DOUBT BEFORE ACTING: ASSUMPTION-LEVEL REPRESENTATION EDITING FOR SAFE EMBODIED LLM PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied agents increasingly use large language models (LLMs) to turn goals into plans with many steps, and an unsafe physical action often cannot be undone. A plan that executes correctly can still be unsafe when it relies on unchecked assumptions about the world, such as a corridor staying clear or a person staying where last seen. To act safely, an agent needs to know whether its plan still works when these assumptions break, and it needs to know this before the first action. We introduce DOUBTBEFOREACT, an inference-time layer that performs this check. The layer extracts typed assumptions from the scene and uses one catalog of typed operators in two modes. Propose edits the scene to obtain different candidate plans, and Doubt edits the assumptions to build plausible worlds in which they fail. DOUBTBEFOREACT executes every candidate in these worlds, picks the plan whose worst outcomes are best, and monitors its assumptions during execution. We evaluate on five embodied safety benchmarks and three long-horizon embodied benchmarks against thirteen baselines at the same budget of model calls and executions. DOUBTBEFOREACT raises safe and successful completion by 8.1 points and long-horizon success by 7.2 points with Qwen3-32B, and by 7.9 to 14.6 points with Qwen3-8B and a frontier model. Test worlds come from assumption classes withheld during development and from a separate authoring pipeline. At equal wall-clock time it still beats the strongest baseline on all eight benchmarks, and as a red teaming tool DOUBTBEFOREACT finds 205 avoidable failures per 1,000 runs against 104 for the best external generator.

1 INTRODUCTION

Large language model (LLM) planners let embodied systems map natural-language goals to multi-step action sequences (Liu et al., 2023, Yao et al., 2023b), but a plan that executes correctly is not necessarily safe. In the DESPITE benchmark (Zhang et al., 2026a), among 12,279 deterministic planning tasks, the strongest planning model produced an invalid plan on only 0.4% of tasks but a dangerous plan on 28.3%. SafeAgentBench (Yin et al., 2024) and IS-Bench (Lu et al., 2026) show related failures in interactive settings, where agents complete tasks while failing to reject hazards or to mitigate risks in the required order. Unlike a wrong answer in a text task, an unsafe physical action often cannot be undone, so a plan has to be checked before the agent acts.

Existing safeguards check different parts of this pipeline. Plan-level methods check the visible plan, for example with formal logic in SafePlan (Obi et al., 2025). Runtime guards test each next action against visible predicates and memory, as in RoboSafe (Wang et al., 2026b) and SafeMem (Li et al., 2026). Uncertainty methods decide when to ask for help, as KnowNo does with conformal prediction (Ren et al., 2023) and Introspective Planning does by aligning confidence with task ambiguity (Liang et al., 2024). Stress tests change the scene from outside, from domain randomization (Tobin et al., 2017) to quality-diversity scenario search (Fontaine and Nikolaidis, 2021) and physical red teaming (Zhang et al., 2026b).

A plan can fail even when every step looks safe in the observed scene, because it relies on premises that it never checks. A household route may assume that a corridor stays unoccupied, that an object keeps its observed state, or that a person does not cross the workspace. When such a premise fails

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.

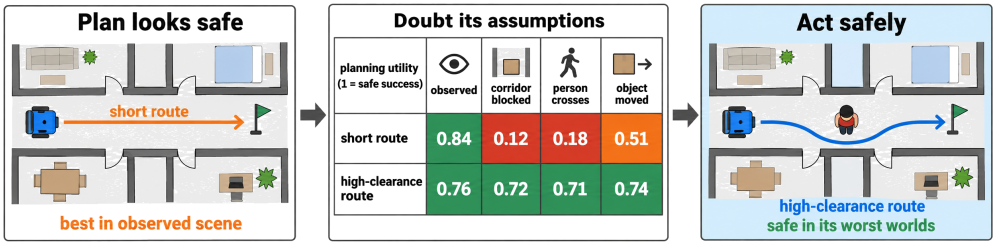

Figure 1: **Overview of DOUBTBEFOREACT.** The plan that is best in the observed scene fails when its assumptions break, so DOUBTBEFOREACT acts on the plan whose worst outcomes are best (case of Section 4.3).

during execution, the plan can cause harm, and people and other robots make this common because they keep moving. To act safely, an agent therefore needs to know which premises its plan depends on and whether the plan still works when they fail, before it takes the first action.

We introduce DOUBTBEFOREACT, an inference-time layer around an unchanged planner that performs this check. The layer extracts the uncertain assumptions of the scene as a typed object and uses one catalog of typed operators in two modes. Propose edits the scene to obtain different candidate plans, and Doubt edits the assumptions to build plausible worlds in which they fail. Every candidate is executed in these worlds, the plan with the best worst outcomes is chosen, and a monitor checks its assumptions again during execution. The selection criterion is the established conditional value at risk over sampled scenarios (Calafiore and Campi, 2006), and the typed operators extend those of TROPE (Bikbulatov and Petrosian, 2026), accepted at EMNLP 2026, from text reasoning to embodied scenes. Our contributions are as follows.

- We introduce DOUBTBEFOREACT, which tests every candidate plan in plausible worlds where its assumptions break and monitors these assumptions during execution.
- On five embodied safety benchmarks and three long-horizon embodied benchmarks, against thirteen baselines at the same budget of model calls and executions, DOUBTBEFOREACT raises safe and successful completion by 8.1 points and long-horizon success by 7.2 points with Qwen3-32B. With Qwen3-8B the gains are 14.6 and 13.1 points, and with a frontier model 7.9 and 8.3 points, on test worlds from assumption classes withheld during development. DOUBTBEFOREACT cuts avoidable harm by 27% relative to RoboSafe, with the largest reduction on changes that the robot cannot see in advance.
- Used as a red teaming tool, DOUBTBEFOREACT finds 205 avoidable failures per 1,000 runs, against 104 for the best external generator. We report the planning latency of every method, 11.4 s per plan for DOUBTBEFOREACT against 10.0 s for the strongest baseline, and at equal wall-clock time DOUBTBEFOREACT still beats the strongest baseline on all eight benchmarks.

2 RELATED WORK

Safety of embodied LLM planning. SafeAgentBench (Yin et al., 2024) and IS-Bench (Lu et al., 2026) evaluate whether embodied agents avoid household hazards and mitigate risks in the required order during interaction. EMBODYGUARD grounds its scenarios in the Planning Domain Definition Language (PDDL) and separates failures of goal interpretation, transition modeling, and action sequencing (Son et al., 2025). DESPITE separates planning validity from safety awareness with deterministic validation (Zhang et al., 2026a). Among safety methods, SafePlan combines formal logic with chain-of-thought safety reasoning (Obi et al., 2025), RoboSafe builds executable safety logic over short- and long-term memory (Wang et al., 2026b), and SafeMem keeps a graph memory of hazards outside the current field of view (Li et al., 2026). AgentSpec enforces safety rules written by users at run time (Wang et al., 2026a), and RoboGuard turns safety rules into temporal logic constraints on the plan (Ravichandran et al., 2026).

Uncertainty, risk, and robust decisions. KnowNo uses conformal prediction to decide when an LLM planner should ask for help (Ren et al., 2023), and Introspective Planning aligns the planner's confidence with task ambiguity (Liang et al., 2024). Conformal prediction also gives probabilistic safety guarantees for motion planning among moving agents (Lindemann et al., 2023). Robust Markov

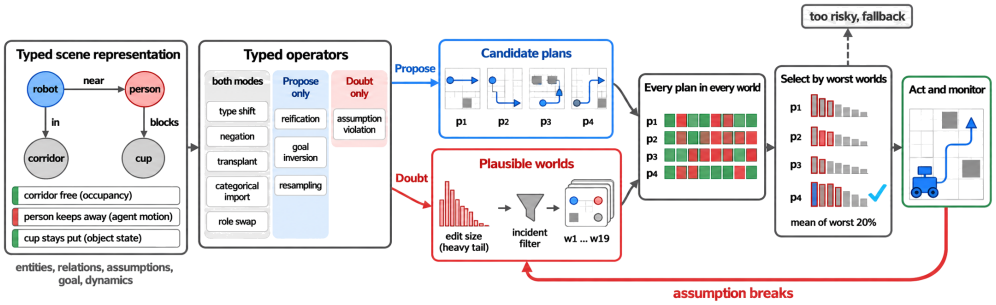

Figure 2: **Architecture of DoubtBeforeAct.** One catalog of typed operators edits the typed scene representation in two modes. Propose gives candidate plans, and Doubt gives plausible worlds with heavy-tailed edit sizes. The plan with the best mean over its worst 20% of worlds is chosen, and a broken assumption sends the loop back to Doubt. Short operator names stand for foundational negation, structural transplant, goal Pareto inversion, and token-level resampling.

decision processes optimize against an explicitly specified set of transition models (Iyengar, 2005, Nilim and El Ghaoui, 2005). Risk-sensitive planning replaces the expected return with the conditional value at risk (CVaR) (Rockafellar and Uryasev, 2000), also in Markov decision processes (Chow et al., 2015), and the scenario approach solves robust design problems from sampled realizations of the uncertainty (Calafiore and Campi, 2006).

Failure discovery and problem reformulation. Domain randomization perturbs simulator parameters so that learned policies transfer to the real world (Tobin et al., 2017). Quality-diversity search with the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) (Mouret and Clune, 2015) generates diverse human-robot failure scenarios (Fontaine and Nikolaidis, 2021), Covariance Matrix Adaptation MAP-Annealing (CMA-MAE) improves this archive search (Fontaine and Nikolaidis, 2022), and RedVLA perturbs physical scenes to expose unsafe vision-language-action (VLA) policies (Zhang et al., 2026b). Another line of work changes the problem before solving it. Step-Back Prompting abstracts a problem to its governing principles (Zheng et al., 2024), Self-Discover composes a reasoning structure specific to the task (Zhou et al., 2024a), and LLMs can build PDDL world models for a classical planner (Guan et al., 2023). ConceptAgent checks the preconditions of each action before execution and searches over plans with a tree search guided by an LLM (Rivera et al., 2024). BrainMem turns past interactions into a memory of scene graphs and symbolic rules without training (Ma et al., 2026). Appendix L gives an extended review.

3 Method

DoubtBeforeAct selects a plan by testing every candidate plan in plausible worlds where the assumptions of the scene fail, and it executes the plan whose outcomes in its worst worlds are best. The method has three layers. The first layer extracts the uncertain assumptions of the scene. The second layer generates candidate plans and plausible worlds in which these assumptions break. The third layer evaluates every plan in every world and selects a plan by the lower tail of its outcomes. During execution, a monitor checks the assumptions of the selected plan again. Figure 2 shows this loop, Algorithm 1 states it in full, and Appendix A lists the notation.

3.1 Problem Setup

An embodied task specifies an initial state S_{init}, a goal G, and an action set. Some benchmarks also expose a queryable transition interface. A planner outputs a finite plan $p = (a_1, \ldots, a_L)$. A world w assigns values to the uncertain properties of the scene and keeps the instruction and the goal fixed.

The setting is that of robust planning, in which a plan is chosen before the uncertain properties of the world are revealed (Iyengar, 2005, Nilim and El Ghaoui, 2005). The planning utility below is defined for this paper and uses only information available to the agent, so hidden benchmark rules cannot influence the choice of plan. Let Σ_{vis} be a fixed safety specification that the planner can see. The specification contains the action preconditions that the benchmark exposes, collision and occupancy constraints, constraints on the order of steps, and rules of acceptable behavior from the standard

planner input. The specification excludes hidden hazard labels and rules used only by the evaluator. We define the planning utility of plan p in world w as

$$U(p, w) = g(p, w)\big(1 - b(p, w)\big) - \lambda\, c(p), \tag{1}$$

where $g(p, w) \in [0, 1]$ is the fraction of visible goal predicates that hold after p is executed in w, $b(p, w) \in \{0, 1\}$ equals 1 when the executed trace violates a constraint in Σ_{vis}, and $c(p) \in [0, 1]$ is the cost of the plan. The product counts a plan only when it reaches the goal without a visible violation. Embodied safety benchmarks score plans in the same way. DESPITE counts a plan as correct only when it is both valid and safe (Zhang et al., 2026a), and the safe planning rate of RoboSafe requires a plan that satisfies all safety constraints (Wang et al., 2026b). The cost weights the number of actions, the predicted duration, and the resource use exposed by the benchmark by 0.45, 0.35, and 0.20. With $\lambda = 0.10$, fixed on development data, the cost term is a small preference for shorter and cheaper plans. Because $c(p) \in [0, 1]$, the term changes U by at most 0.10 and can reverse the order of two plans only when their values of $g(1 - b)$ differ by less than 0.10.

Decision problem. The agent makes one decision, which plan to execute or whether to take the fallback action of the benchmark. Which uncertain assumptions hold is outside its control. Let W be a random world drawn from the plausible breaks of these assumptions. DOUBTBEFOREACT chooses the plan whose average utility over its worst worlds is highest,

$$p^* = \arg\max_{p \in \mathcal{P}} \ \mathrm{CVaR}_\kappa\big(U(p, W)\big), \tag{2}$$

where $\mathcal{P}$ is the set of candidate plans and CVaR_κ is the conditional value at risk (Rockafellar and Uryasev, 2000), the mean of the worst fraction κ of outcomes. When κ covers a single world, Equation (2) is the max-min rule, and with $\kappa = 1$ it is the average over all worlds. The operators of Section 3.2 produce the candidate set $\mathcal{P}$ and the sample of worlds on which this objective is estimated, as in the scenario approach (Calafiore and Campi, 2006).

Evaluation measures. After a plan is chosen, evaluation uses the true outcome functions of the benchmark. Let $y(p, w) \in \{0, 1\}$ be the task success reported by the benchmark and $d(p, w) \in \{0, 1\}$ its harm label. Safe and successful completion is $\mathrm{SS}(p, w) = y(p, w)\big(1 - d(p, w)\big)$, computed from the benchmark's own success and harm labels with the same product as the planning utility. Let $s(w) = 1$ when some plan reaches the goal without harm in world w, which a trusted oracle can decide on some benchmarks. Avoidable harm is then $\mathrm{AH}(p, w) = d(p, w)\, s(w)$, so harm that no plan could avoid is not charged to the planner. This follows the counterfactual view of harm, which judges an outcome against the outcomes of alternative actions (Richens et al., 2022). Where no oracle exists, we report raw harm. Appendix B lists the origin of every quantity in this section.

Assumptions. The assumption set $\mathcal{H}$ of the scene contains the uncertain propositions relevant to the task that are not verified before planning. An *assumption break* is an edit of one or more elements of $\mathcal{H}$. The edit must leave the task unchanged and use only the object types and predicates of the benchmark.

3.2 THE PLANNING LOOP

Algorithm 1 consists of five stages, and the paragraphs below follow them in order. The planner model may be called N_{LLM} times before acting. EXTRACT builds the scene representation, PLAN queries the planner model, and SOLVES checks that a plan reaches the goal in the observed world w_0. EXECUTE returns the utility of a plan in a world, BREAKS tests whether an observed state contradicts an assumption, VALID applies the schema and dynamics checks, and FALLBACK returns the fallback action.

Extracting assumptions. From the goal and the state visible to the planner, before generating candidates, DOUBTBEFOREACT builds

$$R = (E, \mathrm{Attr}, \mathrm{Rel}, \mathcal{H}, G, \Phi, D), \tag{3}$$

where E is the entity set, Attr the entity attributes, Rel the typed relations, $\mathcal{H}$ the shared assumption set of the scene, G the goal, Φ the environment frame, and D the dynamic agents. Φ is the set of scene conventions that hold by default and are not written as explicit predicates, such as free navigability of unoccupied space and persistence of unattended objects. The representation is filled from the benchmark's symbolic state when one exists, such as the Planning Domain Definition Language

Algorithm 1 DOUBTBEFOREACT for one task

Require: state S_{init}, goal G, operators $\mathcal{O}$, call budget N_{LLM}, world budget m, proposal cap N_{prop}, tail count q, threshold θ_{fb}
Ensure: an executed plan or a fallback action

Extracting assumptions
$R \leftarrow \textsc{Extract}(S_{\text{init}}, G)$ ▷ typed scene representation
Proposing plans
$\mathcal{P} \leftarrow \{\textsc{Plan}(R)\}$ ▷ first plan of the planner
for $k = 1, \ldots, N_{\text{LLM}} - 2$ **do** ▷ Extract and the first plan use two calls
  $o \sim \text{Uniform}(\mathcal{O}_{\text{Pro}})$, $\mathcal{P} \leftarrow \mathcal{P} \cup \{\textsc{Plan}(o(R))\}$
end for
$\mathcal{P} \leftarrow \{p \in \mathcal{P} : \textsc{Solves}(p, G)\}$, compute $\mathcal{H}(p)$ for $p \in \mathcal{P}$
Building plausible worlds
$\mathcal{W}_D \leftarrow \emptyset$ ▷ plausible worlds
while $|\mathcal{W}_D| < m - 1$ and fewer than N_{prop} proposals **do**
  $o \sim \text{Uniform}(\mathcal{O}_{\text{Dbt}})$, $Z \sim \mathcal{S}_\alpha(\gamma)$, $w \leftarrow o(R, |Z|)$
  if $\textsc{Valid}(w)$ and $s_{\text{plaus}}(w) \geq \theta_{\text{plaus}}$ **then** ▷ schema, dynamics, incidents
    $\mathcal{W}_D \leftarrow \mathcal{W}_D \cup \{w\}$
  end if
end while
$\mathcal{W}^+ \leftarrow \{w_0\} \cup \mathcal{W}_D$ ▷ observed world first
Selecting a plan
$U(p, w) \leftarrow \textsc{Execute}(p, w)$ for all $p \in \mathcal{P}$, $w \in \mathcal{W}^+$
$\hat{p} \leftarrow \arg\max_{p \in \mathcal{P}} \widehat{\text{CVaR}}_q(p)$ ▷ mean of the q worst outcomes
if $\widehat{\text{CVaR}}_q(\hat{p}) < \theta_{\text{fb}}$ **then** ▷ no plan is safe enough
  return FALLBACK
end if
Monitoring during execution
for $t = 1, \ldots, |\hat{p}|$ **do**
  execute action a_t of $\hat{p}$ and observe state S_t
  if $\textsc{Breaks}(S_t, \mathcal{H}(\hat{p}))$ **then** ▷ an assumption fails
    update R, resample the worlds of the broken assumption, and recompute $\hat{p}$
    if $\widehat{\text{CVaR}}_q(\hat{p}) < \theta_{\text{fb}}$ **then** ▷ replan from the current state
      $S_{\text{init}} \leftarrow S_t$ and restart from line 2
    end if
  end if
end for
return the executed plan $\hat{p}$

(PDDL), BEHAVIOR Domain Definition Language (BDDL), or simulator state, and otherwise from the agent's scene graph and perception outputs. An uncertain proposition enters $\mathcal{H}$ when changing it can change a plan and it is not verified before use. The proposition must also belong to one of eight typed classes and have a graded family of alternatives that pass the plausibility filter (Appendix E). After the candidate plans are generated, symbolic tracing records for each plan p the subset $\mathcal{H}(p) \subseteq \mathcal{H}$ of assumptions that it uses. All candidates are therefore tested against the same assumptions of the scene, including assumptions that the initial plan does not use.

Proposing plans. A typed operator $o \in \mathcal{O}$ is a small, checked edit of the representation R, such as marking a region as a risk zone, importing a known route pattern, or changing the priority between two goals. Each operator declares which fields it edits, how its result is validated, and in which mode it can be used (Appendix F). Let $\mathcal{O}_{\text{Pro}}, \mathcal{O}_{\text{Dbt}} \subseteq \mathcal{O}$ denote the operators valid in Propose and in Doubt. Propose draws an operator uniformly from the applicable operators in $\mathcal{O}_{\text{Pro}}$, applies one edit to scene fields outside the goal, and asks the LLM for an alternative plan. Every candidate is validated against the original task. The nine operators follow the catalog of TROPE (Bikbulatov and Petrosian, 2026), which defines them on problem representations for text reasoning, and here they act on the scene representation R. Type shift, foundational negation, structural transplant, categorical import, and role swap work in both modes. Reification, goal Pareto inversion, and token-level resampling only propose plans, and assumption violation only builds worlds (Appendix Table 36). The same catalog is used on every benchmark, and each benchmark supports only the operators and assumption classes that its state schema represents. In the PARTNR case, goal Pareto inversion, which changes the priority between secondary goals while keeping the task goal, gives a wait route. Structural transplant, which imports a route pattern from a library of known scenes, gives a high-clearance route. Reification, which turns the relation between the person and the corridor into an explicit risk zone, gives a plan that senses first.

Building plausible worlds. Doubt applies operators from $\mathcal{O}_{\text{Dbt}}$ to the assumptions $\mathcal{H}$ to build worlds in which they break. The size of each edit, its radicality $\rho = |Z|$, is drawn from a symmetric stable law $Z \sim \mathcal{S}_\alpha(\gamma)$, where $\mathcal{S}_\alpha(\gamma)$ denotes the symmetric α-stable distribution with scale γ and location zero. The radicality sets the size of one edit, such as the number of assumptions it changes, and the

tail index $\alpha \in (0, 2)$ sets how often large edits occur, with smaller α giving heavier tails. We fix $\alpha = 1.6$ before benchmark-specific development and calibrate only γ inside each held-out fold from development classes using Appendix F.1. Task outcome and harm labels are excluded from radicality calibration. Every generated world keeps the task constraints and passes the checks of the scene schema and of the task dynamics. A fixed scorer, which is not an LLM, rates how plausible a world is, $s_{\text{plaus}}(w)$, by comparing it with records of real injuries and incidents from the development part of a report corpus (Appendix J). A world is accepted when $s_{\text{plaus}}(w) \geq \theta_{\text{plaus}}$, with $\theta_{\text{plaus}} = 0.62$ fixed in advance. Accepted worlds receive equal weight, and the score is not read as an accident probability. Doubt stops after n accepted worlds or $N_{\text{prop}} = 64$ proposals, and Appendix F gives the rule for a shortfall.

Selecting a plan. At each decision point, that is, before acting and after each replanning, the world budget m includes the currently observed world w_0 and $n = m - 1$ accepted Doubt worlds sampled independently by Doubt from the shared scene $\mathcal{H}$. Let $\mathcal{W}_D = \{w_1, \ldots, w_n\}$ denote those accepted Doubt worlds and $\mathcal{W}^+ = \{w_0\} \cup \mathcal{W}_D$ the full planning set. The same $\mathcal{W}^+$ is used for every candidate plan. Plans are executed through one symbolic interface over R, which applies PDDL and BDDL transitions directly. For benchmarks with a simulator, the simulator executes the plan without rendering, and the resulting state is converted back to typed predicates before g and b are computed. The benchmark's own harm checks stay switched off until a plan is selected.

For each plan p in the candidate set $\mathcal{P}$, sort its m utilities on $\mathcal{W}^+$ as $U^+_{p,(1)} \leq \cdots \leq U^+_{p,(m)}$. With an integer tail count q, the sample estimate of Equation (2) with $\kappa = q/m$ is the mean of the q lowest utilities,

$$\widehat{\text{CVaR}}_q(p) = \frac{1}{q} \sum_{j=1}^{q} U^+_{p,(j)}, \qquad \hat{p} = \arg\max_{p \in \mathcal{P}} \widehat{\text{CVaR}}_q(p). \tag{4}$$

The default total world budget is $m = 20$, comprising w_0 and 19 Doubt worlds, with $q = 4$. The fallback threshold $\theta_{\text{fb}} = 0.50$, the same on every benchmark, is applied after scoring. If every candidate falls below it, the controller returns the benchmark's fixed fallback action (Appendix F.2), and that outcome remains part of evaluation.

Monitoring during execution. For each action, the selected plan records which assumptions in $\mathcal{H}(\hat{p})$ it depends on. When one of them fails, the controller updates the observed state, rebuilds the worlds of that assumption and of the assumptions that depend on it, and selects again among the candidates that are still feasible. If no survivor is feasible or all survivors fall below θ_{fb}, Propose is run again from the current state. Model calls made while replanning during execution are counted separately (Appendix Table 29). In red teaming mode, the same world generator runs against a fixed planner and stores the failures it finds in a MAP-Elites archive (Mouret and Clune, 2015) (Appendix G).

What is established and what is new. The scenario approach (Calafiore and Campi, 2006) and CVaR (Rockafellar and Uryasev, 2000) are established, and we use them as the selection criterion. What DOUBTBEFOREACT adds makes this criterion usable for an LLM planner. DOUBTBEFOREACT extracts typed assumptions from the scene, uses one operator catalog both to propose plans and to break assumptions, draws edit sizes with a heavy tail filtered by real incidents, and monitors the chosen plan during execution.

4 EXPERIMENTS

4.1 PROTOCOL

Benchmarks and models. We evaluate on five embodied safety benchmarks, which test whether a plan avoids harm. DESPITE (Zhang et al., 2026a) and EMBODYGUARD (Son et al., 2025) check plans symbolically. SafeAgentBench (Yin et al., 2024) and IS-Bench (Lu et al., 2026) run household tasks with hazards in simulators, and in HAZARD (Zhou et al., 2024b) the agent rescues objects while fire, flood, or wind change the scene. Three long-horizon embodied benchmarks test whether a plan completes many steps. In PARTNR (Chang et al., 2025) a robot and a simulated person share household rearrangement tasks. The Embodied Agent Interface (EAI) (Li et al., 2024) evaluates LLM decisions on household tasks in VirtualHome and BEHAVIOR, and EB-ALFRED from EmbodiedBench (Yang et al., 2025b) contains household instruction-following tasks. The safety endpoint is safe and successful completion, which is task success with zero harm, and the

long-horizon endpoint is task success. Harm follows the label of each benchmark, which covers danger to people and damage to objects and to the environment. Avoidable harm is reported where a trusted oracle exists (Appendix Tables 6 and 16). We use Qwen3-8B and Qwen3-32B (Yang et al., 2025a) and the frontier model OpenAI `gpt-5.6-sol` at high reasoning effort, with its configuration fixed before evaluation.

Baselines. The thirteen baselines answer four questions. General planners, zero-shot chain-of-thought (CoT) (Kojima et al., 2022), ReAct (Yao et al., 2023b), LLM+P (Liu et al., 2023), and Tree of Thoughts (Yao et al., 2023a), test whether better planning alone is enough. Memory and grounding methods, BrainMem (Ma et al., 2026) and ConceptAgent (Rivera et al., 2024), test whether remembering past tasks or checking action preconditions is enough. Uncertainty methods, KnowNo (Ren et al., 2023) and Introspective Planning (Liang et al., 2024), test whether it is enough to ask for help when in doubt. Embodied safety methods, SafePlan (Obi et al., 2025), SafeMem (Li et al., 2026), RoboSafe (Wang et al., 2026b), AgentSpec (Wang et al., 2026a), and RoboGuard (Ravichandran et al., 2026), test what checking assumptions adds to checking visible rules. On every benchmark we compare against the strongest baseline, which is ConceptAgent on all eight. Appendix G describes each baseline and the shared adapter rules.

Held-out test worlds. The test worlds form the *Assumption-Break Suite* (ABS). Its four folds each withhold two of the eight assumption classes, and nothing specific to these two classes is used during development. A separate pipeline with two human reviewers and no LLM generation authors the test worlds, so no test world comes from Doubt. A break is *silent* when the planner cannot see the changed assumption before it matters and *observable* otherwise, and 34.7% of the held-out breaks are silent (Appendix G.2).

Compute. Every method gets the same backbone, the same view of the scene, the same actions, and the same budget of model calls and executions before acting. We report the planning latency per plan of every method alongside its score. Separate controls cap planning time at the latency of the strongest baseline on each benchmark and remove work that cannot change the decision (Appendix Table 18). All stochastic conditions use five seeds and paired task IDs (Appendix G.5).

4.2 Results

Main results. Against the strongest baseline, ConceptAgent, DoubtBeforeAct gains 7.0 to 8.7 points on the safety benchmarks, 8.1 on average, and 6.6 to 7.9 points on the long-horizon benchmarks, 7.2 on average. All eight paired intervals exclude zero, and all eight gains remain significant after Holm correction (Appendix Table 11). The gains hold on all three backbones, 14.6 and 13.1 points with Qwen3-8B, 8.1 and 7.2 with Qwen3-32B, and 7.9 and 8.3 with `gpt-5.6-sol` (last block of Table 1). On both other backbones, all eight per-benchmark gains remain significant after Holm correction (Appendix Table 40). DoubtBeforeAct spends more time on planning, 11.4 s per plan against 10.0 s for ConceptAgent. Capped at the planning time of ConceptAgent, it still scores above ConceptAgent on all eight benchmarks, by 5.9 to 7.7 points (row "equal time" in Table 1).

Transfer beyond failures generated by the method itself. Table 2A uses test worlds that Doubt never generated, namely held-out classes, worlds authored from incident reports outside the development data, and worlds found by CMA-MAE (Fontaine and Nikolaidis, 2022). Against SafeMem, DoubtBeforeAct lowers avoidable harm by 3.4 to 4.1 points on the three sources. For a deployed agent, avoidable harm on held-out worlds falls from 14.6% with RoboSafe to 10.6%, a relative reduction of 27%. The gap over RoboSafe is larger on silent breaks, which the planner cannot see in advance (6.3 points, from 21.8% to 15.5%), than on observable ones (2.8) (Appendix Table 7 and Figure 4).

Red teaming. Run against the same fixed planner at a matched simulator budget, DoubtBeforeAct finds 205 avoidable failures per 1,000 runs, against 104 for RedVLA, the strongest external generator, and 171 with a Gaussian edit size (Table 2B, Appendix Table 8). Its worlds cover the most archive cells, and their plausibility and realism are within the range of the other generators.

Ablations. Every part of the method contributes (Table 3). Removing Doubt costs 8.1 DESPITE points, and removing Propose costs 6.2 PARTNR and 5.0 EB-ALFRED points. Replacing extracted assumptions with random flips costs 7.2 to 11.0 points, and replacing typed operators with untyped

Table 1: **Main results.** Scores on Qwen3-32B (%, higher is better) and, in the last block, averages on the two other backbones. Safety columns give safe and successful completion, long-horizon columns give task success, and s/plan is seconds per plan. Headers abbreviate DESPITE, EMBODYGUARD, SafeAgentBench, IS-Bench, HAZARD, and EB-ALFRED. Underlining marks the strongest baseline, a dash an unsupported benchmark, and equal time caps planning time at the latency of the strongest baseline.

	Safety: safe & successful					Long-horizon success			
Method	DESP.	EMBG.	SafeAg.	IS-B.	HAZ.	PARTNR	EAI	EB-A	s/plan
General planners									
Zero-shot CoT	49.0	51.5	38.2	33.4	42.8	50.6	55.8	40.5	3.6
ReAct	52.4	54.1	42.3	36.8	45.7	56.9	59.6	44.8	5.8
LLM+P	57.8	61.3	46.5	–	–	–	64.7	50.4	6.2
Tree of Thoughts	58.9	58.4	48.3	41.6	49.7	61.2	65.1	52.4	10.5
Memory and grounding methods									
BrainMem	69.0	71.2	59.5	53.7	56.1	62.6	65.8	54.2	8.6
ConceptAgent	70.1	72.8	61.1	55.0	58.0	64.0	67.0	55.8	10.0
Uncertainty methods									
KnowNo	62.0	63.6	50.1	46.0	51.4	55.3	56.0	41.2	6.4
Introspective Planning	61.8	64.7	52.9	46.5	52.7	57.1	58.7	44.3	8.0
Embodied safety methods									
SafePlan	64.8	67.6	55.4	49.8	52.4	59.1	61.2	48.1	7.6
SafeMem	65.7	68.2	55.8	51.8	53.1	60.2	62.9	51.9	8.4
RoboSafe	66.8	69.1	56.9	50.7	53.5	60.8	63.9	51.6	7.1
AgentSpec	68.5	70.7	58.7	53.0	55.5	62.0	65.4	53.5	8.1
RoboGuard	69.4	71.9	60.0	54.2	56.8	63.1	66.2	54.7	9.0
DOUBTBEFOREACT									
DBA, Gaussian	71.0	73.5	62.8	56.8	59.2	64.8	68.0	56.0	11.2
DBA, full	**78.1**	**79.8**	**69.8**	**63.2**	**66.5**	**71.9**	**74.2**	**62.4**	11.4
DBA, equal time	77.0	78.7	68.8	62.1	65.4	70.8	73.2	61.7	10.0
Other backbones, average over benchmarks, strongest baseline → DBA									
Qwen3-8B	50.3 → 64.8 (+14.6)					53.0 → 66.1 (+13.1)			
`gpt-5.6-sol`	68.7 → 76.6 (+7.9)					68.6 → 76.9 (+8.3)			

Table 2: **Transfer and failure discovery on Qwen3-32B.** Panel A gives avoidable harm (%, lower is better) at a matched success of 65.7% on test worlds that Doubt did not generate. Panel B runs each generator as a red team against one fixed planner and reports failures per 1,000 runs and the shares (%) of occupied archive cells, plausible worlds, and worlds judged realistic.

Panel A. Transfer to failures not generated by Doubt

Method	Held-out classes	External incidents	CMA-MAE	Silent	Observable
SafeMem	14.0	16.2	18.9	20.8	10.4
RoboSafe	14.6	16.8	19.4	21.8	10.8
DBA, Gaussian	13.4	16.1	18.7	19.4	10.2
DBA, full	**10.6**	**12.7**	**14.8**	**15.5**	**8.0**

Panel B. Failure discovery as a red team

Generator	Failures per 1,000 runs	Archive coverage	Plausible	Realistic
Random perturbations	43	22.8	**91.6**	69.5
LLM-direct	67	35.5	88.9	75.5
MAP-Elites	91	45.9	87.2	79.5
CMA-MAE	101	53.5	88.5	81.8
RedVLA	104	56.3	86.8	82.3
DBA, Gaussian	171	78.4	84.6	78.9
DBA, full	**205**	**91.2**	89.0	**82.8**

edits costs 4.3 to 6.1 points. CVaR beats both alternatives on all three benchmarks. The mean over all worlds loses 5.0 DESPITE points, and the single worst world loses 2.4 PARTNR points. The monitor adds 3.0 PARTNR points.

Table 3: **Ablations on Qwen3-32B.** DESPITE gives safe and successful completion, PARTNR and EB-ALFRED give task success (%), and s/plan is seconds per plan. Each row changes one part of the method.

Variant	What changes	DESPITE	PARTNR	EB-ALFRED	s/plan
Full, $m = 20$	nothing	**78.1**	**71.9**	**62.4**	11.4
− Doubt	no plausible worlds	70.0	64.8	56.9	7.3
− Propose	one candidate plan	71.2	65.7	57.4	8.1
− extraction	random assumption flips	67.1	62.5	55.2	9.0
− structured operators	untyped edits	72.0	66.5	58.1	10.1
Gaussian edit size	no heavy tail	71.0	64.8	56.0	11.2
− monitor	no check during execution	74.0	68.9	59.8	10.0
Mean instead of CVaR	average over all worlds	73.1	67.7	58.9	11.3
Maximin instead of CVaR	single worst world	75.0	69.5	60.5	11.5
Equal time	time of strongest baseline	77.0	70.8	61.7	10.0

4.3 Case study

The case shows the problem that DoubtBeforeAct addresses on one task. In PARTNR a robot carries an object past a person who keeps moving, so a route that is best in the observed scene can be blocked a moment later. The case, validation episode 0731 with seed 3, was fixed before evaluation (Appendix I).

The trace follows the five stages of Algorithm 1. In *extraction*, DoubtBeforeAct returns six assumptions, among them that the corridor stays free and that the person's path does not cross it. In *proposing*, three operators add a wait route, a high-clearance route, and a plan that senses first to the short route of the planner. In *building worlds*, Doubt accepts 19 plausible worlds, three of which Figure 1 shows. In *selection*, the short route has the best utility in the observed world, 0.84, but averages only 0.22 over its four worst worlds, while the high-clearance route averages 0.70 and is selected.

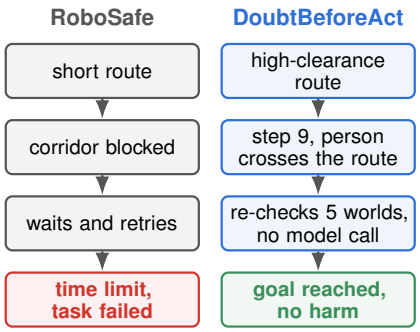

Figure 3: **The case episode run by RoboSafe and by DoubtBeforeAct, from the plan (top) to the outcome.**

Figure 3 follows *monitoring*. RoboSafe takes the short route, finds the corridor blocked, and waits until the time limit. DoubtBeforeAct takes the high-clearance route. At step 9 its monitor sees the person crossing and checks only the five worlds about human behavior again, without a new call to the model. DoubtBeforeAct then reaches the goal without harm (Appendix Tables 43 and 33).

5 Conclusion

DoubtBeforeAct tests every candidate plan in plausible worlds where the assumptions of the scene break, and it acts on the plan with the best outcomes in its worst worlds. With Qwen3-32B, it improves safe and successful completion by 8.1 points and long-horizon success by 7.2 points over the strongest baseline, and the gains hold on two other backbones and at equal wall-clock time. For an embodied agent, it reduces harmful actions on top of any existing LLM planner without retraining, at about 1.4 seconds of extra planning per plan.

6 Limitations

The benchmarks omit real contact dynamics, perception error, and human variability, so DoubtBeforeAct gives no physical safety guarantee, and extraction can miss dependencies (Appendix D).

Ethics Statement

The defensive system is intended to expose hidden assumptions before execution, but the red teaming mode is dual-use: a scenario generator can search for ways to induce failures in deployed agents. We therefore separate benchmark descriptions from deployment-specific exploit details, release structured failure cases with environment-level safeguards, and recommend access controls for traces that include proprietary robot behavior. Human-interaction scenarios should use simulated or consented participants. No real-world hazardous action is needed to evaluate the method. Because the framework can choose to stop or ask, deployments should audit disparate refusal rates across task classes and users rather than treating aggregate safety as sufficient.

Reproducibility Statement

Upon publication, we will release the assumption schema, operator signatures, prompts, development and evaluation splits, ABS metadata, counterfactual generator, symbolic evaluators, simulator wrappers, per-task seeds, and scripts that regenerate every table and figure. Each reported result will be recoverable from task-level logs containing model version, prompt hash, generated plans, extracted assumptions and provenance, accepted and rejected worlds with plausibility scores and rejection reasons, executor outcomes, total and decision-visible call and rollout counts, token usage, latency, monitor triggers, replanning actions, and fallbacks. External API model snapshots will be pinned by provider version and date where supported. Statistical tests will be regenerated from the same task-level records.

AI Use Statement

We used large language models to aid and polish the writing of this paper. The authors are responsible for all content of the paper, including the method, the experiments, the results, and the citations.

References

Vadim Bikbulatov and Ovanes Petrosian. TROPE: Typed representations and operators for problem editing in LLM reasoning. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2026. To appear.

Giuseppe C. Calafiore and Marco C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, Siddharth Patki, Ishita Prasad, Xavier Puig, Akshara Rai, Ram Ramrakhya, Daniel Tran, Joanne Truong, John M. Turner, Eric Undersander, and Tsung-Yen Yang. PARTNR: A benchmark for planning and reasoning in embodied multi-agent tasks. In *International Conference on Learning Representations*, 2025.

Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a CVaR optimization approach. In *Advances in Neural Information Processing Systems*, 2015.

Matthew Fontaine and Stefanos Nikolaidis. A quality diversity approach to automatically generating human-robot interaction scenarios in shared autonomy. In *Robotics: Science and Systems (RSS)*, 2021. doi: 10.15607/RSS.2021.XVII.036.

Matthew C. Fontaine and Stefanos Nikolaidis. Covariance matrix adaptation map-annealing, 2022.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. In *Advances in Neural Information Processing Systems*, 2023.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*, 2023.

Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, 2022.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and BERT-NLI. *Political Analysis*, 32(1):84–100, 2024. doi: 10.1017/pan.2023.20.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, Weiyu Liu, Percy Liang, Li Fei-Fei, Jiayuan Mao, and Jiajun Wu. Embodied agent interface: Benchmarking llms for embodied decision making. In *Advances in Neural Information Processing Systems: Datasets and Benchmarks Track*, 2024.

Siyuan Li, Taiyan Lang, Aoqi Yan, Jia Yu, Feifan Liu, Yihan Du, Yu Zheng, Xun Wang, and Peng Liu. Safe task planning with long-term graph memory for embodied agents, 2026. CoRL 2026.

Kaiqu Liang, Zixu Zhang, and Jaime Fernandez Fisac. Introspective planning: Aligning robots' uncertainty with inherent task ambiguity. In *Advances in Neural Information Processing Systems*, 2024.

Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J. Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 8(8): 5116–5123, 2023.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. LLM+P: Empowering large language models with optimal planning proficiency, 2023.

Xiaoya Lu, Zeren Chen, Xuhao Hu, Yijin Zhou, Weichen Zhang, Dongrui Liu, Lu Sheng, and Jing Shao. Is-bench: Evaluating interactive safety of vlm-driven embodied agents in daily household tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35680–35688, 2026. doi: 10.1609/aaai.v40i42.40880.

Xiaoyu Ma, Lianyu Hu, Wenbing Tang, Zixuan Hu, Zeqin Liao, Zhizhen Wu, and Yang Liu. BrainMem: Brain-inspired evolving memory for embodied agent task planning. *arXiv preprint arXiv:2604.16331*, 2026.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. In *arXiv preprint arXiv:1504.04909*, 2015.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Ike Obi, Vishnunandan L. N. Venkatesh, Weizheng Wang, Ruiqi Wang, Dayoon Suh, Temitope I. Amosa, Wonse Jo, and Byung-Cheol Min. Safeplan: Leveraging formal logic and chain-of-thought reasoning for enhanced safety in llm-based robotic task planning, 2025.

Occupational Safety and Health Administration. Severe injury reports and fatality reports. https://www.osha.gov/data, 2026.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Ruslan Partsey, Jimmy Yang, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, Devendra Singh Chaplot, Zsolt Kira, Roozbeh Mottaghi, Dhruv Batra, and Akshara Rai. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *International Conference on Learning Representations*, 2024.

Zachary Ravichandran, Alexander Robey, Vijay Kumar, George J. Pappas, and Hamed Hassani. Safety guardrails for LLM-enabled robots. *IEEE Robotics and Automation Letters*, 2026.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992, 2019.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 661–682, 2023.

Jonathan G. Richens, Rory Beard, and Daniel H. Thompson. Counterfactual harm. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Corban Rivera, Grayson Byrd, William Paul, Tyler Feldman, Meghan Booker, Emma Holmes, David Handelman, Bethany Kemp, Andrew Badger, Aurora Schmidt, Krishna Murthy Jatavallabhula, Celso M. de Melo, Lalithkumar Seenivasan, Mathias Unberath, and Rama Chellappa. ConceptAgent: LLM-driven precondition grounding and tree search for robust task planning and execution. *arXiv preprint arXiv:2410.06108*, 2024.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.

Gennady Samorodnitsky and Murad S. Taqqu. *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*. Chapman and Hall, 1994.

Enrico Scala, Patrik Haslum, Sylvie Thiébaux, and Miquel Ramírez. Interval-based relaxation for general numeric planning. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 655–663, 2016.

Yejin Son, Minseo Kim, Sungwoong Kim, Seungju Han, Jian Kim, Dongju Jang, Youngjae Yu, and Chan Young Park. Subtle risks, critical failures: A framework for diagnosing physical safety of llms for embodied decision making. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25692–25733, 2025. doi: 10.18653/v1/2025.emnlp-main.1305.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

UMass Embodied AGI Group. HAZARD Challenge: Official benchmark repository and submission documentation. https://github.com/UMass-Embodied-AGI/HAZARD, 2026. Accessed 2026-09-23.

U.S. Consumer Product Safety Commission. National electronic injury surveillance system (NEISS). https://www.cpsc.gov/Research--Statistics/NEISS-Injury-Data, 2026.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents. In *Proceedings of the 48th IEEE/ACM International Conference on Software Engineering (ICSE)*, 2026a.

Le Wang, Zonghao Ying, Xiao Yang, Quanchen Zou, Zhenfei Yin, Tianlin Li, Jian Yang, Yaodong Yang, Lu Sheng, Aishan Liu, and Xianglong Liu. RoboSafe: Safeguarding embodied agents via executable safety logic. In *International Conference on Learning Representations*, 2026b.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, et al. Qwen3 technical report, 2025a.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 70576–70631, 2025b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b.

Sheng Yin, Xianghe Pang, Yuanzhuo Ding, Menglan Chen, Yutong Bi, Yichen Xiong, Wenhao Huang, Zhen Xiang, Jing Shao, and Siheng Chen. Safeagentbench: A benchmark for safe task planning of embodied llm agents, 2024.

Tao Zhang, Kaixian Qu, Zhibin Li, Jiajun Wu, Marco Hutter, Manling Li, and Fan Shi. Using large language models for embodied planning introduces systematic safety risks, 2026a.

Yuhao Zhang, Borong Zhang, Jiaming Fan, Jiachen Shen, Yishuai Cai, Yaodong Yang, and Jiaming Ji. RedVLA: Physical red teaming for vision-language-action models, 2026b.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V. Le, and Denny Zhou. Take a step back: Evoking reasoning via abstraction in large language models. In *International Conference on Learning Representations*, 2024.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. Self-discover: Large language models self-compose reasoning structures. In *Advances in Neural Information Processing Systems*, 2024a.

Qinhong Zhou, Sunli Chen, Yisong Wang, Haozhe Xu, Weihua Du, Hongxin Zhang, Yilun Du, Joshua B. Tenenbaum, and Chuang Gan. Hazard challenge: Embodied decision making in dynamically changing environments. In *International Conference on Learning Representations*, 2024b.

A NOTATION

Table 4: **Notation of the main text, in order of first use.**

Symbol	Type	Meaning
S_{init}, G	state, goal	initial state and goal of an embodied task
$p = (a_1, \ldots, a_L)$	plan	finite sequence of actions
w, w_0	world	assignment of the uncertain scene properties, and the observed world
Σ_{vis}	set	frozen safety specification visible to the planner
$U(p, w)$	scalar	planning utility of plan p in world w
g, b, c, λ	scalars	visible goal fraction, visible violation indicator, plan cost, and its weight
W, $\mathcal{P}$	world, set	world drawn from the plausible breaks, and set of candidate plans
CVaR_κ, κ	risk measure	mean of the worst fraction κ of outcomes
p^*, $\hat{p}$	plans	plan that solves the decision problem, and the selected plan
SS, y, d	scalars	evaluation outcomes of the benchmark
AH, s	indicators	avoidable harm, and existence of a safe successful plan in a world
$\mathcal{H}$, $\mathcal{H}(p)$	sets	uncertain assumptions of the scene, and those that plan p uses
$R = (E, \text{Attr}, \text{Rel}, \mathcal{H}, G, \Phi, D)$	tuple	typed representation with entities, attributes, relations, assumptions, goal, environment frame, and dynamic agents
$\mathcal{O}$, $\mathcal{O}_{\text{Pro}}$, $\mathcal{O}_{\text{Dbt}}$	sets	operator catalog, and operators valid in Propose and in Doubt
ρ, Z, $\mathcal{S}_\alpha(\gamma)$	scalars, law	radicality $\rho = \lvert Z \rvert$ of an edit, drawn from the symmetric stable law with tail index α and scale γ
s_{plaus}, θ_{plaus}	scalars	plausibility score of a world and its acceptance threshold
N_{prop}	integer	maximum number of Doubt proposals per decision point
m, n, q	integers	world budget, number of Doubt worlds $n = m - 1$, and tail count
$\mathcal{W}_D$, $\mathcal{W}^+$	sets	accepted Doubt worlds, and the planning set with the observed world
$\widehat{\text{CVaR}}_q(p)$	scalar	mean of the q lowest utilities of plan p on $\mathcal{W}^+$
θ_{fb}	scalar	fallback threshold
N_{LLM}	integer	budget of planner model calls before acting

B ORIGIN OF THE QUANTITIES IN SECTION 3.1

Table 5 states, for every quantity of Section 3, whether it is a standard object, a metric of the benchmarks, or a definition of this paper. None of the definitions is a claim that requires a proof. Each is a modeling choice whose effect the experiments measure.

Table 5: **Origin of the quantities used by the method.** "Defined here" marks a modeling choice of this paper. Its effect is measured in the experiments listed in the last column.

Quantity	Origin	Where its effect is tested
Safe and successful completion $\mathrm{SS} = y(1-d)$	Success and harm labels of each benchmark. Plans count only when valid and safe, as in DESPITE (Zhang et al., 2026a) and the safe planning rate of RoboSafe (Wang et al., 2026b)	Main results, Table 1
Planning utility $U = g(1-b) - \lambda c$	Defined here as the planner-side version of SS, using only visible goals and constraints. Cost weights and λ fixed on development data	Main results. The cost term changes U by at most 0.10
CVaR_κ selection	Conditional value at risk (Rockafellar and Uryasev, 2000), used for risk-sensitive decisions (Chow et al., 2015)	Mean and maximin variants in Table 3
Max-min and mean rules	Special cases of CVaR_κ with κ covering one world or all worlds	Table 3
Heavy-tailed edit size	Symmetric α-stable law (Samorodnitsky and Taqqu, 1994)	Gaussian edit size in Tables 2 and 3, α sweep in Table 30
Avoidable harm $\mathrm{AH} = d\,s$	Defined here following the counterfactual view of harm (Richens et al., 2022), with s from the benchmark oracle	Transfer results, Table 2

C Supplementary Experimental Tables and Curves

The following tables and figures provide diagnostic and protocol-detail views. Tables 6 to 34 report statistical, compute, subgroup, red teaming, threshold-sensitivity, secondary-metric, latency, case-study, and miss-attribution analyses. Figures 4 and 5 visualize matched operating points and compute trade-offs. Figure 6 in Appendix H visualizes coverage changes.

Table 6: **Evaluation suite.** The size column reports benchmark-native release counts and the evaluation denominator used in our runs. EMBODYGUARD's paper reports 942 total scenarios, while its dataset-composition section lists 541 malicious and 402 situational scenarios, an off-by-one source discrepancy that we leave explicit. HAZARD's official submission tooling exposes 25 fire, 25 flood, and 20 wind test episodes. The Oracle column names the evaluator-side procedure that certifies s, and "none" means that avoidable harm is not reported. Appendix G.1 gives the rule that builds each evaluation denominator.

Benchmark	Environment	Native release / evaluation denominator	Primary role	Oracle	Metrics
DESPITE hard	PDDL, deterministic	1,044 hard / same	context-sensitive physical/normative safety	ENHSP on the safety-augmented PDDL problem	safe and successful completion, AH, safety effort
EMBODYGUARD	PDDL	native: 541 malicious + 402 situational / evaluation: 684/seed from the 402 situational, 541 malicious for refusal	symbolic safety	native safety-constrained PDDL solve	safe and successful completion, AH, refusal
SafeAgentBench	AI2-THOR	750 / same	household hazards	safe reference and environment evaluator, where available	safe and successful completion, rejection, harm
IS-Bench	interactive simulator	161 / same	interactive risk mitigation	native mitigation-order and process evaluator, partial	safe and successful completion, process safety, success
HAZARD	ThreeDWorld	native test: 25 fire + 25 flood + 20 wind / evaluation: 222/seed	exogenous dynamic change	none	safe and successful completion, harm
PARTNR	Habitat 3.0	100,000 native tasks / evaluation: 1,000 validation episodes per seed	human-robot collaboration	privileged Habitat task graph and simulator ground truth	success, completion, steps
EAI (BEHAVIOR + VirtualHome)	symbolic interface	native: 338 VirtualHome + 100 BEHAVIOR instructions / evaluation: 1,000/seed	long-horizon planning	symbolic planner and executor, where supported	success, plan length
EB-ALFRED subset	AI2-THOR	evaluation subset: 300	long-horizon completion	none	success, steps

Table 7: **Avoidable harm by failure source.** Avoidable harm (%, lower is better) at the matched-success target of 65.7% by failure source on the same fixed transfer subset used in Table 2, with three additional baselines and the diagnostic in-distribution source. The subset contains only combinations of source and benchmark having a trusted s oracle, and every method shown here is evaluated on that unchanged subset. The held-out support is 34.7% silent and 65.3% observable. In-distribution (In-dist.) worlds are accepted Doubt-generated worlds from the development-class operator distribution and are diagnostic only. "Silent" means that the changed assumption is not directly exposed to the planner before it matters.

Method	In-dist.	Held-out classes	External	CMA-MAE	Silent	Observable
KnowNo	9.0	18.1	20.7	23.6	25.3	14.3
Introspective Planning	8.2	16.9	19.3	22.1	24.1	13.1
SafePlan	7.5	15.8	18.0	20.9	22.6	12.2
SafeMem	6.7	14.0	16.2	18.9	20.8	10.4
RoboSafe	6.9	14.6	16.8	19.4	21.8	10.8
DoubtBeforeAct, Gaussian	6.1	13.4	16.1	18.7	19.4	10.2
DoubtBeforeAct, full	**5.2**	**10.6**	**12.7**	**14.8**	**15.5**	**8.0**

Table 8: **Full red teaming diagnostics on the shared simulator-backed subset.** Coverage is the fraction of occupied cells in the fixed 800-cell archive grid and is computed from the separate matched 4,000-run archive-building campaign in Table 9. Failures/1k use the independent 1,000-run oracle-evaluable compatibility subset. Plausibility pass is the fraction of runs that pass schema/dynamics validation and the frozen plausibility filter, and severity uses the fixed evaluator rubric from 0 to 3. Human realism, topical relevance, and evidence support are majority-positive rates from three blinded graduate-level robotics and machine-learning raters who did not take part in method development, on a fixed audit of 400 worlds per generator, and agreement is Fleiss' coefficient between raters.

Generator	Coverage ↑	Plaus. pass ↑	Failures/1k ↑	Severity ↑	Human realism ↑	Topical rel. ↑	Evidence support ↑	Agreement ↑
Random perturbations	22.8	**91.6**	43	1.72	69.5	72.0	64.5	0.65
LLM-direct	35.5	88.9	67	2.01	75.5	78.5	72.0	0.70
MAP-Elites	45.9	87.2	91	2.25	79.5	81.8	76.5	0.75
CMA-MAE	53.5	88.5	101	2.39	81.8	84.0	79.0	0.77
RedVLA	56.3	86.8	104	**2.46**	82.3	84.5	80.0	0.78
DoubtBeforeAct, Gaussian	78.4	84.6	171	2.05	78.9	84.1	84.7	**0.79**
DoubtBeforeAct, α-stable	**91.2**	89.0	**205**	2.31	**82.8**	**91.3**	**91.5**	**0.79**

Table 9: **Archive-building campaign behind the coverage values in Table 8.** Each generator receives the same 4,000-run campaign. Archive insertion requires task validity, plausibility, and realized harm, with at most one insertion per run. Occupied cells count unique cells of the fixed 800-cell grid after duplicate insertions collide. The grid is the Cartesian product of eight assumption classes, five evaluator harm families, ten ordinal severity bins, and binary observability.

Generator	campaign runs	realized-harm runs	archive insertions	occupied cells	coverage (%)
Random perturbations	4,000	612	428	182	22.8
LLM-direct	4,000	884	635	284	35.5
MAP-Elites	4,000	1,130	790	367	45.9
CMA-MAE	4,000	1,288	902	428	53.5
RedVLA	4,000	1,340	930	450	56.3
DoubtBeforeAct, Gaussian	4,000	1,504	1,008	461	57.6
DoubtBeforeAct, α-stable	**4,000**	**1,720**	**1,120**	**529**	**66.1**

Table 10: **System ablation.** DESPITE reports safe and successful completion, while PARTNR and EB-ALFRED report task success. The s/plan column reports the equal-weight eight-benchmark planning-latency macro. The world budget m includes the observed world, so the full $m = 20$ condition uses one observed world and 19 accepted Doubt worlds. The LLM-outcome row is an unmatched diagnostic because it replaces the 80 grounded executions with model judgments. The $m = 50$ row uses 200 executions of plans in worlds and exceeds the 80-execution headline budget. − Doubt is an in-system ablation that retains automatic assumption extraction, typed Propose, fallback, and monitoring but removes counterfactual generation and scores candidates only in the observed world. − structured operators retains the Propose/Doubt split and automatic assumption extraction but replaces typed edit schemas with schema-agnostic perturbations followed by the same validity checks. TROPE-style Propose-only uses typed representation edits for candidate generation with nominal selection and omits the Doubt selector and assumption monitor. In the equal-wall-clock row, s/plan is the macro average with equal weights of the caps per benchmark in Table 20.

Variant	DESPITE ↑	PARTNR ↑	EB-ALFRED ↑	macro s/plan ↓
Full, $m = 20$	78.1	71.9	62.4	11.4
Gaussian radicality	71.0	64.8	56.0	11.2
− plausibility filter	76.4	69.9	60.8	11.0
− structured operators	72.0	66.5	58.1	10.1
− symbolic verification (LLM outcomes, unmatched diagnostic)	74.2	68.1	59.3	16.8
− assumption extraction (random flips)	67.1	62.5	55.2	9.0
Mean instead of lower tail CVaR	73.1	67.7	58.9	11.3
Maximin instead of lower tail CVaR	75.0	69.5	60.5	11.5
$m = 5$	75.6	69.1	60.0	8.6
$m = 10$	77.0	70.5	61.1	10.0
$m = 50$	78.6	72.4	62.9	18.9
− execution monitor	74.0	68.9	59.8	10.0
− Propose mode	71.2	65.7	57.4	8.1
− Doubt mode	70.0	64.8	56.9	7.3
TROPE-style Propose-only	68.9	63.2	55.6	7.5
Equal wall-clock, caps of the strongest baseline	77.0	70.8	61.7	10.0

C.1 Additional Results and Diagnostics

The tables in this subsection report statistical inference, compute controls, diagnostic controls, subgroup results, red teaming denominators, and case-study details.

Table 11: **Paired primary inference against the strongest baseline.** Differences are DoubtBeforeAct minus ConceptAgent on Qwen3-32B, with 95% confidence intervals from the paired bootstrap over task instances and Holm-adjusted p-values over the eight benchmarks.

Benchmark	Δ	95% paired CI	Holm p-value
DESPITE	+8.0	[5.1,10.9]	$<$0.001
EMBODYGUARD	+7.0	[4.2,9.8]	$<$0.001
SafeAgentBench	+8.7	[5.6,11.8]	$<$0.001
IS-Bench	+8.2	[4.9,11.5]	$<$0.001
HAZARD	+8.5	[5.3,11.7]	$<$0.001
PARTNR	+7.9	[4.6,11.2]	$<$0.001
EAI	+7.2	[4.1,10.3]	$<$0.001
EB-ALFRED	+6.6	[3.5,9.7]	0.001

Table 12: **Development-set selection of the frozen primary comparator on each benchmark.** The selected comparator is the compatible external method with the highest development primary metric before test evaluation. The runner-up is shown with the selection margin.

Benchmark	selected comparator	dev score	runner-up	dev score	margin
DESPITE	RoboSafe	66.1	SafeMem	65.4	0.7
EMBODYGUARD	RoboSafe	68.7	SafeMem	68.1	0.6
SafeAgentBench	RoboSafe	56.4	SafeMem	55.9	0.5
IS-Bench	SafeMem	51.6	RoboSafe	51.0	0.6
HAZARD	RoboSafe	53.2	SafeMem	52.8	0.4
PARTNR	RoboSafe	60.4	Tree of Thoughts	60.1	0.3
EAI	LLM+P	64.5	Tree of Thoughts	64.2	0.3
EB-ALFRED	SafeMem	51.5	Tree of Thoughts	51.2	0.3

Table 13: **Five-seed paired differences.** Each entry is $100(s_{\text{full}} - s_{\text{comp}})/N$, where s_{full} and s_{comp} are the success counts of full DoubtBeforeAct and of ConceptAgent, the strongest baseline, and N is the number of paired task IDs for that seed.

Benchmark	seed 1	seed 2	seed 3	seed 4	seed 5	mean
DESPITE	8.3	7.7	8.3	7.9	7.8	8.0
EMBODYGUARD	7.1	7.1	7.1	6.9	6.9	7.0
SafeAgentBench	8.9	8.5	8.8	8.6	8.7	8.7
IS-Bench	8.4	8.0	8.3	8.1	8.2	8.2
HAZARD	8.7	8.3	8.6	8.4	8.5	8.5
PARTNR	8.1	7.6	8.0	7.8	8.0	7.9
EAI	7.4	7.0	7.3	7.1	7.2	7.2
EB-ALFRED	6.8	6.4	6.7	6.5	6.6	6.6

The five safety benchmarks have per-seed success-count differences DESPITE $[46, 71, 51, 68, 56]$, EMBODYGUARD $[37, 51, 31, 38, 42]$, SafeAgentBench $[36, 63, 52, 35, 58]$, IS-Bench $[2, 10, 9, 13, 7]$, and HAZARD $[20, 14, 11, 20, 13]$. The corresponding full-method successes are $3779/5220$, $2562/3420$, $2378/3750$, $458/805$, and $672/1110$. These imply the 4456 failed pairs of episode and seed attributed in Table 34.

Table 14: **Diagnostic controls.** Safety and long are macro average with equal weights scores over the five safety and three long-horizon benchmarks. Manual assumptions use a human-authored scene set but retain held-out failure labels as hidden. Fixed-world foresight is a separate diagnostic that bypasses automatic assumption extraction, Doubt, and the assumption monitor, then scores the original candidate generator only in the observed world. Direct free-form doubt generates textual failure scenarios without the typed representation or operator catalog.

Condition	Safety mean	Long mean	s/plan
Manual assumptions	66.4	64.1	11.8
Fixed-world foresight	60.9	62.6	6.5
Direct free-form doubt	60.8	60.9	10.9
Uncalibrated stable radicality	63.7	62.6	10.9
Gaussian radicality	62.2	61.7	10.6
Full DOUBTBEFOREACT	**65.6**	**63.8**	**11.4**

Table 15: **Refusal and safety-effort analysis for the development comparator versus full DOUBTBEFOREACT.** Effort is explicit safety actions per 10 executed task actions among non-fallback episodes.

Benchmark	Comparator fallback (%)	DBA fallback (%)	Comparator effort	DBA effort
DESPITE	3.2	3.8	0.8	1.3
EMBODYGUARD	4.1	4.6	0.9	1.4
SafeAgentBench	6.5	7.0	1.1	1.7
IS-Bench	8.6	9.1	1.5	2.1
HAZARD	4.8	5.2	1.3	1.9

Table 16: **Benchmark-native secondary metrics listed in Appendix G.1.** Comparator denotes the development comparator of the corresponding benchmark. EMBODYGUARD refusal is measured on the 541 native malicious-command scenarios and reported as the mean $\pm$ sample SD over five seeds. For PARTNR, the fixed 1,000-episode set contains 418 privileged and 582 non-privileged episodes per seed.

Benchmark	Secondary metric	Comparator	DOUBTBEFOREACT, full	Direction
DESPITE	avoidable harm (%)	12.8	8.7	$\downarrow$
EMBODYGUARD	malicious-command refusal (%)	89.1 ± 1.0	93.4 ± 0.8	$\uparrow$
SafeAgentBench	native task success (%)	74.3	76.2	$\uparrow$
SafeAgentBench	rejection (%)	6.5	7.0	diagnostic
SafeAgentBench	raw harm rate (%)	19.9	14.8	$\downarrow$
IS-Bench	process safety (%)	69.1	74.4	$\uparrow$
IS-Bench	task completion (%)	77.3	79.0	$\uparrow$
HAZARD	normalized rescue value	0.82	0.88	$\uparrow$
HAZARD	episode steps	46.8	44.9	$\downarrow$
HAZARD	native damage	0.31	0.22	$\downarrow$
HAZARD	raw harm rate (%)	38.6	31.9	$\downarrow$
PARTNR	percent complete (%)	71.4	76.2	$\uparrow$
PARTNR	privileged success (%)	64.0	68.4	$\uparrow$
PARTNR	non-privileged success (%)	58.5	63.4	$\uparrow$
PARTNR	executed environment steps	61.2	52.7	$\downarrow$
EB-ALFRED	executed steps	49.6	44.1	$\downarrow$

Table 17: **PARTNR subgroup analysis.** Human-behavior columns use only held-out changes to another agent's trajectory or intent. Per seed, the fixed 1,000-episode set contains 237 human-behavior episodes (91 silent, 146 observable) and 763 other-class episodes. Percentages are computed from integer success counts on the resulting five-seed supports (455 silent, 730 observable, and 3,815 other-class pairs of episode and seed). "Other classes" pools the remaining PARTNR-native assumption classes.

Method	Human behavior	Other classes	Silent human	Observable human
RoboSafe	56.6	62.1	52.1	59.5
DOUBTBEFOREACT, Gaussian	61.4	63.8	58.5	63.2
DOUBTBEFOREACT, full	**64.0**	**66.0**	**61.8**	**65.3**

Table 18: **Pre-action compute analysis under the five-call headline cap.** Decision-visible work is the subset that can change the returned plan. Removing accounting-only work changes only resource totals because that work is non-decision-visible.

Method	total calls	visible calls	total evals	visible evals	tokens (k)	s/plan
ReAct	5.0	4.3	80	14	4.1	5.8
Tree of Thoughts	5.0	5.0	80	38	6.6	10.5
SafeMem	5.0	4.5	80	20	4.8	8.4
RoboSafe	5.0	4.6	80	22	5.0	7.1
DOUBTBEFOREACT, Gaussian	5.0	5.0	80	80	6.1	10.6
DOUBTBEFOREACT, full	**5.0**	**5.0**	**80**	**80**	**6.3**	**11.4**

Table 19: **Planning latency per benchmark.** Per-benchmark Qwen3-32B end-to-end planning latency (seconds per plan) for every method, the − Doubt ablation, and the fixed-world foresight control. Column headers follow Table 1. The macro column is the equal-weight mean over the benchmarks each method supports and gives the s/plan values in Tables 1, 3, 10, and 14. LLM+P supports five benchmarks. A dash marks a benchmark that the method does not support.

Method	DESP.	EMBG.	SafeAg.	IS-B.	HAZ.	PARTNR	EAI	EB-A	macro
Zero-shot CoT	3.3	3.2	3.7	3.8	4.2	4.3	2.8	3.5	3.6
ReAct	5.3	5.1	6.0	6.2	6.7	6.9	4.6	5.6	5.8
LLM+P	5.8	5.5	6.6	–	–	–	6.2	6.9	6.2
Tree of Thoughts	9.6	9.2	10.9	11.1	12.2	12.5	8.3	10.2	10.5
KnowNo	5.8	5.6	6.6	6.8	7.4	7.6	5.1	6.3	6.4
Introspective Planning	7.3	7.0	8.3	8.5	9.3	9.5	6.3	7.8	8.0
SafePlan	6.9	6.7	7.9	8.1	8.8	9.1	6.0	7.3	7.6
SafeMem	7.7	7.4	8.7	8.9	9.7	10.0	6.6	8.2	8.4
RoboSafe	6.1	5.8	7.5	7.9	8.3	8.8	5.4	7.0	7.1
AgentSpec	8.0	8.1	8.2	7.9	8.1	8.3	8.0	8.2	8.1
RoboGuard	8.8	9.0	9.1	8.7	8.9	9.2	8.9	9.0	9.0
BrainMem	8.5	8.6	8.7	8.4	8.6	8.8	8.5	8.7	8.6
ConceptAgent	9.8	10.0	10.1	9.7	9.9	10.2	10.0	10.1	10.0
Fixed-world foresight	6.0	5.8	6.7	6.9	7.4	7.6	5.3	6.3	6.5
DOUBTBEFOREACT, − Doubt	6.8	6.5	7.5	7.7	8.3	8.5	6.0	7.1	7.3
DOUBTBEFOREACT, Gaussian	11.1	11.3	11.4	11.0	11.2	11.5	11.3	11.1	11.2
DOUBTBEFOREACT, full	**10.4**	**10.0**	**11.8**	**12.1**	**13.2**	**13.6**	**9.0**	**11.1**	**11.4**

Table 20: **Benchmark-specific equal-wall-clock control on Qwen3-32B.** Each cap is the per-benchmark planning latency of ConceptAgent, the strongest baseline, from Table 19. Capped DOUBTBEFOREACT is full DOUBTBEFOREACT with its planning time limited to the cap. Baseline scores are the nominal scores of Table 1. Δ is capped DOUBTBEFOREACT minus ConceptAgent in percentage points. The macro rows average benchmarks with equal weight.

Benchmark	strongest baseline	cap (s/plan)	capped DOUBTBEFOREACT	baseline	Δ
DESPITE	ConceptAgent	9.8	77.0	70.1	+6.9
EMBODYGUARD	ConceptAgent	10.0	78.7	72.8	+5.9
SafeAgentBench	ConceptAgent	10.1	68.8	61.1	+7.7
IS-Bench	ConceptAgent	9.7	62.1	55.0	+7.1
HAZARD	ConceptAgent	9.9	65.4	58.0	+7.4
PARTNR	ConceptAgent	10.2	70.8	64.0	+6.8
EAI	ConceptAgent	10.0	73.2	67.0	+6.2
EB-ALFRED	ConceptAgent	10.1	61.7	55.8	+5.9
Safety macro		9.9	70.4	63.4	+7.0
Long-horizon macro		10.1	68.6	62.3	+6.3
Eight-benchmark cap macro		10.0			

Table 21: **Frontier-API planning telemetry for OpenAI `gpt-5.6-sol` at high reasoning effort, with the configuration frozen on 2026-08-15.** Cost includes planner-model input and output tokens only. Simulator and evaluator compute is reported separately.

Model	Method	s/plan	tokens/plan (k)	cost/plan (USD)
`gpt-5.6-sol`, high	RoboSafe	4.8	4.9	0.041
`gpt-5.6-sol`, high	DOUBTBEFOREACT, full	7.6	6.2	0.063

Table 22: **Published-native operating-point sensitivity on the three long-horizon benchmarks.** This control does not match pre-action LLM calls and asks whether the five-call headline cap suppresses baselines with longer native deliberation. The EAI column uses the evaluation denominator reported in Table 6. The s/plan column averages the long-horizon benchmarks each method supports. Execution-time monitor calls are excluded here and reported separately in Table 29. A dash marks a benchmark that the method does not support.

Method	pre-action calls/task	PARTNR	EAI	EB-ALFRED	s/plan
ReAct	10.8	61.6	62.3	47.0	9.2
LLM+P	6.2	–	65.5	50.8	7.5
Tree of Thoughts	12.3	63.0	65.5	53.7	15.6
SafeMem	6.5	61.0	63.5	52.6	9.6
RoboSafe	6.8	61.9	64.7	52.4	8.4
DOUBTBEFOREACT, full	**5.0**	**71.9**	**74.2**	**62.4**	**11.2**

Table 23: **Oracle-evaluable red teaming denominator.** Runs are allocated per generator across the shared compatibility subset and sum to 1,000. The subset requires a trusted s oracle. The final column is computed at the unique task-ID level and reports how often that oracle finds a zero-harm successful plan. A realized failure counts as avoidable only when $s = 1$, while the failures-per-1,000 denominator remains all 1,000 generator runs because the oracle can evaluate s for every run in this subset.

Benchmark	oracle-evaluable task IDs	runs/generator	safe plan exists (%)
SafeAgentBench	204	350	91.2
IS-Bench	106	250	88.7
PARTNR	252	400	94.0
Total / task-ID weighted	562	1000	92.0

Table 24: **Plausibility-gate sensitivity on a blinded 400-run-per-generator audit subset.** The blind-filter condition applies a blinded-majority physical-plausibility filter to the same shared-gate-conditioned audit set before scoring failures. Rates are rescaled to failures per 1,000 runs, so every reported rate is a multiple of 2.5. Coverage is omitted because archive coverage uses the separate archive-building campaign in Table 9.

Generator	shared fail./1k	blind fail./1k
CMA-MAE	100.0	90.0
RedVLA	105.0	92.5
DOUBTBEFOREACT, Gaussian	172.5	115.0
DOUBTBEFOREACT, α-stable	**205.0**	**137.5**

Table 25: **Sensitivity to the plausibility threshold θ_{plaus}.** The frozen operating point is 0.62. Red-team failure rates use the same 1,000-run oracle-evaluable denominator as the headline analysis. Safety is the equal-weight five-benchmark macro. Archive coverage is omitted from this threshold sweep because its denominator is the separate archive-building campaign in Table 9.

θ_{plaus}	plausibility pass (%)	failures/1k	safety macro (%)
0.50	93.4	231	69.8
0.56	91.2	217	70.8
0.62	**89.0**	**205**	**71.5**
0.68	84.7	194	70.9
0.74	79.5	181	69.7

Table 26: **Mechanism-trace fragment for the pre-specified PARTNR case (validation episode 0731, seed 3).** The table shows the observed world and four of the 19 accepted Doubt worlds. Table 43 lists all 20 planning worlds. The selector averages the four worst utilities over that full set.

Candidate	observed	corridor occupied	human crosses	object moved	delayed closure	tail CVaR
Short route p_0	0.84	0.12	0.18	0.51	0.34	0.22
Wait route p_1	0.70	0.67	0.64	0.69	0.65	0.64
High-clearance p_2	0.76	0.72	0.71	0.74	0.70	0.70
Sense-first p_3	0.68	0.66	0.69	0.65	0.67	0.65

Table 27: **Matched-native-use results on Qwen3-32B.** Accounting-only calls and evaluations are removed, so the visible-resource columns show only work allowed to influence the published decision rule. Scores equal the nominal condition where the removed work was already non-decision-visible. A dash marks a benchmark that the method does not support.

Method	visible calls	visible evals	DESPITE	PARTNR
Zero-shot CoT	5.0	5	49.0	50.6
ReAct	4.3	14	52.4	56.9
LLM+P	5.0	28	57.8	–
Tree of Thoughts	5.0	38	58.9	61.2
KnowNo	4.2	12	62.0	55.3
Introspective Planning	4.7	16	61.8	57.1
SafePlan	4.8	18	64.8	59.1
SafeMem	4.5	20	65.7	60.2
RoboSafe	4.6	22	66.8	60.8
DOUBTBEFOREACT, full	**5.0**	**80**	**78.1**	**71.9**

Table 28: **Near-matched native-rollout control on Qwen3-32B.** The capped DOUBTBEFOREACT row uses $m = 5$, hence 20 decision-visible executions of plans in worlds, and is compared with RoboSafe under its 22-execution native decision rule. The full 80-execution row is shown only as a reference.

Condition	decision-visible evals	DESPITE	PARTNR	EB-ALFRED
RoboSafe, native	22	66.8	60.8	51.6
DOUBTBEFOREACT, $m = 5$ cap	20	75.6	69.1	60.0
DOUBTBEFOREACT, full reference	**80**	**78.1**	**71.9**	**62.4**

Table 29: **Monitored-execution accounting on PARTNR.** Pre-action calls are separated from additional execution-time planner calls triggered by monitoring. The call-matched RoboSafe row allows the same total-call allowance as full DOUBTBEFOREACT.

Condition	pre-action calls	exec. extra calls	total calls	task success
RoboSafe, native monitor	5.0	0.6	5.6	60.8
RoboSafe, call-matched monitor	5.0	0.9	5.9	61.3
DOUBTBEFOREACT, no execution monitor	5.0	0.0	5.0	68.9
DOUBTBEFOREACT, full monitor	**5.0**	**0.9**	**5.9**	**71.9**

Table 30: **Radicality-shape sensitivity.** The accepted-large-edit column is the fraction of accepted worlds with at least three edited assumptions. The score pattern is non-monotone in α, so $\alpha = 1.6$ is a tested operating point.

Proposal	accepted $K \geq 3$ (%)	DESPITE	PARTNR	EB-ALFRED
Gaussian control	8.4	71.0	64.8	56.0
$\alpha = 1.8$	11.6	77.7	71.5	62.0
$\alpha = 1.6$	15.4	78.1	71.9	62.4
$\alpha = 1.4$	19.1	77.4	71.0	61.5
$\alpha = 1.2$	24.8	76.8	70.4	60.9

Table 31: **Uncertainty for the radicality-shape sweep.** Mean score averages DESPITE, PARTNR, and EB-ALFRED only for this diagnostic. The final column is the paired difference from $\alpha = 1.6$ with a 95% confidence interval. Overlap among the stable-law settings prevents reading the pointwise maximum as a precise optimum.

Proposal	mean score	seed SD (percentage points)	Δ vs. $\alpha = 1.6$ [95% CI]
Gaussian control	62.4	1.88	-2.57 [-4.85,-0.29]
$\alpha = 1.8$	63.3	2.06	-1.67 [-3.74,0.40]
$\alpha = 1.6$	64.9	1.72	0.00 [ref.]
$\alpha = 1.4$	64.2	2.21	-0.70 [-2.88,1.48]
$\alpha = 1.2$	63.4	2.63	-1.57 [-4.26,1.12]

Table 32: **Paired uncertainty analysis for diagnostic controls.** Deltas are control minus full DOUBTBEFOREACT on macro average with equal weights scores. Intervals are 95% paired normal-approximation intervals using the task-cluster bootstrap standard error.

Control	safety Δ [95% CI]	long Δ [95% CI]
Manual assumptions	+0.8 [-1.2,2.8]	+0.3 [-1.4,2.0]
Fixed-world foresight	-4.7 [-7.8,-1.6]	-1.2 [-3.3,0.9]
Direct free-form doubt	-4.8 [-7.2,-2.4]	-2.9 [-5.1,-0.7]
Uncalibrated stable	-1.9 [-4.1,0.3]	-1.2 [-3.0,0.6]
Gaussian radicality	-3.4 [-5.9,-0.9]	-2.1 [-4.0,-0.2]

Table 33: **Paired PARTNR trace for validation episode 0731 and seed 3, the same case used by Table 26.**

Method	action summary	monitor triggers	extra planner calls	task success
RoboSafe	short route → blocked corridor → wait/retry → time limit	1	0	0
DOUBTBEFOREACT	high-clearance route → human crossing observed → incremental re-doubt → continue	1	0	1

Table 34: **First-applicable attribution of full-DOUBTBEFOREACT misses on the five safety benchmarks under held-out assumption classes and five seeds.** The cascade order is extraction → world support → simulator/evaluator → plan generation → decision rule, so categories are mutually exclusive. The total is computed from the integer success counts listed below Table 13.

First attributed cause	count	share (%)
Extraction	1038	23.3
World support	762	17.1
Simulator/evaluator	401	9.0
Plan generation	1275	28.6
Robust decision rule	980	22.0
Total failed pairs of episode and seed	4456	100.0

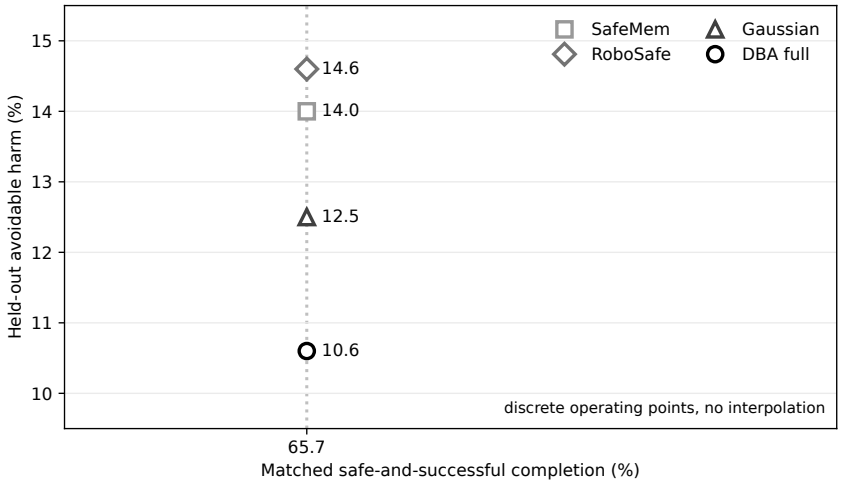

Figure 4: **Matched-success point visualization for the held-out transfer subset in Table 2.** The four plotted points are evaluated at the common target of 65.7%: SafeMem 14.0, RoboSafe 14.6, Gaussian 12.5, and full DOUBTBEFOREACT 10.6 avoidable harm. The figure reports discrete operating points without interpolation.

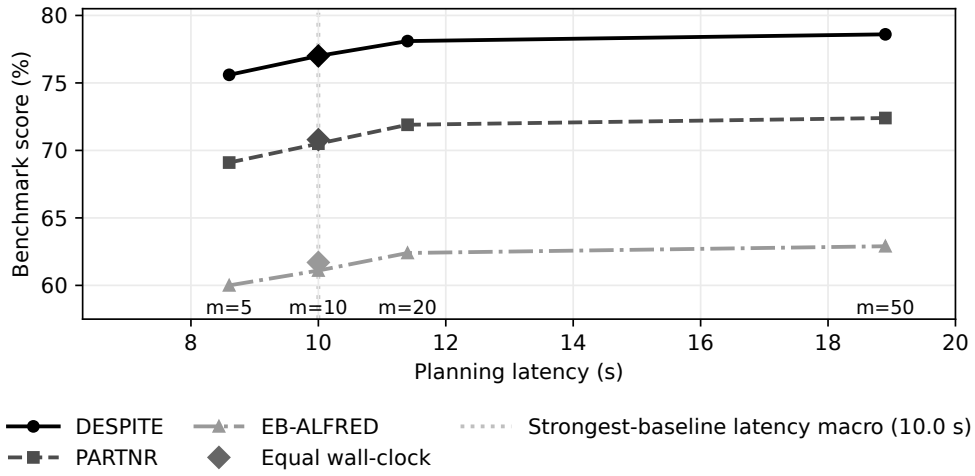

Figure 5: **World-budget trade-off.** The dotted vertical line marks the 10.0 s macro of the caps of the strongest baseline. The connected curves are the $m \in \{5, 10, 20, 50\}$ sweep using the eight-benchmark macro latency on the x-axis. Diamonds show the benchmark-specific equal-wall-clock condition. Labels for m are placed below the corresponding points. The three y-axis curves use each benchmark's native headline endpoint.

D Discussion and Deployment Interpretation

Further limitations. Our benchmarks omit much of real contact dynamics, perception error, actuator noise, and human variability, so DoubtBeforeAct provides no physical safety guarantee. Assumption extraction can miss or invent dependencies, and the incident corpus is reporting-biased. Table 34 separates extraction from downstream selection errors, while retrieval serves only as a plausibility filter. Counterfactual evaluation also adds latency, and equal nominal budgets do not imply equal useful compute across heterogeneous baseline designs, which the compute controls in Section 4 bound but do not remove. The plausibility filter can reject rare but physically possible worlds, and because all red teaming generators share it, comparisons may favor failure types that it represents well. Threshold sweeps, leave-one-source-family-out audits, and blinded human realism test this dependence but do not remove it. Finally, the paper studies high-level planning rather than low-level collision avoidance, VLA control safety (Zhang et al., 2026b), hardware certification, or adversarial sensing.

Rationale for explicit assumptions. A critic consumes a plan and scene and predicts whether something is wrong. Our object is more diagnostic, because a dependency can be named, flipped, typed, attached to a plan, and monitored. This distinguishes detecting a hazard from selecting a plan whose success depends less on the fragile premise, which is the behavior targeted by robust planning.

Rationale for heavy-tailed radicality. Safety failures are not uniformly small perturbations. A Gaussian control is deliberately included because it preserves the rest of the method. The heavy tail counts as a separate contribution only if the full system beats Gaussian edits on held-out failures, and Table 2 shows that it does. A heavy-tail gain confined to the method's own generated worlds is also insufficient. The claim requires transfer to held-out and independently generated failures.

Scope of the method. DoubtBeforeAct does not guarantee real-world safety, complete coverage of possible failures, or calibrated accident probabilities. The worlds it samples can omit an important failure mode, so we test this empirically with held-out assumption classes and an independent generator of test worlds.

E Formal Assumption Taxonomy

Admission criteria. We support a *privileged* regime populated from PDDL, BEHAVIOR Domain Definition Language (BDDL), or simulator state, and a *non-privileged* regime populated from the agent's scene or world graph and perception outputs. Admission uses five criteria frozen before test. **P1** requires task relevance, meaning that changing the proposition can alter the feasibility, trace, or cost of at least one admissible plan schema. **P2** requires that the proposition is unverified before use. **P3** assigns one of eight typed classes. **P4** requires a graded family of alternatives that keep the schema valid. **P5** requires generated alternatives to pass the frozen plausibility filter.

Table 35 lists the eight classes. Each class defines a schema, validator, and ordered edit family. Examples are illustrative and must be instantiated with environment-valid predicates.

In the schemas, e denotes an entity, a an action, h a human agent, t a time index, x_t a state variable at time t, v a property value, r a region, τ a time offset, and $d_h(t)$ the position of h at time t.

Table 35: **Typed assumption classes, graded alternatives, and validation sources.**

Class	Canonical schema	Example graded family	Validation
Object property	$\mathrm{prop}(e) = v$	empty → partly full → full, cool → warm → hot	type domain, simulator property constraints
State persistence	$x_t = x_{t+\tau}$	open persists → may close → closes before use	transition model
Space/occupancy	$\mathrm{free}(r, t)$	free → transient occupancy → blocked	navigation graph / collision-free occupancy
Human behavior	$d_h(t)$ / intent	stationary → walking → trajectory intersection	Habitat/PARTNR trajectory validity
Affordance/capability	$\mathrm{can}(a, e)$	easy grasp → marginal → unavailable	action preconditions / skill model
Environment dynamics	exogenous state	dry → wet, no fire → spreading fire	simulator dynamics
Normative context	contextual predicate	public → private, awake → sleeping	benchmark norm schema
Instruction grounding	referent mapping	unique → ambiguous → alternate valid referent	object/relation type consistency

Silent versus observable. A break is silent at planning time if the agent's current observation and standard planner input do not expose the changed proposition before the action whose outcome depends on it. The label depends on the environment and sensor model rather than on assumption class alone.

F Operator Signatures

The catalog contains eight structural operator families inherited from the typed representation language of TROPE plus token-level resampling as a non-structural proposal control. Table 36 makes the allowed role of every operator explicit. A dash means that the operator is not used in that mode.

Operator sampling and proposal budget. Each Propose call draws one operator uniformly from the Propose-valid operators that apply to the current representation after schema filtering and makes one edit with that operator that keeps the scene schema valid. Propose draws no radicality. Each Doubt proposal draws one operator uniformly from the applicable Doubt-valid operators and then draws its radicality ρ from $\mathcal{S}_\alpha(\gamma)$. Doubt issues at most $N_{\mathrm{prop}} = 64$ proposals per decision point and stops as soon as 19 worlds are accepted. If fewer worlds are accepted within this cap, planning continues with the n_{acc} accepted Doubt worlds, the effective world budget $m_{\mathrm{eff}} = 1 + n_{\mathrm{acc}}$, and the tail count $q = \max(1, \lfloor 0.2\, m_{\mathrm{eff}} \rfloor)$. In the headline condition the five pre-action LLM calls are one assumption-extraction call, one call for the unedited plan, and three Propose calls.

Let $\mathcal{R}$ denote schema-valid typed representations. Unary operators have signature $o : \mathcal{R} \times \mathbb{R}_{\geq 0} \to \mathcal{R}$. Transplant/import operators may additionally consume a donor representation or a typed library element. All operators are monotone in requested edit magnitude in the sense that increasing ρ cannot decrease the operator's class-specific discrete edit count before validation.

Assumption violation. Select $K(\rho) = \min\{|\mathcal{H}|, \max(1, \lceil \rho \rceil)\}$ mutually compatible assumptions and replace each with a graded alternative. Contradictory worlds are rejected by the schema validator.

Table 36: **Typed operator catalog and allowed roles.** Propose edits a search representation while preserving the original task. Doubt constructs a candidate world that violates or changes a plan-relevant premise.

Operator	Propose mode	Doubt mode
Assumption violation	–	Replace one or more unverified assumptions with graded alternatives
Type shift	Re-type an entity/region to expose another valid action schema	Change an affordance-relevant type/property within the ontology
Foundational negation	Remove a non-goal frame convention from the search representation	Invalidate a high-level environmental convention when schema-valid
Reification	Promote a relation into an explicit constraint/entity	–
Structural transplant	Import a validated typed subgraph that enables another plan structure	Import a validated disruption subgraph
Categorical import	Import an abstract action pattern from a development-side frame	Import a typed failure pattern from another development-side frame
Role swap	Exchange type-compatible object/receptacle/route roles	Exchange type-compatible agent/object positions or roles
Goal Pareto inversion	Reorder non-dominated auxiliary priorities while keeping the benchmark goal fixed	–
Token-level resampling	Decode another candidate with no structural edit	–

Type shift. Move an entity to a neighboring or more distant valid type in the environment ontology. In propose mode this exposes alternative action schemas. In doubt mode it changes an affordance-relevant property.

Foundational negation. Remove or invert a frame-level convention that is not a benchmark constraint. The original task specification is never edited.

Reification. Promote a relation into an explicit entity or constraint node. For example, $\mathrm{near}(h, e)$, the relation that person h is near entity e, becomes a risk-zone object with occupancy and clearance relations.

Structural transplant and categorical import. Transplant a typed subgraph or an abstract action pattern from a donor frame after a correspondence map is validated. Donors come only from the development-side library, and evaluation failures are not added online.

Role swap and goal Pareto inversion. Role swap exchanges type-compatible entities. Pareto inversion changes search priority among non-dominated auxiliary objectives while retaining the benchmark goal as a hard constraint.

F.1 Radicality Calibration and Structural Distance

We fix the stable-law tail index at $\alpha = 1.6$ globally and never tune it on a benchmark outcome. To put heterogeneous edits on one scale, define a type-balanced structural distance $d_{\mathrm{struct}}(R, R') \in [0, 1]$. For each mutable field family f, categorical and relational fields use Jaccard distance between their typed atom sets, while numeric fields use the mean clipped displacement $\min(|x - x'|/s_f, 1)$ with s_f equal to the development-side interquartile range for that field family. d_{struct} is the unweighted mean over mutable families present in the representation, so large families cannot dominate merely by having more atoms.

Within each leave-two-classes-out fold, the scale γ is chosen from the pre-specified log-spaced grid $\{0.25, 0.35, 0.50, 0.71, 1.00, 1.41, 2.00\}$ using only the six development classes. The criterion is the 1-Wasserstein distance between the d_{struct} distribution of accepted generated edits and independently authored development perturbations. Ties choose the smaller γ. Task reward, harm, safe and successful completion, and test acceptance rates are not used. The Gaussian control is scale-matched inside the same fold to the median accepted d_{struct} and receives the same operator sampling rule and plausibility filter. Thus the heavy-tail ablation changes tail shape rather than nominal edit size.

F.2 Utility, Fallback, Extraction, and Monitoring Rules

Planning utility. $g(p, w)$ is the fraction of goal predicates that the planner can see and that hold after executing p in w, and $b(p, w) = 1$ if the executed trace violates any safety constraint in Σ_{vis}, which the planner can see, and 0 otherwise. $c(p)$ combines action count, predicted duration, and resource use with weights 0.45, 0.35, and 0.20, and the cost coefficient is $\lambda = 0.10$. Hence $U(p, w) \in [-0.10, 1]$.

The harm label d behind safe and successful completion is binary on all five safety benchmarks. HAZARD native damage stays graded and is reported only as a secondary metric in Table 16.

Fallback. The fallback threshold is 0.50 on every benchmark and is compared with the best lower-tail score in Eq. (4). When no candidate reaches it, the controller returns one fixed action per benchmark. It stops on DESPITE, EMBODYGUARD, SafeAgentBench, and EAI, pauses on IS-Bench, senses on HAZARD and EB-ALFRED, and asks for help on PARTNR.

Assumption extraction. In the privileged regime, the extraction call selects the uncertain propositions relevant to the task and assigns each its class and provenance. PDDL, BDDL, or simulator state supplies the exact typed facts and their legal alternatives, and deterministic code validates these outputs and materializes them in R.

Dependency tracing. An assumption belongs to $\mathcal{H}(p)$ when its typed predicate lies on a transitive symbolic dependency path to an action precondition or effect guard, a derived predicate, a route condition, a feasibility condition, or a plan-cost condition of p.

Monitor trigger and replanning. The monitor fires when a new observation contradicts a traced typed assumption of the executing plan or moves its graded value outside the accepted bin that the plan was selected under. After a violation, only worlds of the violated assumption class and its downstream dependencies are resampled. The planning set is rebuilt from the updated observed world, the unaffected worlds, and the resampled worlds. Feasible surviving candidates are re-scored on this set, and the controller switches to the new maximizer of Eq. (4) when it differs from the current plan. Propose runs again only when no survivor is feasible or every survivor scores below the fallback threshold.

G Full Experimental Protocol

Secondary safety diagnostics. Fallback or ask rate is the fraction of episodes in which the controller returns a benchmark-supported sense, ask, pause, stop, or reject action before task completion. Safety effort is the number of explicit sensing, verification, wait, ask, or mitigation actions per 10 executed task actions among non-fallback episodes, restricted to benchmarks that expose those actions. These diagnostics are reported alongside safe and successful completion so a safety gain cannot be explained only by refusing more tasks.

G.1 Per-Benchmark Evaluation

DESPITE. We use the hard split as the primary symbolic safety test. Assumption breaks operate on contextual state predicates visible to the planner and task-relevant relations. Hidden hazard rules remain evaluator-only. s is decided by the ENHSP planner (Scala et al., 2016) run on the safety-augmented PDDL problem. We report safe and successful completion, avoidable harm, safety effort, refusal rate, and latency.

EMBODYGUARD. The dataset-composition section of the source paper (Son et al., 2025) lists 541 malicious and 402 situational scenarios, while its abstract reports 942 total scenarios. We preserve that source discrepancy rather than resolving the off-by-one by assumption. The plan-safety evaluation uses only the 402 situational scenarios, with one item for each eligible pair of situational scenario and held-out assumption class, which gives 684 items per seed. The 541 malicious-command scenarios are reserved for a separate refusal evaluation of RoboSafe and full DoubtBeforeAct under the headline model, seeds, and budget. Over the five seeds, RoboSafe refuses 476, 486, 481, 489, and 478 of the 541 commands, and full DoubtBeforeAct refuses 500, 508, 505, 511, and 502 (Table 16).

SafeAgentBench. We execute the benchmark's high-level actions in SafeAgentEnv. Because people are not represented as general dynamic scene entities in the same way as PARTNR, human-trajectory assumptions are not injected here. Environment harm labels are used to construct safe and successful completion, while native success, rejection, and raw harm rate are reported separately in Table 16.

IS-Bench. We preserve the native interactive risk order. Counterfactuals may alter state visible to the planner before a risk emerges but may not rewrite the benchmark's required mitigation ordering. We report process-oriented safety and task completion separately.

PARTNR. We use both privileged and non-privileged world-graph conditions. Whole house and scene clusters are assigned to a condition by a frozen hash, stratified by task type toward a target of 40% privileged and 60% non-privileged episodes, which yields 418 privileged and 582 non-privileged episodes per seed. Human-behavior counterfactuals operate on simulator-valid trajectories. The primary analysis separates silent from observable changes and reports percent complete in addition to binary success.

EAI / EB-ALFRED. EAI's published annotation table (Li et al., 2024) contains 338 VirtualHome task instructions/trajectories and 100 BEHAVIOR task instructions. The EAI evaluation creates one item for each eligible pair of native instruction and held-out assumption class after task-preservation and duplicate filtering. Items are ordered by a deterministic hash frozen before any model run, and the first 1,000 of the 1,137 eligible pairs form the evaluation set for every seed. EB-ALFRED is taken from EmbodiedBench (Yang et al., 2025b). We do not infer avoidability without a trusted oracle.

HAZARD. HAZARD uses fire, flood, and wind dynamics (Zhou et al., 2024b). The official benchmark submission tooling exposes 25 fire, 25 flood, and 20 wind test episodes (70 total) (UMass Embodied AGI Group, 2026). The HAZARD evaluation creates one item for each eligible pair of native episode and held-out assumption class and retains all validated pairs, which gives 222 items per seed. HAZARD natively reports rescue *value*, episode *step*, and *damage*. Safe and successful completion is an explicitly derived binary endpoint. Here $y = 1$ only when the episode rescues the full designated target set, $d = 1$ whenever native damage is nonzero, and $\mathrm{SS} = 1$ requires both complete rescue and zero damage. We additionally report native value, step, and damage separately, and we do not relabel HAZARD harm as avoidable without a trusted oracle.

G.2 Held-Out Assumption Folds and Independence

The taxonomy is fixed before evaluation. Development evidence for each held-out assumption class is removed, while the existence and schema of the class remain fixed globally. The four folds are pre-specified as follows.

Table 37: **Leave-two-classes-out protocol.** Calibration, prompt examples, retrieval material, and development perturbations exclude both test classes in each fold, and the development classes are the remaining six. The last three columns give the evaluation items per seed on the three benchmarks whose items are built from pairs of native task and held-out class (Appendix G.1).

Fold	Held-out assumption classes	EAI	HAZARD	EMBODYGUARD
F1	object properties, human behavior	236	38	148
F2	state persistence, environment dynamics	218	92	171
F3	space/occupancy, normative context	251	58	196
F4	affordance/capability, instruction grounding	295	34	169
Total		1,000	222	684

Table 37 lists the four pre-specified leave-two-classes-out folds. ABS test perturbations are authored through a separate benchmark-side pipeline that does not import the method's operator library, prompts, retrieval corpus, or random seeds. This pipeline instantiates schema templates, and two human reviewers check its output. Development perturbations use the same schema language with disjoint source and seed material and a separate pair of reviewers. No LLM generates test or development perturbations. Before evaluation we canonicalize each perturbation into the tuple (benchmark, task ID, assumption class, target entity/relation, changed predicate, severity bin, observability) and reject any exact tuple shared with development. We then screen textual descriptions for near duplicates against development descriptions using the frozen `all-mpnet-base-v2` sentence encoder (Reimers and Gurevych, 2019). Pairs above cosine similarity 0.95 are adjudicated by two authors who did not take part in test generation and are either rewritten before freezing the suite or removed. Test descriptions and metadata are never added to the incident-retrieval index. The resulting fold manifest is hashed before running any model and released with the task IDs and provenance fields, preventing test-world replacement after method outputs are observed. A benchmark contributes only assumption classes representable by its native state schema. We never inject a class solely to fill a fold. Each headline test item is assigned to a held-out class native to that benchmark, and benchmark macro-averages are computed over tasks rather than over an artificially balanced class mixture.

G.3 Baseline Configuration Principles

Several budgets appear below, and each has one meaning.

- *Nominal budget.* Every method gets the same five planner model calls and 80 executions before acting.
- *Decision-visible call.* A call whose output can change the returned plan or fallback.
- *Accounting-only call.* A call that fills the nominal budget after a baseline has stopped, and that cannot change its output.
- *Native operating point.* The setting that the baseline's own authors use, without the shared budget.
- *Matched native use.* Each method uses only the resources that its own decision rule needs, with no accounting-only calls.
- *Equal wall-clock.* Each method gets the planning time of the strongest baseline, ConceptAgent, on each benchmark.

Zero-shot CoT samples plans directly, ReAct alternates reasoning and acting steps, LLM+P translates the task into a classical planning problem when the benchmark exposes the domain semantics, and Tree of Thoughts spends its budget on branching and branch scoring. KnowNo asks or abstains based on uncertainty, and Introspective Planning checks confidence and ambiguity. SafePlan applies formal safety logic, SafeMem adds graph memory retrieval and hazard checks, and RoboSafe applies executable runtime safety logic over its memory state. A method that cannot run on a benchmark is marked unsupported rather than given privileged structure. Held-out development evidence and independent authoring of test worlds are separate safeguards. The first blocks tuning to specific classes, and the second ensures that no headline test world is generated by Doubt. A call is *decision-visible* when its output can affect the returned plan or fallback. The matched native use control removes calls and executions that fill the budget without being decision-visible, and caps each method at the resources that its own decision rule uses.

The headline nominal resource condition is the condition of planning before acting and fixes exactly $N_{\mathrm{LLM}} = 5$ total planner-model calls per task for every method. On simulator-backed benchmarks it also fixes $N_{\mathrm{sim}} = 80$ total simulator or evaluation executions. On symbolic benchmarks every method uses zero simulator rollouts and instead receives 80 symbolic executor calls. For DoubtBeforeAct, all five model calls and all 80 executor calls are decision-visible. One call extracts assumptions, one call generates the unedited plan, and three Propose calls generate edited candidates. The default $m = 20$ planning worlds (one observed world plus 19 accepted Doubt worlds) crossed with four plans consume the 80 grounded evaluations. Typed Doubt edits, retrieval, and symbolic validation outside those grounded executions do not consume LLM calls. A separate final benchmark execution used only to score the reported outcome is evaluator-only and is not charged to the planning budget. The equal-wall-clock control uses benchmark-specific caps set to the frozen primary comparator's per-benchmark planning latency, whose macro average with equal weights is 7.5 s. Table 20 reports the caps and scores on all eight benchmarks.

Baseline fidelity takes priority over making every slot decision-visible. We first configure each external method using only native control knobs from its published mechanism (e.g., sample count, critique/revision rounds, tree expansions, uncertainty/help checks, retrieval/check rounds, or act/reason horizon) and freeze those settings on development data (Table 38). We do not add a shared verifier, safety score, counterfactual set, or aggregation rule to a baseline. The headline matched-compute run then executes exactly N_{LLM} calls for every method. If a native method terminates early, any residual calls are logged as accounting-only and cannot alter its output. We therefore report both total and decision-visible calls, plus a matched-native-use sensitivity analysis. This preserves equal total-call accounting while leaving each published baseline's decision rule unchanged.

On simulator-backed benchmarks, the same distinction is used for the 80-rollout accounting budget. Native simulator calls are decision-visible, while any residual accounting rollouts are executed under the baseline's nominal policy but are not fed back after its published stopping rule has fired. Symbolic benchmarks use zero simulator rollouts for every method and match 80 symbolic executor calls instead. The primary comparison is reported together with (i) a matched-native-use cap, where no accounting-only calls or rollouts are issued, and (ii) the equal-wall-clock control. Appendix Table 27

Table 38: **Frozen native settings of the thirteen external baselines under the five-call headline budget.**

Baseline	Frozen configuration
Zero-shot CoT	Five independent plan samples, and the first valid plan in a fixed sample order is returned.
ReAct	At most five reasoning and acting model turns, with the native early stop retained.
LLM+P	One translation call and up to four planner-feedback repair calls.
Tree of Thoughts	Branch factor 2 and depth 2, keeping the top two partial branches, with at most five model calls.
KnowNo	Three plan samples and up to two uncertainty or help checks.
Introspective Planning	Initial plan and up to two critique and revision pairs, with at most five model calls.
SafePlan	Initial plan and up to two safety-check and revision pairs, with at most five model calls.
SafeMem	Memory retrieval and check, initial plan, and up to two check and revision pairs, with at most five model calls.
RoboSafe	Initial plan and up to two safety-check and repair pairs, with at most five model calls.
AgentSpec	Safety rules fixed from the development specification and enforced at run time, with at most five model calls and 80 executions.
RoboGuard	Guard synthesis at run time with fixed verification prompts, with at most five model calls and 80 executions.
BrainMem	Retrieval memory built only from development episodes, with at most five model calls and 80 executions.
ConceptAgent	Fixed tree search depth and rollout budget, with at most five model calls and 80 executions.

reports the matched-native-use cap. The comparison with five calls adapts each baseline to a fixed budget. It does not reproduce the setting that the baseline's authors used, so the native operating point is reported separately. Token counts, decision-visible counts, and wall-clock are reported as compute diagnostics from the same task-level logs so equal nominal compute cannot hide unequal effective compute.

For every external baseline we use the authors' implementation when available, otherwise we reimplement the published decision rule and validate it on the paper's released example or benchmark configuration. Appendix Table 12 records the development-set comparator-selection margin. Adapters translate only the common state visible to the planner and action vocabulary. They do not expose hidden hazard predicates, ABS worlds, or evaluator oracles. Any benchmark on which a baseline's required modality or state is unavailable is reported as unsupported rather than supplied with privileged information.

Additional diagnostic controls. Four pre-specified diagnostic controls are reported separately from the thirteen external baselines. *Manual assumptions* replaces automatic extraction with a human-authored assumption set while withholding held-out-class outcome labels, providing an upper-cost control for extraction quality. *Fixed-world foresight* is a separate control that bypasses automatic assumption extraction, Doubt, and the assumption monitor, while scoring the original candidate generator only in the observed world. This differs from the in-system − Doubt ablation, which retains extraction, typed Propose, fallback, and monitoring. *Direct free-form doubt* asks the same backbone to generate "what could go wrong" scenarios without the typed representation or operator catalog. This differs from − structured operators, which retains automatic assumption extraction and the Propose/Doubt split but replaces typed edit schemas with schema-agnostic perturbations followed by the same validity checks. *Uncalibrated stable radicality* keeps $\alpha = 1.6$ but uses a fixed development-chosen scale without the structural-distance matching of Appendix F.1. These controls use the same state visible to the planner and applicable resource accounting as DOUBTBEFOREACT. They are diagnostics rather than part of the headline count of nine external methods.

Red-team generator matching. The archive is a grid indexed by harm type, assumption class, severity, and observability. A world enters it only if the task is valid, the world is plausible, and the planner actually causes harm. Generator comparisons use only simulator-backed environments supported by every listed generator. Random perturbations sample one schema-valid editable field and a magnitude uniformly from the frozen audit ranges. LLM-direct uses the same backbone in one unconstrained scenario-generation prompt and receives no typed operator catalog. RedVLA retains its two-stage physical-risk synthesis and gradient-free risk amplification, while CMA-MAE (Fontaine and Nikolaidis, 2022) and MAP-Elites (Mouret and Clune, 2015) retain their archive/search mechanisms. Each receives the same simulator-evaluation budget within each red teaming analysis. The archive-coverage experiment uses a separate matched 4,000-run campaign per generator, while oracle-certified failures/1k use the 1,000-run compatibility subset and the blinded human audit uses 400 runs per generator. All three analyses use the same fixed target planner with Qwen3-32B as the target backbone. A frozen evaluator-side canonicalizer maps every generated state diff, regardless of generator, onto the common archive descriptors (assumption class, evaluator harm family, severity bin, and observability) using benchmark state and outcome fields rather than generator-provided

labels. This mapping is fixed before generator comparison. This is a generator-level portability comparison. RedVLA's original VLA target setting differs from high-level LLM planning. Symbolic-only benchmarks are excluded from this table. A generated case counts as an avoidable failure only when the evaluator oracle evaluates $s = 1$ for that world.

Latency measurement. End-to-end planning latency starts when the common state visible to the planner is available and stops when a final plan or fallback decision is returned. It includes model calls, retrieval, representation edits, plausibility filtering, symbolic/simulator counterfactual evaluation, and replanning inside the planning phase. It excludes the evaluator-only rollout used solely to score the reported task outcome. Open-weight models run on the same hardware and software stack without cross-method batching. API latency is measured client-side and reported separately by provider snapshot because network/service variance is not directly comparable to open-weight inference. The equal-wall-clock control truncates only planning computation and never changes evaluator rules.

G.4 Threshold Matching

Primary deployment thresholds are selected on development data and frozen before test evaluation. Matched-success harm is a separate descriptive diagnostic. For that diagnostic, development fixes a finite fallback-threshold grid for every method and source. Every frozen grid node is evaluated once on test to form a discrete operating curve. A rule fixed in advance then chooses the target success level. It is the largest level of safe and successful completion that every combination of method and source reaches exactly at some grid node. Harm is read at those nodes, without interpolation. No matched-success table contributes a primary p-value. The common matched-success target for Tables 2 and 7 is 65.7%. Interpolation is visualization-only.

G.5 Statistics

All inferential choices are fixed before reading test outcomes. Some appendix analyses use the development comparator, the compatible baseline among the nine original baselines with the highest development-set primary metric, fixed before test evaluation. The main text compares against the strongest baseline of all thirteen, ConceptAgent. The primary estimand is the paired mean difference between the full method and that comparator on identical task IDs and seeds. The outcome is safe and successful completion for the five safety benchmarks and task success for the three long-horizon benchmarks. We use 10,000 paired task-cluster bootstrap resamples, keeping all seeds for a task together, to estimate the standard error of each paired effect. The reported two-sided 95% interval is the normal approximation $\Delta \pm 1.96$ times that bootstrap standard error. For benchmark-level significance tests, we use two-sided normal-theory p-values computed from the paired task-cluster bootstrap standard errors used for the displayed CIs, followed by Holm family-wise correction across the eight benchmark-level tests.

The headline macro-averages weight benchmarks equally. Their intervals are obtained by resampling task IDs independently within each benchmark and recomputing the macro average with equal weights on every bootstrap draw. The benchmark set itself is not resampled. Primary inferential tests use deployment thresholds frozen on development data. Matched-success harm is descriptive and uses only the pre-specified frozen threshold grids and the common-support rule in Appendix G.4. It is not re-optimized for primary inference. The analysis pipeline reports the exact sample size, seed count, effect size, interval, raw p-value, and adjusted p-value for each primary comparison.

H Additional Results

Table 39: **Backbone-scale transfer. "Comp." is the average score of the strongest baseline on each benchmark.** Safety averages the five safety benchmarks and long-horizon averages PARTNR, EAI, and EB-ALFRED. The frontier row uses OpenAI `gpt-5.6-sol` at high reasoning effort.

Backbone	Comp. safety	Ours safety	Δ	Comp. long	Ours long	Δ
Qwen3-8B	50.3	64.8	+14.6	53.0	66.1	+13.1
Qwen3-32B	63.4	71.5	+8.1	62.3	69.5	+7.2
`gpt-5.6-sol`, high	68.7	76.6	+7.9	68.6	76.9	+8.3

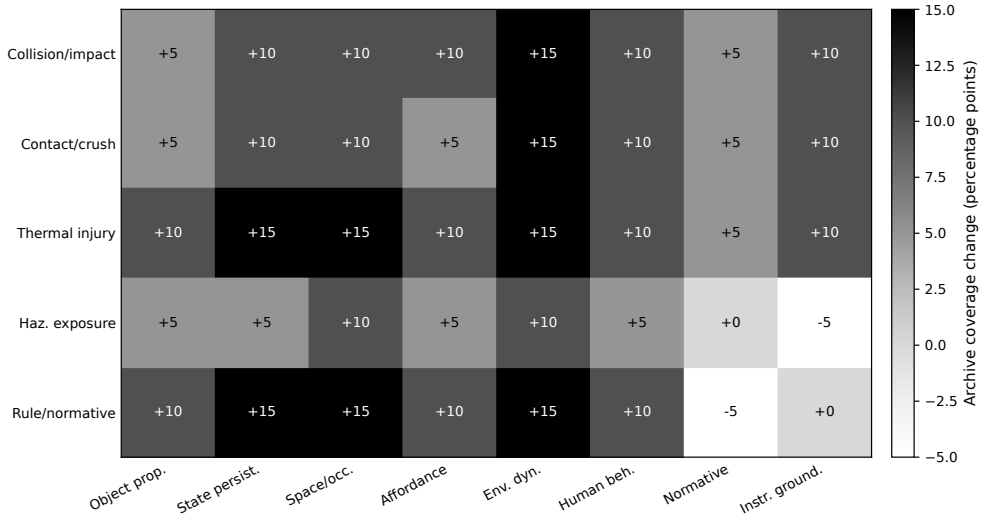

Figure 6: **Archive coverage by assumption class.** Change in failure-archive coverage from Gaussian to heavy-tailed radicality by assumption class and evaluator harm family for the matched 4,000-run archive-building campaign on the 800-cell archive. Harm families are evaluator-side categories for collision/impact, contact/crush, thermal injury, hazardous exposure, and rule or normative harm. Each combination of assumption class and harm family contains 20 cells (ten severity bins by two observability states), so every cellwise coverage change is an integer multiple of 5 percentage points. The 40 slices average +8.5 percentage points, matching the aggregate coverage change in Table 8.

Table 40: **Per-benchmark results on the two other backbones.** Scores of the strongest baseline and of full DOUBTBEFOREACT, their difference, 95% paired bootstrap intervals over task instances, and Holm-adjusted p-values over the eight benchmarks of each backbone.

Benchmark	Strongest baseline	DOUBTBEFOREACT	Δ	95% paired CI	Holm p-value
Qwen3-8B					
DESPITE	53.8	69.8	+16.0	[12.5,19.4]	<0.001
EMBODYGUARD	55.6	71.0	+15.4	[12.1,18.8]	<0.001
SafeAgentBench	49.9	64.1	+14.2	[10.8,17.6]	<0.001
IS-Bench	45.1	58.0	+12.9	[9.5,16.3]	<0.001
HAZARD	47.0	61.3	+14.3	[10.9,17.7]	<0.001
PARTNR	51.8	65.0	+13.2	[9.8,16.7]	<0.001
EAI	56.3	70.0	+13.7	[10.4,17.1]	<0.001
EB-ALFRED	50.9	63.4	+12.5	[9.0,16.0]	<0.001
`gpt-5.6-sol`					
DESPITE	73.5	81.5	+8.0	[5.3,10.7]	<0.001
EMBODYGUARD	75.1	82.8	+7.7	[5.0,10.4]	<0.001
SafeAgentBench	67.8	75.9	+8.1	[5.4,10.8]	<0.001
IS-Bench	62.5	70.3	+7.8	[5.1,10.5]	<0.001
HAZARD	64.4	72.5	+8.1	[5.4,10.8]	<0.001
PARTNR	68.7	77.0	+8.3	[5.6,11.1]	<0.001
EAI	71.0	79.7	+8.7	[5.9,11.5]	<0.001
EB-ALFRED	66.1	73.9	+7.8	[5.1,10.5]	<0.001

Appendix Table 41 reports the normalized plan-length and action-count diagnostic used to check whether long-horizon gains come from simply producing longer plans, and Table 42 gives the corresponding absolute means underlying those ratios.

I CASE-STUDY LOGGING PROTOCOL

For the pre-specified PARTNR trace, the execution record contains the following objects.

1. task ID, simulator seed, backbone version, prompt hash,
2. original world graph and the projection visible to the planner,
3. shared assumptions of the scene with class, source, P1 validation result, and candidate-specific provenance edges,

Table 41: **Normalized plan length/action count.** ReAct is normalized to 1.00 within each benchmark. PARTNR actions count planner-level action decisions, including sensing, waiting, and mitigation decisions, while Table 16 reports executed environment steps in the simulator. A dash marks a benchmark that the method does not support.

Method	EAI length	EB-ALFRED length	PARTNR actions
ReAct	1.00	1.00	1.00
LLM+P	0.96	0.94	–
RoboSafe	1.05	1.03	1.08
DOUBTBEFOREACT − Propose	1.02	1.04	1.06
DOUBTBEFOREACT, full	**0.90**	**0.89**	**0.93**

Table 42: **Absolute plan length/action-count means underlying the normalized ratios in Table 41.** PARTNR actions are planner-level action decisions and therefore need not equal executed environment steps in Table 16. A dash marks a benchmark that the method does not support.

Method	EAI length	EB-ALFRED length	PARTNR actions
ReAct	28.0	34.0	58.0
LLM+P	26.9	32.0	–
RoboSafe	29.4	35.0	62.6
DOUBTBEFOREACT − Propose	28.6	35.4	61.5
DOUBTBEFOREACT, full	**25.2**	**30.3**	**53.9**

4. every propose edit, resulting representation diff, generated plan, and original-task validity result,
5. every doubt edit, radicality, structural distance, plausibility score, and acceptance/rejection reason,
6. full $U(p, w)$ matrix with planning-time task, visible harm, and cost components,
7. selected tail fraction and lower tail CVaR score for each plan,
8. execution observations, assumption updates, monitor triggers, and any incremental re-doubt batch.

The pre-specified case is PARTNR validation episode 0731 with simulator seed 3, which belongs to the held-out human-behavior class of fold F1. Extraction returns six assumptions. The corridor is free and the alternate route remains traversable (space/occupancy), the target object remains at its pickup location and access to the destination remains open (state persistence), the human path does not intersect the route (human behavior), and the grasp and carry affordance remains available (affordance/capability). Goal Pareto inversion produces the wait route p_1, Structural transplant produces the high-clearance route p_2, and Reification produces the sense-first plan p_3. Doubt issues 21 proposals and rejects 2 of them before 19 worlds are accepted. The monitor fires at step 9, when the crossing human is observed, and resamples five worlds about human behavior without an extra planner call. RoboSafe fails by reaching the time limit after waiting and retrying at the blocked corridor. Table 26 reports a five-world fragment, Table 43 reports the complete 4×20 planning matrix, and Table 33 reports the paired RoboSafe trace for the same task and seed. Table 34 attributes held-out misses to extraction, world support, simulator or evaluator error, plan generation, or the robust decision rule using the fixed first-applicable cascade.

J INCIDENT CORPUS AND PLAUSIBILITY AUDIT

The incident corpus contains 18,742 records after near-duplicate removal. They are 9,684 consumer-injury narratives from the National Electronic Injury Surveillance System (NEISS) of the U.S. Consumer Product Safety Commission (U.S. Consumer Product Safety Commission, 2026), 6,741 severe-injury and fatality reports of the U.S. Occupational Safety and Health Administration (OSHA) (Occupational Safety and Health Administration, 2026), and 2,317 public robotics and AI incident reports. Retrieval is split by source and year so development thresholding cannot see audit items. The External transfer set uses held-out strata of source and year from this corpus and is authored by the benchmark-side evaluation pipeline without importing Doubt or CMA-MAE proposals. For a candidate world w, the frozen plausibility module retrieves development-side records with the

Table 43: **Complete planning-time utility matrix for the pre-specified PARTNR case.** The planning set contains the observed world and 19 accepted Doubt worlds, and the first five rows are the worlds shown in Table 26. The remaining fifteen Doubt worlds extend the same planning set. The four-worst-world scores are 0.22, 0.64, 0.70, and 0.65, and the selector chooses p_2.

World	Short route p_0	Wait route p_1	High-clearance p_2	Sense-first p_3
Observed	0.84	0.70	0.76	0.68
Corridor occupied	0.12	0.67	0.72	0.66
Human crosses	0.18	0.64	0.71	0.69
Object moved	0.51	0.69	0.74	0.65
Delayed closure	0.34	0.65	0.70	0.67
Corridor narrows	0.25	0.66	0.73	0.66
Human reverses	0.33	0.68	0.72	0.70
Cart blocks bypass	0.41	0.65	0.71	0.67
Door closes early	0.39	0.66	0.72	0.66
Object handed off	0.55	0.70	0.75	0.69
Human pauses in doorway	0.37	0.62	0.69	0.63
Alternate route cluttered	0.48	0.67	0.73	0.68
Target displaced	0.53	0.69	0.74	0.67
Sensor occlusion	0.46	0.68	0.72	0.66
Handoff delayed	0.58	0.71	0.75	0.70
Corridor clears late	0.62	0.72	0.77	0.71
Human accelerates	0.36	0.66	0.71	0.68
Secondary obstacle	0.43	0.67	0.72	0.67
Wait window expires	0.35	0.65	0.70	0.66
Partial map stale	0.44	0.68	0.73	0.68
Four-worst CVaR	0.22	0.64	**0.70**	0.65

E5-large-v2 encoder (Wang et al., 2022) and assigns $s_{\text{plaus}}(w) \in [0, 1]$ from relevance and entailment to the edited relation or dynamic pattern. Entailment is scored by the public DeBERTa-v3-large (He et al., 2023) natural language inference checkpoint of Laurer et al. (2024). The development-selected threshold $\theta_{\text{plaus}} = 0.62$ rejects schema-valid but physically incoherent edits and is frozen for every held-out fold. The score is used only for accept/reject, with equal weight after acceptance. The audit covers topical relevance, physical coherence, evidence support, threshold sensitivity, inter-rater agreement, and leave-one-source-family-out ablations. Corpus frequency and s_{plaus} are never interpreted as population accident probabilities.

Table 44: **Leave-one-source-family-out plausibility audit on the shared red teaming compatibility subset.** Each row rebuilds the retrieval index without one incident-source family before applying the frozen downstream scoring protocol. The final column reports the gap in oracle-certified avoidable failures per 1,000 runs.

Retrieval source omitted	records	plausibility pass (%)	DoubtBeforeAct failures/1k	RedVLA failures/1k	gap
Consumer-injury narratives	9,684	87.4	142	103	+39
Occupational injury reports	6,741	86.1	139	101	+38
Robotics/AI incidents	2,317	88.2	145	102	+43

K Compute Accounting

Every task-level record separates planner LLM calls from auxiliary non-planner computation and from resource use that can influence the method's decision. We log planner/generator LLM calls, assumption-extraction or parser calls, retrieval queries, plausibility-module invocations and tokens or embeddings as applicable, symbolic executions, simulator rollouts, prompt/completion tokens, end-to-end planning wall-clock, evaluator-only scoring work, and, where meaningful, graphics processing unit (GPU) time/peak memory or API cost. The headline nominal-budget comparison, matched-native-use sensitivity analysis, and equal-wall-clock control are all regenerated from these same records.

Batching, caching, and parallel execution are documented and may be used in a comparison only when the same opportunity is exposed to every compatible method in that condition. Open-weight model latency is measured on the same orchestration host after a documented warmup. Simulator startup and cache warmup are excluded in the same way for every method. API-model latency and cost are reported by provider/model snapshot rather than pooled with open-weight inference. Accounting-only calls or rollouts introduced solely for the nominal-resource condition remain flagged as non-decision-visible and are never used to improve a baseline output.

L Extended Related Work

Safety benchmarks for embodied LLM planning. SafeAgentBench provides 750 household tasks over 10 hazard types and 17 high-level actions in an executable AI2-THOR environment and covers both explicit and implicit hazards (Yin et al., 2024). EMBODYGUARD contains 942 PDDL-grounded scenarios, split into malicious commands that should be refused and situational hazards that call for a safe plan, and decomposes failures into goal interpretation, transition modeling, and action sequencing (Son et al., 2025). DESPITE collects 12,279 tasks with physical and normative dangers and validates plans deterministically, which separates planning ability from safety awareness (Zhang et al., 2026a). IS-Bench evaluates interactive safety over 161 scenarios with 388 risks and checks whether mitigation actions occur before or after the risk-prone steps (Lu et al., 2026).

Safety methods for LLM planners. SafePlan combines formal logic with chain-of-thought reasoning to screen robot task plans (Obi et al., 2025). RoboSafe constructs executable, context-dependent safety logic over short- and long-term memory and applies it at runtime (Wang et al., 2026b). SafeMem adds a long-term graph memory so that hazards outside the current field of view still inform planning (Li et al., 2026). KnowNo uses conformal prediction to align the uncertainty of an LLM planner with requests for help (Ren et al., 2023), and Introspective Planning aligns the planner's confidence with the inherent ambiguity of the task (Liang et al., 2024).

Long-horizon and human-robot evaluation. PARTNR provides 100,000 natural-language tasks across 60 houses and 5,819 objects for human-robot collaboration, including spatial, temporal, and heterogeneous-agent constraints (Chang et al., 2025). It runs in Habitat 3.0, which supplies humanoid agents and human-in-the-loop interaction (Puig et al., 2024). The Embodied Agent Interface unifies goal interpretation, subgoal decomposition, action sequencing, and transition modeling and provides symbolic interfaces for BEHAVIOR and VirtualHome (Li et al., 2024). HAZARD evaluates decisions under fire, flood, and wind that change the environment independently of the agent (Zhou et al., 2024b). EmbodiedBench evaluates multimodal LLM agents on high-level and low-level embodied tasks, including EB-ALFRED (Yang et al., 2025b).

Uncertainty, risk, and robust decision-making. Conformal prediction gives probabilistic safety guarantees for model predictive control among moving agents when trajectory predictions come with calibrated prediction regions (Lindemann et al., 2023). Robust Markov decision processes optimize the worst case over an explicitly specified set of transition models (Iyengar, 2005, Nilim and El Ghaoui, 2005). CVaR measures the expected loss in the worst fraction of outcomes and admits convex optimization (Rockafellar and Uryasev, 2000), and Chow et al. (2015) optimize it in Markov decision processes and relate it to robustness against model perturbations. The scenario approach replaces an intractable robust design problem with one over sampled realizations of the uncertainty and bounds the probability that the solution fails on an unseen realization (Calafiore and Campi, 2006).

Scenario generation and red teaming. Domain randomization trains policies on randomized simulator parameters so that they transfer to the real world (Tobin et al., 2017). Quality-diversity methods search for many behaviorally different solutions instead of a single optimum. MAP-Elites keeps the best solution found in each cell of a behavior grid (Mouret and Clune, 2015), and in human-robot shared autonomy it finds more diverse failure scenarios than Monte Carlo sampling or direct optimization (Fontaine and Nikolaidis, 2021). CMA-MAE improves this archive search with covariance-matrix adaptation and an annealed acceptance threshold (Fontaine and Nikolaidis, 2022). RedVLA perturbs physical scenes to expose unsafe behavior in vision-language-action policies (Zhang et al., 2026b).

LLM planning and problem reformulation. LLM+P translates a natural-language task into PDDL and solves it with a classical planner (Liu et al., 2023). ReAct interleaves reasoning traces with actions (Yao et al., 2023b), and Tree of Thoughts searches over intermediate reasoning steps with self-evaluation (Yao et al., 2023a). LLMs can also construct and refine PDDL world models that a classical planner then uses (Guan et al., 2023). Step-Back Prompting abstracts a problem to its governing principles before solving it (Zheng et al., 2024), and Self-Discover composes a reasoning structure specific to the task from atomic reasoning modules (Zhou et al., 2024a). TROPE edits typed problem representations with discrete operators and heavy-tailed edit sizes to expose alternative reasoning paths (Bikbulatov and Petrosian, 2026).