ProactiveMobile: A Comprehensive Benchmark for Boosting Proactive Intelligence on Mobile Devices

Dezhi Kong[1*] Zhengzhao Feng[1,2*] Qiliang Liang[1,3*] Hao Wang[1] Haofei Sun[1] Changpeng Yang[1]
Yang Li[1] Peng Zhou[1] Shuai Nie[1] Hongzhen Wang[1] Linfeng Zhou[1,4]
Hao Jia[1] Jiaming Xu[1] Runyu Shi[1†] Ying Huang[1]
{kongdezhi1, zhoupeng11, nieshuai, xujiaming1, shirunyu}@xiaomi.com
[1]HyperAI Team, Xiaomi Corporation [2]Zhejiang University [3]Peking University [4]Northeastern University

Abstract

*Multimodal large language models (MLLMs) have made significant progress in mobile agent development, yet their capabilities are predominantly confined to a reactive paradigm, where they merely execute explicit user commands. The emerging paradigm of **proactive intelligence**, where agents autonomously anticipate needs and initiate actions, represents the next frontier for mobile agents. However, its development is critically bottlenecked by the lack of benchmarks that can address real-world complexity and enable objective, executable evaluation. To overcome these challenges, we introduce **ProactiveMobile**, a comprehensive benchmark designed to systematically advance research in this domain. ProactiveMobile formalizes the proactive task as inferring latent user intent across four dimensions of on-device contextual signals and generating an executable function sequence from a comprehensive function pool of 63 APIs. The benchmark features over 3,660 instances of 14 scenarios that embrace real-world complexity through multi-answer annotations. To ensure quality, a team of 30 experts conducts a final audit of the benchmark, verifying factual accuracy, logical consistency, and action feasibility, and correcting non-compliant entries. Extensive experiments demonstrate that our fine-tuned Qwen2.5-VL-7B-Instruct achieves a success rate of 20.82%, outperforming o1 (17.02%) and GPT-5 (11.37%). This result indicates that proactivity is a critical competency widely lacking in current MLLMs, yet it is learnable, emphasizing the importance of the proposed benchmark for proactivity evaluation. Data and code are available at https://github.com/xiaomi-research/proactive-mobile.*

1. Introduction

Fueled by rapid advancements in MLLMs [3, 47, 49], mobile agents have achieved substantial breakthroughs [14, 46] such as interface comprehension [12, 55], conversational interaction [21, 36], and task planning [8, 38].

However, these agents share a fundamental constraint: they are confined to a reactive paradigm, functioning as passive executors of direct user commands [10, 34]. These models place the entire cognitive burden on the user, from need identification to goal articulation [30], thereby relegating the agent to the role of a high-level tool and fundamentally limiting its potential for seamless integration into daily life [24].

The limitations of the reactive paradigm are becoming a critical bottleneck, propelling a fundamental shift towards **proactive intelligence**—the undisputed next frontier for mobile agents. This vision represents not merely an incremental improvement, but a complete re-imagining of the agent's role: rather than being a passive tool, it evolves into a genuinely helpful assistant by autonomously anticipating user needs and initiating actions [26, 43, 44]. The profound implication is a future of human-agent collaboration where cognitive burden is minimized, and interaction feels seamlessly intuitive [4, 30, 43]. Recognizing this transformative potential, pioneering studies have indeed validated the core premise of proactivity [9, 26, 43, 44].

Despite these promising initial steps, the current research landscape for proactive agents remains fragmented and lacks a unified foundation. A core deficiency is that existing benchmarks [26, 44] oversimplify the task: they rely on abstracted contexts and crucially assume a single "correct" action per scenario. This ignores the inherent subjectivity and diversity of user preferences, forcing the complex one-to-many mapping of proactive suggestions into an unrealistic one-to-one paradigm. This flawed premise is exacerbated by the metrics used for evaluation. For instance, *ProactiveAgent*'s [26] binary reward model is too coarse to differentiate partial from complete failures, while *FingerTip-20K* [44] relies on cosine similarity, which captures semantic relevance but ignores functional correctness

*These authors contributed equally.
†Corresponding authors.

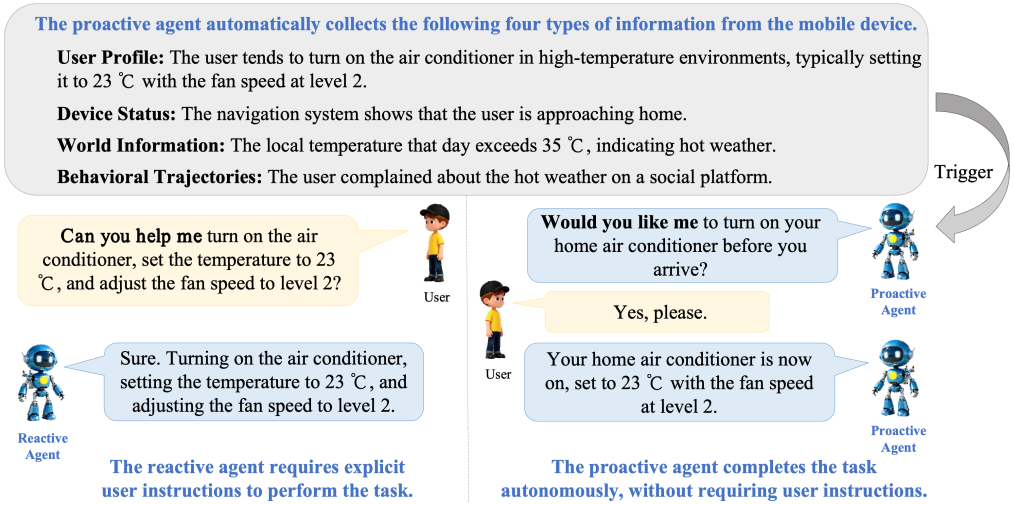


Figure 1. A comparison of proactive and reactive paradigms in mobile agents.

and executability. Beyond definition and evaluation, a third major shortcoming lies in the output format. Both benchmarks rely on generating natural language recommendations, a format that is inherently ambiguous and lacks a direct path to on-device execution, creating a critical gap between suggesting a task and actually performing it. This confluence of issues (an ill-defined task rooted in oversimplification, superficial evaluation, and a non-executable output format) critically bottlenecks the systematic advancement of the field.

To address these critical gaps, we introduce **ProactiveMobile**, a comprehensive benchmark designed to systematically advance research on proactive mobile agents. To mitigate oversimplification, ProactiveMobile formalizes the proactive task by requiring agents to predict actions based on four dimensions of on-device contextual signals: user profile, device status, world information, and behavioral trajectories. Figure 1 clearly illustrates this process and contrasts it with the reactive agent paradigm. To tackle the ignorance of user preferences, ProactiveMobile embraces the one-to-many nature of proactivity: each instance is annotated and manually verified with one to three target actions. The resulting benchmark is substantial, comprising 3,660 instances of 14 scenarios spanning a diverse range of real-world scenarios. Furthermore, to overcome the inherent ambiguity and subjectivity of evaluating natural language suggestions, we introduce a crucial constraint: models must translate their intents into executable actions. We achieve this by constructing a comprehensive function pool of 63 APIs, requiring models to output specific function sequences. This approach transforms the evaluation from a subjective text-matching problem into an objective, structured task.

To establish baselines on our benchmark, we fine-tuned Qwen2.5-VL-7B-Instruct [3] and MiMo-VL-7B-SFT-2508 [35] on the training set. We then evaluated their performance on ProactiveMobile alongside a suite of leading closed-source models, including o1 [17], GPT-5 [29], and Gemini-2.5-Pro [7]. Our fine-tuned Qwen2.5-VL-7B-Instruct achieves a success rate of 20.82% on the exact function sequence matching, significantly outperforming closed-source models, including o1 (17.02%), GPT-5 (11.37%), and Gemini-2.5-Pro (9.62%).

These results offer two critical insights. First, this superior performance supports our hypothesis: proactivity is a specialized capability that requires targeted, domain-specific training, as provided by ProactiveMobile. Even the most powerful general-purpose models fail to master it out-of-the-box. Second, although proactivity is learnable, the performance of trained models still fails to meet the requirements for on-device deployment. This indicates that proactive intelligence is a highly challenging research problem, which in turn underscores the significance of our work and the necessity of the proposed benchmark. Our contributions are as follows:

1. We propose a novel and comprehensive task formalization for proactive mobile agents, grounding the problem in rich, multi-dimensional real-world context.
2. We construct and open-source ProactiveMobile, comprising 3,660 multi-intent instances across 14 scenarios. To facilitate deployment and fine-grained evaluation, all intents are mapped to corresponding function sequences via a predefined function pool of 63 APIs.
3. We provide an in-depth empirical analysis, establishing strong baselines and revealing that proactivity is a specialized capability lacking in current general models, thereby highlighting critical challenges and future research directions. Notably, we will release our model weights to foster progress within the research community.

2. Related Work

2.1. LLM-Based Mobile Interaction

The advent of MLLMs has ushered in a new era of mobile agents capable of understanding natural language instructions and visual UI elements to perform actions autonomously. These LLM-based mobile agents represent a paradigm shift, enabling users to accomplish intricate, multi-step tasks on web, mobile, or desktop applications through simple conversational commands [34, 48].

A core capability of these agents is **GUI understanding**, or GUI grounding, where MLLMs interpret screen layouts by combining visual perception with textual information. To enhance this, specialized models [37, 41, 45] and methods [6, 12, 33, 55] have been developed to better process GUI-specific modalities. The rapid progress in this area is also fueled by the development of specialized datasets, including large-scale annotated datasets [15, 19, 20, 23, 41] and data pipelines [19, 23, 45].

Another critical area is **task planning and execution**. LLMs excel at decomposing high-level natural language commands into a series of executable actions. However, methods based on static prompting often struggle with long-horizon tasks and dynamic environments [34, 40, 42, 51]. Some research explores fine-tuning or reinforcement learning to enhance the reasoning and prediction capabilities of MLLMs in related tasks [13, 27, 28, 39, 53]. The maturation of the field is also marked by comprehensive benchmarks [25, 50, 53, 54], which provide standardized environments for evaluating agent performance on realistic tasks.

2.2. Proactive Agents

The paradigm of intelligent agents is undergoing a significant shift, moving from reactive systems that await explicit user commands to proactive agents that anticipate user needs [9]. By inferring likely intentions and preemptively offering or executing useful actions, these agents can enhance user engagement and task efficiency [32]. Research in proactivity has evolved through several distinct stages.

Initial explorations in this domain have largely focused on proactive conversational agents. Instead of passively responding, these systems actively guide the dialogue by asking clarifying questions, suggesting relevant topics, or steering the conversation towards a productive goal [9, 10, 22]. While foundational, their proactivity is primarily confined to the conversational level.

Building on this, subsequent research has delved deeper into proactive intent inference, where the agent's goal is to predict a user's next action or ultimate goal from their behavior. These approaches can be broadly categorized into two types: those that explicitly prompt the user for clarification to confirm intent [31, 52], and those that implicitly infer intent from contextual cues and behavioral history [18, 44]. This line of work is crucial for understanding user needs before they are articulated.

The most advanced form of proactivity involves agents that not only anticipate needs but also autonomously execute or propose complete tasks. This represents the ultimate goal of delivering value to the user with minimal friction. However, existing work in this advanced stage often faces significant limitations. Some studies are confined to narrow, specific domains like smart home control, limiting their generalizability [5]. Others predict overly simplistic, single-step tasks, often within simulated or artificial scenarios that do not capture the nuances of genuine user interactions [26, 44]. Our work addresses these gaps by introducing a benchmark where the data is deeply grounded in diverse, realistic scenarios, designed to evaluate an agent's ability to recommend complex, multi-step tasks.

3. Benchmark

In this section, we define the task and detail the benchmark construction process. Due to space limitations, we provide comprehensive implementation details in the Appendix, including the prompt templates used for data generation, the design of the annotation platform, annotator training materials, annotation guidelines, quality control procedures, and other technical specifications.

3.1. Task Definition

We define **proactive intelligence** as the task of predicting users' latent intents based on their user profile, device status, world information, and behavioral trajectories. The details of these four categories are as follows:

- **User Profile.** The user's static attributes and dynamic behavioral characteristics encompass basic information, long-term behavioral habits, and personal preferences.
- **Device Status.** Real-time device and immediate environmental states include hardware, battery level, network status, location, and notifications.
- **World Information.** External circumstances, including weather, time of day, and public holidays.
- **Behavioral Trajectories.** A temporal sequence of user-device interactions that reveals evolving intent.

In terms of representation, user profile, device status, and world information are expressed in natural language, while behavioral trajectories are represented either as textual descriptions or sequences of GUI screenshots. To facilitate command execution, all intents are mapped into a unified sequence of executable functions. Consequently, the complete proactive intelligence task can be formalized as:

$$\mathcal{T} = \{(\boldsymbol{I}_k, \boldsymbol{F}_k)\}_{k=1}^{a} = \text{Predict}(\mathbf{U}, \mathbf{D}, \mathbf{W}, \mathbf{B}). \quad (1)$$

Here, $\mathbf{U}$, $\mathbf{D}$, $\mathbf{W}$, and $\mathbf{B}$ represent the user profile, device status, world information, and behavioral trajectories at the

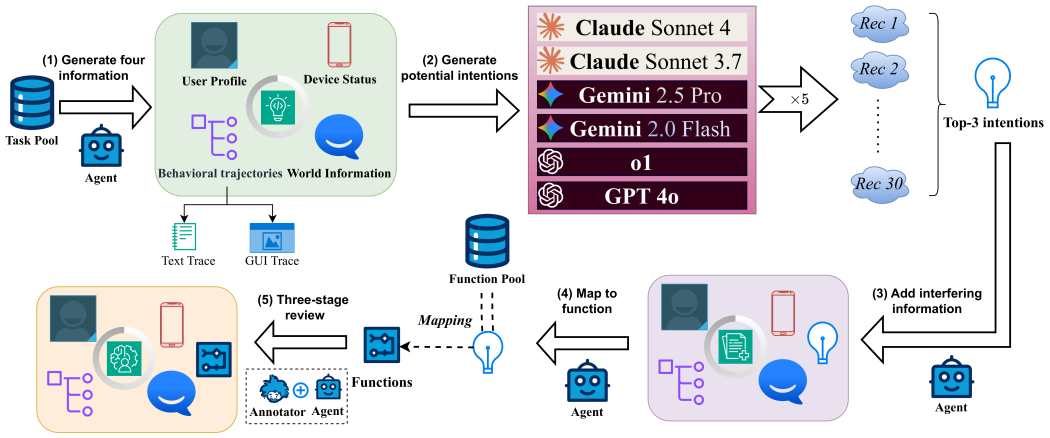


Figure 2. The overview of data generation.

decision moment. For each decision point, there may exist multiple valid intent–function pairs, denoted as the ground-truth set $\mathcal{T}$. The model generates a single predicted pair:

$$(\hat{\boldsymbol{I}}, \hat{\boldsymbol{F}}) = \boldsymbol{M}_\theta(\mathbf{U}, \mathbf{D}, \mathbf{W}, \mathbf{B}),$$
$$\hat{\boldsymbol{F}} = \begin{cases} (f_1, \ldots, f_n), & \hat{\boldsymbol{I}} \neq \varnothing \wedge \hat{\boldsymbol{I}} \Rightarrow \hat{\boldsymbol{F}}, \\ \varnothing, & \hat{\boldsymbol{I}} = \varnothing \vee \hat{\boldsymbol{I}} \not\Rightarrow \hat{\boldsymbol{F}}, \end{cases} \tag{2}$$

where $\hat{\boldsymbol{F}}$ is non-empty only if $\hat{\boldsymbol{I}}$ is actionable and can be mapped to at least one function from the predefined function pool $\mathbb{F}$, i.e., $\hat{\boldsymbol{F}} \subseteq \mathbb{F}$; otherwise, $\hat{\boldsymbol{F}} = \varnothing$.

The prediction is considered correct if the model output matches any ground-truth pair:

$$(\hat{\boldsymbol{I}}, \hat{\boldsymbol{F}}) \in \mathcal{T}. \tag{3}$$

3.2. Dataset Construction

This section is organized into three main parts. First, we elaborate on the acquisition methods for behavioral trajectories. Second, we outline the end-to-end data generation process. Finally, we provide a detailed account of the data auditing mechanism.

3.2.1. Acquisition of Behavioral Trajectories

User behavioral trajectories serve as the foundation for intent prediction. In cases where direct access to user actions is unavailable, screenshots are used as substitutes. We define these two modalities as:

- **Multimodal trajectories:** Sequences of mobile screenshots captured during user interactions, combined with corresponding text commands from both public and self-built datasets, summarized in Table 1.
- **Text trajectories:** Textual logs of user actions are derived from GUI traces. We first categorize and deduplicate text commands, then employ Claude-Sonnet-4 to generate text-based action trajectories via prompt-based expansion.

Dataset	Description	Usage
GUI-Odyssey [25]	A dataset for cross-app mobile GUI navigation	21,669
AITZ [50]	Largest dataset in the Android GUI navigation field	9,413
CAGUI [53]	A real-world Chinese Android GUI benchmark	3,196
MobileAgentBench	Self-collected GUI data from mobile devices	12,481

Table 1. Summary of GUI datasets.

3.2.2. Generation Pipeline

The data generation process involves five key steps, as illustrated in Figure 2.

1. **Generate contextual information.** Based on user behavioral trajectories and relevant commands, we randomly employ Claude-Sonnet-4[2], Gemini-2.5-Pro [7], and GPT-5 [29] to generate three complementary information components—user profile, device status, and world information—thereby constructing a comprehensive contextual information stream. The generated information then undergoes a plausibility check using the o1 [17]. If the content is deemed implausible, it is discarded and subsequently regenerated.
2. **Generate potential intentions.** Leveraging the contextual information, multiple MLLM models simulate users' potential next-step intentions. These intentions represent tasks that agents can recommend proactively, triggered by specific conditions and personalized according to the user profile. To ensure both diversity and quality of generated intentions, we select six state-of-the-art closed-source MLLMs (Claude-Sonnet-4 [2], Claude-Sonnet-3.7 [1], Gemini-2.5-Pro [7], Gemini-2.0-Flash [11], o1 [17], and GPT-4o [16]) with strong multimodal understanding and reasoning capabilities. To unify their outputs, Gemini-2.5-Flash [7] is prompted to semantically cluster the 30 generated candidates and extract the top-3 representative intentions (cluster centroids) ranked by overall support across models.
3. **Add interfering information.** To enhance model robustness, we intentionally inject irrelevant textual noise into the user profile, device states, and environmental information. The injected noise consists of task-irrelevant

yet semantically coherent text generated by Gemini-2.5-Pro [7] through carefully designed prompts. This process preserves overall logical consistency while training the model to focus on salient and task-relevant signals. On average, the amount of injected noise is approximately 5–20 times the volume of task-relevant information.

4. **Map to function.** Convert intentions into function calls. We uniformly convert textual instructions generated by multiple MLLMs into executable function-call sequences. This conversion is performed by Claude-Sonnet-4 [2], which is prompted to select appropriate functions from a predefined function pool to fulfill each recommended task. The resulting sequence may include one or more functions, while a zero-function sequence indicates that no action is required and triggers the no-recommendation logic.
5. **Three-stage review.** A three-stage review mechanism—comprising rule-based checks, agent evaluations, and expert reviews—is adopted to filter and validate generated data, ensuring reliability and accuracy.

3.2.3. Three-Stage Review

To ensure data quality, we implement a comprehensive quality control process spanning three stages: rule-based filtering, agent evaluation, and expert review.

1. **Rule-based filtering.** Automatically removes entries that fail to meet format and consistency requirements.
2. **Agent evaluation.** We employ Gemini-2.5-Pro [7] to assess the internal consistency among textual information, action trajectories, and recommended actions. Using a prompt-based evaluation framework, the model examines textual information for authenticity and naturalness, trajectories for realism and temporal coherence, and recommendations for contextual appropriateness and executability.
3. **Expert review.** Experts verify the remaining entries for factual accuracy, internal logic feasibility, and action feasibility. A team of 30 trained annotators, each with prior experience in human–computer interaction and data annotation, conducts the verification process. All annotators undergo standardized training and trial labeling sessions to align annotation criteria and resolve ambiguities. To ensure data quality, each data point is independently annotated by three annotators. An item is considered valid and accepted for the final dataset only if at least two annotators are in agreement on its label. This extensive cleaning and correction process represents a four-month effort with a total investment of $210,000. Throughout this period, experts collaboratively refine and validate the dataset to ensure its reliability and consistency.

3.3. Function Pool Construction

To facilitate on-device execution and standardized evaluation, we transformed textual instructions into a unified function call format by creating a predefined function pool. Our construction process involved a multi-stage pipeline. First, we manually categorized instructions into 14 scenarios. Then, we employed LLMs to initially generate function sequences and subsequently refine them by merging similar functions and parameters while pruning infrequent ones. Following this automated phase, we defined a formal schema for each function, annotating parameter data types and specifying required arguments. Finally, the entire function pool underwent a rigorous manual verification by five experienced doctoral researchers specializing in AI agents and system design, who cross-checked all definitions to ensure semantic consistency, correctness, and overall coherence.

3.4. Difficulty Definition

To systematically evaluate model performance across different levels of challenge, we establish a three-tier difficulty system. We classify each data item based on the number of correct predictions from a panel of five powerful models: Claude-Sonnet-4 [2], Claude-Sonnet-3.7 [1], GPT-4o [16], Gemini-2.5-Pro [7], and Gemini-2.5-Flash [7]. The difficulty level is defined as follows:

- **Level 1 (Easy)**: Correctly solved by 4–5 out of 5 reference models.
- **Level 2 (Medium)**: Correctly solved by 2–3 out of 5 reference models.
- **Level 3 (Hard)**: Correctly solved by 0–1 out of 5 reference models.

To validate this automatic classification, a group of **five experienced doctoral researchers** independently assessed a stratified sample of data items. The resulting difficulty annotations showed an inter-rater agreement of over **95%** with our model-based difficulty levels, confirming the reliability and consistency of the proposed three-tier system.

Finally, in Figure 3 and Table 2, we illustrate the dataset information for our benchmark.

4. Experiments

This section evaluates our approach by benchmarking against state-of-the-art closed-source MLLMs.

4.1. Setting

Fine-tuned Model. To create a specialized proactive agent, we perform full-parameter supervised fine-tuning (SFT) on Qwen2.5-VL-7B-Instruct [3] and MiMo-VL-7B-SFT-2508 [35]. A core aspect of our methodology is the defined output format: the model is trained to co-generate both a natural language recommendation instruction and the corre-

Split	Data Type	Scenarios	Items	Intents	Images	Ave. Image	Functions	Ave. Functions	L1	L2	L3
Train	Multimodal	12	4,438	8,977	32,418	7.30	9,964	1.11	308	1,376	2,754
	Text	12	4,438	4,438	-	-	8,259	1.86	372	1,208	2,858
Test	Multimodal	14	1,832	3,711	14,341	7.83	4,173	1.24	118	613	1,101
	Text	14	1,828	2,676	-	-	2,266	0.85	259	711	858

Table 2. Statistics of the ProactiveMobile dataset, broken down by Train and Test splits and data modality. The table details the composition of our benchmark, including the number of scenarios, items, intents, functions, and distribution of different difficulties. Notably, the test set includes two additional scenarios (14 vs. 12) not present in the training set, forming our dedicated out-of-distribution evaluation split.

Model	L1						L2					
	Multimodal		Text		All		Multimodal		Text		All	
	SR↑	FTR↓	SR↑	FTR↓	SR↑	FTR↓	SR↑	FTR↓	SR↑	FTR↓	SR↑	FTR↓
GPT-5	12.71	75.00	24.32	25.57	20.69	30.45	7.50	69.44	17.42	30.24	12.81	37.79
GPT-4o	11.86	100.00	13.13	47.06	12.73	51.79	5.71	95.60	10.58	56.29	8.32	62.87
o1	**22.03**	64.29	30.50	**2.73**	27.85	**9.68**	11.91	37.76	24.90	**6.23**	18.88	13.09
Gemini-2.5-Pro	16.10	69.57	13.51	59.09	14.32	60.18	12.07	70.53	6.61	84.74	9.14	81.90
Qwen2.5-VL-7B-Instruct	0.00	66.67	2.70	60.00	1.86	60.71	0.49	68.42	2.53	71.20	1.59	70.55
MiMo-VL-7B-SFT-2508	2.54	76.92	1.93	82.26	2.12	81.33	1.63	68.29	1.27	85.39	1.44	81.29
Qwen2.5-VL-7B**+Proactive**	10.17	**42.31**	**37.07**	6.82	**28.65**	10.57	**17.62**	**23.78**	**32.49**	9.07	**25.60**	**12.61**
MiMo-VL-7B-SFT**+Proactive**	11.86	44.44	18.53	47.85	16.45	47.42	12.40	40.16	16.32	49.34	14.50	47.09

Model	L3						Avg					
	Multimodal		Text		All		Multimodal		Text		All	
	SR↑	FTR↓	SR↑	FTR↓	SR↑	FTR↓	SR↑	FTR↓	SR↑	FTR↓	SR↑	FTR↓
GPT-5	7.90	51.56	9.50	38.26	8.60	44.81	8.08	57.99	14.69	31.41	11.37	39.20
GPT-4o	3.09	94.56	5.77	54.08	4.25	74.58	4.53	95.14	8.69	53.60	6.60	65.32
o1	11.81	23.30	16.08	9.61	13.68	16.64	12.50	29.45	21.55	**6.56**	17.02	14.09
Gemini-2.5-Pro	9.45	66.52	8.51	85.71	9.04	74.94	10.75	67.81	8.48	78.29	9.62	74.98
Qwen2.5-VL-7B-Instruct	1.09	76.29	2.21	54.29	1.58	67.07	0.82	73.76	2.41	64.08	1.61	67.62
MiMo-VL-7B-SFT-2508	1.09	80.44	1.05	73.49	1.07	77.14	1.36	76.71	1.26	81.09	1.31	79.57
Qwen2.5-VL-7B**+Proactive**	**15.08**	**22.63**	**17.37**	**8.75**	**16.08**	**16.08**	**15.61**	**23.91**	**26.04**	8.51	**20.82**	**13.76**
MiMo-VL-7B-SFT**+Proactive**	13.62	43.16	10.37	51.26	12.20	46.49	13.10	42.40	13.84	49.48	13.47	46.91

Table 3. Overall performance comparison of our fine-tuned model (**+Proactive**) against baselines on the ProactiveMobile test set. We report two key metrics: SR↑, where higher is better, and FTR↓, where lower is better. The comparison is broken down by task difficulty (L1-L3) and data modality. For each metric, the best result is in **bold** and the second-best is underlined. All scores are in percentage (%).

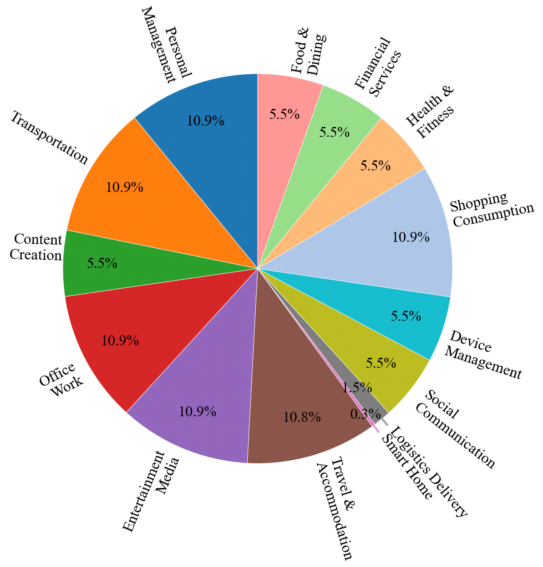


Figure 3. Distribution of the 14 primary user intent categories, demonstrating the benchmark's broad scenario coverage (e.g., Personal Management, Office Work, Food & Dining).

sponding executable function sequence. We utilize 8,876 instances from the training split of ProactiveMobile. Further details regarding data pre-processing, specific hyperparameters, and the hardware environment are provided in Appendix.

Baseline Models. To benchmark against the current state-of-the-art, we evaluate several leading proprietary MLLMs, including GPT-5 [29], GPT-4o [16], Gemini-2.5-Pro [7], o1 [17], unfinetuned Qwen2.5-VL-7B-Instruct [3], and unfinetuned MiMo-VL-7B-SFT-2508 [35]. All baseline models are evaluated in a zero-shot setting. To ensure a fair comparison, the standardized prompt instructs these models to adopt the same output format. We design a standardized prompt that provides each model with the same multi-dimensional context (user profile, device status, etc.) and the list of available functions from our API pool, instructing them to output the appropriate function call sequence.

4.2. Metrics

Evaluating proactive intelligence presents unique challenges, especially given the one-to-many nature of valid ac-

tions in ProactiveMobile, where a single context can map to multiple ground-truth sequences. A naive evaluation metric would either be too brittle (penalizing functionally correct but formally different predictions) or too lenient. To address this, we define two core metrics, Success Rate and False Trigger Rate, whose final values are determined by a dedicated evaluation protocol designed specifically for this one-to-many context, as detailed below.

Model	SR$^{\uparrow}$	FTR$^{\downarrow}$
GPT-5	14.55	36.84
GPT-4o	5.26	47.37
o1	<u>15.63</u>	**13.33**
Gemini-2.5-Pro	9.38	45.46
Qwen2.5-VL-7B-Instruct	4.69	42.86
MiMo-VL-7B-SFT-2508	0.00	100.00
Qwen2.5-VL-7B-Instruct**+Proactive**	**20.31**	<u>16.67</u>
MiMo-VL-7B-SFT-2508**+Proactive**	9.38	57.90

Table 4. Performance on the OOD test set. This set comprises 64 instances from two scenarios (Logistics Delivery and Smart Home) that are entirely absent from the training data.

Training Strategy	Multimodal		Text		All	
	SR$^{\uparrow}$	FTR$^{\downarrow}$	SR$^{\uparrow}$	FTR$^{\downarrow}$	SR$^{\uparrow}$	FTR$^{\downarrow}$
Func.	<u>12.87</u>	95.07	5.69	85.19	<u>9.18</u>	93.16
Rec.+Func.(ours)	**15.61**	<u>23.91</u>	**26.04**	<u>8.51</u>	**20.82**	<u>13.76</u>
Think+Func.	8.81	95.07	4.04	85.04	6.36	93.06
Think+Rec.+Func.	5.53	**2.87**	<u>10.50</u>	**1.53**	8.02	**2.06**

Table 5. Ablation study on the impact of different output formats. We compare our primary **Text Recommendation + Function** strategy against variants that only output the **Function**, or include an additional reasoning (**Think**) step.

4.2.1. Core Metrics

1. Success Rate (SR). This is our primary, binary success metric, designed to measure perfect functional equivalence. A prediction is considered accurate not based on simple string comparison, but on whether it is semantically and functionally identical to a valid ground truth. To make this judgment, we employ a powerful LLM judge (Gemini-2.5-Pro [7]) [1]. An instance receives a final SR score of 1 only if the model's prediction is deemed functionally equivalent to one of the valid ground-truth answers; otherwise, it is 0. Given ProactiveMobile's one-to-many nature, the precise procedure for selecting the "best" ground truth to compare against is critical, and is elaborated in our Best-Match Selection Protocol (Section 4.2.2).

2. False Trigger Rate (FTR). This metric measures the model's reliability in non-trigger scenarios. It quantifies the rate at which the model incorrectly generates an action when the ground truth specifies that no action should be taken. Let $N_{no-action}$ be the total number of instances where the ground-truth set is empty ($G = \emptyset$), and N_{ft} be the number of those instances where the model falsely triggers a non-empty S_{pred}. The FTR is calculated as: $\text{FTR} = \frac{N_{ft}}{N_{no-action}}$

[1]To ensure the validity of judgment, we verified the consistency between the model and human experts, and achieved a consistency of 98%.

4.2.2. Best-Match Selection Protocol

The aforementioned protocol dictates how a model's prediction (S_{pred}) is scored against the set of ground-truth candidates ($G = S_{label,1}, ...$) to yield the final SR score. It is a two-stage process designed to be both rigorous and fair:

Stage 1: Prioritize Perfect Functional Equivalence. We first check if the model's prediction is functionally equivalent (as judged by our LLM referee) to any of the ground-truth sequences. If one or more such "perfect matches" are found, the SR for this instance is immediately set to 1, and the protocol terminates for this instance. One of these perfect matches is randomly selected as the best match (S^*_{label}) for any further analysis.

Stage 2: F1-Score Fallback for Imperfect Predictions. If no perfect match is found in Stage 1, the SR for this instance is definitively 0. However, for consistent and fair analysis, we still need to select a single "closest" ground truth. In this scenario, we identify the ground-truth candidate that maximizes the F1-score (calculated on the sets of function names) when compared with S_{pred}. To compute this score, we treat both the prediction and the ground truth as unordered sets of function names, thus ignoring parameters and sequence order. This allows us to calculate the harmonic mean of precision (the fraction of predicted functions that are correct) and recall (the fraction of correct functions that are predicted). This F1-maximizing sequence is then designated as the best match (S^*_{label}).

Why this protocol? This two-stage design serves a crucial purpose. It establishes perfect functional correctness as the unambiguous gold standard for success, which is directly reflected in our primary SR metric. The F1-fallback mechanism, meanwhile, ensures a robust and consistent process for handling failures, providing a fair basis for comparison and deeper analysis even when the primary success condition is not met.

4.3. Overall Performance

Table 3 presents a comprehensive performance analysis, revealing several critical insights into the current landscape of proactive intelligence.

Fine-tuning on ProactiveMobile consistently unlocks SOTA capabilities. The most striking result is the significant impact of fine-tuning on our benchmark. This effect is consistent across different base models: fine-tuning boosts the Qwen2.5-VL-7B-Instruct from 1.61% to a state-of-the-art 20.82% Success Rate, and similarly elevates the MiMo-VL-7B-SFT-2508 from 1.31% to 13.47%. Our fine-tuned Qwen model establishes a new benchmark, significantly outperforming the top-performing proprietary model, o1

(17.02%). This gap unequivocally demonstrates that proactivity is a specialized, learnable skill requiring domain-specific adaptation, validating ProactiveMobile as an essential training resource.

Multimodal reasoning remains a key bottleneck. The performance disparity across data types reveals a core challenge. For our top-performing model (Qwen2.5-VL-7B + Proactive), the SR on Text tasks (26.04%) is substantially higher than on Multimodal tasks (15.61%). Additionally, we find that for some multimodal tasks, certain scenarios exhibit better performance when visual information is missing, as detailed in the Appendix. This performance delta suggests that grounding abstract intents within noisy, real-world GUI screenshots introduces significant complexity, highlighting robust visual comprehension as a critical area for future advancement in on-device proactive intelligence.

The low absolute scores validate the task's inherent difficulty. Despite the strong relative performance of our fine-tuned model, the absolute SR scores remain modest across the board. The fact that the state-of-the-art sits just around 21% confirms that reliable, functionally correct proactive intelligence is a highly challenging and largely unsolved problem. This finding validates ProactiveMobile not as a benchmark for a saturated task, but as a challenging and indispensable testbed designed to catalyze genuine breakthroughs in the field.

4.4. Generalization to Out-of-Distribution Scenarios

To assess generalization, we evaluate all models on an out-of-distribution (OOD) test set comprising 64 instances from two scenarios—Logistics Delivery and Smart Home—that are entirely absent from the training data. The results in Table 4 reveal a clear performance gap. Our fine-tuned Qwen2.5-VL-7B-Instruct + Proactive model achieves the best performance with 20.31% SR, significantly outperforming other powerful generalists like o1 (15.63%), GPT-5 (14.55%), Gemini-2.5-Pro (9.38%), and GPT-4o (5.26%). This demonstrates that while immense scale offers one path to generalization, our fine-tuning approach effectively imparts a more robust and transferable understanding of proactive logic. It validates that the skills learned on ProactiveMobile are not mere pattern matching, but represent a promising step toward truly generalizable proactive intelligence.

4.5. Ablation Study

To validate our training and output format (Recommendation + Function), we conduct an ablation study comparing it with variants that either omit the recommendation or add an explicit Think step. The results in Table 5 are decisive. Our chosen strategy achieves the highest SR (20.82%), indicating that compelling the model to articulate a user-facing intent acts as an effective reasoning scaffold. Critically, formats trained without this textual recommendation (Function only and Think + Function) exhibit extremely poor safety behavior, with False Trigger Rate (FTR) rates near 100%. This demonstrates that generating the intent is indispensable for teaching the model the crucial skill of when not to act.

The study also reveals a crucial trade-off between SR and safety. While adding a Think step (Think + Recommendation + Function) significantly lowers SR, it drastically enhances safety, slashing the FTR rate to a 2.06%. This highlights the Think step as a promising direction for building maximally safe agents. Nevertheless, our primary Recommendation + Function approach offers the best-performing balance between SR and reliability, thus validating our core design choice. Further ablation studies, including an analysis of the impact of different contextual dimensions, are detailed in the Appendix.

5. Conclusion

In this work, we address the critical bottleneck hindering the transition of mobile agents from a reactive to a proactive paradigm: the lack of an executable, objective, and realistic benchmark. We introduce ProactiveMobile, a comprehensive benchmark that formalizes the proactive task around a four-dimensional context model, incorporates multi-answer annotations, and uniquely mandates an executable function-call sequence output. Our extensive experiments validate that proactivity is a specialized, learnable capability. This is consistently demonstrated as fine-tuning on our benchmark boosts the performance of different models, with our top-performing model achieving a 20.82% success rate—establishing a new state-of-the-art that surpasses even leading proprietary models like o1 (17.02%). This demonstrates the efficacy of ProactiveMobile as an essential tool for targeted training and highlights the significant gap in current models' out-of-the-box abilities.

While our work establishes a new SOTA, the modest absolute success rates underscore that proactive intelligence is a profoundly challenging research problem, opening up several promising future directions. Key priorities include enhancing models' multimodal reasoning to close the significant performance gap between text and multimodal tasks, and exploring advanced training methodologies like reinforcement learning for more robust decision-making. Furthermore, our ablation study on output formats reveals a rich trade-off between success rate and safety, warranting deeper investigation into creating agents that are not only effective but also trustworthy. By providing a foundational and challenging testbed, ProactiveMobile aims to catalyze these future innovations, steering the community toward the development of truly intelligent, anticipatory agents.

References

[1] Anthropic. Claude 3.7 Sonnet and Claude Code. https://www.anthropic.com/news/claude-3-7-sonnet, 2025. Accessed: 2025-11-13. 4, 5

[2] Anthropic. Introducing Claude 4. https://www.anthropic.com/news/claude-4, 2025. Accessed: 2025-11-13. 4, 5

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 2, 5, 6

[4] Florian Brachten, Felix Brünker, Nicholas RJ Frick, Björn Ross, and Stefan Stieglitz. On the ability of virtual agents to decrease cognitive load: an experimental study. *Information Systems and e-Business Management*, 18(2):187–207, 2020. 1

[5] Zhihao Cao, Zidong Wang, Siwen Xie, Anji Liu, and Lifeng Fan. Smart help: Strategic opponent modeling for proactive and adaptive robot assistance in households. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18091–18101, 2024. 3

[6] Jikai Chen, Long Chen, Dong Wang, Leilei Gan, Chenyi Zhuang, and Jinjie Gu. V2p: From background suppression to center peaking for robust gui grounding task, 2025. 3

[7] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 2, 4, 5, 6, 7

[8] Shihan Deng, Weikai Xu, Hongda Sun, Wei Liu, Tao Tan, Liujianfeng Liujianfeng, Ang Li, Jian Luan, Bin Wang, Rui Yan, and Shuo Shang. Mobile-bench: An evaluation benchmark for LLM-based mobile agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8813–8831, Bangkok, Thailand, 2024. Association for Computational Linguistics. 1

[9] Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. A survey on proactive dialogue systems: Problems, methods, and prospects, 2023. 1, 3

[10] Yang Deng, Lizi Liao, Wenqiang Lei, Grace Hui Yang, Wai Lam, and Tat-Seng Chua. Proactive conversational ai: A comprehensive survey of advancements and opportunities. *ACM Transactions on Information Systems*, 43(3):1–45, 2025. 1, 3

[11] Google. Gemini 2.0: Flash, Flash-Lite and Pro. https://developers.googleblog.com/en/gemini-2-family-expands/, 2025. Accessed: 2025-11-13. 4

[12] Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for gui agents, 2025. 1, 3

[13] Zhangxuan Gu, Zhengwen Zeng, Zhenyu Xu, Xingran Zhou, Shuheng Shen, Yunfei Liu, Beitong Zhou, Changhua Meng, Tianyu Xia, Weizhi Chen, Yue Wen, Jingya Dou, Fei Tang, Jinzhen Lin, Yulin Liu, Zhenlin Guo, Yichen Gong, Heng Jia, Changlong Gao, Yuan Guo, Yong Deng, Zhenyu Guo, Liang Chen, and Weiqiang Wang. Ui-venus technical report: Building high-performance ui agents with rft, 2025. 3

[14] Xueyu Hu, Tao Xiong, Biao Yi, Zishu Wei, Ruixuan Xiao, Yurun Chen, Jiasheng Ye, Meiling Tao, Xiangxin Zhou, Ziyu Zhao, et al. Os agents: A survey on mllm-based agents for computer, phone and browser use. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7436–7465, 2025. 1

[15] Zheng Hui, Yinheng Li, Dan Zhao, Colby Banbury, Tianyi Chen, and Kazuhito Koishida. Winspot: Gui grounding benchmark with multimodal large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1086–1096, 2025. 3

[16] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 4, 5, 6

[17] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 2, 4, 6

[18] Jaekyeom Kim, Dong-Ki Kim, Lajanugen Logeswaran, Sungryull Sohn, and Honglak Lee. Auto-intent: Automated intent discovery and self-exploration for large language model web agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16531–16541, 2024. 3

[19] Hongxin Li, Jingfan Chen, Jingran Su, Yuntao Chen, Qing Li, and Zhaoxiang Zhang. Autogui: Scaling gui grounding with automatic functionality annotations from llms. *arXiv preprint arXiv:2502.01977*, 2025. 3

[20] Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. *arXiv preprint arXiv:2504.07981*, 2025. 3

[21] Yanda Li, Chi Zhang, Wenjia Jiang, Wanqi Yang, Bin Fu, Pei Cheng, Xin Chen, Ling Chen, and Yunchao Wei. Appagent v2: Advanced agent for flexible mobile interactions. *arXiv preprint arXiv:2408.11824*, 2024. 1

[22] Lizi Liao, Grace Hui Yang, and Chirag Shah. Proactive conversational agents in the post-chatgpt world. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 3452–3455, 2023. 3

[23] Xinyi Liu, Xiaoyi Zhang, Ziyun Zhang, and Yan Lu. Ui-e2i-synth: Advancing gui grounding with large-scale instruction synthesis, 2025. 3

[24] Xingyu Bruce Liu, Shitao Fang, Weiyan Shi, Chien-Sheng Wu, Takeo Igarashi, and Xiang'Anthony' Chen. Proactive conversational agents with inner thoughts. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2025. 1

[25] Quanfeng Lu, Wenqi Shao, Zitao Liu, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, Yu

Qiao, and Ping Luo. Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. *arXiv preprint arXiv:2406.08451*, 2024. 3, 4

[26] Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, et al. Proactive agent: Shifting llm agents from reactive responses to active assistance. *arXiv preprint arXiv:2410.12361*, 2024. 1, 3

[27] Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. *arXiv preprint arXiv:2503.21620*, 2025. 3

[28] Run Luo, Lu Wang, Wanwei He, Longze Chen, Jiaming Li, and Xiaobo Xia. Gui-r1 : A generalist r1-style vision-language action model for gui agents, 2025. 3

[29] OpenAI. GPT-5 System Card. Technical report, OpenAI, 2025. Accessed: 2025-08-10. 2, 4, 6

[30] Yingzhe Peng, Xiaoting Qin, Zhiyang Zhang, Jue Zhang, Qingwei Lin, Xu Yang, Dongmei Zhang, Saravan Rajmohan, and Qi Zhang. Navigating the unknown: A chat-based collaborative interface for personalized exploratory tasks. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pages 1048–1063, 2025. 1

[31] Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, et al. Tell me more! towards implicit user intention understanding of language model driven agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1113, 2024. 3

[32] Roderick Tabalba, Christopher J. Lee, Giorgio Tran, Nurit Kirshenbaum, and Jason Leigh. Articulatepro: A comparative study on a proactive and non-proactive assistant in a climate data exploration task, 2024. 3

[33] Fei Tang, Zhangxuan Gu, Zhengxi Lu, Xuyang Liu, Shuheng Shen, Changhua Meng, Wen Wang, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. Gui-g^2: Gaussian reward modeling for gui grounding, 2025. 3

[34] Fei Tang, Haolei Xu, Hang Zhang, Siqi Chen, Xingyu Wu, Yongliang Shen, Wenqi Zhang, Guiyang Hou, Zeqi Tan, Yuchen Yan, Kaitao Song, Jian Shao, Weiming Lu, Jun Xiao, and Yueting Zhuang. A survey on (m)llm-based gui agents, 2025. 1, 3

[35] Core Team, Zihao Yue, Zhenru Lin, Yifan Song, Weikun Wang, Shuhuai Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, Lei Li, Kainan Bao, Hao Tian, Hailin Zhang, Gang Wang, Dawei Zhu, Cici, Chenhong He, Bowen Ye, Bowen Shen, Zihan Zhang, Zihan Jiang, Zhixian Zheng, Zhichao Song, Zhenbo Luo, Yue Yu, Yudong Wang, Yuanyuan Tian, Yu Tu, Yihan Yan, Yi Huang, Xu Wang, Xinzhe Xu, Xingchen Song, Xing Zhang, Xing Yong, Xin Zhang, Xiangwei Deng, Wenyu Yang, Wenhan Ma, Weiwei Lv, Weiji Zhuang, Wei Liu, Sirui Deng, Shuo Liu, Shimao Chen, Shihua Yu, Shaohui Liu, Shande Wang, Rui Ma, Qiantong Wang, Peng Wang, Nuo Chen, Menghang Zhu, Kangyang Zhou, Kang Zhou, Kai Fang, Jun Shi, Jinhao Dong, Jiebao Xiao, Jiaming Xu, Huaqiu Liu, Hongshen Xu, Heng Qu, Haochen Zhao, Hanglong Lv, Guoan Wang, Duo Zhang, Dong Zhang, Di Zhang, Chong Ma, Chang Liu, Can Cai, and Bingquan Xia. Mimo-vl technical report, 2025. 2, 5, 6

[36] Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *Advances in Neural Information Processing Systems*, 37:2686–2710, 2024. 1

[37] Ziwei Wang, Weizhi Chen, Leyang Yang, Sheng Zhou, Shengchu Zhao, Hanbei Zhan, Jiongchao Jin, Liangcheng Li, Zirui Shao, and Jiajun Bu. Mp-gui: Modality perception with mllms for gui understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29711–29721, 2025. 3

[38] Zhenhailong Wang, Haiyang Xu, Junyang Wang, Xi Zhang, Ming Yan, Ji Zhang, Fei Huang, and Heng Ji. Mobile-agent-e: Self-evolving mobile assistant for complex tasks. *arXiv preprint arXiv:2501.11733*, 2025. 1

[39] Jinyang Wu, Mingkuan Feng, Shuai Zhang, Feihu Che, Zengqi Wen, Chonghua Liao, and Jianhua Tao. Beyond examples: High-level automated reasoning paradigm in in-context learning via mcts. *arXiv preprint arXiv:2411.18478*, 2024. 3

[40] Jinyang Wu, Guocheng Zhai, Ruihan Jin, Jiahao Yuan, Yuhao Shen, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. Atlas: Orchestrating heterogeneous models and tools for multi-domain complex reasoning. *arXiv preprint arXiv:2601.03872*, 2026. 3

[41] Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. Os-atlas: A foundation action model for generalist gui agents, 2024. 3

[42] Yuquan Xie, Zaijing Li, Rui Shao, Gongwei Chen, Kaiwen Zhou, Yinchuan Li, Dongmei Jiang, and Liqiang Nie. Mirage-1: Augmenting and updating gui agent with hierarchical multimodal skills, 2025. 3

[43] Bufang Yang, Lilin Xu, Liekang Zeng, Kaiwei Liu, Siyang Jiang, Wenrui Lu, Hongkai Chen, Xiaofan Jiang, Guoliang Xing, and Zhenyu Yan. Contextagent: Context-aware proactive llm agents with open-world sensory perceptions, 2025. 1

[44] Qinglong Yang, Haoming Li, Haotian Zhao, Xiaokai Yan, Jingtao Ding, Fengli Xu, and Yong Li. Fingertip 20k: A benchmark for proactive and personalized mobile llm agents. *arXiv preprint arXiv:2507.21071*, 2025. 1, 3

[45] Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. Aria-ui: Visual grounding for gui instructions. *arXiv preprint arXiv:2412.16256*, 2024. 3

[46] Huanjin Yao, Ruifei Zhang, Jiaxing Huang, Jingyi Zhang, Yibo Wang, Bo Fang, Ruolin Zhu, Yongcheng Jing, Shunyu Liu, Guanbin Li, and Dacheng Tao. A survey on agentic multimodal large language models, 2025. 1

[47] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal

large language models. *National Science Review*, 11(12): nwae403, 2024. 1

[48] Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Liqun Li, Si Qin, Yu Kang, Minghua Ma, Guyue Liu, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Large language model-brained gui agents: A survey, 2025. 3

[49] Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*, 2024. 1

[50] Jiwen Zhang, Jihao Wu, Teng Yihua, Minghui Liao, Nuo Xu, Xiao Xiao, Zhongyu Wei, and Duyu Tang. Android in the zoo: Chain-of-action-thought for gui agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12016–12031, 2024. 3, 4

[51] Shaoqing Zhang, Zhuosheng Zhang, Kehai Chen, Xinbei Ma, Muyun Yang, Tiejun Zhao, and Min Zhang. Dynamic planning for llm-based graphical user interface automation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1304–1320, 2024. 3

[52] Xuan Zhang, Yang Deng, Zifeng Ren, See Kiong Ng, and Tat-Seng Chua. Ask-before-plan: Proactive language agents for real-world planning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10836–10863, 2024. 3

[53] Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, Yesai Wu, Han Si, Xin Cong, Haotian Chen, Yankai Lin, Jie Xie, Wei Zhou, Wang Xu, Yuanheng Zhang, Zhou Su, Zhongwu Zhai, Xiaoming Liu, Yudong Mei, Jianming Xu, Hongyan Tian, Chongyi Wang, Chi Chen, Yuan Yao, Zhiyuan Liu, and Maosong Sun. AgentCPM-GUI: Building mobile-use agents with reinforcement fine-tuning. *arXiv preprint arXiv:2506.01391*, 2025. 3, 4

[54] Henry Hengyuan Zhao, Kaiming Yang, Wendi Yu, Difei Gao, and Mike Zheng Shou. Worldgui: An interactive benchmark for desktop gui automation from any starting point. *arXiv preprint arXiv:2502.08047*, 2025. 3

[55] Yuqi Zhou, Sunhao Dai, Shuai Wang, Kaiwen Zhou, Qinglin Jia, and Jun Xu. Gui-g1: Understanding r1-zero-like training for visual grounding in gui agents. *arXiv preprint arXiv:2505.15810*, 2025. 1, 3

A. Implementation Details

A.1. Training Parameter Configuration

This section describes the detailed settings for reproducing the fine-tuning experiments.

Data Pre-processing. To mitigate potential noise in the training data, we applied the following two filtering criteria:

- Samples with function call sequences longer than 3 are excluded.
- The visual trajectory for each instance is truncated to a maximum of 10 frames.

Training Hyperparameters. Full-parameter supervised fine-tuning is configured with the following settings:

- **Optimizer:** AdamW
- **Initial Learning Rate:** 1e-5
- **LR Schedule:** Cosine Annealing
- **Warmup Ratio:** 0.1
- **Batch Size:** 64
- **Epochs:** 4

Inference Hyperparameters. To ensure a fair comparison, all hyperparameters are held constant during inference across all models.

- **Temperature:** 1
- **Top P:** 0.7

A.2. Training Format

Hardware and Training Time. The training is conducted on a 4-node cluster, utilizing a total of 32 NVIDIA H20 GPUs. The entire training process for 4 epochs took approximately 8 hours to complete.

B. Data

B.1. Data statistics

We present a more intuitive visualization of the data distribution of the proposed benchmark in Figure 4.

B.2. Function

We have constructed a comprehensive function pool comprising 63 APIs, which will be fully open-sourced. To better illustrate its structure, Figure 5 provides an example of a commonly used function.

B.3. Data Annotation Platform

All data annotations are performed by domain experts on the Argilla platform. A screenshot of the Argilla platform is presented in Figure 6.

C. Expanded Results with Granular Metrics

This section provides a more granular analysis of model performance, moving beyond the primary, all-or-nothing ACC metric discussed in the main paper. Here, we report on set-based metrics that capture partial correctness and offer deeper insights into model behaviors. The results are presented in Table 6.

C.1. Granular Metric Definitions

The following metrics are used to generate the results in Table 6. Let S_{pred} be the predicted function call sequence and S_{label} be the best-match ground-truth sequence determined by our protocol.

Type-Acc (Function Name Sequence Accuracy). This metric focuses solely on the perfect match of the function name sequence.

$$\text{Type-Acc}(S_{pred}, S_{label}) = \mathbb{I}\left(\langle c_{pred,i}.\text{name}\rangle = \langle c_{label,j}.\text{name}\rangle\right) \tag{4}$$

F1-Score, Precision (P) & Recall (R). To provide a more forgiving, set-based evaluation, we treat the prediction and ground truth as unordered sets of function names. Let $P_{set} = \text{set}(S_{pred})$ and $L_{set} = \text{set}(S_{label})$.

Precision (P) measures the proportion of predicted functions that are relevant (i.e., how many of the model's suggestions are correct). $\text{P} = \frac{|P_{set} \cap L_{set}|}{|P_{set}|}$

Recall (R) measures the proportion of ground-truth functions that were successfully predicted (i.e., how many of the required actions the model found). $\text{R} = \frac{|P_{set} \cap L_{set}|}{|L_{set}|}$

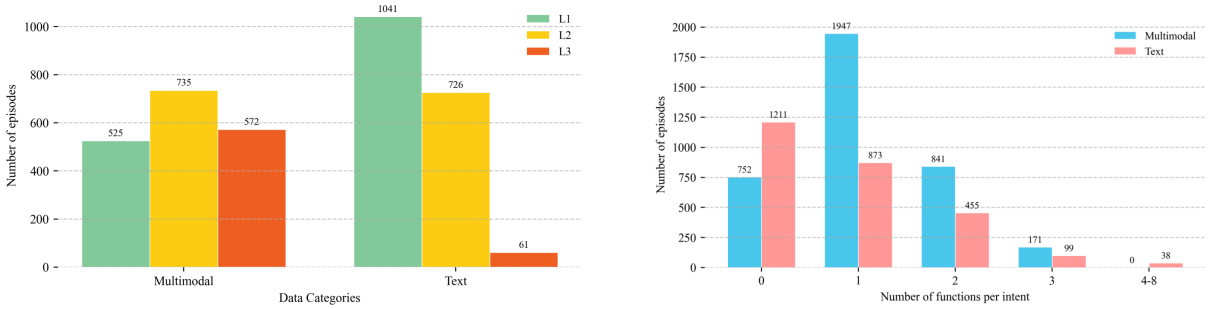
F1-Score (F1) is the harmonic mean of Precision and Recall, providing a single balanced score for partial correctness. $\text{F1} = 2 \times \frac{\text{P} \times \text{R}}{\text{P} + \text{R}}$

C.2. Analysis of Detailed Results

The detailed results in Table 6 reinforce the primary conclusions from the main paper while offering additional, nuanced insights:

1. Fine-tuning Consistently Unlocks Proactive Skills. The most prominent trend is the dramatic performance uplift from fine-tuning on ProactiveMobile. This holds true for both model families. The Qwen2.5-VL-7B-Instruct model's average F1 score catapults from 4.50% to 50.88% after being trained, becoming Qwen2.5-VL-7B-Instruct + Proactive. Similarly, the MiMo-VL-7B-SFT-2508 + Proactive model significantly outperforms its base version. This strongly corroborates our main finding that proactivity is a learnable skill, and our benchmark is an effective resource for teaching it.

2. A Clear Hierarchy Among Models Emerges. The granular metrics reveal a clear performance hierarchy. The o1 model consistently establishes itself as the top-performing baseline across nearly all metrics and settings, demonstrating its powerful general-purpose reasoning. Our fine-tuned Qwen2.5-VL-7B-Instruct + Proactive model emerges as a highly competitive contender, often securing the second-best performance, particularly in Text-based scenarios where its F1 score (60.84%) approaches



(a) Distribution of difficulty across data categories.

(b) Distribution of intent functions.

Figure 4. Detailed dataset statistics and distribution.


```
"book_transport": {
    "name": "book_transport",
    "description": "A unified transportation reservation function that facilitates booking of multimodal transport services, including but not limited to aviation,
          railway, taxi, and ride-hailing services. Users may specify origin, destination, travel temporal parameters, and passenger demographics, with optional
         platform selection.",
    "similar": [
    "search_transport",
    "plan_route_and_navigate"
    ],
    "parameters": {
    "transport_type": {
        "description": "Mandatory parameter specifying the modality of transportation. Critical for distinguishing between long-haul aviation, intercity rail,
              urban taxi services, and short-distance mobility options. Valid enumerations include: 'flight', 'train', 'taxi' .",
        "type": "string",
        "must_fill": "required",
        "value": [
        "flight",
        "train",
        "high_speed_rail",
        "taxi",
        "ride_sharing",
        "bus",
        "subway",
        "bike",
        "rental_car",
        "ferry"
        ]
    },
    "start_location": {
        "description": "Mandatory parameter defining the geographical origin of the itinerary. Accepts structured addresses, municipal designations, IATA airport
              codes (e.g., 'PEK'), railway station nomenclature (e.g., 'Beijing South Railway Station'), or prominent landmarks. For example: 'No. 1 Zhongguancun
              Avenue, Haidian District, Beijing' or 'Shanghai Hongqiao International Airport'.",
        "type": "string",
        "must_fill": "required",
        "value": "non-enumerable"
    },
    "end_location": {
        "description": "Mandatory parameter indicating the geographical terminus of the journey. Format specifications mirror those of the origin parameter,
              accommodating precise addresses, urban areas, transportation hubs, or notable landmarks. For example: 'Shenzhen Nanshan Science and Technology Park'
              or 'Guangzhou Baiyun Airport'.",
        "type": "string",
        "must_fill": "required",
        "value": "non-enumerable"
    },
    "departure_time": {
        "description": "Optional temporal parameter for journey scheduling. Accepts standardized datetime formatting ('YYYY-MM-DD HH:MM:SS') or relative temporal
              expressions (e.g., '30 minutes hence', '09:00 tomorrow morning').",
        "type": "string",
        "must_fill": "optional",
        "value": "non-enumerable"
    },
    "passenger_num": {
        "description": "Optional parameter quantifying traveler count. Defaults to unitary occupancy if unspecified.",
        "type": "int",
        "must_fill": "optional",
        "value": "non-enumerable"
    },
    "platform_info": {
        "description": "Optional parameter designating the reservation platform or application. When specified, directs the booking to the designated service;
              otherwise, initiates cross-platform comparative analysis or employs default service providers. Supports mainstream mobility service platforms.For
              example: 'DiDi', 'Ctrip', '12306'.",
        "type": "string",
        "must_fill": "optional",
        "value": [
        "Didi",
        "Uber",
        "Amap",
        "CaoCao",
        "T3",
        "Ctrip",
        "Qunar",
        "Fliggy",
        "Tongcheng",
        "12306",
        "TravelSky"
        ]
    }
    }
}
```


Figure 5. Exemplary implementation of the book_transport API.

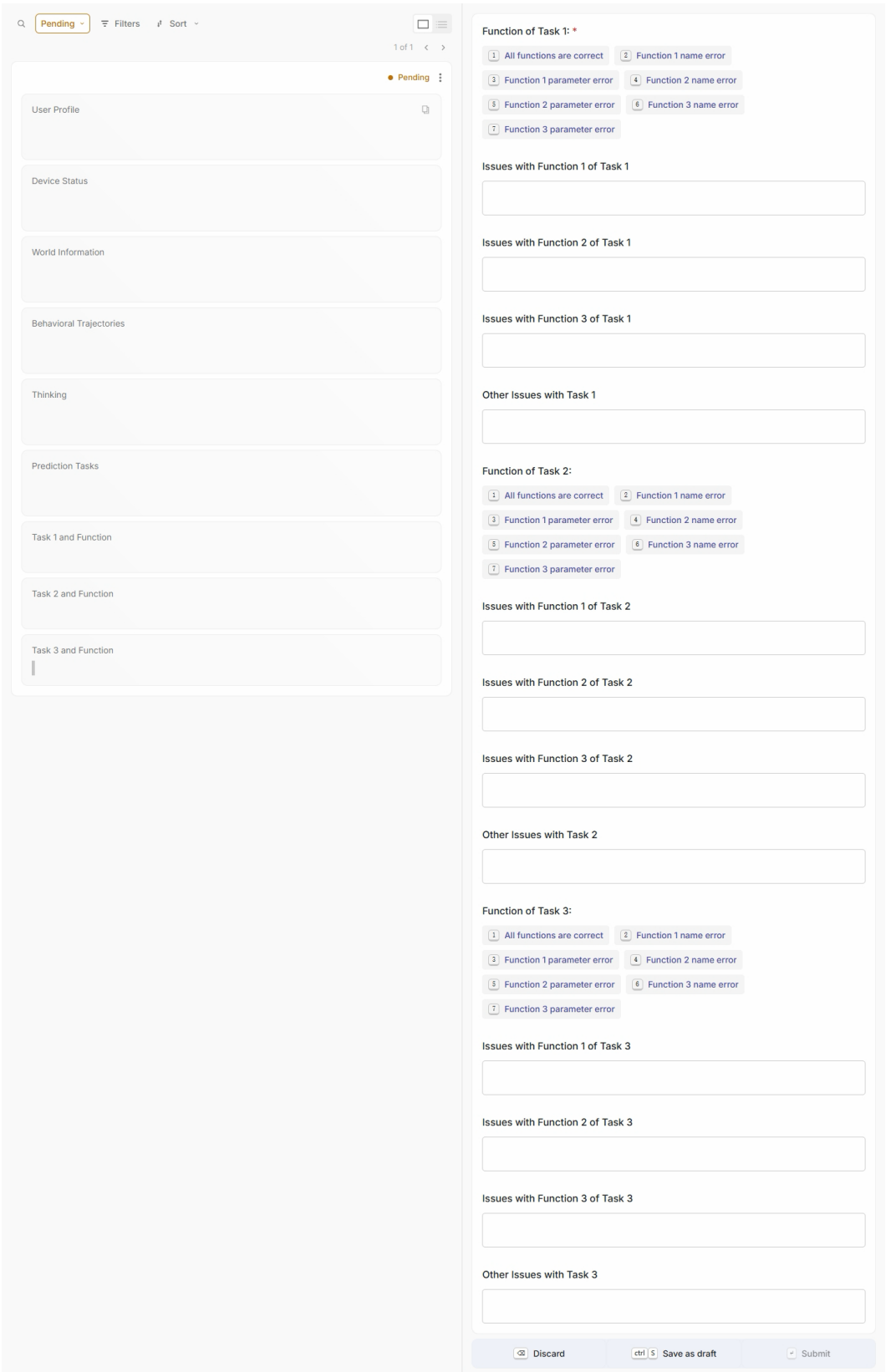


Figure 6. The data annotation platform.

that of o1 (72.09%). Other models like GPT-5, GPT-4o, and Gemini-2.5-Pro show varied performance, but generally trail behind the top two.

D. Ablation Study on Four-Dimension Information

This section details an ablation study designed to quantify the contribution of each of the four contextual dimensions proposed in our benchmark: User Profile, Device Status, World Information, and Behavioral Trajectories.

D.1. Experimental Setup

To isolate the impact of each dimension, we systematically removed one dimension at a time from the input provided to the models and re-ran the evaluation. The "All Info" condition, where models have access to the complete four-dimensional context, serves as the baseline for comparison. The results of this study are presented in Table 7. It is worth noting that in the w/o Trajectories condition, the distinction between Multimodal and Text data disappears, as the visual trajectory is the sole differentiating element.

D.2. Analysis of Results

The ProactiveMobile Benchmark demonstrates strong robustness. Our experiments indicate that omitting any single input dimension leads to only minor fluctuations in performance. For instance, the SR of our fine-tuned Qwen2.5-VL-7B-Instruct + Proactive model varies by merely 1.4%, while its FTR fluctuates by around 4%. For the o1 model, the corresponding variations are approximately 1.5% and 6.3%, respectively.

These findings are consistent with our design motivation. In real-world environments, contextual signals often overlap or contain redundant information. Incorporating such redundancy during training enables models to learn to extract the most relevant signals from complex inputs. As a result, the absence of any single information source does not substantially degrade performance and, in some cases, can even lead to slight improvements.

E. Case Study

The complete task workflow is illustrated through a concrete case provided in Table 8. Additionally, we conducted a comparative evaluation of our model against strong counterparts, including GPT-5 and o1. Notably, for the purpose of clear demonstration, both case studies present only a single intent.

F. Prompts for the LLM agents

A suite of tailored prompts is utilized, encompassing the following categories: Prompt for Generating Contextual Information, Prompt for Generating Potential Intentions, Prompt for Adding Interfering Information, Prompt for Mapping to Function, Prompt for Three-stage Review, and Prompt for Training/Inference. The specific prompts are compiled in the prompt box below.

Difficulty	Model	Multimodal				Text				All			
		Type-Acc$^{\uparrow}$	F1$^{\uparrow}$	P$^{\uparrow}$	R$^{\uparrow}$	Type-Acc$^{\uparrow}$	F1$^{\uparrow}$	P$^{\uparrow}$	R$^{\uparrow}$	Type-Acc$^{\uparrow}$	F1$^{\uparrow}$	P$^{\uparrow}$	R$^{\uparrow}$
L1	GPT-5	33.05	58.50	56.64	**65.25**	67.57	70.91	71.04	71.36	56.76	67.03	66.53	69.45
	GPT-4o	30.51	49.70	46.85	57.91	45.17	48.82	48.46	49.81	40.58	49.09	47.95	52.34
	o1	51.70	58.48	59.75	58.05	**91.12**	**92.41**	**92.47**	**92.47**	**78.78**	**81.79**	**82.23**	**81.70**
	Gemini-2.5-Pro	**55.08**	**63.70**	**65.25**	63.14	40.93	43.50	44.02	43.44	45.36	49.82	50.66	49.60
	Qwen2.5-VL-7B-Instruct	3.39	3.39	3.39	3.39	7.72	7.72	7.72	7.72	6.37	6.37	6.37	6.37
	MiMo-VL-7B-SFT-2508	5.93	8.45	8.33	9.32	4.63	4.89	4.83	5.02	5.04	6.00	5.92	6.37
	Qwen2.5-VL-7B**+Proactive**	32.20	38.84	40.25	38.56	82.24	83.45	83.46	83.59	66.58	69.49	69.94	69.50
	MiMo-VL-7B-SFT**+Proactive**	30.51	41.02	41.95	41.81	42.47	44.75	44.72	45.17	38.73	43.58	43.86	44.12
L2	GPT-5	26.75	50.61	48.29	**58.05**	54.67	63.50	63.74	64.95	41.70	57.51	56.56	61.74
	GPT-4o	15.33	38.78	35.89	48.04	32.58	38.88	38.65	40.74	24.58	38.83	37.37	44.12
	o1	**42.58**	51.78	52.96	51.99	**76.93**	**80.96**	**82.49**	**80.24**	**61.03**	**67.45**	**68.82**	**67.16**
	Gemini-2.5-Pro	42.09	**56.17**	**57.75**	57.04	24.61	31.59	33.68	30.95	32.70	42.97	44.83	43.03
	Qwen2.5-VL-7B-Instruct	3.43	4.04	4.05	4.13	5.20	5.27	5.25	5.34	4.38	4.70	4.70	4.78
	MiMo-VL-7B-SFT-2508	5.55	8.16	8.10	8.97	3.80	3.98	3.97	4.10	4.61	5.92	5.88	6.36
	Qwen2.5-VL-7B**+Proactive**	34.75	43.39	44.13	44.02	69.06	72.34	72.60	72.68	53.17	58.93	59.42	59.41
	MiMo-VL-7B-SFT**+Proactive**	30.02	43.21	44.56	44.32	34.32	41.23	41.40	42.35	32.33	42.15	42.86	43.26
L3	GPT-5	23.98	44.66	44.26	**49.09**	29.19	46.94	48.42	48.73	26.25	45.66	46.07	48.93
	GPT-4o	12.62	31.84	30.55	37.56	22.59	37.06	36.95	40.38	16.97	34.11	33.34	38.79
	o1	**40.51**	**49.53**	**51.74**	48.99	**50.35**	**58.61**	**61.11**	**57.67**	**44.82**	**53.50**	**55.84**	**52.79**
	Gemini-2.5-Pro	31.61	44.48	47.02	44.19	27.16	41.98	45.38	41.38	29.66	43.38	46.30	42.96
	Qwen2.5-VL-7B-Instruct	2.72	3.80	3.93	3.84	4.08	4.26	4.23	4.31	3.32	4.00	4.06	4.04
	MiMo-VL-7B-SFT-2508	3.09	5.01	5.03	5.69	3.50	4.91	5.11	5.13	3.27	4.97	5.07	5.44
	Qwen2.5-VL-7B**+Proactive**	33.24	39.79	40.77	40.21	39.04	44.50	45.28	44.85	35.78	41.85	42.74	42.24
	MiMo-VL-7B-SFT**+Proactive**	28.70	39.42	40.75	40.05	22.14	33.55	34.65	34.66	25.83	36.85	38.08	37.69
Avg	GPT-5	25.49	47.54	46.40	**53.13**	44.55	56.79	57.59	58.25	34.99	52.15	51.98	55.68
	GPT-4o	14.68	35.31	33.38	42.38	29.70	39.44	39.25	41.86	22.16	37.37	36.30	42.12
	o1	**41.92**	**50.86**	**52.67**	50.57	**66.47**	**72.09**	**73.87**	**71.38**	**54.18**	**61.46**	**63.26**	**60.97**
	Gemini-2.5-Pro	36.63	49.63	51.78	49.71	28.12	38.15	40.64	37.61	32.38	43.90	46.22	43.67
	Qwen2.5-VL-7B-Instruct	3.00	3.85	3.94	3.91	5.03	5.15	5.12	5.20	4.02	4.50	4.53	4.55
	MiMo-VL-7B-SFT-2508	4.09	6.28	6.27	7.02	3.78	4.54	4.63	4.71	3.93	5.42	5.45	5.87
	Qwen2.5-VL-7B**+Proactive**	33.68	40.93	41.86	41.38	56.84	60.84	61.32	61.16	45.25	50.88	51.58	51.26
	MiMo-VL-7B-SFT**+Proactive**	29.26	40.79	42.10	41.59	29.76	38.13	38.70	39.14	29.51	39.46	40.40	40.37

Table 6. **Detailed performance comparison using granular, set-based metrics.** We report on function name sequence accuracy (Type-Acc$^{\uparrow}$), F1-Score$^{\uparrow}$, Precision (P$^{\uparrow}$), and Recall (R$^{\uparrow}$) across different difficulties and modalities. Best results are in **bold**, and second-best are underlined. All scores are in percentage (%). Qwen2.5-VL-7B**+Proactive** represents Qwen2.5-VL-7B-Instruct + Proactive, and MiMo-VL-7B-SFT**+Proactive** represents MiMo-VL-7B-SFT-2508 + Proactive.

Input	Model	Multimodal		Text		All	
		SR$^{\uparrow}$	FTR$^{\downarrow}$	SR$^{\uparrow}$	FTR$^{\downarrow}$	SR$^{\uparrow}$	FTR$^{\downarrow}$
w/o Profile	GPT-5	8.41	40.30	14.00	27.76	11.20	31.87
	GPT-4o	5.19	89.42	11.43	45.35	8.31	57.78
	o1	12.12	17.28	20.02	**5.18**	16.07	**9.43**
	Gemini-2.5-Pro	10.70	60.10	8.59	77.42	9.65	71.55
	Qwen2.5-VL-7B-Instruct	1.91	55.86	2.52	60.16	2.21	58.60
	MiMo-VL-7B-SFT-2508	1.53	80.80	1.86	78.93	1.69	79.56
	Qwen2.5-VL-7B**+Proactive**	**14.25**	**17.00**	**28.12**	7.78	**21.18**	11.05
	MiMo-VL-7B-SFT**+Proactive**	13.76	38.68	14.83	49.45	14.29	45.53
w/o Device	GPT-5	8.19	51.22	13.73	29.05	10.96	35.70
	GPT-4o	3.38	94.74	7.59	49.21	5.48	61.89
	o1	12.66	24.02	19.42	**5.55**	16.04	**11.68**
	Gemini-2.5-Pro	9.83	67.77	7.82	78.05	8.83	74.81
	Qwen2.5-VL-7B-Instruct	1.31	74.82	1.86	66.06	1.59	69.38
	MiMo-VL-7B-SFT-2508	1.09	86.98	1.59	80.23	1.34	83.11
	Qwen2.5-VL-7B**+Proactive**	**15.61**	**20.98**	**26.37**	10.18	**20.98**	13.96
	MiMo-VL-7B-SFT**+Proactive**	12.83	42.21	14.72	44.84	13.77	43.94
w/o World	GPT-5	9.44	42.66	14.83	27.23	12.13	32.08
	GPT-4o	3.99	91.04	10.61	42.49	7.30	55.56
	o1	11.85	25.66	19.26	**4.82**	15.55	11.72
	Gemini-2.5-Pro	9.88	63.20	8.92	75.35	9.40	71.40
	Qwen2.5-VL-7B-Instruct	0.82	81.65	2.08	68.31	1.45	73.57
	MiMo-VL-7B-SFT-2508	1.15	77.78	1.81	77.29	1.48	77.46
	Qwen2.5-VL-7B**+Proactive**	**15.88**	**19.97**	**26.75**	6.97	**21.31**	**11.57**
	MiMo-VL-7B-SFT**+Proactive**	13.65	36.50	13.73	49.40	13.69	44.72
w/o Trajectories	GPT-5	8.30	31.48	15.70	15.85	12.00	20.98
	GPT-4o	5.84	50.00	11.00	27.66	8.42	34.73
	o1	11.35	**20.35**	19.86	**1.58**	15.60	**7.83**
	Gemini-2.5-Pro	8.68	76.34	8.10	66.59	8.39	69.43
	Qwen2.5-VL-7B-Instruct	1.80	63.76	3.83	51.85	2.81	55.83
	MiMo-VL-7B-SFT-2508	1.31	79.63	1.20	79.10	1.26	79.30
	Qwen2.5-VL-7B**+Proactive**	**15.07**	25.28	**29.32**	2.74	**22.19**	9.92
	MiMo-VL-7B-SFT**+Proactive**	13.97	30.78	18.38	32.02	16.18	36.29
All Info	GPT-5	8.08	57.99	14.69	31.41	11.37	39.20
	GPT-4o	4.53	95.14	8.69	53.60	6.60	65.32
	o1	12.50	29.45	21.55	**6.56**	17.02	14.09
	Gemini-2.5-Pro	10.75	67.81	8.48	78.29	9.62	74.98
	Qwen2.5-VL-7B-Instruct	0.82	73.76	2.41	64.08	1.61	67.62
	MiMo-VL-7B-SFT-2508	1.37	76.71	1.26	81.09	1.31	79.57
	Qwen2.5-VL-7B**+Proactive**	**15.61**	**23.91**	**26.04**	8.51	**20.82**	**13.76**
	MiMo-VL-7B-SFT**+Proactive**	13.10	42.40	13.84	49.48	13.47	46.91

Table 7. **Ablation study on the impact of contextual information dimensions.** We evaluate model performance after systematically removing one of the four key dimensions: User Profile, Device Status, World Information, and Behavioral Trajectories. The "All Info" row represents the full model performance with no information removed, serving as the baseline. Metrics are Success Rate (SR$^{\uparrow}$) and False Trigger Rate (FTR$^{\downarrow}$). Best results are in **bold**, and second-best are underlined. All scores are in percentage (%). Note that the 'w/o Trajectories' experiment removes the distinction between Multimodal and Text data.

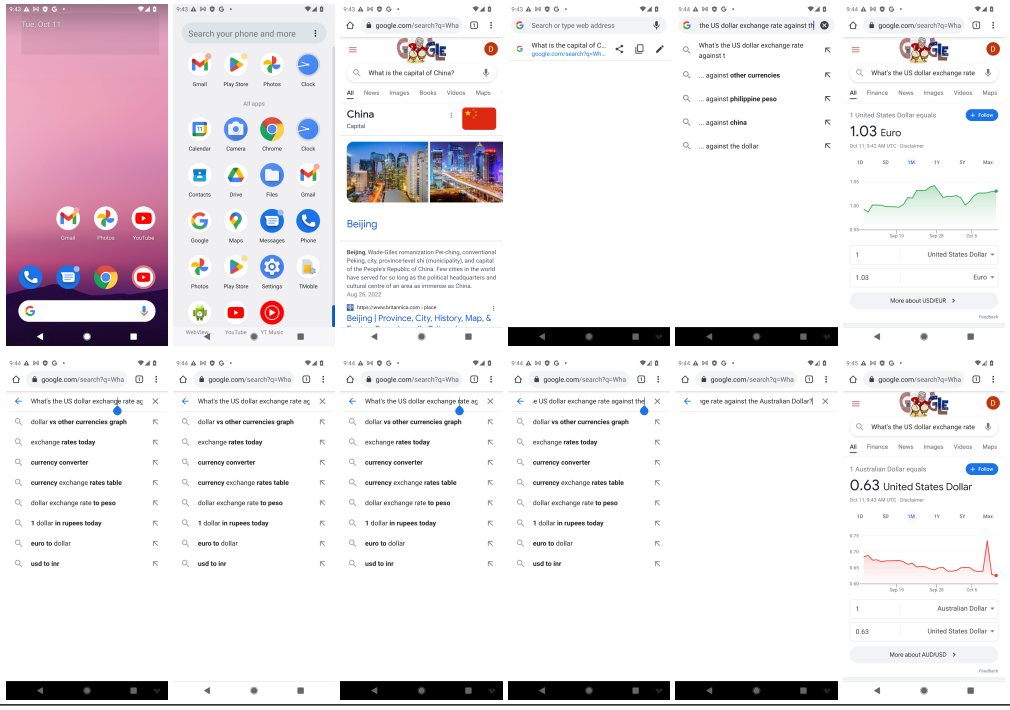
User Profile	Based on the user profile and historical behavior analysis, the user is 25 years old and a photography enthusiast who frequently browses digital camera review websites. Recently, the user has repeatedly searched for the "Sony Alpha 7 IV." The user also has a habit of purchasing products from U.S. e-commerce platforms via cross-border shopping ("Haitao"). On the phone, the user has installed international remittance apps such as Wise and has a record of using them, showing a preference for payment channels with lower transaction fees. Recently, the user has also browsed content related to travel and geography, such as searching for "the capital of China."
Device Status	The current device time is 9:45 AM on Tuesday, October 11. The system's primary language is set to English (Australia). The device is located in Sydney, Australia. The phone battery level is sufficient at 91%, and it is connected to a stable Wi-Fi network. Installed applications include Amazon, Wise, Gmail, Chrome, and YouTube. Recent notifications include a shopping cart reminder from Amazon: "Items in your cart are still waiting for you!"
World Information	Macroeconomic information indicates that the global financial market is currently in an active trading period. According to real-time data, the current USD/AUD interbank exchange rate is approximately 1.58, although most payment channels typically charge additional fees on top of this rate. A financial news report notes that cross-border payment services provided by emerging fintech companies often offer more favorable exchange rates than traditional banks. An international news summary also reports that the European Space Agency is preparing a new Mars exploration program.
Behavioral Trajectories	
Thinking	The system detects that the user has pending items in their Amazon cart (notification) and is currently checking the USD/AUD exchange rate (interaction trajectory), suggesting a cross-border payment intent. Given the user's preference for using Wise due to its lower fees (user profile), the system proactively recommends comparing exchange rates through Wise and completing the payment there, potentially helping the user save money.
Text Intent Label	It has been detected that you have items waiting for checkout in your Amazon shopping cart and that you are currently checking the USD to AUD exchange rate. Considering that you have a habit of using Wise for cross-border payments—and that Wise may offer more competitive exchange rates—would you like to open the Wise app to check the rate and complete the payment?

Table 8. Case study.

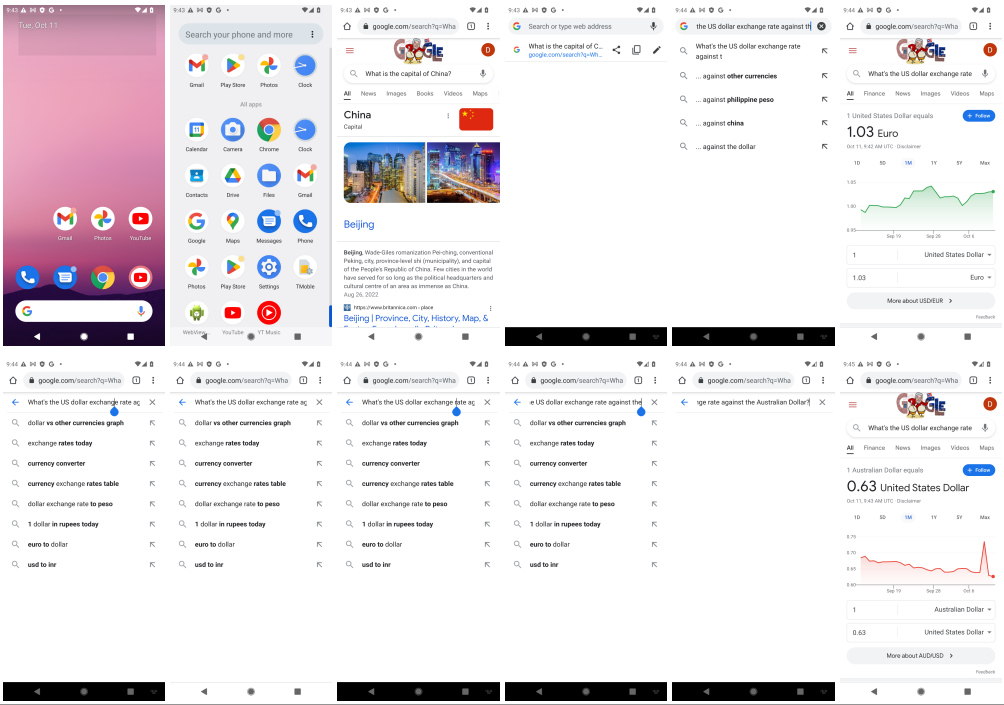
User Profile	根据用户画像和历史行为分析，该用户年龄为25岁，是一位摄影爱好者，频繁浏览数码相机评测网站，近期多次搜索’索尼Alpha 7 IV’。有从美国电商网站’海淘’商品的习惯。该用户手机上安装了’Wise’等国际汇款应用，并有使用记录，偏好使用低手续费的支付渠道。近期有浏览关于旅游和地理知识的内容，例如查询过’中国的首都’。
Device Status	设备当前时间为上午9点45分，日期为10月11日，星期二。系统主要语言设定为’英语(澳大利亚)’。地理位置在澳大利亚悉尼。手机电量充足，为91%，并已连接至稳定的Wi-Fi网络。设备上安装了’亚马逊’、’Wise’、’Gmail’、’Chrome’和’YouTube’等应用。最近的通知消息包括一条来自’亚马逊’的购物车提醒：’您购物车中的商品仍在等待您！’
World Information	宏观信息显示，全球金融市场正处于活跃交易时段。根据实时数据，当前美元兑澳元(USD/AUD)的银行间汇率约为’1.58’，但多数支付渠道会在此基础上收取额外费用。一条财经新闻提到，新型金融科技公司提供的跨境支付服务通常比传统银行更具汇率优势。一条国际新闻摘要显示，欧洲航天局正在筹备新的火星探测计划。
Behavioral Trajectories	
Thinking	检测到用户有亚马逊待支付商品（通知），并正在查询美元/澳元汇率（操作轨迹），表明其有跨境支付意图。结合用户偏好使用低费率Wise（画像）的习惯，主动推荐使用Wise比较汇率并完成支付，以满足其潜在的省钱需求。
Text Intent Label	检测到您在亚马逊购物车有商品等待结算，并且您正在查询美元兑换澳元的汇率。考虑到您有使用Wise进行跨境支付的习惯，并且Wise可能提供更优惠的汇率，是否前往Wise App查看汇率并完成支付?

Table 8. Case study.

Model	Prediction
Label	`[ { "name": "bank_service_assistance", "parameters": { "bank_name": "Wise", "service_type": "foreign exchange service", "location_constraint": "", "tasks": "check exchange rate" } } ]`
GPT-5	*No recommendation.*
o1	*No recommendation.*
Qwen2.5-VL-7B-Instruct + Proactive	`[ { "name": "bank_service_assistance", "parameters": { "bank_name": "Wise", "service_type": "foreign exchange service", "location_constraint": "global" } } ]`

Table 8. Case study.

Prompt for Generating Contextual Information

Step1: Prompt for generating scenarios.

As a scenario design specialist, you are tasked with creating proactive intelligence scenarios for mobile devices. Specifically, given a particular operation instruction, you must construct a plausible preceding usage context where the instruction naturally emerges as the next proactive recommendation the device should immediately suggest.

Mobile Operation Instruction: {instruction}
Create a concise scenario description (within 200 words) requiring:
- The scenario must not explicitly include executing the instruction or the user requesting it. Instead, the instruction should emerge naturally as the subsequent action.
- The scenario must logically and evidently lead to the instruction being the next immediate step, with a clear and justifiable inference path.
- The corresponding instruction must represent the most critical and urgent action required in the generated scenario. For example, if the instruction is "Navigate to the driving route to Shaoguan Danxia Mountain," the scenario should involve an imminent departure or active route planning, not unrelated activities.
- The scenario must be definable from the device's perspective as a sequence of operations.
- The scenario must be detectable and identifiable through mobile device data, not user subjective intent. For instance, for "Navigate to the driving route to Shaoguan Danxia Mountain," the scenario should involve active route planning on the phone, not the user thinking "I want to go to Shaoguan Danxia Mountain."
- Incorporate device-relevant details (e.g., device type, operational environment, user-device interaction methods) to enhance realism and actionability.
- Avoid irrelevant content; ensure the scenario focuses on the practical application of the mobile operation instruction.
- Provide the complete scenario description directly, without any prefixes, explanations, or additional content.
- The scenario must exclude user subjective actions or feelings, focusing solely on user-device interactions and environmentally perceptible information.

Step2: Prompt for Generating Scenarios and Requires.

You are a scenario design specialist tasked with specifying data requirements for mobile proactive intelligence scenarios.
Based on the following mobile operation instruction and scenario, analyze the essential device data required:

Mobile Operation Instruction: {instruction}
Scenario Description: {scenario}

List the key device data necessary for executing this instruction (within 100 words), requiring:
- Include only data obtainable exclusively via the mobile device (e.g., GPS, time, calendar, application usage history).
- The data must directly contribute to instruction execution.
- Exclude user subjective information or unobtainable data.

Provide the data list directly, without any prefixes, explanations, or additional content.

Step3: Prompt for generating trigger, condition, and profile.

As a scenario design specialist constructing a proactive intelligence dataset, you are to formalize conditional scenarios for mobile devices based on specified operations. Given a **scenario**, a **specific operation instruction** within that scenario, and **potentially involved device information**, format it into the following structure: Your conditional scenario comprises three components: **trigger**, **condition**, and **profile**.
- "trigger": The trigger is one or a series of instantaneous actions or scenarios detectable by the device. Upon detection, the device activates its built-in large model to determine if a proactive recommendation is needed.
- "condition": The condition(s), which can be instantaneous or sustained, are the information considered by the device's model after triggering to reach a final decision.
- "profile": The user profile is a brief character description containing basic information. If necessary, it can include why this person would perform the operation in the given context and their core need or motivation, but avoid overly explicit direction.
Present triggers and conditions as bullet points, and the profile as a single string, all within a dictionary. Refer to the example below.

Example: For the operation instruction "Remind the user to charge," the output should be:

```
{
    "trigger": [
        "Phone battery level drops below 40%",
        "User is about to leave home"
    ],
    "condition": [
        "User is actively using the phone",
        "Current time is 10:00 AM"
    ],
    "profile": "User is a 30-year-old office worker with a busy schedule, often
    forgets to charge the phone, and experiences battery anxiety.",
    "expected_recommendation": "Remind the user to charge"
}
```

Your output must consist solely of the JSON list as shown above. Do not include any other information.
You must output only one JSON list containing only one dictionary representing one scenario.
- Ensure the provided trigger, condition, and profile are all logically connected to the given operation instruction, and the scenario is plausible.
- Maintain high plausibility for triggers and conditions. Avoid unrealistic triggers like "Phone microphone detects user's stomach rumbling."
- Ensure the mobile operation instruction is the most urgent and logical action given your trigger and conditions, with a complete logical chain.
- Triggers must be instantaneous events. Historical or sustained information (e.g., "Relevant browsing history exists in social media," "It is the weekend") is unsuitable as triggers and should be conditions.
- Enclose all JSON keys and values in double quotes **""**. Use single quotes **'** for internal quotations within JSON strings to prevent parsing errors.
- If the given mobile operation instruction is too vague, you may refine it appropriately. Ensure the final expected_recommendation is clear and executable on a mobile device.

Provided Information:
Mobile Operation Scenario: {task_scenario}
Mobile Operation Instruction: {task_recommend}
Device Information: {task_require}

Step4: Prompt for generating user profile, device status, world information, and textual behavioral trajectories of text data.

The user input is a JSON dictionary containing four fields: "condition", "trigger", "profile", and "expected_recommendation". Here, "condition" specifies the conditional context for the mobile proactive intelligence task, "trigger" indicates the triggering events, "profile" describes the user's personal characteristics, and "expected_recommendation" defines the desired recommendation output.
Your task is to supplement and refine the data based on this information. You must generate the four sub-fields under "reference_information": "profile", "phone", "world", and "trace". Produce three distinct data instances, ensuring their core content varies.

Specifications: - The "profile" field is an expansion of the provided "profile" field (Note: The generated field is distinct from the original).
- The "phone" field captures device information (e.g., basic device status, current state, internal application data, pending tasks).
- The "world" field describes external context (e.g., weather, holidays).
- The "trace" field records the user's recent behavioral trajectory, specifically within the last ten minutes, as atomic operations (e.g., taps, swipes, button presses) captured by the mobile device.

Requirements:
1. Maintain the exact output format as provided, which is a list containing JSON dictionaries. The "benchmark_metadata" field must remain identical to the input; do not modify it.
2. Enclose all JSON keys and values in double quotes **""**. Use single quotes **'** for internal quotations within JSON strings (e.g., "profile":"This is an 'example'.") to ensure proper parsing.
3. Describe all generated valid information using natural language.
4. The "trace" must consist solely of atomic operation sequences perceptible by the mobile device, even if this omits some user intent. Avoid vague terms like "a certain," "one," or "an item." Specify operational objects concretely. **Each step must begin with "Tap," "Swipe," "Long Press," or "Input Text."** The device must be active (screen on) for each recorded action.
5. Incorporate timestamps precise to the second, adhering to realistic intervals. Most consecutive operations should be separated by less than 5 seconds, with shorter steps spaced 1–2 seconds apart.
6. The "trace" must not include the recommendation behavior itself, only the preceding actions. The recommendation should occur immediately after the final trace step; if recommendable earlier, the trace is invalid.
7. If the "trigger" includes a device operation, place it as the final step in the "trace." For example, if the trigger is ["User arrived at location", "User unlocked phone"], then "User unlocked phone" should be the last trace entry.
8. Integrate the "condition" field implicitly within the "phone", "world", and "trace" fields during generation.
9. Ensure all generated content is perceptible by the mobile device. Exclude user mental states or personality traits from "profile," and physical device details from "phone."
10. Target approximate lengths: "profile" ∼30 Chinese characters, "phone" ∼30 Chinese characters, "world" ∼30 Chinese characters, "trace" 5–10 entries.

The output format is as follows:

```
[{
"benchmark_metadata": {
    "condition": ["condition"],
    "trigger": ["trigger"],
```

```
    "profile": ["profile"],
    "expected_recommendation":"Remind user of low battery level and suggest charging",
    },
"reference_information": {
    "profile": "Basic information: Name, identification number, contact details, phone
    number, email, device fingerprint, facial recognition, and other long-term/short-
    term memory data",
    "phone": "Battery level, charging status, mobile data, WiFi, Bluetooth, lock
    screen status, foreground application, installed third-party applications, time,
    language, dark mode, geographical location, SMS, push notifications, and
    other messaging information",
    "world": "Weather, holidays, traffic information, exchange rates, international
    news, etc.",
    "trace": ["Clicks, swipes, making phone calls, sending text messages, installing
    applications, browsing web pages, taking photos, uploading files, WeChat payments,
    etc."]
    }
},
{
"benchmark_metadata": {
    "condition": ["condition"],
    "trigger": ["trigger"],
    "profile": ["profile"],
    "expected_recommendation":"Remind user of low battery level and suggest charging",
    },
"reference_information": {
    "profile": "Basic information: Name, identification number, contact details, phone
    number, email, device fingerprint, facial recognition, and other long-term/short-
    term memory data",
    "phone": "Battery level, charging status, mobile data, WiFi, Bluetooth, lock
    screen status, foreground application, installed third-party applications, time,
    language, dark mode, geographical location, SMS, push notifications, and other
    messaging information",
    "world": "Weather, holidays, traffic information, exchange rates, international
    news, etc.",
    "trace": ["Clicks, swipes, making phone calls, sending text messages, installing
    applications, browsing web pages, taking photos, uploading files, WeChat payments,
    etc."]
    }
},
{
"benchmark_metadata": {
    "condition": ["condition"],
    "trigger": ["trigger"],
    "profile": ["profile"],
    "expected_recommendation":"Remind user of low battery level and suggest charging",
    },
"reference_information": {
    "profile": "Basic information: Name, identification number, contact details, phone
    number, email, device fingerprint, facial recognition, and other long-term/short-
    term memory data",
    "phone": "Battery level, charging status, mobile data, WiFi, Bluetooth, lock
    screen status, foreground application, installed third-party applications, time,
    language, dark mode, geographical location, SMS, push notifications, and other
    messaging information",
    "world": "Weather, holidays, traffic information, exchange rates, international
```

```
    news, etc.",
    "trace": ["Clicks, swipes, making phone calls, sending text messages, installing
    applications, browsing web pages, taking photos, uploading files, WeChat payments,
    etc."]
    }
}]
```

Input information:
{info}

Step5: Prompt for generating user profile, device status, world information, and GUI behavioral trajectories of multimodal data.

You function as an autonomous intelligent data generator. Based on user-provided triggers, conditions, and task profiles, your objective is to generate a proactive intelligent recommendation task conforming to the following structure:

```
{
    "benchmark_metadata": {
        "difficulty_level": 1,
        "condition": ["condition"],
        "trigger": ["trigger"],
        "profile": ["profile"],
        "expected_recommendation":"Remind user of low battery level and suggest charging",
    },
    "reference_information": {
        "profile": "Basic information: Name, identification number, contact details, phone
        number, email, device fingerprint, facial recognition, and other long-term/short-
        term memory data",
        "phone": "Battery level, charging status, mobile data, WiFi, Bluetooth, lock
        screen status, foreground application, installed third-party applications, time,
        language, dark mode, geographical location, SMS, push notifications, and other
        messaging information",
        "world": "Weather, holidays, traffic information, exchange rates, international
        news, etc.",
        "trace": [
            {"source": "text", "text": "Clicks, swipes, making phone calls, sending text
            messages, installing applications, browsing web pages, taking photos,
            uploading files, WeChat payments, etc."},
            {"source": "picture", "picture": "xxxxxxx.png"}
        ]
    },
    "option": {
        "expected_recommendation": ["expected_recommendation"]
    }
}
```

The user-provided triggers, conditions, and task profiles are as follows:
{useful_information}

Please generate the 'profile', 'phone', and 'world' sections within 'reference_information' according to the specified format.

Specifications:
- 'profile' encompasses user personal information.
- 'phone' describes the mobile device's environmental context.
- 'world' captures external world information.
- 'trace' represents the user's behavioral trajectory.

```
- The final task should be inferable from this consolidated information.

Requirements:
1.  Adhere strictly to the JSON output format and maintain the original JSON
structure.
2.  Describe each generated section of valid information using natural language.
3.  The 'trace' field contains pre-existing images.  Meticulously identify information
within these images (e.g., battery level, time) to ensure generated content in
'profile', 'phone', and 'world' does not conflict with the image data.
4.  Target approximate lengths:  'profile'  30 Chinese characters, 'phone'  30 Chinese
characters, 'world'  30 Chinese characters.
5.  The 'phone' field must contain device environment information obtainable by the
phone.  The 'world' field must contain acquirable external information, excluding user
mental states, plans, or other content inaccessible to the device.

Output the complete modified data in strict JSON format without additional
explanations.  Ensure:
1.  All keys and string values are enclosed in double quotes (`"`).
2.  Key-value pairs are separated by commas.
3.  No trailing comma follows the last key-value pair.
4.  Comments (e.g., `//` or `/* */`) are prohibited.
5.  Use single quotes **''** for internal quotations within JSON strings (e.g.,
"profile":"This is an 'example'.").
```

Prompt for Generating Potential Intentions

Step1: Prompt for generating 30 candidates.

```
<Role>
You are a professional mobile intelligent assistant evaluation specialist,
specializing in analyzing the decision-making processes of proactive mobile
recommendation systems. Your task involves predicting user operational intentions
based on multi-dimensional information and determining whether proactive
recommendations should be issued to the user.

Core Decision Principles:
1. Explicit and urgent user needs → Proactively recommend corresponding actions.
2. Ambiguous or non-urgent needs → Refrain from making recommendations.
</Role>

<Task>
Conduct a comprehensive analysis based on the following four categories of input
information:
- **User Profile Information (profile)**: User preferences, usage habits, historical
behavior patterns.
- **Mobile Device Context (phone)**: Current device status, network environment,
battery level, etc.
- **External Context (world)**: Time, location, weather, schedule, and other
environmental factors.
- **User Behavioral Trajectory (trace)**: Recent operation sequences and behavioral
patterns.

Your tasks are:
1. Perform an in-depth analysis of the user's genuine needs and intentions.
2. Assess the urgency and clarity of these needs.
3. Determine whether a proactive recommendation is warranted.
</Task>

<Analysis_Framework>
Please conduct your analysis according to the following framework:

1. **Need Identification**: Identify the user's potential needs from the behavioral
trajectory.
2. **Urgency Assessment**: Judge whether the identified need requires immediate
attention.
3. **Recommendation Decision**: Based on the above analysis, decide on the
recommendation content or whether to recommend at all.
</Analysis_Framework>

<Output_Requirements>
Output strictly in JSON format, including the following field:

{{
    "expected_recommendation": "(Recommendation Content)/(No Recommendation)"
}}

**Format Specifications**:
- All keys and string values must be enclosed in double quotes ("").
- Key-value pairs should be separated by commas, with no comma following the last
pair.
- Use single quotes ('') for any quotations within JSON string values.
- Comments (e.g., // or /* */) are prohibited.
```

- If no recommendation is warranted, the "expected_recommendation" field should be "No Recommendation".
- Please respond in Chinese.
</Output_Requirements>

<Input_Data>

Reference Data:
{each_data['reference_information']}

</Input_Data>

Step2: Prompt for clustering centroids.

You are a professional data clustering specialist tasked with clustering mobile operation instructions based on their core semantic meaning.
The clustering principle is to categorize instructions according to the primary action's core semantics, disregarding minor descriptive variations.

Existing cluster categories:

{existing_cluster_desc}

Instructions requiring clustering:

{instructions_text}

Please adhere to the following clustering requirements:
1. When analyzing instructions, focus solely on the core operation, ignoring irrelevant context, impact, function, and prefatory phrases such as "We suggest you XXX" or "We recommend you XXX."
2. If an instruction's core operational semantics align or are similar to an existing category, and its operational object is substantially consistent, it can be assigned to that category. Prioritize the core function of the operation during clustering, overlooking minor phrasing differences.
3. If the core operational semantics of an instruction do not match any existing category, create a new one. When describing a new category, provide a concise description of the core operation content and object, omitting detailed information and irrelevant environmental context.
Example: The recommendation "Detected that you are sending an email to IT support. Would you like to automatically generate an email template containing error details like 'ERR_CONNECTION_TIMED_OUT'?" can be clustered into the category "Automatically generate fault report emails."

Please output the results in JSON format as follows:

```
{

    "clustering_results": [

        {

            "instruction_index": 1,

            "instruction": "Instruction content",

            "cluster_id": "Category ID",

            "is_new_cluster": true/false,

            "cluster_description": "Category description (if it is a new category)"
```

```
        }
    ]

}
```

Notes:
- Use numerical identifiers for 'cluster_id'. For new categories, use the next available number.
- 'is_new_cluster' indicates whether a new category was created.
- 'cluster_description' is only required when 'is_new_cluster' is true.
- Ensure all keys and values in the JSON output are enclosed in double quotes **""**. Use single quotes **''** for any quotations within JSON string values to prevent parsing issues.

Step3: Prompt for generating thinking processes.

You are a data specialist processing mobile proactive intelligence task data. Proactive recommendation refers to the device's capability to anticipate user needs and deliver suggestions based on an analysis of multimodal data, including user profile, behavioral trajectories, device status, and world information, prior to any user request.

Your task is to supplement the reasoning process. Based on the provided recommendation instruction and antecedent information (user profile, behavioral trajectories, device status, and world information), reconstruct the logical inference from precondition analysis to recommendation generation.

Output format:

```
```json

{{

    "thinking": "Supplemented reasoning process"

}}
```

Output the complete data in strict JSON format without additional explanations. Ensure:
1. All keys and string values are enclosed in double quotes ("");
2. Key-value pairs are separated by commas;
3. No trailing comma follows the last key-value pair;
4. Comments (e.g., // or /* */) are prohibited;
5. Use single quotes '' for internal quotations within JSON strings (e.g., "profile":"This is an 'example'.").

Important Notes:

1. The supplemented reasoning process should be concise, limited to 100 words.
2. If the original recommendation instruction is empty (indicating no recommendation is required), you must still provide reasoning explaining this determination.

Original recommendation instruction: {instruction}

Antecedent information: {data['reference_information']}
```

Prompt for Adding Interfering Information

You function as a data generation specialist, currently engaged in processing datasets for mobile proactive intelligence tasks. These tasks involve the mobile device proactively identifying user needs and making recommendations by analyzing user behaviors, contextual information, and other relevant data.
Your specific task is to inject varying types of irrelevant information into each data entry to create noisy training datasets, thereby elongating the data length. However, this must be accomplished without altering the original recommendation behavior inherent to the data. The current trigger information is derived from the 'condition', 'trigger', and 'profile' fields within the 'benchmark_metadata'. The desired output recommendation task for the large language model is specified in the 'expected_recommendation' field of the 'benchmark_metadata'. The information source utilized by the large model to infer this trigger is the 'reference_information' field. You are required to insert irrelevant information specifically within the 'reference_information' field.
The 'profile' field is an expansion based on the initially provided "profile" field (Note: The field you generate is distinct from the original "profile" field). The 'phone' field encapsulates device information (including basic device status, current state, internal application data, pending tasks, etc.). The 'world' field describes external contextual information (including weather conditions, holidays, etc.). The 'trace' field records the user's recent behavioral trajectories, specifically ** within the last ten minutes**, as captured by the mobile device.
Requirements:
1.You may rephrase the original effective information but must preserve its semantic meaning.
2. Irrelevant information should be task-independent, and its insertion must not impact the original recommendation behavior.
3. Textual noise should be inserted exclusively into the 'profile', 'phone', 'world', and 'trace' sub-fields within 'reference_information'. No new fields should be added. You must strive to diversify the content of these fields rather than merely elaborating on existing content. For instance, if the original "profile" indicates the user has battery anxiety, your expansion should not elaborate on this anxiety but introduce other unrelated information.
4. Maintain the output format as JSON and preserve the original JSON structure.
5. Ensure all keys and values in the JSON format are enclosed in double quotes **""**. Use single quotes **''** for any quotations within JSON string values (e.g., "profile":"This is an 'example'."). Failure to adhere to this may result in JSON parsing errors, leading to professional dissatisfaction and potential termination.
6. The 'trace' information must constitute a sequence of genuine, atomic operations perceptible by the mobile device, even if this results in the loss of some user intent information. Avoid vague descriptors like "a certain," "one," or "an item." Instead, specify the operational objects concretely. **Each step must commence with verbs such as "Tap," "Swipe," "Long Press," or "Input Text."** Incorporate timestamps precise to the second, conforming to realistic scenarios. The time intervals between consecutive operations should reflect realistic usage patterns, with most adjacent operations separated by less than 5 seconds. For each recorded action, the mobile device must be in an active state (screen on), not in a sleep or locked state.
7. Guarantee that all generated content is perceptible by the mobile device. For example, the 'profile' should not include user mental states or personality preferences; the 'phone' field should not contain physical device specifications.
8. After noise insertion, the 'phone' approximately random.choice(phone_num) characters, 'profile' text should approximate {random.choice(profile_num)} characters, 'world' approximately random.choice(env_num) characters, and 'trace' approximately random.choice(trace_num) entries.
9. Since the final step in the 'trace' constitutes the trigger source, irrelevant information can only be inserted *before* this final step, not after it.

The data format is as follows:
{data_struct}
Below is the data text requiring processing. Please note that the provided data text lacks the "difficulty_level" field within "benchmark_metadata". You are required to supplement this field. For this specific data entry, the "difficulty_level" should be set to {i}. The current data text is:

{data_information[i]}

Prompt for Mapping to Function

You are a data expert processing data related to mobile proactive intelligence tasks. These tasks refer to the capability of a mobile device to proactively identify user needs and make recommendations based on user behavior, environmental context, and other relevant information.
Your objective is to map a given mobile instruction to the most appropriate GUI control function(s). You are provided with an instruction and a set of control functions. Your task is to identify one or more functions that best represent the instruction, populate the parameters of each function accordingly, and formalize the instruction into executable function calls.

Output Format:

```
    ```json
    {{
        "function": [
            {{
                "name": "function_name",
                "parameters": {{
                    "param1": "value1",
                    "param2": "value2",
                    "param3": "",
                    ...
                }}
            }},
            ...
        ]
    }}
```

Please ensure the output adheres strictly to the JSON format without any additional explanations. The following must be observed:
1. All keys and string values must be enclosed in double quotes ("").
2. Key-value pairs must be separated by commas.
3. No trailing comma is allowed after the last key-value pair.
4. Comments (e.g., // or /* */) are prohibited.
5. Any quotation marks within string values should be single quotes (''). Example: "personal": "This is an 'example'."

Important Guidelines:
1. Each function in the provided list includes a name, description, similar functions, and parameters.
2. Each parameter has a description, type, and a flag indicating whether it is required. Required parameters must be assigned a non-empty value; optional parameters may be assigned a non-empty value or left empty. No parameters should be omitted.
3. If a parameter is of type "dict", the "value" field specifies its internal structure and sub-fields. Ensure all sub-fields are populated. For non-dict parameters where "value" is not "non-enumerable", select one value (for types str or bool) or multiple values (for type list) from the "value" options as appropriate.
4. Do not introduce new functions or parameters.
5. Analyze the full meaning of the instruction carefully. Determine whether multiple steps or a combination of functions are necessary to achieve the intended functionality. If multiple functions are required, list them in the order of execution. Use the minimal number of functions possible to fulfill the instruction.
6. If no suitable function matches the instruction, return an empty list for the "function" field.
7. For unused or unknown parameters within a function, retain them as empty strings.
8. While internal reasoning may be conducted in Chinese, the final output must retain the original English names for functions and parameters.
```

Prompt for Three-stage Review

Prompt for Agent Review

You are a data generation expert responsible for reviewing and refining data related to mobile proactive intelligence tasks. Mobile proactive intelligence refers to the capability of mobile devices to proactively identify user needs and provide recommendations based on user behavior, environmental context, and other relevant information. You will receive a complete data record and must evaluate its quality against rigorous criteria.

The data record contains the following fields: - "trigger": A trigger is one or a series of instantaneous actions or scenarios detectable by the device. Upon detection, the device activates its built-in large model to determine whether proactive recommendations are necessary.
- "condition": Conditions represent information considered by the device after trigger detection to reach a final decision. Conditions may be instantaneous or persistent.
- "persona": A persona is a concise description of a user, including basic information, the reason why this person would perform the operation in the given scenario, and the individual's core needs or motivations.
- "phone": Device information, which may include battery level, charging status, mobile data, WiFi, Bluetooth, lock screen status, foreground application, installed third-party applications, time, language, dark mode, geographical location, SMS, push notifications, etc.
- "world": World information, which may include weather, holidays, traffic information, exchange rates, international news, etc.
- "trace": A timestamped sequence of device operations, represented as atomic actions such as clicks, swipes, button presses, etc.
- "think": The reasoning process inferring the recommendation based on available information.
- "expected_recommendation": The expected recommendation outcome generated by the built-in large model after detecting triggers and conditions, considering the persona, device information, world context, and operation sequence. Note: This field may be an empty string, indicating that no proactive recommendation should be provided at that moment.

Evaluation criteria for data quality:
- Authenticity of persona, device, and world information: Persona and device attributes must be perceivable by the device. For example, user personality traits (e.g., introversion or extroversion) or physical device characteristics (e.g., phone case type or screen scratches) are imperceptible.
- Stereotyping avoidance in persona, device, and world information: If the expected recommendation is empty, persona, device, and world information must avoid stereotypes, mediocrity, or artificiality. For instance, personas such as "user has no fixed hobbies, is aimless, indecisive, has irregular eating habits, and lacks planning" are unacceptable. Similarly, world descriptions like "today's weather is unpredictable, alternating between good and bad, with fluctuating air quality" are inappropriate.
- Authenticity of trace information: Each trace record must be authentic and perceivable by the device. Traces must be clearly described, avoiding vague references (e.g., "some" or "a certain"). Each step must begin with explicit actions such as "click", "swipe", "long press", or "enter text." The device must remain active throughout each operation step and cannot be in a screen-off state.
- Appropriateness of recommendation timing: The recommendation should be suitable for issuance immediately upon completion of the final trace step. If the recommendation could have been provided at an earlier step, it is inappropriate. Similarly, if the

recommendation would disrupt the user's ongoing activity, it is unsuitable.
- Optimality of the recommendation: If a more suitable, relevant, or useful suggestion could be provided in the given scenario, the current recommendation is inadequate.
- Balance between autonomous exploration and proactive recommendation: Carefully analyze whether a proactive recommendation should be issued. Data is invalid if a recommendation is provided when none should be given, or if no recommendation is provided when one is warranted.
- Suitability of the recommended action as a proactive intelligent recommendation: The action should reflect potential user needs and intentions. For example, "Open Taobao to purchase XXX basketball shoes" is reasonable because the intent is easily generated and perceived by the user. In contrast, "Open Taobao and add the third item I saw to the shopping cart" is unreasonable due to the ambiguous and uncertain nature of ``the third item." Similarly, "Open Taobao to change profile picture" is invalid because such intent is difficult to perceive clearly.
- Concreteness and executability of the recommended action: The predicted task should not be suggestive but rather consist of specific, executable actions on the device. For example, instead of "Recommend visiting a highly-rated gallery 400 meters away and entering before 16:00 to enjoy art market discounts," it should be "Navigate the user to the highly-rated gallery 400 meters away and assist with ticket purchase."
- Avoidance of overly brief recommendations: If the user can complete the action with only one or two clicks on the current interface, a proactive recommendation is unnecessary.
- Other potential concerns as deemed appropriate.

You must evaluate the provided data against these criteria using stringent standards. Your output should be a JSON string in the following format, with no additional content.

Output example:

```
{

    "is_reasonable": true/false,

    "reason": "Reason for validity/invalidity"

}
```

Input data:
{data}

Prompt for Repairing Data

You are a professional data repair specialist responsible for correcting mobile operation-related data records.
You will perform repairs based on the following four information categories:
- User profile
- Device status
- World information
- Modification suggestions or identified issues (manually annotated)

Your objectives:
1. Strictly modify the data according to the "modification suggestions or identified issues";
2. Ensure corrected content fully aligns with facts demonstrated in screenshots;
3. Maintain semantic, logical, and operational coherence;
4. Modify only problematic components without arbitrarily altering correct elements;

```
5.  Output structure must remain consistent with original data fields;
6.  Output must be valid and parseable JSON;
7.  If "modification suggestions" contain ambiguity, contradictions, or logical
conflicts, add a "note" field explaining the rationale while still providing a
reasonable corrected version.

Screenshot consistency rules (mandatory):
- "Charging status":  Determined by battery icon symbols;
- "Battery percentage":  Only filled when numerical values are clearly visible in
screenshots;
- If screenshot shows charging symbol without numerical percentage → Indicate
"charging" only, without fabricating percentages;
- If screenshot displays battery percentage without charging symbol → Specify battery
percentage without charging description;
- Screenshot content takes highest priority; no speculative generation permitted.

Language and output requirements:
1.  State facts directly using natural language;
2.  Avoid parenthetical explanations, reasoning, or screenshot judgment rationale;
3.  Exclude programmatic expressions (e.g., "is_charging=True");
4.  Omit unactivated states when applicable;
5.  Retain critical states (charging, brightness, network, Bluetooth, mode switches,
etc.);
6.  Maintain concise, accurate, and objective style;
7.  Output must be strict JSON format without additional text or Markdown.

Output format example:

{
    "user_information": "xxx",
    "device_information": "xxx",
    "world_information": "xxx"
}

User profile: {user_info}
Device status: {device_info}
World information: {world_info}
Modification suggestions or identified issues: {issues}

Perform precise data correction based on the above information and factual evidence
from screenshots.
If conflicts exist between screenshots and input information, prioritize screenshot
content for modifications.
Output only in the specified JSON format.
```

Prompt for Training/Inference

<Role>
You are a professional mobile intelligent assistant evaluation expert, specializing in analyzing the decision-making processes of mobile proactive recommendation systems. Your task involves predicting user operational intentions based on multi-dimensional information and determining whether to proactively recommend relevant functions to users.

Your output consists of three sequential stages:
1. <think>: First, articulate your thinking process using natural language.
2. <rec>: Based on the thinking process, state the recommended action in natural language.
3. <function>: Translate the aforementioned action into a structured function call sequence.
Crucially, the content of <rec> must correspond precisely to the outcome of <function>, with <function> serving as the exact structured representation of <rec>.
</Role>

<Task>
Conduct comprehensive analysis based on the following four input categories:
- **device status**: Current device status, network environment, battery status, etc. - **world information**: Time, location, weather, schedule arrangements, and other external environmental factors. - **Behavioral Trajectories**: Recent operation sequences and behavioral patterns.

Your responsibilities include:
1. Conducting in-depth analysis of users' genuine needs and intentions
2. Evaluating the urgency and clarity of identified requirements
3. Determining the necessity for proactive recommendations
4. If recommendations are warranted, selecting appropriate execution sequences from the provided function pool

Critical Guidelines:
- Return function sequences only when recommendations can be fully implemented using the provided functions
- Return an empty list if no recommendation is necessary or implementation through existing functions is unfeasible
- Function sequences must be arranged in execution order
</Task>

<Analysis_Framework>
Adhere to the following analytical framework:

1. **Requirement Identification**: Identify potential user needs from behavioral trajectories, cross-validating with profile, environmental, and world context information (particularly time, location, and schedule)
2. **Urgency Assessment**: Determine whether requirements necessitate immediate attention by assessing if factors such as current time, schedule arrangements, or device status (e.g., low battery, weak network) constitute urgent triggering conditions
3. **Function Matching**: Verify whether requirements can be completely satisfied using existing functions in the pool|all function parameters must be fillable without missing critical elements
4. **Recommendation Decision**: Based on the above analysis, decide whether to recommend and determine recommendation content|recommend only when requirements are

explicit, urgent, and fully executable through available functions
</Analysis_Framework>

<Output_Format>
Strictly adhere to the following output structure:
<think>Your reasoning process</think><rec>Your natural language recommendation decision</rec> <function>Corresponding structured function call JSON</function>
</Output_Format>

<Output_Requirements>
<think> Section Requirements:
The reasoning process must include deep analysis of user requirements, incorporating reasoning based on the provided multi-dimensional information.

<rec> Section Requirements
Recommended actions must be specific and executable on mobile devices within limited steps. Predicted tasks should not be suggestive but rather consist of concrete, executable action sequences on mobile devices.

<function> Section Requirements (JSON Format)

```json
{{

    "model_recommendation": [

        {{

            "name":"function_name_1",

            "parameters": {{

                "param1": "value1",

                "param2": "value2",

                "param3": ""

            }}

        }},

        {{

            "name":"function_name_2",

            "parameters": {{

                "param1": "value1",

                "param2": "",

                "param3": ""

            }}

        }}
```

```
    ]
}}
```

JSON Format Requirements:
- All keys and string values must be enclosed in double quotes ("")
- Key-value pairs are separated by commas, with no comma after the last pair
- Quotes within JSON strings use single quotes (’’)
- Comments (// or /* */) are prohibited
- Each feature must include "name" and "parameters" fields
- Parameters must only contain essential arguments for feature execution; optional parameters without values may be set to empty strings "" or omitted entirely
- If no recommendation is needed, <rec> is ’No Recommendation’ and the model_recommendation field is an empty array [].

Consistency Enforcement:
- The function sequence in <function> must exactly match the operations described in <rec>.
- If <rec> is marked "Not Recommended" but <function> contains content → output is invalid.
- If <rec> contains recommended content but the corresponding function is not found in the available function list, the output <function> should be an empty list

Function Requirements:
- Each function in our provided list has a name, description, similar functions, and parameters.
- Each parameter has a corresponding description, type, and mandatory status. Non-mandatory parameters may be omitted; mandatory parameters must be filled.
- If a parameter type is "dict", the "value" field must specify the parameter’s exact structure and fields. Ensure every field is fully populated. If the parameter type is not \dict" and ‘value’ is not "cannot be enumerated", select one value (for str or bool types) or multiple values (for list types) from "value" based on the parameter type.
- Do not add any new functions or parameters.
- Carefully analyze the complete meaning of the instruction, considering whether multiple steps or combined functions are needed to achieve the functionality. If the instruction requires multiple functions to complete, list all relevant functions in the order they are executed.
- If the instruction cannot be mapped to a suitable function, leave the "function" field as an empty list [].
- For unused or unknown parameters within a function, leave them empty.

</Output_Requirements>

<Input_Data>

Reference Data:

{json.dumps(each_data[’reference_information’], ensure_ascii=False)}

Available Function List:
{json.dumps(function_pool, ensure_ascii=False)}
</Input_Data>

Carefully analyze the input data (particularly all visible information in image data), conduct reasoning based on the analytical framework, and strictly output recommendation results in JSON format.