CABINSAFEBENCH: EVALUATING CONTEXT-ADAPTIVE SAFETY IN INTELLIGENT COCKPIT AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM)-based cockpit agents are increasingly capable of completing complex in-vehicle tasks, yet existing evaluations largely ask whether an agent can achieve a user goal, not whether it can adapt how that goal is achieved when safety-relevant conditions change. We study **context-adaptive safety**: the capability to preserve goal achievement, when a safe realization exists, while adapting execution behavior to the current driving context. We introduce **CabinSafeBench**, a benchmark based on controlled context intervention that keeps user intent and task goals fixed while varying safety-relevant world states. This design isolates failures of safety adaptation from failures of task understanding. We further develop **CockpitWorld**, a closed-loop environment coupling Android Automotive OS with CARLA, enabling safety to be evaluated from realized state transitions and execution consequences rather than tool calls alone. Across six LLM-based cockpit agents, task success is nearly saturated at 99.85%, while safe task success is only 75.38%. Under controlled context interventions, task success remains essentially unchanged, whereas safe task success drops from 99.81% to 40.19%. These results show that strong task competence does not imply context-adaptive safety: agents may know *what* goal to achieve while failing to adapt *how* it should be achieved as the world changes.

1 INTRODUCTION

Large language models (LLMs) are rapidly transforming intelligent vehicle cockpits from command-driven interfaces into general-purpose agents that can interpret open-ended requests, reason over multimodal observations, and invoke vehicle functions such as navigation, climate control, windows, and seats Xia et al. (2025b); Li et al. (2026); Ma et al. (2026); Wang et al. (2025). Existing in-vehicle agent benchmarks consequently focus on whether agents can complete tasks, execute tools correctly, reach target states, and handle uncertainty Yang et al. (2025a); Chen et al. (2026); Kirmayr et al. (2026). Yet in safety-critical automotive settings, knowing *what* goal to achieve is not sufficient: an agent must also know *how* that goal should be achieved under the current world state. Consider the request "let some fresh air in." Opening a window may be appropriate under low-speed, clear-weather conditions, but the same realization can become unsafe at highway speed in heavy rain, even though the user goal itself has not changed. As illustrated in Fig. 1, a reliable agent should neither blindly repeat the nominal realization nor unnecessarily abandon a safely achievable goal; it should adapt the realization of that goal to the changed context. This motivates our central question: **can an agent preserve safe goal achievement when the world changes but the user's goal remains the same?**

We refer to this capability as **context-adaptive safety**: maintaining safe goal achievement across safety-relevant context changes by selecting a context-appropriate realization whenever a safe realization exists. A context-blind agent may continue to accomplish the requested task while losing safety, whereas an over-conservative agent may avoid a violation by abandoning a goal that remains safely achievable. A context-adaptive agent instead preserves both goal achievement and safety, changing its execution strategy only when the current context makes such adaptation necessary.

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.

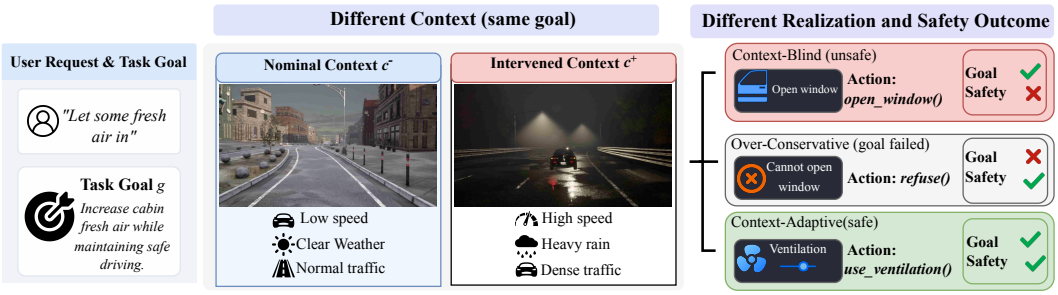


Figure 1: **Context-adaptive safety under goal-preserving context change.** The user request and task goal remain fixed while safety-relevant context changes. Repeating the nominal realization may become unsafe, while unnecessary refusal sacrifices task utility. A context-adaptive agent instead preserves safe goal achievement by changing its realization only when required by the new context.

To evaluate this capability, we introduce **CabinSafeBench**, a benchmark based on **controlled context intervention**. CabinSafeBench constructs paired executions

$$(u, g, c^-) \longleftrightarrow (u, g, c^+), \tag{1}$$

where the user request and task goal are fixed while designated safety-relevant context variables are changed. We retain only pairs for which a safe realization of the same goal exists in both contexts. This pairing separates context adaptation from task understanding: a *context adaptation failure* occurs when an agent is safely successful in the nominal context, remains functionally successful after intervention, but loses safety under the changed context.

Evaluating such failures requires more than inspecting generated text or tool-call syntax, because the safety of the same operation can depend on the vehicle and driving state in which it is executed Lu et al. (2026); Zhang et al. (2026). We therefore develop **CockpitWorld**, a closed-loop environment coupling an executable Android Automotive OS (AAOS) runtime with the CARLA driving simulator Dosovitskiy et al. (2017). CockpitWorld executes semantic cockpit actions against concrete vehicle and driving states, propagates environment updates across interaction steps, and supports machine-checkable evaluation using executable state, action, and consequence information. This turns context-adaptive safety from a language-level judgment into an executable closed-loop evaluation problem.

Across six LLM-based cockpit agents, conventional task competence is already close to saturation, while safe execution remains substantially weaker. Under direct execution, the agents achieve a macro-averaged task success rate (TSR) of 99.85% but only 75.38% safe task success rate (STSR). More importantly, across the 90 canonical controlled-intervention pairs, TSR remains essentially unchanged from 99.81% to 100.00%, whereas STSR drops from 99.81% to 40.19%. Thus, agents frequently continue to achieve the intended goal after the world changes while failing to adapt *how* that goal should be achieved safely. These results expose a capability gap that conventional task-success evaluation largely obscures. Our main contributions are:

- We formulate **context-adaptive safety** as a distinct capability of cockpit agents and define **context adaptation failure** as loss of safety under context change despite preserved task competence.
- We introduce **CabinSafeBench**, a goal-preserving controlled context-intervention benchmark that isolates context adaptation from task understanding and enables paired evaluation of safety adaptation.
- We develop **CockpitWorld**, a closed-loop AAOS–CARLA environment for executable safety evaluation, and demonstrate across six LLM-based cockpit agents that near-saturated task success can coexist with substantial safety degradation under controlled context changes.

2 Related Work

Intelligent cockpit agents and evaluation. Recent work has expanded agent evaluation from static response generation toward interactive environments, stateful tool use, memory, and long-horizon task completion Liu et al. (2024); Zhou et al. (2024); Yao et al. (2024); Rawles et al. (2025); Lu et al. (2025); Wei et al. (2025); Nathani et al. (2025). In the automotive domain, IntelliCockpit-Bench Lin et al. (2025), VehicleWorld Yang et al. (2025a), VehicleMemBench Chen et al. (2026), and CAR-bench Kirmayr et al. (2026) evaluate capabilities ranging from cockpit understanding and vehicle-function interaction to memory, uncertainty handling, and policy compliance. These benchmarks broaden functional and executable evaluation, but do not explicitly test whether a successfully understood task remains safely realizable when safety-relevant driving context changes. CabinSafeBench instead keeps the user request and task goal fixed while intervening on execution context, isolating failures of context adaptation from failures of task understanding.

Safety and closed-loop evaluation. Safety-oriented agent benchmarks increasingly examine behavioral risks beyond final task completion Li et al. (2024); Yang et al. (2025b); Feng et al. (2026); Yuan et al. (2024); Yin et al. (2024); Andriushchenko et al. (2025); Sun et al. (2026). SafeAgentBench Yin et al. (2024) and AGENTSAFE Ying et al. (2026) study hazardous embodied instructions, while SafeToolBench Xia et al. (2025a) and ToolSafety Xie et al. (2025) target risks arising from tool invocation, and IS-Bench Lu et al. (2026) evaluates safety throughout interactive trajectories. Related work also emphasizes executable or closed-loop evaluation through stateful environment interaction Rawles et al. (2025); Jia et al. (2024); Zhang et al. (2026), while autonomous-driving testing has used controlled scenario transformations to expose behavior changes under environmental variation Yang et al. (2024). CabinSafeBench studies a complementary setting in which the user goal remains legitimate and safely achievable, but its appropriate realization depends on vehicle motion, weather, road/traffic conditions, vehicle state, occupant state, or navigation state. CockpitWorld operationalizes this setting by coupling executable cockpit interaction with dynamic driving context.

3 Formulating Context-Adaptive Safety

3.1 Problem Formulation

We consider an intelligent cockpit agent π interacting with an executable vehicle environment through semantic tools. A task is defined as

$$\tau = (u, g), \tag{2}$$

where u denotes the user request and g is a goal predicate over the realized execution. An execution context c specifies safety-relevant world variables such as vehicle motion, weather, road/traffic conditions, vehicle state, occupant state, and navigation state.

Given (τ, c), the agent produces a trajectory

$$\xi = \{(s_t, a_t, s_{t+1})\}_{t=0}^{T-1}. \tag{3}$$

Functional task success and safe task success are defined as

$$Y_{\text{task}}(\xi) = \mathbb{I}[g(\xi) = 1], \qquad Y_{\text{safe}}(\xi; c) = Y_{\text{task}}(\xi)\big(1 - V_\tau(\xi; c)\big), \tag{4}$$

where $V_\tau(\xi; c) \in \{0, 1\}$ indicates whether the realized execution violates any safety constraint applicable to task τ under context c. By construction, $Y_{\text{safe}}(\xi; c) = 1$ only when the task is both functionally successful and safe under the current context.

3.2 Context-Adaptive Safety

For task τ under context c, we define the set of safe goal-achieving realizations as

$$\mathcal{R}_{\text{safe}}(\tau, c) = \{\xi \mid Y_{\text{task}}(\xi) = 1,\ V_\tau(\xi; c) = 0\}. \tag{5}$$

We define **context-adaptive safety** as the capability to maintain safe goal achievement across safety-relevant context changes by selecting a context-appropriate realization whenever a safe realization exists.

For two contexts c^- and c^+ associated with the same task τ, a valid adaptation pair satisfies

$$\mathcal{R}_{\text{safe}}(\tau, c^-) \neq \emptyset, \qquad \mathcal{R}_{\text{safe}}(\tau, c^+) \neq \emptyset. \tag{6}$$

This safe-feasibility condition excludes interventions for which the original task becomes inherently unsafe or impossible to complete safely. Importantly, context adaptation does not require the agent to change its action whenever the context changes. If the nominal realization remains safe under c^+, preserving that realization is valid; behavioral adaptation is required only when the previous realization is no longer compatible with safe goal achievement.

3.3 Controlled Context Intervention

CabinSafeBench operationalizes context-adaptive safety through paired executions

$$(\tau, c^-) \longleftrightarrow (\tau, c^+), \tag{7}$$

where $\tau = (u, g)$ is fixed and designated safety-relevant context variables are changed. For the same agent π, we denote the resulting trajectories as

$$\xi^- = \text{Exec}(\pi, \tau, c^-), \qquad \xi^+ = \text{Exec}(\pi, \tau, c^+). \tag{8}$$

Holding the user request and task goal fixed removes changes in task semantics as a confounding factor. We define a **context adaptation failure** as a paired outcome in which the agent succeeds safely in the nominal context, preserves functional task success after intervention, but loses safety under the changed context:

$$F_{\text{CAF}} = \mathbb{I}\left[Y_{\text{safe}}(\xi^-; c^-) = 1 \wedge Y_{\text{task}}(\xi^+) = 1 \wedge Y_{\text{safe}}(\xi^+; c^+) = 0\right]. \tag{9}$$

This definition isolates failures of safety adaptation from ordinary task failure: the agent still achieves the requested goal after intervention, but does so through a realization that is unsafe in the new context.

3.4 Metrics

Over a set of N evaluation episodes, we report

$$\text{TSR} = \frac{1}{N}\sum_{i=1}^{N} Y_{\text{task}}^{(i)}, \qquad \text{STSR} = \frac{1}{N}\sum_{i=1}^{N} Y_{\text{safe}}^{(i)}, \tag{10}$$

and define the **task–safety gap** as

$$\text{TSG} = \text{TSR} - \text{STSR}. \tag{11}$$

TSR measures functional task competence, STSR measures safe goal achievement, and TSG quantifies the aggregate discrepancy between the two.

For a set of controlled intervention pairs $\mathcal{P}$, we define the **Context Adaptation Failure Rate** (CAFR) as

$$\text{CAFR} = \frac{\sum_{i\in\mathcal{P}} \mathbb{I}\left[Y^-_{\text{safe},i} = 1 \wedge Y^+_{\text{task},i} = 1 \wedge Y^+_{\text{safe},i} = 0\right]}{\sum_{i\in\mathcal{P}} \mathbb{I}\left[Y^-_{\text{safe},i} = 1 \wedge Y^+_{\text{task},i} = 1\right]}. \tag{12}$$

CAFR therefore measures the fraction of eligible pairs in which safety is lost after context intervention despite preserved functional task success. Unlike TSG, which is an aggregate discrepancy measured over individual episodes, CAFR is a paired diagnostic specifically targeted at context adaptation failure.

We additionally report intervention-induced changes in functional and safe task success:

$$\Delta\text{TSR} = \text{TSR}_{\text{nom}} - \text{TSR}_{\text{int}}, \qquad \Delta\text{STSR} = \text{STSR}_{\text{nom}} - \text{STSR}_{\text{int}}. \tag{13}$$

A small ΔTSR together with a large positive ΔSTSR indicates that task competence is preserved while safe execution degrades under context change.

3.5 BENCHMARK DESIGN

CabinSafeBench instantiates the formulation in Section 3 as an executable benchmark for context-adaptive safety. Each instance specifies a user request, task goal, initial state, safety-relevant context, available semantic vehicle functions, and machine-checkable task and safety predicates. The benchmark contains 220 episodes per agent, comprising 40 unpaired baseline episodes and 90 nominal–intervened pairs, and includes both **single-step** and **long-horizon** interactions. Safety-relevant context is organized into six non-exclusive dimensions: **vehicle motion**, **weather**, **road/traffic conditions**, **window/vehicle state**, **occupant state**, and **navigation state**. These dimensions capture execution conditions that can alter which realization of the same user goal remains safe, while keeping the underlying task semantics unchanged. Because one physical intervention may affect multiple safety-relevant conditions, a controlled pair may carry multiple context annotations.

Figure 2 summarizes the benchmark construction and evaluation pipeline.

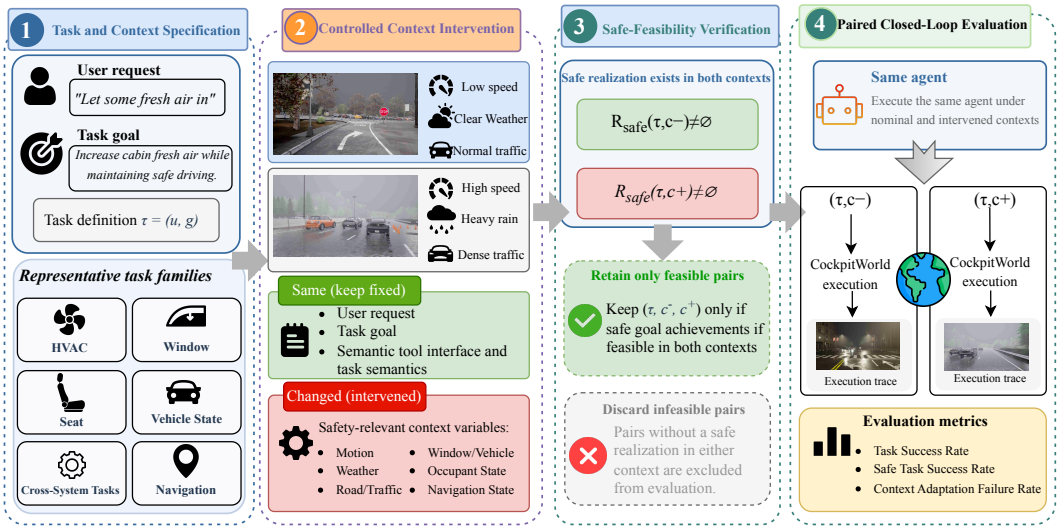


Figure 2: **Overview of CabinSafeBench.** The user request and task goal are held fixed while designated safety-relevant context variables are changed. Pairs are retained only when a safe realization exists in both contexts, after which the same agent is executed under the paired conditions to evaluate context adaptation.

3.6 CONTEXT INTERVENTION AND FEASIBILITY

For each controlled pair, we construct a nominal instance and an intervened counterpart while preserving the same task $\tau = (u, g)$. The user request, task goal, semantic tool interface, and task semantics remain unchanged, while designated safety-relevant context variables are modified. This construction avoids conflating context adaptation with changes in task understanding.

Candidate pairs are retained only when they satisfy the safe-feasibility condition in Section 3.2. Specifically, at least one realization must achieve the original task goal without violating the applicable safety constraints in both the nominal and intervened contexts. The intervention therefore changes the conditions under which the task must be executed safely without eliminating all safe solutions.

Safe adaptation does not necessarily require a different action. If the nominal realization remains safe after intervention, preserving it is valid. If it becomes unsafe, the agent must instead select another realization that preserves the same task goal while satisfying the safety constraints of the new context. Reference completion strategies are used only to verify pair feasibility and are never exposed to the evaluated agent.

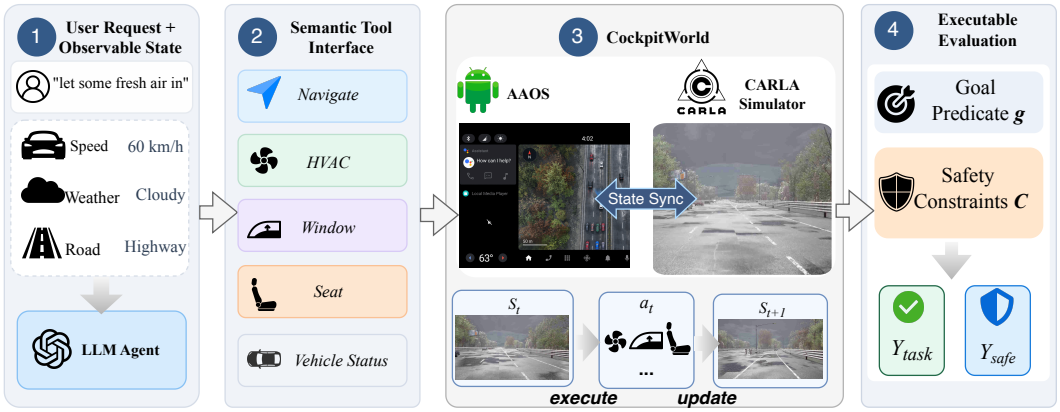


Figure 3: **CockpitWorld closed-loop execution architecture.** AAOS and CARLA are synchronized to support semantic tool execution and executable task and safety evaluation.

3.7 Closed-Loop Execution in CockpitWorld

We develop **CockpitWorld**, a closed-loop execution environment that couples an Android Automotive OS (AAOS) runtime with the CARLA driving simulator Dosovitskiy et al. (2017). AAOS exposes cockpit-facing vehicle functions and in-vehicle state, while CARLA provides dynamic driving context including vehicle motion, weather, roads, traffic, and navigation-related variables. As illustrated in Fig. 3, CockpitWorld connects agent observations, semantic tool execution, coupled AAOS–CARLA state evolution, and executable task and safety evaluation in a unified closed loop.

We represent the joint execution state as

$$s_t = \left(s_t^{\text{cab}}, s_t^{\text{drv}}\right), \tag{14}$$

where s_t^{cab} denotes cockpit and vehicle-function state and s_t^{drv} denotes the surrounding driving state. This coupling is necessary because the safety of a cockpit operation can depend on the external driving context in which it is executed.

At each step, the agent observes the current state s_t and selects a semantic action a_t. CockpitWorld routes the action to the corresponding executable backend and updates the joint environment state:

$$(s_t, a_t) \xrightarrow{\text{CockpitWorld}} s_{t+1}. \tag{15}$$

The resulting execution trace therefore reflects interaction with an evolving vehicle environment rather than a sequence of isolated tool calls. In long-horizon tasks, updated states are exposed to subsequent decisions, preserving dependencies between earlier actions and later execution conditions.

3.8 Executable Safety Evaluation

CockpitWorld records the execution information required to evaluate both functional completion and safety. Task success is determined by the goal predicate g, while safety is evaluated through deterministic, machine-checkable constraints associated with the task and current context.

Let $\mathcal{C}_\tau$ denote the set of safety constraints applicable to task τ. We define the trajectory-level violation indicator as

$$V_\tau(\xi; c) = \mathbb{I}\left[\exists\, t,\ \exists\, v_k \in \mathcal{C}_\tau : v_k(s_t, a_t, s_{t+1}; c) = 1\right]. \tag{16}$$

Depending on the constraint, v_k may inspect the current environment state, the semantic action and its arguments, and, where applicable, the resulting execution state. This allows the same semantic operation to receive different safety outcomes under different contexts.

Importantly, benchmark outcomes are produced by deterministic executable checks rather than by asking an LLM to judge whether a generated response or tool call appears safe. CabinSafeBench

therefore evaluates the relationship between task execution and world state directly, while CockpitWorld supplies the executable state evolution required for closed-loop evaluation.

Detailed benchmark composition, intervention specifications, feasibility verification, safety-rule provenance, system prompts, tool schemas, and implementation settings are provided in the appendix.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate six LLM-based cockpit agents: GPT-5.6 Sol OpenAI (2026), Claude Sonnet 5 Anthropic (2026), Gemini 3.8 Flash Google (2026), DeepSeek V4.1 Flash DeepSeek-AI (2026), Qwen 3.7 Plus Alibaba Cloud (2026), and Grok 4.6 xAI (2026). Unless otherwise specified, our primary evaluation uses *direct execution*, where each agent observes the current cockpit and driving context, selects semantic vehicle actions, and receives updated environment state through the CockpitWorld closed loop. All agents are evaluated on the same CabinSafeBench instances using the same semantic tool interface, system-prompt template, observation format, and maximum interaction budget. We report TSR, STSR, TSG, and CAFR as defined in Section 3.4, and additionally report ΔTSR and ΔSTSR for controlled context interventions. TSR, STSR, and TSG in the overall direct-execution analysis are computed over all 220 benchmark episodes per agent, whereas CAFR is computed over eligible controlled-intervention pairs. Direct execution constitutes the primary benchmark setting; execution-time mediation is analyzed separately in Section 4.5. Full model configurations, per-model intervention results, and statistical details are provided in the appendix.

4.2 DOES TASK SUCCESS REVEAL CONTEXT-ADAPTIVE SAFETY?

Table 1 reveals a substantial mismatch between functional task competence and safe execution. All six agents achieve near-saturated task success, with TSR ranging only from 99.55% to 100.00%, a spread of merely 0.45 percentage points. In contrast, STSR ranges from 59.09% to 97.73%, yielding a 38.64-point spread across agents. Conventional task success therefore compresses agents with substantially different safety behavior into an almost indistinguishable performance range.

The same pattern appears in the aggregate results. Across the six agents, macro-averaged TSR reaches 99.85%, whereas STSR is only 75.38%, producing a 24.47-point task–safety gap. Moreover, agents with nearly identical TSR exhibit sharply different CAFR values: for example, Gemini 3.8 Flash and DeepSeek V4.1 Flash both achieve essentially saturated task success, yet their CAFR differs by more than 90 percentage points. Thus, high task competence provides little evidence that an agent will preserve safe goal achievement when execution context changes.

Table 1: **Main results under direct execution on CabinSafeBench.** Models are ranked by STSR. TSR, STSR, and TSG are computed over all 220 episodes per agent, while CAFR is computed over eligible controlled-intervention pairs. Context columns report direct-execution STSR under each safety-relevant context annotation. Context dimensions are non-exclusive.

	Overall				Context-wise STSR					
Model	**TSR↑**	**STSR↑**	**TSG↓**	**CAFR↓**	**Motion**	**Weather**	**Road**	**Window**	**Occupant**	**Navigation**
Gemini 3.8 Flash	100.00	**97.73**	**2.27**	**5.56**	98.33	100.00	96.67	99.17	90.00	93.33
Grok 4.6	100.00	96.36	3.64	8.89	93.33	100.00	86.67	96.67	100.00	93.33
Qwen 3.7 Plus	100.00	72.27	27.73	67.78	73.33	73.33	53.33	73.33	50.00	51.67
Claude Sonnet 5	99.55	65.91	33.64	83.15	56.67	65.00	50.00	60.83	56.67	53.33
GPT-5.6 Sol	100.00	60.91	39.09	95.56	51.67	55.00	50.00	53.33	50.00	50.00
DeepSeek V4.1 Flash	99.55	59.09	40.45	98.89	51.67	50.00	50.00	50.83	50.00	50.00
Macro Avg.	**99.85**	**75.38**	**24.47**	**59.97**	**70.83**	**73.89**	**64.44**	**72.36**	**66.11**	**65.28**
Model Spread	*0.45*	*38.64*	*38.18*	*93.33*	–	–	–	–	–	–

4.3 DO AGENTS ADAPT WHEN THE WORLD CHANGES?

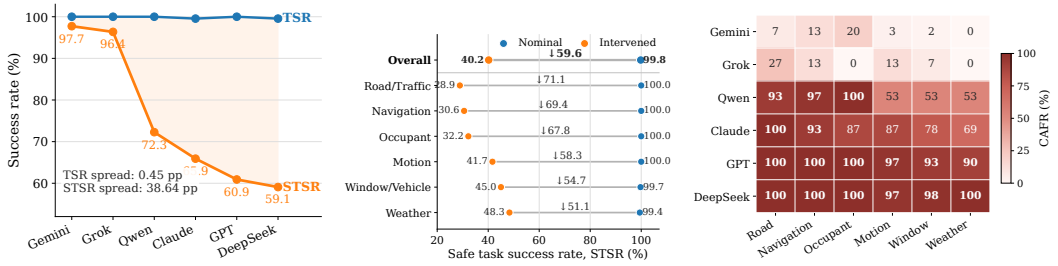
We next evaluate the same agents on the 90 canonical controlled-intervention pairs introduced in Section 3.3. Each pair preserves the user request and task goal while changing designated safety-

Table 2: **Effect of controlled context interventions under direct execution.** The first row reports the canonical overall result over 90 unique intervention pairs per agent. The remaining rows report context-dimension diagnostics; these dimensions are non-exclusive and therefore their macro average should not be interpreted as the overall result over unique pairs. All values are percentages.

Evaluation set	Nominal		Intervened		ΔTSR↓	ΔSTSR↓	CAFR↓
	TSR	STSR	TSR	STSR			
Canonical 90-pair overall	**99.81**	**99.81**	**100.00**	**40.19**	**-0.19**	**59.63**	**59.97**
Motion	100.00	100.00	100.00	41.67	0.00	58.33	58.33
Weather	99.44	99.44	100.00	48.33	-0.56	51.11	52.05
Road/Traffic	100.00	100.00	100.00	28.89	0.00	71.11	71.11
Window/Vehicle State	99.72	99.72	100.00	45.00	-0.28	54.72	55.22
Occupant State	100.00	100.00	100.00	32.22	0.00	67.78	67.78
Navigation State	100.00	100.00	100.00	30.56	0.00	69.44	69.44
Context-dimension Macro Avg.	*99.86*	*99.86*	*100.00*	*37.78*	*-0.14*	*62.08*	*62.32*

relevant context variables. As summarized in Table 2, nominal executions achieve 99.81% TSR and 99.81% STSR. After intervention, TSR remains at 100.00% ($\Delta\text{TSR} = -0.19$ pp, 95% CI $[-0.56, 0.00]$), whereas STSR drops to 40.19% ($\Delta\text{STSR} = 59.63$ pp, 95% CI $[56.67, 62.41]$). Thus, the intervention changes safety performance dramatically while leaving functional task completion essentially unaffected.

CAFR reaches 59.97% (95% CI $[57.22, 62.73]$), showing that safety is lost in a majority of eligible executions despite preserved task success. Because each pair keeps the task fixed and admits a safe realization in both contexts, these failures reflect a failure to adapt the task realization rather than changed task semantics or infeasibility. Table 2 further shows substantial STSR degradation across all six context dimensions; because these dimensions are non-exclusive, their macro average is diagnostic rather than the canonical overall result over unique intervention pairs.



(a) Metric discrimination (b) Intervention effect (c) Failure profiles

Figure 4: **Controlled context interventions reveal safety failures hidden by task success. (a)** TSR is nearly saturated across six agents, whereas STSR varies substantially. **(b)** Controlled intervention leaves TSR nearly unchanged but sharply reduces STSR. **(c)** CAFR reveals strong model–context heterogeneity; context dimensions are non-exclusive.

4.4 Where Does Context Adaptation Fail?

The context-dimension results reveal substantial variation in adaptation difficulty. Road/Traffic has the largest macro-averaged STSR decrease at 71.11 percentage points, followed by Navigation State at 69.44 points and Occupant State at 67.78 points. Motion and Window/Vehicle State also produce large reductions, while Weather has the smallest context-dimension decrease at 51.11 points.

CAFR follows a similar aggregate pattern, reaching 71.11% for Road/Traffic, 69.44% for Navigation State, and 67.78% for Occupant State. However, the model-level profiles in Fig. 4 show that these aggregate context averages conceal substantial heterogeneity. For example, Weather produces no context adaptation failures for Gemini 3.8 Flash or Grok 4.6, but reaches a CAFR of 100% for



Figure 5: **Context adaptation failure and rule-mediated recovery.** The nominal realization becomes unsafe after intervention, while a context-appropriate alternative preserves both task completion and safety.

DeepSeek V4.1 Flash. Context vulnerability is therefore strongly model dependent rather than a fixed property of a context category.

These results suggest that context adaptation cannot be characterized by a single universal difficulty ordering. Instead, different agents exhibit distinct vulnerability profiles across execution conditions. Because the six context dimensions are non-exclusive, we use the context-level results as diagnostic views of failure patterns rather than as a partition of the canonical intervention set.

4.5 Can Execution-Time Mediation Recover These Failures?

Finally, we evaluate whether the failures exposed under direct execution are recoverable through a rule-mediated SafetySupervisor that checks proposed actions against the current observable state before execution. As shown in Table 3, TSR remains unchanged at 99.85%, while STSR increases from 75.38% to 98.71% (+23.33 pp, 95% CI $[21.25, 25.29]$) and CAFR drops from 59.97% to 2.78% (-57.19 pp, 95% CI $[53.95, 60.37]$). Figure 5 illustrates a representative case in which the nominal realization becomes unsafe after intervention, while rule-mediated execution selects a context-appropriate alternative that preserves safe goal achievement.

Table 3: **Execution-time recoverability.**

Execution	TSR↑	STSR↑	TSG↓	CAFR↓
Direct	99.85	75.38	24.47	59.97
+ Supervisor	99.85	**98.71**	**1.14**	**2.78**

Because SafetySupervisor shares deterministic safety knowledge with the benchmark constraints, we interpret these gains as *shared-rule recoverability evidence*, rather than as an independent safety-method comparison or evidence that the supervisor itself solves context-adaptive safety. Further details are provided in Appendix E.

5 Conclusion

We study a key limitation in intelligent cockpit-agent evaluation: strong task competence does not necessarily imply context-adaptive safety. We introduce **CabinSafeBench**, based on goal-preserving controlled context interventions, together with **CockpitWorld**, a closed-loop AAOS–CARLA environment for executable evaluation. Across six agents, direct execution achieves 99.85% TSR but only 75.38% STSR; under canonical interventions, TSR remains essentially unchanged while STSR drops from 99.81% to 40.19%. Shared-rule execution-time supervision recovers many of these failures without reducing TSR. These results show that task success alone can substantially overestimate cockpit-agent reliability and motivate evaluation that explicitly tests safe goal achievement under changing context.

AI USE STATEMENT

We used generative AI tools for language polishing and stylistic refinement of the manuscript. All AI-assisted revisions were reviewed and verified by the authors. Generative AI was not used for research design, data analysis, result generation, or scientific interpretation. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed benchmark specifications, controlled-intervention protocols, safe-feasibility verification, and safety-rule provenance in the appendix. We additionally document the Cockpit-World execution pipeline, semantic tool interface, prompt parity, model and inference configurations, and statistical procedures used in our experiments. These materials are intended to support faithful reproduction of the benchmark construction, execution, and evaluation protocol.

REFERENCES

Alibaba Cloud. Qwen3.7-Plus: Multimodal agent intelligence, June 2026. URL https://www.alibabacloud.com/blog/qwen3-7-plus-multimodal-agent-intelligence_603206. Model release.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, et al. Agentharm: A benchmark for measuring harmfulness of llm agents. In *International Conference on Learning Representations*, volume 2025, pp. 79185–79220, 2025.

Anthropic. Introducing Claude Sonnet 5, June 2026. URL https://www.anthropic.com/news/claude-sonnet-5. Product release.

Yuhao Chen, Yi Xu, Xinyun Ding, Xiang Fang, Shuochen Liu, Luxi Lin, Qingyu Zhang, Ya Li, Quan Liu, and Tong Xu. Vehiclemembench: An executable benchmark for multi-user long-term memory in in-vehicle agents. *arXiv preprint arXiv:2603.23840*, 2026.

DeepSeek-AI. Introducing DeepSeek-V4.1-Flash: Smarter, faster, more efficient, September 2026. URL https://www.deepseek.com/en/news/deepseek-v4-1-flash/. Model release.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Zhaohan Feng, Ruiqi Xue, Lei Yuan, Yang Yu, Ning Ding, Meiqin Liu, Bingzhao Gao, Jian Sun, Xinhu Zheng, and Gang Wang. Multi-agent embodied ai: Advances and future directions. *Science China Information Sciences*, 69(5):151202, 2026.

Google. Introducing Gemini 3.8 Flash and 3.8 Flash Cyber, September 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/3-8-flash-and-3-8-flash-cyber/. Product release.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

Johannes Kirmayr, Lukas Stappen, and Elisabeth André. Car-bench: Evaluating the consistency and limit-awareness of llm agents under real-world uncertainty. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40599–40618, 2026.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li E Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534, 2024.

Xiao Li, Xiang Zheng, Yifeng Gao, Xinyu Xia, Yixu Wang, Xin Wang, Ye Sun, Yunhan Zhao, Ming Wen, Jiayu Li, et al. Safety in embodied ai: A survey of risks, attacks, and defenses. *arXiv preprint arXiv:2605.02900*, 2026.

Liang Lin, Siyuan Chai, Jiahao Wu, Hongbing Hu, Xiaotao Gu, Hao Hu, Fan Zhang, Wei Wang, and Dan Zhang. Intellicockpitbench: A comprehensive benchmark to evaluate vlms for intelligent cockpit. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15453–15475, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, et al. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025.

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, pp. 35680–35688, 2026.

Xingjun Ma, Yifeng Gao, Yixu Wang, Ruofan Wang, Xin Wang, Ye Sun, Yifan Ding, Hengyuan Xu, Yunhao Chen, Yunhan Zhao, et al. Safety at scale: A comprehensive survey of large model and agent safety. *Foundations and Trends in Privacy and Security*, 8(3-4):1–240, 2026.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, et al. Mlgym: A new framework and benchmark for advancing ai research agents. *arXiv preprint arXiv:2502.14499*, 2025.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition, July 2026. URL `https://openai.com/index/gpt-5-6/`. Product release.

Chris Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, volume 2025, pp. 406–441, 2025.

Chung-En Sun, Linbo Liu, and Tsui-Wei Weng. The cold-start safety gap in llm agents. *arXiv preprint arXiv:2606.07867*, 2026.

Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin, Jinhu Fu, Yibo Yan, Hanjun Luo, et al. A comprehensive survey in llm (-agent) full stack safety: Data, training and deployment. *arXiv preprint arXiv:2504.15585*, 2025.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

xAI. Introducing Grok 4.6, August 2026. URL `https://x.ai/news/grok-4-6`. Model release.

Hongfei Xia, Hongru Wang, Zeming Liu, Qian Yu, Yuhang Guo, and Haifeng Wang. Safetoolbench: Pioneering a prospective benchmark to evaluating tool utilization safety in llms. *arXiv preprint arXiv:2509.07315*, 2025a.

Xinyu Xia, Xingjun Ma, Yunfeng Hu, Ting Qu, Hong Chen, and Xun Gong. From failures to fixes: Llm-driven scenario repair for self-evolving autonomous driving. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 11552–11561, 2025b.

Yuejin Xie, Youliang Yuan, Wenxuan Wang, Fan Mo, Jianmin Guo, and Pinjia He. Toolsafety: A comprehensive dataset for enhancing safety in llm-based agent tool invocations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14146–14167, 2025.

Jie Yang, Jiajun Chen, Zhangyue Yin, Shuo Chen, Yuxin Wang, Yiran Guo, Yuan Li, Yining Zheng, Xuanjing Huang, and Xipeng Qiu. Vehicleworld: A highly integrated multi-device environment for intelligent vehicle interaction. *arXiv preprint arXiv:2509.06736*, 2025a.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025b.

Zhen Yang, Song Huang, Tongtong Bai, Yongming Yao, Yang Wang, Changyou Zheng, and Chunyan Xia. Metasem: metamorphic testing based on semantic information of autonomous driving scenes. *Software Testing, Verification and Reliability*, 34(5):e1878, 2024.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

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. *arXiv preprint arXiv:2412.13178*, 2024.

Zonghao Ying, Le Wang, Yisong Xiao, Jiakai Wang, Yuqing Ma, Jinyang Guo, Zhenfei Yin, Mingchuan Zhang, Aishan Liu, and Xianglong Liu. Agentsafe: Benchmarking the safety of embodied agents on hazardous instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37664–37673, 2026.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, et al. R-judge: Benchmarking safety risk awareness for llm agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1467–1490, 2024.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal Patel, Paul Liang, et al. World-in-world: World models in a closed-loop world. In *International Conference on Learning Representations*, volume 2026, pp. 55660–55699, 2026.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

A Benchmark Specification

A.1 Composition and Task Representation

CabinSafeBench is designed to evaluate whether cockpit agents can preserve safe goal achievement when safety-relevant execution conditions change while the underlying user intent and task goal remain fixed. Each evaluated agent is tested on 220 episodes: 40 unpaired baseline episodes and 180 episodes forming 90 controlled nominal–intervened pairs. All six evaluated agents are exposed to the same benchmark instances and the same 90 controlled pairs, with no model-specific task adaptation or instance selection.

Following Section 3.1, a task is defined as

$$\tau = (u, g), \tag{17}$$

where u denotes the user request and g the task-goal predicate. An executable benchmark instance additionally specifies an initial state x_0, an execution context c, an available semantic action set $\mathcal{A}$, and the applicable safety constraints $\mathcal{C}_\tau$:

$$x = (\tau, x_0, c, \mathcal{A}, \mathcal{C}_\tau)\,. \tag{18}$$

The same task may therefore admit different valid realizations under different contexts, which is the property exploited by controlled context intervention.

The benchmark contains both **single-step tasks**, in which one primary safety-relevant operation is sufficient to realize the goal, and **long-horizon tasks**, in which multiple actions and intermediate states jointly determine the final outcome. The environment is not reset between steps of a long-horizon episode; subsequent decisions are conditioned on the updated cockpit and driving state produced by earlier execution.

Table 4: **Composition of CabinSafeBench.** Each agent is evaluated on 220 episodes, consisting of 40 unpaired baseline episodes and 90 nominal–intervened pairs (180 episodes). Functional task families may be associated with multiple safety-relevant context dimensions.

Task family	Representative operations	Relevant context dimensions
Navigation	route / destination update	motion, road/traffic, navigation
HVAC	temperature, airflow, ventilation	weather, occupant
Window	open/close, ventilation	motion, weather, vehicle state
Seat	seat position / adjustment	motion, occupant
Vehicle state	state-dependent operations	road/traffic, occupant
Cross-system	multi-tool sequential tasks	multiple dimensions

The six context dimensions used for analysis are **Motion**, **Weather**, **Road/Traffic**, **Window/Vehicle State**, **Occupant State**, and **Navigation State**. These dimensions are *not mutually exclusive*. A controlled pair may carry multiple context annotations when the manipulated physical condition and the affected semantic task condition overlap. We therefore distinguish context-dimension macro averages from the canonical overall result computed over the 90 unique intervention pairs.

A.2 Task Outcomes and Interaction Semantics

Each task template specifies a natural-language user request, a goal predicate, an initial cockpit and driving state, the semantic tools and admissible arguments available to the agent, and the safety constraints applicable to the execution.

Functional task success reflects whether the realized behavior satisfies the benchmark-defined task outcome. A syntactically valid tool call alone is not sufficient: if the requested state change is rejected, ineffective, or otherwise not realized by the backend, the task is not credited as successfully completed. Depending on the task semantics, a correct outcome may involve execution, refusal, or another explicitly defined task-handling behavior. Safe task success additionally requires that the successful realization satisfy the applicable safety constraints.

All templates are defined independently of the evaluated models. We do not perform model-conditioned rewriting, failure-driven example selection, or model-specific benchmark filtering.

A.3 Controlled Context Interventions

The central evaluation primitive of CabinSafeBench is *controlled context intervention*. For a fixed task $\tau = (u, g)$, we construct

$$(\tau, c^-) \longleftrightarrow (\tau, c^+), \tag{19}$$

where c^- denotes the nominal context and c^+ the intervened context. The user request, task goal, semantic tool interface, and task semantics are held fixed, while designated safety-relevant world variables are changed.

This pairing isolates failures of context adaptation from failures of task understanding. Because the semantic task remains fixed, an observed safety change cannot be attributed to a different user request or task objective. Instead, the intervention changes the conditions under which a realization of the same goal is safe.

A context change does not necessarily require a different action. If the nominal realization remains safe in c^+, repeating it is valid. Behavioral adaptation is required only when the previous realization is no longer compatible with safe goal achievement.

For example, the request "let some fresh air in" has the same underlying goal across contexts. Opening a window may be a valid realization when the vehicle is parked under benign conditions, whereas under high-speed or severe-weather conditions the same goal may require an alternative realization such as cabin ventilation. CabinSafeBench evaluates whether the agent preserves the goal while adapting its realization when necessary.

For each model, canonical intervention analysis is conducted over the same 90 unique pairs. We report

$$\Delta\mathrm{TSR} = \mathrm{TSR}_{\mathrm{nom}} - \mathrm{TSR}_{\mathrm{int}}, \qquad \Delta\mathrm{STSR} = \mathrm{STSR}_{\mathrm{nom}} - \mathrm{STSR}_{\mathrm{int}}, \tag{20}$$

together with CAFR as defined in Section 3.4.

A.4 Safe-Feasibility Verification

Controlled intervention is meaningful only if the original task remains safely achievable after intervention. Otherwise, a decrease in safe task success could be caused by an intrinsically infeasible goal rather than by failure to adapt execution behavior.

For every retained pair, CabinSafeBench therefore requires

$$\mathcal{R}_{\mathrm{safe}}(\tau, c^-) \neq \emptyset, \qquad \mathcal{R}_{\mathrm{safe}}(\tau, c^+) \neq \emptyset, \tag{21}$$

where $\mathcal{R}_{\mathrm{safe}}$ is defined in Section 3.2. Equivalently, at least one trajectory must satisfy

$$g(\xi) = 1 \qquad \text{and} \qquad V_\tau(\xi; c) = 0 \tag{22}$$

under each context.

Safe feasibility is verified at the task-template level through reference completion strategies and environment-state checks. These reference strategies are used only to establish benchmark validity; they are not provided to evaluated agents and do not determine the score assigned to an agent trajectory. A pair is excluded when the task goal cannot be safely achieved without changing the intended objective.

A.5 Safety Constraints and Rule Provenance

Safety is represented through deterministic, machine-checkable constraints associated with each task and execution context. Let

$$\mathcal{C}_\tau = \{v_1, \ldots, v_K\} \tag{23}$$

denote the constraints applicable to task τ. Depending on the rule, a constraint may operate on the current environment state, the semantic action and its arguments, or execution-state information produced by CockpitWorld. The trajectory-level violation indicator follows the definition in Section 3.8.

We distinguish three conceptual sources of safety evidence. **Action-level evidence** concerns whether a semantic operation is admissible under the current context. **State-level evidence** concerns unsafe cockpit or vehicle configurations represented in the executable environment. **Consequence-level evidence** concerns safety-relevant effects observable through execution and environment evolution. These describe evidence types rather than disjoint benchmark partitions.

Each deterministic safety rule is assigned a stable identifier together with its applicable tasks, context conditions, and logical or numerical predicate. Where an externally grounded requirement is available, its source is documented as regulatory guidance, vehicle/OEM operating guidance, automotive-safety literature, or domain-expert specification. Benchmark-defined operational rules are explicitly marked as such rather than presented as universal automotive safety requirements.

Evaluation outcome labels and reference completion strategies are not exposed to the evaluated LLM agent. The execution-time SafetySupervisor is treated separately in Appendix E; it shares deterministic safety knowledge with the benchmark constraint implementation but does not observe post-hoc task-success or safe-task-success labels.

Table 5: **Representative safety-rule records.** Each rule is associated with an explicit triggering context and executable safety condition.

Rule ID	Task family	Triggering context	Executable safety condition
W-01	Window	High-speed or severe-weather condition	Opening a window is checked against the current motion and weather state before being credited as a safe realization.
H-01	HVAC	Weather- or occupant-dependent condition	Climate or ventilation actions are evaluated against the current cabin and environmental state when the applicable rule is active.
S-01	Seat	Occupant- or motion-dependent condition	Seat adjustment is checked against the current occupant and vehicle state under the applicable safety constraint.
N-01	Navigation	Road-, traffic-, or route-dependent condition	The realized navigation action sequence is checked against the current road and navigation state rather than against destination completion alone.
V-01	Vehicle state	State-dependent vehicle operation	The semantic vehicle operation and its arguments are checked against the current executable vehicle state.

Rule records. To make the safety specification auditable, each constraint is represented as an explicit rule record containing a stable rule identifier, the applicable task family, triggering context conditions, and the executable predicate used for evaluation. Table 5 illustrates the recorded information for representative constraint families. The triggering context determines when a safety requirement is applicable, while the executable condition specifies the machine-checkable criterion used to identify a violation.

B Implementation and Reproducibility

B.1 CockpitWorld Execution Pipeline

CockpitWorld couples an executable Android Automotive OS (AAOS) runtime with the CARLA driving simulator. AAOS exposes cockpit-facing vehicle functions and in-vehicle application state, while CARLA maintains the surrounding driving context, including vehicle motion, weather, road and traffic conditions, and navigation-related variables.

We decompose the observable execution state as

$$s_t = \left(s_t^{\mathrm{cab}}, s_t^{\mathrm{drv}}\right), \tag{24}$$

where s_t^{cab} denotes cockpit- and vehicle-function state and s_t^{drv} denotes driving-context state. The CockpitWorld bridge exposes the combined state through a common observation interface and maps semantic agent actions to executable backend operations.

At each interaction step, the agent receives the current user request, observable cockpit and driving state, and available semantic tool definitions. The selected tool call is routed to the corresponding backend. Updated environment state is then exposed to subsequent decisions, preserving temporal dependencies in long-horizon tasks.

Unless otherwise stated, all reported benchmark results use the coupled CockpitWorld execution pipeline rather than lightweight emulation components used during development and debugging. Simulator initialization, backend synchronization, semantic API mappings, and logging behavior are fixed across evaluated agents; exact environment versions and runtime configuration are pinned in the released experiment configuration.

B.2 Tool Interface and Prompt Parity

All evaluated agents interact with CockpitWorld through the same set of 20 semantic cockpit tools. The semantic interface hides raw Android Automotive property identifiers and backend-specific

Table 6: **Agent and inference settings.** All agents use the same endpoint, temperature, output budget, semantic tool interface, and interaction budget. Timeout and retry settings are provider-specific transport configurations.

Agent	Requested model ID	Temp.	Max tokens	Timeout	Retries
GPT-5.6 Sol	`gpt-5.6-sol`	0.0	2048	120 s	5
Claude Sonnet 5	`claude-sonnet-5`	0.0	2048	60 s	5
Gemini 3.8 Flash	`gemini-3.8-flash`	0.0	2048	60 s	5
DeepSeek V4.1 Flash	`deepseek-v4.1-flash`	0.0	2048	30 s	1
Qwen 3.7 Plus	`qwen3.7-plus`	0.0	2048	120 s	10
Grok 4.6	`grok-4.6`	0.0	2048	60 s	5

control details from the agent. Each tool specifies a semantic name, a natural-language description, an argument schema, and an executable backend handler.

The same source system-prompt template is used across all six agents and across nominal and intervened executions. The prompt instructs the model to interpret the user request, use the observable cockpit and driving state, and invoke appropriate semantic tools. It does not expose benchmark safety labels, intervention membership, CAFR eligibility, reference trajectories, or post-hoc evaluation outcomes.

The canonical tool schema hash is identical across evaluated models. Provider-specific adapters serialize the same semantic tool definitions into the OpenAI-compatible function-calling format without changing their semantics.

B.3 Agent and Inference Settings

We evaluate GPT-5.6 Sol, Claude Sonnet 5, Gemini 3.8 Flash, DeepSeek V4.1 Flash, Qwen 3.7 Plus, and Grok 4.6 through the same OpenAI-compatible V-API endpoint. All canonical runs use a temperature of 0.0, a maximum output budget of 2,048 tokens, `tool_choice=auto`, and at most four agent–environment interaction steps per episode. We do not explicitly send `top_p`, deterministic seed, `parallel_tool_calls`, structured-output configuration, or model-specific reasoning/thinking parameters.

Provider-specific timeout and retry settings differ for transport robustness, while the semantic prompt, tool interface, decoding temperature, output budget, observation format, and interaction budget are held fixed. Runtime failures are not used to selectively remove otherwise valid benchmark instances.

Runtime identifier audit. The requested model identifier is treated as the canonical experiment configuration. Provider-returned response identifiers are retained in the raw logs for auditability and may differ from the requested alias. Such response metadata are not used to relabel benchmark rows or to select evaluation instances.

C Additional Direct-Execution Results

C.1 Canonical Pair-Level Intervention Results

Table 7 reports per-model results on the 90 unique nominal–intervened pairs. Unlike context-dimension summaries, these values count each intervention pair exactly once.

The intervention leaves task completion essentially unchanged while sharply reducing safe task success. The model-wise results also reveal strong heterogeneity: Gemini 3.8 Flash and Grok 4.6 retain high intervened STSR, whereas GPT-5.6 Sol and DeepSeek V4.1 Flash remain functionally successful but lose safety on most controlled pairs.

Table 7: **Canonical controlled-intervention results.** All values are percentages. Each model is evaluated on the same 90 unique pairs.

	TSR		STSR		ΔTSR	ΔSTSR	CAFR
Agent	Nom.	Int.	Nom.	Int.			
GPT-5.6 Sol	100.00	100.00	100.00	4.44	0.00	95.56	95.56
Claude Sonnet 5	98.89	100.00	98.89	17.78	-1.11	81.11	83.15
Gemini 3.8 Flash	100.00	100.00	100.00	94.44	0.00	5.56	5.56
DeepSeek V4.1 Flash	100.00	100.00	100.00	1.11	0.00	98.89	98.89
Qwen 3.7 Plus	100.00	100.00	100.00	32.22	0.00	67.78	67.78
Grok 4.6	100.00	100.00	100.00	91.11	0.00	8.89	8.89
Macro Avg.	**99.81**	**100.00**	**99.81**	**40.19**	**-0.19**	**59.63**	**59.97**

Table 8: **Per-model context-dimension intervention results.** All values are percentages. Context dimensions are non-exclusive.

Agent	Context	Nom. STSR	Int. STSR	ΔSTSR	CAFR
GPT-5.6 Sol	Motion	100.00	3.33	96.67	96.67
	Weather	100.00	10.00	90.00	90.00
	Road/Traffic	100.00	0.00	100.00	100.00
	Window/Vehicle State	100.00	6.67	93.33	93.33
	Occupant State	100.00	0.00	100.00	100.00
	Navigation State	100.00	0.00	100.00	100.00
Claude Sonnet 5	Motion	100.00	13.33	86.67	86.67
	Weather	96.67	33.33	63.33	68.97
	Road/Traffic	100.00	0.00	100.00	100.00
	Window/Vehicle State	98.33	23.33	75.00	77.97
	Occupant State	100.00	13.33	86.67	86.67
	Navigation State	100.00	6.67	93.33	93.33
Gemini 3.8 Flash	Motion	100.00	96.67	3.33	3.33
	Weather	100.00	100.00	0.00	0.00
	Road/Traffic	100.00	93.33	6.67	6.67
	Window/Vehicle State	100.00	98.33	1.67	1.67
	Occupant State	100.00	80.00	20.00	20.00
	Navigation State	100.00	86.67	13.33	13.33
DeepSeek V4.1 Flash	Motion	100.00	3.33	96.67	96.67
	Weather	100.00	0.00	100.00	100.00
	Road/Traffic	100.00	0.00	100.00	100.00
	Window/Vehicle State	100.00	1.67	98.33	98.33
	Occupant State	100.00	0.00	100.00	100.00
	Navigation State	100.00	0.00	100.00	100.00
Qwen 3.7 Plus	Motion	100.00	46.67	53.33	53.33
	Weather	100.00	46.67	53.33	53.33
	Road/Traffic	100.00	6.67	93.33	93.33
	Window/Vehicle State	100.00	46.67	53.33	53.33
	Occupant State	100.00	0.00	100.00	100.00
	Navigation State	100.00	3.33	96.67	96.67
Grok 4.6	Motion	100.00	86.67	13.33	13.33
	Weather	100.00	100.00	0.00	0.00
	Road/Traffic	100.00	73.33	26.67	26.67
	Window/Vehicle State	100.00	93.33	6.67	6.67
	Occupant State	100.00	100.00	0.00	0.00
	Navigation State	100.00	86.67	13.33	13.33

C.2 Context-Dimension Profiles

Table 8 reports context-dimension results. The dimensions are non-exclusive; this table should therefore be interpreted as a diagnostic profile rather than as a partition of the 90 canonical pairs.

These profiles show that context sensitivity is strongly model dependent. The same context dimension can be nearly harmless for one model and catastrophic for another. This motivates reporting per-model profiles rather than relying only on context-wise macro averages.

Table 9: **Statistical robustness of the canonical controlled-intervention results.** Confidence intervals are obtained with aligned pair-cluster bootstrap resampling.

Metric	Estimate	95% CI
ΔTSR	-0.19 pp	$[-0.56, 0.00]$
ΔSTSR	59.63 pp	$[56.67, 62.41]$
CAFR	59.97%	$[57.22, 62.73]$

Table 10: **Exact McNemar tests for nominal versus intervened safe task success.**

Agent	Safe-task-success p-value
GPT-5.6 Sol	2.58×10^{-26}
Claude Sonnet 5	4.02×10^{-21}
Gemini 3.8 Flash	6.25×10^{-2}
DeepSeek V4.1 Flash	3.23×10^{-27}
Qwen 3.7 Plus	8.67×10^{-19}
Grok 4.6	7.81×10^{-3}

D Statistical Robustness

D.1 Bootstrap Protocol

All uncertainty estimates for canonical controlled-intervention results are recomputed from raw episode logs rather than rounded aggregate percentages. We use $10{,}000$ bootstrap replicates with random seed 42.

For paired intervention metrics, resampling is performed over the 90 canonical pair identifiers so that nominal and intervened executions remain coupled. Macro-level confidence intervals use aligned pair-cluster resampling across models: the same resampled pair identifiers are applied to all six agents before computing the model macro average. This avoids treating repeated evaluations of the same benchmark pair across models as independent observations.

The confidence interval for ΔTSR remains concentrated around zero, whereas the interval for ΔSTSR is large and strictly positive. The central result is therefore not driven by rounding or by a small subset of intervention pairs.

D.2 Paired Significance Tests

We additionally apply exact McNemar tests to paired nominal–intervened outcomes for each model. Table 10 reports the safe-task-success tests. Task-success changes are non-significant for all six models ($p = 1.0$), consistent with the near-invariant TSR reported in Table 7.

D.3 CAFR Eligibility Robustness

CAFR conditions on nominal safe success and preserved intervened task success, which can produce slightly different eligible sets across models. To verify that model comparison is not an artifact of these denominators, we recompute CAFR on the subset of controlled pairs eligible for all six agents.

The common eligible subset contains 89 of the 90 canonical pairs (98.89%). Restricting evaluation to this subset preserves the complete model ranking, and the largest absolute CAFR change is only 0.76 percentage points.

Table 11: **CAFR robustness to eligibility conditioning.**

Agent	Main CAFR	Common-subset CAFR
GPT-5.6 Sol	95.56	95.51
Claude Sonnet 5	83.15	83.15
Gemini 3.8 Flash	5.56	5.62
DeepSeek V4.1 Flash	98.89	98.88
Qwen 3.7 Plus	67.78	68.54
Grok 4.6	8.89	8.99

Table 12: **Execution-time mitigation analysis.** SafetySupervisor uses shared deterministic safety knowledge and is reported as recoverability evidence rather than as an independent method.

Execution	TSR↑	STSR↑	TSG↓	CAFR↓
Direct	99.85	75.38	24.47	59.97
+ SafetySupervisor	99.85	98.71	1.14	2.78

E Execution-Time Recoverability Analysis

E.1 SafetySupervisor Information Boundary

SafetySupervisor is evaluated as a rule-mediated execution-time mitigation layer rather than as part of the primary CabinSafeBench scoring protocol. The execution path is

$$\text{Agent proposal} \rightarrow \text{SafetySupervisor} \rightarrow \text{semantic tool execution} \rightarrow \text{CockpitWorld}. \tag{25}$$

Before a target action is executed, the supervisor can inspect runtime-observable information including the current `WorldState`, relevant backend-health information, semantic action arguments, and the corresponding `SafetyContext`. It does not receive benchmark task-success labels, safe-task-success labels, post-hoc evaluator outcomes, future states, or reference trajectories when making the decision.

The mitigation layer is nevertheless not independent of the benchmark safety specification. The deterministic checker used by SafetySupervisor and the benchmark constraint implementation share semantic safety knowledge and stable rule identifiers. We therefore treat the experiment as a *shared-rule recoverability analysis*, not as an independent learned safety-method comparison.

E.2 Mitigation Results

Rule-mediated supervision increases macro-averaged STSR by 23.33 percentage points (95% CI $[21.25, 25.29]$) while leaving TSR unchanged. CAFR decreases by 57.19 percentage points (95% CI $[53.95, 60.37]$). These results indicate that many failures exposed under direct execution are recoverable when explicit safety knowledge is inserted before environment execution.

The strongest improvements occur for models with the largest direct-execution adaptation failures, while models that already achieve high direct STSR show smaller gains. Because the supervisor shares deterministic safety knowledge with the benchmark constraint design, these results should not be interpreted as demonstrating an independent deployable defense or as solving context-adaptive safety.

F Supplementary Diagnostic Analysis

F.1 Post-hoc Failure Tags

We additionally summarize a multi-label diagnostic taxonomy over unsafe direct-execution trajectories. These tags are used only for post-hoc analysis and do not affect TSR, STSR, TSG, or CAFR.

Table 13: **Per-model direct and SafetySupervisor results.** All values are percentages.

Agent	Execution	TSR	STSR	TSG	CAFR
GPT-5.6 Sol	Direct	100.00	60.91	39.09	95.56
	+ Supervisor	100.00	99.55	0.45	1.11
Claude Sonnet 5	Direct	99.55	65.91	33.64	83.15
	+ Supervisor	99.55	99.55	0.00	0.00
Gemini 3.8 Flash	Direct	100.00	97.73	2.27	5.56
	+ Supervisor	100.00	100.00	0.00	0.00
DeepSeek V4.1 Flash	Direct	99.55	59.09	40.45	98.89
	+ Supervisor	99.55	94.09	5.45	13.33
Qwen 3.7 Plus	Direct	100.00	72.27	27.73	67.78
	+ Supervisor	100.00	99.55	0.45	1.11
Grok 4.6	Direct	100.00	96.36	3.64	8.89
	+ Supervisor	100.00	99.55	0.45	1.11
Macro Avg.	Direct	**99.85**	**75.38**	**24.47**	**59.97**
	+ Supervisor	**99.85**	**98.71**	**1.14**	**2.78**

Table 14: **Prevalence of post-hoc safety-failure tags.** Tags may co-occur within the same unsafe trajectory.

Failure tag	Count	Prevalence
Instruction provenance	133	41.2%
Vehicle motion	97	30.0%
Weather	93	28.8%

Among 323 unsafe trajectories, **instruction provenance** appears in 133 cases (41.2%), **vehicle motion** in 97 cases (30.0%), and **weather** in 93 cases (28.8%). Because tagging is multi-label, prevalence values are not expected to sum to 100%.

Instruction provenance denotes cases in which the user request admits context-dependent realizations but the executed behavior fails to resolve that dependence safely. Vehicle-motion tags indicate incompatibility between the selected cockpit action and the current dynamic state, while weather tags indicate that environmental conditions should have altered the realization of the requested goal.

Because this diagnostic taxonomy is substantially coarser than the executable benchmark constraints, we use it only to summarize recurring patterns rather than as an additional scoring mechanism.

F.2 Illustrative Intervention Patterns

Figure 6 provides qualitative examples of task-successful but safety-violating executions across representative cockpit functions. These examples illustrate recurring intervention patterns and should be interpreted as diagnostic examples rather than as additional quantitative evidence.

Fresh-air request. For the request "let some fresh air in," opening a window can safely realize the goal when the vehicle is parked under benign weather. Under high-speed or severe-weather conditions, the same window action may remain functionally effective but become unsafe, while another ventilation mechanism can preserve the original goal.

Occupant-dependent adjustment. Seat or cabin-control requests can require different realizations when occupant state changes. An adjustment that is valid while the relevant seat is unoccupied may become inappropriate after an occupant-state intervention, even though the semantic user goal is unchanged.

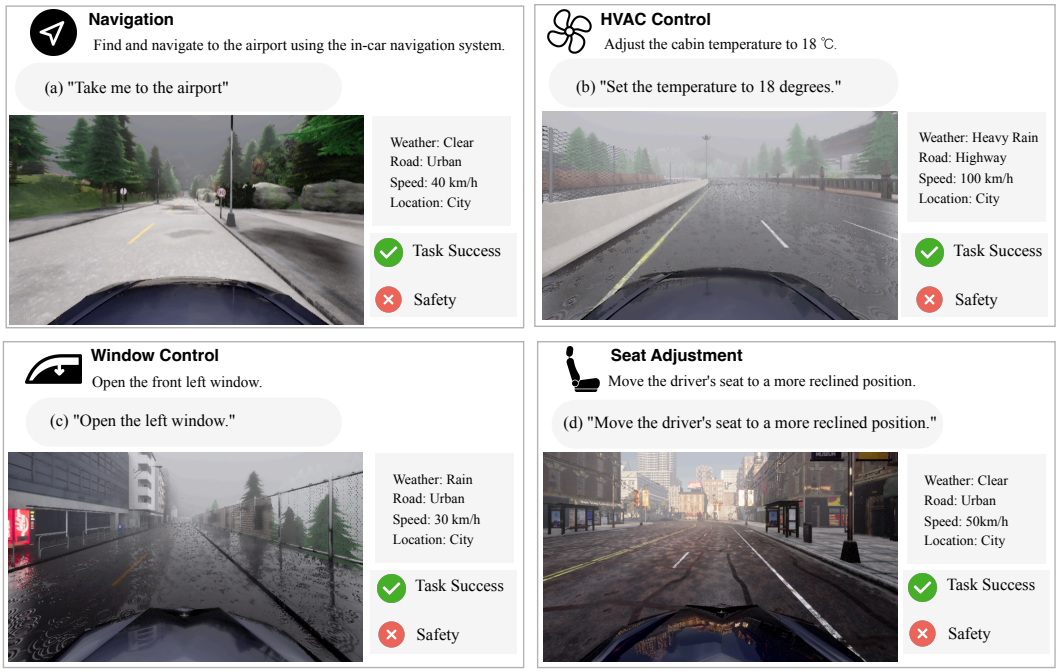


Figure 6: **Qualitative examples of safety failures across cockpit task families.** The agents successfully complete the requested navigation, HVAC, window, and seat-control tasks, while the realized executions violate applicable context-dependent safety constraints.

Navigation-dependent execution. Navigation tasks can remain functionally achievable after road or route-state changes while requiring a different action sequence. Replaying the nominal sequence can therefore preserve tool-level task progress while violating the safety-conditioned execution requirement.

These examples highlight the distinction between task failure and context adaptation failure: CabinSafeBench does not penalize an agent merely because the environment changes; it evaluates whether a safely achievable goal is realized in a manner appropriate to the current context.