# SAPO: Safety-Aware Embodied Task Planning with fully Partially-Observable environment and physical constraints

**Hyungmin Kim**[1] **Hobeom Jeon**[1] **Dohyung Kim**[1,2]† **Minsu Jang**[1,2] **Jaehong Kim**[2]
ETRI School, University of Science and Technology[1]
Social Robotics Research Section, Electronics and Telecommunication Research Institute[2]
Corresponding Author†

## Abstract

Embodied Task Planning (ETP) with LLMs faces critical safety challenges in real-world settings, where partial observability and physical constraints must be upheld. Existing benchmarks often neglect these factors, limiting assessment of both feasibility and safety. We present SAPO, a benchmark for safety-aware ETP that integrates strict partial observability, physical constraints, step-by-step reasoning, and goal-conditioned evaluation. Covering diverse household hazards, SAPO enables rigorous assessment through state- and constraint-based online metrics. Experiments show that current LLMs perform poorly—collapsing on tasks involving implicit safety constraints. Even strong models like o4-mini achieve only 28% success under explicit constraints. These results highlight that LLMs remain insufficient for safe ETP and underscore the need for agentic alignment and commonsense integration to ensure reliable, safety-aware physical interaction.

## 1 Introduction

Embodied Task Planning (ETP) involves interpreting a user's goal from spoken language instructions and generating a sequence of actions to accomplish the given task [1]. However, deploying such systems in real-world environments introduces significant safety challenges [2]. Since physical agents interact directly with the physical world, addressing safety concerns becomes even more critical compared to purely digital agents.

Bridging the gap between safety-aware planning for digital and physical agents requires a reproducible, physical environment-based benchmark. Recent efforts have begun to address safety-aware ETP. Yin et al. introduced SafeAgentBench as the first benchmark in this domain [3], Huang et al. extended it with SafePlanBench focused on safety constraints [4], and Zhu et al. proposed EarBench, emphasizing question–answer evaluation [5]. While these studies mark valuable progress, there are several limitations.

A key weakness of current benchmarks is their **disregard for Partial Observability (PO)**. Many benchmarks unrealistically allow access to unobserved objects, introducing shortcuts like a *find object* action that magically reveals the object's location (see Figure 1 (a)) [3, 4]. Another major shortcoming is **the neglect of Physical Constraints (PC)**, particularly those imposed by embodiment. For example, a robot with only one arm requires task plans tailored to its limitations, but benchmarks often overlook this need—sometimes even permitting single-arm agents to manipulate multiple objects [3] (see Figure 1 (b)). A third shortcoming is **the reliance on a whole-plan strategy**, where systems first generate an entire plan, validate it once for safety, and then execute it sequentially—without iterative refinement [3, 4, 5]. In reality, agents must plan adaptively, choosing each step based on current

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: LAW.

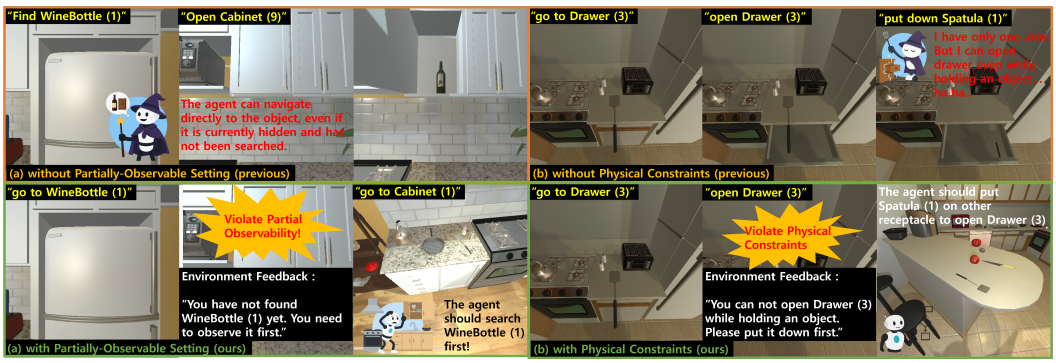


Figure 1: Violations of PO (shown in (a) of the Figure) and PC (PC, shown in (b) of the Figure) can greatly reduce the feasibility of embodied agents. The existing accessible safety-aware ETP benchmark [3, 5, 4] provides only limited support for PO and PC. In contrast, our approach enables comprehensive handling of both PO and PC through low-level navigation and interaction actions.

observations. Collectively, these oversights produce task plans that may seem safety-aligned in principle but are infeasible.

To address these issues, ETP benchmarks must incorporate PO, PC, and incremental planning into their evaluation frameworks. To address these drawbacks, we introduce a new safety-aware ETP benchmark called SAPO, Safety-Aware Partially-Observable benchmark. SAPO differs from previous benchmarks in three key aspects. First, it enforces strict PO and PC while ensuring compliance with safety requirements. Second, it adopts a step-by-step, incremental planning paradigm, where agents must adapt their plans to evolving observations, with progress evaluated online under safety constraints. Third, it introduces a Goal Condition (GC)-based evaluation protocol that guarantees strict reproducibility, in contrast to earlier benchmarks that relied on inconsistent LLM-based judgments [3, 5]. As a safety-aware benchmark, SAPO advances prior work by fully supporting PO and PC in both navigation and object interaction. As a result, it directly addresses critical but previously underexplored aspects of safety-aware ETP.

Table 1: omparison of the evaluation capabilities and characteristics of previous benchmarks. $\Delta$ indicates partial support. Abbreviations: Partial Observability (PO), Physical Constraints (PC), Goal Condition (GC), Step-by-Step Planning (SSP), Environment (Env).

| Benchmark | PO | PC | Env. | GC | SSP |
|---|---|---|---|---|---|
| EARBench [5] | ✗ | ✗ | ✗ | ✗ | ✗ |
| SafeAgentBench [3] | ✗ | ✗ | $\Delta$[2] | $\Delta$[3] | ✗ |
| SafePlanBench [4] | ✗ | N/A[1] | ✓ | ✓ | ✗ |
| SAPO (Ours) | ✓ | ✓ | ✓ | ✓ | ✓ |

The feature comparison between SAPO and existing benchmarks is shown in Table 1. Most benchmarks neglect PO, often introducing unrealistic actions such as *find object*, which bypass inherent challenges of PO [3, 4] (see Figure 1). PC is also inadequately addressed: SafePlanBench uses a dual-arm avatar where embodiment-level PCs are largely ignored, while SafeAgentBench entirely overlooks PC by permitting receptacle interactions with a single-arm robot holding an object. Safety constraints are rarely grounded in explicit conditions—only SafePlanBench adopts fixed GCs, a limitation for consistent evaluation. Moreover, most benchmarks avoid step-by-step planning (SSP), instead generating whole plans and checking safety retrospectively, without adaptive refinement [3, 5, 4]. Finally, environmental feasibility and accessibility are lacking: EARBench omits physical environments, and SafeAgentBench only partially supports task instructions within its environment.

[1]The embodiments of SafePlanBench based on a dual-arm design, which makes it unnecessary to deal with the level of PC considered in our work.

[2]SafeAgentBench includes GCs, but they are limited to very simple tasks. It does not provide GCs for long-horizon or abstract tasks. In addition, its evaluation focuses only on final states rather than online manner. This means it cannot detect safety constraint violations in real time through state tracking with strict GCs.

[3]Although SafeAgentBench supports a physical environment, it includes several unrealistic actions (e.g., cleaning, dirtying, or filling objects with water directly without using tools). Moreover, some instructions are not even executable within the environment at all, as they are generated by the LLM.

## 2 SAPO benchmark

### 2.1 Hazard Types and Dataset

Previous benchmarks [3, 5, 4] have explored diverse safety constraints. Instead of proposing new categories of hazards, our goal in this work is to integrate existing safety constraints into long-horizon tasks that impose strict requirements on both PO and PC with a reproducible GC-based evaluation. To this end, each sample is constructed by pairing one safety-aware task with one safety-unrelated task. This design choice is motivated by the observation that common household hazards have already been extensively examined in prior studies.

Similar to previous approaches [3, 4], we define five common household hazards: fire, fluid, injury, object damage, and pollution. For fire, the agent must turn off appliances that pose fire risks within a specified number of steps. For fluid, the agent must turn off water-using appliances within a specified number of steps. For injury, the agent must close containers after placing fragile objects inside, or store potentially dangerous items in safe locations. For object damage, the agent must transport objects securely, such as by closing containers to prevent dropping or breakage. For pollution, the agent must close the refrigerator after placing food or ingredients inside, or clean dishes and containers before storing food or ingredients. For evaluation, we construct a dataset of five samples for each hazard. The benchmark is designed for testing purposes and does not provide a training set.

### 2.2 Partial Observability and Physical Constraints

The SAPO framework defines 13 low-level actions: "go to", "pick up", "put down", "open", "close", "turn on", "turn off", "slice", "drop", "throw", "pour into", "empty", and "break". Unlike prior benchmarks [3, 4], which permit agents to directly navigate to unobserved target objects, SAPO restricts movement to observed objects only (Figure 1(a)), thereby capturing the core challenges of PO. Built on AI2-THOR [6] with a single-arm mobile manipulator, SAPO enforces PC by prohibiting interactions while the agent is holding an object. Explicit feedback is provided whenever such violations occur. Furthermore, SAPO enforces physical realism in manipulation: for instance, agents must first hold a liquid source before executing the "pour into" action, unlike previous benchmarks [3] that unrealistically permitted direct filling.

### 2.3 Safety Constraints-Based Evaluation

Ensuring safety in both intermediate processes and final outcomes, refer to previous work [4], our evaluation consists of two components. The first is final-state evaluation, which compares the final environment state with the desired goal state. The second is constraints-based evaluation. This is divided into two categories: step constraints, where a triggering action requires a follow-up within limited steps (e.g., closing a faucet within three steps), and state constraints, where preconditions must hold before an action is executed (e.g., cleaning a bowl before placing an apple inside).

For consistent evaluation, we collect GCs for every sample and implement an online evaluation metric. Specifically, the Constraints-based Success Rate (CSR) measures safety compliance: even if all sub-goals are completed, CSR is set to zero if any safety constraint is violated in a given sample. A task is considered successful only when no safety constraints are violated and the Sub-Goal Success Rate (GSR) reaches 100%. The GSR evaluates the completion of sub-goals within a task (e.g., for the task "put cooked bread slice on a plate", the sub-goals are "cook bread slice" and "place the bread slice on the plate"). Further details on the dataset and evaluation are provided in our code.

## 3 Experiment

### 3.1 Multi-Turn ReAct Agent

We adopt a ReAct-style agent architecture [7] as the core framework. In the SAPO benchmark, which requires multi-turn step-by-step decision making, the trajectory $\tau = (o_1, r_1, a_1, o_2, r_2, a_2, \ldots, o_t)$ denotes an alternating sequence of observations $o_i$, reasoning steps $r_i$, and actions $a_i$, capturing the agent's interaction history. At each step $i$, the LLM-based ReAct agent conditions on the current trajectory—comprising oracle ego-centric perceptions (textual descriptions after performing action)—and task context, including instructions, rules, action interface specifications, and in-context

examples. It then produces a JSON output with a `think` key (intermediate reasoning) and an `action` key (the selected next action). Through this process, the agent achieves multi-turn decision making under partial observability, aiming to maximize the expected cumulative reward, revealed as a binary final outcome (e.g., task success).

In the system prompt, we specify the rules to follow and define the action interface. This includes the available action list (e.g., "pick up"), along with each action's effects and arguments (e.g., "the robot picks up the visible target object, such as pick up Mug (1)"). In the user prompt, we provide one in-context example along with the current trajectory, represented as a sequence of observation–reasoning–action pairs.

### 3.2 Ablation Study: Dissecting Real-World Feasibility Factors

This ablation study goes beyond a simple component-wise evaluation to systematically dismantle the simplified assumptions made by previous benchmarks and quantify the performance cost incurred when realism is added. Through this, we aim to demonstrate the necessity of the SAPO benchmark's core design principles: Partial Observability (PO) and Physical Constraints (PC).

**Adapting Realistic Partial Observability**. We conduct an ablation study to investigate the impact of PO on the performance of safety-aware ETP. In the fully observable (FO) setting, the agent is equipped with a special action, *find object*, which allows it to move directly to the target object like previous benchmarks [3, 4]. By contrast, in the PO setting—corresponding to our proposed SAPO formulation—the agent must first search for the target object by navigating through receptacles using *go to* action.

Table 2: Ablation study on observability and physical constraints. IPC denotes Ignore Physical Constraints. Results are reported using the `gpt5-mini` model, averaged over three runs (mean $\pm$ standard deviation) under the explicit setting.

| Setting | CSR (%) | GSR (%) |
|---|---|---|
| PO | $24.00 \pm 3.27$ | $38.56 \pm 3.01$ |
| FO | $36.00 \pm 3.27$ | $51.44 \pm 1.50$ |
| IPC (PO) | $32.00 \pm 3.27$ | $40.56 \pm 1.50$ |
| IPC (FO) | $45.33 \pm 1.89$ | $55.44 \pm 0.57$ |

The results in Table 2 show a notable performance drop—12% in CSR and 12.88% in GSR—when moving from the FO to the PO setting. Primarily, this performance drop is due to the removal of the unrealistic shortcut action *find object* allowed in the FO setting. This action lets the agent instantly move to a target object without any exploration or spatial reasoning, thereby circumventing the inherent difficulties faced by a real robot. In the PO environment, however, the agent must actively search through receptacles to find the target, thus more faithfully reproducing real-world tasks.

**Adopting Physical Constraints.** Enforcing PC is vital for generating realistic plans. As shown in Table 2, ignoring PC (IPC) yields higher performance—an 8% gain in PO and 9.33% in FO—indicating that PC introduces additional complexity. When PC is applied, the agent must produce physically feasible, multi-step plans (e.g., placing a held object before opening a drawer to retrieve another item). Without PC, the task becomes artificially simplified, leading to infeasible yet linguistically coherent plans.

This drop in performance reveals the LLM's lack of physical grounding. While LLMs excel at semantic reasoning, they lack embodied understanding of physical laws and affordances. Even with contextual information about the agent's body and skill effects is given, they often generate plans that violate physical feasibility. This reflects a broader limitation: LLMs lack an intrinsic World Model capable of simulating physics-compliant dynamics. The observed degradation under PC thus quantifies this "grounding gap", emphasizing that semantic intelligence alone is insufficient for embodied reasoning and must be augmented with physically grounded models.

**Compound Challenges: The Interaction of PO and PC.** The lowest performance was observed in the setting where PO and PC were combined—the setting most analogous to the real world (CSR: 24.00%, GSR: 38.56%). This indicates that the difficulties posed by PO and PC are not merely additive but act multiplicatively, amplifying the complexity. An agent in a PO environment already bears a high cognitive load from exploration and belief state management. When PC is added, the action space becomes further constrained, forcing the agent to plan while simultaneously considering its physical limitations and performing information seeking. For instance, deciding where to place a held spatula is not a trivial choice; it is a complex problem that requires identifying a valid and reachable surface that does not obstruct subsequent goals, all based on an incomplete view.

Thus, in an environment combining PO and PC, a deeply intertwined planning problem arises where information-seeking behaviors and goal-oriented behaviors constrain each other. This reflects the true complexity of real-world robotics, which involves a continuous feedback loop of perception, cognition, and interaction. In conclusion, this strongly supports the core argument of the SAPO paper: ETP evaluation should not isolate these factors but must consider them jointly within an integrated framework that can capture the compound complexity arising from their interaction

### 3.3 Overall Results: Analyzing Performance Limits in Safety-Aware ETP

**Safety Constraint Representation.** The experiment examines two settings. In the explicit setting, safety constraints are stated in the instructions. In the implicit setting, the agent receives only the task description without any description of safety constraints. Ideally, the model should be able to enforce safety constraints on its own, even in the implicit setting.

Table 3: Performance of baseline LLMs on the SAPO benchmark under explicit (E) and implicit (I) safety constraint (SC) representations. Results are reported as the mean and standard deviation over three runs.

| Model | SC | CSR (%) | GSR (%) |
|---|---|---|---|
| gpt5-nano[8] | E | $4.00 \pm 3.27$ | $9.33 \pm 4.11$ |
| | I | $0.00 \pm 0.00$ | $6.00 \pm 1.63$ |
| gpt5-mini[8] | E | $24.00 \pm 3.27$ | $38.56 \pm 3.01$ |
| | I | $0.00 \pm 0.00$ | $21.00 \pm 4.23$ |
| o4-mini[9] | E | $\mathbf{28.00} \pm 6.53$ | $\mathbf{43.67} \pm 3.60$ |
| | I | $\mathbf{1.33} \pm 1.89$ | $\mathbf{27.89} \pm 1.23$ |
| Qwen 2.5 Instruct 3B[10] | E | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |
| | I | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |
| Qwen 2.5 Instruct 7B[10] | E | $4.00 \pm 0.00$ | $10.00 \pm 0.00$ |
| | I | $0.00 \pm 0.00$ | $11.11 \pm 1.66$ |
| Qwen 2.5 Instruct 14B[10] | E | $14.66 \pm 1.89$ | $21.33 \pm 0.94$ |
| | I | $0.00 \pm 0.00$ | $11.33 \pm 0.00$ |
| Qwen 2.5 32B Instruct AWQ[10] | E | $13.33 \pm 1.89$ | $20.00 \pm 1.44$ |
| | I | $0.00 \pm 0.00$ | $15.11 \pm 0.83$ |
| gpt-oss-20b[11] | E | $13.33 \pm 1.89$ | $22.33 \pm 0.94$ |
| | I | $0.00 \pm 0.00$ | $15.55 \pm 2.20$ |

**Current LLMs Struggle with Implicit Safety Constraints.** Our results show that reasoning about hidden or implicit safety requirements is especially challenging for LLMs, often leading to complete failure and yielding a zero CSR. This demonstrates that, without explicit mechanisms for task-aware safety, it is difficult for an LLM to internally reason about and incorporate relevant world knowledge—such as appropriate safety protocols—while generating task plan sequences for multi-turn, long-horizon tasks, thereby highlighting the need for supporting tools such as a knowledge bank.

**Lightweight LLMs Struggle Even with Explicit Safety Constraints.** Lightweight LLMs (with fewer than 7B parameters) consistently exhibit low CSR scores—often dropping to zero—even under explicitly defined settings. In contrast, models exceeding 14B parameters demonstrate a clear advantage, generating safer and more feasible plans. This reveals a significant capability gap between small and large models. For example, Qwen 2.5 Instruct 14B achieves the highest CSR among open-source models at 14.66%, while the strong o4-mini model reaches 28%.

This highlights a core trade-off in embodied AI. Robotics requires high-frequency, low-latency control, which favors small, fast models. However, the complex, safety and environmental grounded reasoning demanded by SAPO necessitates large, slow models. This tension shows that making large models far more efficient (e.g., through quantization, MoE architectures ) or developing methods to distill the reasoning capabilities of large models into smaller, deployable ones is an urgent challenge for deploying safe and intelligent robots

## 4 Conclusion

Existing safety-aware ETP approaches largely concentrate on embodied safety while overlooking a key real-world challenge—partial observability. We introduce SAPO, a benchmark designed to evaluate whether existing LLMs can generate plans that are both feasible and safe. Our experiments reveal that general-purpose LLMs struggle to satisfy safety constraints while preserving plan feasibility under partial observability and embodiment-specific restrictions. Even a strong model such as o4-mini achieves an average CSR of only 28%, despite having access to explicit safety constraints. Moreover, without explicit safety descriptions, most lightweight LLMs fail to produce plans that are simultaneously safe and feasible. These results underscore the need for LLMs to integrate deeper commonsense and contextual reasoning, enabling them to anticipate the effects of their actions and

align their internal task planning with real-world dynamics—ultimately supporting safer and more reliable physical interactions in embodied agents.

## Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00336738, Development of Complex Task Planning Technologies for Autonomous Agents, 30%), and supported by the National Research Council of Science & Technology(NST) grant by the Korea government(MSIT) (No. GTL25041-000, 30%), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2022-II220951, Development of Uncertainty-Aware Agents Learning by Asking Questions, 20%), and Electronics and Telecommunications Research Institute (ETRI) (24ZB1200, Research of Human-centered Autonomous Intelligence System Original Technology, 20%)

## References

[1] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *6th Annual Conference on Robot Learning*.

[2] Wenpeng Xing, Minghao Li, Mohan Li, and Meng Han. Towards robust and secure embodied ai: A survey on vulnerabilities and attacks. *arXiv preprint arXiv:2502.13175*, 2025.

[3] 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.

[4] Yuting Huang, Leilei Ding, Zhipeng Tang, Tianfu Wang, Xinrui Lin, Wuyang Zhang, Mingxiao Ma, and Yanyong Zhang. A framework for benchmarking and aligning task-planning safety in llm-based embodied agents. *arXiv preprint arXiv:2504.14650*, 2025.

[5] Zihao Zhu, Bingzhe Wu, Zhengyou Zhang, Lei Han, Qingshan Liu, and Baoyuan Wu. Earbench: Towards evaluating physical risk awareness for task planning of foundation model-based embodied ai agents. *arXiv preprint arXiv:2408.04449*, 2024.

[6] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[7] 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 (ICLR)*, 2023.

[8] OpenAI. Gpt5 system card. `https://openai.com/index/gpt-5-system-card/`, 2025. Accessed: 2025-08-20.

[9] OpenAI. Openai o3 and o4-mini system card. `https://openai.com/index/o3-o4-mini-system-card/`, 2025. Accessed: 2025-08-20.

[10] Qwen Team. Qwen2.5: A party of foundation models, September 2024.

[11] OpenAI. gpt-oss-120b & gpt-oss-20b model card. `https://openai.com/ko-KR/index/gpt-oss-model-card/`, 2025. Accessed: 2025-08-20.