

# NaturalGAIA: A Verifiable Benchmark and Hierarchical Framework for Long-Horizon GUI Tasks

Zihan Zheng<sup>1\*</sup>, Tianle Cui<sup>1\*</sup>, Taoran Wang<sup>1</sup>, Fengtao Wang<sup>1</sup>,  
Jiahui Pan<sup>1</sup>, Lewei He<sup>1†</sup>, Qianglong Chen<sup>2†</sup>

<sup>1</sup>South China Normal University   <sup>2</sup>Zhejiang University

zhengzihan994@m.scnu.edu.cn   helewei@m.scnu.edu.cn   chenqianglong@zju.edu.cn

## Abstract

Despite significant advances in LLM-driven GUI agents, the field remains constrained by the challenge of reconciling high-fidelity realism with verifiable evaluation accuracy. To address this, we introduce NaturalGAIA, a verifiable evaluation dataset grounded in real-world human GUI interaction intents. By decoupling logical causal pathways from linguistic narratives, it rigorously simulates natural human intent, characterized by cognitive non-linearity and contextual dependencies. Furthermore, we propose LightManus-Jarvis, a hierarchical collaborative framework where LightManus manages dynamic topological planning and context evolution, while Jarvis ensures execution precision via hybrid visual-structural perception. Experiments demonstrate that our approach achieves a Weighted Pathway Success Rate of 45.6%, significantly outperforming the state-of-the-art baseline (21.1%), while reducing token consumption by 75% and execution time by 76%. These results validate the efficacy of the macro-planning and micro-execution paradigm in handling complex naturalized tasks. Our code is publicly available at: <https://github.com/KeLes-Coding/NatureGAIA>.

## 1 Introduction

Although LLM-driven GUI agents have advanced significantly (DeepSeek-AI et al., 2025; OpenAI et al., 2024; Qwen et al., 2025), current research faces an "Evaluation-Realism Dilemma" (Mohammadi et al., 2025; Gera et al., 2025). Realistic benchmarks like OSWorld (Xie et al., 2024) and RealWebAssist (Ye et al., 2025) lack deterministic ground truth due to reliance on unstable MLLM judges or manual verification, preventing accurate measurement of the "reasoning-execution gap" (Dong et al., 2025; Liu et al., 2025). Conversely,

traditional static benchmarks (Deng et al., 2023; He et al., 2024a; Zhou et al., 2023) use simplified, de-contextualized instructions that fail to capture the cognitive non-linearity of human intent (Ye et al., 2025; Zhang et al., 2025a). However, these fail to capture the cognitive non-linearity of human intent (Dong et al., 2025; Wei et al., 2025), leading to overestimated success rates that do not reflect robustness in unstructured environments (Dai et al., 2025; Kartik et al., 2025; He et al., 2025).

Simultaneously, existing architectures face stability issues in long-range, noisy workflows. Mainstream end-to-end vision models (Wang et al., 2025a; Niu et al., 2024) often prioritize visual "shortcut learning" over generalizable procedural knowledge (Zhang et al., 2023; Cheng et al., 2024). This monolithic design suffers from a dual bottleneck: the lack of dynamic context management causes semantic drift in long sequences (Zhang et al., 2025a; Cai et al., 2025), while pure visual perception is prone to "coordinate hallucinations" in resource-constrained environments, hindering fine-grained precision (Yang et al., 2025; Yin et al., 2025). Therefore, balancing macroscopic planning coherence with microscopic execution precision (He et al., 2024b) remains a core unresolved challenge.

To address the limitations of existing evaluation methods, we introduce NaturalGAIA, a dynamic dataset currently comprising 276 constructed tasks. Its core innovation lies in separating underlying logical structure from linguistic representation via knowledge-driven causal paths. By integrating a naturalized narrative layer with a multi-level evaluation framework, NaturalGAIA rigorously assesses agent performance and isolates failure sources, ranging from intent parsing to execution planning. This design ensures a focused evaluation of core capabilities, including implicit intent understanding, long-horizon planning, and complex tool usage.

To overcome the execution challenges inherent

\*Equal contribution.

†Corresponding Authors.



Figure 1: Overview of the **LightManus-Jarvis** framework executing a task from **NaturalGAIA**. **Task Parsing:** The parser decomposes abstract user intents into a structured Task Topology composed of atomic tasks. **Workflow Management:** The manager dynamically schedules tasks across heterogeneous agents, employing Context Evolution to bridge information gaps between steps. **Execution & Evaluation:** Jarvis execute actions via hybrid perception, assessed by a hierarchical framework for success rates and error attribution.

in long-horizon tasks, we propose LightManus, a hierarchical collaborative framework. Adopting a "macro-planning-micro-execution" paradigm, the system operates on two levels: at the macro level, a workflow manager orchestrates task topologies and manages context evolution; at the micro level, Jarvis serves as an efficient execution kernel, achieving precise, coordinate-independent atomic actions via hybrid perception and chain-of-thought reasoning. This architecture effectively bridges the gap between high-level semantic dependencies and low-level operational barriers.

We evaluate our approach using NaturalGAIA. Experimental results demonstrate that the LightManus-Jarvis framework (driven by Claude-Sonnet-4.5) achieves a Weighted Pathway Success Rate (WPSR) of 45.6%, significantly outperforming mainstream baselines such as PC-Agent and Mobile-Agent-e, which score 13.1% and 21.1% respectively. Furthermore, our architecture improves operational efficiency by reducing token consumption by approximately 75% compared to existing GUI agents. These findings highlight the effectiveness of the hierarchical architecture in managing long-term context and precise execution, while also revealing the remaining challenges in handling

highly ambiguous naturalized narratives.

The main contributions of this paper can be summarized as follows:

- 1) Propose a new benchmark: NaturalGAIA, which separates logic and language by causal paths and narrative layers to evaluate the task performance of intelligent agents.
- 2) We design the LightManus-Jarvis layered architecture, integrating dynamic workflow scheduling and a high-precision GUI execution kernel.
- 3) We complete a comprehensive empirical evaluation, revealing the key shortcomings of current agents in complex tasks.

## 2 Related Work

GUI agent research is transitioning from static web parsing (Zhou et al., 2023; Deng et al., 2023) to dynamic OS control (Xie et al., 2024; Chai et al., 2025; Rawles et al., 2024), yet reconciling ecological validity with reproducibility remains a core challenge (Riddell et al., 2024; Fang et al., 2024; Lin et al., 2025; Zheng et al., 2025). Specifically, realistic benchmarks (Ye et al., 2025; Xu et al.,

2024) often lack deterministic verification (Drouin et al., 2024), relying on unstable or costly evaluations (Dai et al., 2025; Lù et al., 2025). Conversely, deterministic approaches (Sun et al., 2025; Valmeekam et al., 2023; Pan et al., 2025) tend to simplify tasks, failing to assess open-world planning (Dong et al., 2025; Kartik et al., 2025) or safety (Andriushchenko et al., 2025). regarding architecture, end-to-end LMMs (Wang et al., 2025a; Niu et al., 2024; Andreux et al., 2025) offer strong generalization but face computational constraints on mobile devices (Yang et al., 2025; Yin et al., 2025; Zhang et al., 2025b). Meanwhile, modular frameworks (He et al., 2024b; Li et al., 2025; Hong et al., 2024; Shang et al., 2025) improve interpretability but struggle with context drift and cross-modal alignment in long sequences (Huang et al., 2025; Mao et al., 2025; Zhang et al., 2025a; Cai et al., 2025). Moreover, integrating symbolic logic with neural planning (Choi et al., 2025; Wei et al., 2025) remains a significant unresolved bottleneck.

To address these challenges, we propose NaturalGAIA, which decouples logical pathways from linguistic narratives to enable verifiable evaluation under realistic ambiguity. We further introduce the LightManus-Jarvis architecture, which coordinates macro-level context evolution with micro-level hybrid perception to ensure both long-horizon planning consistency and precise on-device execution.

### 3 NaturalGAIA

NaturalGAIA evaluates robustness by decoupling logical structure from linguistic presentation. We employ a three-stage pipeline to instantiate interaction "Naturalness" across five dimensions—Long-horizon Chaining, Cross-domain Grounding, Cognitive Non-linearity, Informational Redundancy, and Contextual Dependency—unifying logical rigor with realistic ambiguity.

#### 3.1 Methodology for Building Naturalized Causal Pathways

As shown in the figure 2. To ensure scalability, we replace ad-hoc manual creation with a systematic process addressing the dimensions above.

**Path Generation and Atomic Task Definition (Long-horizon Chaining)** We formalize a Causal Path (CP) as a traversal sequence  $P = \{e_1 \xrightarrow{r_1} e_2 \dots \xrightarrow{r_n} e_{n+1}\}$  on a knowledge graph. Each step, retrieving  $e_{i+1}$  from  $e_i$  via  $r_i$ , is an

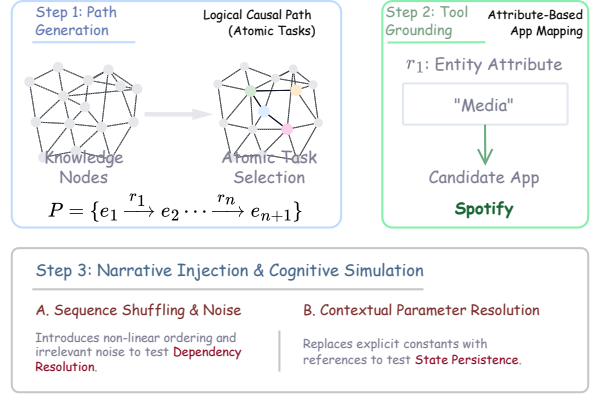


Figure 2: NaturalGAIA build process. This is a three-stage build process, namely Path Generation, Tool Grounding, and Narrative Injection & Cognitive Simulation.

*atomic task*—the smallest interaction unit—where  $e_{i+1}$  is the deterministic ground truth. By instantiating dependencies via structured knowledge graphs (e.g., Wikidata), we prevent overfitting and enable **long-horizon reasoning**, ensuring every task has a verifiable solution grounded in real-world data.

**Attribute-Based Tool Mapping (Cross-domain Grounding)** To address cross-domain scenarios, we map atomic tasks to applications based on target entity attributes. For instance, a task retrieving music metadata is mapped to candidate apps (e.g., Spotify) via the "media" attribute. This separates logical dependencies from interface implementations, ensuring NaturalGAIA evaluates core reasoning rather than interface memorization.

#### Narrative Injection and Cognitive Simulation

To simulate the remaining cognitive dimensions, we wrap rigid CPs in a natural language layer introducing two key challenges:

**Non-Linear Sequencing & Noise Filtering:** Human intent rarely adheres to a strict logical sequence. We simulate this by shuffling the sequence of atomic intents and injecting irrelevant noise. This compels the agent to filter noise from the disordered input and reconstruct the valid execution topology, thereby testing its capacity for *dependency resolution*.

**Context-Dependent Parameter Resolution:** We introduce referential ambiguity by replacing explicit constants with context-bound references. This necessitates dynamic context grounding: the agent must recognize that operation targets are anchored to prior execution outputs, requiring a persistent state to resolve abstract pointers into con-

crete parameters.

**Running Example:** Figure 3 illustrates our end-to-end pipeline. By extracting a deterministic causal path and wrapping it into a Naturalized Query that conceals intermediate nodes, we force the agent to implicitly resolve hidden entities to reach the final verifiable goal.

The specific building principles and processes for NaturalGAIA are given in the appendices A and B.

### 3.2 Multi-level Evaluation Framework

To precisely evaluate agent performance within NaturalGAIA and meticulously identify the root causes of errors, this study constructs a hierarchical evaluation framework. This framework enables comprehensive scrutiny from macroscopic, difficulty-weighted task completion down to microscopic error attribution.

#### Level 1: Weighted Pathway Success Rate

**(WPSR)** To account for task complexity, we introduce a difficulty score  $D_{j,i}$  for each task instance  $i$  of a Causal Pathway  $CP_j$ . This score is proportional to the number of nodes in the pathway and the number of distinct applications involved in its corresponding task. WPSR is defined by weighting each successful completion ( $S_{\text{task}}(j, i) = 1$ ) by its normalized difficulty  $w_{j,i} = D_{j,i} / \sum_{k,l} D_{k,l}$ , where the sum in the denominator is over all task instances across all pathways.

$$\text{WPSR} = \sum_{j,i} w_{j,i} \cdot S_{\text{task}}(j, i) \quad (1)$$

WPSR serves as a holistic metric for an agent’s final task completion capability, weighted by difficulty.

**Level 2: Fine-grained Traversal Metrics** To quantify partial progress and resilience, we introduce two complementary metrics that assess traversal quality.

**Mean Atomic Tasks Completion Ratio (MATCR):** This metric quantifies an agent’s ability to successfully complete sequences of atomic tasks. For each task sequence  $j$  within the benchmark, a completion ratio  $R_j = k_j / n_j$  is calculated, where  $k_j$  is the number of consecutive atomic tasks successfully executed from the start, and  $n_j$  is the total number of atomic tasks in sequence  $j$ .

$$\text{MATCR} = \frac{1}{N} \sum_{j=1}^N \frac{k_j}{n_j} \quad (2)$$

#### Positional-Weighted Atomic Tasks Success

**Rate (p-ATSR):** This metric evaluates an agent’s ability to maintain long-term coherence by assigning greater weight to successes in the later stages of a Causal Pathway. Let  $n_j$  be the total number of atomic tasks in pathway  $CP_j$ . We introduce a positional weight, denoted as  $p(i)$ , which is a monotonically increasing function of the step index  $i$ . The p-ATSR is then defined as:

$$\text{p-ATSR} = \frac{\sum_{j=1}^N \sum_{i=1}^{n_j} p(i) \cdot S_{\text{atomic}}(j, i)}{\sum_{j=1}^N \sum_{i=1}^{n_j} p(i)} \quad (3)$$

where  $S_{\text{atomic}}(j, i) = 1$  for a success at step  $i$  of pathway  $j$ , and 0 otherwise.

In summary, MATCR assesses an agent’s foundational reliability by quantifying its average execution length, whereas p-ATSR places greater emphasis on its long-term coherence by rewarding success in the later stages of a task.

#### Level 3: Error Attribution Analysis

For atomic tasks that fail, as identified by the Level 2 metrics, this stage involves analyzing the atomic operation sequence to attribute Execution Errors (EE) to one of three primary types:

*Knowledge Deficit (KD):* Lack of domain/procedural knowledge for the operation.

*Perceptual Error (PE):* Correct intent but failed information extraction from the UI.

*Operational Error (OE):* Correct perception but imprecise action execution.

### 3.3 Task Difficulty Stratification

To enable systematic evaluation, we stratify tasks into three levels based on the topological complexity of their underlying Causal Pathways, specifically considering path length (number of nodes) and cross-application dependencies.

**Level 1 (Basic):** Short pathways (1-2 nodes) confined to 1-2 applications, testing basic sequential execution.

**Level 2 (Intermediate):** Pathways of 3-4 nodes requiring transitions across 3-5 applications, challenging context maintenance.

**Level 3 (Advanced):** Long pathways (5-7 nodes) spanning up to 7 applications. These require advanced planning and memory to manage intricate inter-node dependencies.

Finally, we constructed 276 tasks. Detailed statistics and examples are provided in Appendix C and C.3.

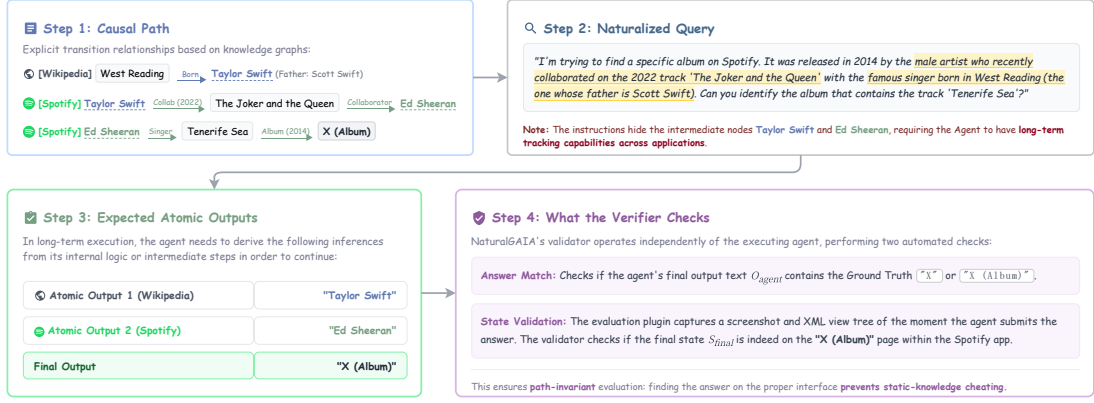


Figure 3: An end-to-end running example illustrating the NaturalGAIA pipeline: (i) Causal Path extraction, (ii) Naturalized Query generation simulating cognitive ambiguity, (iii) Expected Atomic Outputs required for progression, and (iv) What the Verifier Checks to ensure path-invariant evaluation.

## 4 The LightManus-Jarvis Collaborative Framework

To address the complex semantic dependencies and cross-platform barriers inherent in long-horizon tasks, we propose the collaborative LightManus-Jarvis architecture. This framework adopts a "macro-planning-micro-execution" paradigm: LightManus functions as the system's macro-hub responsible for global topology planning, cross-app scheduling, and long-term memory management; while Jarvis is designed as a specialized micro-kernel for high-precision instruction grounding on mobile devices.

### 4.1 LightManus: Macro-Scheduling and Memory-Driven Flow Management

LightManus aims to bridge the gap between abstract user intents and concrete device operations. It serves not only as an initial task decomposer but also as a dynamic scheduling hub, maintaining global context memory and adjusting subsequent plans in real-time based on subtask feedback.

#### Semantic Parsing and Topology Generation

The core of the perception layer is the Task Parser ( $\mathcal{P}_\theta$ ). Faced with long-horizon user intents  $I_{user}$ —which often imply complex logical structures such as strict sequential dependencies, parallel execution, or cross-application operations—we model  $\mathcal{P}_\theta$  as a constrained planning generator. Based on the capability library  $\mathbb{A}$ , it deconstructs unstructured  $I_{user}$  into a logically complete task topology  $\mathcal{T}$ :

$$\mathcal{T} = \mathcal{P}(I_{user}, \mathbb{A}) = \langle \tau_1, \tau_2, \dots, \tau_N \rangle \quad (4)$$

Each atomic task  $\tau_i$  is formalized as a quintuple  $\langle \mathcal{D}_i, \mathcal{C}_i, \alpha_i^*, \xi_i, \sigma_i \rangle$ , providing the downstream execution with the operation descriptor, context dependency slot, optimal executor route, runtime environment, and lifecycle state, respectively.

**Memory-Driven Context Evolution** At the execution layer, to address information fragmentation caused by heterogeneous environment isolation in long-term tasks, LightManus abandons static linear execution in favor of an "Execute-Perceive-Evolve" closed-loop strategy.

**Context Evolution Mechanism:** Traditional static instructions fail when subsequent steps depend heavily on the output of preceding ones (e.g., using prior search results as current query keywords). Therefore, we introduce a Context Evolution module  $\mathcal{E}_\phi$  that is tightly coupled with the system's *global memory*. Before scheduling task  $\tau_{i+1}$ , the system injects the execution feedback  $r_i$  from the preceding task  $\tau_i$  into the global memory and dynamically updates the original description of  $\tau_{i+1}$ , generating a semantically enhanced descriptor  $\mathcal{D}'_{i+1}$ :

$$\mathcal{D}'_{i+1} \leftarrow \mathcal{E}_\phi(\tau_{i+1}, \mathcal{D}_{raw}, r_i) \quad (5)$$

This mechanism enables implicit parameter passing, transforming unstructured natural language feedback from upstream into executable structured parameters downstream, effectively suppressing semantic drift during long-sequence execution. The final system response  $\Omega$  is a recursive aggregation process driven by real-time feedback, supplemented by a consistency check  $\mathcal{V}(r_t)$  to ensure evidence reliability.

## 4.2 Jarvis: High-Efficiency Android GUI Execution Agent

As the execution core on mobile devices, Jarvis is designed to address the common problems of **high inference latency** and severe **coordinate illusions** in existing pure-vision solutions. Through hybrid perception and structured mapping, Jarvis achieves efficient and robust single-agent execution.

**Hybrid Visual-Structural Perception** To overcome the limitations of pure vision while preserving visual semantics, the Observer module employs a hybrid strategy to construct the observation  $o_t$ :

$$o_t = \langle \mathcal{V}(I_t), \mathcal{T}(X_t) \rangle \quad (6)$$

Specifically,  $\mathcal{V}(I_t)$  processes the adaptively compressed screenshot  $I_t$  to capture unstructured visual semantics with low token overhead. Meanwhile, to eliminate coordinate illusions,  $\mathcal{T}(X_t)$  leverages the **Android Accessibility interface** to retrieve the underlying layout tree  $X_t$ , pruning it via a viewport semantic filter  $S_t = \text{Nodes}(X_t) \cap \mathcal{V}_{\text{vis}} \cap \mathcal{I}_{\text{int}}$ . The retained elements are mapped to quadruples (UID, Class, Text, Bounds), where the unique UID provides an unambiguous anchor for the LLM, significantly improving operation accuracy.

**Reasoning and Atomic Action Generation** The Agent Core is driven by a multimodal Large Language Model (LLM). The input prompt  $\mathcal{P}_t = \mathcal{I} \oplus h_{t-1} \oplus o_t$  integrates the context-enhanced instruction  $\mathcal{I}$ , historical trajectory memory  $h_{t-1}$ , and the current hybrid observation.

**Chain-of-Thought Decision Making:** Jarvis enforces an explicit Chain-of-Thought (CoT) mechanism. The policy function  $\pi_\theta$  first generates a natural language reasoning path  $th_t$ , analyzing the gap between the current interface state and the task objective, and then generates standardized action instructions  $a_t$ :

$$(th_t, a_t) \sim \pi_\theta(\cdot | \mathcal{P}_t) \quad (7)$$

**Atomic Action Space:** To ensure execution determinism, we map semantic actions  $a_t$  to underlying ADB control signals. The defined atomic action space  $\mathcal{A}$  includes:

- TAP( $u$ ): Precise clicking based on UID coordinate resolution, completely avoiding pixel prediction errors;
- INPUT( $u, \text{txt}$ ): A hybrid input strategy that prioritizes simulating a physical keyboard

to trigger predictive text, while reverting to broadcast injection for special characters;

- SWIPE/DRAG for gesture control and SYS( $k$ ) for system-level navigation.

In summary, Jarvis achieves a precise mapping from abstract instructions to physical signals through a closed loop of Perception  $\rightarrow$  Reasoning  $\rightarrow$  Action:

$$\begin{aligned} (\hat{th}, \hat{a}) &= \underset{(th,a)}{\operatorname{argmax}} P_\theta((th, a) | \mathcal{P}_t), \\ a_t^{\text{phy}} &= \mathcal{E}_{\text{act}}((\hat{th}, \hat{a}), \phi(X_t)) \end{aligned} \quad (8)$$

We present a complete execution trajectory in the Appendix D.

## 5 Experiment

Baseline	Env	Obs Space
PCA	Desktop	Scrn (Vision)
MAe	Android	Scrn (Vision)
UI-TARS	Android	Scrn + XML (Hybrid)
LightManus-MAe	Android	Scrn (Vision)
LightManus-Jarvis	Android	Scrn + XML (Hybrid)

Table 1: Comparison of observation spaces across baselines.

### 5.1 Setup

**Experimental Environment.** Experiments are executed on an Ubuntu 20.04 server. The Android testing environment is an Android Studio emulator (Android 11.0, 6 vCPUs, 8 GB RAM). The Windows environment is a VMware virtual machine (Windows 11, 8 vCPUs, 12 GB RAM).

**Baselines & Observation Spaces.** We compare LightManus with PC-Agent (PCA) (He et al., 2024b), Mobile-Agent-e (MAe) (Wang et al., 2025b), and UI-TARS (Qin et al., 2025). An ablation variant, LightManus-MAe, isolates the macro-planner’s contribution. As Table 1, our independent verifier strictly enforces a unified State Validation standard (capturing screenshots and XML) across all models to ensure an absolutely consistent evaluation.

**Foundation Models.** We test multiple MLLM backends for generalizability: closed-source models (Gemini 2.5/3 Pro/Flash, GPT-5.2, Claude-Sonnet-4.5) and open-source models (Qwen3-max, Qwen3-plus, and Qwen3.5-397b-a17b).

Method	Level-1				Level-2				Level-3				Overall			
	SR (P@1/4)	WPSR	MAT CR	ATSR	SR (P@1/4)	WPSR	MAT CR	ATSR	SR (P@1/4)	WPSR	MAT CR	ATSR	SR (P@1/4)	WPSR	MAT CR	ATSR
<b>PC-Agent</b>																
Gemini-2.5-Pro	40.0 / 66.7	42.9	60.8	59.1	10.0 / 40.0	10.0	43.1	32.0	0.0 / 20.0	7.6	24.9	15.9	20.0 / 45.7	13.1	45.5	25.7
<b>Mobile-Agent-e</b>																
Gemini-2.5-Pro	46.7 / <b>100.0</b>	58.9	74.2	68.9	10.0 / <u>60.0</u>	20.0	50.0	41.0	0.0 / 30.0	12.7	24.4	17.5	22.9 / 68.6	21.1	53.0	30.4
Gemini-2.5-flash	40.0 / 80.0	46.4	60.0	56.1	0.0 / 50.0	15.0	42.5	33.0	0.0 / 20.0	12.7	15.0	13.8	17.1 / 54.3	18.0	42.1	24.4
<b>LightManus_Mobile-Agent-e</b>																
Gemini-2.5-Pro	<u>73.3</u> / <b>100.0</b>	63.4	71.7	71.3	20.0 / <b>70.0</b>	27.5	53.1	44.5	10.0 / 40.0	20.3	29.9	25.2	40.0 / 74.3	28.3	54.4	36.3
Gemini-2.5-flash	53.3 / 80.0	63.4	74.2	71.3	0.0 / <u>60.0</u>	22.5	50.0	42.0	0.0 / 30.0	12.7	20.5	16.1	22.9 / 60.0	22.5	51.9	30.1
GPT-5.2	60.0 / -	57.1	66.7	61.0	30.0 / -	30.0	30.0	30.0	20.0 / -	18.6	20.0	17.6	40.0 / -	27.2	42.9	26.4
Claude-Sonnet-4.5	60.0 / -	57.1	60.0	56.1	<u>40.0</u> / -	40.0	57.5	49.0	20.0 / -	20.3	36.7	28.4	42.9 / -	31.1	52.6	37.7
<b>LightManus_Jarvis</b>																
Gemini-2.5-pro	<u>73.3</u> / <b>100.0</b>	<u>75.0</u>	78.3	77.4	<u>40.0</u> / <b>70.0</b>	42.5	<u>70.6</u>	63.0	20.0 / <u>50.0</u>	27.5	50.8	42.2	48.6 / <u>77.1</u>	38.3	<u>68.3</u>	52.4
Gemini-2.5-flash	66.7 / <u>93.3</u>	69.6	75.8	73.8	30.0 / <u>60.0</u>	37.5	65.6	57.0	10.0 / 30.0	20.3	32.5	27.5	40.0 / 65.7	32.0	60.5	41.5
Gemini-3.0-pro	<u>73.3</u> / <b>100.0</b>	<b>81.2</b>	83.3	<u>82.3</u>	<u>40.0</u> / <b>70.0</b>	45.0	<b>75.0</b>	<b>67.5</b>	<u>30.0</u> / <b>60.0</b>	34.7	<u>55.5</u>	<b>46.7</b>	<u>51.4</u> / <b>80.0</b>	44.1	<b>73.0</b>	<b>57.0</b>
Gemini-3.0-flash	<b>86.7</b> / <u>93.3</u>	<b>81.2</b>	<b>86.7</b>	<b>84.8</b>	30.0 / <b>70.0</b>	<u>47.5</u>	68.8	60.8	<u>30.0</u> / 40.0	27.1	40.1	32.2	<b>54.3</b> / 71.4	40.4	68.2	46.7
GPT-5.2	66.7 / -	64.3	76.7	70.7	<u>40.0</u> / -	40.0	40.0	40.0	<b>40.0</b> / -	<b>40.7</b>	40.0	41.2	<u>51.4</u> / -	43.7	55.7	44.3
Claude-Sonnet-4.5	66.7 / -	64.3	73.3	68.3	<b>50.0</b> / -	<b>50.0</b>	70.0	<u>65.0</u>	<b>40.0</b> / -	<u>39.0</u>	<b>56.7</b>	<u>45.6</u>	<b>54.3</b> / -	<b>45.6</b>	67.6	<u>53.9</u>
Qwen3-max	53.3 / 86.7	58.0	62.5	61.0	30.0 / <b>70.0</b>	37.5	39.4	38.5	10.0 / 30.0	20.3	20.0	20.6	34.3 / 65.7	30.5	43.8	30.6
Qwen3-vl-plus	46.7 / 73.3	45.5	55.8	53.7	20.0 / 40.0	27.5	43.8	38.0	10.0 / 30.0	12.7	18.9	15.2	28.6 / 51.4	21.5	41.8	26.4
Qwen3.5-397b-a17b	60.0 / <u>93.3</u>	59.8	64.2	62.8	30.0 / <u>60.0</u>	37.5	46.2	43.8	20.0 / 40.0	22.9	23.8	23.9	40.0 / 74.3	32.2	47.5	34.3
UI-TARS-1.5-7b	33.3 / 80.0	42.0	56.7	53.0	0.0 / 30.0	10.0	41.2	31.0	0.0 / 20.0	7.6	14.2	12.4	14.3 / 48.6	13.0	40.1	22.6

Table 2: Main results on the NaturalGAIA. SR (P@1/4) denotes Success Rate at Pass@1 and Pass@4. **Bold** indicates the best performance, and underline indicates the second best. Note that for Claude and GPT series, only Pass@1 is reported due to computational constraints.

Model	Token Usage			Duration (s)
	Input	Output	Total	
MAe	377,200	50,780	427,980	2015.4
LightManus-MAe	378,300	51,850	430,150	2046.9
LightManus-Jarvis	98,700	13,900	112,600	505.8

Table 3: Efficiency analysis on long-horizon Level-3 tasks using Gemini-3.0-Flash. The reported values represent the average metrics across 5 identical L3 tasks (averaging 5.4 atomic tasks).

## 5.2 Main Results

Table 2 presents the experimental results of different agent architectures and foundation models on NaturalGAIA. Overall, the proposed LightManus-Jarvis architecture significantly outperforms baseline methods across all metrics. The results not only validate the effectiveness of the architecture itself but also reveal complex interactions between model capabilities and task difficulty.

**Architecture Effectiveness: Asymmetry of Gains** Our empirical results strongly corroborate the superiority of the "macro-planning-micro-execution" paradigm. Using Gemini-2.5-Pro as the foundation model, merely introducing the LightManus scheduler boosts the WPSR of MAe

from 21.1% to 28.3%; further integrating the Jarvis executor yields a significant leap to 38.3%. This asymmetry indicates that optimizing planning alone fails to resolve perceptual hallucinations; Jarvis’s high-precision atomic operations are crucial for success.

Regarding long-horizon coherence (measured by p-ATSR), LightManus-Jarvis (52.4%) also significantly outperforms MAe (30.4%). A more compelling finding is that even with the lightweight Gemini-2.5-Flash, our architecture’s p-ATSR (41.5%) surpasses MAe equipped with a stronger foundation model. This proves that our architecture maintains long-sequence context consistency, effectively compensating for the limitations of weaker foundation models.

### Model Performance: Task-Driven Divergence

The performance of different models does not scale linearly with parameter size but exhibits a divergence trend highly correlated with task characteristics.

Lightweight models (e.g., Gemini3-Flash) demonstrate superior immediate instruction-following capabilities in Level-1 basic tasks (WPSR 81.2%), even outperforming the larger

Gemini-3.0-Pro. Our analysis suggests that for short-horizon, linear tasks, efficient intuitive instruction mapping is more effective than complex logical scrutiny, as the latter can sometimes lead to "over-interpretation" of simple directives.

However, when facing Level-3 complex tasks laden with cognitive non-linearity and cross-domain dependencies, models with strong logical reasoning capabilities (e.g., Claude-Sonnet-4.5 and Gemini-3.0-Pro) exhibit irreplaceable robustness. For instance, while Claude-Sonnet-4.5 performed typically in Level-1 due to hypersensitivity to instruction noise, its powerful logical deduction and noise filtering abilities enabled it to achieve the highest Mean Atomic Tasks Completion Ratio (MATCR 56.7%) in Level-3. By examining execution logs, we found that while some open-source models (e.g., Qwen3-Max) possess reliable single-step execution skills, their long-horizon logical coherence decays as steps increase, making them struggle with complex contextual dependencies. Similarly, while UI-TARS-1.5-7B performs adequately on Level-1, its Pass@1 drops to 0% on Level-3, directly exposing its vulnerabilities in long-horizon logic decay.

**Task Difficulty and Error Propagation** Performance declines non-linearly as task difficulty rises, validating the challenges introduced by the "Naturalness" of NaturalGAIA. For instance, Gemini-3.0-Pro's WPSR drops from 81.2% (Level-1) to 34.7% (Level-3). However, its relatively high MATCR (55.5%) in Level-3 reveals a disparity between local step execution and global task completion. We attribute this failure to two distinct factors consistent with our framework: (1) *Intent Parsing Failures*, where models struggle to decompose ambiguous, natural language instructions into correct planning topologies; and (2) *Cumulative Error Propagation*, where minor deviations in atomic operations accumulate over long horizons, causing the execution chain to fracture. This highlights that high success rates depend on both precise semantic disambiguation and robust error-correction mechanisms.

**Supplementary Experiments.** To validate generalizability, we evaluated LightManus-Jarvis on the independent AndroidWorld benchmark and compared it against Mind2Web on a pure-web subset (details in Appendix E).

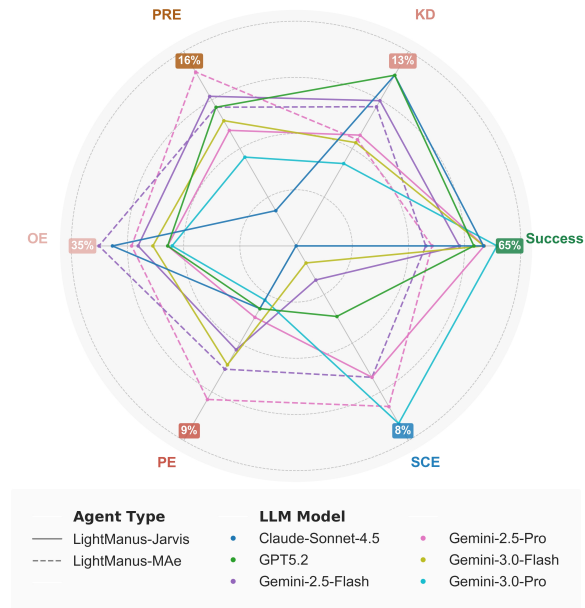


Figure 4: A comparative analysis of the failure reasons of different GUI agents and LLM backends. The chart categorizes errors into three main types: Knowledge Deficit (KD), Planning & Reasoning Errors (PRE), Operational Errors (OE), Perceptual Errors (PE), and Structural Compliance Errors (SCE)

### 5.3 Error analysis

To analyze the bottlenecks limiting agent performance, we categorize failure cases into five types: Knowledge Deficit (KD), Planning & Reasoning Errors (PRE), Operational Errors (OE), Perceptual Errors (PE), and Structural Compliance Errors (SCE). Figure 4 and the detailed statistics reveal three key observations. We have provided specific examples of several errors in the Appendix F.

#### Execution Remains the Primary Bottleneck

Across all architectures, OE consistently constitutes the primary failure mode, ranging from 19% to 30% even in advanced models like Claude-Sonnet-4.5. This bottleneck stems not merely from layout variability, but from the dynamic instability of mobile environments. First, high-density GUI contexts often induce visual hallucinations regarding element coordinates. Second, a temporal mismatch exists between reasoning and execution: the inherent latency of LLM inference allows interface states to drift (e.g., sudden pop-ups) before the action is applied. Furthermore, reliance on discrete static observations rather than real-time monitoring leaves agents vulnerable to transient state changes, resulting in precise actions being executed on obsolete states.

### **Efficacy of the Specialized Execution Module**

Comparing LightManus-MAe with LightManus-Jarvis validates the design of our Jarvis micro-kernel. For Gemini-2.5-Pro, upgrading to the Jarvis architecture reduced PE from 7.1% to 3.3% and OE from 25.6% to 20.0%. This reduction demonstrates that the hybrid visual-structural perception mechanism in Jarvis effectively mitigates "coordinate illusions" and improves the accuracy of element localization, confirming that a specialized execution agent is superior to general-purpose visual execution.

**The Reasoning-Execution Trade-off** We observe a distinct divergence between abstract planning and concrete execution. Claude-Sonnet-4.5, despite achieving minimal PRE (2.9%), struggles with high OE and KD (11.4%). In contrast, the Gemini-3.0 series maintains a more robust balance. Crucially, we find that KD is inversely correlated with model scale. Larger models leverage extensive parametric memory to recall diverse app-specific workflows, evidenced by the lower KD in Gemini-3.0-Pro (5.5%) compared to its Flash counterpart (6.9%). This suggests that robust agent performance relies on the synergy of two distinct capabilities: logical reasoning for topological planning and scale-driven domain knowledge for procedural grounding.

### **5.4 Efficiency Analysis**

Beyond accuracy metrics, we evaluate the operational efficiency of the proposed framework. As presented in Table 3, when handling the most complex long-sequence Level-3 tasks, Jarvis demonstrates superior efficiency compared to the GUI agent MAe. Specifically, Jarvis reduces total token consumption by approximately 74% (112,600 vs. 427,980) and shortens task duration by 75% (505.8s vs. 2015.4s). Furthermore, compared to MAe, the ablation variant LightManus-MAe introduces negligible overhead (+31.5s, +1.1k tokens), proving that our macro-planner provides global context management with extremely low computational cost. By replacing redundant visual encoding with a lightweight hybrid perception mechanism, our approach significantly minimizes inference latency and deployment costs.

## **6 Conclusion and Discussion**

This study highlights the "Evaluation-Realism Dilemma" constraining GUI agent research. By

introducing NaturalGAIA, we provide a methodology to decouple logical rigidity from linguistic naturalness. Furthermore, the LightManus-Jarvis framework validates the efficacy of the "macro-planning-micro-execution" paradigm in mitigating semantic drift and execution noise. Experiments demonstrate that this hierarchical approach significantly enhances robustness and efficiency compared to existing baselines. We hope this work offers valuable insights into the reasoning-execution gap and serves as a foundation for developing more robust, ecologically valid autonomous systems.

### **Limitations**

Our work entails limitations in both benchmark construction and agent efficacy. First, the pursuit of verifiable determinism constrains task diversity. To ensure rigorous ground truth, we exclude dynamic or irreversible scenarios (e.g., e-commerce, real-time messaging), which consequently narrows the scope of supported applications compared to open-ended environments. Second, scalability is bottlenecked by manual verification. Although task generation is automated, validating the topological logic currently relies on human experts, as existing agents lack the capability for reliable automated quality assurance. Finally, regarding the agent, inference latency and cumulative error propagation remain unresolved. The temporal mismatch between MLLM inference and dynamic interface changes can lead to state drift, and minor execution deviations in long-horizon tasks still pose risks of cascading failure.

### **Acknowledgments**

This work was supported by the Brain Science and Brain-like Intelligence Technology-National Science and Technology Major Project under Grant 2022ZD0208900, the National Natural Science Foundation of China under Grant 52308250, and the Guangdong Basic and Applied Basic Research Foundation under grant 2026A1515012965.

### **References**

Mathieu Andreux, Märt Bakler, Yanael Barbier, Hamza Benchekroun, Emilien Biré, Antoine Bonnet, Riaz Bordie, Nathan Bout, Matthias Brunel, Aleix Cambay, Pierre-Louis Cedoz, Antoine Chassang, Gautier Cloix, Ethan Connelly, Alexandra Constantinou, Ramzi De Coster, Hubert de la Jonquiere, Au-

- rélien Delfosse, Maxime Delpit, and 34 others. 2025. [Surfer 2: The next generation of cross-platform computer use agents](#). *Preprint*, arXiv:2510.19949.
- Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Eric Winsor, Jerome Wynne, Yarin Gal, and Xander Davies. 2025. [Agentharm: A benchmark for measuring harmfulness of llm agents](#). *Preprint*, arXiv:2410.09024.
- Zhicheng Cai, Xinyuan Guo, Yu Pei, Jiangtao Feng, Jinsong Su, Jiangjie Chen, Ya-Qin Zhang, Wei-Ying Ma, Mingxuan Wang, and Hao Zhou. 2025. [Flex: Continuous agent evolution via forward learning from experience](#). *Preprint*, arXiv:2511.06449.
- Yuxiang Chai, Hanhao Li, Jiayu Zhang, Liang Liu, Guangyi Liu, Guozhi Wang, Shuai Ren, Siyuan Huang, and Hongsheng Li. 2025. [A3: Android agent arena for mobile gui agents](#). *Preprint*, arXiv:2501.01149.
- Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. [SeeClick: Harnessing GUI grounding for advanced visual GUI agents](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9313–9332, Bangkok, Thailand. Association for Computational Linguistics.
- Wonje Choi, Jinwoo Park, Sanghyun Ahn, Daehee Lee, and Honguk Woo. 2025. [Nesyc: A neuro-symbolic continual learner for complex embodied tasks in open domains](#). *Preprint*, arXiv:2503.00870.
- Yutong Dai, Krithika Ramakrishnan, Jing Gu, Matthew Fernandez, Yanqi Luo, Viraj Prabhu, Zhenyu Hu, Silvio Savarese, Caiming Xiong, Zeyuan Chen, and Ran Xu. 2025. [Scuba: Salesforce computer use benchmark](#). *Preprint*, arXiv:2509.26506.
- DeepSeek-AI and 1 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. [Mind2web: Towards a generalist agent for the web](#). *Preprint*, arXiv:2306.06070.
- Lingzhong Dong, Ziqi Zhou, Shuaibo Yang, Haiyue Sheng, Pengzhou Cheng, Zongru Wu, Zheng Wu, Gongshen Liu, and Zhuosheng Zhang. 2025. [Say one thing, do another? diagnosing reasoning-execution gaps in vlm-powered mobile-use agents](#). *Preprint*, arXiv:2510.02204.
- Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. 2024. [Workarena: How capable are web agents at solving common knowledge work tasks?](#) *Preprint*, arXiv:2403.07718.
- Qihang Fang, Chengcheng Tang, Shugao Ma, and Yan-chao Yang. 2024. [Bid: Boundary-interior decoding for unsupervised temporal action localization pre-trainin](#). *Preprint*, arXiv:2403.07354.
- Ariel Gera, Odellia Boni, Yotam Perlit, Roy Bar-Haim, Lilach Eden, and Asaf Yehudai. 2025. [JuStRank: Benchmarking LLM judges for system ranking](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 682–712, Vienna, Austria. Association for Computational Linguistics.
- Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024a. [WebVoyager: Building an end-to-end web agent with large multimodal models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890, Bangkok, Thailand. Association for Computational Linguistics.
- Yanheng He, Jiahe Jin, and Pengfei Liu. 2025. [Efficient agent training for computer use](#). *Preprint*, arXiv:2505.13909.
- Yanheng He, Jiahe Jin, Shijie Xia, Jiadi Su, Runze Fan, Haoyang Zou, Xiangkun Hu, and Pengfei Liu. 2024b. [Pc agent: While you sleep, ai works - a cognitive journey into digital world](#). *arXiv preprint arXiv:2412.17589*.
- Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. [Metagtpt: Meta programming for a multi-agent collaborative framework](#). *Preprint*, arXiv:2308.00352.
- Kung-Hsiang Huang, Haoyi Qiu, Yutong Dai, Caiming Xiong, and Chien-Sheng Wu. 2025. [Gui-kv: Efficient gui agents via kv cache with spatio-temporal awareness](#). *Preprint*, arXiv:2510.00536.
- NVJK Kartik, Garvit Sapra, Rishav Hada, and Nikhil Pareek. 2025. [Agentcompass: Towards reliable evaluation of agentic workflows in production](#). *Preprint*, arXiv:2509.14647.
- Ziyue Li, Yuan Chang, Gaihong Yu, and Xiaoqiu Le. 2025. [Hiplan: Hierarchical planning for llm-based agents with adaptive global-local guidance](#). *Preprint*, arXiv:2508.19076.
- Haojia Lin, Xiaoyu Tan, Yulei Qin, Zihan Xu, Yuchen Shi, Zongyi Li, Gang Li, Shaofei Cai, Siqi Cai, Chaoyou Fu, Ke Li, and Xing Sun. 2025. [Cuareward-bench: A benchmark for evaluating reward models on computer-using agent](#). *Preprint*, arXiv:2510.18596.
- Jiyuan Liu, Jielin Song, Yunhe Pang, Zhiyu Shen, and Yanghui Rao. 2025. [CARE: A disagreement detection framework with concept alignment and reasoning enhancement](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language*

- Processing*, pages 13275–13290, Suzhou, China. Association for Computational Linguistics.
- Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, Arkil Patel, Dongchan Shin, Alejandra Zambrano, Karolina Stańczak, Peter Shaw, Christopher J. Pal, and Siva Reddy. 2025. [Agentrewardbench: Evaluating automatic evaluations of web agent trajectories](#). *Preprint*, arXiv:2504.08942.
- Yansheng Mao, Yufei Xu, Jiaqi Li, Fanxu Meng, Haotong Yang, Zilong Zheng, Xiyuan Wang, and Muhan Zhang. 2025. [Lift: Improving long context understanding of large language models through long input fine-tuning](#). *Preprint*, arXiv:2502.14644.
- Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. 2025. [Evaluation and benchmarking of llm agents: A survey](#). In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6129–6139, New York, NY, USA. Association for Computing Machinery.
- Runliang Niu, Jindong Li, Shiqi Wang, Yali Fu, Xiyu Hu, Xueyuan Leng, He Kong, Yi Chang, and Qi Wang. 2024. [Screenagent: A vision language model-driven computer control agent](#).
- OpenAI and 1 others. 2024. [Openai o1 system card](#). *Preprint*, arXiv:2412.16720.
- Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Hogan. 2025. [Spider: Scalable physics-informed dexterous retargeting](#). *Preprint*, arXiv:2511.09484.
- Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, and 16 others. 2025. [Ui-tars: Pioneering automated gui interaction with native agents](#). *Preprint*, arXiv:2501.12326.
- Qwen and 1 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Christopher Rawles, Sarah Clinckemahillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. 2024. [Androidworld: A dynamic benchmarking environment for autonomous agents](#). *Preprint*, arXiv:2405.14573.
- Martin Riddell, Ansong Ni, and Arman Cohan. 2024. [Quantifying contamination in evaluating code generation capabilities of language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14116–14137, Bangkok, Thailand. Association for Computational Linguistics.
- Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. 2025. [Agentsquare: Automatic llm agent search in modular design space](#). *Preprint*, arXiv:2410.06153.
- Qiushi Sun, Zhoumianze Liu, Chang Ma, Zichen Ding, Fangzhi Xu, Zhangyue Yin, Haiteng Zhao, Zhenyu Wu, Kanzhi Cheng, Zhaoyang Liu, Jianing Wang, Qintong Li, Xiangru Tang, Tianbao Xie, Xiachong Feng, Xiang Li, Ben Kao, Wenhai Wang, Biqing Qi, and 2 others. 2025. [Scienceboard: Evaluating multimodal autonomous agents in realistic scientific workflows](#). *Preprint*, arXiv:2505.19897.
- Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. [Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change](#). *Preprint*, arXiv:2206.10498.
- Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Juntong Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, Wanjun Zhong, Yining Ye, Yujia Qin, Yuwen Xiong, Yuxin Song, Zhiyong Wu, Aoyan Li, Bo Li, Chen Dun, and 93 others. 2025a. [Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning](#). *Preprint*, arXiv:2509.02544.
- Zhenhailong Wang, Haiyang Xu, Junyang Wang, Xi Zhang, Ming Yan, Ji Zhang, Fei Huang, and Heng Ji. 2025b. [Mobile-agent-e: Self-evolving mobile assistant for complex tasks](#). *arXiv preprint arXiv:2501.11733*.
- Hui Wei, Zihao Zhang, Shenghua He, Tian Xia, Shijia Pan, and Fei Liu. 2025. [PlanGenLLMs: A modern survey of LLM planning capabilities](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19497–19521, Vienna, Austria. Association for Computational Linguistics.
- Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. [Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments](#). *Preprint*, arXiv:2404.07972.
- Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. 2024. [Agenttrek: Agent trajectory synthesis via guiding replay with web tutorials](#).
- Zhen Yang, Zi-Yi Dou, Di Feng, Forrest Huang, Anh Nguyen, Keen You, Omar Attia, Yuhao Yang, Michael Feng, Haotian Zhang, Ram Ramrakhya, Chao Jia, Jeffrey Nichols, Alexander Toshev, Yinfei Yang, and Zhe Gan. 2025. [Ferret-ui lite: Lessons from building small on-device gui agents](#). *Preprint*, arXiv:2509.26539.

Suyu Ye, Haojun Shi, Darren Shih, Hyokun Yun, Tanya Roosta, and Tianmin Shu. 2025. [Realwebassist: A benchmark for long-horizon web assistance with real-world users](#). *Preprint*, arXiv:2504.10445.

Zecheng Yin, Hao Zhao, and Zhen Li. 2025. [Hypernav: Hybrid perception for object-oriented navigation in unknown environment](#). *Preprint*, arXiv:2510.22917.

Chi Zhang, Zhao Yang, Jiakuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2023. [Appagent: Multimodal agents as smartphone users](#). *Preprint*, arXiv:2312.13771.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. 2025a. [Agentic context engineering: Evolving contexts for self-improving language models](#). *Preprint*, arXiv:2510.04618.

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, and 6 others. 2025b. [AgentCPM-GUI: Building mobile-use agents with reinforcement fine-tuning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 155–180, Suzhou, China. Association for Computational Linguistics.

Zihan Zheng, Tianle Cui, Chuwen Xie, Jiahui Pan, Qianglong Chen, and Lewei He. 2025. [PlanningArena: A modular benchmark for multidimensional evaluation of planning and tool learning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31047–31086, Vienna, Austria. Association for Computational Linguistics.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, and 1 others. 2023. [Webarena: A realistic web environment for building autonomous agents](#). *arXiv preprint arXiv:2307.13854*.

## A Design Principles for NaturalGAIA

To ensure the benchmark rigorously evaluates agentic capabilities in realistic scenarios, we formulate a set of strict design principles governing the generation of high-complexity tasks (referred to as Level-3 complexity). These principles enforce structural integrity, logical determinism, and ecological validity.

### I. Structural Complexity and Cross-Domain Scope

- **Long-Horizon Composition:** To challenge the agent’s context maintenance capabilities, each main task must be composed of a sequence of atomic interactions (typically  $N \geq 6$ ). These atomic tasks are explicitly delimited in the logical skeleton, preventing ambiguous task boundaries.

- **Multi-App Collaboration:** The task execution must necessitate context switching between distinct application environments. We enforce a minimum diversity constraint (e.g., interacting with at least three distinct applications), ensuring the agent is evaluated on cross-domain grounding rather than single-app proficiency.

## II. Determinism of Atomic Outcomes

- **Explicit Textual Output:** Every atomic task must yield a verifiable result presentable in text format. This enables automated evaluation of intermediate steps.

- **Uniqueness and Objectivity:** The output space of each atomic task must be constrained to a single, unambiguous ground truth (Deterministic Causality). Ambiguous queries yielding multiple valid answers are strictly pruned during the generation phase.

- **Temporal Stability and Freeze-Frame Strategy:** To ensure reproducibility, tasks prioritize time-invariant knowledge (e.g., historical data, physical constants). For scenarios inevitably involving time-sensitive information (e.g., "current stock price"), we employ a **Freeze-Frame Strategy**: the information is solidified at the moment of capture into a local static snapshot (e.g., a cached page or screenshot). The ground truth is then validated against this static context rather than the live web, eliminating drift caused by temporal latency.

## III. Logical Dependency Chain

- **Strict State Dependency:** The task structure enforces a rigid sequential dependency, where the output  $O_t$  of the atomic task at step  $t$  serves as a mandatory input parameter for step  $t + 1$ . This creates an unbreakable "information flow."

- **Chain Irreducibility:** The dependency chain is designed such that a failure in any preceding node renders subsequent steps theoretically executing impossible. This "Fail-Stop" mechanism ensures that agents cannot bypass reasoning steps through hallucination or guessing.

#### IV. Grounding Specification

- **App-Specific Designation:** Each atomic task is explicitly mapped to a specific target application within the defined schema. The benchmark rejects ambiguous instructions that do not imply a clear tool selection.
- **Verifiable Execution Path:** Alongside the final answer, the benchmark records the canonical operation path (the sequence of edges traversed in the knowledge graph) as the Golden Path. This facilitates granular error analysis beyond simple success/failure metrics.

#### V. Ecological Validity of Tools

- **Popularity-Based Selection:** To ensure the benchmark reflects real-world utility, the selection of applications follows the principle of ecological validity. We prioritize applications with high market penetration (e.g., top-ranked apps in global stores<sup>1</sup>), ensuring the agent learns to interact with interfaces relevant to the general user population.

### B Details of Task Creation Pipeline

The construction of NaturalGAIA is grounded in the semantic structure of Wikidata, utilizing its RDF (Resource Description Framework) triples to instantiate a realistic GUI environment. We formalize the pipeline into six stages: Semantic Substrate Analysis, Information Space Definition, Subgraph Construction, Causal Skeleton Generation, Natural Language Mapping, Human Verification Protocol, and a Case Study.

#### B.1 Data Structure and Semantic Substrate

We select Wikidata as the semantic substrate for our benchmark. Structurally, Wikidata is a labeled multidigraph where the fundamental data unit is an RDF triple  $(s, p, o)$ , denoting a subject  $s$  linked to an object  $o$  via a predicate  $p$ . This structure

<sup>1</sup>Data reference: Google Play Store rankings via <https://play.google.com/store>.

provides two methodological foundations for task generation:

1. **Deterministic Causality:** Unlike unstructured text, RDF triples provide unambiguous logical transitions. For instance, if a task requires identifying a "director," the predicate  $P57$  provides a deterministic edge from a movie entity to a person entity, eliminating ambiguity in logical reasoning.
2. **Ontological Isomorphism:** The class hierarchy in Wikidata (defined via  $P31$  *instance of* and  $P279$  *subclass of*) mirrors real-world taxonomies. This allows us to project the global graph into specific domain contexts using strict ontological rules rather than arbitrary node connections.

#### B.2 Definition of Application Information Spaces

To simulate a multi-application environment, we define an **Information Space** ( $\mathcal{I}_{app}$ ) for each application. An information space is not merely a collection of entities but a constrained subgraph defined by a tuple  $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ :

- **Root Classes ( $\mathcal{T}$ ):** The set of entity types allowed within the application (corresponding to `type_filter` in configuration). For example, a music app includes classes such as "Musical Recording" and "Artist".
- **Navigable Relations ( $\mathcal{R}$ ):** The specific Wikidata predicates that function as interactive transitions within the GUI (corresponding to `actions`). An edge in the graph is instantiated as a clickable link only if its predicate belongs to  $\mathcal{R}$ .
- **Attribute Space ( $\mathcal{A}$ ):** The set of properties visible on an entity's profile page (corresponding to `constraints`). These attributes serve as metadata for filtering and verification.

Figure 5 illustrates a configuration snippet for the "Spotify" application. The action `view_artist` maps the predicate  $P175$  to a transition from a *Track* to an *Artist*, while properties like  $P577$  (publication date) constitute the observable attribute space.

A critical property of our design is that the information space of any vertical application is a proper subset of the general knowledge space

```

"Spotify": {
  "description": "Music streaming
  service...",
  "entities": {
    "Track": {
      "type_filter": ["Q218818", "Q7366"
      ],
      // Root Classes: Musical Recording
      , Song
      "constraints": ["P175", "P577", "P
      136"],
      // Attribute Space: Performer,
      Date, Genre
      "actions": {
        "view_artist": {
          "target": "Artist",
          "relation": "P175",
          // Navigable Relation: P175 (
          Performer)
          "intent": "find the performing
          artist"
        }
      }
    }
  }
}

```

Figure 5: Configuration snippet defining the Information Space for the Spotify application.

(e.g., Wikipedia), denoted as  $\mathcal{I}_{vertical} \subset \mathcal{I}_{wiki}$ . This ensures that Wikipedia acts as a connectivity guarantor: when no direct path exists between atomic tasks in vertical apps, the agent can utilize Wikipedia as a hub to bridge disconnected components, enabling long-horizon task continuity.

### B.3 Subgraph Construction with Type Inference

We employ a seed-based expansion strategy to construct the evaluation subgraph. To ensure adherence to the defined schema, we implement a deep type inference mechanism during traversal. Instead of retrieving all neighbors, the crawler verifies the class hierarchy of each neighbor using the SPARQL property path `wdt:P31/wdt:P279*`. A node is included in the subgraph only if its ancestral classes intersect with the target application’s Root Classes ( $\mathcal{T}$ ). This filters out ontological noise and ensures logical consistency within the generated environment.

### B.4 Causal Skeleton Generation and Cardinality Constraints

We generate the logical skeleton of a task via constrained random walks on the constructed subgraph. A step in the causal path is defined as  $S_t = (v_p, e, v'_p, \mathbb{C})$ , generated as follows:

1. **Primary Node Transition** ( $v_p \rightarrow v'_p$ ): The primary node represents the focal entity of the current interface (e.g., an Album). The system selects an edge  $e$  based on the application’s navigable relations  $\mathcal{R}$  to transition to the next primary node  $v'_p$  (e.g., the Artist).
2. **Application Switching**: The predicate of edge  $e$  determines the next active application. For instance, selecting  $P175$  activates the music application, whereas  $P57$  activates the movie database.
3. **Cardinality Constraints and Secondary Nodes** ( $\mathbb{C}$ ): Relations in Wikidata are often one-to-many (e.g., an artist has multiple albums). To uniquely identify the target node  $v'_p$ , the system retrieves distinctive attributes from the attribute space  $\mathcal{A}$  of  $v'_p$  (e.g., release year, genre). These attributes form the set of Secondary Nodes  $\mathbb{C}$ .

In the final task, these Secondary Nodes serve as discriminative constraints (e.g., "the pop album released in 2014"). The agent must utilize these visible attributes to filter candidates and resolve branching ambiguities.

## B.5 Natural Language Mapping and Cognitive Simulation

The logical skeleton is transformed into a natural language instruction using a Large Language Model (LLM). This process follows a "Hide-and-Describe" principle to simulate cognitive challenges:

- **Hiding Intermediate Nodes**: Intermediate primary nodes in the path are redacted from the instruction. The agent is provided only with the start node and the final intent, requiring it to infer the traversal path.
- **Constraint Description**: The discriminative constraints ( $\mathbb{C}$ ) selected in the previous stage are converted into natural language descriptors. For example, the structured constraint ( $P577 : 2014$ ) is rewritten as the relative clause "released in 2014".
- **Narrative Injection**: To simulate diverse user intents, the logical conditions are wrapped in varying narrative templates (e.g., "Exploration," "Vague Recollection"), increasing the pragmatic complexity of the instruction.

## B.6 Human Verification Protocol

While the pipeline described above is automated, ensuring the fidelity of the benchmark requires a Human-in-the-Loop (HITL) verification stage. We employ a rigorous two-step validation protocol for the generated tasks:

**Step 1: Solvability Check** Human annotators act as the agents. They are provided with the initial instruction and the environment constraints (the specific versions of the Apps). Annotators must attempt to solve the task following the generated logical path. Any task where the ground truth cannot be deterministically reached—due to interface updates, data obsolescence, or graph ambiguity—is flagged and discarded.

**Step 2: Consistency Verification** Reviewers inspect the alignment between the generated *Natural Language Query* (from Section A.5) and the *Logical Skeleton* (from Section A.4). They verify that:

- The natural language query does not accidentally leak the hidden intermediate entities (Spoiler Check).
- The query contains sufficient linguistic cues (e.g., accurate descriptions of constraints) to logically deduce the next step without ambiguity.

Only tasks that pass both solvability and consistency checks are included in the final NaturalGAIA release.

## B.7 Case Study: From Graph to Natural Instruction

We illustrate the pipeline with a generated example.

**1. Logical Skeleton Generation** The generator performs a random walk involving Wikipedia and Spotify:

- **Step 1 (Wikipedia): Place**  $\rightarrow$  **Person**. Transition: *West Reading*  $\xrightarrow{P_{19}^{-1}}$  *Taylor Swift*. Constraint ( $\mathbb{C}_1$ ): Father is *Scott Swift*; Occupation is *Singer-songwriter*.
- **Step 2 (Spotify): Person**  $\rightarrow$  **Track**. Transition: *Taylor Swift*  $\xrightarrow{P_{175}^{-1}}$  *The Joker and the Queen*. Constraint ( $\mathbb{C}_2$ ): Released in 2022; Co-artist is *Ed Sheeran*.
- **Step 3 (Spotify): Track**  $\rightarrow$  **Co-Artist**. Transition: *The Joker and the Queen*  $\xrightarrow{P_{175}}$  *Ed*

*Sheeran*. Constraint ( $\mathbb{C}_3$ ): Genre includes *Pop music*.

- **Step 4 & 5 (Spotify): Artist**  $\rightarrow$  **Track**  $\rightarrow$  **Album (Goal)**. Transition: *Ed Sheeran*  $\rightarrow$  *Tenerife Sea*  $\rightarrow$  *X (Album)*. Constraint ( $\mathbb{C}_5$ ): Released in 2014; Label is *Asylum Records*.

**2. Naturalization** The LLM converts the skeleton into a user query, hiding intermediate entities like "Taylor Swift" and "Ed Sheeran":

"I'm trying to find a specific album on Spotify. It was released in **2014** by the male artist who recently collaborated on the 2022 track 'The Joker and the Queen' with the famous singer born in **West Reading** (the one whose father is **Scott Swift**). Can you identify the album that contains the track '**Tenerife Sea**'?"

**3. Resolution Logic** The agent must resolve the dependency chain: (1) Identify *Taylor Swift* via the "West Reading/Scott Swift" constraint on Wikipedia; (2) Locate the 2022 collaboration to identify *Ed Sheeran*; (3) Navigate to Sheeran's profile to find the 2014 album containing "Tenerife Sea". This demonstrates the benchmark's ability to create rigorous long-horizon tasks requiring cross-app information retrieval and state tracking.

## C Details of Data Statistics and Environment

### C.1 Task Complexity Distribution

Table 4 presents the statistical characteristics of the NaturalGAIA dataset across varying difficulty levels. The dataset is stratified into three levels based on the length of the causal chain. For each stratum, we quantify the average count (Avg) and distribution range of both *Tools Involved* and *Atomic Tasks*. The data exhibits a strong positive correlation between difficulty level and structural complexity: Level-3 tasks require, on average, interaction with 4.4 distinct applications and a sequence of 5.7 atomic operations. This distribution confirms that NaturalGAIA successfully captures a wide spectrum of complexity, from simple single-step retrievals to long-horizon, cross-domain reasoning chains.

### C.2 Application Ecosystem and Information Spaces

To ensure ecological validity and broad coverage, NaturalGAIA integrates a diverse ecosystem of

Table 4: Statistical Distribution of Task Complexity in NaturalGAIA

Difficulty Level	Count	Unique Tools		Atomic Tasks	
		Avg	Range	Avg	Range
Level 1 (Simple)	133	1.5	1–2	1.9	1–2
Level 2 (Medium)	92	2.3	2–4	3.7	3–4
Level 3 (Hard)	51	4.4	3–7	5.7	3–7
<b>Overall</b>	<b>276</b>	<b>2.6</b>	<b>1–7</b>	<b>3.1</b>	<b>1–7</b>

**25 applications**, prioritizing platforms that host Immutable Historical Records or Fixed Physical Attributes. As detailed in Table 5, the selection strictly adheres to the "*Determinism Principle*": apps like *FlightAware* (Historical Flights) and *Steam* (Software Specs) provide indisputable verification sources that are immune to temporal drift.

### C.3 Task example of NaturalGAIA

Figure 6 presents concrete examples of tasks corresponding to the three difficulty levels. These examples illustrate the increasing requirements in terms of the number of atomic tasks, the diversity of applications involved, and the complexity of planning and execution across the levels.

### D Example of task execution trajectory

As shown in the fig10 and code1, we demonstrate the complete execution trajectory of a GUI execution task that consists of 3 atomic tasks.

## E Systematic Benchmark Comparison

To clearly demonstrate NaturalGAIA’s unique niche, Table 6 provides a systematic comparison with current mainstream benchmarks from multiple dimensions. Compared to existing works like AndroidWorld and Mind2Web, NaturalGAIA’s core advantage lies in providing fine-grained traversal metrics (MATCR, ATSR) to measure the agent’s intermediate inference progress. Furthermore, by employing logico-linguistic decoupling and causal paths, it maintains an infinite number of dynamic data instances while ensuring strict state verifiability (combining device state with text match), effectively preventing data contamination.

### E.1 Generalization on External Benchmarks (AndroidWorld)

To dispel concerns regarding the framework’s feasibility and generalizability, we deployed

LightManus-Jarvis on AndroidWorld, a completely independent and dynamic verifiable benchmark. As shown in Table 7, under a unified benchmark, our system equipped with Qwen3.5-397b-a17b achieved a success rate of 63.3%, significantly outperforming the standard MAE. This confirms that the core contribution of our system lies in the decoupling of macro-planning and micro-execution, rather than simply relying on the design of the underlying accessibility tree.

### E.2 Comparative Analysis with Web-Specific Frameworks (Mind2Web)

Mind2Web is primarily designed for static Web DOM tree parsing. To ensure a fair comparison, we extracted a subset of web tasks from NaturalGAIA that can be completed using only a browser and tested them using the Mind2Web framework with the same Qwen3.5 model. As shown in Table 8, LightManus-Jarvis significantly outperforms Mind2Web (Overall SR 31.4% vs 22.9%). This performance gap stems from two fundamental differences: (1) Jarvis’s hybrid vision-structure awareness effectively avoids element positioning failures caused by relying solely on dynamic, non-standard DOM trees; (2) The LightManus macro-planner mitigates semantic drift in long-sequence execution through Context Evolution, enabling it to maintain higher path completion quality in Level-2 and Level-3 tasks compared to Mind2Web’s static memory.

## F Error Type Examples

### Planning & Reasoning Errors (PRE)

**Original Task:** "I’m trying to settle a debate about a film score. I need to identify a specific movie where a certain musician served as the music director in 2019. I’ve totally forgotten his name, but I know he was the arranger for the title track of an album released on June 5, 2015. That album

Table 5: The Application Ecosystem of NaturalGAIA

Category	Applications	Primary Information Space (Sources of Ground Truth)
Global Knowledge	Wikipedia, Google Search	<b>Universal Facts:</b> Historical events, geographical data, official entity names.
Computational Knowledge	WolframAlpha	<b>Scientific Facts:</b> Historical weather data, astronomical timestamps, chemical element properties, mathematical constants.
Multimedia (Audio)	Spotify, Apple Music, NetEase Cloud	<b>Discography Data:</b> Album release dates, track duration, artist labels, explicit content tags.
Multimedia (Video)	YouTube, Bilibili	<b>Video Metadata:</b> Upload timestamps, channel/Uploader names, video duration, view count snapshots.
Geo-Spatial	Google Maps, Amap (Gaode)	<b>POI &amp; Navigation:</b> Business addresses, subway exit numbers, historical traffic data, distance estimates.
Movies & TV	IMDb, Douban	<b>Filmography:</b> Cast lists, director credits, runtime, release years, award history.
Finance & Commerce	Yahoo Finance, Amazon	<b>Market Data:</b> Historical stock closing prices, product dimensions, ISBNs, publication dates.
Sports	ESPN / NBA App	<b>Match Records:</b> Final scores, game dates, player statistics, MVP awards.
Travel & Logistics	Booking.com, FlightAware, TripAdvisor	<b>Itinerary Facts:</b> Hotel check-in policies, historical flight arrival times, gate records, historical establishment awards.
Digital Entertainment	Steam, Epic Games	<b>Software Specs:</b> Release dates, developer names, minimum system requirements (OS/RAM), achievement lists.
System Utilities	Messages, Contacts, Calendar, Photos, Calculator, Files, Clock	<b>Local Context:</b> Sender IDs, meeting slots, EXIF timestamps, file paths, alarm settings.

Benchmark	Environment	Tasks	Verifiable State	Linguistic Decoupling	Evaluation Granularity
Mind2Web	Desktop Web	2350	None (Action Match)	No	Single-level
OSWorld	Desktop/Web	369	Cloud/Device State	No	Single-level
Web Voyager	Desktop Web	643	LLM Judge	No	Single-level
AndroidWorld	Android	116	Device State	No	Single-level
NaturalGAIA (Ours)	Android/Desktop	276	Device State + Text Match	Yes (Causal Paths)	Multi-level (WPSR, MATCR, ATSR)

Table 6: Systematic comparison of NaturalGAIA with existing GUI agent benchmarks.

Agent	Model	Obs. Space	SR (P@1)
MAe	Qwen3.5-397b-a17b	Screenshot	46.7
LightManus-MAe	Qwen3.5-397b-a17b	Screenshot	50.0
LightManus-Jarvis	Qwen3.5-397b-a17b	Scrn + A11y tree	63.3
M3A	UI-TARS-1.5-7b	Scrn + A11y tree	33.3
LightManus-Jarvis	UI-TARS-1.5-7b	Scrn + A11y tree	36.7

Table 7: Cross-platform verification experiments on the AndroidWorld benchmark.

belongs to the same singer who performed 'Crush You into an Apple'.

Could you act as a detective? First, use Google Search to identify the singer of that song. I also need her debut year for my notes, so please look that up on Wikipedia. Then, trace back to her 2015 album, find that arranger's name via Wikipedia, and

finally use Google Search to tell me which movie he directed the music for in 2019."

#### Atomic tasks partitioned by LLM:

- "Use Google Search to determine which album includes the song 'Crush You into an Apple', and identify the performer of that album from the search results snippet/page."
- "Open the Wikipedia app and search for the singer of 'Alien from the Sky' to find the year they debuted."
- "Use Google Search to determine which album Joker Xue released on 2015-06-05."
- "Use Wikipedia to open the page for the album identified in task 3 as being released in 2005,

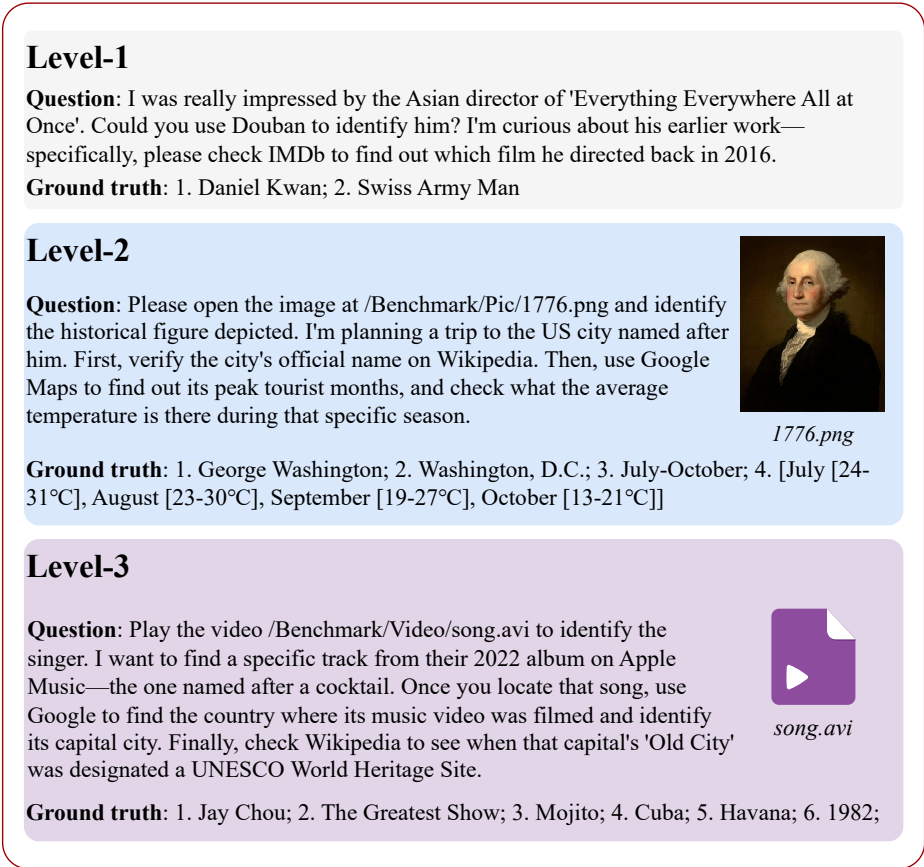


Figure 6: Task examples of different difficulty levels in NaturalGAIA

Method	Level-1				Level-2				Level-3				Overall			
	SR (P@1/4)	WPSR	MATCR	ATSR	SR (P@1/4)	WPSR	MATCR	ATSR	SR (P@1/4)	WPSR	MATCR	ATSR	SR (P@1/4)	WPSR	MATCR	ATSR
Mind2Web	40.0 / 86.7	49.1	58.3	54.9	10.0 / 50.0	20.0	36.9	33.0	10.0 / 30.0	17.8	29.2	27.9	22.9 / 60.0	22.7	43.9	32.6
LightManus-Jarvis	46.7 / 86.7	56.2	61.7	59.8	30.0 / 70.0	32.5	40.0	38.0	10.0 / 40.0	20.3	27.5	26.8	31.4 / 68.6	28.8	45.7	34.0

Table 8: Performance comparison between LightManus-Jarvis and Mind2Web on a pure web subset of NaturalGAIA.

and identify the arranger of the album's title track (the song with the same name as the album)."

- "Use Google Search to search the arranger identified in task 4 and find which film they served as music director for in 2019."

**Error analysis:** The Planning & Reasoning Error (PRE) in this case stems from inconsistent task granularity during decomposition. The agent combined the dependent steps of "identifying the album" and "identifying the performer" into a single atomic task, causing a 1-step alignment shift for all subsequent IDs.

**Structural Compliance Errors(SCE)**

**The action given by LLM:**

"action": "clear\_text(uid(5))"

**Error analysis:** "The returned action format is incorrect. It should be clear\_text(5); the formatting error caused the action to fail."

**Operational Errors(OE)**

"The 'Expand All' button should have been clicked, but it clicked on Project Satan instead. The click position was incorrect."

**Knowledge Deficits(KD)**

**Error analysis:** The LLM lacks specific knowledge on how to operate Map applications. For the task of searching for the nearest major commercial district near Jiangnanxi Metro Station, Guangzhou, Guangdong Province, China, the model only queried the station name itself. Consequently, it failed to retrieve nearby commercial information. The correct search query should have

been: major commercial districts near Jiangnanxi Metro Station, Guangzhou, Guangdong Province, China.

### **Perceptual Errors**

**Task Description:** Use Google to find out which dynasty China was in during the year 1776.

**Error Analysis:** The current screenshot already contains the required answer. However, the LLM misjudged the situation, and failed to extract the available information and instead returned to the home screen, leading to an unnecessary execution step.



Figure 7: Operational Error Example



### 江南西地铁站

4.4 ★★★★★ (13)

Open now

Directions

Save

Share

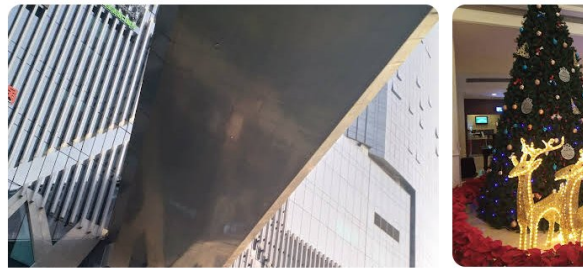


Figure 8: Operational Error Example

4:27



Search Wikipedia

1



## 清朝

中國歷史上最後一個帝制王朝，由滿族建立和主導，存在於17世紀至20世紀初

清朝 (1636年5月15日或1644年6月6

日<sup>[註 1]</sup>—1912年2月12日)，國号大清

(满语: ᡩᠠᡳᠴᡳᠩ, 穆麟德轉寫: daicing

gurun)<sup>[書 4]</sup>，并使用中國 (满语: ᡩᠠᡳᠴᡳᠩ, 转



Save



Language



Find in article

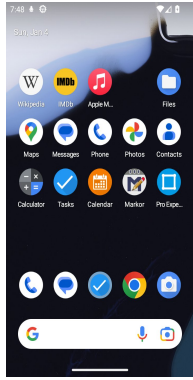


Theme

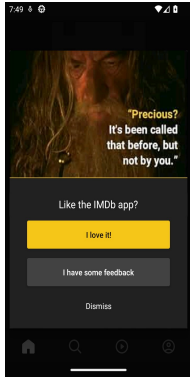


Contents

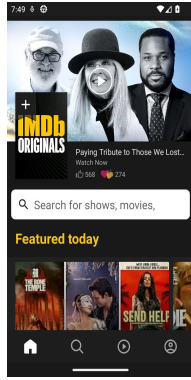
Figure 9: Operational Error Example



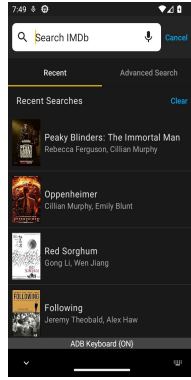
T1-S1



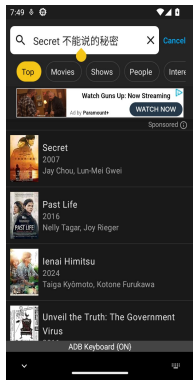
T1-S2



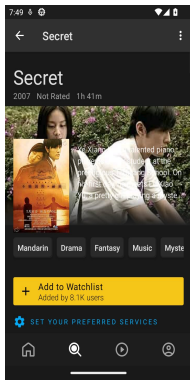
T1-S3



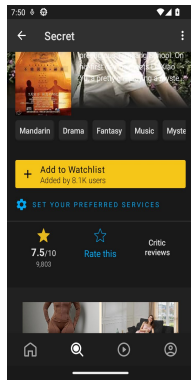
T1-S4



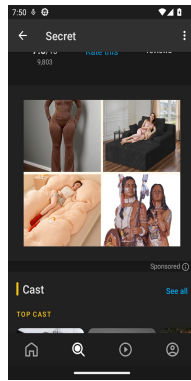
T1-S5



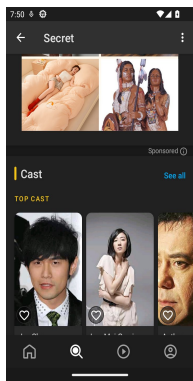
T1-S6



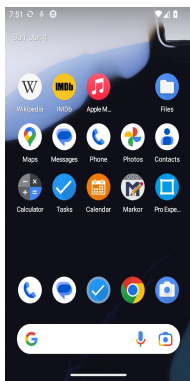
T1-S7



T1-S8



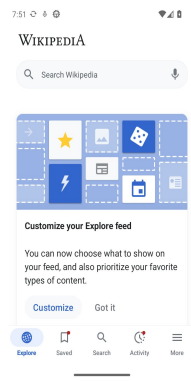
T1-S9



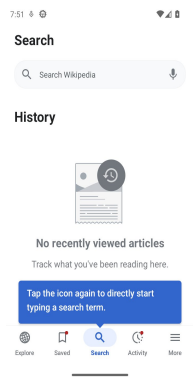
T2-S1



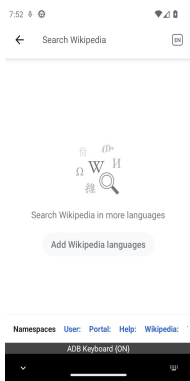
T2-S2



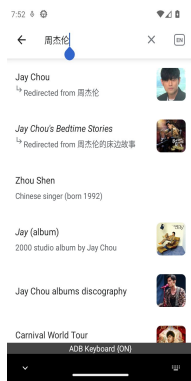
T2-S3



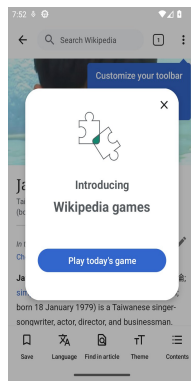
T2-S4



T2-S5



T2-S6



T2-S7

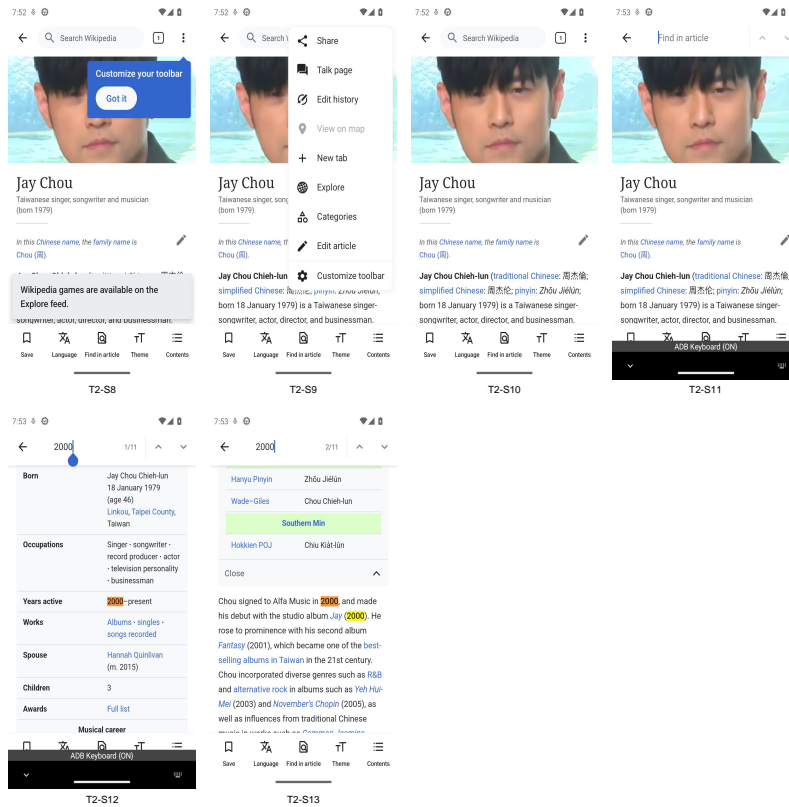


Figure 10: Image from image\_10-42\_1\_1.jpg to image\_10-42\_2\_13.jpg

```

1 {
2   "images": [
3     "image_-10-42_1_1.jpg",
4     "image_-10-42_1_2.jpg",
5     "image_-10-42_1_3.jpg",
6     "image_-10-42_1_4.jpg",
7     "image_-10-42_1_5.jpg",
8     "image_-10-42_1_6.jpg",
9     "image_-10-42_1_7.jpg",
10    "image_-10-42_1_8.jpg",
11    "image_-10-42_1_9.jpg",
12    "image_-10-42_2_1.jpg",
13    "image_-10-42_2_2.jpg",
14    "image_-10-42_2_3.jpg",
15    "image_-10-42_2_4.jpg",
16    "image_-10-42_2_5.jpg",
17    "image_-10-42_2_6.jpg",
18    "image_-10-42_2_7.jpg",
19    "image_-10-42_2_8.jpg",
20    "image_-10-42_2_9.jpg",
21    "image_-10-42_2_10.jpg",
22    "image_-10-42_2_11.jpg",
23    "image_-10-42_2_12.jpg",
24    "image_-10-42_2_13.jpg"
25  ],
26  "original_task": [
27
28  ]
29
30  ===Task1===
31
32  "overall_task": [
33    "Use IMDB to search for the movie 'Secret' and identify the male lead actor."
34  ]
35

```

```

36  ===Step1===
37
38  <image_-10-41_1_1.jpg>
39
40  **Thought**
41  We need to use the IMDb app to search for the movie, so first open IMDb from the
    home screen.
42
43  **Action**
44  tap(5)
45
46  ===Step2===
47
48  <image_-10-41_1_2.jpg>
49
50  **Thought**
51  IMDb opened successfully but a feedback popup is blocking the app. I need to
    dismiss it to access search.
52
53  **Action**
54  tap(4)
55
56  ===Step3===
57
58  <image_-10-41_1_3.jpg>
59
60  **Thought**
61  The feedback popup is gone and the IMDb home screen is visible with a prominent
    search field. Next step is to open the search input to look up the movie title.
62
63  **Action**
64  tap(13)
65
66  ===Step4===
67
68  <image_-10-41_1_4.jpg>
69
70  **Thought**
71  Search screen is open with an editable field labeled 'Search IMDb'. To find the
    movie 'Secret', I should type the title into this field and submit.
72
73  **Action**
74  input_text(3, "Secret")
75
76  ===Step5===
77
78  <image_-10-41_1_5.jpg>
79
80  **Thought**
81  The query is entered and results are shown. The top result is the movie 'Secret'
    (2007) with cast listed. Open this movie's page to confirm the male lead actor.
82
83  **Action**
84  tap(31)
85
86  ===Step6===
87
88  <image_-10-41_1_6.jpg>
89
90  **Thought**
91  Now on the IMDb title page for 'Secret' (2007). To identify the male lead actor,
    I need to find the cast/credits section on this page, which likely requires
    scrolling down to the 'Top cast' area.
92
93  **Action**
94  swipe("UP", "MEDIUM")
95
96  ===Step7===
97
98  <image_-10-41_1_7.jpg>

```

```

99
100 **Thought**
101 The swipe did not reveal the cast section yet; we're still near the top with
plot, genres, and rating. To find the male lead actor, I need to scroll further
down until the 'Top cast' appears.
102
103 **Action**
104 swipe("UP", "LONG")
105
106 ===Step8===
107
108 <image_-10-41_1_8.jpg>
109
110 **Thought**
111 The swipe was successful: the Cast section with TOP CAST is now visible. To
identify the male lead actor, I need to view the top cast names; they should be
in the visible list below, so I will scroll slightly to reveal the first cast
entries.
112
113 **Action**
114 swipe("UP", "MEDIUM")
115
116 ===Step9===
117
118 <image_-10-41_1_9.jpg>
119
120 **Thought**
121 The swipe revealed the Top Cast list with actor names. The first listed male
actor is Jay Chou, which indicates the male lead. The task is to identify the
male lead actor, so I can complete now.
122
123 **Action**
124 finish("On IMDb for 'Secret', the male lead actor listed in Top Cast is Jay Chou
.")
125
===Task2===
126
127 "overall_task": [
128     "Use Wikipedia to search for Jay Chou and find which album he released in
2000."
129 ]
130
131 ===Step1===
132
133 <image_-10-41_2_1.jpg>
134
135 **Thought**
136 We need to use Wikipedia to search for Jay Chou, so first open the Wikipedia app
from the home screen.
137
138 **Action**
139 tap(4)
140
141 ===Step2===
142
143 <image_-10-41_2_2.jpg>
144
145 **Thought**
146 Wikipedia app opened to an onboarding language selection screen that blocks
access to search. To proceed quickly to the app and search for Jay Chou, skip
onboarding.
147
148 **Action**
149 tap(8)
150
151 ===Step3===
152
153 <image_-10-41_2_3.jpg>
154
155 **Thought**
156

```

```

157 Onboarding was skipped successfully; we are now on Wikipedia Explore with a
158 banner. To search for Jay Chou, open the Search tab.
159
160 **Action**
161 tap(15)
162
163 ===Step4===
164
165 <image_-10-41_2_4.jpg>
166
167 **Thought**
168 We're on Wikipedia's Search tab with a visible 'Search Wikipedia' field. Next
169 step is to tap the search field and type Jay Chou to find his page and then
170 identify the 2000 album.
171
172 **Action**
173 tap(4)
174
175 ===Step5===
176
177 <image_-10-41_2_5.jpg>
178
179 **Thought**
180 The tap opened the Wikipedia search interface with the editable search text
181 field visible. Next we need to enter Jay Chou to pull up the correct article.
182
183 **Action**
184 input_text(4, "Jay Chou")
185
186 ===Step6===
187
188 <image_-10-41_2_6.jpg>
189
190 **Thought**
191 Search results for Jay Chou are displayed. To find which album he released in 20
192 00, open the main Jay Chou article result.
193
194 **Action**
195 tap(10)
196
197 ===Step7===
198
199 <image_-10-41_2_7.jpg>
200
201 **Thought**
202 A popup about Wikipedia games is blocking the article content, so I need to
203 close it before reading the Jay Chou page to find the 2000 album.
204
205 **Action**
206 tap(5)
207
208 ===Step8===
209
210 <image_-10-41_2_8.jpg>
211
212 **Thought**
213 A modal popup titled "Introducing Wikipedia games" is still blocking the Jay
214 Chou article; I need to close it using the X in the popup to access the content
215 and find the 2000 album.
216
217 **Action**
218 tap(12)
219
220 ===Step9===
221
222 <image_-10-41_2_9.jpg>
223
224 **Thought**
225 The popups are gone, but a three-dot overflow menu is open covering the article.
226 I need to dismiss this menu to access the article content and find the 2000

```

```

album.
218
219 **Action**
220 back()
221
222 ===Step10===
223
224 <image_-10-41_2_10.jpg>
225
226 **Thought**
227 The overflow menu is now dismissed and I can see the Jay Chou article. To find
the album released in 2000 quickly, I should use the built-in "Find in article"
to search for 2000 or "studio albums."
228
229 **Action**
230 tap(46)
231
232 ===Step11===
233
234 <image_-10-41_2_11.jpg>
235
236 **Thought**
237 The Find in article bar is open with an input field. To locate the relevant
section about albums released in 2000, I should search within the article for
the year 2000.
238
239 **Action**
240 input_text(56, "2000")
241
242 ===Step12===
243
244 <image_-10-41_2_12.jpg>
245
246 **Thought**
247 The in-page find for "2000" is active and shows 1/11 matches; we're currently on
the infobox match (Years active 2000-present), not the album release. I should
jump to the next match to find where the 2000 album is mentioned.
248
249 **Action**
250 tap(89)
251
252 ===Step13===
253
254 <image_-10-41_2_13.jpg>
255
256 **Thought**
257 The next match (2/11) highlights text stating he made his debut with the studio
album 'Jay' (2000), which answers which album he released in 2000. I should end
the find mode and report the album.
258
259 **Action**
260 finish("On Jay Chou's Wikipedia page, the text says he made his debut with the
studio album 'Jay' (2000). So the album released in 2000 is: Jay.")
261 }

```

Listing 1: Example of Execution Trajectory