EVIL-SAFE: A Benchmark for Embodied Vision-Language Safety Inspection by Free Exploration in Home Environment

Anonymous ACL submission

Abstract

Embodied agents can identify and report safety hazards in home settings. Accurately evaluating their ability to perform home safety checks is essential, yet current benchmarks have two major shortcomings. First, they oversimplify the task by using textual descriptions instead of visual inputs, hindering proper evaluation of vision-language model (VLM)-based agents. Second, they rely on a single static viewpoint, limiting exploration and potentially missing hazards that are occluded from fixed angles. To address these issues, we introduce EVIL-SAFE, a benchmark with 12,900 instances covering five common home hazards. EVIL-SAFE provides dynamic first-person view images from simulated home environments, offering multiple dynamic perspectives in complex settings by allowing embodied agents to freely explore rooms, thereby enabling more comprehensive inspection. Our evaluation of mainstream VLMs on EVIL-SAFE reveals significant limitations: even the top model achieves only a 10.23% F1 score, struggling particularly with hazard recognition and exploration planning. We hope EVIL-SAFE will support future research on home safety inspection.[1]

1 Introduction

Homes often present safety hazards due to human negligence, potentially posing a serious threat to residents (Stewart, 2001; Josephson et al., 1991; Goldstick et al., 2022). While regular inspections can prevent these issues, manual checks are time-consuming and labor-intensive. Fortunately, the recent development of vision language models (VLMs) (Alayrac et al., 2022; Liu et al., 2023; Wang et al., 2024; Bai et al., 2025a) has enabled VLM-based embodied agents to perform various practical tasks such as visual exploration, navigation, and embodied question-answering (Duan et al., 2022; Chen et al., 2019; Batra et al., 2020; Ye et al., 2021; Zhao et al., 2025). The embodied VLM agents show great promise for diverse applications, especially within home environments (Yin et al., 2024; Liu et al., 2024). Consequently, the automation of safety inspections using embodied VLMs agents is a promising new area of research.

However, the evaluation of embodied VLM agents in home safety inspection tasks still has significant flaws. Specifically, previous evaluation benchmarks exhibit notable limitations (Mullen Jr et al., 2024; Hassan et al., 2024) primarily in two aspects. First, they convert visual data into textual modalities such as object relationship graphs, for processing by text-only large language models (LLMs). This modality transformation discards critical spatial information, as nuanced spatial concepts are simplified into inadequate positional relationship descriptions in text, thus failing to evaluate the general visual understanding capabilities of VLM-based embodied agents. Second, they rely on fixed-view cameras for hazard identification. The fixed and limited field-of-view is susceptible to occlusion, potentially causing the embodied VLM agents to overlook hazards. To address the lack of visual presentations and flexible viewpoints in existing home inspection benchmarks, we propose EVIL-SAFE, a comprehensive benchmark for **E**mbodied **VI**sion-**L**anguage **SAFE**ty inspection by free exploration in home environment. The construction of EVIL-SAFE combines human annotation and rule-based generation, obtaining a large scale dataset with significant diversity. Throughout and after the construction process, human reviews are adopted to ensure the correctness and high quality of the benchmark. Collectively, EVIL-SAFE contains 12,900 safety inspection tasks with diverse variations based on the simulated environment of VirtualHome (Puig et al., 2018, 2020), covering five prevalent domestic hazard categories: fire, electric shock, falling object, trip, and child safety. Each instance represents a room en-

[1] Our dataset and code will be released soon.

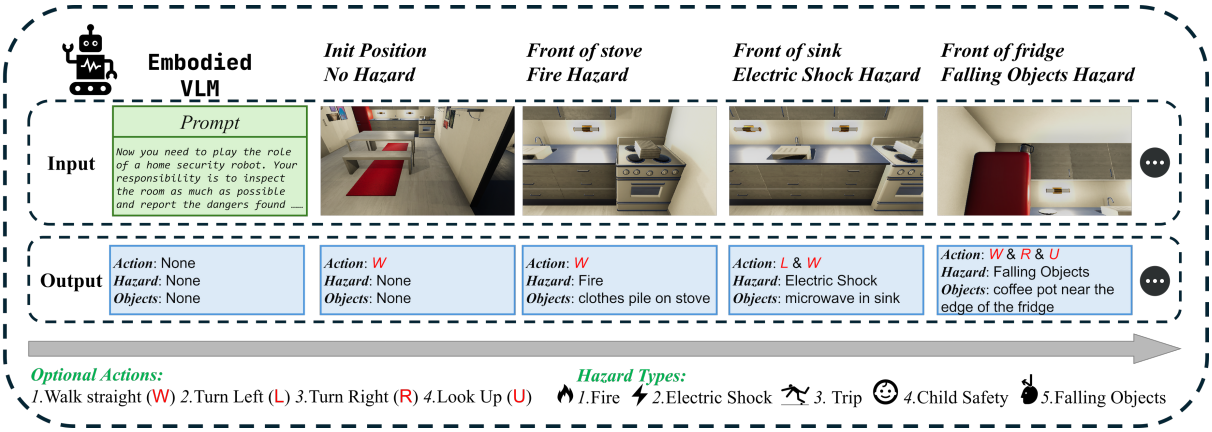


Figure 1: Schematic diagram of the home safety inspection. VLM agents are tasked with identifying the objects that pose a safety hazard given the first-person perspective image from the environment, and select the next action from the action list to iteratively inspect the entire room.

vironment containing multiple hazards, and agents are instructed to autonomously explore the room to report these hazards. During the inspection, the VLM-based agents interacts with the environment to acquire egocentric visual perspectives, identifies hazards within the current field-of-view, and autonomously determines subsequent actions. The process is shown in Figure 1.

We conduct a systematic evaluation on mainstream VLMs using EVIL-SAFE. Our results show that existing VLMs have significant capability deficiency in identifying potential home safety hazards under the paradigm of free exploration with visual feedback. Even the top-performing proprietary VLMs like Qwen-VL-Max and GPT-4o achieve an F1 score under 10%. Our finding indicates that current VLMs are not yet reliable for real-world applications in home safety inspection.

To offer a deeper understanding of the deficiencies of current VLMs in home safety inspection tasks, we conduct an in-depth analysis of the effectiveness of free exploration by embodied VLM agents. Our key findings highlight the importance of free exploration in these tasks, while also revealing a significant weakness in the exploration effectiveness of current embodied VLMs, particularly in complex environments and over a larger number of interaction steps.

Our main contributions are as follows:

- We introduce EVIL-SAFE, a novel benchmark for embodied VLMs that enables visual free-exploration for home safety inspection. It contains 12,900 instances across five common household hazard categories, constructed through a rigorous design and review process.
- We perform a comprehensive evaluation of prevalent VLMs on EVIL-SAFE, revealing their significant limitations in this task and establishing it as a highly challenging benchmark.
- We conduct an in-depth analysis of VLM agents' free exploration during safety inspections to understand the root of their deficiencies. We demonstrate that effective free exploration remains a major challenge, especially in complex, multi-step environments.

2 Related Work

2.1 Vision-Language Model on Embodied AI

Vision-Language Models (VLMs) are widely used in embodied AI to enable agents to perceive, interact with, and understand their environments (Ma et al., 2024; Li et al., 2025b; Shao et al., 2025). Typical applications include vision-language navigation (Anderson et al., 2018; Zhu et al., 2020; Hong et al., 2021; Zhang et al., 2025), embodied question answering (Das et al., 2018; Li et al., 2025c; Zhao et al., 2025), and executing language-driven physical tasks ranging from simple instruction following to long-horizon planning (Gao et al., 2022; Li et al., 2023; Patel et al., 2025; Yang et al., 2025; Sripada et al., 2025).

In this context, we propose EVIL-SAFE, introducing a novel and challenging multi-task scenario for embodied VLMs: comprehensive home safety inspection. Unlike prior tasks, EVIL-SAFE requires the agent to proactively patrol a room to discover as many safety hazards as possible. This demands not only hazard identification but also au-

Dataset	Samples	Hazard Categories	Scene		Task Types			
			Num.	SG	VI	FE	MDS	MT
SafetyDetect	1,000	3	7	✔	✘	✘	✘	✘
M-CoDAL	908	16	✘	✘	✔	✘	✘	✘
SafeAgentBench	750	10	1	✘	✔	✘	✘	✘
SafePlan-Bench	2,027	8	1	✘	✘	✘	✘	✘
EmbodyGuard	942	12	942	✘	✘	✘	✘	✔
IS-Bench	388	10	161	✘	✔	✘	✘	✔
EVIL-SAFE	12,900	5	12	✔	✔	✔	✔	✔

Table 1: Comparison of EVIL-SAFE with existing safety related datsets, including SafetyDetect (Mullen Jr et al., 2024), M-CoDAL (Hassan et al., 2024), SafeAgentBench (Huang et al., 2025), SafePlan-Bench (Huang et al., 2025), EmbodyGuard (Son et al., 2025) and IS-Bench (Lu et al., 2025). In the **Scene**, Num. represents the total number of scenes, and SG respectively represent Scalable Generation. In the **Task Types**, VI, FE, MDS, MT respectively represent Visual Interaction, Free Exploration, Multiple Dangerous Scenarios and Multiple Turns.

tonomous decision-making, advanced spatial perception, and exploration planning, presenting a benchmark of significant practical utility.

2.2 Safety of Embodied AI

With the extensive development of embodied AI, its safety has become a critical research focus. Recent studies in this area broadly fall into two categories: enhancing the safety of embodied AI itself (Yin et al., 2024; Liu et al., 2024; Zhang et al., 2024a; Zhu et al., 2024; Zhang et al., 2024b) and utilizing embodied AI to perform human safety-related tasks (Li et al., 2025a; Zhou et al., 2024; Mullen Jr et al., 2024; Hassan et al., 2024). Research in the first category typically examines the safety of agent operations, employing methods like prompt injection or jailbreaking to execute dangerous human instructions (Yin et al., 2024; Zhang et al., 2024a) or generate risky specific actions (Zhu et al., 2024; Zhang et al., 2024b). The second category involves deploying embodied AI to address real-world hazardous situations, such as analyzing traffic accident causes (Li et al., 2025a), rescuing items in disaster environments (Zhou et al., 2024), or inspecting homes for unsafe conditions (Mullen Jr et al., 2024).

Our work belongs to the second category, aiming to utilize embodied AI for home safety inspection. Compared to prior work, our embodied environment presents a more flexible and challenging scenario. Specifically, the agent is required to autonomously patrol a home to identify safety hazards based on visual feedback, placing higher demands on spatial awareness and path planning capabilities. A detailed comparison is provided in Table 1.

3 EVIL-SAFE

3.1 Task Definition

We propose a home safety hazard inspection task in which an embodied agent actively navigates a simulated 3D home environment to identify and report safety hazards. Following real-world home safety guidelines, we define five categories of common household hazards in our benchmark. Each category represents a specific configuration of item placement that poses a safety risk.

- **Fire Hazards**: Flammable materials are located close to active or potential heat sources. Examples include curtains or stacks of paper placed next to a lit stove, and a pile of dry cloth near a burning candle.
- **Electric Shock Hazards**: Appliances or power devices in contact with water, which may cause electric shock or short circuits. Examples include an appliance in a sink or a toilet.
- **Falling Object Hazards**: Items positioned in a way that they may fall from height and cause injury or damage. Examples include a coffee pot placed at the edge of a refrigerator, or a box positioned at the edge of a shelf.
- **Trip Hazards**: Objects or clutter on the floor that could cause someone to stumble or lose balance during normal movement. Examples include a bar of soap left in a hallway.

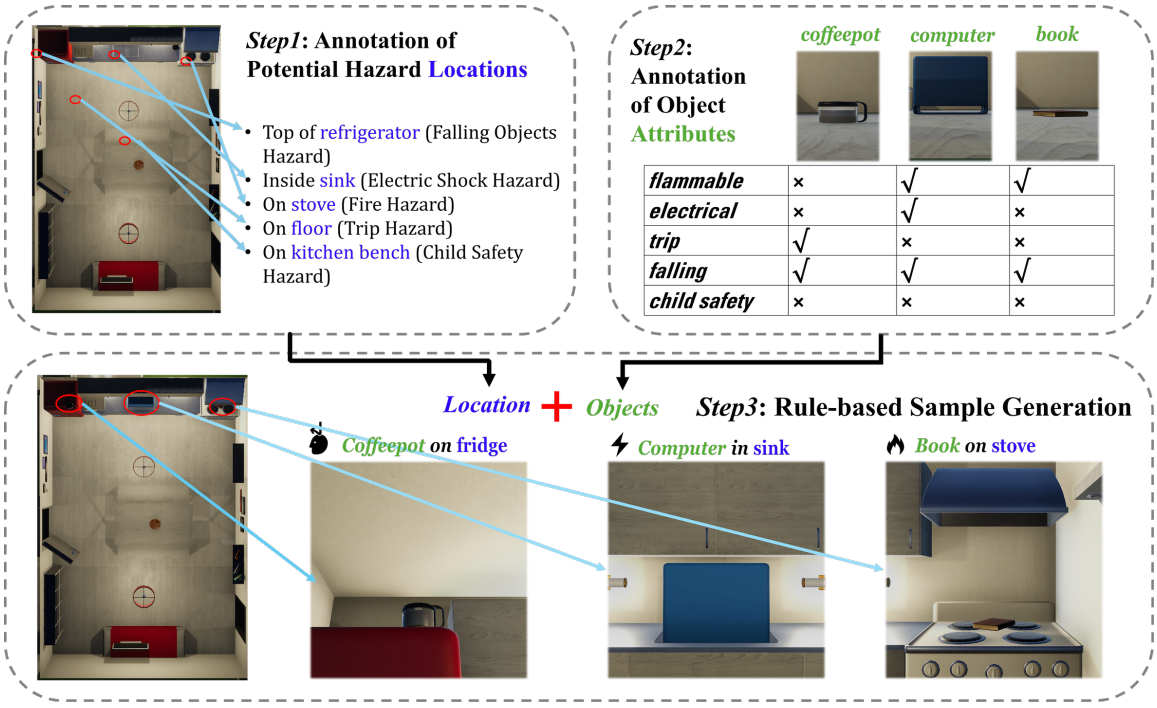


Figure 2: Annotation process of EVIL-SAFE.

- **Child Safety Hazards**: Placement of dangerous or harmful items within easy reach of a child. Examples include a bottle of alcohol on a low table, or sharp kitchen knives placed on TV stand.

Formally, let the initial state denoted as s_0 with a ground-truth hazard set $\mathcal{H}$. At each discrete time step t, the agent policy π is to identify hazards $\hat{\mathcal{H}}_t$ with the current observation, and select an action $a_t \in \mathcal{A}$. The state is then transitioned following the transition function f, updating the observation.

$$\hat{\mathcal{H}}_t, a_t \sim \pi(\cdot|s_t), \quad s_{t+1} = f(s_t, a_t). \tag{1}$$

The action space $\mathcal{A}$ consists of basic navigation primitives such as `move-forward`, `turn-left`, `turn-right`, and `look-up`, as detailed in Appendix A. After executing a sequence of actions $\{a_0, a_1, \ldots, a_{T-1}\}$ within a step budget T, the final identified hazard set is defined as the union of the hazard sets identified at each step.

$$\hat{\mathcal{H}} = \bigcup_{i=0}^{T-1} \hat{\mathcal{H}}_t. \tag{2}$$

Task performance is evaluated by comparing the reported hazards $\hat{\mathcal{H}}$ against the ground-truth hazards $\mathcal{H}$ using the precision, recall, and F1 score. A hazard is considered correctly reported if both its category and the name of the associated item are correct. Note that we do not require a perfect match for the item name. Instead, we use a rule-based matching system for a more flexible and reliable evaluation, as detailed in Appendix A.

$$\text{Precision}(\hat{\mathcal{H}}, \mathcal{H}) = \frac{||\hat{\mathcal{H}} \cap \mathcal{H}||}{||\hat{\mathcal{H}}||}, \tag{3}$$

$$\text{Recall}(\hat{\mathcal{H}}, \mathcal{H}) = \frac{||\hat{\mathcal{H}} \cap \mathcal{H}||}{||\mathcal{H}||}, \tag{4}$$

$$\text{F1}(\hat{\mathcal{H}}, \mathcal{H}) = \frac{2 \times ||\hat{\mathcal{H}} \cap \mathcal{H}||}{||\hat{\mathcal{H}}|| + ||\mathcal{H}||}. \tag{5}$$

3.2 Construction and Quality Control Procedure

EVIL-SAFE is constructed based on the engine of VirtualHome by combining manual annotation and rule-based generation. The careful reviews are conducted through out the construction process to ensure the correctness and quality of the benchmark. The construction process consists of three stages, as shown in Figure 2.

Annotation of Potential Hazard Locations Firstly, the spatial locations within the virtual environment that are likely to contain safety hazards are annotated, such as the top of a refrigerator, inside a sink, or on a stove. The process is performed by two annotators, each responsible for six

	Fire	Electric Shock	Trip	Falling object	Child Safety
# of Hazard Locations	17	3	51	13	19
# (and %) of Samples	958 (60%)	417 (26%)	1486 (94%)	973 (61%)	716 (45%)
# Hazard Locations	22	10	58	24	22
# (and %) Samples	9051 (70%)	3558 (28%)	11019 (85%)	8836 (69%)	8999 (70%)

Table 2: Statistics of hazard types in EVIL-SAFE for subset (top) and full set (bottom).

rooms across three environments (12 rooms in total). To ensure annotation quality, the annotators cross-verify each other's annotation case by case, filter out locations with low risks, and the final annotations reflect consensus between the annotators. Each identified location is assigned exactly one hazard type tag. In total, we obtain 136 annotated hazard locations across all environments.

Annotation of Object Attributes Then, the common objects in the virtual environment are assigned with a predefined set of attributes, including flammable, electrical, tripping hazard, falling object, and child safety hazard. An object may be assigned multiple attributes. In total, 367 objects across the three environments are separately annotated by two annotators, and any disagreement between them is referred to a third annotation for a consolidated assignment.

Rule-based Sample Generation Finally, the final samples of EVIL-SAFE are generated following the combination rule of locations and objects. Specifically, based on the potential hazard types associated with these locations, we place suitable objects with corresponding attributes at the sampled positions. For instance, a location tagged as fire hazard (e.g., a stove) may have an object of paper placed on it, while a location tagged as falling object hazard (e.g., the top of a refrigerator) may have a glass cup assigned to it.

For quality control of the final samples, we conduct tests in two ways. First, we randomly select 100 samples, examine every hazard of the gold labels in the virtual environment. It is verified that all the hazard points marked are indeed risky and the placement of items is visually discoverable. Second, we randomly select other 100 sample, and manually conduct inspection with no golden label given. The Precision, Recall, and F1 scores achieve 82.29%, 69.50%, and 75.36% for human inspection, validating that the tasks in EVIL-SAFE are solvable.

3.3 Dataset Statistics

Collectively, EVIL-SAFE contains 12,900 samples generated from 12 unique scenes, covering five types of hazards. Considering its relatively large size, we select a small subset sized 1,580 as detailed in Appendix A for quick experiments. Hazard statistics are shown in Table 2.

4 Experiments

4.1 Settings

Models Following others work(Zhu et al., 2024; Yin et al., 2024), and considering the information loss during image-to-text conversion, we chose VLMs over LLMs as the foundation models for embodied home safety inspection agents. We comprehensively test mainstream VLMs. Open-sourced models are locally deployed, including Qwen2.5-VL-7B (Bai et al., 2025b), InternVL2.5-4B, InternVL2.5-8B (Chen et al., 2024), Llama3.2-11B-V (Dubey et al., 2024), and Gemma3-12B (Team et al., 2025). proprietary models are called through API, including Qwen-VL-Max (Bai et al., 2023) and GPT-4o (Hurst et al., 2024).

Inference The open-sourced models are locally deployed with `transformers` (Wolf et al., 2020). We set temperature as 0.6, top-p as 0.9 during sampling for all models. We don't use greedy decoding to avoid the endless repeation of the generated actions.

Agent Design The interaction flow between the VLM agents and the virtual environments is illustrated in Figure 1. In the design of VLM embodied agents, we allow the VLM to control the agent's free exploration by generating navigation primitives such as `move-forward`, `turn-left`, `turn-right`, and `look-up`. The agents perform a 10-turn dialogue-based room inspection,where environment transmits a first-person perspective image to the VLM, and the VLM identifies and reports safety hazards based on the image, then autonomously decides the next action. Agent

Models	Subset			Others			All		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Qwen2.5-VL-7B	5.16	2.42	2.91	1.61	0.87	1.02	1.97	1.03	1.21
InternVL2.5-4B	7.98	2.73	3.66	4.10	1.25	1.69	4.57	1.43	1.93
InternVL2.5-8B	9.38	3.05	4.06	7.42	1.91	2.87	7.67	2.05	3.01
Llama3.2-11B-V	**9.77**	17.84	**11.84**	7.87	11.11	8.78	8.11	11.93	9.16
Gemma3-12B	9.00	**18.14**	11.46	8.51	**12.93**	**10.06**	8.57	**13.57**	**10.23**
Qwen-VL-Max	7.26	3.15	4.03	5.13	1.64	2.23	5.15	1.76	2.44
GPT-4o	7.67	4.44	5.42	**10.30**	6.15	6.89	**9.91**	5.89	6.67

Table 3: Main results of embodied VLMs on EVIL-SAFE. Best scores among all models are shown in bold. Prec and Rec refer to Precision and Recall respectively.

prompts, and other implementation details are listed in Appendix B.

Metrics We report the micro average of precision, recall, and F1 scores as the final metrics following Equation 5. Specifically, we calculate the metrics for each task instance, and average across the dataset sourced from EVIL-SAFE.

4.2 Main Result

In the main experiment, we evaluate VLMs using the complete EVIL-SAFE dataset. The results are shown in Table 3. In safety hazard identification, all models scores below 20% on Precision, Recall, and F1. Comparing to human inspectors that obtaining 82.29%, 69.50%, and 75.36% Precision, Recall, and F1 on a subset, it can be concluded that current VLMs have a very poor performance on home safety inspection tasks. Even the commercial models Qwen-VL-Max and GPT-4o, which perform well on many tasks, achieved low scores comparable to smaller models. Gemma3-12B achieved the best performance among all, with a recalling 13.57% of all hazards. Qwen2.5-VL-8B achieved the lowest score, whose action selection is single-minded, usually selecting and repeatedly executing only one action. Furthermore, Qwen2.5-VL-8B was more likely to choose the passive “None" answer rather than proactively identifying safety hazards. To further understand the poor performance of current VLMs, we conduct a analysis on their free exploration behaviors in Section 5, and introduce case studies in Appendix D.

As detailed in Appendix A, the subset contains more hazard points in obvious places which are easier to notice. Comparison between results of different test sets show that the subset generally obtains higher scores, which meets our expectations. However, the final results are still very low compared to human scores (75.36% F1 score), again validating the tasks of home safety inspection from EVIL-SAFE pose strong challenges to VLMs.

5 Analysis

In the introduction to EVIL-SAFE, the free exploration paradigm plays a crucial role in our proposed EVIL-SAFE. To gain a deeper understanding of the role of free exploration in home safety inspection tasks, we conducted an in-depth analysis to answer four research questions (RQs) regarding free exploration. More additional experiments can be found in the appendix E.

- **RQ1**: Is free exploration useful for home safety inspection task?
- **RQ2**: How is the free exploration performance of current embodied VLMs?
- **RQ3**: How does the multi-turn interaction affect exploration effectiveness?

RQ1: Importance of Free Exploration

Intuitively, a fixed single viewpoint of agents can cause problems such as blurring of distant objects and obstruction of safety hazards by other objects. To validate the impact of agent free exploration on inspection capabilities, we design an inspection experiment without it. The experiment is conducted with 10% randomly sampled data of EVIL-SAFE, ensuring that each type of room and environment is included. We fix the agent in a corner of the room and rotated it to ensure that the agent’s first-person perspective could see the entire room.

The results are shown in Table 4. When deprived of the ability to explore freely, all models suffer

Models	Precision			Recall			F1		
	w/	w/o	Δ	w/	w/o	Δ	w/	w/o	Δ
Qwen2.5-VL-7B	1.97	11.09	-9.12	1.03	0.41	+0.62	1.21	0.79	+0.42
InternVL2.5-4B	4.57	3.58	+0.99	1.43	0.55	+0.88	1.93	0.94	+0.99
InternVL2.5-8B	7.67	4.03	+3.64	2.05	0.63	+1.42	3.01	1.08	+1.93
Llama3.2-11B-V	8.11	5.64	+2.47	11.93	1.51	+10.42	9.16	2.32	+6.84
Gemma3-12B	8.57	19.17	-10.60	13.57	3.41	+10.16	10.23	5.69	+4.54
Qwen-VL-Max	5.15	16.74	-11.59	1.76	1.13	+0.63	2.44	2.12	+0.32
GPT-4o	9.91	14.00	-4.09	5.89	2.33	+3.56	6.67	3.96	+2.71

Table 4: The performance with (w/) and without (w/o) free exploration, and the corresponding difference between the two paradigms (Δ).

from a consistent and significant decrease in F1 scores. Although Qwen2.5-VL, Qwen-VL-Max, and Gemma3-12B models achieve improvements in precision compared to when deprived, they suffer from a more significant recall drop due to insufficient environmental information and occlusion, finally scoring lower F1. These results indicate that enabling the VLM to control the agent's free exploration is a key factor in ensuring the model's effectiveness in safety inspection tasks.

RQ2: Performance of Free Exploration

Given that free exploration plays a significant positive role in home safety inspection by offering flexible viewpoints, we are interested in the performance of current VLM-based agents in effectively conduct exploration. In doing so, we introduce the Navigation (Nav) metric to reflect the agent's ability to sufficiently navigate the entire environment. Specifically, we use the built-in function `get_visible_objects` in VirtualHome to record the objects visible to the agent, and then calculate the ratio of the number of hazards into the occurrence of the agent view, to the total number of hazards.

$$\text{Navigation} = \frac{||\mathcal{O}_{vis} \cap \mathcal{H}||}{||\mathcal{H}||}, \tag{6}$$

where $\mathcal{O}_{vis}$ refers to the set of all objects visible to the agent, and $\mathcal{H}$ is the ground truth hazard set. The experimental results are shown in Table 5.

First, all the models perform badly on navigation, with less than 50% of the risk points included in the observation throughout the inspection. The poor navigation scores partially explain the deficiencies of current embodied VLMs in home inspection task. Since embodied VLMs have significant difficulties navigating to extensively observe the items, it becomes less likely for them to identify the hazards and get a high recall rate.

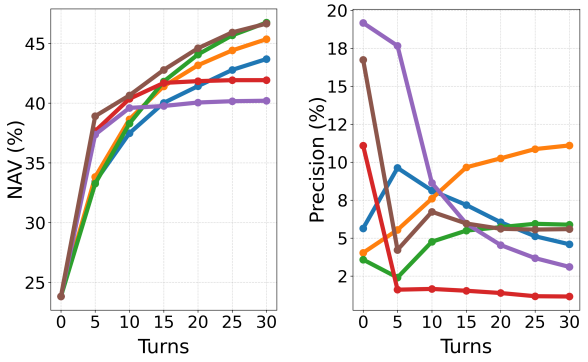
Second, according to the experimental results, the model performs different on navigation and F1 scores in different rooms. All models perform poorly in the living room, likely due to the large number of items such as sofas and TV tables, which complicates the environment and presents a greater challenge for the models. In the bathroom, most models achieve relatively strong scores, likely due to the smaller size and fewer items of the room which makes it easier for the models to inspect the entire room. The difference between rooms indicate that current VLM agents still struggle with conducting effective navigation in complex environments.

RQ3: Free Exploration under Multi-Turn Interaction

The free exploration in EVIL-SAFE involves multi-turn interaction with the virtual environment. A natural question under this paradigm is, how will the effectiveness of free exploration be affected as the number of turn grows. Therefore, we conduct an investigation by setting the maximum number of turns to 30, and calculating the model score every five turns. This maximum number of turns is set to 30, as empirical evidences suggest that the models tend to output content unrelated to the task, such as descriptions of the environment, and the performance will not change significantly when the number of turns exceeds 30. The model's Navigation, Precision, Recall, and F1 scores as a function of the number of turns as line graphs are shown in Figure 3.

Models	Kitchen		Bathroom		Bedroom		Livingroom		All	
	Nav	F1	Nav	F1	Nav	F1	Nav	F1	Nav	F1
Qwen2.5-VL-7B	36.77	0.70	54.12	1.24	40.98	3.73	23.73	0.79	33.64	1.21
InternVL2.5-4B	38.36	2.58	53.27	2.69	44.55	2.02	37.10	0.63	39.45	1.93
InternVL2.5-8B	37.44	3.10	53.31	5.03	42.24	2.43	40.88	2.53	40.32	2.86
Llama3.2-11B-V	37.06	9.94	53.69	**12.07**	41.41	6.91	38.46	8.86	39.24	9.16
Gemma3-12B	43.72	9.21	**60.78**	11.18	47.33	**10.78**	25.22	**11.68**	39.54	**10.23**
Qwen-VL-Max	36.77	3.29	54.12	5.11	43.98	2.51	44.03	0.93	41.24	2.44
GPT-4o	**44.22**	**10.45**	52.64	10.01	**51.36**	3.34	**51.89**	2.87	**49.94**	6.67

Table 5: The navigation (Nav) and F1 scores across different types of rooms. The navigation score of an agent is defined as the ratio of the number of observed hazards to that of all hazards. The highest scores among models are shown in bold.



(a) Navigation (b) Precision

(c) Recall (d) F1

Figure 3: VLM performance changes as a function of the number of action turns. Sub-figure (a), (b), (c), and (d) shows the change of Navigation, Precision, Recall, and F1 score of each VLM with action turns.

As the number of interaction turn grows, the F1 score does not necessarily increases monotonically with it. Instead, the performance general reaches its high value or even its peak at the first few steps, and then saturates or even decrease along with more steps. Analysis on other metrics other than F1 scores offers further insights about the long-horizon free exploratino process. Along with the turn number increases, the Navigation score and Recall score gain per five turns gradually slows down, indicating that the free exploration and hazard identification are less and less effective for later turns in the whole process. Meanwhile, the Precision score significantly decreases when turn number grows, also demonstrating the decrease of inspection quality after a certain number of turns.

Generally, the analysis highlights another weakness of current embodied VLMs in the free exploration of home safety inspection tasks, namely the effectiveness drop under a long horizon of multi-turn interaction. These VLM-based agents lack clear and solid planning to conduct a well-organized inspection, but conduct exploration and identification in a arbitrary way, thus obtaining little gain in a long-sequence task.

6 Conclusion

To address the limitations of fixed viewpoints and missing visual information in home safety evaluation, we propose EVIL-SAFE, a benchmark featuring first-person perception and interactive free exploration for embodied VLMs. EVIL-SAFE includes 12,900 samples across five common household hazard categories, with quality ensured through human review.

Using EVIL-SAFE, we systematically evaluate mainstream VLMs, revealing significant shortcomings in hazard identification and exploration. Our in-depth analysis of free exploration patterns highlights its importance while exposing VLMs' weaknesses in navigation within complex, multi-turn environments.We hope EVIL-SAFE can establish a foundation for embodied safety and guide improvements in VLM navigation and understanding.

Limitation

We introduced EVIL-SAFE, designed to evaluate a model's inspection capabilities in a home environment. Because EVIL-SAFE includes a large number of diverse scenarios and requires multiple rounds of real-time interaction with the virtual environment, the evaluation consumes significant resources. Furthermore, due to limitations in the model's context length, we cannot retain all interaction records, only key information.

Ethics Statement

Our work belongs to LLM embodied agent work. This type of work has risks. However, as a benchmark in its previous application field, our work does not introduce additional risks on top of the basic risks. We have developed a security check-related benchmark that will help advance the theory and application of embodied VLM security and has significant positive potential impact.

Our work falls under the category of embodied agents controlled by VLMs, a field that carries certain inherent risks. However, as a benchmark built upon existing theory and application, our study introduces little additional risks beyond current ones. Moreover, this work proposes EVIL-SAFE specifically for security checks, which aims to enhance the safety of embodied VLM systems and thus has a substantial positive impact.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025a. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025b. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. 2020. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*.

Tao Chen, Saurabh Gupta, and Abhinav Gupta. 2019. Learning exploration policies for navigation. *arXiv preprint arXiv:1903.01959*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10.

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. 2022. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S Sukhatme. 2022. Dialfred: Dialogue-enabled agents for embodied instruction following. *IEEE Robotics and Automation Letters*, 7(4):10049–10056.

Jason E Goldstick, Rebecca M Cunningham, and Patrick M Carter. 2022. Current causes of death in children and adolescents in the united states. *New England journal of medicine*, 386(20):1955–1956.

Sabit Hassan, Hye-Young Chung, Xiang Zhi Tan, and Malihe Alikhani. 2024. Coherence-driven multimodal safety dialogue with active learning for embodied agents. *arXiv preprint arXiv:2410.14141*.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. 2021. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1643–1653.

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

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Karen R Josephson, Diana A Fabacher, and Laurence Z Rubenstein. 1991. Home safety and fall prevention. *Clinics in geriatric medicine*, 7(4):707–732.

Cheng Li, Keyuan Zhou, Tong Liu, Yu Wang, Mingqiao Zhuang, Huan-ang Gao, Bu Jin, and Hao Zhao. 2025a. Avd2: Accident video diffusion for accident video description. *arXiv preprint arXiv:2502.14801*.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, and 1 others. 2023. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR.

Haoran Li, Yuhui Chen, Wenbo Cui, Weiheng Liu, Kai Liu, Mingcai Zhou, Zhengtao Zhang, and Dongbin Zhao. 2025b. Survey of vision-language-action models for embodied manipulation. *arXiv preprint arXiv:2508.15201*.

Pengna Li, Kangyi Wu, Jingwen Fu, and Sanping Zhou. 2025c. Regnav: Room expert guided image-goal navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4860–4868.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Shuyuan Liu, Jiawei Chen, Shouwei Ruan, Hang Su, and Zhaoxia Yin. 2024. Exploring the robustness of decision-level through adversarial attacks on llm-based embodied models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8120–8128.

Xiaoya Lu, Zeren Chen, Xuhao Hu, Yijin Zhou, Weichen Zhang, Dongrui Liu, Lu Sheng, and Jing Shao. 2025. Is-bench: Evaluating interactive safety of vlm-driven embodied agents in daily household tasks. *Preprint*, arXiv:2506.16402.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. 2024. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*.

James F Mullen Jr, Prasoon Goyal, Robinson Piramuthu, Michael Johnston, Dinesh Manocha, and Reza Ghanadan. 2024. "don't forget to put the milk back!" dataset for enabling embodied agents to detect anomalous situations. *IEEE Robotics and Automation Letters*.

Shivansh Patel, Xinchen Yin, Wenlong Huang, Shubham Garg, Hooshang Nayyeri, Li Fei-Fei, Svetlana Lazebnik, and Yunzhu Li. 2025. A real-to-sim-to-real approach to robotic manipulation with vlm-generated iterative keypoint rewards. *arXiv preprint arXiv:2502.08643*.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8494–8502.

Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Joshua B. Tenenbaum, Sanja Fidler, and Antonio Torralba. 2020. Watch-and-help: A challenge for social perception and human-ai collaboration. *Preprint*, arXiv:2010.09890.

Rui Shao, Wei Li, Lingsen Zhang, Renshan Zhang, Zhiyang Liu, Ran Chen, and Liqiang Nie. 2025. Large vlm-based vision-language-action models for robotic manipulation: A survey. *arXiv preprint arXiv:2508.13073*.

Yejin Son, Minseo Kim, Sungwoong Kim, Seungju Han, Jian Kim, Dongju Jang, Youngjae Yu, and Chan Young Park. 2025. Subtle risks, critical failures: A framework for diagnosing physical safety of LLMs for embodied decision making. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25692–25733, Suzhou, China. Association for Computational Linguistics.

Venkatesh Sripada, Samuel Carter, Frank Guerin, and Amir Ghalamzan. 2025. Scene exploration by vision-language models. *Preprint*, arXiv:2409.17641.

Jill Stewart. 2001. Home safety. *The journal of the Royal Society for the Promotion of Health*, 121(1):16–22.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, and 1 others. 2025. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*.

Joel Ye, Dhruv Batra, Erik Wijmans, and Abhishek Das. 2021. Auxiliary tasks speed up learning point goal navigation. In *Conference on Robot Learning*, pages 498–516. PMLR.

Sheng Yin, Xianghe Pang, Yuanzhuo Ding, Menglan Chen, Yutong Bi, Yichen Xiong, Wenhao Huang, Zhen Xiang, Jing Shao, and Siheng Chen. 2024. Safeagentbench: A benchmark for safe task planning of embodied llm agents. *arXiv preprint arXiv:2412.13178*.

Hangtao Zhang, Chenyu Zhu, Xianlong Wang, Ziqi Zhou, Shengshan Hu, and Leo Yu Zhang. 2024a. Badrobot: Jailbreaking llm-based embodied ai in the physical world. *arXiv preprint arXiv:2407.20242*, 3.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, and 1 others. 2025. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*.

Wenxiao Zhang, Xiangrui Kong, Thomas Braunl, and Jin B Hong. 2024b. Safeembodai: a safety framework for mobile robots in embodied ai systems. *arXiv preprint arXiv:2409.01630*.

Baining Zhao, Ziyou Wang, Jianjie Fang, Chen Gao, Fanhang Man, Jinqiang Cui, Xin Wang, Xinlei Chen, Yong Li, and Wenwu Zhu. 2025. Embodied-r: Collaborative framework for activating embodied spatial reasoning in foundation models via reinforcement learning. *arXiv preprint arXiv:2504.12680*.

Qinhong Zhou, Sunli Chen, Yisong Wang, Haozhe Xu, Weihua Du, Hongxin Zhang, Yilun Du, Joshua B Tenenbaum, and Chuang Gan. 2024. Hazard challenge: Embodied decision making in dynamically changing environments. *arXiv preprint arXiv:2401.12975*.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. 2020. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10012–10022.

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

Appendix

A Benchmark Details

A.1 Virtual Environment

We utilize VirtualHome as the virtual home environment, and the agent within this environment is simulated using the “Character" module from VirtualHome. Specifically, an agent can be added using the `add_character` function. We adopt the `FIRST_PERSON` camera of the “Character" as the primary egocentric viewpoint for all actions except for the “Look Up" action.

Regarding the actions described in Section 3.1, we have modified the original actions provided by VirtualHome as follows:

Walk Straight In VirtualHome, the “walkforward" action moves the agent forward by only one step per execution. However, a single step results in minimal change in the agent’s field of view, and traversing an entire room would require an excessive number of meaningless steps. To address this, we define “Walk Straight" as moving forward three steps in sequence, i.e., executing three consecutive “walkforward" actions.

Turn Left & Turn Right In VirtualHome, each “turnleft" or “turnright" action rotates the agent by 30 degrees. Similar to the walking action, this would lead to inefficient and repetitive rotations. Therefore, we define “Turn Left" and “Turn Right" as rotating the agent by 90 degrees, achieved by executing three consecutive “turnleft" or “turnright" actions.

Look Up VirtualHome does not natively provide a “look up" action. To implement this, we attach an upward-facing camera, named “up_camera", to the agent during its initialization. The relative position of this camera is set to $Position = [0, 1.5, 0]$ and its rotation to $Rotation = [-15, 0, 0]$. When the agent performs a “Look Up" action, the image captured by the “up_camera" is used.

A.2 Small Subset Selection

During the location selection phase, annotators identify 57 conspicuous locations, such as tabletops and floors. Subsequently, from all samples, we

form a subset comprising entries where over 50% of hazard locations are located in such conspicuous locations. This subset contains 1,586 data instances and is characterized as relatively simpler compared to the complete dataset.

A.3 Evaluation Details

To address the issue that objects in the environment may be referred to by multiple names, which can lead to evaluation inaccuracies if a model uses a synonym instead of the standard name defined in the environment, we create a mapping that links alternative names of objects to their canonical names used in the virtual environment.

This mapping is constructed using GPT-4o, which generated preliminary synonym associations based on the standard object names. The initial mappings were then reviewed and refined by two human annotators to ensure relevance and accuracy, resulting in a finalized mapping table. An example of this mapping is provided in the Table 6.

Standard Name	Mapping Names
Wallshelf	Bookshelf, Rack, Shelf
Candle	Wax, Light, flame
Computer	PC, Laptop, Desktop
Toaster	Bread heater, Bread maker, Tost machine
Clothespants	Pants, Trousers, Jeans
Fridge	Refrigerator

Table 6: Examples of Mapping

During evaluation, object names predicted by the model are first mapped to their standard canonical names before scoring is performed.

B Experiment Details

B.1 Inference details

During model inference, we utilize the PyTorch framework and the Transformers library, with the versions of the Python libraries aligned with the requirements of each respective model. The inference process is conducted using six NVIDIA TITAN RTX GPUs in parallel. Executing inference on the entire dataset requires approximately 24 to 48 hours.

B.2 Prompt

Our experiments used multiple prompts. To ensure reproducibility, this section demonstrates the prompts used.

In the main experiment, we used EVIL-SAFE to evaluate a mainstream VLM. To ensure the VLM fully understood the task, we split the inspection task into Task 1 and Task 2, describing each task's requirements in detail. For ease of evaluation, we specified the output format and required the VLM to strictly adhere to it. We also provided three examples to facilitate contextual learning for the VLM. As shown in Figure 4.

In the Without Action analysis experiment, we deleted Task 2 and made appropriate modifications to the prompt to ensure that the VLM could complete the task, as shown in Figure 5.

B.3 Usage of Data

In Section 4, we evaluated all models except GPT-4o using the full dataset. In Section 5, a randomly selected 10% subset of the data was used for the experiments in Sections 5.1 and 5.3. The specific data selection and processing procedures for individual experiments are detailed below.

GPT-4o For both the evaluation and analysis experiments involving GPT-4o, we used a randomly selected sample of 100 instances, ensuring the sample included data from all room types.

Data for the Without-exploration Experiment In Section 5.1, we utilized the selected subset of data and adjusted the agent's viewpoint to ensure a complete view of the entire room. The agent's starting position for each data instance was consistent with the initial position used in the Section 4 experiments. The rotation angles applied in each room are specified in Table 7.

Room	Env0	Env1	Env2
Kitchen	R30°	R30°	R30°
Bedroom	R30°	R30°	R30°
Livingroom	L30°	L60°	L60°
Bathroom	R30°	R30°	L30°

Table 7: Initialize of Without-exploration. R stands for "turn right", and L stands for "turn left".

Additionally, the camera configuration was adjusted. A camera with a relative position of $Position = [0, 1.5, 0]$ and rotation of $Rotation = [0, 0, 0]$ was attached to the agent, serving as the primary image capture camera.

Prompt for HomeSafeBench

You are a home security robot responsible for inspecting the room thoroughly and reporting any dangers found. The input image represents your current first-person perspective. Please complete the following two tasks **in sequence** and follow the specified output formats exactly.

Task 1: Identify Safety Hazards

- Detect if there are any safety hazards in your current view.
- Hazard categories to identify: [fire, electric shock, falling object, trip, child safety].
- For each hazard detected, output the hazard category and the name of the object causing it (only the object name).
- If multiple hazards exist, output all of them. Use the format:
  `Task1: {{hazard1, object1; hazard2, object2; ...}}`
  where each `{{hazard, object}}` pair is enclosed in braces and multiple pairs are separated by semicolons.
- If no hazards are detected, reply with:
  `Task1: {{None}}`

Task 2: Choose Your Next Operation

- Select one operation from: [walk straight, turn left, turn right, look up].
- Choose an operation that helps you inspect the room as much as possible without bumping into objects.
- Output format:
  `Task2: {{your next operation}}`

Examples:

Example 1:

Task1: {{fire, stove}}, Task2: {{turn right}}.

Example 2:

Task1: {{slips and falls, box}}, Task2: {{walk straight}}.

Example 3:

Task1: {{falling objects, coffeepot; electric shock, microwave}}, Task2: {{turn right}}.

Strictly follow the above formats without adding any other content or explanation.

Your previous action was: {history}. Please complete the two tasks accordingly.

Figure 4: Prompt using in EVIL-SAFE

Prompt for Without Action

You are a home security robot responsible for inspecting the room thoroughly and reporting any dangers found. The input image represents your current first-person perspective. Please complete the following task and follow the specified output format exactly.

Task: Identify Safety Hazards

- Detect if there are any safety hazards in your current view.
- Hazard categories to identify: [fire, electric shock, falling object, trip, child safety].
- For each hazard detected, output the hazard category and the name of the object causing it (only the object name).
- If multiple hazards exist, output all of them. Use the format:

  `Task: {{hazard1, object1; hazard2, object2; ...}}`

  where each {{hazard, object}} pair is enclosed in braces and multiple pairs are separated by semicolons.

Examples:

Example 1:

Task: {{fire, stove}}

Example 2:

Task: {{slips and falls, box}}

Example 3:

Task: {{falling objects, coffeepot; electric shock, microwave}}

Strictly follow the above formats without adding any other content or explanation.

Figure 5: Prompt using in without action

Figure 6: Example of Hazard Type. The left image in the first row illustrates a fire hazard, the center depicts an electric shock hazard, while the right image demonstrates a falling object hazard. The second row presents a child safety hazard on the left and a trip hazard on the right, respectively.

C Dataset Details

In this section, we employ examples of hazard locations and types to enhance the clarity of the dataset's details.

C.1 Example of Hazard Type

EVIL-SAFE contains five distinct hazard types, and we provide a representative example for each category, as illustrated in Figure 6.

C.2 Example of Hazard location

EVIL-SAFE dataset comprises 3 distinct environments, each containing 4 different room types, resulting in a total of 12 scenarios. During the data construction phase, the process requires the initial annotation of hazard locations. To illustrate the specifics of our annotation methodology, for all 12 scenarios, a number of Hazard locations are selected, ensuring that at least one instance of each type of hazard is included. An example of a hazard location is presented in Figure 7.

D Case study

To analyze the inaccurate recognition of VLMs, we selected some representative results and analyzed the errors of the model in Navigation and Objects Identification respectively.

Navigation Table 3 and Table 5 shows that all models achieve relatively low recall and Nav scores, suggesting that their suboptimal performance may stem from poor navigation when patrolling rooms. The figure below illustrates several example trajectories of the models operating within the rooms. Figure 8 shows two trajectories produced by GEMMA. In both examples, although the navigation paths differ, the model consistently misses the slip hazard placed on the floor. Notably, the hazard is never observed throughout the entire process.

Objects Identification In addition to navigation errors, models may also produce object recognition mistakes or fail to identify hazards. For example, as shown in Figure 9, even when the model *qwen-max* navigates correctly and observes the hazard *(a computer in the sink, belonging to the **electric shock hazard** category)* within its field of view, it still fails to report it.

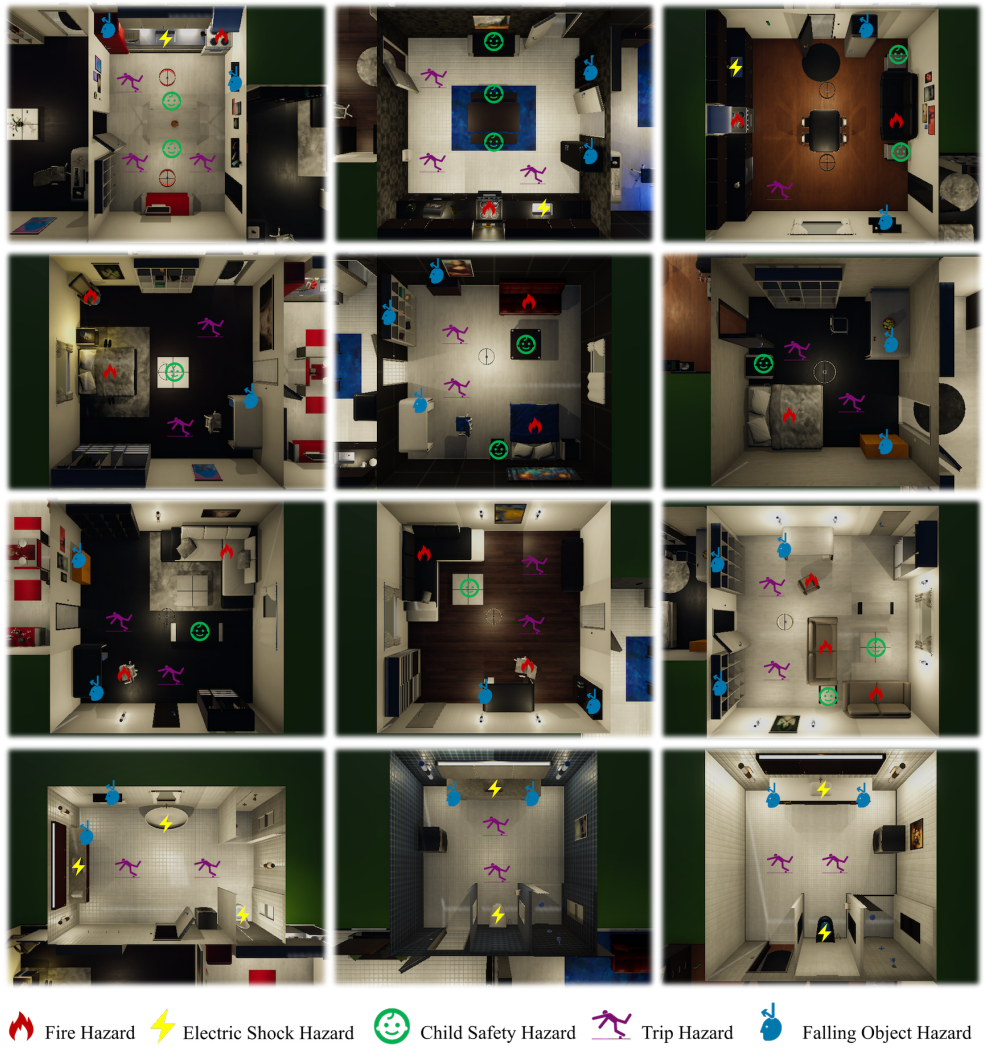
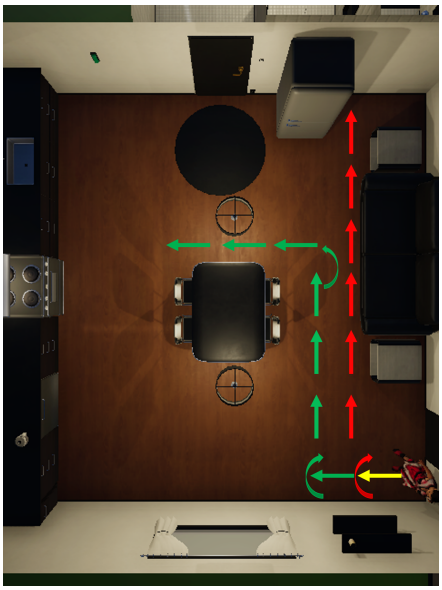
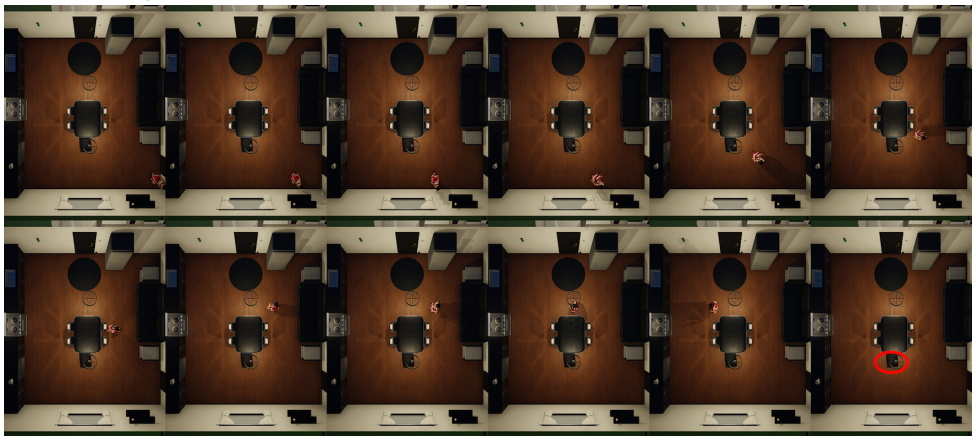


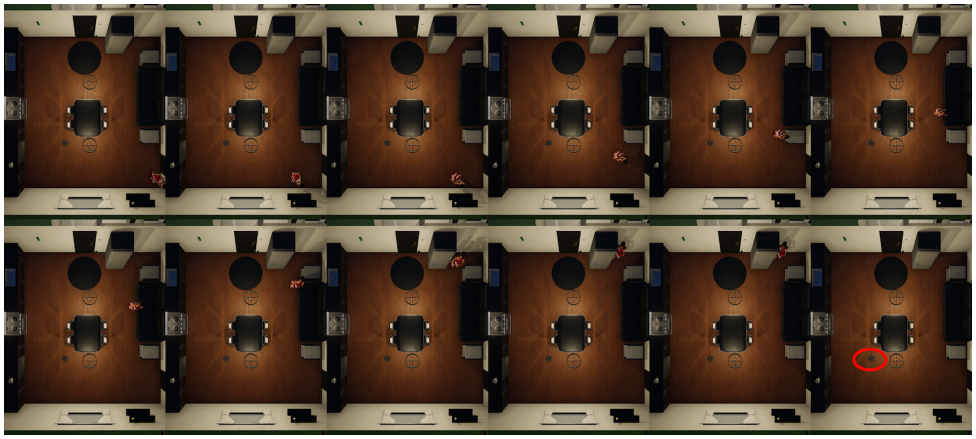
Figure 7: Example of hazard location



(a) Brief paths (Different colors represent different paths, with yellow indicating the same action).



(b) Detailed path (green one in 8a).



(c) Detailed path (red one in 8a).

Figure 8: Example paths of Gemma3 (Hazards are highlighted with red circles).

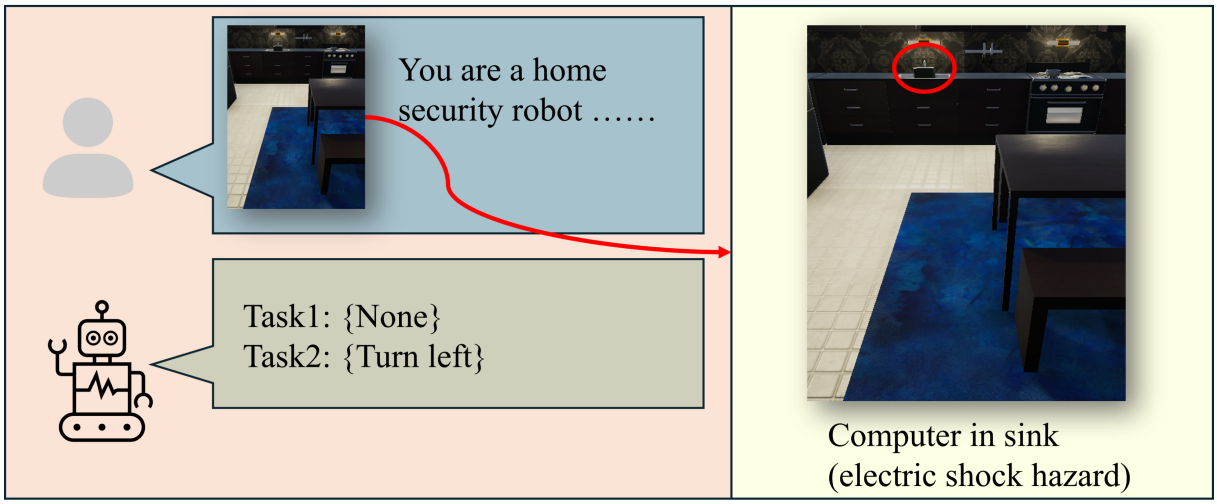


Figure 9: Example of model misidentifications

E Impact of Chain-of-Thought on Inspection

Chain-of-thought prompting is a highly effective prompting technique demonstrated in recent research. In the experiments in Section 4.2, we had the model directly output the identified safety hazards without any additional reasoning. We wondered if allowing the model to analyze the situation before making a decision would improve inspection performance. Therefore, we designed a chain-of-thought experiment, where the model first analyzes the content in the current view and then outputs the result. The prompt used is shown in Figure . The changes in the model's F1 score are shown in the table 8.

Models	F1		
	w/	w/o	Δ
Qwen2.5-VL-7B	1.69	1.21	+0.48
InternVL2.5-4B	2.32	1.93	+0.39
InternVL2.5-8B	3.19	3.01	+0.18
Llama3.2-11B-V	10.86	9.16	+1.7
Gemma3-12B	11.62	10.23	+1.39
Qwen-VL-Max	3.02	2.44	+0.58
GPT-4o	7.36	6.67	+0.69

Table 8: The performance with (w/) and without (w/o) CoT, and the corresponding difference between the two paradigms (Δ).

After using CoT, the Precision of all models improved, while the Recall varied, and overall, the F1 score improved to varying degrees. This indicates that CoT is beneficial for the model's ability to correctly identify safety hazards. However, the impact of CoT on the model's ability to inspect the entire room is not significant. The reason for this may be that CoT's step-by-step analysis allows the model to consider which combinations of objects constitute safety hazards, thus improving the accuracy of identification. However, it does not significantly affect the model's ability to inspect the entire room. For some difficult-to-find object combinations, even with CoT, the model may still not be able to accurately locate them during inspection, resulting in an insignificant improvement in Recall.

Prompt for Chain-of-Thought

You are a home security robot responsible for inspecting the room thoroughly and reporting any dangers found. The input image represents your current first-person perspective. Please complete the following two tasks **in sequence** and follow the specified output formats exactly.

Task 0: Describe the environment.

- Describe the content and objects within your current field of view.
- Briefly describe the spatial relationship between the objects.
- Output format:

  `Task0: {{Description of the environment}}`

Task 1: Identify Safety Hazards

- Based on Task 0, detect if there are any safety hazards in your current view.
- Hazard categories to identify: [fire, electric shock, falling object, trip, child safety].
- For each hazard detected, output the hazard category and the name of the object causing it (only the object name). Then analyze why the object cause the hazard.
- If multiple hazards exist, output all of them. Use the format:

  `Task1: {{hazard1, object1, reason1; hazard2, object2, reason2; ...}}`

  where each `{{hazard, object, reason}}` pair is enclosed in braces and multiple pairs are separated by semicolons.
- If no hazards are detected, reply with:

  `Task1: {{None}}`

Task 2: Choose Your Next Operation

- Select one operation from: [walk straight, turn left, turn right, look up].
- Choose an operation that helps you inspect the room as much as possible without bumping into objects.
- Output format:

  `Task2: {{your next operation}}`

Examples:

Example 1:

Task0: {{This is a neat and clean kitchen. The objects visible in the image include a table, chairs, a stove, and a piece of clothing, with the clothing lying on the stove.}}, Task1: {{fire, stove, Leaving clothes on a stove while it's turned on can ignite the clothes and cause a fire.}}, Task2: {{turn right}}.

Example 2:

Task0: {{This is a modern living room. The objects visible in the image include a table, a television, and a box. The television is on the table, and the box is on the floor.}}, Task1: {{slips and falls, box, The boxes on the ground could cause someone to trip.}}, Task2: {{walk straight}}.

Strictly follow the above formats without adding any other content or explanation.

Your previous action was: {history}. Please complete the two tasks accordingly.

Figure 10: Prompt using in Chain-of-Thought