WHERE TO REPAIR? WORLD-MODEL-GUIDED HIERARCHICAL INTERVENTION FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective self-correction in LLM agents requires anticipating potential failures and identifying the appropriate scope of intervention. We propose HiCARE, a framework for hierarchical counterfactual attribution and repair before execution. Our framework organizes the agent loop into hierarchical decision levels and uses a world model to simulate the outcomes of proposed actions. When a simulated outcome indicates failure, the world model evaluates counterfactual interventions at different levels to attribute the predicted failure and assess how alternative decisions could resolve it. An intervention controller balances the expected improvement against the cost of modifying each level, selecting targeted repairs within a bounded deliberation budget. Revised proposals are reassessed through world-model simulation before the final action is executed in the real environment. This process connects predictive failure attribution with counterfactual intervention, enabling anticipatory self-correction across decision levels. Experiments on DiscoveryWorld's Combinatorial Chemistry task and τ-Knowledge show that our framework achieves better task performance than the evaluated baselines.

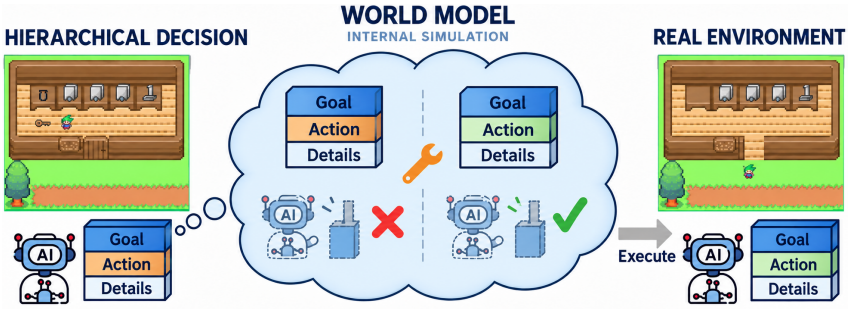


Figure 1: **Motivation of HiCARE.** A world model allows an agent to simulate a hierarchical decision, repair a predicted failure at the appropriate scope, and execute only the revised decision in the real environment.

1 INTRODUCTION

Recognizing a likely failure is not the same as knowing where to repair. Large language model (LLM) agents connect reasoning to external actions, allowing them to gather information and revise plans through interaction (Yao et al., 2022). Yet an action is the endpoint of several dependent decisions: what to achieve, which operation to use, and how to execute it. This nested structure has been exploited for task decomposition (Prasad et al., 2024; Hu et al., 2025a), but not for deciding how much of a decision to revisit. When a proposed action is unlikely to succeed, correcting its arguments may be sufficient—or entirely irrelevant. Reliable self-correction therefore requires deciding not only whether to intervene, but also how much of the decision to reconsider. Consider the following hypothetical scenario:

"A laboratory agent is preparing an experiment. If it is about to measure the wrong sample, changing the target may suffice. If the selected instrument cannot measure the desired property, the mea-

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.

surement operation must change. If the experiment cannot distinguish the competing hypotheses, the experimental design itself must be reconsidered..."

These situations require repairs at different scopes: changing a sample cannot rescue an uninformative design, while redesigning an experiment is unnecessary when only the target is wrong. The challenge is to determine how much of a decision to reconsider before committing an action.

Existing work provides important ingredients, but each leaves the repair-scope question unaddressed. Self-Refine (Madaan et al., 2023) and Reflexion (Shinn et al., 2023) improve outputs through self-generated feedback and stored reflections, yet both revise retrospectively without distinguishing which part of a decision is at fault. World-model-based approaches make deliberation anticipatory—RAP (Hao et al., 2023) searches simulated reasoning paths and WALL-E 2.0 (Zhou et al., 2026) refines actions via learned symbolic dynamics—but evaluate alternatives at a single, fixed level of abstraction. CoEx (Kim & Hwang, 2025) couples hierarchical planning with a neurosymbolic belief state, yet its hierarchy organizes task decomposition rather than intervention scope. Closest to our setting, CausalFlow (Bonagiri et al., 2026) performs step-level counterfactual interventions on failed traces but differs in two decisive respects: its counterfactuals are constructed *after* a trace has failed, and its intervention granularity is fixed at the step level rather than chosen among scopes. In contrast, we aim to select the scope of a counterfactual repair *before* execution, on a decision not yet committed. These advances establish the value of feedback, predictive models, and targeted correction, but also motivate a complementary control question.

Given a pending structured decision, which intervention scope offers the greatest predicted improvement relative to the commitments it disrupts? As illustrated in Fig. 1, we therefore treat repair scope as an explicit decision variable and use a world model to counterfactually evaluate scope-specific repairs under a shared criterion. Broader commitments are reopened only when their additional expected benefit justifies the greater disruption. This perspective turns failure prediction into a prospective decision about where and how broadly to intervene before execution.

We propose **HiCARE**–**Hi**erarchical **C**ounterfactual **A**ttribution and **R**epair before **E**xecution–a framework that uses a world model to diagnose and repair structured decisions before they affect the environment. HiCARE organizes each decision proposal into ordered repair scopes. The world model predicts the local outcomes of the proposal and its admissible counterfactual repairs, while a controller selects the scope with the largest predicted utility gain relative to its disruption cost. A repair preserves coarser commitments, regenerates dependent finer components, and is reassessed before the agent commits an action. Observed outcomes then close two alignment loops: prediction errors update the relative trust assigned to the world model; validated recurring repairs are distilled into structural constraints that prevent similar proposals without updating policy parameters.

Our contributions are threefold: (i) We introduce HiCARE, which formulates pre-execution self-correction as cost-sensitive intervention over hierarchical decision scopes, enabling targeted repairs that preserve unaffected commitments. (ii) We develop a counterfactual controller that uses an error-adaptive world model to compare admissible repairs and selects an intervention by trading predicted utility gain against disruption cost, without environment rollback. (iii) We use execution feedback to jointly align world-model trust and future proposals, and establish a conditional guarantee connecting bounded prediction error to improvement in true local expected utility.

2 Related Work

Self-Improving Agents and Experience Reuse. Recent work improves agents by optimizing different persistent artifacts. *Harness-level* methods rewrite the agent scaffold: STOP (Zelikman et al., 2023) uses an LM improver to rewrite its own scaffold, while the Darwin Gödel Machine (Zhang et al., 2026a) and Hyperagents (Zhang et al., 2026b) extend self-modification to the agent codebase and its improvement mechanism. *Workflow-level* methods search over agent structure: ADAS (Hu et al., 2025b), AgentSquare (Shang et al., 2025), and AFlow (Zhang et al., 2025) search over agent code, modules, or workflow graphs, and Meta^n (Kim et al., 2026) grows the depth of a solver–improver stack until convergence. *Memory-level* methods accumulate reusable experience: AWM (Wang et al., 2024) induces workflows, ACE (Zhang et al., 2026c) evolves context into a structured playbook, and Recuris (Yu et al., 2026) localizes and validates patches to memory components from execution traces. *Exploration-level* methods improve how agents gather experience: Dream-

RSI (Zheng et al., 2026) turns discovery histories into replay simulators for improving exploration policies. *HiCARE instead takes a lighter-weight approach. It distills outcome-validated repairs into reusable constraints on future proposals, improving subsequent decisions without rewriting the harness, restructuring the workflow, or updating planner parameters.*

World-Model Planning and Counterfactual Intervention. *Model-based language agents* use simulated outcomes or search to improve deliberation, as in RAP (Hao et al., 2023) and LATS (Zhou et al., 2023). K-Search (Cao et al., 2026) further updates an intrinsic world model from execution feedback to guide high-level program search. *Hierarchical methods* instead organize task execution or memory around subgoals, including ADaPT (Prasad et al., 2024) and HiAgent (Hu et al., 2025a). Their hierarchies decompose a task *downward* into finer subgoals. Our hierarchy is used in the opposite direction: given a pending decision, it selects which level to revise *backward*, treating repair granularity rather than task decomposition as the control variable. WALL-E 2.0 (Zhou et al., 2026) refines actions using learned symbolic dynamics, and CoEx (Kim & Hwang, 2025) co-evolves planning with a neurosymbolic belief state, but both evaluate alternatives at a single, fixed level of abstraction. StepFinder (Zhu et al., 2026) identifies root-cause steps from failed multi-agent traces through temporal semantic modeling, but provides post-hoc step localization rather than pre-execution counterfactual repair. CausalFlow (Bonagiri et al., 2026) is the closest counterfactual analogue, yet it differs in two respects that make our contribution non-incremental. First, its counterfactuals are constructed *after* a trace has already failed, whereas ours are constructed *before* execution on a decision that has not yet been committed. Second, its intervention granularity is fixed at the step level, whereas ours is chosen among scopes on a shared predicted-utility scale. *Therefore, HiCARE instead makes repair granularity an explicit control decision, preserving unaffected commitments while escalating from local edits to broader replanning only when warranted.*

3 Problem Setup and Decision Hierarchies

We consider an agent interacting with a partially observable environment (Spaan, 2012). At step t, the agent receives an observation o_t and maintains a *public state* s_t constructed only from its interaction history $h_t = (o_0, a_0, \ldots, o_t)$ and memory. A policy π_θ (an LLM planner) proposes an initial structured decision $z_t^{(0)} \sim \pi_\theta(\cdot \mid s_t)$. After optional repairs, the agent commits a decision z_t of the same form and extracts an executable action:

$$z_t = \big(x_t^{(1)}, \ldots, x_t^{(D)}\big), \qquad a_t = \mathrm{Exec}(z_t). \tag{1}$$

An episode terminates at step T upon success or exhaustion of the environment budget and induces a trajectory $\xi = (s_0, z_0, a_0, \ldots, s_T)$. Let Π denote the complete closed-loop agent, including the planner and the repair controller. The environment evaluates and assigns a terminal success indicator $Y_{\mathrm{succ}}(\xi) \in \{0, 1\}$, which equals one exactly when the task's goal conditions are satisfied. The primary objective is the task success probability

$$J_{\mathrm{task}}(\Pi) = \mathbb{E}_{\xi \sim P(\cdot \mid \Pi, \mathrm{env})}[Y_{\mathrm{succ}}(\xi)] = \Pr(Y_{\mathrm{succ}} = 1). \tag{2}$$

Executing a_t produces the next observation and a realized one-step *outcome*,

$$(o_{t+1}, \boldsymbol{y}_t^*) \sim P_{\mathrm{env}}(\cdot \mid h_t, a_t), \qquad \boldsymbol{y}_t^* \in \{0, 1\}^d, \tag{3}$$

where $\boldsymbol{y}_t^*$ is a low-dimensional vector of predefined feedback indicators such as action feasibility, constraint violations, and task progress. Its conditional randomness captures hidden environment state and stochastic dynamics. We use the predicted utility $U(s, z)$ defined in Sec. 4 from these local outcomes as a one-step surrogate for the terminal objective J_{task}. These local outcomes provide diagnostic signals for screening decisions. However, they signal *whether* a decision is likely to fail, not *how much* of it should be revised.

To close this gap, we therefore represent an agent's decision as a hierarchy of components, where each level defines an intervention scope, which is a component that can be revised while preserving higher-level commitments.

Definition 1 (Decision hierarchy)**.** *A decision hierarchy is a tuple $\mathcal{H} = (\mathcal{L}, \prec, c, \Delta)$ where $\mathcal{L} = \{\ell_1, \ldots, \ell_D\}$ is a finite set of D scopes (the depth D is task-dependent); $\prec$ is a total coarseness order $\ell_1 \prec \cdots \prec \ell_D$ from finest to coarsest; $c : \mathcal{L} \to \mathbb{R}_{>0}$ is a strictly increasing scope cost,*

$$c_{\ell_1} < c_{\ell_2} < \cdots < c_{\ell_D}; \tag{4}$$

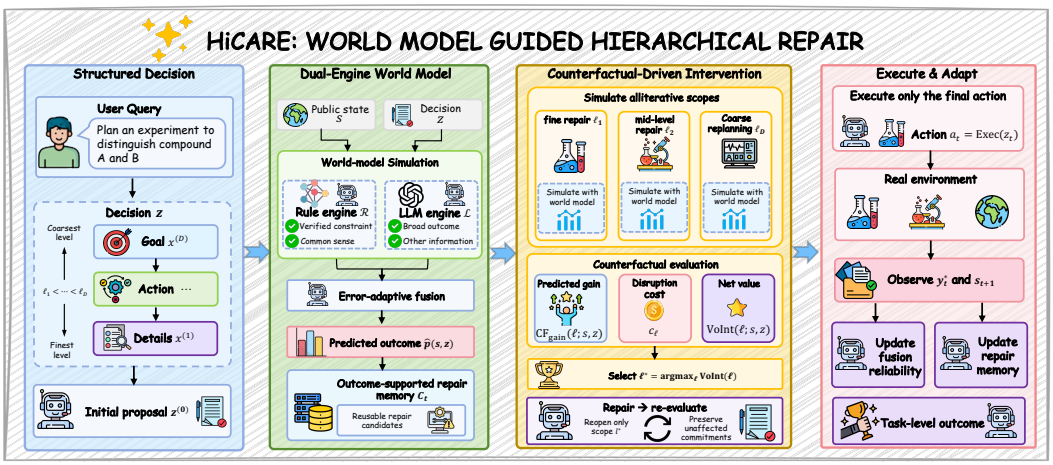


Figure 2: **Overview of HiCARE.**

$\Delta_\ell(s, z)$ is a finite set of admissible perturbations at scope ℓ, containing the null perturbation $\emptyset$.

The hierarchy factorizes the structured decision $z = (x^{(1)}, \ldots, x^{(D)})$ of Eq. (1), where scope ℓ_k governs component $x^{(k)}$, and *coarser components condition finer ones*—the feasible set of $x^{(k)}$ depends on $x^{(k+1)}, \ldots, x^{(D)}$. We write $z[\ell_k \leftarrow \delta]$ for the decision obtained by resetting $x^{(k)}$ according to $\delta \in \Delta_{\ell_k}(s, z)$ and re-deriving the finer components $x^{(k-1)}, \ldots, x^{(1)}$ consistently; its executed action is $\mathrm{Exec}(z[\ell_k \leftarrow \delta])$. The null edit leaves the decision unchanged, $z[\ell \leftarrow \emptyset] = z$. Because a coarser repair forces re-selection of everything below it, the cost ordering Eq. (4) reflects disruptiveness rather than an arbitrary weighting.

4 HiCARE: World-Model-Guided Hierarchical Repair Before Agent Execution

Building on the decision hierarchy defined before, we propose HiCARE, with complete framework shown in Fig. 2, which evaluates scope-specific counterfactual repairs with an adaptive world model and refines each pending decision before execution.

4.1 World Model with Error-Adaptive Fusion

World-model inference. For a hypothetical decision z with $a = \mathrm{Exec}(z)$, the environment produces a binary outcome $\boldsymbol{y}_t^* \in \{0, 1\}^d$ only after execution. For each coordinate j, let $p_j^*(s, z)$ denote the true, environment-induced probability that outcome j occurs when decision z is executed from public state s:

$$p_j^*(s, z) := \mathbb{E}[y_{j,t}^* \mid s_t = s, z_t = z] = \Pr(y_{j,t}^* = 1 \mid s_t = s, z_t = z). \tag{5}$$

This probability marginalizes over any hidden state and environmental stochasticity and is not available to the agent before execution. The agent observes only a single realization $y_{j,t}^*$ afterward. Let $\boldsymbol{p}^*(s, z)$ collect these true probabilities. Before execution, world model W estimates them from the public state and the proposed decision:

$$\hat{\boldsymbol{p}}(s, z) = W(s, z) \in [0, 1]^d, \tag{6}$$

where $\hat{p}_j(s, z)$ estimates $p_j^*(s, z)$. Thus, W predicts local outcome probabilities rather than a binary realization or a complete future trajectory. We suppress the environment-step index except when describing online updates.

Dual-engine prediction. The world model W is instantiated with a state-grounded rule engine R and a LLM engine L. Two engines provide complementary predictive coverage. The rule engine captures verifiable environment constraints, while the LLM engine generalizes to outcomes beyond the scope of explicit rules. For each outcome coordinate $j \in [d] := \{1, \ldots, d\}$ and engine

$e \in \{R, L\}$, $\hat{p}_{e,j}(s,z) \in [0,1]$ denotes the engine's probability estimate, while $m_{e,j}(s,z) \in \{0,1\}$ indicates whether that estimate is available. We assume that at least one engine is available for every evaluated coordinate (e.g., $\sum_{e\in\{R,L\}} m_{e,j}(s,z) \geq 1$). A hard-rule indicator $h_j(s,z) \in \{0,1\}$, satisfying $h_j(s,z) \leq m_{R,j}(s,z)$, specifies whether a verified rule deterministically determines coordinate j. At environment step t, the fused prediction is

$$\hat{p}_j(s,z) = \begin{cases} \hat{p}_{R,j}(s,z), & h_j(s,z) = 1, \\ \sum_{e\in\{R,L\}} \omega_{e,j,t}(s,z)\hat{p}_{e,j}(s,z), & h_j(s,z) = 0, \end{cases} \tag{7}$$

where $\omega_{e,j,t}(s,z)$ is the error-adaptive weight defined below, with unavailable engines assigned zero weight. Thus, verified hard rules override the mixture, while all other predictions are fused according to the engines' observed reliability.

Error-adaptive fusion. The two engines provide complementary predictions, but their empirical reliability may differ across outcome coordinates and can only be estimated from execution feedback. A fixed mixture may therefore overweight the less accurate engine. We instead adapt their coordinate-wise fusion weights according to recent prediction errors. We initialize the loss estimates as $\bar{L}_{e,j,0} = L_0$ and obtain the initial fusion weights $\omega_{e,j,0}$ from the common loss value. After executing the committed decision z_t and observing the realized binary outcome $\boldsymbol{y}_t^*$, we update each engine's coordinate-wise Brier loss using an exponential moving average (Aydore et al., 2019). Let $\hat{p}_{e,j,t} := \hat{p}_{e,j}(s_t, z_t)$. Then

$$\bar{L}_{e,j,t+1} = \begin{cases} (1-\alpha)\bar{L}_{e,j,t} + \alpha\big(\hat{p}_{e,j,t} - y^*_{j,t}\big)^2, & m_{e,j}(s_t,z_t) = 1, \\ \bar{L}_{e,j,t}, & m_{e,j}(s_t,z_t) = 0, \end{cases} \qquad 0 < \alpha \leq 1. \tag{8}$$

The updated losses determine the fusion weights for the next environment step:

$$\omega_{e,j,t+1}(s,z) = \frac{m_{e,j}(s,z)\exp(-\beta\bar{L}_{e,j,t+1})}{\sum_{e'\in\{R,L\}} m_{e',j}(s,z)\exp(-\beta\bar{L}_{e',j,t+1})}, \qquad \beta \geq 0. \tag{9}$$

Thus, engines with lower recent prediction error receive greater weight, while unavailable engines receive zero weight. This mechanism adapts the relative trust assigned to the two engines but does not by itself guarantee probabilistic calibration.

Counterfactual evaluation. Given the fused world model, we next determine whether and at which scope a pending decision should be repaired before execution. For a counterfactual edit $z' = z[\ell \leftarrow \delta]$, the world model W re-evaluates all applicable engines and returns the fused probability prediction $\hat{\boldsymbol{p}}(s,z')$ according to Eq. (7). These evaluations occur entirely within the world model and only the final committed decision is executed in the environment.

4.2 Counterfactual-Driven Attribution and Control

Given a pending decision z at public state s, a controller evaluates which intervention scope can best repair its predicted outcome before execution. Let $\boldsymbol{w} \in \mathbb{R}^d$ be a fixed utility weighting specified from the semantics of the observable outcome coordinates, shared across tasks in the same family. Desirable coordinates receive positive weights and undesirable ones negative weights, with $\|\boldsymbol{w}\|_1 = 1$. The predicted local utility is

$$U(s,z) = \langle \boldsymbol{w}, \hat{\boldsymbol{p}}(s,z) \rangle. \tag{10}$$

For each scope $\ell \in \mathcal{L}$, the controller selects the admissible counterfactual edit with the largest predicted improvement:

$$\delta^\star_\ell(s,z) \in \arg\max_{\delta\in\Delta_\ell(s,z)} \{U(s, z[\ell \leftarrow \delta]) - U(s,z)\}. \tag{11}$$

The corresponding counterfactual gain is

$$\mathrm{CFgain}(\ell; s, z) := U(s, z[\ell \leftarrow \delta^\star_\ell(s,z)]) - U(s,z) \geq 0. \tag{12}$$

The inequality follows because the null edit belongs to $\Delta_\ell(s, z)$. Since all scopes are evaluated using the same world model and utility function, their gains are directly comparable. This defines a repairability-based attribution. It identifies the scope at which an admissible edit is predicted to help most, rather than claiming to recover the causal source of failure.

Let $c_\ell := c(\ell)$ denote the disruption cost of scope ℓ, normalized to the same scale as U. We define the value of intervening (VoInt) as

$$\mathrm{VoInt}(\ell; s, z) := \mathrm{CFgain}(\ell; s, z) - c_\ell. \tag{13}$$

The controller commits z unchanged if $U(s, z) \geq \tau$ or $\max_{\ell \in \mathcal{L}} \mathrm{VoInt}(\ell; s, z) < \eta$, where $\eta > 0$ is the intervention margin. Otherwise, it selects

$$\ell^\star \in \arg\max_{\ell \in \mathcal{L}} \mathrm{VoInt}(\ell; s, z), \qquad z' := z[\ell^\star \leftarrow \delta^\star_{\ell^\star}(s, z)], \tag{14}$$

with ties broken in favor of the finer scope. Thus, every applied repair has positive predicted net value, while a coarser scope is chosen only when its additional gain offsets its greater disruption. The revised decision is then re-evaluated within the self-correcting loop in Sec. 4.4.

4.3 Bidirectional Adaptation and Outcome-Supported Repair Memory

Execution feedback supports two forms of adaptation. *Forward reliability adaptation* updates $\bar{L}_{e,j,t}$ and $\omega_{e,j,t+1}$ using Eqs. (8)–(9), increasing the influence of engines with lower recent prediction error. *Backward repair memory* builds on the broader idea that language agents can improve by retaining and reusing experience-derived knowledge (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2024). Let $\psi_t = \phi(s_t, z_t^{(0)})$ denote an abstract pattern of the initial proposal, and let $r_t = \rho(z_t^{(0)} \to z_t)$ abstract its repair sequence while removing instance-specific identities.

For each pattern–repair pair (ψ, r), let $N_{\psi,r}$ count its applications and $S_{\psi,r}$ count those satisfying $\langle \boldsymbol{w}, \boldsymbol{y}_t^* \rangle \geq \tau_{\mathrm{val}}$. The committed repair memory is

$$\mathcal{C}_t := \{(\psi, r) : N_{\psi,r} \geq n_{\mathrm{commit}},\ S_{\psi,r}/N_{\psi,r} \geq q_{\mathrm{commit}}\}. \tag{15}$$

For a matching future proposal, these templates augment the admissible repairs:

$$\Delta_\ell^{\mathcal{C}_t}(s, z) := \Delta_\ell(s, z) \cup \{\mathrm{Inst}(r; s, z) : (\phi(s, z), r) \in \mathcal{C}_t,\ \mathrm{scope}(r) = \ell\}. \tag{16}$$

Memory-derived repairs are still evaluated by the world model and applied only when their VoInt justifies intervention. Thus, the memory provides outcome-supported reuse rather than causal identification or permanent exclusion of proposals.

4.4 The Self-Correcting Loop

At environment step t, the planner proposes $z_t^{(0)} \sim \pi_\theta(\cdot \mid s_t)$. The world-model statistics and repair memory $\mathcal{C}_t$ remain fixed during the inner loop, where candidate edits are drawn from $\Delta_\ell^{\mathcal{C}_t}(s_t, z)$ and evaluated using the current fusion weights.

At round k, the controller stops if

$$U(s_t, z_t^{(k)}) \geq \tau, \quad \text{or} \quad \max_{\ell \in \mathcal{L}} \mathrm{VoInt}(\ell; s_t, z_t^{(k)}) < \eta, \quad \text{or} \quad k = K. \tag{17}$$

Otherwise, it applies Eq. (14) and re-evaluates the repaired proposal. Let $\kappa_t \leq K$ be the first stopping round. The agent then commits

$$z_t = z_t^{(\kappa_t)}, \qquad a_t = \mathrm{Exec}(z_t), \qquad (o_{t+1}, \boldsymbol{y}_t^*) \sim P_{\mathrm{env}}(\cdot \mid h_t, a_t). \tag{18}$$

Thus, counterfactual repairs remain internal to the world model, and only the final action is executed.

The observed outcome updates the fusion statistics through Eqs. (8)–(9). If a repair was applied, its abstract pair $(\phi(s_t, z_t^{(0)}), \rho(z_t^{(0)} \to z_t))$ updates the repair memory through Eq. (15). Planner parameters remain fixed and the adaptation occurs only through the fusion statistics and repair memory. Algorithm. 1 in Appendix. A summarizes the procedure. We further establish a reliable and near-optimal intervention guarantee. When the intervention margin sufficiently exceeds local world-model error, HiCARE selects repairs with positive true net benefit that remain close to the best admissible alternative. Please see Appendix. B for details.

5 EXPERIMENTS

We evaluate the task performance, interaction efficiency, and world-model sensitivity of HiCARE through four questions. **Q1** (Effectiveness): How does HiCARE compare with existing agent baselines in task success and score in interactive scientific discovery? **Q2** (Generality): Does HiCARE also improve task completion in knowledge-intensive workflows with varying document and action requirements? **Q3** (Efficiency): How effectively does HiCARE use limited environment interaction budgets to achieve higher task scores? **Q4** (Sensitivity): Does HiCARE retain its performance advantage when the LLM used for world-model prediction is replaced?

5.1 EXPERIMENTAL SETUP

Datasets. We evaluate HiCARE on the *Combinatorial Chemistry* task in DiscoveryWorld (Jansen et al., 2024) under the *Normal* setting and a 21-task stratified subset of τ-Knowledge (Shi et al., 2026), covering interactive scientific discovery and document-grounded tool use, respectively. The τ-Knowledge subset contains seven tasks in each of three difficulty groups, defined by the sum of required document and action counts: **Low** (≤ 14), **Medium** (15–25), and **High** (> 25).

Baselines. We compare HiCARE with six agent baselines: Random, Hypothesizer (Jansen et al., 2024), ReAct(Yao et al., 2022), Plan-and-Execute (Wang et al., 2023), StepFinder (Zhu et al., 2026), and LATS (Zhou et al., 2023). All six baselines are evaluated on the same setting.

5.2 EFFECTIVENESS

To address **Q1**, we compare HiCARE with six agent baselines on the *Combinatorial Chemistry* task in DiscoveryWorld under the *Normal* setting across 5 seeds, reporting task success rate, average task score, and average environment interaction steps. We make three observations: **Obs. ❶** HiCARE achieves more consistent task completion than all evaluated baselines. In table 1, it succeeds in all evaluated runs, whereas Plan-and-Execute and Hypothesizer, the strongest baselines by success rate, succeed in fewer than half. **Obs. ❷** High task scores do not necessarily indicate successful completion. LATS achieves the highest baseline average score of 0.92 but succeeds in only 20% of runs, below both Plan-and-Execute and Hypothesizer. **Obs. ❸** Higher completion rates are accompanied by shorter environment trajectories. HiCARE uses fewer environment interaction steps than every evaluated baseline, averaging 34.4% fewer than ReAct, the baseline with the fewest steps.

Table 1: **Performance on Combinatorial Chemistry (Normal) in DiscoveryWorld.** Best results are highlighted in **bold**.

Method	Success Rate	Avg. Score	Avg. Steps
Random	0%	0.07	200
ReAct	0%	0.82	126.8
P + E	40%	0.78	128.8
Hypothesizer	40%	0.9	171.4
StepFinder	20%	0.8	177.8
LATS	20%	0.92	165
HiCARE	**100%**	**1**	**83.2**

> **Insight: The benefit of HiCARE is reflected in complete execution rather than partial progress alone.** The joint gains in task success and interaction length support hierarchical pre-execution intervention as a promising way to improve execution reliability without relying on longer environment trajectories.

5.3 GENERALITY

To address **Q2**, we compare HiCARE with six agent baselines on the 21-task stratified τ-Knowledge subset, evaluating task completion, document and action recall, and interaction overhead across three difficulty groups. We make three observations: **Obs. ❶** The completion advantage extends across knowledge-intensive workflows of varying difficulty. In table 2, HiCARE leads in all three difficulty groups, with an overall pass rate of 71.4% versus 28.6% for StepFinder, the strongest baseline by overall pass rate. **Obs. ❷** Completion gains are accompanied by stronger action recall, but not uniformly higher document recall. HiCARE leads in action recall across all groups, reaching 85.9% overall. Hypothesizer achieves slightly higher overall document recall but completes substantially fewer tasks. **Obs. ❸** Higher completion rates do not uniformly coincide with fewer interactions or

Table 2: **Performance on the stratified τ-Knowledge subset.** We report pass rate, document and action recall, average turns and tool calls, and error counts. Low, Medium, and High each contain seven tasks and are defined by the sum of required document and action counts. The highest pass rate and recall values within each group and overall are highlighted in **bold**.

Method	Pass	Doc Recall	Action Recall	Avg. Turns	Avg. Tools	Errors
Difficulty: Low						
ReAct	42.90%	63.30%	46.00%	19	9.6	0
Plan-and-Execute	28.60%	**76.40%**	46.00%	18.6	13	0
Random	0%	0%	0%	92	90	112
Hypothesizer	28.60%	72.40%	62.70%	43.9	24	2
LATS	28.60%	63.30%	50.80%	34.6	17.1	0
StepFinder	57.10%	60.50%	76.20%	18.1	8.6	0
HiCARE	**85.70%**	63.30%	**84.10%**	21.3	11.6	0
Difficulty: Medium						
ReAct	14.30%	**72.00%**	66.40%	33.4	24.6	7
Plan-and-Execute	0%	67.10%	61.30%	28.3	18.3	5
Random	0%	3.00%	0%	92	90	112
Hypothesizer	14.30%	69.70%	74.90%	36.3	24.6	7
LATS	14.30%	68.20%	65.90%	27	20.7	3
StepFinder	14.30%	64.20%	67.30%	29.3	20.7	6
HiCARE	**71.40%**	70.70%	**83.60%**	33	25.1	5
Difficulty: High						
ReAct	0%	65.20%	62.00%	30.1	21.7	3
Plan-and-Execute	14.30%	70.50%	73.10%	35.7	23.6	2
Random	0%	1.10%	0%	92	90	112
Hypothesizer	0%	72.60%	58.00%	41.6	33.3	17
LATS	14.30%	65.20%	56.10%	28.6	22.4	2
StepFinder	14.30%	65.60%	66.70%	33.1	21.4	6
HiCARE	**57.10%**	**76.90%**	**90.10%**	49.4	35.1	6
Difficulty: Overall						
ReAct	19.00%	66.80%	58.10%	27.5	18.6	10
Plan-and-Execute	14.30%	71.40%	60.10%	27.5	18.3	7
Random	0%	1.40%	0%	92	90	336
Hypothesizer	14.30%	**71.50%**	65.20%	40.6	27.3	26
LATS	19.00%	65.60%	57.60%	30	20.1	5
StepFinder	28.60%	63.40%	70.10%	26.9	16.9	12
HiCARE	**71.40%**	70.30%	**85.90%**	34.6	24	11

errors. In the overall results, HiCARE uses more turns and tool calls than StepFinder and records more errors than LATS, despite outperforming both in task completion.

> **Insight: Reliable knowledge-intensive execution requires more than document coverage.** The cross-domain results support the applicability of HiCARE to complete workflows, where retrieved evidence must be used to carry out the required actions. They motivate aligning repair with the full task objective rather than maximizing retrieval coverage or minimizing interaction in isolation.

5.4 Efficiency

To address **Q3**, we evaluate HiCARE and six agent baselines on the *Combinatorial Chemistry* task in DiscoveryWorld under the *Normal* setting, reporting average task scores with environment interaction limits of 100, 200, 300, 400, and 500 steps. We make three observations: **Obs. ❶** HiCARE

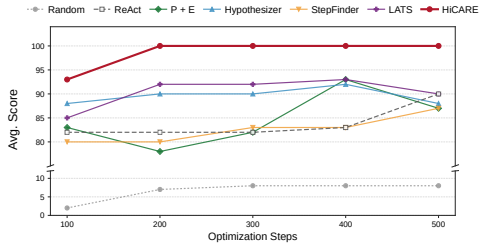


Figure 3: **Task scores under different interaction budgets.** Average task scores on *Combinatorial Chemistry* (*Normal*) in DiscoveryWorld with environment interaction limits of 100–500 steps. Scores are multiplied by 100 for display.

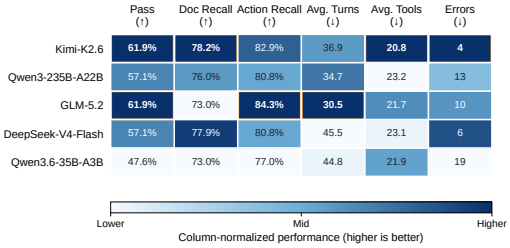


Figure 4: **Sensitivity to world-model choice.** Overall results of HiCARE with five alternative world models on the τ-Knowledge. Cell annotations report the original values. Colors are normalized within each column.

retains its score advantage under both restrictive and larger interaction budgets. In fig. 3, it achieves the highest average task score at every evaluated limit, including the most restrictive 100-step setting. **Obs. ❷** HiCARE matches the best observed baseline performance with a smaller allowed interaction budget. It attains an average score of 0.93 with a 100-step limit, matching Plan-and-Execute and LATS at 400 steps. These are budget limits, not executed step counts. **Obs. ❸** Larger interaction budgets do not consistently improve the observed baseline scores. Baseline performance plateaus or fluctuates, while HiCARE reaches an average score of 1.00 at 200 steps and maintains it at all larger evaluated limits.

> **Insight: Expanding the interaction horizon alone does not close the performance gap.** The results motivate improving the decisions committed to the environment rather than relying only on additional execution opportunities. The observed advantage concerns environment interaction efficiency, not total model computation.

5.5 Sensitivity

To address **Q4**, we evaluate HiCARE with five alternative LLMs for world-model prediction on the 21-task stratified τ-Knowledge subset, examining whether its performance advantage persists after model replacement. We make three observations: **Obs. ❶** The task-completion advantage persists across all evaluated LLM replacements. In fig. 4, all five configurations outperform the strongest reported baseline in table 2, with pass rates ranging from 47.6% to 61.9% compared with 28.6%. **Obs. ❷** The retained advantage extends to both document and action coverage. Every replacement configuration exceeds the highest baseline document recall and action recall reported in table 2. The benefit is therefore evident in both task completion and coverage of the required evidence and actions. **Obs. ❸** Strong task completion remains attainable under different interaction profiles. Kimi-K2.6 and GLM-5.2 share the highest pass rate, despite differing in interaction behavior. GLM-5.2 uses fewer turns, while Kimi-K2.6 uses fewer tool calls and records fewer errors.

> **Insight: The effectiveness of HiCARE is not confined to a single world-model LLM.** The consistent advantages over the reported baselines support the applicability of hierarchical pre-execution repair across the evaluated prediction models, even though their absolute performance and interaction profiles differ.

6 Conclusion

We introduced HiCARE, a training-free framework that treats self-correction as cost-sensitive intervention over a hierarchy of decision scopes. Before execution, HiCARE uses a fused world model to evaluate scope-specific counterfactual repairs and selects the intervention scope that best balances predicted improvement against disruption cost. Execution feedback then adapts predictor reliability and records reusable repair patterns. Experiments demonstrate the potential of hierarchical pre-execution repair to improve task completion while reducing unnecessary environment interaction.

AI USE STATEMENT

We used generative AI tools to assist in the preparation and presentation of this work. In addition to the language models used as components of the methodology and experiments described in the paper, we used generative AI tools for language editing, including grammar correction and improvements to clarity, readability, and organization. We also used these tools to assist in writing and debugging code for processing and analyzing experimental outputs, aggregating evaluation metrics, generating figures, and formatting result tables. The research questions, methodology, experimental design, interpretation of the results, and conclusions were determined by us. All AI-assisted text and code were manually reviewed, and the reported results were verified against the original experimental outputs. We take full responsibility for the content of this paper.

REFERENCES

Sergul Aydore, Tianhao Zhu, and Dean P Foster. Dynamic local regret for non-convex online forecasting. *Advances in neural information processing systems*, 32, 2019.

Akash Bonagiri, Devang Borkar, Gerard Janno Anderias, Setareh Rafatirad, and Houman Homayoun. Causalflow: Causal attribution and counterfactual repair for llm agent failures. *arXiv preprint arXiv:2605.25338*, 2026.

Shiyi Cao, Ziming Mao, Joseph E Gonzalez, and Ion Stoica. K-search: Llm kernel generation via co-evolving intrinsic world model. *arXiv preprint arXiv:2602.19128*, 2026.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 8154–8173, 2023.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32779–32798, 2025a.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025b.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *Advances in neural information processing systems*, 37:10088–10116, 2024.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002.

Minsoo Kim and Seung-won Hwang. Coex-co-evolving world-model and exploration. In *EMNLP (Findings)*, pp. 21629–21651, 2025.

Zae Myung Kim, Young-Jun Lee, Seungyeon Jwa, and Dongyeop Kang. Meta^n: Recursive self-improvement through emergent depth. *arXiv preprint arXiv:2608.24735*, 2026.

Sam Lobel and Ronald Parr. An optimal tightness bound for the simulation lemma. *arXiv preprint arXiv:2406.16249*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. Adapt: As-needed decomposition and planning with language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4226–4252, 2024.

Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. Agentsquare: Automatic llm agent search in modular design space. In *International Conference on Learning Representations*, volume 2025, pp. 3841–3865, 2025.

Quan Shi, Alexandra Zytek, Pedram Razavi, Karthik Narasimhan, and Victor Barres. τ-knowledge: Evaluating conversational agents over unstructured knowledge. *arXiv preprint arXiv:2603.04370*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Matthijs TJ Spaan. Partially observable markov decision processes. In *Reinforcement learning: State-of-the-art*, pp. 387–414. Springer, 2012.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 2609–2634, 2023.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

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

Zhaochen Yu, Yingcheng Wu, Zhenfei Yin, Kaiyuan Chen, Zhe Zhao, Mengdi Wang, Shuicheng Yan, and Ling Yang. Recursive experiential-working memory evolution for long-horizon agent harnesses. *arXiv preprint arXiv:2608.24876*, 2026.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-taught optimizer (stop): Recursively self-improving code generation. *arXiv preprint arXiv:2310.02304*, 2023.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, volume 2026, pp. 104223–104294, 2026a.

Jenny Zhang, Bingchen Zhao, Wannan Yang, Jakob Foerster, Jeff Clune, Minqi Jiang, Sam Devlin, and Tatiana Shavrina. Hyperagents. *arXiv preprint arXiv:2603.19461*, 2026b.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026c.

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

Tong Zheng, Xidong Wu, Zheng Zhang, Zhankui He, Chaoyi Zhang, Benjamin Coleman, Ruoqiao Wei, Di Bai, Haolin Liu, Rui Liu, et al. Dream-rsi: Recursive self-improvement through evolving worlds. *arXiv preprint arXiv:2609.14858*, 2026.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.

Siyu Zhou, Tianyi Zhou, Yijun Yang, Guodong Long, Deheng Ye, Jing Jiang, and Chengqi Zhang. Wall-e: World alignment by neurosymbolic learning improves world model-based llm agents. *Advances in Neural Information Processing Systems*, 38:65352–65391, 2026.

Taiyu Zhu, Yifan Wu, Weilin Jin, Ying Li, and Gang Huang. Stepfinder: A temporal semantic framework for failure attribution in multi-agent systems. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6946–6957, 2026.

APPENDIX

A ALGORITHM

Algorithm 1 One environment step of the HiCARE self-correcting loop.

Require: Public state s_t; planner π_θ; world model W_t; hierarchy $\mathcal{H}$; repair memory $\mathcal{C}_t$; counts $\{N_{\psi,r}, S_{\psi,r}\}$

Ensure: Committed transition and updated fusion statistics and repair memory

Sample an initial proposal

$$z_t^{(0)} \sim \pi_\theta(\cdot \mid s_t).$$

for $k = 0, \dots, K$ **do**

  $\hat{\boldsymbol{p}}^{(k)} \leftarrow W_t(s_t, z_t^{(k)}), \quad U^{(k)} \leftarrow \langle \boldsymbol{w}, \hat{\boldsymbol{p}}^{(k)} \rangle$

  if $U^{(k)} \geq \tau$ **or** $k = K$ **then**

    $\kappa_t \leftarrow k$; **break**

  end if

  for all $\ell \in \mathcal{L}$ **do**

$$\delta_\ell^\star \in \arg \max_{\delta \in \Delta_\ell^{\mathcal{C}_t}(s_t, z_t^{(k)})} U\Big(s_t, z_t^{(k)}[\ell \leftarrow \delta]\Big)$$

$$\mathrm{CFgain}_\ell \leftarrow U\Big(s_t, z_t^{(k)}[\ell \leftarrow \delta_\ell^\star]\Big) - U^{(k)}$$

    $\mathrm{VoInt}_\ell \leftarrow \mathrm{CFgain}_\ell - c_\ell$

  end for

  if $\max_{\ell \in \mathcal{L}} \mathrm{VoInt}_\ell < \eta$ **then**

    $\kappa_t \leftarrow k$; **break**

  end if

  $\ell_k^\star \leftarrow \arg\max_{\ell \in \mathcal{L}} \mathrm{VoInt}_\ell$ ▷ break ties toward the finer scope

  $z_t^{(k+1)} \leftarrow z_t^{(k)}[\ell_k^\star \leftarrow \delta_{\ell_k^\star}^\star]$

end for

Commit $z_t \leftarrow z_t^{(\kappa_t)}$ and execute $a_t \leftarrow \mathrm{Exec}(z_t)$

Observe $(o_{t+1}, \boldsymbol{y}_t^*) \sim P_{\mathrm{env}}(\cdot \mid h_t, a_t)$

Update $\{\bar{L}_{e,j,t+1}, \omega_{e,j,t+1}\}$ using Eqs. (8)–(9)

if $\kappa_t > 0$ **then**

  $\psi_t \leftarrow \phi(s_t, z_t^{(0)}), \quad r_t \leftarrow \rho(z_t^{(0)} \to z_t)$

  $N_{\psi_t, r_t} \leftarrow N_{\psi_t, r_t} + 1$

  if $\langle \boldsymbol{w}, \boldsymbol{y}_t^* \rangle \geq \tau_{\mathrm{val}}$ **then**

    $S_{\psi_t, r_t} \leftarrow S_{\psi_t, r_t} + 1$

  end if

end if

Construct $\mathcal{C}_{t+1}$ using Eq. (15)

return $(a_t, o_{t+1}, \boldsymbol{y}_t^*, \mathcal{C}_{t+1}, \{\bar{L}_{e,j,t+1}, \omega_{e,j,t+1}\})$

B THEORETICAL ANALYSIS AND CORRESPONDING PROOFS

We analyze a single controller decision at a fixed public state s, under a fixed world-model snapshot and repair memory $\mathcal{C}$. Recall that

$$U(s, z) = \langle \boldsymbol{w}, \hat{\boldsymbol{p}}(s, z) \rangle, \qquad U^*(s, z) = \langle \boldsymbol{w}, \boldsymbol{p}^*(s, z) \rangle$$

denote the predicted and true local utilities, respectively. For a proposal z, let

$$\mathcal{Q}_{\mathcal{C}}(s, z) := \{(\ell, \delta) : \ell \in \mathcal{L},\ \delta \in \Delta_\ell^{\mathcal{C}}(s, z)\}$$

be the finite set of admissible repairs. For $q = (\ell, \delta)$, write $z_q := z[\ell \leftarrow \delta]$ and define

$$V(q; s, z) := U(s, z_q) - U(s, z) - c_\ell, \qquad V^*(q; s, z) := U^*(s, z_q) - U^*(s, z) - c_\ell.$$

Local world-model accuracy. Following the bounded-model-error principle in model-based planning (Kearns & Singh, 2002; Lobel & Parr, 2024), define

$$\mathcal{Z}_{\mathrm{cf}}(s,z) := \{z\} \cup \{z_q : q \in \mathcal{Q}_{\mathcal{C}}(s,z)\}.$$

We say that W is *locally uniformly ε-accurate* if

$$\|\hat{\boldsymbol{p}}(s,\tilde{z}) - \boldsymbol{p}^*(s,\tilde{z})\|_\infty \le \varepsilon, \qquad \forall \tilde{z} \in \mathcal{Z}_{\mathrm{cf}}(s,z). \tag{19}$$

This condition is local to the proposal and candidate repairs considered in the current controller step. Since $\|\boldsymbol{w}\|_1 = 1$, it implies

$$|U(s,\tilde{z}) - U^*(s,\tilde{z})| \le \varepsilon \qquad \forall \tilde{z} \in \mathcal{Z}_{\mathrm{cf}}(s,z). \tag{20}$$

Proposition 1 (Reliable and near-optimal intervention)**.** *Suppose the controller intervenes with*

$$\hat{q} \in \arg\max_{q \in \mathcal{Q}_{\mathcal{C}}(s,z)} V(q;s,z), \qquad V(\hat{q};s,z) \ge \eta.$$

Under Eq. (19),

$$V^*(\hat{q};s,z) \ge \eta - 2\varepsilon, \tag{21}$$
$$V^*(\hat{q};s,z) \ge \max_{q \in \mathcal{Q}_{\mathcal{C}}(s,z)} V^*(q;s,z) - 2\varepsilon. \tag{22}$$

Consequently, if $\eta > 2\varepsilon$, every applied repair has positive true net value. Since $c_\ell \ge 0$, it also strictly improves the true local utility. Moreover, the selected repair is within 2ε of the best admissible repair under the true utility.

Proof of Proposition 1. By Eq. (20), every considered decision $\tilde{z} \in \mathcal{Z}_{\mathrm{cf}}(s,z)$ satisfies

$$|U(s,\tilde{z}) - U^*(s,\tilde{z})| \le \varepsilon.$$

Let $\hat{q} = (\hat{\ell}, \hat{\delta})$ be the repair selected by the controller. Its true net value satisfies

$$\begin{aligned} V^*(\hat{q};s,z) &= U^*(s,z_{\hat{q}}) - U^*(s,z) - c_{\hat{\ell}} \\ &\ge U(s,z_{\hat{q}}) - U(s,z) - c_{\hat{\ell}} - 2\varepsilon \\ &= V(\hat{q};s,z) - 2\varepsilon \\ &\ge \eta - 2\varepsilon, \end{aligned}$$

which proves Eq. (21).

For near-optimality, let

$$q^\star = (\ell^\star, \delta^\star) \in \arg\max_{q \in \mathcal{Q}_{\mathcal{C}}(s,z)} V^*(q;s,z).$$

Because $\hat{q}$ maximizes the predicted net value and the baseline $U(s,z)$ is shared by all repairs,

$$U(s,z_{\hat{q}}) - c_{\hat{\ell}} \ge U(s,z_{q^\star}) - c_{\ell^\star}.$$

Therefore,

$$\begin{aligned} V^*(\hat{q};s,z) &= U^*(s,z_{\hat{q}}) - c_{\hat{\ell}} - U^*(s,z) \\ &\ge U(s,z_{\hat{q}}) - c_{\hat{\ell}} - U^*(s,z) - \varepsilon \\ &\ge U(s,z_{q^\star}) - c_{\ell^\star} - U^*(s,z) - \varepsilon \\ &\ge U^*(s,z_{q^\star}) - c_{\ell^\star} - U^*(s,z) - 2\varepsilon \\ &= V^*(q^\star;s,z) - 2\varepsilon. \end{aligned}$$

This proves Eq. (22). □

Repair-memory compatibility. Memory-derived repairs enter $\mathcal{Q}_{\mathcal{C}}(s,z)$ as additional candidates and are evaluated by the same counterfactual controller. Proposition 1 therefore applies equally to newly generated and memory-retrieved repairs.

C LIMITATION, BROADER IMPACT, AND FUTURE WORK

Limitations Our current formulation assumes that the decision hierarchy and its repair scopes are specified from the task structure. Although this provides an explicit and controllable basis for hierarchical intervention, automatically discovering and adapting the hierarchy remains an important direction for future work. In addition, our evaluation focuses on the current benchmark settings; further experiments across more diverse domains, longer-horizon tasks, and larger-scale agent systems are needed to establish broader generality.

Broader Impact HiCARE may improve the reliability and interpretability of autonomous agents by detecting predicted failures and repairing structured decisions before they affect the environment. This property is particularly useful in settings where erroneous actions are costly or difficult to reverse. However, inaccurate world models, misspecified utilities, or incorrect rules may lead to inappropriate interventions or unwarranted confidence in repaired decisions. Deployment in high-stakes domains should therefore combine calibrated uncertainty, human oversight, and environment-specific validation rather than treating simulated outcomes as guarantees.

Future Work Future work will study how agents can learn variable-depth decision hierarchies and admissible repair operators instead of relying on predefined structures. A second direction is the joint adaptation of the hierarchy and world model, where prediction errors guide scope refinement and learned abstractions improve multi-scale prediction. We also plan to develop uncertainty-aware and budget-sensitive intervention policies, strengthen repair memory through causal validation, connect local repair utility more directly to terminal task success, and evaluate HiCARE across a broader range of embodied, tool-use, and long-horizon agent benchmarks.