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 1.

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 2 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 3.

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 1.

C.1. Granular Metric Definitions

The following metrics are used to generate the results in Table 1. 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.

$$\text{Type-Acc}(S_{pred}, S_{label}) = \mathbb{I}\left(\langle c_{pred,i}.\text{name}\rangle = \langle c_{label,j}.\text{name}\rangle\right) \tag{1}$$

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). $\text{R} = \frac{|P_{set} \cap L_{set}|}{|L_{set}|}$

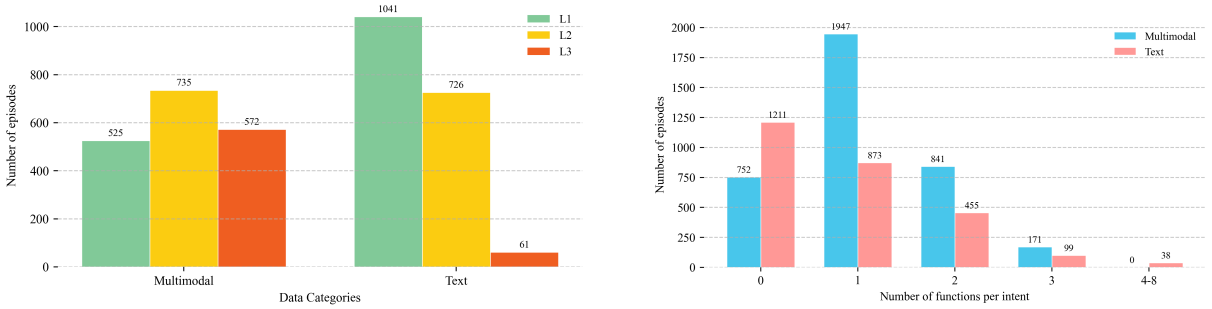
F1-Score (F1) is the harmonic mean of Precision and Recall, providing a single balanced score for partial correctness. $\text{F1} = 2 \times \frac{\text{P} \times \text{R}}{\text{P} + \text{R}}$

C.2. Analysis of Detailed Results

The detailed results in Table 1 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 1. 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 2. Exemplary implementation of the book_transport API.

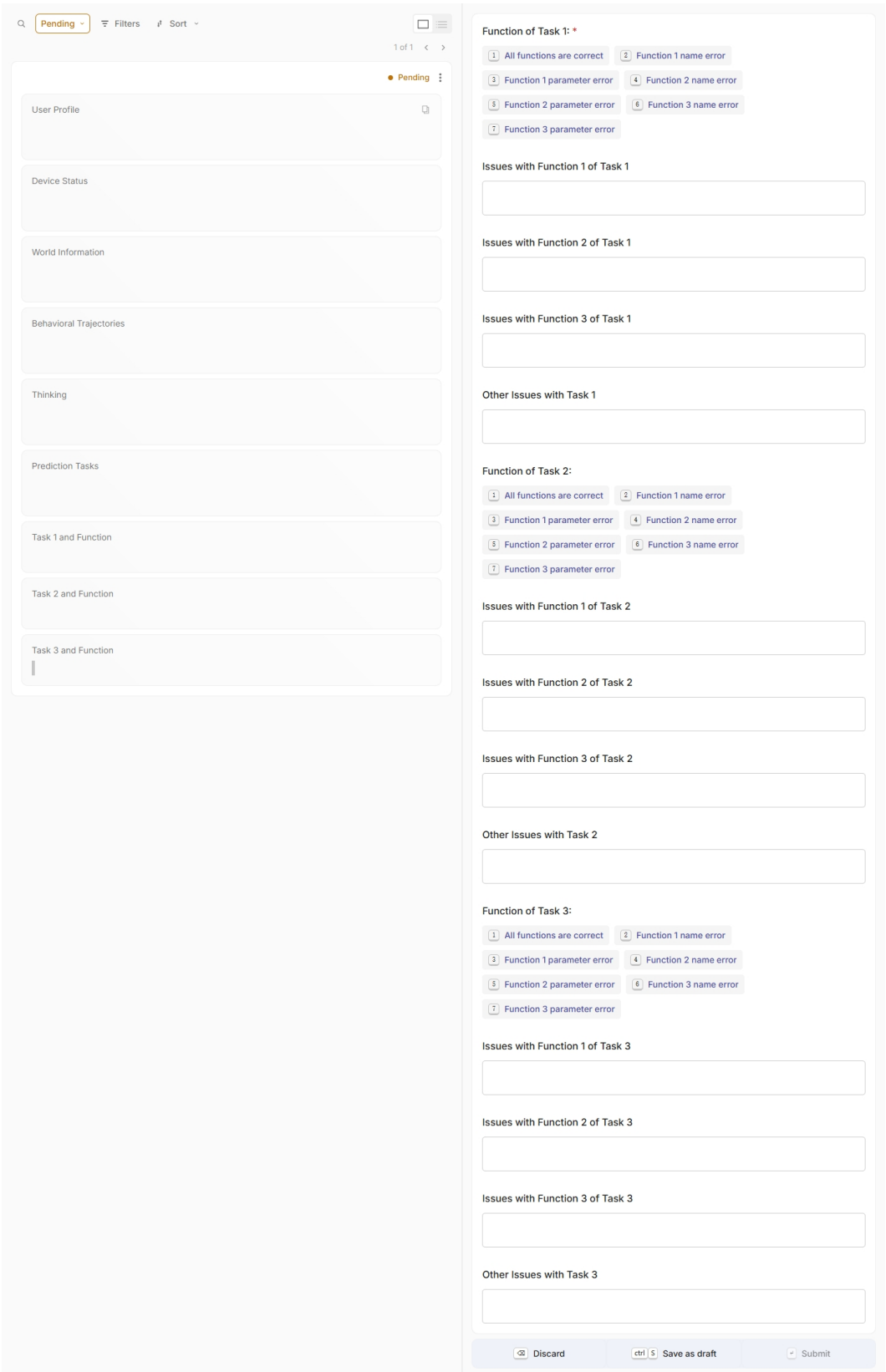


Figure 3. The data annotation platform.

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 2. 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 3. 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 1. **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 2. **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.

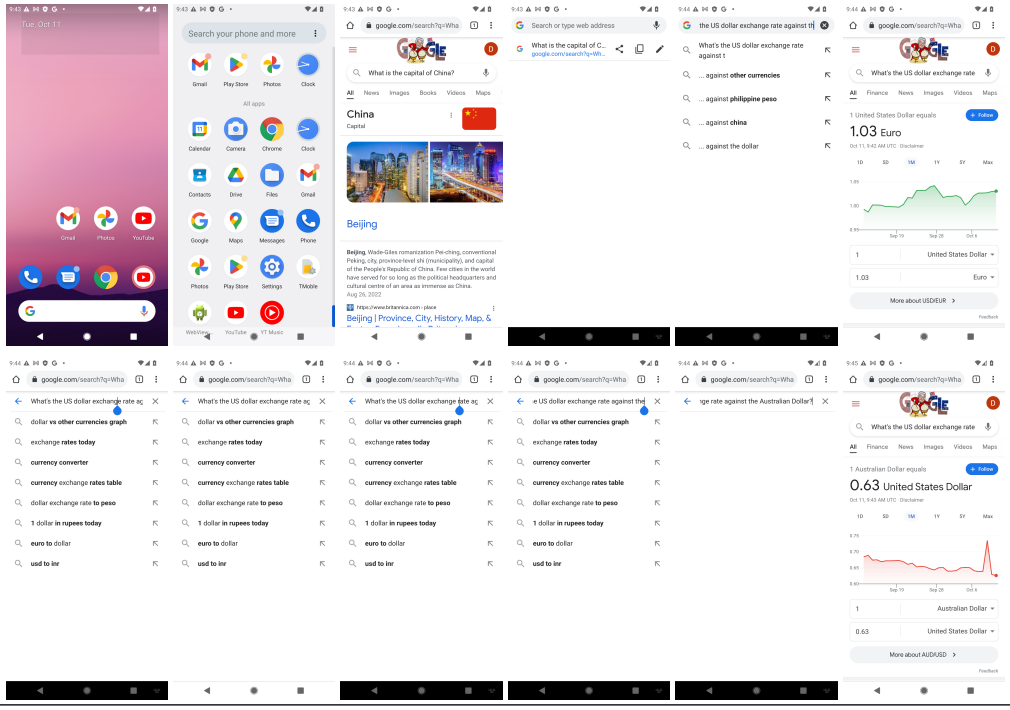
User Profile	Based on the user profile and historical behavior analysis, the user is 25 years old and a photography enthusiast who frequently browses digital camera review websites. Recently, the user has repeatedly searched for the "Sony Alpha 7 IV." The user also has a habit of purchasing products 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 3. Case study.

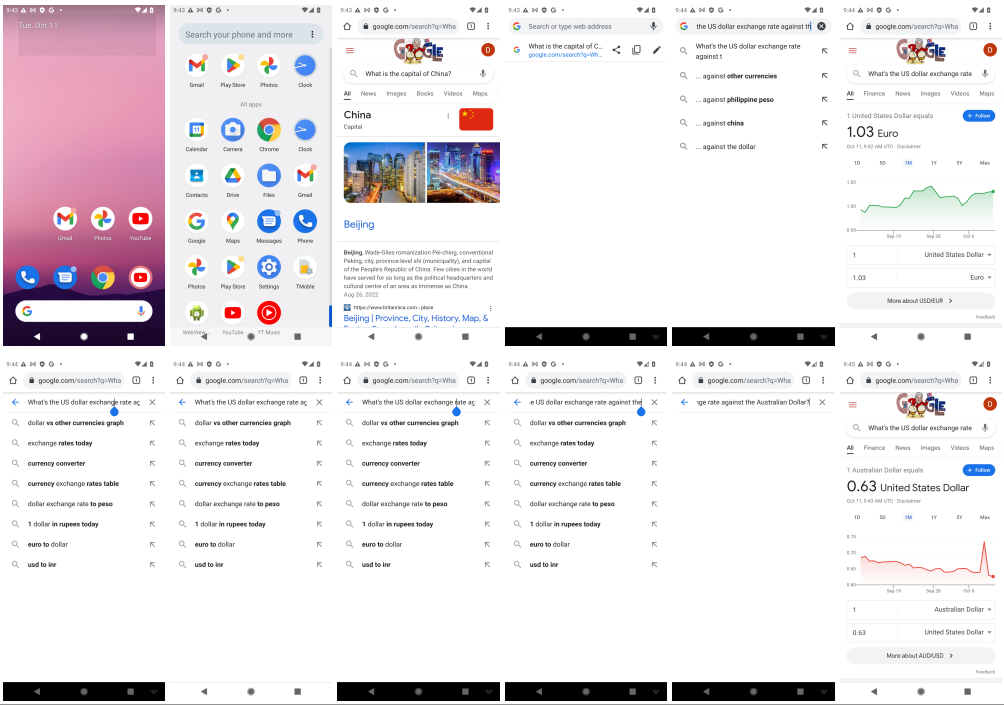
User Profile	根据用户画像和历史行为分析，该用户年龄为25岁，是一位摄影爱好者，频繁浏览数码相机评测网站，近期多次搜索’索尼Alpha 7 IV’。有从美国电商网站’海淘’商品的习惯。该用户手机上安装了’Wise’等国际汇款应用，并有使用记录，偏好使用低手续费的支付渠道。近期有浏览关于旅游和地理知识的内容，例如查询过’中国的首都’。
Device Status	设备当前时间为上午9点45分，日期为10月11日，星期二。系统主要语言设定为’英语(澳大利亚)’。地理位置在澳大利亚悉尼。手机电量充足，为91%，并已连接至稳定的Wi-Fi网络。设备上安装了’亚马逊’、’Wise’、’Gmail’、’Chrome’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 3. 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 3. 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.