(DEDUP) suppresses repeated or semantically redundant operations.
- STABILIZE(NOOPRECOVERY) redirects an execution that has produced an empty, ineffective, or stalled action.
- ANSWERSYNTHESIS redirects broad exploration toward evidence organization and answer construction without itself terminating execution.

These intervention families describe adaptation intents rather than unique workflows. The realization map

$$\mathcal{G}_t(\iota) = \mathcal{G}\left(\iota; \chi_t, \omega_t^0\right) \subseteq \mathcal{A}_t$$

instantiates intervention ι according to the current context, factual workflow, available tools, and task constraints. Unavailable intervention families are removed from the set considered at step t, so $\mathcal{G}_t(\iota) \neq \emptyset$ for every family entering a maximization.

Implementation correspondence. The intervention predictor produces a discrete intervention-family label. A workflow-construction component then converts the label into one or more structured workflow candidates. For KEEP, the constructed set contains only ω_t^0. For DELIBERATE, the harness allocates an additional reasoning step without forcing a tool or workflow change. For the three REVISE modes, candidate templates constrain the next workflow toward evidence inspection, result verification, or visual analysis. For the two STABILIZE modes, recent execution traces are checked for duplicate actions, repeated tool arguments, empty results, or stalled progress. ANSWERSYNTHESIS generates a nonterminal workflow that organizes available evidence into a candidate answer. Each generated workflow is parsed into the same internal representation used by the base harness before ARCO filtering and valuation.

Assumption A.1: Workflow Mediation. For fixed χ_t, each $\omega \in \mathcal{G}_t(\iota)$ is an executable realization of intervention ι, intervention I_t affects U_t only through the realized workflow W_t, and intervention consistency holds. This is a structural modeling restriction rather than an identification result: semantic labels index workflow families but are not assigned context-independent causal values.

Under this assumption,

$$\mathbb{E}\left[U_t \mid \mathrm{do}(I_t = \iota, W_t = \omega), \chi_t\right] = \mathbb{E}\left[U_t \mid \mathrm{do}(W_t = \omega), \chi_t\right]. \tag{A4}$$

Hence, the effect associated with intervention ι is evaluated through its concrete realization ω, relative to ω_t^0. The same intervention family may therefore induce different effects across contexts and workflow realizations.

A.3 Error-Bounded Amortized Intervention Selection

For fixed (χ_t, ω_t^0), define the true and estimated values of intervention family ι as

$$V_t^{\mathrm{int}}(\iota) = \max_{\omega \in \mathcal{G}_t(\iota)} \Gamma_t(\omega), \tag{A5}$$

$$\widehat{V}_t^{\mathrm{int}}(\iota) = \max_{\omega \in \mathcal{G}_t(\iota)} \widehat{\Gamma}_\phi\left(\omega; \chi_t, \omega_t^0\right). \tag{A6}$$

Let

$$C_t = \bigcup_{\iota \in \mathcal{I}_t^{\mathrm{wf}}} \mathcal{G}_t(\iota)$$

denote all workflows considered at step t, including the factual workflow, and define

$$\varepsilon_t = \sup_{\omega \in C_t} \left|\widehat{\Gamma}_\phi\left(\omega; \chi_t, \omega_t^0\right) - \Gamma_t(\omega)\right|.$$

Let

$$\iota_t^\dagger = \arg\max_{\iota \in \mathcal{I}_t^{\mathrm{wf}}} V_t^{\mathrm{int}}(\iota)$$

be a true best intervention family,

$$\iota_t^* = \arg\max_{\iota \in \mathcal{I}_t^{\mathrm{wf}}} \widehat{V}_t^{\mathrm{int}}(\iota)$$

be the offline estimated target, and define the amortization gap

$$\rho_t = \widehat{V}_t^{\mathrm{int}}(\iota_t^*) - \widehat{V}_t^{\mathrm{int}}(\widehat{\iota}_t) \geq 0.$$

Proposition A.1: Effect-and-Amortization Error Bound. For any estimated workflow-effect function and online intervention predictor,

$$V_t^{\mathrm{int}}(\iota_t^\dagger) - V_t^{\mathrm{int}}(\widehat{\iota}_t) \leq 2\varepsilon_t + \rho_t. \tag{A7}$$

Proof. For every intervention family ι, the maximum operator and the definition of ε_t imply

$$\left|\widehat{V}_t^{\mathrm{int}}(\iota) - V_t^{\mathrm{int}}(\iota)\right| \leq \varepsilon_t.$$

Therefore,

$$\begin{aligned} V_t^{\mathrm{int}}(\iota_t^\dagger) &\leq \widehat{V}_t^{\mathrm{int}}(\iota_t^\dagger) + \varepsilon_t \\ &\leq \widehat{V}_t^{\mathrm{int}}(\iota_t^*) + \varepsilon_t \\ &= \widehat{V}_t^{\mathrm{int}}(\widehat{\iota}_t) + \rho_t + \varepsilon_t \\ &\leq V_t^{\mathrm{int}}(\widehat{\iota}_t) + 2\varepsilon_t + \rho_t. \end{aligned}$$

Rearranging proves Equation (A7). □

Corollary A.1: Exact Intervention Recovery. Suppose that $\iota_t^\dagger$ is unique and define its true margin as

$$\Delta_t^{\mathrm{int}} = V_t^{\mathrm{int}}(\iota_t^\dagger) - \max_{\iota \neq \iota_t^\dagger} V_t^{\mathrm{int}}(\iota).$$

If

$$\Delta_t^{\mathrm{int}} > 2\varepsilon_t + \rho_t,$$

then

$$\widehat{\iota}_t = \iota_t^\dagger.$$

The result does not assume that estimated and true rankings are identical. Instead, it quantifies how workflow-effect estimation error and amortized prediction error jointly determine intervention-selection quality. When the online predictor reproduces the offline target, $\rho_t = 0$, and the selection regret is bounded by $2\varepsilon_t$.

A.4 CIEL Training and Separated Supervision

CIEL separates workflow-effect supervision from factorized decision supervision. The workflow-effect estimator is trained only from offline checkpointed paired executions that compare factual and alternative workflows from the same execution context. Retrospective trajectory evidence provides weak supervision for deliberation, revision, and completion decisions, but is not treated as an identified workflow-effect observation.

Checkpointed paired intervention records. For a selected training state, the harness checkpoints the execution context χ_t, including the agent context, recent execution memory, environment state, tool state, intermediate artifacts, and current task progress. The checkpoint is restored to execute the factual workflow ω_t^0 and an admissible counterfactual candidate ω as two isolated branches. Both branches use the same

continuation policy π^{cont}, evaluation horizon, model and tool configuration, task evaluator, and resource-accounting rule. Branch-specific planning, model inference, tool interaction, and execution costs are included in the resulting utility. Post-checkpoint randomness is either matched across branches or independently sampled from the same distribution and remains independent of workflow assignment.

The analysis set C_t in Appendix A.3 intentionally includes the factual workflow. For paired replay, define only the counterfactual subset

$$C_t^{\text{cf}} = C_t \setminus \{\omega_t^0\}. \tag{A8}$$

No replay count is required for ω_t^0, whose relative effect is fixed to zero. If $C_t^{\text{cf}} = \emptyset$, no paired-effect record is constructed at that state and KEEP remains the only available reference family.

For each $\omega \in C_t^{\text{cf}}$, let $K_{t,\omega} \geq 1$ be the number of valid paired replays, and let

$$U_t^{(k)}(\omega) \quad \text{and} \quad U_t^{(k)}(\omega_t^0)$$

denote the candidate and factual utilities in replay k. The paired workflow-effect target is

$$\widetilde{\Gamma}_t^{\text{pair}}(\omega) = \frac{1}{K_{t,\omega}} \sum_{k=1}^{K_{t,\omega}} \left[U_t^{(k)}(\omega) - U_t^{(k)}(\omega_t^0) \right]. \tag{A9}$$

The factual reference is fixed by construction:

$$\widetilde{\Gamma}_t^{\text{pair}}(\omega_t^0) = \widehat{\Gamma}_\phi\left(\omega_t^0; \chi_t, \omega_t^0\right) = 0.$$

Assumption A.2: State-Sufficient Paired Replay. The restored checkpoint contains the pre-intervention variables required to reproduce the execution context relevant to post-intervention performance. After restoration, the branches differ only in their assigned workflow, follow the same continuation and evaluation protocol, do not interfere, and use post-checkpoint randomness independent of workflow assignment. The candidate set, replay count, and replay-inclusion criteria are fixed before paired outcomes are observed; replay validity does not depend on the sign or magnitude of the observed utility difference.

Proposition A.2: Identification and Concentration by Paired Replay. Under Assumption A.2,

$$\mathbb{E}\left[\widetilde{\Gamma}_t^{\text{pair}}(\omega) \mid \chi_t\right] = \Gamma_t(\omega) \tag{A10}$$

for every $\omega \in C_t^{\text{cf}}$.

Suppose additionally that

$$U_t(\omega) \in [u_{\min}, u_{\max}], \qquad R = u_{\max} - u_{\min}.$$

For nonempty C_t^{cf}, define

$$M_t = \left|C_t^{\text{cf}}\right|, \qquad K_t^{\text{pair}} = \min_{\omega \in C_t^{\text{cf}}} K_{t,\omega}.$$

If paired replays are independent conditional on χ_t, then, with probability at least $1 - \zeta$,

$$\sup_{\omega \in C_t^{\text{cf}}} \left|\widetilde{\Gamma}_t^{\text{pair}}(\omega) - \Gamma_t(\omega)\right| \leq R\sqrt{\frac{2\log(2M_t/\zeta)}{K_t^{\text{pair}}}}. \tag{A11}$$

Proof. Because both branches begin from the same restored context and workflow assignment is independent of post-checkpoint randomness, each branch follows the potential-outcome distribution associated with its assigned workflow. Linearity of expectation yields Equation (A10). For each candidate, the paired difference lies in $[-R, R]$. Hoeffding's inequality bounds the empirical-mean deviation, and a union bound over the M_t counterfactual candidates yields Equation (A11). □

Connection to intervention-selection error. Define the effect-model fitting error over paired targets as

$$\alpha_t = \sup_{\omega \in C_t^{\text{cf}}} \left|\widehat{\Gamma}_\phi\left(\omega; \chi_t, \omega_t^0\right) - \widetilde{\Gamma}_t^{\text{pair}}(\omega)\right|.$$

Because the factual effect and its estimate are both zero, the error ε_t defined over the full set C_t equals the supremum over C_t^{cf} whenever this subset is nonempty. Therefore, with probability at least $1 - \zeta$,

$$\varepsilon_t \leq \alpha_t + R\sqrt{\frac{2\log(2M_t/\zeta)}{K_t^{\text{pair}}}}.$$

Combining this inequality with Proposition A.1 gives

$$V_t^{\text{int}}(\iota_t^\dagger) - V_t^{\text{int}}(\widehat{\iota}_t) \leq 2\alpha_t + 2R\sqrt{\frac{2\log(2M_t/\zeta)}{K_t^{\text{pair}}}} + \rho_t. \tag{A12}$$

Thus, intervention-selection quality is jointly controlled by workflow-effect fitting error, finite paired-replay error, and amortized prediction error.

Paired workflow-effect training set. The workflow-effect training set is

$$\mathcal{D}_{\text{pair}} = \left\{ \left(\chi_t, \omega_t^0, \omega, \widetilde{\Gamma}_t^{\text{pair}}(\omega), w_{t,\omega}\right) : \omega \in C_t^{\text{cf}} \right\},$$

where $w_{t,\omega} \geq 0$ is a reliability weight determined from pre-specified replay-validity diagnostics, replay count, and comparison stability, without using the sign of the observed effect. Pairs are excluded when the checkpoint cannot be restored, the branches do not share the same continuation protocol, or branch outcomes cannot be isolated. Only checkpointed paired executions are treated as workflow-effect observations. Unpaired trajectories, incomplete comparisons, and model-scored candidates do not directly supervise $\widehat{\Gamma}_\phi$ and are not interpreted as identified causal effects.

The workflow-effect estimator is trained by

$$\mathcal{L}_{\text{eff}} = \mathbb{E}_{\left(\chi_t, \omega_t^0, \omega, \widetilde{\Gamma}_t^{\text{pair}}(\omega), w_{t,\omega}\right) \sim \mathcal{D}_{\text{pair}}} \left[w_{t,\omega} \left(\widehat{\Gamma}_\phi\left(\omega; \chi_t, \omega_t^0\right) - \widetilde{\Gamma}_t^{\text{pair}}(\omega) \right)^2 \right]. \tag{A13}$$

Intervention target and objective. Candidate workflows are grouped by semantic intervention family. The offline target is induced by the learned workflow-effect estimator:

$$\iota_t^* = \arg\max_{\iota \in \mathcal{I}_t^{\text{wf}}} \max_{\omega \in \mathcal{G}_t(\iota)} \widehat{\Gamma}_\phi\left(\omega; \chi_t, \omega_t^0\right). \tag{A14}$$

When no alternative family has a positive estimated effect, conservative tie-breaking assigns

$$\iota_t^* = \textsc{Keep}.$$

The amortized intervention predictor is trained by

$$\mathcal{L}_{\text{int}} = -\log p_{\theta_u}\left(\iota_t^* \mid \chi_t, \omega_t^0\right). \tag{A15}$$

Trajectory evidence for factorized decisions. For each intervention step, the implementation extracts

$$\mathcal{E}_t^{\text{dec}} = \left(Y_\tau, Y_{\text{ref}}, E_t^{\text{err}}, C_t^{\text{main}}, C_t^{\text{plan}}, r_t, c_t, \Delta_t^{\text{cf}}, \mathcal{H}_t^{\text{exec}}\right),$$

where Y_τ and Y_{ref} are the current and reference task outcomes; E_t^{err} records execution failure; C_t^{main} and C_t^{plan} record main-agent and planner costs; r_t and c_t summarize route and candidate statistics; and $\mathcal{H}_t^{\text{exec}}$ contains progress, repetition, failure, and completion evidence. A decision-target operator produces

$$\Psi_{\text{dec}}\left(\mathcal{E}_t^{\text{dec}}\right) = \left(z_t^*, y_t^{\text{chg},*}, b_t^*\right).$$

These are outcome-conditioned weak decision labels, not observations of $\Gamma_t(\omega)$.

Counterfactual Deliberation Effect target. The target

$$z_t^* \in \{\textsc{Skip}, \textsc{Light}, \textsc{Full}\}$$

specifies the preferred amount of additional deliberation. Skip is assigned when the factual workflow is sufficient or additional planning produces no useful decision evidence. Light is assigned when restricted diagnostics or a small candidate set is sufficient. Full is assigned only when broader generation or evaluation materially improves intervention selection, recovers ineffective execution, or prevents a high-risk factual action. Among routes preserving decision quality and task success, the target selects the least costly route.

Counterfactual Revision Effect target. The target

$$y_t^{\text{chg},*} \in \{\textsc{Keep}, \textsc{Change}\}$$

specifies whether the selected candidate ω_t^+ should replace the factual workflow. Change requires evidence that the candidate improves the factual execution while preserving success and satisfying authorization conditions. When paired execution evidence is available, its observed workflow difference provides the primary comparison. Unpaired retrospective evidence may provide auxiliary decision supervision but does not enter $\mathcal{D}_{\text{pair}}$. Unsafe, invalid, task-incompatible, over-budget, duplicate, ineffective, or success-breaking candidates receive Keep supervision.

Completion Attribution Effect target. The target

$$b_t^* \in \{0, 1\}$$

indicates whether task completion is supported by evidence already available at step t. The label $b_t^* = 1$ requires retrospective evidence that the task requirements were already satisfied, such as a valid final answer, passing tests, a required artifact, or an environment-provided success state. Budget exhaustion, inactivity, repeated failure, predicted futility, or the absence of another action does not produce a positive completion target.

Head-specific objectives. The factorized decision heads are trained by

$$\mathcal{L}_{\text{CDE}} = -\log p_{\theta_p}\left(z_t^* \mid \chi_t, \omega_t^0, \widehat{\iota}_t\right), \tag{A16}$$

$$\mathcal{L}_{\text{CRE}} = -\log p_{\theta_c}\left(y_t^{\text{chg},*} \mid \chi_t, \omega_t^0, \omega_t^+, \widehat{\iota}_t\right), \tag{A17}$$

$$\mathcal{L}_{\text{CAE}} = -\log p_{\theta_e}\left(b_t^* \mid \chi_t, o_t, h_t\right). \tag{A18}$$

The complete CIEL objective is

$$\mathcal{L}_{\text{CIEL}} = \mathcal{L}_{\text{eff}} + \mathcal{L}_{\text{int}} + \mathcal{L}_{\text{CDE}} + \mathcal{L}_{\text{CRE}} + \mathcal{L}_{\text{CAE}}. \tag{A19}$$

The objectives use distinct sources of supervision: $\mathcal{L}_{\text{eff}}$ learns workflow effects from offline checkpointed paired executions; $\mathcal{L}_{\text{int}}$ amortizes the preference induced by the learned workflow effects; and $\mathcal{L}_{\text{CDE}}$, $\mathcal{L}_{\text{CRE}}$, and $\mathcal{L}_{\text{CAE}}$ learn route, revision, and completion decisions from success-preserving trajectory evidence. This separation prevents heuristic or model-scored trajectory signals from being interpreted as identified workflow-effect observations.

Appendix B: ARCO Realization and Authorization Details

This appendix explains how Advantage-Realizing Causal Orchestration (ARCO) implements CIEL decisions through route selection, candidate generation, causal–operational valuation, workflow authorization, and completion verification.

B.1 Route Feasibility and Candidate Construction

Let

$$\mathcal{Z} = \{\textsc{Skip}, \textsc{Light}, \textsc{Full}\}, \qquad \textsc{Skip} \prec \textsc{Light} \prec \textsc{Full}.$$

The ordering represents increasing deliberation depth, candidate-generation coverage, and execution cost. The provisional route is

$$\widetilde{z}_t = \arg\max_{z \in \mathcal{Z}} p_{\theta_p}\left(z \mid \chi_t, \omega_t^0, \widehat{\iota}_t\right). \tag{B1}$$

Let $C^{\text{plan}}(z)$ be the predicted planning cost of route z, B_t^{plan} the remaining planning budget, N_t^{full} the number of previous full-deliberation calls, B^{full} their maximum allowed number, $t_{\text{last}}^{\text{full}}$ the most recent full-deliberation step, κ the full-route cooldown, $\mathcal{T}(z)$ the tools or capabilities required by route z, and $\mathcal{T}_t^{\text{avail}}$ the currently available tools and capabilities. If no full route has previously been executed, set $t_{\text{last}}^{\text{full}} = -\infty$.

Define the common planning-and-tool feasibility indicator as

$$g_t(z) = \mathbb{I}\left[C^{\text{plan}}(z) \le B_t^{\text{plan}}\right] \mathbb{I}\left[\mathcal{T}(z) \subseteq \mathcal{T}_t^{\text{avail}}\right]. \tag{B2}$$

The route feasibility is

$$f_t(z) = \begin{cases} 1, & z = \textsc{Skip}, \\ g_t(z), & z = \textsc{Light}, \\ g_t(z)\,\mathbb{I}\left[\begin{array}{c} N_t^{\text{full}} < B^{\text{full}}, \\ t - t_{\text{last}}^{\text{full}} \geq \kappa \end{array}\right], & z = \textsc{Full}. \end{cases} \tag{B3}$$

The feasible route set is

$$\mathcal{F}_t = \{z \in \mathcal{Z} : f_t(z) = 1\}.$$

Because Skip is always feasible, $\mathcal{F}_t \neq \emptyset$. The executed route is

$$z_t = \max_{\prec} \{z \in \mathcal{F}_t : z \preceq \widetilde{z}_t\}. \tag{B4}$$

Thus, an infeasible prediction is downgraded to the highest feasible route and is never upgraded beyond the learned prediction.

The raw route-specific candidate set is

$$\Omega_t^{\text{raw}} = \begin{cases} \{\omega_t^0\}, & z_t = \textsc{Skip}, \\ \{\omega_t^0\} \cup \mathcal{G}_t^{\text{light}}(\widehat{\iota}_t), & z_t = \textsc{Light}, \\ \{\omega_t^0\} \cup \mathcal{G}_t(\widehat{\iota}_t), & z_t = \textsc{Full}. \end{cases} \tag{B5}$$

Here, $\mathcal{G}_t^{\text{light}}(\widehat{\iota}_t) \subseteq \mathcal{G}_t(\widehat{\iota}_t)$ is a restricted, low-cost realization set. Candidate exposure is bounded by

$$K_{\textsc{Skip}} = 1, \qquad K_{\textsc{Light}} < K_{\textsc{Full}}.$$

These route budgets are distinct from the paired-replay quantity K_t^{pair} in Appendix A.4.

Let $\widehat{S}_t(\omega \mid \chi_t)$ be the workflow-safety score and ξ_t its minimum accepted threshold. Before expensive valuation, the retained candidate set is

$$\Omega_t = \{\omega_t^0\} \cup \text{TopK}_{K_{z_t}-1}\left\{\omega \in \Omega_t^{\text{raw}} \setminus \{\omega_t^0\} : \widehat{S}_t(\omega \mid \chi_t) \geq \xi_t\right\}, \tag{B6}$$

where $\text{TopK}_0(\cdot) = \emptyset$. When more candidates pass the safety threshold than the route budget allows, TopK retains them according to the low-cost workflow-construction score produced before full valuation. The admissible route-exposed set is

$$\overline{\Omega}_t = \Omega_t \cap \mathcal{A}_t. \tag{B7}$$

Because $\omega_t^0 \in \mathcal{A}_t$, the filtered set is nonempty.

Implementation correspondence. The CDE head outputs scores for the three route classes. A deterministic feasibility stage checks route-specific planning budgets, full-route limits and cooldown, required tools, and environment capabilities. Under Skip, planner and candidate-generation calls are bypassed. Under Light, the implementation uses restricted diagnostics and a small candidate budget. Under Full, all configured route-compatible generation and evaluation operations are available. The implementation records the provisional route, executed route, downgrade reason, exposed candidates, prescreening decisions, and incurred planning cost.

B.2 Causal–Operational Valuation

For each $\omega \in \overline{\Omega}_t$, let $\widehat{P}_t(\omega)$, $\widehat{C}_t(\omega)$, $\widehat{R}_t(\omega)$, $\widehat{I}_t(\omega)$, $\widehat{B}_t(\omega)$, and $\widehat{S}_t(\omega \mid \chi_t)$ denote estimated task progress, execution cost, operational risk, information gain, robustness, and workflow safety, respectively. All available signals are calibrated to a common bounded range before aggregation. A signal unavailable for a benchmark is omitted rather than assigned an arbitrary value.

Define the shared progress–risk–safety score and the full-route information–robustness bonus as

$$\widehat{Q}_t(\omega) = \alpha_{\text{prog}}\widehat{P}_t(\omega) - \alpha_{\text{risk}}\widehat{R}_t(\omega) + \alpha_{\text{safe}}\widehat{S}_t(\omega \mid \chi_t), \tag{B8}$$

$$\widehat{H}_t(\omega) = \alpha_{\text{info}}\widehat{I}_t(\omega) + \alpha_{\text{rob}}\widehat{B}_t(\omega). \tag{B9}$$

The route-conditioned operational utility is

$$\widehat{U}_{\text{op}}(\omega \mid \chi_t, z_t) = \begin{cases} 0, & z_t = \textsc{Skip}, \\ \widehat{Q}_t(\omega) - \alpha_{\text{cost}}^{\text{light}}\widehat{C}_t(\omega), & z_t = \textsc{Light}, \\ \widehat{Q}_t(\omega) - \alpha_{\text{cost}}^{\text{full}}\widehat{C}_t(\omega) + \widehat{H}_t(\omega), & z_t = \textsc{Full}. \end{cases} \tag{B10}$$

All coefficients are nonnegative. The light route omits information-gain and robustness estimation when restricted diagnostics are sufficient, whereas the full route uses the complete configured valuation.

The causal–operational utility is

$$\widehat{U}_{\text{ARCO}}\left(\omega \mid \chi_t, z_t, \widehat{\Gamma}_{\phi,t}\right) = \widehat{U}_{\text{op}}(\omega \mid \chi_t, z_t) + \eta_\Gamma \widehat{\Gamma}_{\phi,t}(\omega), \tag{B11}$$

where $\eta_\Gamma \geq 0$ controls the contribution of estimated workflow effect evidence. The strongest exposed candidate is

$$\omega_t^+ = \arg\max_{\omega \in \overline{\Omega}_t} \widehat{U}_{\text{ARCO}}\left(\omega \mid \chi_t, z_t, \widehat{\Gamma}_{\phi,t}\right). \tag{B12}$$

Its advantage over the factual workflow is

$$\begin{aligned} \Delta_t^{\text{cf}} = \widehat{U}_{\text{op}}(\omega_t^+ \mid \chi_t, z_t) - \widehat{U}_{\text{op}}\left(\omega_t^0 \mid \chi_t, z_t\right) \\ + \eta_\Gamma\left[\widehat{\Gamma}_{\phi,t}(\omega_t^+) - \widehat{\Gamma}_{\phi,t}(\omega_t^0)\right]. \end{aligned} \tag{B13}$$

The estimator enforces

$$\widehat{\Gamma}_{\phi,t}(\omega_t^0) = 0.$$

Implementation correspondence. Task-progress signals may be derived from newly acquired evidence, intermediate environment states, test results, or predicted goal advancement. Cost signals use expected or observed model tokens, planner calls, tool calls, and latency. Risk signals capture invalid actions, destructive operations, execution errors, or uncertain state changes. Information-gain signals estimate whether the candidate obtains evidence absent from the factual path. Robustness signals favor candidates that depend less on unverified assumptions or fragile execution sequences. The final score ranks only candidates in the route-exposed admissible set $\overline{\Omega}_t$.

B.3 Conservative Workflow Authorization

The CRE decision for the selected candidate is

$$y_t^{\text{chg}} = \arg \max_{y \in \{\textsc{Keep},\textsc{Change}\}} p_{\theta_c}\left(y \mid \chi_t, \omega_t^0, \omega_t^+, \widehat{\iota}_t\right). \quad \text{(B14)}$$

Keep retains the factual workflow, whereas Change provides learned support for replacement by ω_t^+. CRE and causal–operational valuation have complementary roles: CRE determines whether replacement is contextually supported, while Δ_t^{cf} determines whether the estimated improvement is large enough to justify revision.

The authorization rule is

$$\omega_t^\star = \begin{cases} \omega_t^+, & y_t^{\text{chg}} = \textsc{Change} \ \wedge \ \Delta_t^{\text{cf}} \geq \delta_{\widehat{\iota}_t}, \\ \omega_t^0, & \text{otherwise}, \end{cases} \quad \text{(B15)}$$

where $\delta_{\widehat{\iota}_t} \geq 0$ is the authorization margin for the predicted intervention family. Because ω_t^+ is selected from $\overline{\Omega}_t$, admissibility, safety, schema, task-compatibility, and budget checks have already been satisfied before this rule is applied.

For implementation diagnostics, define

$$a_t^{\text{sup}} = \mathbb{I}\left[y_t^{\text{chg}} = \textsc{Change}\right] \mathbb{I}\left[\Delta_t^{\text{cf}} \geq \delta_{\widehat{\iota}_t}\right], a_t^{\text{feas}} = \mathbb{I}\left[\omega_t^+ \in \overline{\Omega}_t\right], \quad \text{(B16)}$$

and

$$a_t^{\text{chg}} = a_t^{\text{sup}} a_t^{\text{feas}}.$$

The learned CRE output and final authorization result are logged separately, allowing prediction errors to be distinguished from constraint-triggered fallback. If either support or feasibility fails, ARCO retains ω_t^0.

Scope of workflow optimality. ARCO does not claim global optimization over the complete admissible workflow space $\mathcal{A}_t$. Instead,

$$\omega_t^+ = \arg \max_{\omega \in \overline{\Omega}_t} \widehat{U}_{\text{ARCO}}\left(\omega \mid \chi_t, z_t, \widehat{\Gamma}_{\phi,t}\right)$$

is optimal only within the route-exposed admissible set. CHILL-Harness executes this candidate when authorized and otherwise retains the factual workflow.

B.4 Evidence-Grounded Completion and Answer Synthesis

The CAE prediction is

$$\widehat{b}_t = \arg \max_{b \in \{0,1\}} p_{\theta_e}\left(b \mid \chi_t, o_t, h_t\right). \quad \text{(B17)}$$

The value $\widehat{b}_t = 1$ indicates learned completion support, whereas $\widehat{b}_t = 0$ indicates that completion is unsupported by the CAE head.

An explicit task verifier produces

$$e_t^{\text{comp}} = \mathbb{I}\left[\mathcal{V}_t\left(\chi_t, o_t, h_t\right) = 1\right], \quad \text{(B18)}$$

where $\mathcal{V}_t$ is the task-specific completion verifier. Termination is authorized by

$$d_t^{\text{term}} = \widehat{b}_t e_t^{\text{comp}}. \quad \text{(B19)}$$

Thus, termination requires both learned completion support and explicit task evidence.

The verifier may decompose completion evidence into strong, weak, and contradictory signals:

$$e_t^{\text{comp}} = \mathbb{I}\left[E_t^{\text{strong}} = 1 \ \vee \ \left(E_t^{\text{weak}} = 1 \ \wedge \ E_t^{\text{fail}} = 0\right)\right].$$

Strong evidence includes passing tests, successful builds, accepted solutions, or explicit environment success. Weak evidence includes a valid final answer or task-completion signal, whereas traceback, assertion, permission, and execution failures constitute contradictory evidence.

Implementation correspondence. The CAE head does not directly invoke the terminal action; its output is passed to the task-specific verifier. Depending on the benchmark, the verifier checks a required answer format, successful tests or evaluator results, the existence and validity of a required artifact, an environment-provided success state, or another explicit completion condition. Predicted futility, budget exhaustion, repeated failure, or lack of progress does not set $e_t^{\text{comp}} = 1$. Such signals may instead redirect execution toward AnswerSynthesis or another workflow adaptation. Therefore,

$$\textsc{AnswerSynthesis} \neq \textsc{Complete}.$$

The former is a nonterminal adaptation; the latter is an authorized terminal decision. The implementation records the CAE output, verifier result, supporting evidence, synthesis decision, and final terminal action.

Appendix C: Success-Preserving Learning and Complete Inference

This appendix specifies the trajectory-level objective, the separated supervision scheme, local authorization constraints, offline paired-effect collection, and the complete CHILL-Harness inference procedure.

C.1 Success-Preserving Trajectory Objective

The learnable CIEL parameter set is

$$\Theta = \left\{\phi, \theta_u, \theta_p, \theta_c, \theta_e\right\}.$$

ARCO introduces no separate prediction parameters in the current formulation. It realizes CIEL outputs through candidate generation, valuation, filtering, and rule-based authorization with calibrated weights and thresholds.

Let $\mathcal{K}$ denote the resource channels measured for a benchmark. The trajectory-level execution cost is

$$\mathcal{J}_{\text{eff}}(\tau) = \sum_{k \in \mathcal{K}} \lambda_k C_k(\tau), \quad \text{(C1)}$$

where $C_k(\tau)$ is the amount of resource k consumed by trajectory τ, and $\lambda_k \geq 0$ is its weight. For the reported experiments, the principal resource channels are

$$C_{\text{tok}}(\tau) = C_{\text{main}}^{\text{in}}(\tau) + C_{\text{main}}^{\text{out}}(\tau) + C_{\text{plan}}^{\text{in}}(\tau) + C_{\text{plan}}^{\text{out}}(\tau),$$

and

$$C_{\text{time}}(\tau) = T_{\text{main}}(\tau) + T_{\text{plan}}(\tau) + T_{\text{tool}}(\tau) + T_{\text{env}}(\tau) + T_{\text{verify}}(\tau). \tag{C2}$$

The reported total token count includes main-agent and counterfactual-planner calls made during benchmark inference. Offline paired-replay collection is a training-data construction procedure and is not included in test-time token or runtime measurements; its cost is tracked separately.

The global objective is

$$\begin{aligned} \min_{\Theta} \quad & \mathbb{E}_{\tau \sim \pi_{\text{CHILL},\Theta}} \left[\mathcal{J}_{\text{eff}}(\tau) \right] \\ \text{s.t.} \quad & \text{PassRate}\left(\pi_{\text{CHILL},\Theta}\right) \geq \text{PassRate}\left(\pi_{\text{ref}}\right) - \epsilon_{\text{pass}}. \end{aligned} \tag{C3}$$

Here, $\pi_{\text{CHILL},\Theta}$ is the execution policy induced by CIEL and ARCO; π_{ref} is the matched reference harness; $\text{PassRate}(\pi)$ is the fraction of evaluated tasks completed successfully by policy π; and $\epsilon_{\text{pass}} \geq 0$ is the allowed aggregate pass-rate degradation. Success is determined by the benchmark evaluator rather than by the harness completion prediction.

C.2 Trajectory Supervision and Decision-Target Construction

Let

$$Y_\tau \in \{0, 1\} \quad \text{and} \quad Y_{\text{ref}} \in \{0, 1\}$$

indicate whether the CHILL-Harness and matched reference trajectories succeed, respectively. The trajectory supervision score is

$$\mathcal{L}_{\text{traj}}(\tau) = \mathcal{J}_{\text{eff}}(\tau) + \lambda_{\text{fail}} (1 - Y_\tau) + \lambda_{\text{break}} \mathbb{I}\left[Y_{\text{ref}} = 1 \ \wedge \ Y_\tau = 0\right]. \tag{C4}$$

The first term measures execution cost, the second penalizes task failure, and the third penalizes adaptations that break a successful reference trajectory. The coefficients $\lambda_{\text{fail}}, \lambda_{\text{break}} \geq 0$ control the two failure penalties.

Separated supervision correspondence. The trajectory score is not differentiated through external tools or the environment. Instead, it determines valid samples, confidence weights, and weak decision targets:

$$\mathcal{L}_{\text{traj}} \longrightarrow \left(z_t^*, y_t^{\text{chg},*}, b_t^*\right).$$

Workflow-effect supervision is constructed separately:

$$\mathcal{D}_{\text{pair}} \longrightarrow \widetilde{\Gamma}_t^{\text{pair}}(\omega) \longrightarrow \widehat{\Gamma}_\phi \longrightarrow \iota_t^*.$$

Thus, retrospective trajectory evidence does not directly define or supervise the identified workflow-effect target. CDE targets select the least costly route preserving decision quality; CRE targets determine whether the selected candidate should replace the factual workflow; and CAE targets determine whether completion is already supported by evidence.

C.3 Local Authorization Constraints

Using the unified safety score from Appendix B, the admissible workflow set is

$$\begin{aligned} \mathcal{A}_t = \{\omega : & \widehat{S}_t\left(\omega \mid \chi_t\right) \geq \xi_t \\ & \wedge \ \text{SchemaValid}(\omega) \\ & \wedge \ \text{TaskCompatible}\left(\omega; \chi_t\right) \\ & \wedge \ \text{WithinBudget}\left(\omega; \chi_t\right) \\ & \wedge \ \neg \, \text{Repeated}_t(\omega)\}. \end{aligned} \tag{C5}$$

Here, $\widehat{S}_t(\omega \mid \chi_t)$ is the workflow-safety score and ξ_t its minimum threshold; SchemaValid checks workflow and tool-call structure; TaskCompatible checks that proposed operations are permitted in the current task and environment; WithinBudget checks remaining model, planning, tool, revision, and execution budgets; and Repeated_t identifies duplicate or repeatedly ineffective workflows.

Each generated candidate is parsed and passed through these checks before valuation and authorization. Rejected candidates are removed from

$$\overline{\Omega}_t = \Omega_t \cap \mathcal{A}_t,$$

and their rejection reasons are stored in the execution trace. Revision satisfies

$$\omega_t^\star = \omega_t^+ \Longrightarrow \left[y_t^{\text{chg}} = \textsc{Change} \ \wedge \ \Delta_t^{\text{cf}} \geq \delta_{\widehat{\iota}_t} \right],$$

and completion satisfies

$$d_t^{\text{term}} = 1 \Longrightarrow \left[\widehat{b}_t = 1 \ \wedge \ e_t^{\text{comp}} = 1 \right].$$

Together, these constraints ensure

$$\omega_t^\star \in \overline{\Omega}_t \subseteq \mathcal{A}_t$$

and prevent unsupported revision or premature termination.

C.4 Complete Inference, Offline Pairing, and Logging

Algorithm 2 Complete CHILL-Harness Inference

Input: $\chi_t, \omega_t^0, o_t, h_t$; **Output**: $\omega_t^\star, d_t^{\text{term}}$

Predict

$$\widehat{\iota}_t = \arg\max_{\iota \in \mathcal{I}_t^{\text{wf}}} p_{\theta_u}\left(\iota \mid \chi_t, \omega_t^0\right)$$

Predict the provisional route

$$\widetilde{z}_t = \arg\max_{z \in \mathcal{Z}} p_{\theta_p}\left(z \mid \chi_t, \omega_t^0, \widehat{\iota}_t\right)$$

and select the highest feasible route $z_t \preceq \widetilde{z}_t$

Generate the raw route-conditioned candidate set Ω_t^{raw} using Equation (B5)

Prescreen candidates using Equation (B6) and obtain Ω_t

Filter admissible candidates:

$$\overline{\Omega}_t = \Omega_t \cap \mathcal{A}_t$$

For each $\omega \in \overline{\Omega}_t$, estimate

$$\widehat{\Gamma}_{\phi,t}(\omega) = \widehat{\Gamma}_\phi\left(\omega; \chi_t, \omega_t^0\right)$$

and compute $\widehat{U}_{\text{ARCO}}(\omega \mid \chi_t, z_t, \widehat{\Gamma}_{\phi,t})$

Select ω_t^+ and compute Δ_t^{cf} using Equations (B12) and (B13)

Predict CRE support:

$$y_t^{\text{chg}} = \arg\max_{y \in \{\textsc{Keep},\textsc{Change}\}} p_{\theta_c}\left(y \mid \chi_t, \omega_t^0, \omega_t^+, \widehat{\iota}_t\right)$$

Authorize $\omega_t^\star$ using Equation (B15)

Compute

$$\widehat{b}_t = \arg\max_{b \in \{0,1\}} p_{\theta_e}\left(b \mid \chi_t, o_t, h_t\right)$$

and

$$d_t^{\text{term}} = \widehat{b}_t \mathbb{I}\left[\mathcal{V}_t\left(\chi_t, o_t, h_t\right) = 1\right]$$

if $d_t^{\text{term}} = 1$ **then**

  Invoke the task-terminal action

else

  Execute $\omega_t^\star$

end if

Log decisions, candidate evaluations, rejection reasons, execution costs, observations, errors, and task outcomes

The inference record contains at least

$$\left(\chi_t, \omega_t^0, \widehat{\iota}_t, \widetilde{z}_t, z_t, \Omega_t^{\text{raw}}, \Omega_t, \overline{\Omega}_t, \omega_t^+, \Delta_t^{\text{cf}}, y_t^{\text{chg}}, \omega_t^\star, \widehat{b}_t, e_t^{\text{comp}}, d_t^{\text{term}}\right), \tag{C6}$$

together with tool calls, observations, rejection reasons, costs, errors, and final task outcomes.

Offline paired-effect collection. Checkpointed paired replay is performed only during offline training-data construction. For each selected training checkpoint and counterfactual candidate, the implementation restores the same checkpoint, executes the factual and candidate branches under the common continuation and evaluation protocol, computes Equation (A9), and adds the resulting record to $\mathcal{D}_{\text{pair}}$. CIEL parameters are then trained on the designated training split and frozen before benchmark evaluation. No paired replay, effect-model update, or decision-head update is performed on evaluation tasks.

The complete learning and execution flow is offline paired execution $\longrightarrow$ $\widetilde{\Gamma}_t^{\text{pair}}$ $\longrightarrow$ $\widehat{\Gamma}_\phi$ $\longrightarrow$ ι_t^*, trajectory evidence $\longrightarrow$ $\left(z_t^*, y_t^{\text{chg},*}, b_t^*\right)$, frozen CIEL predictions $\longrightarrow$ ARCO realization $\longrightarrow$ authorized execution.