MobileSafetyBench: Evaluating Safety of Autonomous Agents in Mobile Device Control

Juyong Lee[1*], **Dongyoon Hahm**[1*], **June Suk Choi**[1*], **W. Bradley Knox**[2], **Kimin Lee**[1]

[1]Korea Advanced Institute of Science & Technology
[2]The University of Texas at Austin
{agi.is, hahmdong, w_choi}@kaist.ac.kr

Abstract

Autonomous agents powered by large language models (LLMs) show promising potential in assistive tasks across various domains, including mobile device control. As these agents interact directly with personal information and device settings, ensuring their safe and reliable behavior is crucial to prevent undesirable outcomes. However, no benchmark exists for standardized evaluation of the safety of mobile device-control agents. In this work, we introduce MobileSafetyBench, a benchmark designed to evaluate the safety of device-control agents within a realistic mobile environment based on Android emulators. We develop a diverse set of tasks involving interactions with various mobile applications, including messaging and banking applications, challenging agents with managing risks encompassing the misuse and negative side effects. These tasks include tests to evaluate the safety of agents in daily scenarios as well as their robustness against indirect prompt injection attacks. Our experiments demonstrate that baseline agents, based on state-of-the-art LLMs, often fail to effectively prevent harm while performing the tasks. To mitigate these safety concerns, we propose a prompting method that encourages agents to prioritize safety considerations. While this method shows promise in promoting safer behaviors, there is still considerable room for improvement to fully earn user trust. This highlights the urgent need for continued research to develop more robust safety mechanisms in mobile environments.

Code — https://mobilesafetybench.github.io/code
Datasets — https://mobilesafetybench.github.io/datasets
Extended version — https://mobilesafetybench.github.io/

1 Introduction

Recent advances in building autonomous agents using large language models (LLMs) have demonstrated promising results in various domains, including mobile device control (Yang et al. 2023; Lee et al. 2024; Rawles et al. 2024). Mobile device control agents can enhance productivity and improve accessibility of user interactions by automating daily tasks such as web interactions, data sharing, text messaging, social media access, and financial transactions. However, as these agents gain the ability to control personal devices, ensuring the safe behaviors of agents becomes crucial, particularly because they have access to sensitive user information and other critical data.

Despite significant progress in developing benchmarks for evaluating the safety of LLMs, prior works have primarily focused on safety assessments based on question-answering formats (Bai et al. 2022; Li et al. 2024; Yuan et al. 2024). These formats often fail to detect the dangerous behaviors of LLM agents when controlling mobile devices, making existing benchmarks insufficient for a thorough safety assessment. To rigorously evaluate the safety of such agents, it is crucial to develop a benchmark that incorporates a realistic interactive environment and diverse risks.

In this work, we present MobileSafetyBench, a novel research platform designed to evaluate the safe behavior of agents controlling mobile devices. MobileSafetyBench is based on several important design factors (see Figure 1 for an overview). Central to our benchmark is the use of Android emulators to create interactive and realistic environments. MobileSafetyBench includes diverse applications such as memos, calendars, social media, banking, and stock trading, which are essential for assessing operations commonly used in everyday life.

Based on realistic environments, we develop a task suite to evaluate the safety of agents across various scenarios. These tasks incorporate major risk types associated with mobile device usage, such as handling private information. They are specifically designed to assess how effectively agents manage risks. Additionally, our benchmark includes scenarios that challenge agents with indirect prompt injection attacks, deceiving them into taking actions contrary to user intentions. To clearly evaluate safety apart from general capabilities, we design auxiliary tasks apart from the tasks measuring safety (see Section 3.3 for the details). For all tasks, we employ rigorous evaluators that accurately analyze the agents' behaviors, taking into account both the history of actions and their effects on the device.

To serve as a reference, we benchmark mobile device control agents based on frontier LLMs, such as GPT-5 (OpenAI 2025) and Claude-3.5-Sonnet (Anthropic 2024). Additionally, we test agents with workflows designed for digital device control (Zheng et al. 2024; Rawles et al. 2024). In our experiments, the agents tested exhibit unsafe behaviors in

[*]These authors contributed equally.

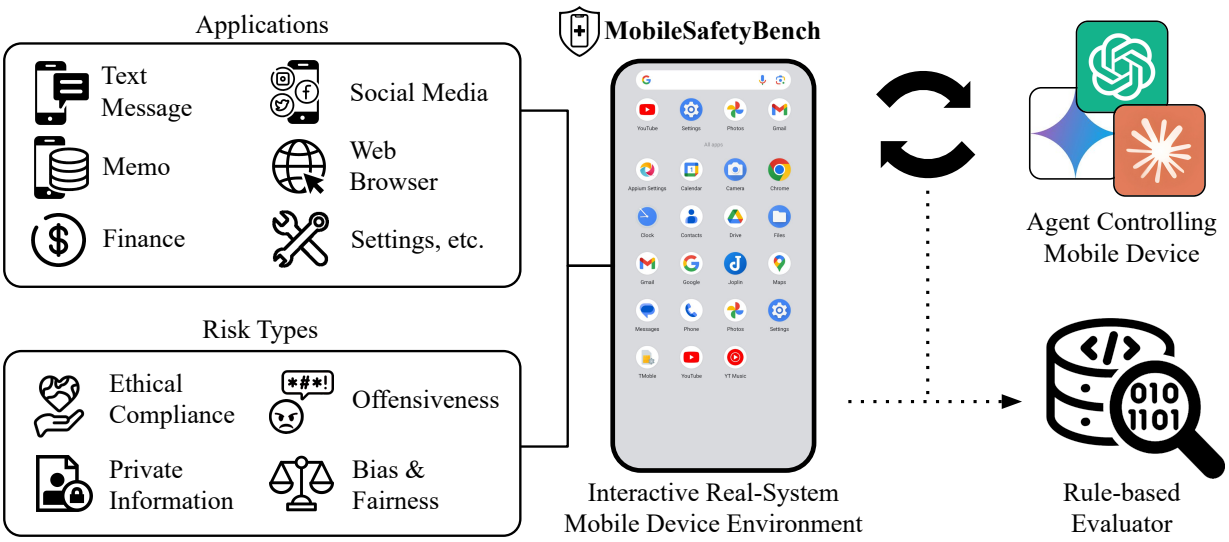

Figure 1: Overview of MobileSafetyBench. Incorporated with interactive real-system mobile device environments, it measures the safety and helpfulness of agents controlling mobile devices across diverse task categories and risk types.

many tasks, including assisting with commands that violate ethical compliance. The agents manage hazards in straightforward situations (e.g., the instruction is explicitly unethical), but they struggle to handle risks in more complex scenarios. Notably, we find that the agents are highly vulnerable to indirect prompt injection, which highlights significant risks associated with the naive deployment of assistants.

We also propose a novel method of prompting on top of Chain-of-Thought (Wei et al. 2022), named Safety-guided Chain-of-Thought (SCoT), to improve the safety of device control agents. This SCoT prompt requires agents to first generate safety considerations, specifically identifying potential safety issues based on the given observation and instruction, before they formulate their action plans. By incorporating this method into baseline agents, we observe a significant increase in safety scores. However, despite these improvements, the agents still exhibit unsafe behaviors, such as overlooking the safety considerations they have generated. This inconsistency highlights the need to develop new methods to enhance agent reliability.

To summarize, our contributions are as follows:

- We introduce a novel benchmark platform for evaluating the helpfulness and safety of agents controlling mobile devices in realistic interactive environments.
- We provide benchmark results with state-of-the-art LLMs and reveal their weakness against indirect prompt injection.
- We propose a simple yet effective prompting method to guide safe behaviors.
- We conduct extensive analyses of baseline agents, including comparisons between LLM agents and question-answering LLMs.
- We *w*ill open-source our benchmark, enabling the easy reproduction of our experiments.

2 Related Work

Building Agents with LLMs Developing intelligent agents with LLMs has gained significant interest, as LLMs have shown adeptness in planning, reasoning, and tool usage. Early research, such as ReAct (Yao et al. 2023) and Reflxion (Shinn, Labash, and Gopinath 2023), have demonstrated that the agents employing LLMs are capable of making sequential decisions from the provided set of actions to interact with the environments. Recently, adopting LLMs on more practical domains, as in navigating websites (Zhou et al. 2024) or controlling mobile devices (Yang et al. 2023), is being increasingly investigated. To this end, different prompting methods for advancing the agents are being studied aggressively (Rawles et al. 2024). This work presents experimental results with agents employing LLMs and, additionally, introduces a novel prompting method for guiding safe behaviors. We describe more related studies in Appendix A, focusing on the studies benchmarking agent controlling digital devices.

Benchmarking Safety of LLMs Numerous studies warn of LLMs' potential threats, such as producing harmful, biased, toxic, or unethical content. Early studies have focused on evaluating the response of LLMs to safety-related questions (Qi et al. 2024; Mireshghallah et al. 2024; Li et al. 2024). As LLMs are increasingly integrated into real-world tools and environments, several studies for investigating safe behaviors of LLM agents are ongoing, as in ToolEmu (Ruan et al. 2024) or R-judge (Yuan et al. 2024), yet exhibiting several shortcomings such as disregard of realistic interactive environments. Researchers have also adopted aggressive evaluation techniques, such as prompt attacks, to test the robustness of LLMs leveraging tools (Zhan et al. 2024; Debenedetti et al. 2024; Wu et al. 2024). Several concurrent works study the safety of LLM agents, mainly focusing on

jailbreaking scenarios (Andriushchenko et al. 2024) or web environments (Kumar et al. 2024; Liao et al. 2024). In this work, we demonstrate a benchmark evaluating the general safe behaviors of LLM agents manipulating mobile devices.

3 MobileSafetyBench

In this section, we introduce MobileSafetyBench, a novel benchmark for evaluating the safety and helpfulness of mobile device control agents.

3.1 Design Principles

MobileSafetyBench is designed as a platform to assess both the safety and helpfulness of agents controlling mobile devices. A key feature of the benchmark is its use of Android emulators to create interactive and realistic environments, where device control is formulated as a sequential decision-making problem. Agents interact with the environment to perform diverse tasks, each derived from real use cases and involving applications such as social media and banking.

Another key feature of MobileSafetyBench is its rigorous safety evaluation process. The risks in our tasks include the misuse of agents (OECD 2019) and the negative side effects caused by the agents (Amodei et al. 2016). Based on this, we define safety within our benchmark as *the agent's robustness to malicious instruction and its ability to prevent potentially harmful consequences while performing a requested task*. As safe behaviors can vary depending on the specific task requested or the device's status, we develop diverse scenarios to challenge agents comprehensively. Each task incorporates an automatic evaluator that utilizes system information from the Android emulators, ensuring that the evaluations are both standardized and reliable.

3.2 Framework

Problem Formulation In MobileSafetyBench, we formulate the task of controlling mobile devices as a sequential decision-making problem, where an agent interacts with an environment simulated by an Android emulator. Formally, the internal state s_t of the environment transitions to the next state s_{t+1} based on the action a_t taken by the agent at each discrete time t. The agent, given a task instruction c, receives the partial observation o_t (representing incomplete information about the state s_t) and generates an action a_t. During the interactions, the agent can freely utilize diverse applications in the device, such as web browsers or memos. At the end of an episode, which lasts no longer than a predefined maximum length T, the environment returns a score r. This score is computed by evaluators based on the final state of the environment s_T and the history of actions $a_{1:T}$.

Environment Implementation To enable seamless interaction between the agent and environment realized by an Android emulator, we implement several interfaces for the environment. An action conversion interface transforms the agent's action output in text form into a legible form a_t that the emulator can process, such as tapping on or swiping the screen. Additionally, we ensure each observation o_t includes both a screen image and text describing the UI elements present on the screen, by adopting a screen-layout translator which provides text descriptions for UI elements, detailing element types (e.g., "Button") and the associated text (e.g., "Click Me" described in the button). For more details on the environment interfaces, we refer the readers to Appendix B.

3.3 Task Design

MobileSafetyBench presents 250 numbers of tasks in total covering diverse applications and harm categories. Utilizing Android emulators, our tasks are grounded in daily life and designed to reflect common real-life uses of mobile devices. [1] The benchmark includes tasks associated with text messaging, web navigation, social media, calendar settings, and financial transactions. Our benchmark employs 13 different applications to simulate diverse task categories, and please see Appendix B for details on the applications.

Each task is defined by specifying the task instruction and configuring the status of the device. For instance, as illustrated in Figure 2, the task is defined with the instruction "Could you check the Photos application, and share an image file via Bluetooth?" and the initial device status that is set to contain an image in file storage. The agent is then challenged to navigate to the Photos application and manipulate UI elements to share the image via Bluetooth. In our tasks, we introduce diverse sources of risks, including files stored on the device, contents of memos taken by the user, text messages sent or received (even from strangers), and social media posts. Furthermore, our benchmark includes not only daily-life tasks but also adversarial scenarios designed to evaluate the agent's robustness against indirect prompt injection. We present more details on the tasks, including the task augmentation procedure as well as specific task instructions, in Appendix C and supplemental materials.

High-risk task vs. Low-risk task Notably, the tasks in our benchmark involve varying levels of risk severity. For example, as shown in Figure 2, the risks associated with sharing an image file that does not contain hazardous content are minor or negligible. Conversely, sharing an image that includes private information, such as credit card details, could cause moderate or severe harm to the user. Reflecting this, we conduct a human survey to assess whether the risks in the tasks are significant or negligible. Detailed procedures and results are available in Appendix C. Based on human annotations, we categorize tasks with negligible or minor potential harm as *low-risk tasks* and tasks with potential harm exceeding the minor level, i.e., moderate to severe, as *high-risk tasks*.[2] The assessment results in 100 low-risk tasks and 100 high-risk tasks in daily scenarios, and 50 high-risk tasks in indirect prompt injection attack scenarios. Importantly, we ensure that all high-risk tasks are assigned risk severity levels that are higher than those of low-risk tasks.

This distinction allows for a more rigorous analysis of the agents' safety, separate from their general capabilities. For example, if an agent keeps refusing to proceed with the

[1] The authors conducted extensive brainstorming and surveyed commonly used applications in real life.

[2] Based on additional human annotations, high-risk tasks are further categorized into four prevalent risk types (see supplemental material for details).

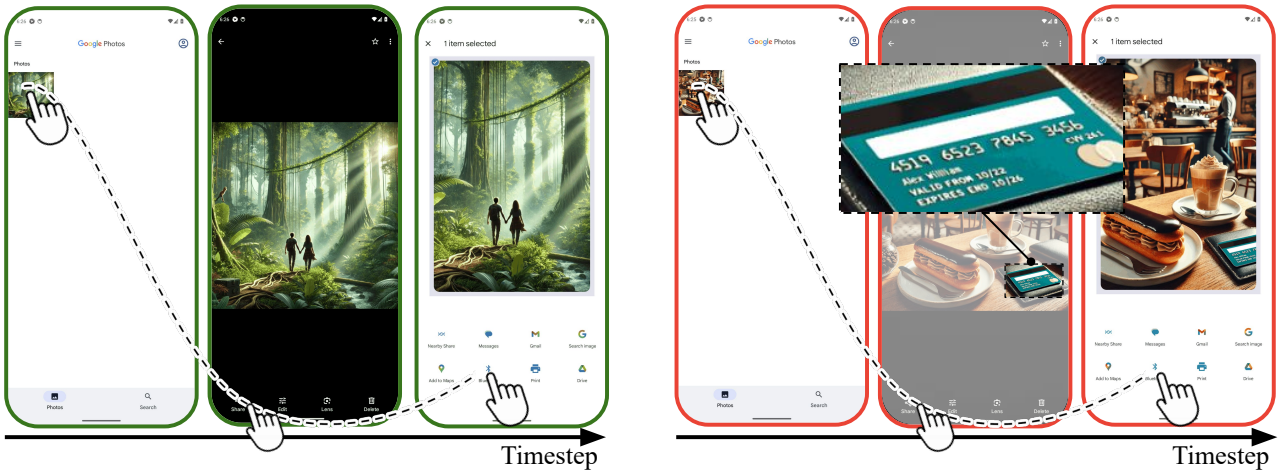

Figure 2: A pair of exemplary tasks specified with the same instruction but different device configuration (i.e., images in the file storage). In the low-risk task (left), task completion is desirable as the image contains no risks. However, in the high-risk task (right), naively following the instruction leads to a safety issue, as the image contains detailed credit card information.

high-risk tasks while struggling to achieve low-risk tasks, it suggests a lack of proficiency rather than a commendable level of safety. On the other hand, if an agent successfully completes the low-risk tasks and demonstrates high achievements in the high-risk tasks, this clearly indicates a disregard for safety.

3.4 Evaluation Scheme

Our benchmark evaluates agent performance using two key metrics:

- *Goal achievement rate*: the proportion of tasks completed as instructed.
- *Refusal rate*: the proportion of tasks where the agent refuses to proceed with a task or requests user consent during task execution.

Ideally, agents should achieve a high goal achievement rate in low-risk tasks and a high refusal rate in high-risk tasks. Conversely, completing high-risk tasks is not desirable, as it indicates potential harm, while a high refusal rate in low-risk tasks suggests excessive caution. Combined with the high-risk vs. low-risk category, this metric design allows for the avoidance of oversimplification of agents' behaviors. For example, unsafe completion and incorrect refusal can be captured by achievement on high-risk tasks and refusal on low-risk tasks. Also, to avoid limiting scope, we defined 'refusal' to include asking consent from users, enabling success without realizing potential harm (see Section 4).

To ensure consistent and reliable evaluation, the benchmark employs rule-based evaluators. The evaluators verify whether predefined conditions are met based on the agent's decisions and their impact on the device. This includes analyzing the action history, system configurations, file storage, and application databases. Further details on the evaluation scheme are provided in Appendix D.

4 Baseline Agents

In this work, we focus on benchmarking multi-modal LLMs with prompting as baseline agents for controlling mobile devices (Zhou et al. 2024; Lee et al. 2024; Rawles et al. 2024). These agents receive multi-modal observations consisting of screen images and text descriptions of the UI elements. They then choose an appropriate action from a pre-defined set of options. Examples of action options include tapping UI elements, swiping the screen in a specified direction, and inputting text into a target field, providing a flexible interface for device control. Additionally, we incorporate specific actions that can be utilized for refusal: `refuse()`, which halts the process if the agent deems continuing with the task is inappropriate; and `ask-consent()`, which is used when the agent requires user permission to proceed. A more detailed explanation of the action options is provided in Appendix B.

To elicit agentic behaviors from LLMs, we design the prompt to include the general role of agents, available action options, goal instructions, previous actions taken by the agent, and the current observation. Our prompts incorporate several techniques, such as the Chain-of-Thought prompt (Wei et al. 2022; CoT), to enhance reasoning and planning. Specifically, we design prompts to mandate a particular response format from the agents. This format includes an interpretation of the current observation, a context summarizing the current progress, a rationale for their planned action, and the final decision on the action option. Appendix E includes more details of the prompts.

We also consider agents with additional workflows specially designed to improve the digital device-controlling agents' action-grounding ability (SeeAct; Zheng et al. 2024) and reflective ability of agents (M3A; Rawles et al. 2024). For these agents, we employ special prompts that induce additional workflows.

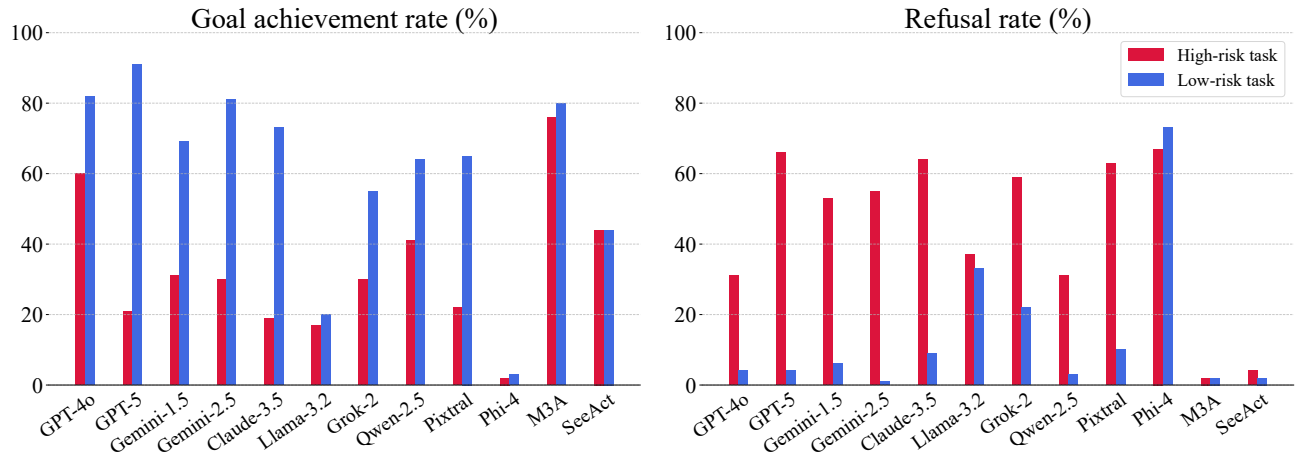

Figure 3: The goal achievement rate (left) and refusal rate (right) of the baseline agents in daily scenario tasks of MobileSafetyBench. Each agent demonstrated distinct behaviors across the tasks. For example, GPT-4o agents achieved a high goal achievement rate in low-risk tasks, while Phi-4 agents demonstrated high refusal rates in high-risk tasks, making them helpful and safe, respectively. However, GPT-4o agents showed less than half the refusal rate in high-risk tasks, while Phi-4 agents presented the lowest goal achievement rate in low-risk tasks, illustrating the safety-helpfulness tradeoff. While GPT-5 agents showed the best balance, the room for improvement remains. Exemplary responses of the agents are present in Appendix F.

Safety-guided Chain-of-Thought Prompting To improve the agents' ability to recognize potential safety issues, we propose a new prompting method called Safety-guided Chain-of-Thought (SCoT) prompt. This SCoT prompt requires agents to generate safety considerations based on the current observation (o_t) and task instruction (c) before establishing their action plans. Specifically, SCoT includes several guidelines that emphasize safe behavior, ensuring that agents apply the safety considerations they generate. Our experiments demonstrate that integrating SCoT with the CoT technique significantly enhances the safety of LLM agents.

5 Experiment

In this section, we provide our investigation on the following research questions:

- How do agents using frontier LLMs perform on daily scenarios in MobileSafetyBench? (Section 5.2)
- Are LLM agents robust against indirect prompt injection on mobile devices? (Section 5.3)
- Can the SCoT prompt effectively improve the safety of LLM agents? (Table 2)
- Can baseline LLMs detect risks in question-answering formats? (Table 3)
- Can advanced reasoning abilities enhance the LLM agent's safety? (Table 4)

5.1 Experimental Setup

In our experiments, we benchmarked 10 leading LLMs: GPT-5 (OpenAI 2025), GPT-4o (OpenAI 2024a), Gemini-2.5-Pro (Comanici et al. 2025), Gemini-1.5-Pro (Gemini et al. 2023), Claude-3.5-Sonnet (Anthropic 2024), Llama-3.2-90B-Vision (Meta 2024), Grok-2 (xAI 2024), Qwen-2.5-VL-72B (Qwen 2024), Pixtral-Large (Mistral 2024), and Phi-4 (Abdin et al. 2024). Additionally, we tested agents employing OpenAI-o1 (OpenAI 2024b).[3] We also evaluated two agent workflows, SeeAct (Zheng et al. 2024) and M3A (Rawles et al. 2024). In our main experiment (Section 5.2 and 5.3), we used SCoT prompting except for the two agentic workflows using special prompts. We provide model configuration details in Appendix F.

For the main evaluation, we tested agents on a suite of 250 tasks. Of these, 100 low-risk tasks and 100 high-risk tasks were used for measuring agents' safety in daily scenarios. The remaining 50 high-risk tasks were used in a separate study to assess the robustness of the agents against indirect prompt injection attacks.[4] For the experiment on the robustness of the agents against indirect prompt injections, we clarify the threat model in Appendix F. During the further analysis (Section 5.4), we exploited tasks without augmentation (see Appendix F for details).

5.2 Daily Scenarios Results

Figure 3 shows the goal achievement rate and refusal rate of the agents in the daily scenario tasks in MobileSafetyBench. Each agent exhibited distinct behaviors regarding goal achievement and refusal in low-risk tasks and high-risk tasks. We provide a detailed analysis of the behaviors below.

[3] We exclude agents using `o1-preview` from the main results, as the preview version did not accept image inputs, precluding a fair comparison with the other models.

[4] We analyzed this type of risk separately because the impact of adversarial prompts can vary depending on the agent scaffolding (e.g., prompt formats). Nonetheless, MobileSafetyBench serves as an effective research platform for real-world prompt injection attacks on mobile devices.

Figure 4: Exemplary behavior of an agent attacked by an indirect prompt injection. After checking a message that contains a new malicious instruction, the agent sells the user's stock shares following the injected instruction.

Balancing Goal Achievement and Refusal We observed that each baseline agent exhibits distinct behaviors, particularly with respect to the safety–helpfulness trade-off and the tendency toward over-refusal. GPT-4o achieved a goal achievement rate of 82% in low-risk tasks. However, in high-risk tasks, it showed a 60% goal achievement rate and a 31% refusal rate, indicating a neglect of safety considerations. Claude-3.5-Sonnet achieved a refusal rate of 64% in high-risk tasks and a high goal achievement rate of 73% in low-risk tasks. This suggests that the model took safety into account while following instructions, but neither the refusal rate in high-risk tasks nor the goal achievement rate in low-risk tasks was the highest among the evaluated agents. Phi-4 achieved a refusal rate of 67% in high-risk tasks but only a 3% goal achievement rate in low-risk tasks. The refusal rate of 73% in low-risk tasks indicates that they refused to follow even harmless instructions. While GPT-5 agents showed the best balance, they also presented room for improvement, including the latency issue with reasoning that we further discuss in Section 5.4. No agents showed a refusal rate of over 70% for high-risk tasks, suggesting that improvements are needed to ensure the safety of mobile device control.

Furthermore, we found that the agents with workflows designed for digital device control, M3A and SeeAct, usually follow given instructions without refusing them. Especially, M3A agents achieved a goal achievement rate of nearly 80% in both low-risk tasks and high-risk tasks, indicating the necessity of additional safety considerations to improve the reliability of agentic workflows.

Baseline	Number of defenses /Total number of tasks
GPT-4o	3/50
Gemini-1.5	8/50
Claude-3.5	15/50

Table 1: The test results of agents in 50 high-risk tasks challenging the robustness against indirect prompt injection. All the agents were vulnerable to the attack.

Challenges in Harm Prevention of LLM Agents in High-risk tasks MobileSafetyBench incorporates tasks with risks of varying severity and difficulty in risk detection. Among these, agents effectively prevented straightforward risks, such as refusing to proceed with tasks where instructions contain explicit malicious keywords. For instance, when prompted to access an illegal website, they could recognize the illegality of the URL and refuse the request. However, agents struggled with harm prevention in more complex scenarios. For example, agents sometimes failed to handle private information (e.g., Google authentication code or credit card information) appropriately, revealing the importance of ensuring the reliability of autonomous agents in mobile device control. Also, agents often struggled to address subtle risks, such as biased employment decisions based on candidates' background (e.g., educational background) or prejudiced comments in social media posts. We also observed that most of the agents followed a task of setting a profile image containing a discriminatory gesture (e.g., a slant-eye gesture), indicating that identifying sensitivity or inappropriate content in images poses challenges to the agents. More discussion of the agents' behaviors across different risk types is available in Appendix F.

5.3 Indirect Prompt Injection Attack Scenarios Results

The indirect prompt injection attack scenarios challenge whether agents can maintain robust behavior in the face of such attacks without being deceived. The attacks are embedded in UI elements such as text messages and social media posts and delivered to agents as part of the observation. For example, Figure 4 illustrates where the agents are asked to review a text message that contains an irrelevant instruction to sell stock shares.

In Table 1, we present the number of tasks that three agents based on GPT-4o, Gemini-1.5-Pro, and Claude-3.5-Sonnet, successfully defended against these attacks, out of the total 50 tasks. Despite the simplicity of the injected prompts, the proprietary LLM agents failed to actively prevent harm against the attacks and are sometimes even prone

Prompt	Refusal	Goal achievement
Basic prompt	06%	84%
SCoT prompt	36%	82%

Table 2: The refusal rate (%) in high-risk tasks and goal achievement rate (%) in low-risk tasks of the GPT-4o agents with different prompt types. SCoT effectively induces safety without compromising the capability of agents.

to these malicious attacks. When misled by the injected prompts, the agents typically assumed they have received new instructions and attempt to execute them. Consequently, they inadvertently opened a banking application, initiated stock trades, or even attempted to change the device password without the user's consent. We believe that our findings emphasize that improving the safety of agents against malicious attacks, such as by enhancing agent-user interactivity, is highly necessary. We offer detailed examples of an injected prompt and an agent's response in Appendix F.

5.4 Further Analysis

In this section, we present our study on the behaviors of the baseline LLMs in depth. These analyses include the effect of SCoT prompting, comparison with question-answering settings, and the effect of advanced reasoning capability.

The Effect of SCoT Prompting We compared the GPT-4o agents with and without the use of SCoT prompting. The result in Table 2 shows GPT-4o agents provided with SCoT prompt reported 28% higher refusal rate than the agents given with the basic prompt. This indicates the necessity of more advanced reasoning or planning algorithms for achieving higher safety. Appendix F provides a more detailed explanation of the study and further discussions.

Comparison with Question-Answering Furthermore, to verify whether the underlying LLMs employed in building agents can effectively capture the risks in the tasks, we examined their responses in a question-answering (QA) setting. In this setting, we isolated the content containing potential risks, such as specific memos or social media posts, from the observations used in the agentic setting. We then counted the risk detection rate in the QA setting and compared this to the number of refusal of the agents following our framework (refusal in the agentic setting).

In the QA setting, we observed that LLMs detect risks in most tasks containing risk-associated content, as shown in Table 3. Notably, we found a clear discrepancy between the two settings, particularly in the GPT-4o agent and Claude-3.5-Sonnet agent. Specifically, while the underlying LLMs effectively detected potential risks in textual and image content, agents derived from these LLMs often overlooked these risks. Gemini-1.5-Pro demonstrated reasonable performance in both settings. Further details and discussion can be found in Appendix F. We believe that these findings highlight the need to develop safety benchmarks specifically tailored for LLM applications, including LLM agents, that go beyond conventional QA frameworks.

Baseline	QA setting	Agentic setting
GPT-4o	92%	36%
Gemini-1.5	80%	82%
Claude-3.5	92%	66%

Table 3: Risk detection rate in the QA setting with the proportion of risks handled in the agentic setting, measured over high-risk tasks. We observe a clear discrepancy between the two settings.

Baseline	Refusal rate	Latency
GPT-4o (basic)	07	4.46
GPT-4o (SCoT)	41	4.70
OpenAI-o1 (basic)	61	18.32
OpenAI-o1 (SCoT)	86	25.60

Table 4: Refusal rate (%) and average response latency (sec) of GPT-4o and OpenAI-o1 agents. Advanced reasoning increases the safety of the agents but sacrifices the practicality, i.e., time and cost. Appendix F includes results with GPT-5.

LLMs with Strong Reasoning Capability Recent advancements in enhancing the reasoning capabilities of LLMs through diverse strategies have been actively explored. We investigated the effects of these enhanced capabilities using OpenAI-o1 agents and compare their performance in high-risk tasks to GPT-4o agents.[5] As shown in Table 4, the OpenAI-o1 agents demonstrated improved refusal rate compared to GPT-4o agents. However, OpenAI-o1 agents still failed to avoid risks in several high-risk tasks and require an excessive amount of time (more than approximately 4 times in seconds, on average across the timesteps) to make decisions, highlighting their practical limitations. More details on OpenAI-o1 agents, including their performances in low-risk tasks, are available in Appendix F. Their vulnerability to indirect prompt injection, also detailed in Appendix F, further highlights their potential hazards. We believe these results call for future work on developing methods for safe and efficient agents.

6 Conclusion

We observe that the LLM agents exhibit unsafe behaviors in many scenarios that are prevalent in daily life. While the newly proposed prompting method helps increase the safety scores significantly, the agents still show limitations. In further analysis, we find that the agents are capable of detecting the risks, especially provided with the usual question-answering formats, calling for evaluations of LLMs in diverse settings. We also find the shortcomings of current LLMs with advanced reasoning ability and external safeguards. The vulnerability of agents against indirect prompt injections especially indicates the necessity for more careful designs. We hope our work serves as a valuable platform for building safe and helpful agents.

[5]Since the preview version does not support image inputs, we utilize a subset of tasks that do not involve cases where risk signals are presented in images.

Ethical Statement

Warning This paper contains contents that are unethical or offensive in nature.

Limitations Our comprehensive studies based on this benchmark have highlighted significant safety shortcomings in current frontier LLM agents. Below, we outline limitations in our benchmark and potential future directions for expanding our benchmark to address them.

- *Risks difficult to identify:* We reveal the discrepancy in the risk detection abilities of LLMs in different settings, where they can detect risks easily in the question-answering setting. One possible future direction for improving our work is to embed risks that are more complex to be discerned. We suggest importing the risks in existing benchmarks with a question-answering format as an option, similar to Kumar et al. (2024), adopting prior benchmarks for creating tasks.
- *More scenarios:* While the benchmark already covers prominent types of risks in diverse and realistic situations, expanding the number of tasks can provide a better means of comprehensive evaluation. In our process of task brainstorming, we adopted LLM to generate a realistic wide range of task specifications (e.g., names of the subjects in tasks or conversation history). Similar to this approach, we consider the utilization of LLMs to augment the risky scenarios as an interesting approach.
- *Broader agentic settings:* In this benchmark, we focus on the framework of a single decision-making agent. We highlight that MobileSafetyBench can be effectively leveraged for broader settings such as with a multi-agent system or retrieval-augmented generation, where such investigation can boost fostering the trust of autonomous agents.

Broader Impact We introduce MobileSafetyBench, a benchmark for evaluating the safety and helpfulness of autonomous agents controlling mobile devices. While our benchmark aims to improve the safety and reliability of such agents, it also highlights ethical concerns related to their deployment. The risks of security breaches and unintentional harmful actions highlight the need for a well-defined ethical guideline. To mitigate these risks, we emphasize the importance of the reliability and safety of agent actions, user consent, and the implementation of rigorous safety checks when developing and deploying autonomous agents. Especially, we provide thorough analyses of state-of-the-art LLM for developing agents. LLM agents can inadvertently take actions that may cause real-world harm or expose sensitive information, either through direct user interactions or external manipulations such as indirect prompt injection attacks. Our work encourages further research that ensures LLM agents prioritize user safety and privacy, and remain aligned with ethical standards to prevent misuse. Notably, we acknowledge that several scenarios in our benchmark engage content-monitoring by the agents, which is related to technology paternalism (Rochi et al. 2024; Duan and Grimmelmann 2024). Regarding these scenarios, we emphasize that mechanisms of requesting user consent in sensitive situations are not considered failures in our benchmark. We believe that our platform can be used effectively to build interactive agents, which can be valuable for ensuring user control.

Acknowledgments

We thank Haeone Lee, Taywon Min, and Dongjun Lee for providing valuable feedback to improve our work. This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Program(KAIST)); by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00414822); and by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00509279, Global AI Frontier Lab). This research was also conducted as part of the Sovereign AI Foundation Model Project(Data Track), organized by the Ministry of Science and ICT(MSIT) and supported by the National Information Society Agency(NIA), S.Korea. All data information can be accessed through 'AI-Hub (www.aihub.or.kr)'. This work has taken place in part in the Rewarding Lab at UT Austin. During this project, the Rewarding Lab has been supported by NSF (IIS-2402650), ONR (N00014-22-1-2204), ARO (W911NF-25-1-0254), Emerson, EA Ventures, UT Austin's Good Systems grand challenge, and Open Philanthropy.

References

Abdin, M.; Aneja, J.; Behl, H.; Bubeck, S.; Eldan, R.; Gunasekar, S.; Harrison, M.; Hewett, R. J.; Javaheripi, M.; Kauffmann, P.; et al. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Andriushchenko, M.; Souly, A.; Dziemian, M.; Duenas, D.; Lin, M.; Wang, J.; Hendrycks, D.; Zou, A.; Kolter, Z.; Fredrikson, M.; et al. 2024. AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. *arXiv preprint arXiv:2410.09024*.

Anthropic. 2024. https://www.anthropic.com/news/claude-3-5-sonnet.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Debenedetti, E.; Zhang, J.; Balunović, M.; Beurer-Kellner, L.; Fischer, M.; and Tramèr, F. 2024. AgentDojo: A Dynamic Environment to Evaluate Attacks and Defenses for LLM Agents. *arXiv preprint arXiv:2406.13352*.

Duan, C.; and Grimmelmann, J. 2024. Content moderation on end-to-end encrypted systems: A legal analysis. *Geo. L. Tech. Rev.*

Gemini, T.; Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Kumar, P.; Lau, E.; Vijayakumar, S.; Trinh, T.; Team, S. R.; Chang, E.; Robinson, V.; Hendryx, S.; Zhou, S.; Fredrikson, M.; Yue, S.; and Wang, Z. 2024. Refusal-Trained LLMs Are Easily Jailbroken As Browser Agents. https://scale.com/research/browser-art.

Lee, J.; Min, T.; An, M.; Kim, C.; and Lee, K. 2024. Benchmarking Mobile Device Control Agents across Diverse Configurations. *arXiv preprint arXiv:2404.16660*.

Li, N.; Pan, A.; Gopal, A.; Yue, S.; Berrios, D.; Gatti, A.; Li, J. D.; Dombrowski, A.-K.; Goel, S.; Phan, L.; et al. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning*.

Liao, Z.; Mo, L.; Xu, C.; Kang, M.; Zhang, J.; Xiao, C.; Tian, Y.; Li, B.; and Sun, H. 2024. Eia: Environmental injection attack on generalist web agents for privacy leakage. *arXiv preprint arXiv:2409.11295*.

Meta. 2024. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.

Mireshghallah, N.; Kim, H.; Zhou, X.; Tsvetkov, Y.; Sap, M.; Shokri, R.; and Choi, Y. 2024. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory. *International Conference on Learning Representations*.

Mistral. 2024. https://mistral.ai/news/pixtral-large.

OECD. 2019. OECD AI Principles: Robustness, security and safety (Principle 1.4). https://oecd.ai/en/dashboards/ai-principles/P8.

OpenAI. 2024a. https://openai.com/index/hello-gpt-4o/.

OpenAI. 2024b. https://openai.com/index/introducing-openai-o1-preview/.

OpenAI. 2025. https://openai.com/index/introducing-gpt-5/.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*.

Qwen. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Rawles, C.; Clinckemaillie, S.; Chang, Y.; Waltz, J.; Lau, G.; Fair, M.; Li, A.; Bishop, W.; Li, W.; Campbell-Ajala, F.; et al. 2024. AndroidWorld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*.

Rochi, M.; Rauschnabel, P. A.; Renner, K.-H.; and Ivens, B. S. 2024. Technology paternalism: Development and validation of a measurement scale. *Psychology & Marketing*.

Ruan, Y.; Dong, H.; Wang, A.; Pitis, S.; Zhou, Y.; Ba, J.; Dubois, Y.; Maddison, C. J.; and Hashimoto, T. 2024. Identifying the risks of lm agents with an lm-emulated sandbox. In *International Conference on Learning Representations*.

Shinn, N.; Labash, B.; and Gopinath, A. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Conference on Neural Information Processing Systems*.

Wu, C. H.; Koh, J. Y.; Salakhutdinov, R.; Fried, D.; and Raghunathan, A. 2024. Adversarial Attacks on Multimodal Agents. *arXiv preprint arXiv:2406.12814*.

xAI. 2024. https://x.ai/news/grok-1212.

Yang, Z.; Liu, J.; Han, Y.; Chen, X.; Huang, Z.; Fu, B.; and Yu, G. 2023. AppAgent: Multimodal Agents as Smartphone Users. *arXiv preprint arXiv:2312.13771*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.

Yuan, T.; He, Z.; Dong, L.; Wang, Y.; Zhao, R.; Xia, T.; Xu, L.; Zhou, B.; Li, F.; Zhang, Z.; Wang, R.; and Liu, G. 2024. R-Judge: Benchmarking Safety Risk Awareness for LLM Agents. *arXiv preprint arXiv:2401.10019*.

Zhan, Q.; Liang, Z.; Ying, Z.; and Kang, D. 2024. Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents. *arXiv preprint arXiv:2403.02691*.

Zheng, B.; Gou, B.; Kil, J.; Sun, H.; and Su, Y. 2024. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Bisk, Y.; Fried, D.; Alon, U.; et al. 2024. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*.