OxyGent: Making Multi-Agent Systems Modular, Observable, and Evolvable via Oxy Abstraction

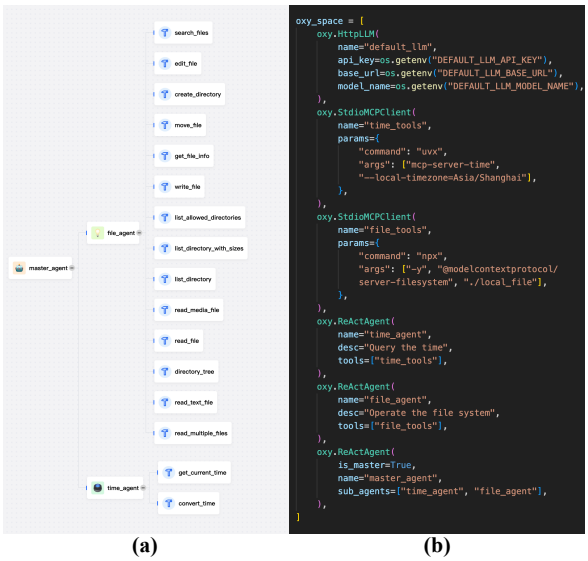
Junxing Hu* **Tianlong Li*** **Lei Yu** **Ai Han†**
JD.com
Beijing, China
junxing.hu@cripac.ia.ac.cn, {litianlong26, yulei116, hanai5}@jd.com

Abstract

Deploying production-ready multi-agent systems (MAS) in complex industrial environments remains challenging due to limitations in scalability, observability, and autonomous evolution. We present OxyGent, an open-source framework driven by two core novelties: a unified Oxy abstraction and the OxyBank evolution engine. The unified abstraction encapsulates agents, tools, LLMs, and reasoning flows as pluggable atomic components, enabling Lego-like scalable system composition and non-intrusive monitoring. To enhance observability, OxyGent introduces permission-driven dynamic planning that replaces rigid workflows with execution graphs generated at runtime, providing adaptive visualizations. Furthermore, to support continuous evolution, OxyBank serves as an AI asset management platform that drives automated data backflow, annotation, and joint evolution. Empirical evaluations and real-world case studies show that OxyGent provides a robust and scalable foundation for MAS. OxyGent is fully open-sourced under the Apache License 2.0 at https://github.com/jd-opensource/OxyGent.

1 Introduction

Large Language Model (LLM) agents are transitioning from experimental prototypes to collaborative Multi-Agent Systems (MAS) capable of solving sophisticated real-world tasks (Wang et al., 2024). However, in large-scale distributed environments such as intelligent assistants (Fu et al., 2024), existing MAS frameworks encounter critical bottlenecks. Traditional development paradigms may lack standardized abstractions, leading to low efficiency and repetitive implementation of core functions. Moreover, rigid plan-and-execute workflows are too brittle to handle the uncertainties of dynamic environments. In addition, without a closed-loop mechanism to evaluate and refine agents using online feedback, performance tends to stagnate or even deteriorate (Gao et al., 2025).



(a) (b)

Figure 1: A multi-agent file management assistant built with OxyGent. (a) MAS visualization. (b) Permission relationships are defined in the implementation code.

To enhance scalability and development efficiency, OxyGent introduces a unified abstraction that encapsulates agents, tools, and reasoning flows as pluggable Oxy nodes. This Lego-like approach empowers researchers to assemble and hot-swap specialized components without complex manual reconfiguration rapidly. Furthermore, by integrating Aspect-Oriented Programming (AOP) (Kiczales et al., 1997), the framework separates core business logic from cross-cutting concerns, maintaining a clean, modular architecture for seamless expansion in production settings. Building on this abstraction to improve observability, OxyGent replaces static DAGs with permission-driven, dynamic planning. By defining potential collaboration spaces through node permissions (Figure 1(b)), the system automatically synthesizes execution paths at runtime and generates real-time call graph

*Equal contribution.
†Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 586–596

visualizations (Figure 1(a)). This transparency allows users to thoroughly inspect decision-making trajectories, ensuring that emergent MAS behaviors remain fully monitorable and manageable even in volatile task environments.

To facilitate the agent evolution, OxyGent introduces OxyBank, a specialized platform for AI asset management. OxyBank serves as the evolutionary engine of OxyGent by providing a systematic pipeline that captures online execution traces and converts them into high-quality training samples. By integrating automated rewarding and human-in-the-loop annotation, OxyBank enables the backflow of domain-specific knowledge into the multi-agent system. This closed-loop process supports the joint evolution of agent strategies, ensuring that the collective intelligence of the ecosystem continuously improves through data-centric learning rather than remaining static after deployment.

While many recent frameworks emphasize engineering integration, OxyGent's central research novelties lie in two foundational pillars: the unified Oxy abstraction and the OxyBank evolution engine. The main contributions are summarized as follows:

- **Unified Oxy Abstraction.** We propose the unified Oxy abstraction to resolve structural heterogeneity in MAS, enabling Lego-like construction and high scalability.
- **OxyBank Evolution Engine.** We introduce an AI asset platform that drives continuous MAS self-improvement via automated data backflow and verifiable evaluation.
- **System Observability.** Built upon the Oxy abstraction, we implement permission-driven dynamic planning and an AOP-infused orchestration paradigm, ensuring deep observability.
- **Empirical Validation & Open Source.** Evaluations demonstrate OxyGent's robustness. We release the complete framework, built-in toolkits, a trace visualization Web UI, and comprehensive reference examples.

2 Related Work

Orchestration Paradigms and Dynamic Planning. Current MAS frameworks focus on balancing agent autonomy with structural control through diverse paradigms. LangGraph (LangChain AI, 2024) serves as a low-level orchestration framework and runtime for building long-running, stateful agents via graph structures. CrewAI (CrewAI-Inc, 2025) utilizes a role-driven approach, combining the collaborative intelligence of "Crews" with the precise control of "Flows" to manage complex processes. Beyond graph and role-based designs, Semantic Kernel (Microsoft, 2024) integrates AI capabilities with enterprise business logic, while MetaGPT (Hong et al., 2024) encodes Standardized Operating Procedures (SOPs) into prompt sequences to mimic software company workflows. Recent research has also explored more fluid architectures. Aime (Shi et al., 2025) replaces fixed planners with dynamic actor instantiation, and AFlow (Zhang et al., 2025) automates the generation of agentic workflows. OxyGent distinguishes itself by unifying agents, tools, and flows into interchangeable atomic nodes governed by permission-driven dynamic planning and real-time execution visualization.

Execution Lifecycle and System Observability. Observability is a critical production-readiness requirement, yet many frameworks prioritize developer experience over runtime monitoring. OpenAI Agents SDK (OpenAI, 2025), the successor to Swarm (OpenAI, 2024), adopts a minimalist set of primitives, including Agents, Handoffs, and Guardrails, and provides built-in tracing for debugging agentic flows. AutoGen (Wu et al., 2024) models interactions as asynchronous message passing between entities, while Pydantic AI (Pydantic, 2024) introduces a type-safe Python framework that enforces strict contracts for tool calling and data exchange. For enterprise-grade monitoring, Strands Agents (AWS, 2025) provides a model-agnostic toolkit optimized for OpenTelemetry integration, similar to the durable execution principles in Dapr Agents (Dapr, 2025). OxyGent implements a comprehensive AOP-based lifecycle management system that modularly injects cross-cutting concerns such as security auditing and performance monitoring into distributed agent joinpoints.

Evolutionary AI Asset Management. The transition toward agents that learn and improve has led to systems that prioritize memory and training loops (Han et al., 2025). Agno (Agno-agi, 2025) serves as specialized infrastructure for reasoning systems, utilizing its AgentOS layer to manage long-term memory and knowledge accumulation across sessions. AWorld (Yu et al., 2025) provides a high-performance interaction runtime designed for large-scale experience generation to enable practical reinforcement learning for agents. EvoA-

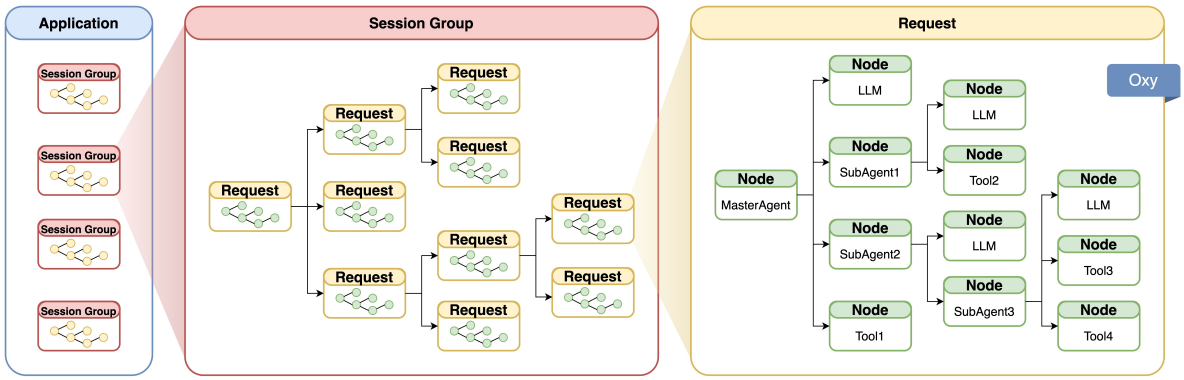


Figure 2: The OxyGent framework provides four data scopes: Application, Session Group, Request, and Node. This multi-level data isolation and sharing ensures data management efficiency and development convenience in MAS.

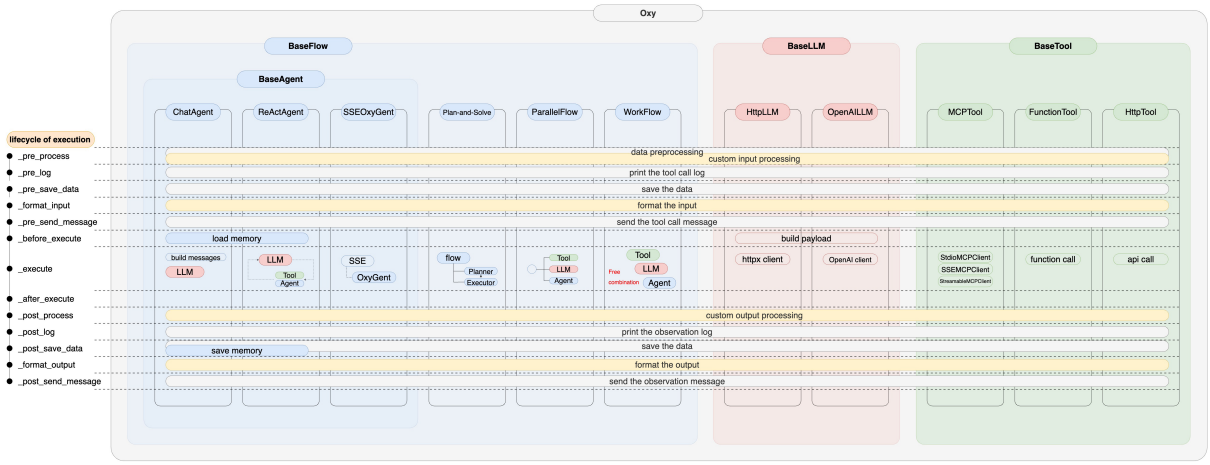


Figure 3: The execution lifecycle of Oxy. A series of management and coordination steps ensures data flow and processing, enabling MAS to dynamically plan and generate flowcharts in real time.

gent (Yuan et al., 2025) and SE-Agent (Lin et al., 2025) have proposed automatic multi-agent extension via evolutionary operators and cross-trajectory refinement, respectively. Moreover, OWL (Hu et al., 2025) improves cross-domain generalization by optimizing a domain-agnostic planner with reinforcement learning from real-world feedback, whereas Chain-of-Agents (Zhang et al., 2024) focuses on distilling multi-agent reasoning patterns into unified foundation models. OxyGent facilitates a continuous build-inference-evolution cycle through OxyBank, an industrial-scale AI asset platform that unifies data annotation and knowledge backflow to improve collective agents.

3 OxyGent

OxyGent aims to transform multi-agent orchestration from static, hard-coded workflows into an elastic, Lego-like assembly of atomic components. This section details the framework's infrastructure, the dynamic planning mechanisms, and the evolutionary loop powered by OxyBank.

3.1 Unified Oxy Abstraction and Hierarchical Data Scopes

To facilitate the flexible construction of complex MAS, OxyGent introduces a unified atomic abstraction where agents, tools, LLMs, and flows are wrapped into pluggable **Oxy** components. Unlike traditional frameworks that treat these entities as distinct software layers, OxyGent provides a consistent interface for all AI atoms, enabling hot-swapping of reasoning logic and capabilities at runtime without refactoring.

A cornerstone of unified abstraction is the four-tier data scoping mechanism, which provides a structured approach to state management in distributed environments (Figure 2). Rather than relying on a single global state, OxyGent isolates data into specific domains to prevent data-privilege leaks and ensure efficiency. The domain description and data processing tools are as follows:

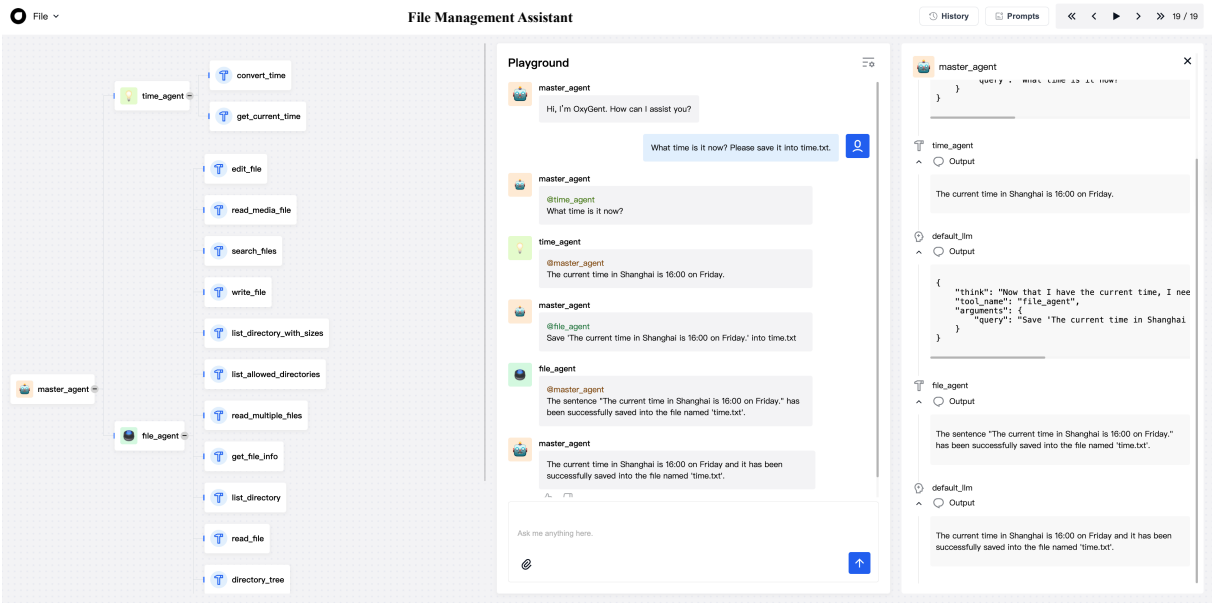


Figure 4: The file management assistant is built on OxyGent. From left to right: MAS visualization, question-answering box, and the output of each node. As shown in Figure 11, each intermediate step of the reasoning process can be rolled back and reviewed.

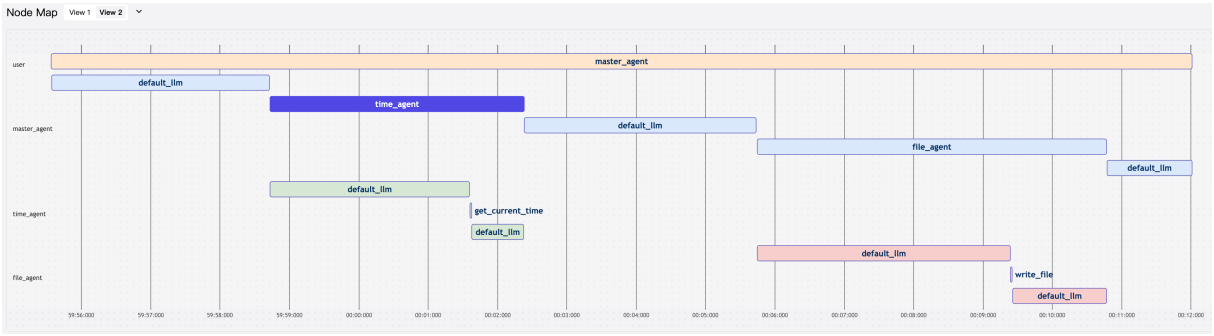


Figure 5: MAS inference monitoring. OxyGent has built-in production-grade time tracking, which displays task distribution and resource congestion in real time, facilitating MAS architecture optimization. More features are introduced in Appendix A.1.

- **Application**: Global context accessible across all MAS instances. Data tools are: `oxy_request.get/set_global_data()`
- **Session Group**: Shared memory among a cluster of related conversations. Data tools are: `oxy_request.get/set_group_data()`
- **Request**: Transient data restricted to a single inference trajectory. Data tools are: `oxy_request.get/set_shared_data()`
- **Node**: Local arguments specific to an individual Oxy component. Data tools are: `oxy_request.get/set_arguments()`

Through hierarchical data isolation and sharing, the efficiency of data management and the ease of development in multi-agent collaborative scenarios are significantly improved.

3.2 Permission-Driven Planning and Standardized Execution Lifecycle

In contrast to fixed workflows, OxyGent utilizes permission-driven dynamic planning to govern agent collaboration. The execution trajectory is not predefined but emerges from the authorization relationships between Oxy nodes. To support this dynamism while maintaining observability, we implement a comprehensive Oxy execution lifecycle as shown in Figure 3. Each node traverses a standardized sequence of management steps, and the example operations for each stage are as follows:

- **Pre-execution Phase**: `_pre_process` is used for data formatting and `_pre_save_data` persists inputs before logic invocation.
- **Core Execution**: The `_execute` hook triggers the actual LLM reasoning or tool usage.

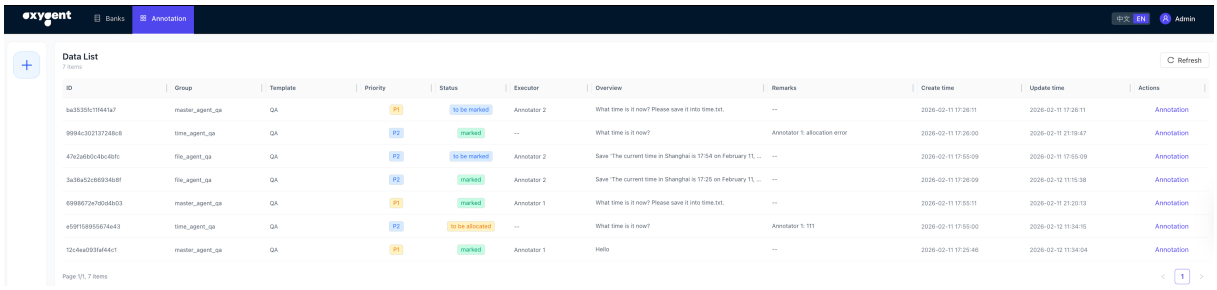


Figure 6: OxyBank, as OxyGent's one-stop AI asset management platform, supports knowledge base construction, data annotation, memory management, multi-agent system evaluation, and enables agent evolution.

- **Post-execution Phase**: `_post_process` for data post-processing and `_format_output` for downstream compatibility.

By utilizing Aspect-Oriented Programming (AOP), OxyGent injects observability and security operators into these lifecycle joinpoints. This enables the system to automatically synthesize real-time execution visualizations of the actual call graphs during runtime (Figure 4). Furthermore, as shown in Figure 5, OxyGent incorporates production-grade time tracking to expose real-time agent-level resource consumption. By decomposing execution into LLM inference, tool and API calls, and inter-agent interactions, it enables efficient bottleneck identification and fine-grained architectural optimization.

3.3 Evolutionary AI Asset Management via OxyBank

The long-term value of an MAS lies in its ability to self-improve. We introduce OxyBank, an AI asset management platform serving as the framework's memory bank (Figure 6). It bridges the gap between raw execution data and continuous self-improvement through the following mechanisms:

Closed-Loop Asset Pipeline. The framework supports autonomous evolution by establishing a closed-loop cycle for annotation, auditing, and data backflow. Online execution traces are automatically captured and precipitated as memory assets within OxyBank. These assets are processed by annotation agents or experts using customized templates (*e.g.*, QA or business-domain tagging). Once audited, the high-quality samples are used to update knowledge bases or fine-tune models. This pipeline ensures that the MAS continuously refines its decision-making strategies based on validated feedback from production.

Multi-Layered Quality Gating. To address potential degradation from noisy traces, OxyBank implements a strict multi-layered quality-gating pipeline. Specifically, it employs MD5-based deduplication at deposit to prevent frequency bias, and infers trace priorities based on call chain metadata (*e.g.*, end-to-end user interactions are prioritized as P0). Crucially, no raw trace can bypass the strict state machine (pending → annotated → approved). The pending or rejected data are strictly isolated from the knowledge base, ensuring that only high-fidelity signals drive the evolution process.

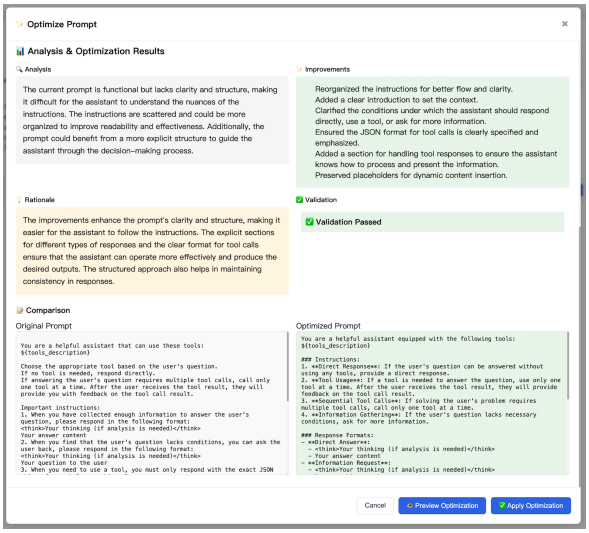


Figure 7: AI-driven Optimize Prompt module in OxyGent, which automatically analyzes and refines agent prompts based on historical execution traces, improving prompt quality without manual engineering.

Autonomous Prompt Optimization. Instead of relying solely on manual prompt engineering, OxyGent integrates an AI-driven Optimize Prompt module as shown in Figure 7. It automates the extraction of best practices from validated traces, allowing agents to iteratively refine their system prompts based on historical execution contexts. This dynamically blends the reliability of human-in-the-loop auditing with the scalability of autonomous LLM optimization, significantly accelerating the agents' adaptation to novel tasks.

Results: Test

Agent name	Model family	organisation	Average score (%)	Level 1 score (%)
OWL++	GPT-4.1		60.8	75.27
Agent_v0.0.4	gpt-4.1		60.47	87.1
csy_v0.23			60.13	75.27
ATH High			59.8	47.31
OxyGent-v0.2	Claude,DeepSeek,GPT	JD	59.14	77.42
SuperCheerlink beta	Claude, Gemini	Cheerlink Labs	58.8	70.97
agent_0711	gpt o3		58.14	81.72
OxyGent-v0.1	Claude,DeepSeek,GPT	JD	57.81	77.42
🏂	claude-37-sonnet, gemini-25-pro, gpt-o4-mini		57.14	78.49
zl-test			57.14	72.04
Agent_v0.0.3	gpt-4.1		57.14	86.02
csy_v0.22			56.48	74.19

Figure 8: On July 22, 2025, OxyGent achieved the second-highest score (59.14%) among open-source methods on the GAIA leaderboard, second only to OWL++, which was the strongest open-source method at the time.

4 Case Studies

4.1 Evaluations on the GAIA Benchmark

Benchmark. GAIA (Mialon et al., 2023) focuses on real-world tasks that require fundamental human-level abilities. It is intuitive for humans, who achieve an average accuracy of 92%, while frontier models like GPT-4, equipped with plugins, struggled at 15%. The benchmark comprises 466 questions across 3 levels of increasing complexity.

Experimental Results. To evaluate the orchestration effectiveness of OxyGent, we tested the framework against the GAIA leaderboard. On July 22, 2025, OxyGent achieved a score of 59.14%, ranking as the **second-highest** open-source method globally at the time, closely trailing the open-source benchmark OWL++ (Hu et al., 2025) (60.8%), as shown in Figure 8. Although the emergence of newer state-of-the-art models and advanced reasoning toolsets has since moved the performance ceiling, these results highlight OxyGent's inherent structural advantages in managing long-chain interactions and multi-modal dependencies efficiently. By leveraging its unified atomic abstraction, the framework enables high performance even with earlier-generation base models. The method is detailed in Appendix A.2. The full implementation and specific configurations used for this evaluation are available for community review at: `https://github.com/jd-opensource/OxyGent/tree/gaia59`.

Ablation Study. To isolate the performance gains derived from our framework's specific designs, we conducted an ablation study on the GAIA benchmark (Table 1). Since the unified Oxy abstraction serves as the non-removable foundation of the system, we evaluated the sequential addition of multi-agent orchestration, permission-driven dynamic planning, and the memory backflow (the predecessor to OxyBank). The results show that dynamic planning significantly boosts Level 2 reasoning (+18.24%), while the memory mechanism predominantly enhances L1 task precision (+15.05%) and overall average scores, validating that the gains stem from OxyGent's architectural designs rather than merely the underlying foundation models.

Method	Avg	Level 1	Level 2	Level 3
Single agent	36.21	61.29	29.56	10.20
+ Multi-agent	42.19	62.37	35.85	24.49
+ Planning	52.16	62.37	54.09	26.53
+ Memory	**59.14**	**77.42**	**56.60**	**32.65**

Table 1: Ablation study on the GAIA benchmark (%).

4.2 Case Analysis in Business Scenarios

Beyond academic benchmarks, OxyGent has demonstrated significant industrial impact through wide-scale deployment in enterprise production and open-source communities. To illustrate the framework's capacity for extreme scale and complexity, we implemented a hierarchical multi-agent classification system for e-commerce customer service, as shown in Figure 9.

Setup and Evaluation. This system comprises over **2,000 agents** tasked with categorizing incoming requests into a taxonomy of more than 2,400 distinct labels. Each category contains as few as 10 samples, making the task a massive few-shot classification problem. We compared OxyGent against a strong single-agent baseline utilizing RAG retrieval coupled with DeepSeek-R1-Distill-Qwen-

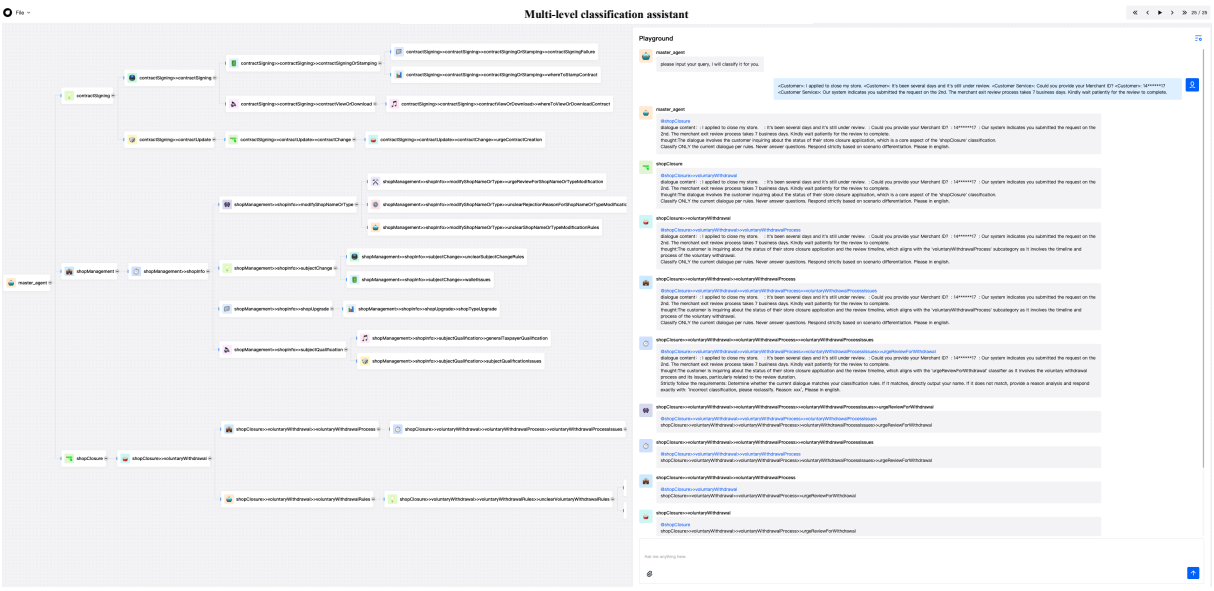


Figure 9: A hierarchical MAS of 2,000+ agents for e-commerce classification, employing a top-down decision chain with dynamic logic and full-chain tracing. "»" denotes a call invocation between Oxy nodes.

32B (Guo et al., 2025). The evaluation protocol continuously samples 2,000+ real-world interactions weekly for AI and human joint assessment.

Results and Trade-offs. The MAS-driven approach improved classification accuracy from 61.3% (baseline) to 85.6%. Furthermore, it demonstrated strong topological self-evolution, autonomously discovering and validating an average of 5.4 new categories per week. Regarding cost and latency tradeoffs, the dynamic generation of the 2,000+ agents keeps memory overhead manageable. While the multi-tiered decision architecture increases average inference latency by 2.3× compared to the single-agent baseline, this trade-off is strictly justified for offline business environments prioritizing high precision.

Failure Cases. We observed that highly ambiguous semantic queries occasionally induce repetitive ReAct loops or hallucinations, which are currently mitigated by our trace replay and manual auditing loops. Due to space constraints, additional industrial scenarios including automated SOPs, RAG-based assistants, and community-contributed tools are detailed in Appendix A.3.

5 Discussion

Decision Routing and Structural Resilience. To resolve execution conflicts when multiple nodes qualify, OxyGent utilizes a centralized orchestrator for dynamic, permission-constrained delegation. Under node failures, the framework prevents global trajectory collapse by encapsulating local errors into reasoning memory, enabling autonomous corrective replanning.

Connections to Concurrent Works. AlphaApollo (Zhou et al., 2026a) focuses on deep agentic reasoning and learning. In contrast, OxyGent provides a general-purpose, modular, and observable framework, designed for reliable industrial evolution via verifiable data backflow. For inspectability, Landscape of Thoughts (Zhou et al., 2026b) provides post-hoc visual analysis of textual reasoning trajectories, whereas OxyGent visualizes multi-agent execution topologically at runtime for immediate architectural debugging. Finally, regarding AR-Bench (Zhou et al., 2025), while we evaluated on GAIA to test general multi-modal assistance, OxyGent's dynamic planning inherently supports the iterative information-seeking required by active reasoning tasks.

6 Conclusion

In this paper, we present OxyGent, an open-source framework for building scalable and self-improving multi-agent systems driven by two core novelties. First, its unified atomic abstraction enables a Lego-like construction process that replaces rigid workflows with permission-driven dynamic planning, offering high observability through real-time execution visualizations. Second, integrated with OxyBank for AI asset management, the framework continuously refines strategies via a validated feedback pipeline, as evidenced by evaluations on the GAIA benchmark and industrial applications.

Limitations. Although OxyBank supports self-evolution, large-scale training currently depends on manual resource configuration. We are actively developing intelligent resource scheduling to achieve a fully automated lifecycle, including agent construction, deployment, and refinement. This will minimize human intervention and further accelerate the evolution of collective intelligence in complex environments.

References

Agno-agi. 2025. Agno. https://github.com/agno-agi/agno.

Anthropic. 2024. Claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet.

AWS. 2025. Strands agents. https://github.com/strands-agents/sdk-python.

CrewAIInc. 2025. Crewai: Framework for orchestrating role-playing, autonomous ai agents. https://github.com/crewAIInc/crewAI.

Dapr. 2025. Dapr agents: A framework for agentic ai systems. https://github.com/dapr/dapr-agents.

Yicheng Fu, Raviteja Anantha, and Jianpeng Cheng. 2024. Camphor: Collaborative agents for multi-input planning and high-order reasoning on device. *arXiv preprint arXiv:2410.09407*.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, and 1 others. 2025. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Ai Han, Junxing Hu, Pu Wei, Zhiqian Zhang, Yuhang Guo, Jiawei Lu, and Zicheng Zhang. 2025. Joyagents-r1: Joint evolution dynamics for versatile multi-llm agents with reinforcement learning. *arXiv preprint arXiv:2506.19846*.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, and 1 others. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. In *ICLR*.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, and 1 others. 2025. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *arXiv preprint arXiv:2505.23885*.

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

Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingtier, and John Irwin. 1997. Aspect-oriented programming. In *European conference on object-oriented programming*, pages 220–242. Springer.

LangChain AI. 2024. Langgraph. https://github.com/langchain-ai/langgraph.

Jiaye Lin, Yifu Guo, Yuzhen Han, Sen Hu, Ziyi Ni, Licheng Wang, Mingguang Chen, Hongzhang Liu, Ronghao Chen, Yangfan He, and 1 others. 2025. Se-agent: Self-evolution trajectory optimization in multi-step reasoning with llm-based agents. *arXiv preprint arXiv:2508.02085*.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*.

Microsoft. 2024. Semantic kernel. https://github.com/microsoft/semantic-kernel.

OpenAI. 2024. Swarm. https://github.com/openai/swarm.

OpenAI. 2025. Openai agents sdk. https://github.com/openai/openai-agents-python.

Pydantic. 2024. Pydantic ai. https://github.com/pydantic/pydantic-ai.

Yexuan Shi, Mingyu Wang, Yunxiang Cao, Hongjie Lai, Junjian Lan, Xin Han, Yu Wang, Jie Geng, Zhenan Li, Zihao Xia, and 1 others. 2025. Aime: Towards fully-autonomous multi-agent framework. *arXiv preprint arXiv:2507.11988*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*.

Chengyue Yu, Siyuan Lu, Chenyi Zhuang, Dong Wang, Qintong Wu, Zongyue Li, Runsheng Gan, Chunfeng Wang, Siqi Hou, Gaochi Huang, and 1 others. 2025. Aworld: Orchestrating the training recipe for agentic ai. *arXiv preprint arXiv:2508.20404*.

Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Dongsheng Li, and Deqing Yang. 2025. Evoagent: Towards automatic multi-agent generation via evolutionary algorithms. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6192–6217.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. 2025. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. 2024. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237.

Zhanke Zhou, Chentao Cao, Xiao Feng, Xuan Li, Zongze Li, Xiangyu Lu, Jiangchao Yao, Weikai Huang, Tian Cheng, Jianghangfan Zhang, and 1 others. 2026a. Alphaapollo: A system for deep agentic reasoning. In *ICLR 2026 Workshop on Lifelong Agents: Learning, Aligning, Evolving*.

Zhanke Zhou, Xiao Feng, Zhaocheng Zhu, Jiangchao Yao, Sanmi Koyejo, and Bo Han. 2025. From passive to active reasoning: Can large language models ask the right questions under incomplete information? In *International Conference on Machine Learning*, pages 78714–78758.

Zhanke Zhou, Zhaocheng Zhu, Xuan Li, Mikhail Galkin, Xiao Feng, Sanmi Koyejo, Jian Tang, and Bo Han. 2026b. Landscape of thoughts: Visualizing the reasoning process of large language models. In *The Fourteenth International Conference on Learning Representations*.

A Appendix

A.1 More Features of OxyGent

Beyond the core orchestration and evolution pipelines, OxyGent provides a suite of tools designed for deep inspection and real-time intervention in AI decision-making.

Interactive Decision Exploration. OxyGent allows researchers to pause execution at any stage to analyze an agent's internal state, including LLM prompts, memory snapshots, and pending tool invocations. Developers can modify variables and re-sample inference paths on-the-fly without restarting the entire system. This enables parallel experimentation with different models or prompts, in which each variant is automatically recorded for seamless comparison and rollbacks, as shown in Figure 10.

Cognitive Transparency. To address the black-box nature of multi-agent systems, OxyGent provides end-to-end visibility from high-level strategies to atomic tool operations. As shown in Figure 11, the system automatically constructs traceable decision graphs that capture the rationale behind each action. This Git-like versioning of agent reasoning enables transparent collaboration and allows intermediate nodes to be regenerated under modified configurations to verify the system's adaptability.

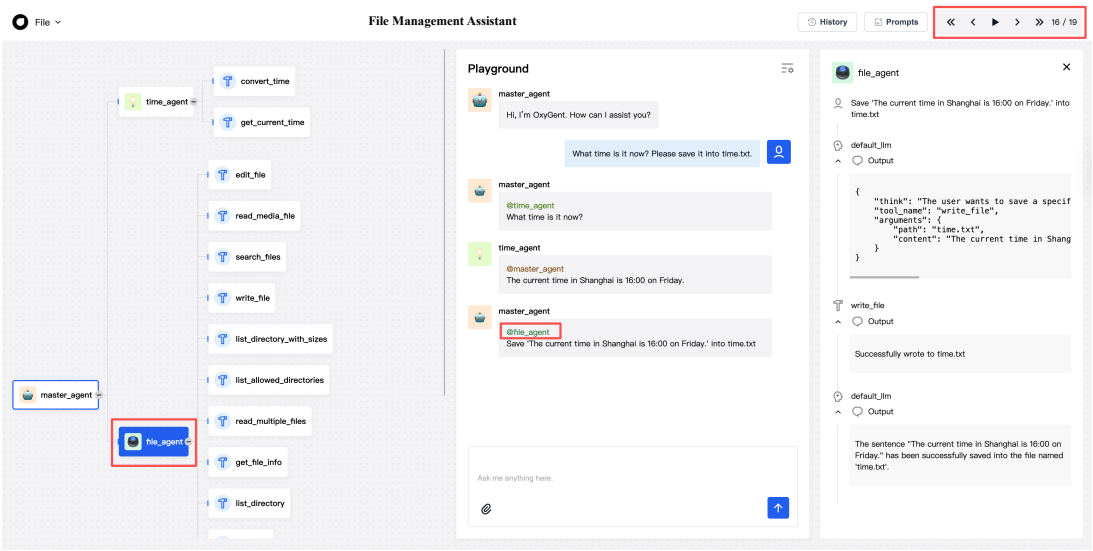


Figure 10: During runtime, the calling nodes on the left will be highlighted. When the inference is complete, any intermediate step can be retraced using the progress button in the upper right corner.

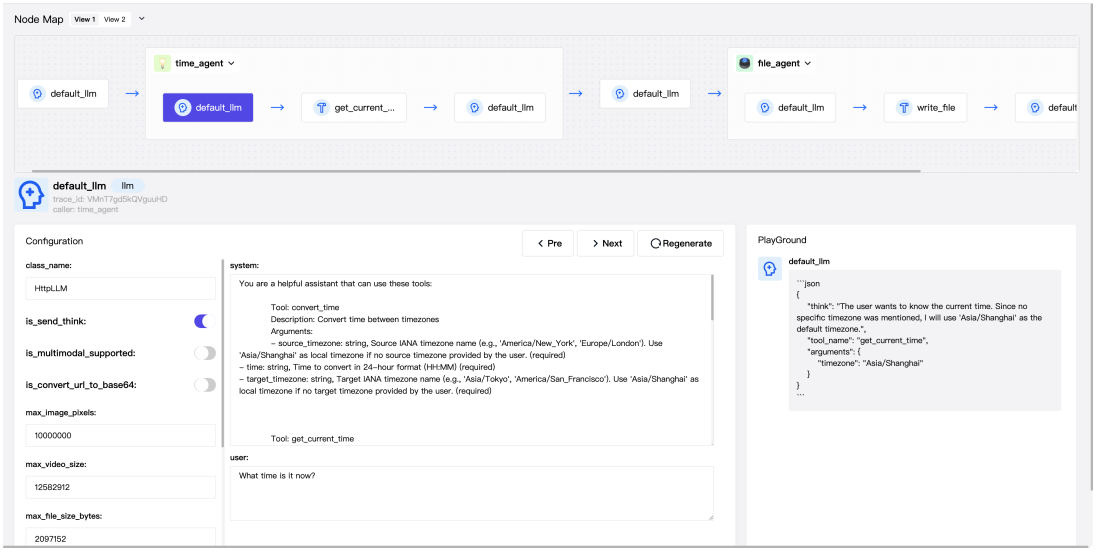


Figure 11: OxyGent can automatically generate traceable decision graphs in real time. It also supports configuration modification and regeneration of any intermediate node, making the MAS more transparent and operable.

A.2 Details of OxyGent-GAIA

The OxyGent-GAIA is built based on DeepSeek-R1 (Guo et al., 2025), GPT-4o (Hurst et al., 2024), and Claude-3.5-Sonnet (Anthropic, 2024). The agentic hierarchy and tool distribution are detailed below:

- **Master Agent** (DeepSeek-R1): The root orchestrator that initiates and governs the entire workflow, managing the following three primary functional units:
  - **Task Agent** (GPT-4o): Triggered first by the Master to perform high-level task decomposition and generate the overall strategic plan for the request.
  - **Coordinator Agent** (GPT-4o): Manages the specific coordination and execution phase by delegating sub-tasks to its specialized sub-agents:
    - **Web Agent** (GPT-4o): Equipped with the *SearchToolkit* (Google, Wikipedia, revisions, and archives), *DocumentProcessingToolkit*, *AsyncBrowserToolkit*, and *VideoAnalysisToolkit* for real-time information retrieval.
    - **Document Processing Agent** (GPT-4o): Specialized in multimodal content extraction using toolkits for documents, images, audio, video, and general code execution.
    - **Reasoning Coding Agent** (GPT-4o): Focused on complex logic and data manipulation. It utilizes the *CodeExecutionToolkit*, powered specifically by Claude-3.5-Sonnet, along with the *ExcelToolkit* and *DocumentProcessingToolkit*.
  - **Answerer Agent (GPT-4o):** Responsible for the final response synthesis. Due to GAIA's strict formatting requirements, this agent ensures precision and has the authority to reject and re-delegate the task if the reasoning evidence is deemed insufficient.

A.3 Extended Case Studies

A.3.1 Commercial Application Patterns

The OxyGent framework has established several standardized patterns for industrial MAS deployment:

- **Standard Operating Procedures (SOP):** Agents are assigned to specific workflow segments (*e.g.*, approval, data validation). Top-level agents monitor progress while lower-level agents execute tasks, improving efficiency in standardized business processes like contract auditing.
- **Advanced RAG Pipelines:** A multi-layered collaboration where specialized agents handle retrieval, answer generation, and quality review separately, significantly reducing hallucinations in enterprise knowledge bases.
- **Automated Data Analytics:** OxyGent orchestrates agents to perform end-to-end data collection, cleaning, modeling, and visualization, streamlining the generation of automated business reports and anomaly detection.
- **Cross-System Tool Orchestration:** In complex DevOps and office automation scenarios, agents dynamically select and invoke external APIs, maintaining system state across multiple platforms.

A.3.2 Community and Developer Feedback

Open-source community feedback highlights the extensibility of OxyGent across diverse niche domains:

- **Natural Language to SQL (NL2SQL):** Developers have utilized OxyGent to build multi-agent clusters that convert natural language queries into complex SQL statements with multi-dimensional consistency verification.
- **Low-Code Automation:** Integration with tools like *JoyCode* has enabled users to perform web searches and file operations via natural language.
- **Database Optimization:** Autonomous agents built on OxyGent are currently used to diagnose "slow SQL" queries and generate governance suggestions to improve system performance.
- **Dynamic Flowchart Generation:** Using local API calls, agents can automatically generate and render core implementation logic into visual flowcharts for better developer communication.