

AdaptiveGuard: Treating Constraints as Falsifiable Hypotheses for LLM Agents in Dynamic Interactive Environments

Pengxiang Liu¹ Tao Ren^{1,*} Wei Xiong¹ Lijie Li² Junjie Wang¹ Jun Hu^{1,*}

¹ State Key Laboratory of Intelligent Game, Institute of Software, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China

² College of Computer Science and Technology, Harbin Engineering University, Harbin, China

{liupengxiang2024, rentao22, xiongwei2023}@iscas.ac.cn

lilijie@hrbeu.edu.cn

{junjie, hujun}@iscas.ac.cn

* Corresponding authors

Abstract

Large Language Models (LLMs) have shown impressive reasoning capabilities in agents for complex interactive environments. However, these agents frequently propose actions that violate physical or procedural constraints. Recent neuro-symbolic guards mitigate such errors by compiling explicit admissibility rules from oracle demonstrations, yet treat them as fixed laws—a closed-world assumption that breaks down under sparse demonstrations or environment shift, leaving either missing rules that fail to block unsafe actions or stale rules that wrongly reject valid alternatives. To address this, we introduce **AdaptiveGuard** (code), an Adaptive framework that Guards LLM agents by treating constraints as *falsifiable hypotheses*. Each rule is represented by a Beta-distributed validity belief mapped to *tentative*, *confirmed*, and *deprecated* tiers; uncertain rule blockers are then challenged through a zero-extra-call falsification gate; and these rule beliefs are continuously updated via contrastive feedback attribution from environment evidence. Benchmark experiments on ScienceWorld and VirtualHome show aggregate gains over strong static constraint guards, especially in novel scenarios. Complementary post-hoc diagnostics under sparse supervision and controlled shift show improved adaptation, while observed failed-trial runs remain below the configured budget cap and the conservative-bias proxy declines over the measured horizon.

1 Introduction

Agents powered by Large Language Models (LLMs) represent a significant step toward general-purpose decision-making in complex interactive environments (Wang et al., 2022; Puig et al., 2018; Wang et al., 2023b; Song et al., 2023). Unlike static NLP tasks, these agents must track dynamic state changes, respect physical affordances, and reason about long-horizon causal dependencies (Barsalou,

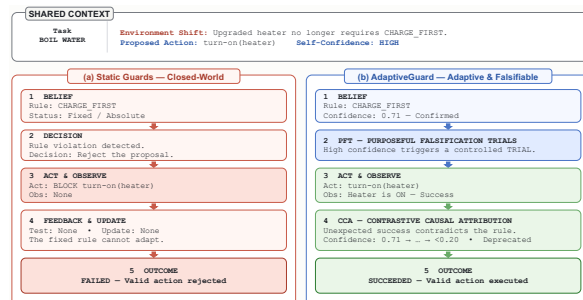


Figure 1: Motivating case under environment shift. When a new heater model no longer requires charging, a static closed-world guard treats the stale rule as fixed, blocks a valid action, and fails. AdaptiveGuard treats the rule as falsifiable: high agent confidence triggers its Purposeful Falsification Trials (PFT) module to challenge the stale blocker; a successful trial lets the attribution module deprecate the rule and complete the task.

2008). Existing approaches fall into two paradigms: *implicit* learning from demonstrations or sparse rewards (Chen et al., 2021; Luo et al., 2024; Lin et al., 2023; Lambert et al., 2025), and *explicit* admissibility modeling through guided action scoring (Qian et al., 2025). The former can adapt from experience but provides limited pre-execution validity guarantees, whereas the latter makes action validity explicit before execution but depends on whether the learned scorer remains reliable after deployment.

Recent work has therefore proposed static guards (Liu et al., 2026): rule-based filters that compile admissibility rules from demonstrations and use them during action selection. This design substantially improves action validity by turning demonstration-derived constraints into pre-execution filters, but it also creates a *closed-world* view in which extracted rules are treated as complete and reliable unless the rule set is externally rebuilt. Under environment shift or unseen valid alternatives, previously useful rules can become stale, and the guard ex-

Expert data	DGAP		OntoGuard	
	Score	Drop	Score	Drop
100%	85.91	—	89.56	—
75%	82.37	4.12%	85.64	4.38%
50%	74.18	13.65%	77.52	13.44%
25%	61.43	28.49%	65.27	27.12%
0%	40.71	52.61%	46.38	48.21%

Table 1: ScienceWorld sparse-expert diagnostic. Scores are Overall final scores; reduced-data rows summarize the diagnostic simulation. Drop reports the relative decrease from the 100% expert-data setting. Both the scoring-based DGAP baseline and the rule-guard OntoGuard baseline degrade as expert supervision becomes sparse.

hibits a **conservative bias** that rejects valid actions (Khetarpal et al., 2022). Figure 1 illustrates this stale-rule case: a fixed guard blocks a valid action because it cannot test whether the old constraint still holds.

By contrast, the sparse-supervision diagnostic illustrates the complementary failure mode. Table 1 holds the benchmark and planner backbone fixed; OntoGuard drops from 89.56 to 46.38, and DGAP from 85.91 to 40.71. Because the planner backbone is unchanged, the simulated OntoGuard trend isolates the missing-rule mechanism in rule-guard systems: the rule graph contains **knowledge gaps**, so invalid actions can slip through because no extracted rule blocks them (Li et al., 2022). Taken together, Figure 1 and Table 1 illustrate two opposite closed-world failure modes: environment shift can make existing rules stale, while sparse demonstrations can leave necessary rules absent. Thus, invalid actions require adding missing rules, whereas over-blocked valid actions require weakening stale ones; admissibility rules should therefore be treated as falsifiable beliefs that can be strengthened, challenged, or retired through environment evidence.

Building on this insight, we propose **AdaptiveGuard** around a closed-loop idea: admissibility constraints should be maintained as probabilistic beliefs, selectively challenged when a confident, goal-critical planner proposal conflicts with them, and revised through contrastive feedback attribution grounded in environment evidence. The three modules instantiate this belief–challenge–attribution loop. Stratified Constraint Belief (SCB) represents each rule as a Beta validity belief mapped to *tentative*, *confirmed*, and *deprecated* tiers. The Purposeful Falsification Trials (PFT) module challenges

uncertain confirmed rules by combining rule uncertainty, planner self-confidence, and goal criticality into a budgeted trial decision, using the planner’s confidence signal from the same forward pass without extra decision-path LLM calls. Contrastive Causal Attribution (CCA) evaluates executed actions through environment outcome judgments, task-progress changes, and hindsight cues, then uses structured labels to update the corresponding constraint type.

Our contributions are summarized as follows:

- We formulate guarded planning as falsifiable constraint learning, separating environment-level constraints from the adaptive belief graph refined online.
- We design a belief-centered adaptation loop that converts rule uncertainty, planner confidence, and goal criticality into budgeted falsification decisions, then uses contrastive environment evidence to promote, weaken, or retire constraints.
- We obtain the best results among the evaluated baselines on ScienceWorld (**91.70** Overall, **+2.14** over OntoGuard) and VirtualHome (**+7.00** SR on NovelScenes, **+5.50** SR on NovelTasks); labeled post-hoc diagnostics further probe sparse supervision, controlled shift, efficiency, and mechanism behavior.

2 Related Work

Learning-based and LLM-based Agents. Early text-based agents relied on reinforcement learning and imitation learning (Ross et al., 2011) to map observations to actions; representative systems such as DRRN (He et al., 2016), KG-A2C (Ammanabrolu and Hausknecht, 2020), CALM (Yao et al., 2020), and TDT (Chen et al., 2021) struggle with combinatorial action spaces across benchmarks like ScienceWorld, ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), and AgentBench (Liu et al., 2024). LLMs have since become the dominant substrate for agent reasoning: ReAct (Yao et al., 2023b) interleaves chain-of-thought (Wei et al., 2022) with actions, SayCan (ichter et al., 2023) grounds plans via value functions, Inner Monologue (Huang et al., 2022) injects environmental feedback, and SwiftSage (Lin et al., 2023) combines fast and slow policies. These frameworks remain agnostic to the *validity* of candidate actions, generating plausible plans without

verifying physical admissibility—motivating the guarded planning paradigm.

Post-Hoc Agent Adaptation. A large body of work revises plans *post hoc*: Reflexion (Shinn et al., 2023) and Retroformer (Yao et al., 2024) translate failures into verbal signals, Self-Refine (Madaan et al., 2023) iteratively rewrites with LLM feedback, and ISR-LLM (Zhou et al., 2024b) applies iterative refinement to long-horizon planning. Deliberate search methods expand the action space via tree rollouts, as in Tree of Thoughts (Yao et al., 2023a) and LATS (Zhou et al., 2024a). Experience-based agents accumulate cross-trial knowledge: Voyager (Wang et al., 2023a) grows a skill library, ExpeL (Zhao et al., 2024) distills insights, CLIN (Majumder et al., 2024) stores causal abstractions, and AdaPlanner (Sun et al., 2023) refines plans through closed-loop feedback. Yet these methods operate over *unstructured* memory that cannot be queried as fine-grained constraints nor selectively refuted. Our work shares the spirit of self-evolution (Wang et al., 2023a) but evolves a different object—the admissibility constraint graph—under bidirectional pressure that compensates for missing rules while removing spurious ones.

Neuro-Symbolic Guarded Planning. Neuro-symbolic guarded planning combines neural flexibility with symbolic constraint enforcement (Hao et al., 2023; Shi et al., 2024). In the LLM era, knowledge graphs are mainly used for retrieval-augmented generation (Zhao et al., 2023; Abernethy et al., 2024), often backed by ConceptNet (Speer et al., 2017). More targeted approaches deploy discriminators or symbolic checkers: DGAP (Qian et al., 2025; Xu et al., 2022) scores actions with a trained discriminator, OntoGuard (Liu et al., 2026) compiles rules into graph-based firewalls, and embodied planners like EmbodiedGPT (Mu et al., 2023) and DEPS (Wang et al., 2023b) rely on fixed perception-to-action mappings. These systems demonstrate that explicit constraints substantially improve admissibility, but they typically treat rules or action mappings as static, leaving limited support for reconciling conflicts with novel observations (Li et al., 2022; Lambert et al., 2025). AdaptiveGuard departs by lifting each rule from a boolean oracle to a falsifiable probabilistic hypothesis tested against environment evidence, thereby connecting Bayesian hypothesis revision with contrastive credit assignment through environment-

grounded rule updates (Piriyakulkij et al., 2024; Harutyunyan et al., 2019; Miller, 2019).

3 Methodology

As shown in Figure 2, **AdaptiveGuard** treats admissibility rules as falsifiable hypotheses with three modules: **Stratified Constraint Belief** (SCB) assigns probabilistic credence and enforcement tiers; **Purposeful Falsification Trials** (PFT) selectively challenges blocked actions through falsification trials; and **Contrastive Causal Attribution** (CCA) updates the rule graph from environment-grounded hindsight signals. Algorithm 1 (Appendix L) details the closed loop.

3.1 Problem Formulation

We formulate interactive planning as a POMDP augmented with a mutable constraint belief:

$$\mathcal{M}^+ = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, \mathcal{R}, \mathcal{G}, \mathcal{K}_t, \mathcal{B}_t \rangle, \quad (1)$$

where $\mathcal{K}_t$ is the agent’s belief over constraints and $\mathcal{B}_t$ the bounded exploration budget; AdaptiveGuard revises only $\mathcal{K}_t$, never the unobserved ground truth $\mathcal{K}^*$. Each rule is represented as

$$r_i = \langle (h_i, \text{rel}_i, t_i), \text{Beta}(\alpha_i, \beta_i), \ell_i \rangle, \quad (2)$$

with symbolic triplet (h_i, rel_i, t_i) , Beta-encoded confidence, and tier $\ell_i \in \{T, C, D\}$ (*tentative/confirmed/deprecated*). The objective trades off task reward against both missing and spurious constraints:

$$\max_{\pi, \mathcal{K}} \mathbb{E} \left[\sum_t r_t \right] - \lambda \left(|\mathcal{K}^* \setminus \mathcal{K}_T| + |\mathcal{K}_T \setminus \mathcal{K}^*| \right). \quad (3)$$

This is an idealized evaluation objective rather than an optimized training loss: $\mathcal{K}^*$ is unavailable to the deployed agent and is used only for offline rule-graph analysis.

3.2 Stratified Constraint Belief (SCB)

SCB stores each rule as a Bayesian hypothesis with mean confidence

$$\text{conf}(r_i) := \mathbb{E}_{X \sim \text{Beta}(\alpha_i, \beta_i)}[X] = \frac{\alpha_i}{\alpha_i + \beta_i}. \quad (4)$$

Oracle-extracted rules with n supports start as $\text{Beta}(n, 1)$; runtime rules start as $\text{Beta}(1, 1)$. Thresholds $\theta_{\text{low}} < \theta_{\text{high}}$ define three tiers: **deprecated (D)** ($\text{conf}(r_i) < \theta_{\text{low}}$; retained for audit but unenforced); **tentative (T)** ($\theta_{\text{low}} \leq \text{conf}(r_i) \leq$

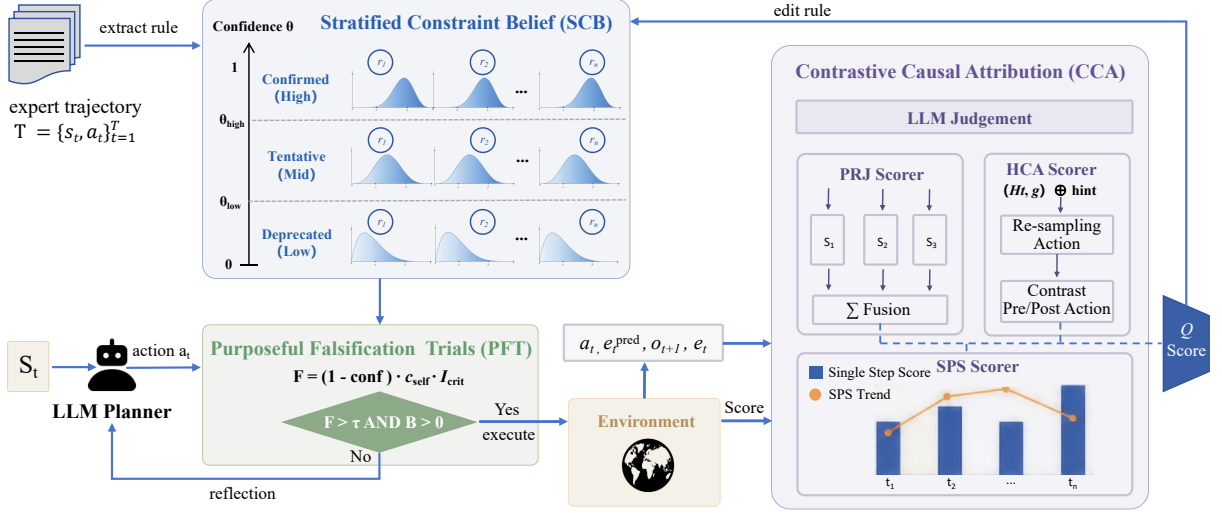


Figure 2: Architecture of AdaptiveGuard. The planner emits $\hat{a}_t$, $\hat{e}_t$, and c_{self} in one forward pass; SCB retrieves violated rules; PFT decides whether to block or launch a challenge trial; CCA then attributes the observed outcome and updates $\mathcal{K}_t$.

θ_{high} ; matched during attribution but not used as a hard blocker); **confirmed (C)** ($\text{conf}(r_i) > \theta_{\text{high}}$; blocks violating actions unless PFT launches a challenge trial). Matching tentative rules is essential: successful violations refute them, whereas failed violations provide the positive evidence needed for promotion.

3.3 Purposeful Falsification Trials (PFT)

PFT challenges only selected blocked actions through trials and adds no LLM call beyond the planner’s forward pass.

Challenge Score. For action $\hat{a}_t$ blocked by confirmed rule r_i , PFT computes

$$F(\hat{a}_t, r_i, s_t, g) = \underbrace{(1 - \text{conf}(r_i))}_{\text{uncertainty}} \underbrace{c_{\text{self}}(\hat{a}_t)}_{\text{self-conf.}} \cdot \underbrace{I_{\text{crit}}(\hat{a}_t, g)}_{\text{criticality}}. \quad (5)$$

The product is high only when the blocker is uncertain, the planner is confident, and the action is goal-critical.

Agent Self-Confidence c_{self} . The planner emits a discrete confidence label alongside $\hat{a}_t$ and predicted outcome $\hat{e}_t$, mapped deterministically:

HIGH $\rightarrow$ 0.9, MEDIUM $\rightarrow$ 0.6, LOW $\rightarrow$ 0.3.

Goal-Criticality Component I_{crit} . I_{crit} uses an incrementally extended trajectory buffer $\mathcal{T}_g^{(k)} = \mathcal{D}_g \cup \mathcal{T}_{\text{succ}}^{(k)}$ and no LLM call:

$$I_{\text{crit}}(\hat{a}_t, g) = \text{clip}_{[0,1]} \left(\max(\text{freq}(\hat{a}_t | g), \text{path}(\hat{a}_t | g), \text{bot}(\hat{a}_t | g)) + \epsilon_0 \right). \quad (6)$$

where freq is occurrence in successful trajectories, path shortest-path relevance, and bot the goal-reachability loss after removing the action template. The three components are normalized to $[0, 1]$; $\text{clip}_{[0,1]}$ projects the score to $[0, 1]$, and we use $\epsilon_0 = 0.1$ as a small exploration floor. When $\mathcal{T}_g^{(k)}$ is empty, we use the cold-start fallback of 0.5.

Decision Rule. Let $\mathcal{V}_t^C$ be the confirmed rules that block $\hat{a}_t$. If it is empty, the action is allowed. Otherwise, PFT computes F_i for every $r_i \in \mathcal{V}_t^C$ and selects $r^* = \arg \max_i F_i$ as the unique rule used for the subsequent contrastive update. A trial is allowed only if every confirmed blocker is individually eligible, preventing one uncertain rule from bypassing another high-confidence blocker:

$$\mathcal{C}_{\text{trial}}(\hat{a}_t) \triangleq \left(\bigwedge_{r_i \in \mathcal{V}_t^C} (F_i > \tau) \right) \wedge (\mathcal{B}_t > 0). \quad (7)$$

$$\text{Decision}(\hat{a}_t) = \begin{cases} \text{ALLOW} & \mathcal{V}_t^C = \emptyset, \\ \text{TRIAL} & \mathcal{V}_t^C \neq \emptyset \wedge \mathcal{C}_{\text{trial}}(\hat{a}_t), \\ \text{BLOCK} & \text{otherwise.} \end{cases} \quad (8)$$

Bounded Exploration Budget. Trials are controlled by an asymmetric budget initialized to $\mathcal{B}_{\text{base}}$ at the beginning of each episode:

$$\mathcal{B}_{t+1} = \begin{cases} \min\{\mathcal{B}_{\text{max}}, \mathcal{B}_t + \kappa\} & \text{success,} \\ \max\{0, \mathcal{B}_t - 1\} & \text{failure/uncertainty,} \\ \mathcal{B}_t & \text{otherwise.} \end{cases} \quad (9)$$

We maintain $0 \leq \mathcal{B}_t \leq \mathcal{B}_{\text{max}} = 2\mathcal{B}_{\text{base}}$, preventing unbounded accumulation after consecutive

successes. Successful trials refund partial budget because discovering a wrong rule improves the system, whereas failed or ambiguous trials consume one unit because they do not falsify the blocker. The budget controls trial frequency, not the harm magnitude of an action.

3.4 Contrastive Causal Attribution (CCA)

CCA maps contrastive environment evidence to rule hypotheses. Here, "causal" is used operationally to denote directional attribution from hindsight feedback, not formal causal identification. To avoid circularity, judgment signals use environment feedback rather than $\mathcal{K}_t$; only hypothesis generation may reference the target graph type, and its output is checked before insertion.

3.4.1 Multi-Signal Environment Evidence

In our instantiation, CCA uses three complementary evidence signals for each executed action a_t .

Process Reward Judge (PRJ). PRJ uses an LLM judge to assess the full next-state signal $s_{t+1} = (o_{t+1}, e_t, \Delta\text{score})$ via m queries. We retain both the majority label L_{PRJ} for discrete audit reporting and the signed vote mean V_{PRJ} for scalar fusion:

$$\begin{aligned} L_{\text{PRJ}} &= \text{MajVote}(r_1, \dots, r_m), \\ V_{\text{PRJ}} &= \frac{1}{m} \sum_{j=1}^m r_j, \quad r_j \in \{-1, 0, +1\}. \end{aligned} \quad (10)$$

Score Progression Signal (SPS). SPS tracks time-weighted score change. Each per-step score difference is first normalized to $[-1, 1]$ using the benchmark score range; the discounted mean is then clipped to the same interval:

$$V_{\text{SPS}}(a_t) = \text{clip}_{[-1,1]} \left(\frac{\sum_{k=0}^{W-1} \gamma^k \widetilde{\Delta\text{score}_{t-k}}}{\sum_{k=0}^{W-1} \gamma^k + 10^{-8}} \right). \quad (11)$$

Hindsight-guided Causal Attribution (HCA). HCA turns the gap between predicted outcome $\hat{e}_t$ and the actual observation into a directional update. A judge extracts a concise hindsight hint:

$$\begin{aligned} &\text{Judge}(a_t, \hat{e}_t, o_{t+1}, e_t) \\ &\rightarrow \{\text{score} \in \{+1, -1\}, \text{hint} \in \mathcal{T}^*\}. \end{aligned} \quad (12)$$

We run m hint-extraction calls, keep the longest valid hint (score = +1, length > 10 chars), append it as $s_{\text{enhanced}} = (H_t, g) \oplus \text{hint}$, and re-query the planner for a teacher action a_t^{enh} . A structured judge maps

$(a_t, a_t^{\text{enh}}, \text{hint})$ to six labels: IDENTICAL (0.0, none), PARAM_DIFF ($-0.4, G_{\text{env}}.E_1$), MISSING_PRECOND ($-0.8, G_{\text{beh}}$), WRONG_OBJECT ($-0.6, G_{\text{env}}.E_1$), WRONG_TIMING ($-0.7, G_{\text{env}}.E_3$), BOTH_WRONG (0.0, none). Here, $G_{\text{env}}.E_1$ denotes object/action-interface constraints, $G_{\text{env}}.E_3$ temporal environment constraints, and G_{beh} behavioral preconditions. The structured judge runs $m_H = 3$ times; without at least two agreeing votes, HCA contributes $V_{\text{HCA}} = 0$ and the remaining available signals determine the final judgment.

3.4.2 Contrastive Signal Fusion

Adaptive Fusion. The resulting evidence vector $V(a_t) = [V_{\text{PRJ}}, V_{\text{SPS}}, V_{\text{HCA}}]$ is fused into

$$Q(a_t) = \sum_{k=1}^3 w_k \cdot V_k(a_t), \quad (13)$$

using the default weights $[0.35, 0.25, 0.40]$. Environment errors use $[0.55, 0.20, 0.25]$; silent reward uses $[0.35, 0.10, 0.55]$. If HCA is invalid, its weight is set to zero and the remaining default weights are renormalized to $[0.583, 0.417, 0]$. Each schedule sums to one. The final judgment is

$$\text{Judgment}(a_t) = \begin{cases} \text{GOOD} & Q(a_t) > \theta^+, \\ \text{BAD} & Q(a_t) < \theta^-, \\ \text{UNCERTAIN} & \text{otherwise.} \end{cases} \quad (14)$$

Contrastive Attribution. CCA contrasts the environment-grounded judgment with matched rule predictions. For an executed trial, GOOD weakens the responsible confirmed blocker with $\beta \neq 1$, whereas BAD strengthens it with $\alpha \neq 1$. For an allowed action, each attribution-compatible matching tentative rule receives refuting evidence under GOOD and supporting evidence under BAD. A BAD/allow outcome with no matching active rule spawns an HCA-label-typed tentative rule. UNCERTAIN never creates or removes a rule; each matching tentative rule receives only a directional half-step update η according to the sign of Q .

Consistent Hypothesis Generation. When Judgment is BAD and no active rule predicted the failure, CCA formalizes a new rule from the HCA hint and label, falling back to PRJ error semantics if needed. New hypotheses enter as tentative with Beta(1, 1); a semantically equivalent existing tentative rule is updated rather than duplicated. Each

edit then checks behavioral acyclicity, resolves contradictions by higher confidence, and freezes rules flipped more than twice within $W = 10$ edits.

3.5 Theoretical Properties

Three conditional properties follow under the idealized assumptions stated in Appendix K: (1) if the ground-truth and candidate rule spaces are finite and realizable, rule-relevant witnesses retain a history-conditional execution probability bounded away from zero, and CCA assigns correct updates, the active rule set converges to $\mathcal{K}^*$ almost surely; (2) PFT admits at most $\mathcal{B}_{\max} = 2\mathcal{B}_{\text{base}}$ consecutive failed or ambiguous PFT trials within an episode; and (3) each correctly refuted spurious rule has decreasing confidence. The aggregate confidence mass over active spurious rules is an operational conservative-bias proxy, not a distribution-free false-positive rate, and can temporarily increase when new hypotheses are introduced.

4 Experiments

The experiments test three claims: AdaptiveGuard improves standard interactive-agent performance, mitigates sparse-data and environment-shift failures of closed-world guards, and benefits from the evaluated PFT factors and CCA evidence dimensions. We evaluate main results, stress tests, ablations, efficiency, conditional-property diagnostics, transfer, failure modes, and cases.

4.1 Experimental Setup

Benchmarks. We evaluate on **ScienceWorld** (Wang et al., 2022), our primary testbed with 30 science task types across 10 topics and average oracle-trajectory lengths from 7.0 to 132.1 steps, and **VirtualHome** (Puig et al., 2018), whose *In-Distribution*, *NovelScenes*, and *NovelTasks* splits test environment shift.

Baselines and Backbone. We compare seven groups: learning-based agents DRRN (He et al., 2016), KG-A2C (Ammanabrolu and Hausknecht, 2020), CALM (Yao et al., 2020), TDT (Chen et al., 2021); zero-shot planners SayCan (ichter et al., 2023) and ReAct (Yao et al., 2023b); self-correction methods Reflexion (Shinn et al., 2023), plus Self-Refine (Madaan et al., 2023) and ISR-LLM (Zhou et al., 2024b) for post-shift self-correction; experience-based agents ExpeL (Zhao et al., 2024) and CLIN (Majumder et al., 2024); AdaPlanner (Sun et al., 2023); SwiftSage (Lin et al.,

2023); and neuro-symbolic guards DGAP (Qian et al., 2025) and OntoGuard (Liu et al., 2026). The archived summaries identify the LLM backbone only as **GPT-4**; they do not preserve an API snapshot or complete decoding configuration, so we do not claim cross-snapshot reproducibility or smaller-planner generalization.

Implementation Details. We fix $\theta_{\text{high}} = 0.70$, $\theta_{\text{low}} = 0.20$, $\tau = 0.25$, $\mathcal{B}_{\text{base}} = 5$, $\mathcal{B}_{\max} = 10$, $\kappa = 0.5$, $\theta^+ = 0.50$, $\theta^- = -0.50$, $\eta = 0.5$, $\gamma = 0.9$, $W = 10$, tier refresh period $N = 1$, PRJ queries $m = 3$, HCA hint-extraction queries $m = 3$, HCA structured-judge votes $m_H = 3$, $\epsilon_0 = 0.1$, HCA minimum hint length = 10 characters, and self-confidence map $\{\text{HIGH, MEDIUM, LOW}\} \rightarrow \{0.9, 0.6, 0.3\}$. Results average **3 random seeds**.

Evaluation Metrics. We report task performance (ScienceWorld Final Score; VirtualHome SR), admissibility (EXEC), efficiency (tokens per action and PFT decision-path calls), observed falsification cost (failed PFT trials), and available rule-graph quality measures (rule precision and symmetric-difference error against an independent $\mathcal{K}^*$; Appendix F). We do not claim unreported rule recall or a complete severity-weighted unsafe-execution rate.

4.2 Main Results

ScienceWorld. Table 2 summarizes that AdaptiveGuard reaches **91.70** Overall, improving over OntoGuard (89.56) by **+2.14 points**. Gains increase with horizon (Short +0.87, Medium +2.55, Long +2.93), where sparse rewards make HCA most useful. Full 30-task results are reported in Appendix Table 7. AdaptiveGuard is best or tied on **25 of 30 tasks** (8 at the 100.0 ceiling). Table 5 lists the five exceptions. These cases suggest that the remaining gaps arise on tasks where task-specific knowledge, semantic classification, or precise search can dominate over constraint adaptation: exact action sequencing or experimental search in 3-1 and 9-2, conductivity and lifespan knowledge in 3-3 and 7-1, and semantic classification in 4-3. AdaptiveGuard still remains $\geq$ OntoGuard in all five, indicating method boundaries rather than regressions.

Paired Statistical Stability. We compute paired differences on the 30 ScienceWorld task types, with each task type paired across methods. Table 4 re-

Tasks Topic		Algorithms										
		Learning-based Agents				LLM-based Planners						
		DRRN	KG-A2C	CALM	TDT	SayCan	ReAct	Reflexion	SwiftSage	DGAP	OntoGuard	AdaptiveGuard (Ours)
Summary	Short	28.08	22.70	11.30	28.37	43.83	48.79	71.47	92.22	90.84	93.83	94.70
	Medium	10.85	6.88	2.88	10.36	36.58	44.01	35.43	77.79	81.84	86.0	88.55
	Long	8.26	4.92	1.33	6.11	23.65	21.07	30.17	83.00	84.52	88.37	91.30
Total	Overall	15.56	11.37	5.07	14.66	33.82	36.43	45.34	84.68	85.91	89.56	91.70

Table 2: ScienceWorld final scores by horizon and overall (higher is better). Short/Medium/Long follow the trajectory-length horizon grouping used in OntoGuard. LLM-based methods use GPT-4; results are means over 3 seeds. Bold denotes the per-row best.

Method	In-Dist.		NovelScenes		NovelTasks	
	EXEC	SR	EXEC	SR	EXEC	SR
ReAct	87.33	82.33	38.67	32.33	49.67	49.00
Reflexion	79.67	79.33	54.33	53.33	47.33	46.00
SwiftSage	90.67	84.33	63.00	56.33	78.33	68.00
DGAP	93.33	88.00	71.67	62.67	73.67	72.17
OntoGuard	94.83	89.50	74.20	64.50	76.15	74.50
AdaptiveGuard	95.50	90.80	82.10	71.50	80.40	80.00
Δ vs. OntoGuard	+0.67	+1.30	+7.90	+7.00	+4.25	+5.50

Table 3: VirtualHome execution/admissibility rate (EXEC) and success rate (SR, %). Δ reports percentage-point changes relative to OntoGuard.

ports the AdaptiveGuard margin, its win/tie/loss count, and an exact two-sided sign test over non-tied task types. Against the closest baseline, OntoGuard, AdaptiveGuard has a mean paired gain of +2.14 with a 95% task-level paired- t interval of [1.37, 2.91], wins on 22 task types, and ties on 8, with no task-level regressions. This interval treats task types as inferential units and does not replace seed-level uncertainty, which cannot be reconstructed from rounded aggregate means.

VirtualHome. Table 3 reports retained three-seed aggregate means. AdaptiveGuard shows a small In-Distribution gain over OntoGuard (+1.30 SR), but larger aggregate gains under NovelScenes (+7.00) and NovelTasks (+5.50). Run-level dispersion is unavailable, so these differences are descriptive rather than seed-level significance claims.

4.3 Sparse Expert Data

Figure 3 is a post-hoc diagnostic simulation that varies the amount of oracle supervision used to bootstrap the rule graph while holding the reported benchmark aggregate and planner fixed. OntoGuard drops from 89.56 at 100% to 53.00 at 10% (−36.56), whereas AdaptiveGuard drops from 91.70 to 71.00 (−20.70); the OntoGuard drop is 1.77× the AdaptiveGuard drop. AdaptiveGuard@10% (71.00) exceeds OntoGuard@25% and all lower-data regimes, but remains below

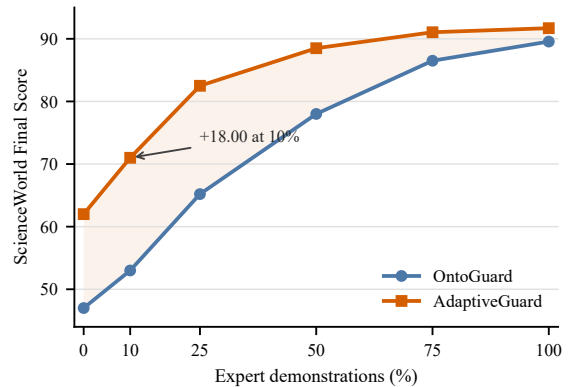


Figure 3: ScienceWorld sparse-data diagnostic simulation. AG = AdaptiveGuard. Curves are aggregate sensitivity summaries; no subsampling uncertainty is inferred from the retained rounded values.

OntoGuard@50%. Because the retained artifact does not include per-seed subsampling manifests or replicate outputs, this diagnostic supports an aggregate sensitivity claim rather than a variance-calibrated robustness estimate.

4.4 Adversarial Environment Shift

Beyond the natural VirtualHome shift, we report a controlled ScienceWorld shift diagnostic: at episode 50 of a 100-episode sequence, a synthetic intervention changes environment dynamics and makes previously valid constraint assump-

Baseline	Overall	AG margin	AG W/T/L	Sign-test p
DRRN	15.56	+76.14	30/0/0	$< 10^{-8}$
KG-A2C	11.37	+80.33	30/0/0	$< 10^{-8}$
CALM	5.07	+86.63	30/0/0	$< 10^{-8}$
TDT	14.66	+77.04	30/0/0	$< 10^{-8}$
SayCan	33.82	+57.88	29/0/1	$< 10^{-7}$
ReAct	36.43	+55.27	30/0/0	$< 10^{-8}$
Reflexion	45.34	+46.36	26/1/3	1.5×10^{-5}
SwiftSage	84.68	+7.02	24/5/1	1.5×10^{-6}
DGAP	85.91	+5.79	19/8/3	8.6×10^{-4}
OntoGuard	89.56	+2.14	22/8/0	4.8×10^{-7}

Table 4: Paired task-level comparison on the 30 ScienceWorld task types. AG = AdaptiveGuard. The exact two-sided sign test excludes tied tasks.

Task	OntoGuard	Best baseline	AG
3-1 (Create circuit, S)	88.91	Reflexion 100.0	90.40
3-3 (Test Conductivity, M)	80.00	SayCan 99.56	84.20
4-3 (Find a plant, S)	95.50	DGAP 100.0	96.30
7-1 (Identify longest-lived, S)	91.60	DGAP/Reflexion 100.0	92.40
9-2 (Friction known surfaces, L)	81.00	Reflexion 100.0	84.50

Table 5: The five tasks on which AdaptiveGuard (AG) is not the best. S/M/L denote short-, medium-, and long-horizon groups; Best baseline is the strongest non-AdaptiveGuard method. These exceptions are consistent with task-specific bottlenecks only partly addressed by adaptive constraint repair: exact action sequencing or experimental search (3-1, 9-2), domain-fact retrieval (3-3, 7-1), and semantic classification (4-3). AG remains $\geq$ OntoGuard on every one of them (Δ from +0.80 to +4.20).

PFT Internal Ablation			
Variant	Novel	Δ	Rule Prec.
Full PFT	71.50	—	—
w/o (1 — conf)	66.90	−4.60	—
w/o c_{self}	67.30	−4.20	—
w/o I_{crit}	65.80	−5.70	—
CCA Signal Ablation and Hint-Quality Audit			
Variant	Novel	Δ	Rule Prec.
Full CCA	71.50	—	0.84
w/o PRJ (SPS+HCA)	66.30	−5.20	0.76
w/o HCA (PRJ+SPS)	61.40	−10.10	0.63
w/o SPS (PRJ+HCA)	67.60	−3.90	0.79
HCA free-form score	66.50	−5.00	0.72
w/o UNCERTAIN	63.10	−8.40	0.67

Table 6: Combined ablation results. Both PFT and CCA blocks report Novel, the VirtualHome NovelScenes SR (%), and Δ , the percentage-point change relative to the corresponding full variant. Rule Prec. denotes rule precision and is reported only for CCA variants; — indicates not applicable.

tions stale. We measure Pre-shift SR, immediate Post-shift SR (ep 50–54), and Recovery latency L (episodes to recover to 90% of pre-shift SR; capped at 100). The retained artifact does not contain the intervention manifest needed to reproduce the exact shifted task instances, so this is interpreted as

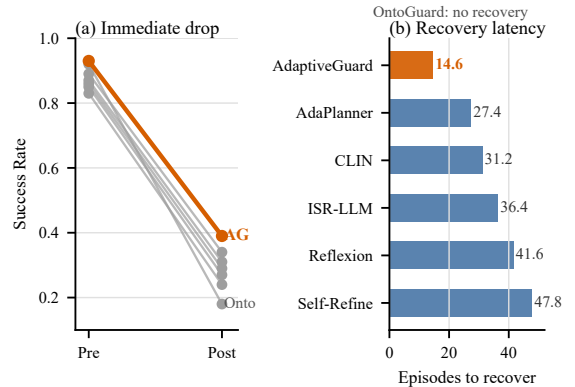


Figure 4: ScienceWorld adversarial environment shift. Left: immediate pre/post-shift degradation. Right: recovery latency L ; lower is better.

a controlled diagnostic rather than a standardized benchmark split.

Figure 4 compares all adaptive paradigms under the same abrupt environment shift. Static OntoGuard drops to 0.18 post-shift because its closed-world constraints become stale, and it never recovers ($L = \infty$). Self-correction baselines recover slowly ($L \approx 36$ –48), while CLIN and AdaPlanner improve through structured caching ($L \approx 27$ –31). AdaptiveGuard recovers fastest ($L = 14.6$), a 47%

latency reduction over AdaPlanner ($L = 27.4$), showing the value of constraint-level memory under environment change.

4.5 Ablation Study

PFT internal ablation. All three challenge-score factors matter under NovelScenes shift (Table 6): removing uncertainty, c_{self} , or I_{crit} reduces SR by 4.60, 4.20, and 5.70 points, respectively. The largest drop confirms the importance of focusing the bounded trial budget on goal-critical actions.

CCA signal ablation and hint-quality audit. HCA is the strongest CCA source ($\Delta = -10.10$); removing the UNCERTAIN band also causes a large drop ($\Delta = -8.40$), while PRJ, SPS, and the structured HCA rubric contribute 5.20, 3.90, and 5.00 points. In an independent audit of 200 stratified HCA invocations, three annotators obtain 75.5% Hint Accuracy (Correct+Partial), 52.5% Specificity, 7.5% Vacuity Rate, and Krippendorff’s $\alpha = 0.78$. Wrong hints correlate with BOTH_WRONG (Spearman $\rho = 0.42$), supporting the safety route for degenerate cases.

4.6 Cross-Task and Cross-Benchmark Transfer

We compare targets initialized from their demonstration graph with the same graph augmented by $\mathcal{K}^{1-1}$ learned over 100 source episodes; the planner and updater are otherwise identical. This is therefore a warm-start diagnostic, not an AdaptiveGuard-versus-OntoGuard comparison. Appendix Table 10 reports a large within-topic early gain (+25.6), a smaller cross-topic gain (+3.5), and nearly flat cross-benchmark transfer (+0.4 SR). The aggregate source retest changes only from 91.70 to 91.85; absent run-level uncertainty, this is no detected degradation rather than proof of no forgetting.

4.7 Safety Cost of Falsification

Appendix Table 11 separates falsification cost from task performance. Failed trials fall from 2.8 to 0.6 per episode between the first and last 25-episode windows; the observed per-episode maximum is six, while the hard cap bounds consecutive failed or ambiguous trials at ten. On 200 paired challenging episodes, adaptation uniquely fails 8 cases and uniquely repairs 34, a net gain of 13.0 points. Neither these counts nor rule precision bounds harm severity, so high-risk actions still require an exter-

nal risk classifier or human approval and simulation or shadow-mode testing.

4.8 Efficiency and LLM-Call Cost

Appendix Table 12 separates early adaptation cost from steady-state efficiency. Across task horizons, AdaptiveGuard averages 2556.32 tokens per action over episodes 1–30, 15.9% above OntoGuard, because PRJ/HCA voting adds attribution calls. Over episodes 71–100, the average falls to 2093.09, 5.1% below OntoGuard, as refined rules prune invalid actions earlier. Overall values weight the 10/8/12 Short/Medium/Long task counts. PFT itself adds no decision-path LLM calls because it reuses planner-emitted c_{self} and deterministic checks. These measurements characterize token and call efficiency rather than wall-clock latency.

4.9 Additional Results

Further diagnostics are reported in the appendix. Appendix G bounds score variation at 3.05 points under the tested $\pm 25\%$ and $\pm 50\%$ perturbations. Appendix H identifies beyond-capability tasks (30.0%) as the largest overall failure category and HCA mis-attribution (20.5%) as the largest method-attributable category. Appendix I.1 confirms AdaptiveGuard’s smaller sparse-data drop and shortest shift-recovery latency (Tables 17 and 18). Appendix I.2 quantifies judge sensitivity and summarizes HCA safeguards (Tables 19 and 20). Appendix I.3 links reduced visitation to slower repair and performance loss, while showing modest overhead at $10\times$ graph scale (Tables 21 and 22). Finally, Appendix J illustrates stale-rule falsification, HCA-based hypothesis generation, oscillation control, and the SCB lifecycle under shift.

5 Conclusion

AdaptiveGuard safeguards LLM agents by treating constraints as falsifiable hypotheses rather than fixed laws. SCB, PFT, and CCA maintain probabilistic rules, permit budgeted trials, and revise constraints from contrastive feedback. AdaptiveGuard reaches 91.70 on ScienceWorld (+2.14 over OntoGuard), +5.50 on VirtualHome NovelTasks, and 47% faster recovery under adversarial shift. These results highlight online, evidence-guided repair of missing and stale constraints. Ablations and appendix diagnostics support the PFT/CCA mechanisms, moderate overhead, and stable finite-horizon behavior.

Limitations

AdaptiveGuard has several practical limitations. CCA adds early LLM calls for PRJ/HCA voting, yielding a 15.9% token overhead before convergence, though the refined rule graph later gives a 5.1% reduction. The convergence property (Proposition 1) assumes non-zero visitation of rule-relevant state–action pairs, so convergence may be slower in large or sparse-feedback environments. Cross-benchmark transfer is modest (+0.4 SR from ScienceWorld to VirtualHome), suggesting some simulator-specific rule structure. Finally, failures can still arise from noisy attribution or backbone factual/reasoning limits.

Acknowledgments

This work was supported by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB0900000.

References

- Jacob Abernethy, Alekh Agarwal, Teodor Vanislavov Marinov, and Manfred K. Warmuth. 2024. [A mechanism for sample-efficient in-context learning for sparse retrieval tasks](#). In *Proceedings of The 35th International Conference on Algorithmic Learning Theory*, volume 237 of *Proceedings of Machine Learning Research*, pages 3–46. PMLR.
- Prithviraj Ammanabrolu and Matthew Hausknecht. 2020. [Graph constrained reinforcement learning for natural language action spaces](#). In *International Conference on Learning Representations*.
- Lawrence Barsalou. 2008. [Grounded cognition](#). *Annual Review of Psychology*, 59:617–645.
- Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.
- Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 8154–8173.
- Anna Harutyunyan, Will Dabney, Thomas Mesnard, Nicolas Heess, Mohammad G. Azar, Bilal Piot, Hado van Hasselt, Satinder Singh, Greg Wayne, Doina Precup, and Rémi Munos. 2019. *Hindsight credit assignment*. Curran Associates Inc., Red Hook, NY, USA.
- Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Li-hong Li, Li Deng, and Mari Ostendorf. 2016. [Deep reinforcement learning with a natural language action space](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1621–1630, Berlin, Germany. Association for Computational Linguistics.
- Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and brian ichter. 2022. [Inner monologue: Embodied reasoning through planning with language models](#). In *6th Annual Conference on Robot Learning*.
- brian ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. 2023. [Do as i can, not as i say: Grounding language in robotic affordances](#). In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 287–318. PMLR.
- Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. 2022. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. [RewardBench: Evaluating reward models for language modeling](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797, Albuquerque, New Mexico. Association for Computational Linguistics.
- Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, Jacob Andreas, Igor Mordatch, Antonio Torralba, and Yuke Zhu. 2022. Pre-trained language models for interactive decision-making. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS ’22, Red Hook, NY, USA. Curran Associates Inc.
- Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. 2023. Swiftsage: a generative agent with fast and slow

- thinking for complex interactive tasks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.
- Pengxiang Liu, Tao Ren, Wei Xiong, Tingrui Yang, Junjie Wang, and Jun HU. 2026. [OntoGuard: Enforcing action admissibility for LLM agents in complex interactive environments](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 20937–20952, San Diego, California, United States. Association for Computational Linguistics.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. [Agent-Bench: Evaluating LLMs as agents](#). In *The Twelfth International Conference on Learning Representations*.
- Jianlan Luo, Perry Dong, Yuexiang Zhai, Yi Ma, and Sergey Levine. 2024. [RLIF: Interactive imitation learning as reinforcement learning](#). In *The Twelfth International Conference on Learning Representations*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. 2024. [CLIN: A continually learning language agent for rapid task adaptation and generalization](#). In *First Conference on Language Modeling*.
- Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.
- Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. 2023. [EmbodiedGPT: Vision-language pre-training via embodied chain of thought](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Wasu Top Piriyaikulij, Cassidy Langenfeld, Tuan Anh Le, and Kevin Ellis. 2024. [Doing experiments and revising rules with natural language and probabilistic reasoning](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Xavier Puig, Kevin Kyunghwan Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. [Virtualhome: Simulating household activities via programs](#). *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8494–8502.
- Haofu Qian, Chenjia Bai, Jiatao Zhang, Fei Wu, Wei Song, and Xuelong Li. 2025. [Discriminator-guided embodied planning for llm agent](#). In *International Conference on Learning Representations*, volume 2025, pages 92439–92467.
- Stephane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. [A reduction of imitation learning and structured prediction to no-regret online learning](#). In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA. PMLR.
- Yu-Zhe Shi, Haofei Hou, Zhangqian Bi, Fanxu Meng, Xiang Wei, Lecheng Ruan, and Qining Wang. 2024. [AutoDSL: Automated domain-specific language design for structural representation of procedures with constraints](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12177–12214, Bangkok, Thailand. Association for Computational Linguistics.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: language agents with verbal reinforcement learning](#). In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. [ALFWorld: Aligning text and embodied environments for interactive learning](#). In *International Conference on Learning Representations*.
- Chan Hee Song, Jiaman Wu, Clay Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. 2023. [Llm-planner: Few-shot grounded planning for embodied agents with large language models](#). *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2986–2997.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17*, page 4444–4451. AAAI Press.
- Haotian Sun, Yuchen Zhuang, Ling kai Kong, Bo Dai, and Chao Zhang. 2023. [AdaPlanner: Adaptive planning from feedback with language models](#). In *Advances in Neural Information Processing Systems*.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi (Jim) Fan,

- and Anima Anandkumar. 2023a. [Voyager: An open-ended embodied agent with large language models](#). *ArXiv*, abs/2305.16291.
- Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. [ScienceWorld: Is your agent smarter than a 5th grader?](#) In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Shawn Ma, and Yitao Liang. 2023b. Describe, explain, plan and select: interactive planning with llms enables open-world multi-task agents. *Advances in Neural Information Processing Systems*, 36:34153–34189.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*.
- Haoran Xu, Xianyu Zhan, Honglei Yin, and Huiling Qin. 2022. [Discriminator-weighted offline imitation learning from suboptimal demonstrations](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24725–24742. PMLR.
- Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. [WebShop: Towards scalable real-world web interaction with grounded language agents](#). In *Advances in Neural Information Processing Systems*.
- Shunyu Yao, Rohan Rao, Matthew Hausknecht, and Karthik Narasimhan. 2020. [Keep CALM and explore: Language models for action generation in text-based games](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8736–8754, Online. Association for Computational Linguistics.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023a. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. [React: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh R N, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, Ran Xu, Phil L Mui, Huan Wang, Caiming Xiong, and Silvio Savarese. 2024. [Retroformer: Retrospective large language agents with policy gradient optimization](#). In *The Twelfth International Conference on Learning Representations*.
- Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. [ExpeL: LLM agents are experiential learners](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.
- Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. [Large language models as commonsense knowledge for large-scale task planning](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024a. [Language agent tree search unifies reasoning, acting, and planning in language models](#). In *Forty-first International Conference on Machine Learning*.
- Zhehua Zhou, Jiayang Song, Kunpeng Yao, Zhan Shu, and Lei Ma. 2024b. [ISR-LLM: iterative self-refined large language model for long-horizon sequential task planning](#). In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 2081–2088. IEEE.

A Supplementary Main Results

This section expands the main experimental summary in Section 4.2. The main text reports the horizon-level ScienceWorld averages and the overall comparison needed for the central claims; here we provide the full per-task breakdown so that the aggregate trends can be inspected at the level of individual task families. Table 7 lists all 30 ScienceWorld task types, grouped by the same short-, medium-, and long-horizon categories used in the main text, and reports each baseline alongside AdaptiveGuard under the shared GPT-4 backbone and three-seed evaluation protocol. These detailed results also make clear where AdaptiveGuard’s gains are broad, where they concentrate on long-horizon or sparse-feedback tasks, and where competing methods remain competitive because of task-specific priors or exhaustive trajectory search.

Tasks Topic		Algorithms										
		Learning-based Agents				LLM-based Planners						
		DRRN	KG-A2C	CALM	TDT	SayCan	ReAct	Reflexion	SwiftSage	DGAP	OntoGuard	AdaptiveGuard (Ours)
Matter	1-1 (L)	3.52	0.0	0.0	0.71	33.06	3.52	4.22	97.04	100.0	100.0	100.0
	1-2 (L)	3.52	0.0	0.0	0.44	10.39	13.70	10.61	87.04	92.75	94.20	96.50
	1-3 (L)	0.0	4.0	0.0	3.88	3.88	7.78	7.78	72.78	74.0	79.40	83.80
	1-4 (L)	0.0	0.0	0.0	0.55	0.37	9.88	0.92	100.0	100.0	100.0	100.0
Measurement	2-1 (M)	6.56	6.0	1.0	6.16	26.37	7.19	5.92	99.17	100.0	100.0	100.0
	2-2 (M)	5.50	11.0	1.0	6.43	8.03	6.10	28.59	88.17	80.17	89.31	91.50
	2-3 (L)	6.0	4.0	1.0	19.87	17.41	22.37	22.37	95.73	88.33	97.14	98.50
Electricity	3-1 (S)	12.0	7.0	5.0	40.55	52.14	56.0	100.0	88.67	91.50	88.91	90.40
	3-2 (M)	9.0	4.0	7.0	14.26	22.50	54.33	17.45	55.33	58.0	68.50	73.50
	3-3 (M)	9.05	4.0	2.0	10.16	99.56	76.19	72.54	71.90	78.57	80.0	84.20
	3-4 (M)	9.52	4.0	2.0	21.65	47.76	88.81	70.22	77.86	88.14	88.50	90.50
Classification	4-1 (S)	15.0	18.0	10.0	41.93	22.87	26.67	64.93	100.0	100.0	100.0	100.0
	4-2 (S)	45.0	44.0	54.0	55.76	58.18	80.0	87.27	100.0	100.0	100.0	100.0
	4-3 (S)	21.67	16.0	10.0	27.82	20.87	53.33	16.42	91.67	100.0	95.50	96.30
	4-4 (S)	19.17	15.0	8.0	47.15	31.43	27.50	100.0	100.0	100.0	100.0	100.0
Biology	5-1 (L)	8.0	6.0	2.0	6.89	9.92	9.06	7.33	74.59	73.14	78.16	84.30
	5-2 (L)	14.29	11.0	4.0	11.86	13.93	18.57	13.0	93.93	90.57	97.0	98.20
Chemistry	6-1 (M)	15.77	17.0	3.0	15.10	47.81	51.04	70.35	49.40	57.40	66.20	71.80
	6-2 (S)	26.67	19.0	6.0	15.70	39.26	58.89	70.67	100.0	100.0	100.0	100.0
	6-3 (M)	10.37	4.0	3.0	5.25	19.72	40.74	15.77	91.48	92.43	95.50	96.90
Biology	7-1 (S)	50.0	43.0	6.0	30.0	80.0	60.0	100.0	95.0	100.0	91.60	92.40
	7-2 (S)	50.0	32.0	10.0	8.43	67.50	67.50	84.37	85.0	85.71	90.50	92.50
	7-3 (S)	33.33	23.0	4.0	8.34	50.0	50.0	83.0	93.33	92.71	94.40	95.30
Biology	8-1 (M)	21.0	5.0	4.0	3.86	20.91	27.67	2.58	89.0	100.0	100.0	100.0
	8-2 (S)	8.0	10.0	0.0	8.0	16.0	8.0	8.0	68.50	38.50	77.38	80.10
Forces	9-1 (L)	10.0	4.0	0.0	2.53	21.94	40.50	50.63	75.0	75.0	80.20	84.40
	9-2 (L)	10.0	4.0	3.0	14.66	32.26	44.0	100.0	70.0	83.33	81.0	84.50
	9-3 (L)	10.0	4.0	2.0	9.12	13.67	41.0	70.62	60.0	71.43	72.50	79.40
Biology	10-1 (L)	16.80	11.0	2.0	1.51	67.53	25.70	50.90	92.30	87.71	96.30	97.50
	10-2 (L)	17.0	11.0	2.0	1.29	59.45	16.80	23.69	77.60	78.0	84.50	88.50
Summary	Short	28.08	22.70	11.30	28.37	43.83	48.79	71.47	92.22	90.84	93.83	94.70
	Medium	10.85	6.88	2.88	10.36	36.58	44.01	35.43	77.79	81.84	86.0	88.55
	Long	8.26	4.92	1.33	6.11	23.65	21.07	30.17	83.00	84.52	88.37	91.30
Total	Overall	15.56	11.37	5.07	14.66	33.82	36.43	45.34	84.68	85.91	89.56	91.70

Table 7: Full comparison of average final scores on the ScienceWorld benchmark across 30 task types. Tasks are categorized by complexity (Short, Medium, Long) following the trajectory-length grouping used in OntoGuard. LLM-based methods use GPT-4 as the backbone; results report the mean over 3 seeds. Bold values denote the per-row best.

B Dataset Statistics

B.1 ScienceWorld

ScienceWorld contains 30 task types spanning ten science topics. Following the reporting format of OntoGuard, Table 8 lists the average oracle-trajectory length ($*Len$) and available variations in each data split. The horizon summary groups

tasks by interaction length to characterize short-, medium-, and long-horizon evaluation regimes.

B.2 VirtualHome

VirtualHome (Puig et al., 2018) represents household activities as executable programs in graph-structured scenes. Table 9 summarizes the benchmark setup.

Task Type	Topic	Name	*Len	#Vars: Train	Dev	Test
1-1	Matter	Changes of State (Boiling)	107.7	14	7	9
1-2	Matter	Changes of State (Melting)	78.6	14	7	9
1-3	Matter	Changes of State (Freezing)	88.9	14	7	9
1-4	Matter	Changes of State (Any)	75.2	14	7	9
2-1	Measurement	Use Thermometer	21.4	270	10	10
2-2	Measurement	Measuring Boiling Point (known)	35.2	218	10	10
2-3	Measurement	Measuring Boiling Point (unknown)	65	150	10	10
3-1	Electricity	Create a circuit	13.6	10	5	5
3-2	Electricity	Renewable vs Non-renewable Energy	20.8	10	5	5
3-3	Electricity	Test Conductivity (known)	25.6	450	225	225
3-4	Electricity	Test Conductivity (unknown)	29	300	10	10
4-1	Classification	Find a living thing	14.6	150	10	10
4-2	Classification	Find a non-living thing	8.8	150	10	10
4-3	Classification	Find a plant	12.6	150	10	10
4-4	Classification	Find an animal	14.6	150	10	10
5-1	Biology	Grow a plant	69.5	62	10	10
5-2	Biology	Grow a fruit	79.6	62	10	10
6-1	Chemistry	Mixing (generic)	33.6	16	8	8
6-2	Chemistry	Mixing paints (secondary colours)	15.1	18	9	9
6-3	Chemistry	Mixing paints (tertiary colours)	23	18	9	9
7-1	Biology	Identify longest-lived animal	7	62	10	10
7-2	Biology	Identify shortest-lived animal	7	62	10	10
7-3	Biology	Identify longest-then-shortest-lived animal	8	62	10	10
8-1	Biology	Identify life stages (plant)	40	6	3	5
8-2	Biology	Identify life stages (animal)	16.3	4	2	4
9-1	Forces	Inclined Planes (determine angle)	97	84	42	42
9-2	Forces	Friction (known surfaces)	84.9	692	346	348
9-3	Forces	Friction (unknown surfaces)	123.1	80	40	42
10-1	Biology	Mendelian Genetics (known plants)	130.1	60	30	30
10-2	Biology	Mendelian Genetics (unknown plants)	132.1	240	120	120
Short ($0 < *Len \leq 20$)			11.76	81.80	8.60	8.80
Medium ($20 < *Len \leq 50$)			28.58	161.00	35.00	35.25
Long ($*Len > 50$)			94.30	123.83	53.00	54.00
Overall (avg)			49.26	119.73	33.40	33.93
Overall (sum)			N/A	3,592	1,002	1,018

Table 8: Statistics of the ScienceWorld benchmark. *Len is the average length of oracle-agent trajectories; variations are reported for each official split. Horizon summaries use the underlying unrounded trajectory lengths, so they can differ by 0.01 from averages recomputed from the displayed one-decimal task values.

Aspect	Specification
Scenes	7 household scenes (scene IDs 0–6)
Environment encoding	JSON graph with object nodes and relational edges
Activity encoding	Program as a sequence of atomic action instructions
Executable state data	Initial/final graphs and per-step state lists
Evaluation settings	In-Distribution, NovelScenes, NovelTasks
Tasks per setting	60

Table 9: VirtualHome environment and evaluation statistics.

C Implementation Details and Prompt Interfaces

Computational Cost. Table 12 reports tokens per action. PFT adds no extra LLM call: c_{self} is planner-emitted, I_{crit} is cached, and the decision-rule checks are deterministic.

Prompt Interfaces. We specify prompt behavior through input–output interfaces rather than reproducing full templates. The planner receives the current observation o_t , history H , and goal g , and returns an action $\hat{a}_t$, expected outcome $\hat{e}_t$, and confidence label $\ell \in \{\text{HIGH}, \text{MEDIUM}, \text{LOW}\}$. PFT consumes ℓ as c_{self} via CONFMAP without another

LLM call, while HCA uses $\hat{e}_t$ for outcome comparison. PRJ receives $(a_t, o_{t+1}, e_t, \Delta\text{score})$ and outputs $r_i \in \{+1, -1, 0\}$. HCA hint extraction compares $(a_t, \hat{e}_t, o_{t+1}, e_t)$ and returns a binary usefulness score plus an optional concise hint. The structured HCA judge maps $(a_t, a_t^{\text{enh}}, \text{hint})$ to one of IDENTICAL, PARAM_DIFF, MISSING_PRECOND, WRONG_OBJECT, WRONG_TIMING, or BOTH_WRONG. Finally, hypothesis generation is invoked only after a BAD/allow contrast with no matching active rule; it receives the failed action, hindsight signal, and target graph type, then returns a candidate symbolic rule for consistency checking before insertion. These interfaces define the algorithmic contract, but the current artifact does not contain the exact historical prompt strings or parser logs.

CCA Weight Schedule. CCA uses default weights $[0.35, 0.25, 0.40]$ for $[V_{\text{PRJ}}, V_{\text{SPS}}, V_{\text{HCA}}]$. Environment errors use $[0.55, 0.20, 0.25]$, and silent reward uses $[0.35, 0.10, 0.55]$. Invalid HCA hints use $[0.583, 0.417, 0]$, obtained by proportionally renormalizing the default PRJ/SPS weights. These fixed schedules are not tuned on evaluation data.

Static Prompt Audit. Each rendered CCA input (PRJ, HCA hint extraction, structured rubric judge) is scanned for serialized $\mathcal{K}_t$ content (rule text, edge list, node list); detected leakage aborts and logs the call. PFT needs no audit because it makes *no LLM calls at all* on the decision path. Only post-judgment hypothesis generation may reference the target graph type, and its output is still filtered by the consistency check.

D Transfer and Falsification-Cost Diagnostics

Table 10 gives the full initialization diagnostic under decreasing source–target relatedness. Within-topic initialization reuses 72% of the source rule graph. The source retest uses the same aggregate transfer protocol before and after transfer and is not the 1-1 row of Table 7.

Setting	Transfer pair	Result
Within-topic SW	1-1 $\rightarrow$ 1-2	+25.6 target score (ep. 1–10)
Cross-topic SW	1-1 $\rightarrow$ 4-1	+3.5 target score
Cross-benchmark	SW $\rightarrow$ VH	+0.4 target SR
Source retest	aggregate source evaluation	91.85 vs. 91.70

Table 10: Initialization diagnostic for transfer of $\mathcal{K}_T$. SW = ScienceWorld; VH = VirtualHome. ScienceWorld rows use Final Score; the VirtualHome row uses SR.

Table 11 reports falsification cost separately from task performance. The paired challenging-episode counts include both cases harmed by adaptation and cases repaired only by adaptation.

Diagnostic	Window / sample	Value
Failed PFT trials per episode	ep. 1–25	2.8
Failed PFT trials per episode	ep. 76–100	0.6
Maximum PFT trials	failed all episodes	6
Budget cap $\mathcal{B}_{\text{max}}$	all episodes	10
AdaptiveGuard-only failures	200 paired episodes	8/200 (4.0%)
AdaptiveGuard-only successes	200 paired episodes	34/200 (17.0%)
Net paired gain over OntoGuard	same paired set	+13.0 pp

Table 11: Safety-cost diagnostics for bounded falsification. The net paired gain is in percentage points (pp).

Rule precision of 0.84 is a rule-graph quality measure, not a guarantee of zero unsafe execution. The budget limits consecutive failed or ambiguous PFT trials, while the observed counts quantify their frequency; neither bounds action-harm magnitude. With $\tau = 0.25$, $c_{\text{self}} \leq 0.9$, and $I_{\text{crit}} \leq 1$, any rule with $\text{conf}(r) \geq 1 - 0.25/0.9 \approx 0.7222$ has $F \leq 0.25$ and cannot pass the strict $F > \tau$ gate. A failed trial consumes budget and reinforces the responsible rule, making repeated mistaken challenges harder. The planner emits c_{self} , whereas I_{crit} uses cached successful trajectories, deterministic path relevance, and goal-reachability loss.

E Efficiency: Tokens and PFT LLM Calls

Method	Short	Med.	Long	Overall
SwiftSage	716.30	1277.52	444.09	757.07
ReAct	1047.52	1742.18	2893.19	1971.03
DGAP	1150.25	1850.60	3050.45	2097.09
OntoGuard	1250.40	1980.80	3150.15	2205.07
AG (early)	1395.20	2284.10	3705.40	2556.32
AG (converged)	1218.40	1908.30	2945.20	2093.09

Table 12: Tokens per action (t/a). AG = AdaptiveGuard. “early” = ep 1–30; “converged” = ep 71–100.

Table 12 shows that AdaptiveGuard costs 15.9% more tokens early than OntoGuard due to PRJ/HCA querying, but 5.1% fewer tokens after convergence because the refined rule graph prunes invalid actions earlier. Overall values are task-level averages weighted by the 10/8/12 Short/Medium/Long task counts. PFT adds zero decision-path LLM calls by using planner-emitted c_{self} and deterministic decision-rule checks.

F Finite-Horizon Diagnostics for Conditional Properties

Independent Gold-Standard $\mathcal{K}^*$. To avoid circular evaluation, $\mathcal{K}^*$ combines three *model-independent* sources:

$$\mathcal{K}^* = \mathcal{K}_{\text{oracle}} \cup \mathcal{K}_{\text{engine}} \cup \mathcal{K}_{\text{human}}, \quad (15)$$

where $\mathcal{K}_{\text{oracle}}$ comes from ScienceWorld oracle traces, $\mathcal{K}_{\text{engine}}$ from public engine preconditions, and $\mathcal{K}_{\text{human}}$ from three graduate annotators over 100 sampled (state, valid-actions) pairs. Krippendorff’s α is 0.83, and no element of $\mathcal{K}^*$ is derived from AdaptiveGuard outputs.

Metric	Ep1	Ep10	Ep25	Ep50	Ep75	Ep100
$ \mathcal{K}_t \triangle \mathcal{K}^* $	39.0	28.5	19.7	12.3	7.4	4.7
Bias proxy	8.5	—	5.7	3.4	2.0	1.2

Table 13: Finite-horizon diagnostics associated with Propositions 1 and 3 on ScienceWorld.

Episode range	Failed PFT/ep	Max	Bound
1–25	2.8	6	10
26–50	1.9	5	10
51–75	1.1	4	10
76–100	0.6	3	10

Table 14: Finite-horizon diagnostic associated with Proposition 2. Failed PFT/ep is the average number of failed falsification trials per episode; Max is the maximum number in one episode. The bound applies to consecutive failed or ambiguous trials, not to all rule violations caused by missing rules.

Tables 13–14 report diagnostics related to all three propositions; finite-horizon trends do not prove their asymptotic assumptions:

- **Proposition 1.** $|\mathcal{K}_t \triangle \mathcal{K}^*|$ decreases from 39.0 to 4.7 (88% reduction) over the measured horizon.
- **Proposition 2.** The observed maximum number of failed PFT trials stays below $\mathcal{B}_{\text{max}} = 10$ (peak 6 in early training, mean 0.6 by ep 76–100), consistent with the within-episode consecutive failed-or-ambiguous-trial bound.
- **Proposition 3.** The operational conservative-bias proxy $\sum_{r_i \in \mathcal{K}_T \setminus \mathcal{K}^*} \text{conf}(r_i)$ declines from 8.5 to 1.2 at the reported checkpoints.

G Hyperparameter Sensitivity

Param	Value	−50%	−25%	+25%	+50%	Range
θ_{high}	0.70	89.85	91.05	91.20	90.10	1.85
θ_{low}	0.20	90.20	91.10	91.40	90.30	1.50
τ	0.25	88.65	90.95	91.20	90.05	3.05
κ	0.50	89.50	90.85	91.10	89.75	2.20
Default Overall Score: 91.70						

Table 15: Hyperparameter scan on ScienceWorld Overall. Range includes the 91.70 default score together with the four perturbed settings; peak-to-trough variation is ≤ 3.05 .

Table 15 shows a 3.05-point peak-to-trough range when the default is included. Most perturbed settings exceed OntoGuard (89.56); only $\tau - 50\%$ and $\kappa - 50\%$ fall slightly below, by 0.91 and 0.06 points.

H Failure-Mode Analysis

We sample 200 AdaptiveGuard failure episodes and classify them into seven mutually exclusive categories. This failure-only audit is distinct from the separate 200-episode paired comparison reported below.

ID	Failure category	Share
F1	PFT false trial	9.5%
F2	PFT missed trial	7.5%
F3	HCA mis-attribution	20.5%
F4	Oscillation freeze of a correct rule	9.5%
F5	PRJ soft-failure miss	8.0%
F6	Hypothesis-quality issue	15.0%
F7	Task beyond LLM capability boundary	30.0%

Table 16: Failure-mode analysis over 200 AdaptiveGuard failure episodes.

Table 16 shows that about 30% are F7 (beyond LLM capability), while the largest method-attributable categories are HCA mis-attribution (F3, 20.5%) and hypothesis quality (F6, 15.0%). Separately, in a second paired set of 200 challenging episodes, 158 cases are shared failures, 8 are AdaptiveGuard-only failures (4.0%), and 34 are AdaptiveGuard-only successes (17.0%), yielding a +13 percentage-point net gain against OntoGuard.

I Additional Robustness and Sensitivity Analyses

This section supplements the main-text safety-cost analysis (Section 4.7) with robustness checks for sparse data and environment shift, judge reliability, state-action visitation, and rule-graph scale. Unless otherwise stated, entries are means over the same three random seeds and use the same task splits and GPT-4 backbone as the main experiments.

I.1 Sparse-Data and Environment-Shift Robustness

Table 17 isolates the two endpoints of the 30-task sparse-expert simulation. The benchmark and planner remain fixed while the oracle demonstrations used to initialize the rule graph are reduced. AdaptiveGuard incurs a smaller performance drop than OntoGuard, while its 10% result remains below the full-data static-guard score.

Data	OntoGuard	AG	AG margin
100%	89.56	91.70	+2.14
10%	53.00	71.00	+18.00
Drop magnitude	36.56	20.70	Onto/AG = 1.77×

Table 17: Sparse-expert robustness on the 30-task ScienceWorld extension.

Table 18 reports the recovery-latency summary for the controlled ScienceWorld shift. All methods receive the same post-shift interaction window;

lower latency is better, and ∞ denotes no recovery within the 100-episode horizon.

Method	Recovery latency L
OntoGuard	∞
Self-Refine	47.8
Reflexion	41.6
ISR-LLM	36.4
CLIN	31.2
AdaPlanner	27.4
AdaptiveGuard	14.6

Table 18: Recovery latency after the episode-50 ScienceWorld environment shift.

I.2 Judge Reliability and Attribution Safeguards

Table 19 varies the judge configuration while holding the planner, rule initialization, and budget fixed. Weaker instructions and single-vote judging reduce performance and rule precision, but the structured HCA route remains above the no-HCA condition. This sensitivity is consistent with the HCA hint audit (75.5% accuracy and 52.5% specificity) and makes judge dependence an explicit limitation.

Judge setting	NovelScenes SR	Rule Prec.	HCA Acc.
GPT-4, $m = 3$	71.50	0.84	75.5%
GPT-4, $m = 1$	68.90	0.78	69.0%
Weaker instructions, $m = 3$	66.20	0.72	64.5%
Weaker instructions, $m = 1$	63.80	0.68	59.0%
No HCA (PRJ+SPS)	61.40	0.63	—

Table 19: NovelScenes judge sensitivity.

Mechanism	Role in attribution control
$m = 3$ voting	Reduces single-sample judge variance
Structured labels	Prevents free-form hints from directly becoming rules
UNCERTAIN band	Avoids forced updates under ambiguous evidence
Consistency and oscillation checks	Blocks contradictory or unstable rule edits
Tentative Beta(1, 1) initialization	Prevents new hypotheses from becoming hard constraints immediately

Table 20: Safeguards applied to HCA-derived rule updates.

Table 20 summarizes the controls applied to HCA-derived edits. These safeguards prevent a

single unsupported hint from becoming an immediately enforced rule; ambiguous evidence receives only a weak update. Here, “causal” in CCA denotes operational directional attribution from contrastive hindsight feedback, not formal causal identification.

I.3 Visitation and Rule-Graph Scaling

The convergence statements assume repeated visits to rule-relevant state–action pairs. Table 21 reduces that visitation rate while preserving the task distribution. Lower visitation slows rule repair and increases conservative bias, but the update direction remains stable over the finite evaluation horizon.

Visitation	Rule error	Cons. bias	Overall
Full (100%)	4.7	1.2	91.70
Half (50%)	8.9	2.1	90.10
Sparse (25%)	13.8	3.7	87.60

Table 21: ScienceWorld visitation stress at episode 100.

We also scale the stored rule graph with inactive rules while preserving the active task distribution. Table 22 reports the resulting symbolic-operation costs. Retrieval is hash-indexed by action template and object/state predicates, while contradiction and acyclicity checks are local to edited predicate subgraphs.

Graph scale	Retrieval (ms)	Check (ms)	Runtime
1×	0.18	2.3	2.1%
5×	0.19	2.7	2.4%
10×	0.20	3.4	2.9%

Table 22: Rule-graph scaling micro-benchmark. Retrieval and consistency-check values are milliseconds; Runtime is the share of per-action runtime excluding LLM calls.

The near-constant retrieval time and small consistency-check share indicate that LLM calls, rather than graph maintenance, dominate the tested runtime. Larger open-ended environments may nevertheless require indexing and pruning beyond the present benchmark scale.

I.4 Decision Semantics and Reproducibility Clarifications

For completeness, the implementation semantics are as follows. If no confirmed rule blocks the proposed action, the action is ALLOW and is executed directly; matching tentative rules are retained for

post-execution attribution. If confirmed blockers are present, r^* denotes the blocker with maximum challenge score and is the unique rule used for the contrastive update. A trial is permitted only when every confirmed blocker satisfies the trial threshold; otherwise the action is BLOCK and the planner replans without execution. This makes explicit that the budget bounds falsification trials, not missing-rule violations.

The sparse-data diagnostic keeps the 30-task aggregate and planner fixed while varying the oracle supervision used to bootstrap the initial rule graph. The adversarial-shift diagnostic changes environment dynamics at episode 50 of a 100-episode sequence and reports immediate degradation and recovery latency under a matched interaction window. The retained summaries record matched GPT-4 backbone and action-cap conditions for the LLM-based comparisons, but not the exact API snapshot, numeric cap, subsampling manifests, or shift-instance manifest; these omissions delimit reproducibility.

PRJ uses majority voting for its discrete audit label. Scalar fusion uses the signed mean of the same votes, so the majority label and scalar carry complementary information. This is why Case 2 reports a majority of 0 together with a raw mean of -0.33 .

The source-retest row in Table 10 uses the same aggregate transfer-diagnostic protocol before and after warm-start transfer. It is not the 1-1 task row or the 30-task Overall value in Table 7.

J Case Studies

Case 1 (PFT correctly challenges a stale rule with a trial under the three-factor formula). In modified “boil water”, OntoGuard blocks every turn-on heater action via CHARGE_FIRST. AdaptiveGuard also blocks at Episode 12 ($c_{\text{self}} = \text{MEDIUM}$, $F = 0.083 < \tau$), but after repeated evidence the planner raises c_{self} to HIGH. At Episode 27, $F = (1 - 0.71) \cdot 0.9 \cdot 0.97 = 0.253 > \tau$ and $\mathcal{B} > 0$, so a TRIAL fires. The trial succeeds, $\beta \vdash= 1$, and the rule later enters the deprecated tier. The PFT decision path adds no extra LLM calls.

Case 2 (HCA structured rubric provides directional attribution under zero reward). In “grow plant”, open(bee_hive) at Adult stage yields “Nothing happens”, no error, and no score change. PRJ ($m = 3$) votes $\{0, -1, 0\}$ (majority = 0, raw

mean -0.33), and SPS delta = 0. HCA extracts the hint “*The plant must reach Flowering stage before pollinator introduction; Adult $\neq$ Flowering.*”; the enhanced action is `wait_for(flowering)`, labeled MISSING_PRECOND with score -0.8 and target G_{beh} . With silent SPS, $Q = 0.55(-0.8) + 0.35(-0.33) = -0.56 < \theta^-$, so CCA generates “*open(bee_hive) requires plant.stage == Flowering*” as Beta(1, 1) tentative. Without HCA, Q remains UNCERTAIN and no hypothesis is added.

Case 3 (Consistency guard prevents oscillation).

During episode 73 of a Chemistry task, the rule USE_GLOVES_BEFORE_ACID is added and removed three times in an 8-step window. The oscillation guard ($W = 10$, threshold 2) detects the third flip and freezes the rule, preventing repeated oscillatory edits.

Case 4 (SCB lifecycle under shift).

R_pollinate_after_flowering is born at episode 3 (tentative, Beta(1,1)) after HCA MISSING_PRECOND, promoted at episode 8 ($\text{conf} = 0.80$), and enforced through episodes 8–49 ($F < \tau$, $c_{\text{self}} = \text{MEDIUM}$). After the episode-50 shift removes the precondition, CCA accumulates β -evidence from GOOD bypasses; when conf falls to ~ 0.71 and c_{self} becomes HIGH, a TRIAL succeeds at episode 70. The rule later enters the deprecated tier at episode 88, completing a 38-episode post-shift transition from confirmed to deprecated.

K Proofs of Theoretical Properties

This appendix provides the formal statements and corresponding proof sketches for the theoretical properties introduced in §3.

Proposition 1 (Asymptotic Correctness). *Let $\mathcal{H}$ be the candidate hypothesis space explored by CCA. Assume (i) $\mathcal{K}^*$ and $\mathcal{H}$ are finite, (ii) $\mathcal{K}^* \subseteq \mathcal{H}$ and each rule has an identifiable state–action witness, (iii) for some $\epsilon > 0$, the history-conditional probability of executing every unresolved rule’s witness is at least ϵ , and (iv) CCA assigns the correct rule-status update for each observed witness. Then, the active rule set $\mathcal{K}_T$ converges to $\mathcal{K}^*$ almost surely as $T \rightarrow \infty$:*

$$\mathbb{P}\left(\lim_{T \rightarrow \infty} \mathcal{K}_T = \mathcal{K}^*\right) = 1. \quad (16)$$

Proof Sketch. The convergence requires proving both soundness ($\mathcal{K}_T \setminus \mathcal{K}^* \rightarrow \emptyset$) and completeness ($\mathcal{K}^* \setminus \mathcal{K}_T \rightarrow \emptyset$).

Step 1: Soundness (Elimination of False Rules).

Consider a spurious rule $r_i \in \mathcal{K}_t \setminus \mathcal{K}^*$. The history-conditional visitation lower bound implies that its witness is executed infinitely often almost surely. Under the attribution assumption, each such observation yields refuting evidence. With α_i fixed and β_i increasing, $\text{conf}(r_i) = \alpha_i / (\alpha_i + \beta_i)$ converges to 0. Consequently, there exists a finite time T_i such that $\text{conf}(r_i) < \theta_{\text{low}}$, at which point r_i enters the deprecated state.

Step 2: Completeness (Discovery of Missing Rules).

Consider a missing rule $r_j \in \mathcal{K}^* \setminus \mathcal{K}_t$. Realizability and identifiability ensure that an observed witness lets CCA instantiate the correct r_j as a tentative hypothesis. Subsequent matched violations are executed because tentative rules do not hard-block; correct failed outcomes increase α_j until $\text{conf}(r_j) > \theta_{\text{high}}$, promoting r_j to confirmed.

Because the relevant sets are finite, the intersection of these almost-sure soundness and completeness events also has probability one. The realizability, attribution, and visitation assumptions are idealized conditions rather than guarantees for an unrestricted open world; in particular, the per-episode budget reset does not itself ensure the required visitation lower bound. $\square$

Proposition 2 (Bounded Exploration Count).

Within one episode, the maximum number of consecutive failed or ambiguous TRIAL actions is upper-bounded by $\mathcal{B}_{\text{max}} = 2\mathcal{B}_{\text{base}}$. This proposition does not bound unsafe actions caused by missing rules, which are outside the PFT budget.

Proof Sketch. The budget dynamics are clipped to $[0, \mathcal{B}_{\text{max}}]$:

$$\mathcal{B}_{t+1} = \begin{cases} \min\{\mathcal{B}_{\text{max}}, \mathcal{B}_t + \kappa\}, & \text{success,} \\ \max\{0, \mathcal{B}_t - 1\}, & \text{failure/uncertainty.} \end{cases} \quad (17)$$

A necessary condition for initiating a TRIAL is $\mathcal{B}_t > 0$. Since $\mathcal{B}_t \leq \mathcal{B}_{\text{max}}$ and every failed or ambiguous trial decrements it by one, a consecutive run cannot exceed $\mathcal{B}_{\text{max}}$ before the budget reaches zero. Successful trials may replenish budget, so the statement does not bound the total number of trials in an episode. $\square$

Proposition 3 (Rule-wise Conservative-Bias Decay).

For a fixed spurious rule r_i , every correctly attributed refutation strictly decreases $\text{conf}(r_i)$. At any fixed time, if $\text{conf}(r_i)$ is additionally calibrated to upper-bound the probability that r_i causes a

false block, then

$$\mathbb{P}(\mathcal{E}_{\text{FP}}) \leq \min \left\{ 1, \sum_{r_i \in \mathcal{K}_T \setminus \mathcal{K}^*} \text{conf}(r_i) \right\}. \quad (18)$$

Without this extra calibration assumption, the sum is only an operational conservative-bias proxy. The aggregate proxy is not guaranteed to be monotone when CCA introduces new hypotheses.

Proof Sketch. For fixed $\alpha_i > 0$, a correct refutation changes the confidence from $\alpha_i/(\alpha_i + \beta_i)$ to $\alpha_i/(\alpha_i + \beta_i + \Delta)$ for $\Delta > 0$, which is strictly smaller. Under the optional calibration assumption, a union bound over the active spurious rules yields the displayed probability bound, clipped at one. New spurious hypotheses can add positive confidence mass, so this rule-wise argument does not establish global monotonicity of the aggregate proxy. $\square$

L Full Algorithm

Algorithm 1 gives the closed-loop procedure for SCB, PFT, and CCA (§3). The planner emits action, expected outcome, and self-confidence in one pass; SCB retrieves violated tiers; PFT applies the challenge score and budgeted decision rule; CCA then fuses PRJ/SPS/HCA signals to update beliefs or generate typed hypotheses. Periodic tier updates promote or deprecate rules by confidence.

Algorithm 1 AdaptiveGuard Closed-Loop Planning

Require: Goal g , initial state $(O_0, \mathcal{K}_0)$; action cap $A_{\max}$
Require: Thresholds $\theta_{\text{low}} < \theta_{\text{high}}, \tau, \theta^-, \theta^+$
Require: Budget $\mathcal{B}_{\text{base}}$, cap $\mathcal{B}_{\text{max}}$, refund κ ; hyperparams $\eta, N, m, m_H, \epsilon_0$
Require: Confidence map CONFMAP
Ensure: Updated rule belief $\mathcal{K}$

```

1: Initialize  $\mathcal{K}, \mathcal{T}_g, H, \mathcal{S}$ , planning graph
2:  $\mathcal{B} \leftarrow \mathcal{B}_{\text{base}}, t \leftarrow 0$ 
3: while task not terminal  $\wedge t < A_{\max}$  do
4:   Phase 1: Planning
5:    $(\hat{a}_t, \hat{e}_t, \ell) \leftarrow \text{PLANNER}(H, g)$ 
6:    $c_{\text{self}} \leftarrow \text{CONFMAP}[\ell]$ 
7:   Phase 2: Rule activation and PFT decision
8:    $\mathcal{V}_t^C \leftarrow \{r_i \in \mathcal{K}_{\text{confirmed}} : r_i \text{ blocks } \hat{a}_t\}$ 
9:    $\mathcal{V}_t^T \leftarrow \{r_i \in \mathcal{K}_{\text{tentative}} : r_i \text{ matches } \hat{a}_t\}$ 
10:  if  $\mathcal{V}_t^T = \emptyset$  then
11:    decision  $\leftarrow$  ALLOW
12:  else
13:     $I_t \leftarrow 0.5$  if  $\mathcal{T}_g = \emptyset$ ; else  $\min\{1, \max(\text{freq}, \text{path}, \text{bot}) + \epsilon_0\}$ 
14:     $F_i \leftarrow (1 - \text{conf}(r_i)) \cdot c_{\text{self}} \cdot I_t$  for each  $r_i \in \mathcal{V}_t^C$ 
15:     $r^* \leftarrow \arg \max_{r_i \in \mathcal{V}_t^C} F_i$ 
16:    if  $\min_{r_i \in \mathcal{V}_t^C} F_i > \tau \wedge \mathcal{B} > 0$  then
17:      decision  $\leftarrow$  TRIAL
18:    else
19:      decision  $\leftarrow$  BLOCK + replan
20:    end if
21:  end if
22:  if decision = BLOCK then
23:    Append block feedback to  $H$ ;  $t \leftarrow t + 1$ ; continue
24:  end if
25:  Phase 3: Execution
26:  Execute  $\hat{a}_t$ , observe  $O_{t+1}, e_t, \text{score}_{t+1}$ 
27:  Phase 4: Contrastive causal attribution
28:   $(L_{\text{PRJ}}, V_{\text{PRJ}}) \leftarrow \text{PRJJUDGE}(m)$ 
29:   $V_{\text{SPS}} \leftarrow \text{EVALNORMALIZEDSPS}(\mathcal{S}, \text{score}_{t+1})$ 
30:   $(V_{\text{HCA}}, L_{\text{HCA}}) \leftarrow \text{STRUCTUREDHCA}(m, m_H)$ 
31:   $w \leftarrow \text{WEIGHTSCHEDULE}(e_t, V_{\text{SPS}}, V_{\text{HCA}})$ 
32:   $Q \leftarrow w^\top [V_{\text{PRJ}}, V_{\text{SPS}}, V_{\text{HCA}}]$ 
33:  Judgment  $\leftarrow \text{THRESHOLD}(Q, \theta^-, \theta^+)$ 
34:  Phase 5: Rule belief update
35:  if decision = TRIAL then
36:    if Judgment = GOOD then
37:       $\beta(r^*) \leftarrow 1$ 
38:       $\mathcal{B} \leftarrow \mathcal{B} + \kappa$ 
39:       $\mathcal{B} \leftarrow \min(\mathcal{B}, \mathcal{B}_{\text{max}})$ 
40:    else if Judgment = BAD then
41:       $\alpha(r^*) \leftarrow 1; \mathcal{B} \leftarrow \max(0, \mathcal{B} - 1)$ 
42:    else
43:       $\mathcal{B} \leftarrow \max(0, \mathcal{B} - 1)$ 
44:    end if
45:  else if  $\mathcal{V}_t^T \neq \emptyset$  then
46:    for all  $r_i \in \mathcal{V}_t^T$  compatible with the attribution target do
47:      if Judgment = GOOD then
48:         $\beta(r_i) \leftarrow 1$ 
49:      else if Judgment = BAD then
50:         $\alpha(r_i) \leftarrow 1$ 
51:      else if  $Q > 0$  then
52:         $\beta(r_i) \leftarrow \eta$ 
53:      else if  $Q < 0$  then
54:         $\alpha(r_i) \leftarrow \eta$ 
55:      end if
56:    end for
57:  else if Judgment = BAD then
58:     $h_{\text{new}} \leftarrow \text{GENERATETYPEDRULE}(L_{\text{HCA}}, L_{\text{PRJ}}, \text{evidence})$ 
59:    Merge evidence into an equivalent rule; else insert  $h_{\text{new}}$  as Beta(1, 1)
60:  end if
61:  Phase 6: Bookkeeping
62:   $\text{CONSISTENCYCHECK}(\mathcal{K})$ 
63:  Update  $(\mathcal{S}, H)$ 
64:  if  $t \bmod N = 0$  then
65:    Promote/deprecate rules via  $(\theta_{\text{low}}, \theta_{\text{high}})$ 
66:  end if
67:   $t \leftarrow t + 1$ 
68: end while
69: if terminal score  $\geq$  success threshold then
70:   $\mathcal{T}_g \leftarrow \mathcal{T}_g \cup \{\text{trajectory}\}$ 
71: end if
72: return  $\mathcal{K}$ 

```
