

OpenManus: Empowering LLM Agent Applications Via Framework and Capability Evolution

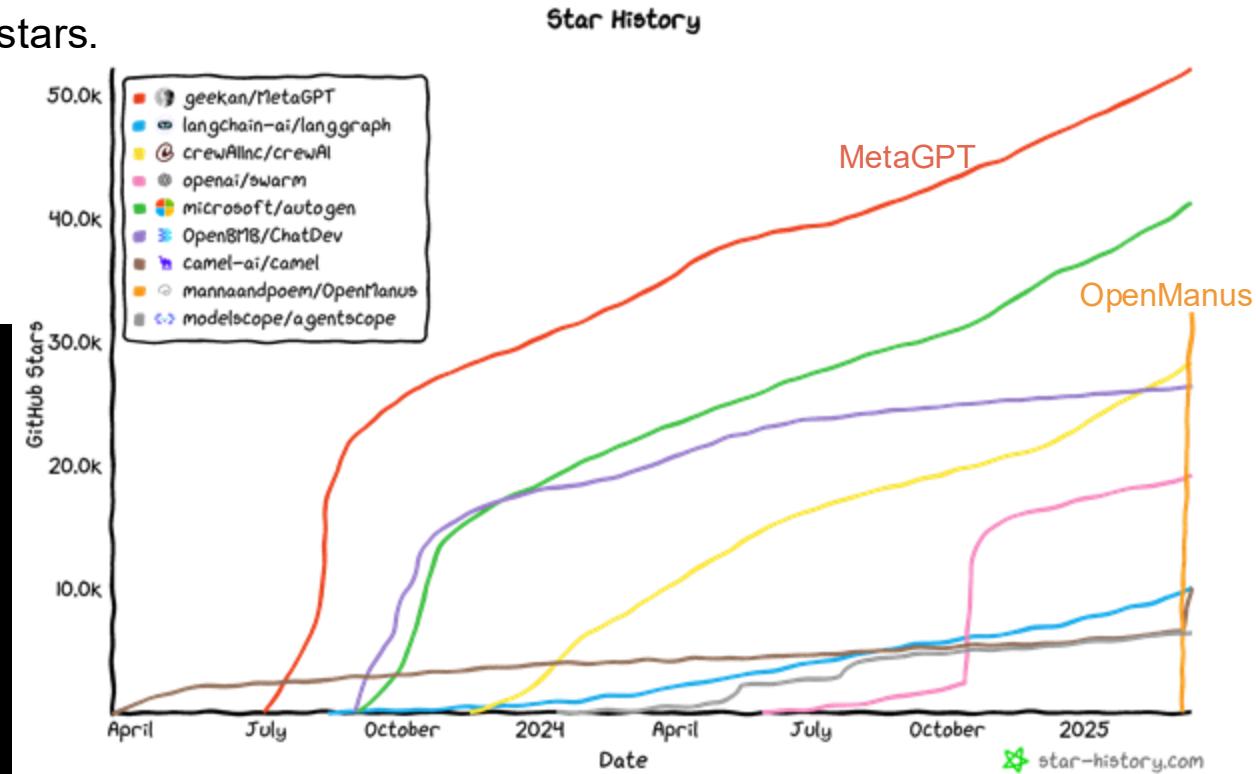
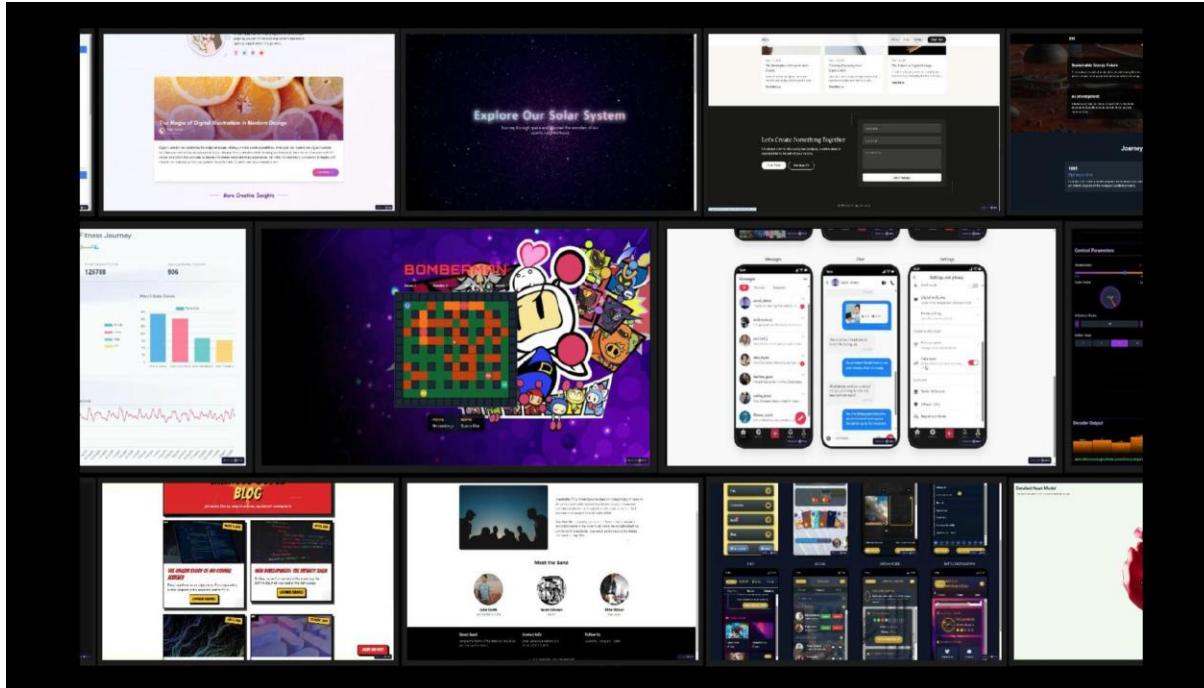
Sirui Hong

MetaGPT: Open-Source Multi-Agent Framework Pioneer

GOSIM

MetaGPT provides the first fully AI agent-based software company , with **natural language programming** capabilities, improved the software development efficiency. (Community Size: 10k+ members)

- MetaGPT, the world's top multi-agent framework with **130k+** stars.
- MetaGPT & AFlow have won ICLR Oral twice.
- MetaGPT, topped ProductHunt's ranking.
- OpenManus gained **10k+** stars on GitHub in a day.



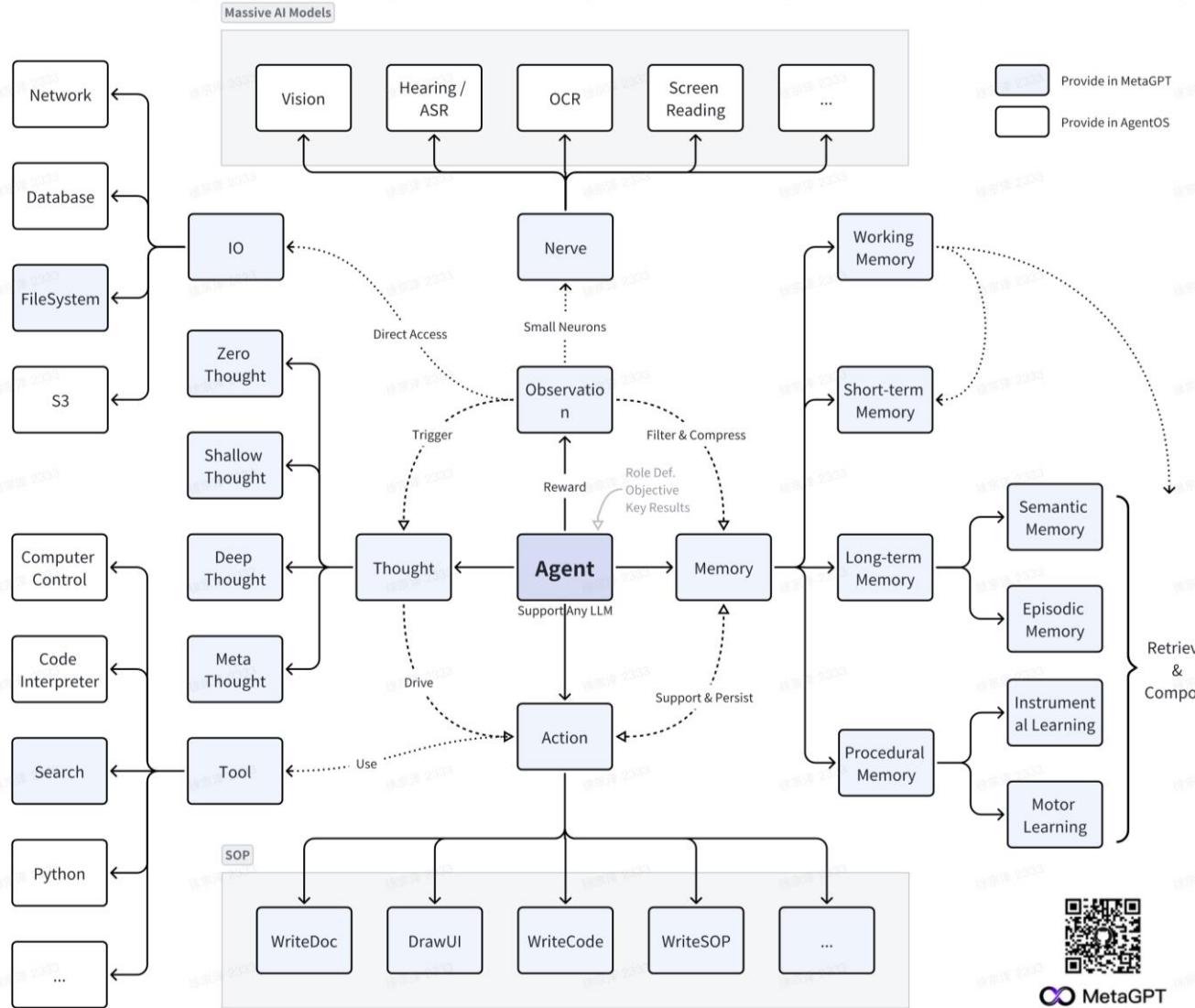
*3-Hour Replication to Production
Explosive Community Adoption*

- **10,000+** stars in first 24 hours
- Active community of **6k+** members

GOSIM AI Paris 2025

What is AI Agent?

GOSIM



Foundation Agent

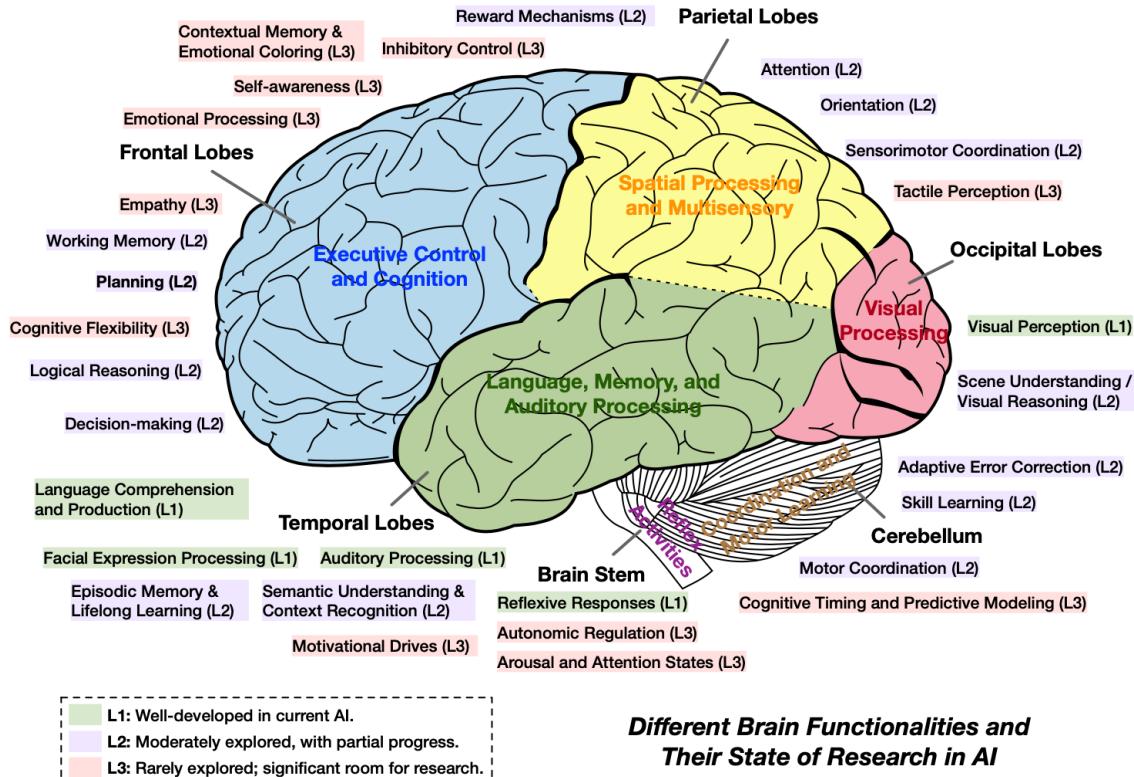
An autonomous intelligent system:

- Actively perceives diverse environmental signals
- Continuously learns from experiences
- Updates internal states such as memory/emotion
- Reasons purposefully about actions (both external and internal)
- Navigates autonomously toward complex, long-term objectives
- Adapts dynamically to changing conditions and requirements



What is AI Agent?

The brain-inspired modules optimize core capabilities and continuously optimize



Each module handles a specific cognitive function, similar to how the human brain is organized.

The “General + Professional” dual-layer design balances versatility and expertise.

[“Advances and Challenges in Foundation Agents: From Brain-Inspired Intelligence to Evolutionary, Collaborative, and Safe Systems.” arXiv preprint arXiv:2504.01990 \(2025\)](#)

Author Affiliation: MetaGPT, Montreal & MILA, NTU, ANL, U Sydney, PSU, MSRA, UIUC, HKUST, USC, Yale, Stanford, UGeorgia, OSU, KAUST, Duke, PIIyU, Google DeepMind, Canada CIFAR AI Chair

GOSIM AI Paris 2025



Chatbots

AI with conversational language



Emerging

equal to or somewhat better than an unskilled human

Level 1

Reasoners
human-level problem solving

Competent
at least 50th percentile of skilled adults

Level 3

Agents
systems that can take actions

Expert
at least 90th percentile of skilled adults

Level 4

Innovators
AI that can aid in invention

Virtuoso
at least 99th percentile of skilled adults

Level 5

AI that can do the work of an organization

Superhuman
outperforms 100% of humans

AI System Level

Level of AGI

Agents: Revolutionizing Information Access & Interaction

GOSIM

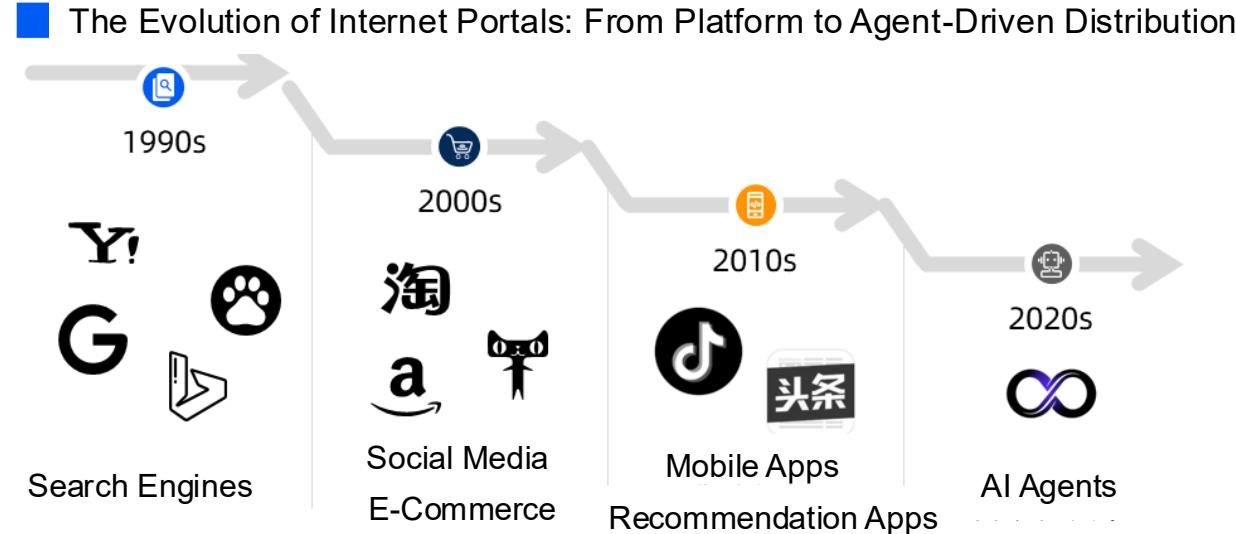
- Human-Machine Interaction Revolution: Simplifying Development and Workflow Processes



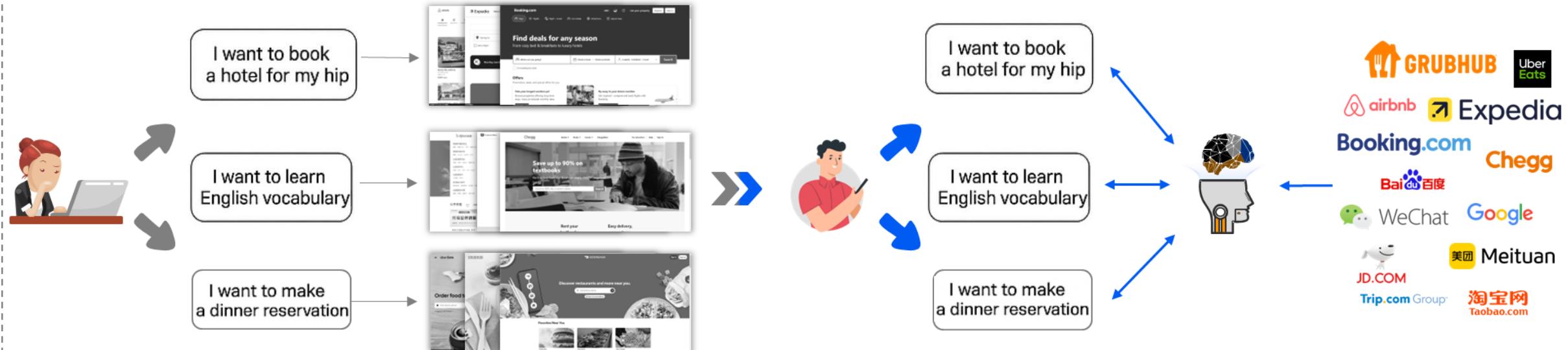
Traditional Menu-Based Interface



Agent-Based Interface



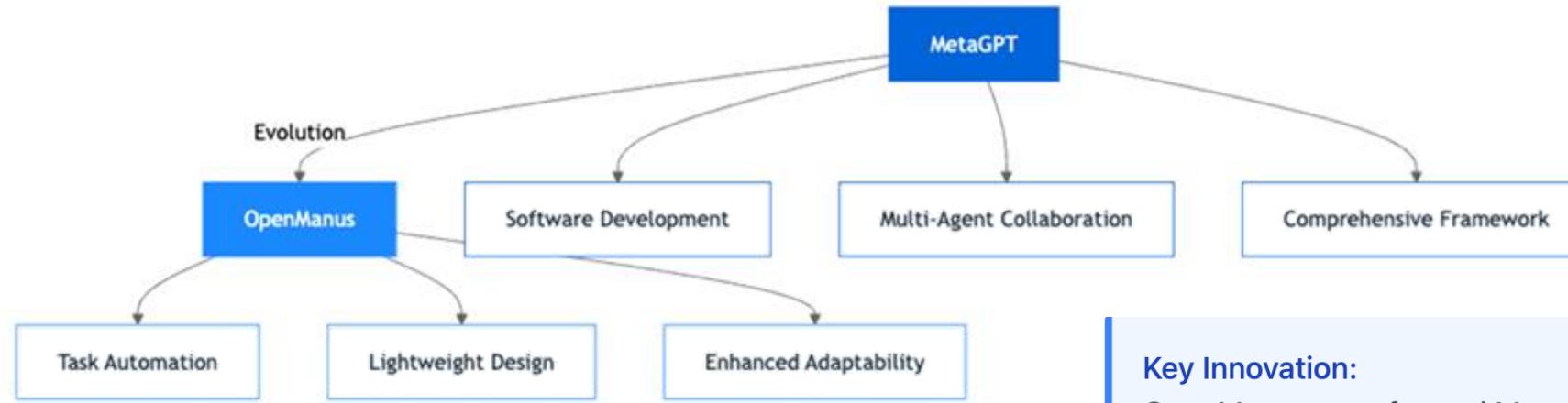
While software needs will continue to exist, the way users interact with software must evolve.



From MetaGPT to OpenManus

GOSIM

MetaGPT → OpenManus



Key Innovation:

OpenManus transformed MetaGPT's comprehensive framework into a streamlined tool focused on adaptability, autonomy, and cross-environment operation.

Integration Points

Tool Selection

Plans incorporate tool recommendations, suggesting which specific tools are appropriate for each step

Memory Management

Integrates with the agent's memory systems to retrieve relevant context and store execution results

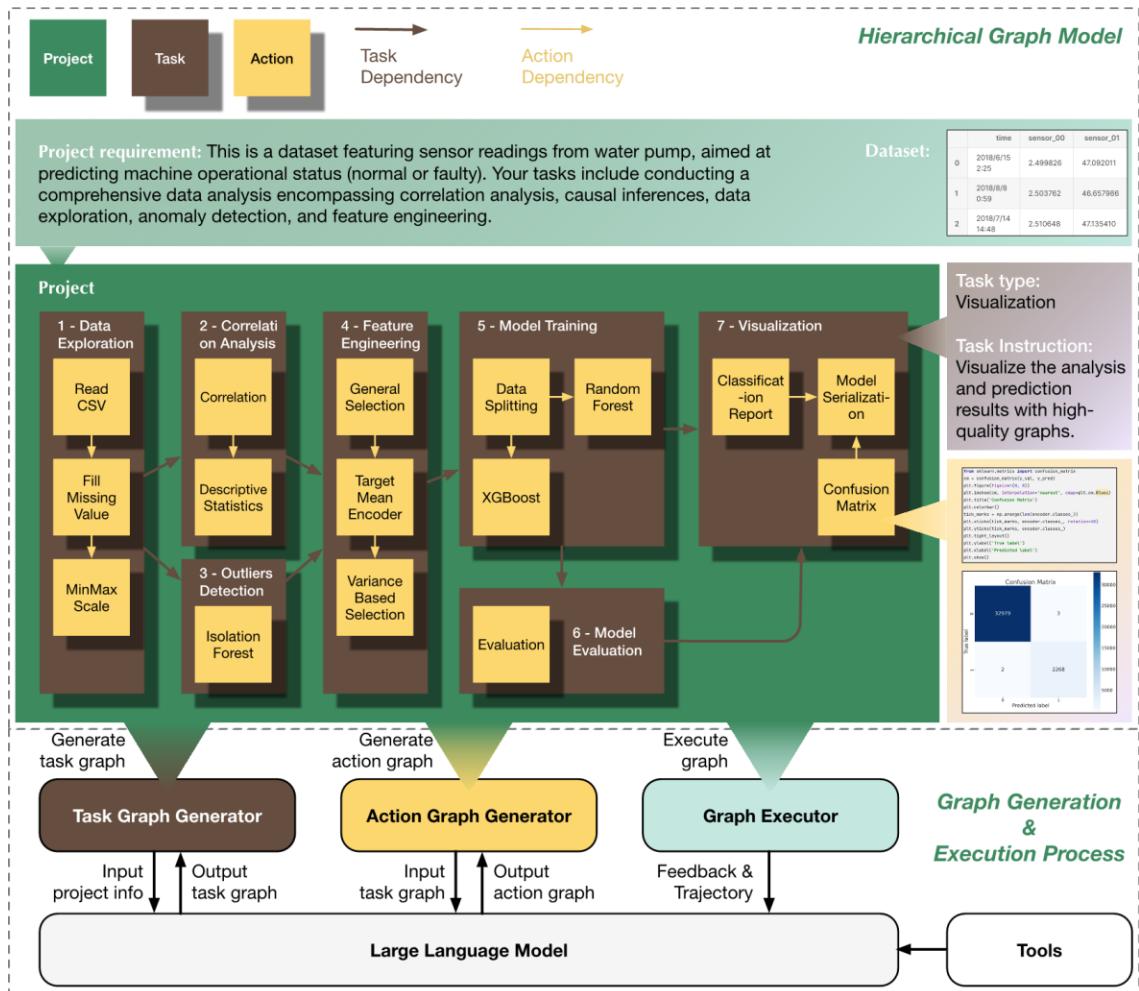
Multi-Agent Coordination

Plans can be shared across multiple specialized agents to enable collaborative task execution



OpenManus: A Foundation of Planning and Tool Usage

GOSIM



Hierarchical Planning Structure

["Data Interpreter: An LLM Agent For Data Science."](#) arXiv preprint arXiv:2402.18679 (2024).

Author Affiliation: MetaGPT, Montreal & MILA, NTU, Yale HKUST, KAUST, Fudan, Qinhua

PlanningTool Functionality

Plan Creation

Analyzes user requests to formulate structured execution plans with numbered steps, creating a clear roadmap for task completion.

Plan Management

Offers commands for listing, retrieving, and updating plans, maintaining a central registry of all ongoing tasks.

Step Execution Tracking

Monitors progress by marking steps as completed, failed, or in-progress, providing visibility into execution status.

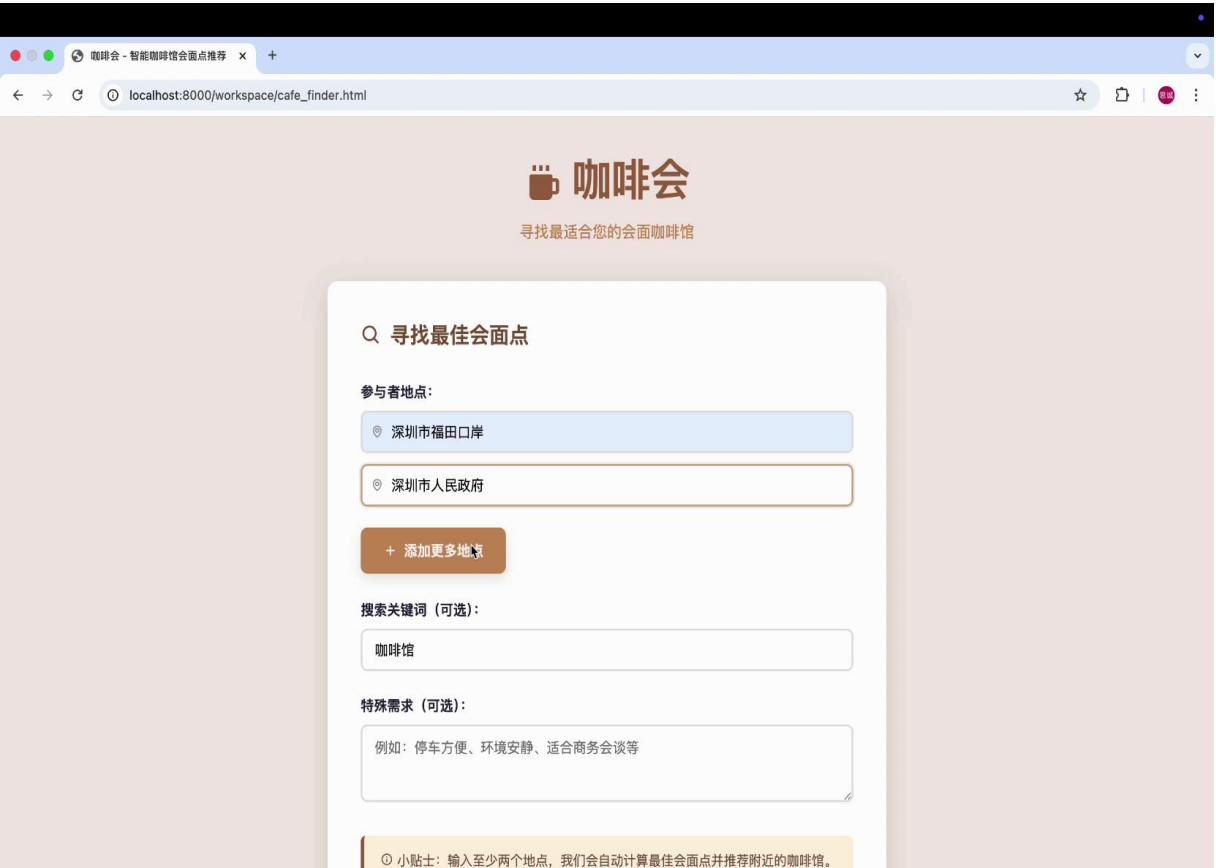
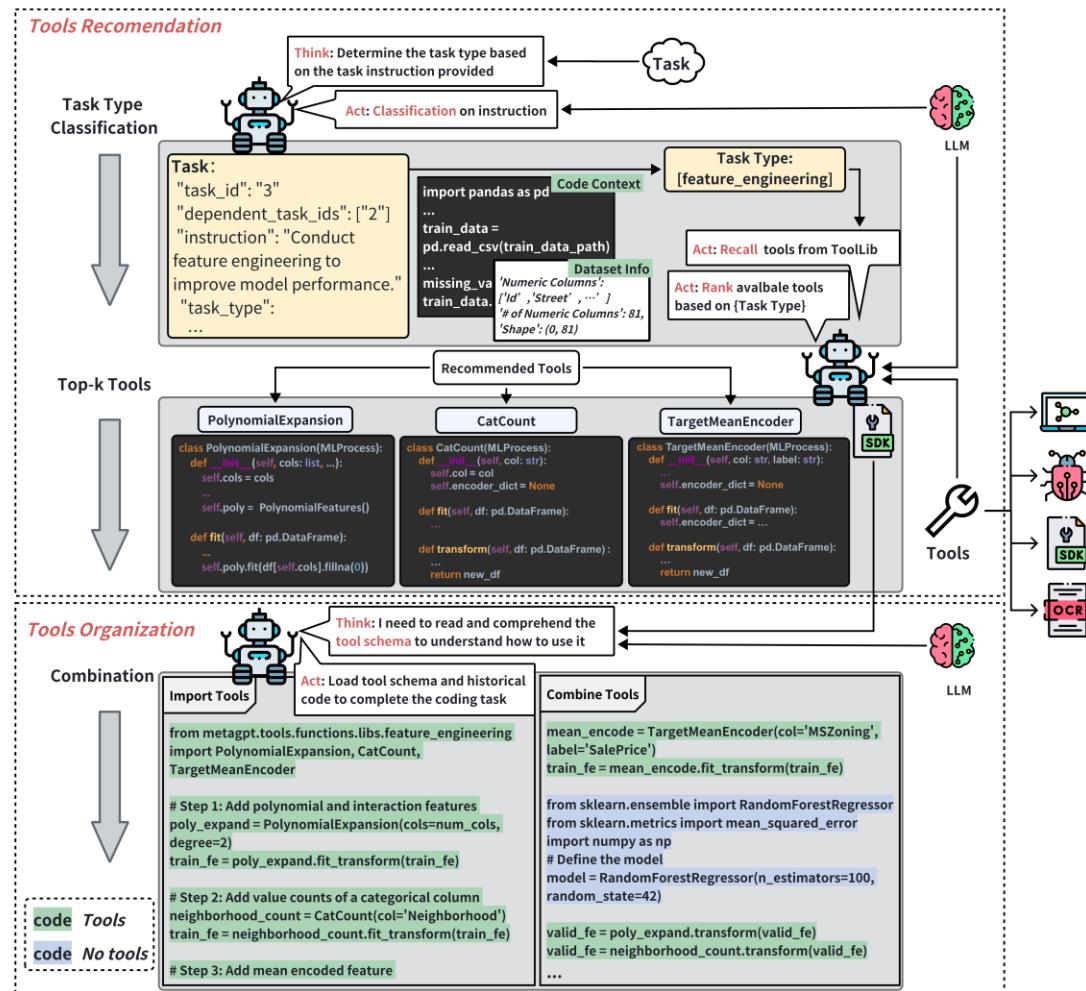
Plan Status Updates

Allows agents to update the overall plan status and record outcomes, ensuring accurate state representation.



OpenManus: A Foundation of Planning and Tool Usage

GOSIM



Integration with Tools and Execution Flow

["Data Interpreter: An LLM Agent For Data Science," arXiv preprint arXiv:2402.18679 \(2024\).](#)

Author Affiliation: MetaGPT, Montreal & MILA, NTU, Yale HKUST, KAUST, Fudan, Qinhua

GOSIM AI Paris 2025

OpenManus: A Foundation of Planning and Tool Usage

GOSIM

ToolAct = ReAct + Tools + Prompt



ToolAct

Completes tasks through combinations of predefined tool calls, offering better safety and control but relatively lower flexibility



OpenManus Implementation

In OpenManus, ToolAct is implemented through carefully designed Prompts and rich Tools to achieve powerful automation capabilities, enabling AI agents to perform complex tasks.



```
# ToolAct implementation example in OpenManus
from openmanus.agent import ToolActAgent
from openmanus.tools import BrowserUseTool, WebSearch

# Define Agent's toolset
tools = [
    BrowserUseTool(), # Browser automation
    WebSearch()        # Web search
]

# Define Agent's behavior logic
prompt = """You are a web research assistant
who helps users find and organize online
information. When facing questions, first
try searching for latest information, then
use browser to get detailed content."""

# Create ToolAct Agent
agent = ToolActAgent(
    tools=tools,
    system_prompt=prompt
)
```



OpenManus: A Foundation of Planning and Tool Usage

GOSIM



Computer-Use Functionality

This tool enables AI agents to interact with computer systems like human users, performing various system-level operations to automate complex workflows. Similar to the computer operation capabilities in Anthropic's Claude models.

- ✓ File system operations (create, read, write, delete)
- ✓ Directory traversal and file finding capabilities
- ✓ System command execution and process management
- ✓ Application control and automation
- ✓ System resource monitoring and management

Shell Command Integration

OpenManus uses a **command-line shell-based approach** similar to SWE agent design, allowing direct access to system resources through shell commands.

Bash command execution

System monitoring

Script execution & generation

Sandboxed environment

File Management Capabilities

File Creation & Editing

Generate and modify files through shell commands like cat, echo, and sed

Directory Management

Create, navigate, and organize directories with mkdir, cd, and ls

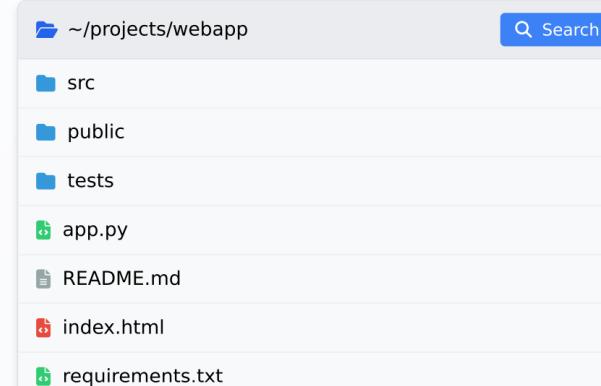
Content Search

Locate specific content using grep, find, and awk commands

Batch Processing

Execute operations on multiple files using wildcards and loops

File Browser Demonstration



```
OpenManus Shell
openmanus@agent:~$ find ./projects -name "*.py" | grep "model"
./projects/webapp/src/models/user_model.py
./projects/webapp/src/models/data_model.py

openmanus@agent:~$ cat ./projects/webapp/src/models/user_model.py
| head -5
class UserModel:
    def __init__(self, username, email):
        self.username = username
        self.email = email
        self.active = True

openmanus@agent:~$ mkdir -p ./projects/webapp/output
```

OpenManus: A Foundation of Planning and Tool Usage

GOSIM



Web Search Tools

- ✓ Supports Google search and web query methods
- ✓ Real-time information retrieval, bypassing knowledge cutoff limitations
- ✓ Smart search result parsing and information extraction
- ✓ Multi-language search and result translation



Browser-Use Tool

- ✓ Playwright-based browser automation tool
- ✓ Mimics human web interaction (clicking, typing, scrolling)
- ✓ Supports form filling, login, data extraction and complex operations
- ✓ Can capture screenshots and parse webpage content
- ✓ Supports multi-tab and multi-session management

Search Tool Usage Example

```
# Using WebSearch tool to get real-time information
from openmanus.tools import WebSearch

search_tool = WebSearch()
results = search_tool.run(
    query="Latest AI technology developments",
    num_results=5
)

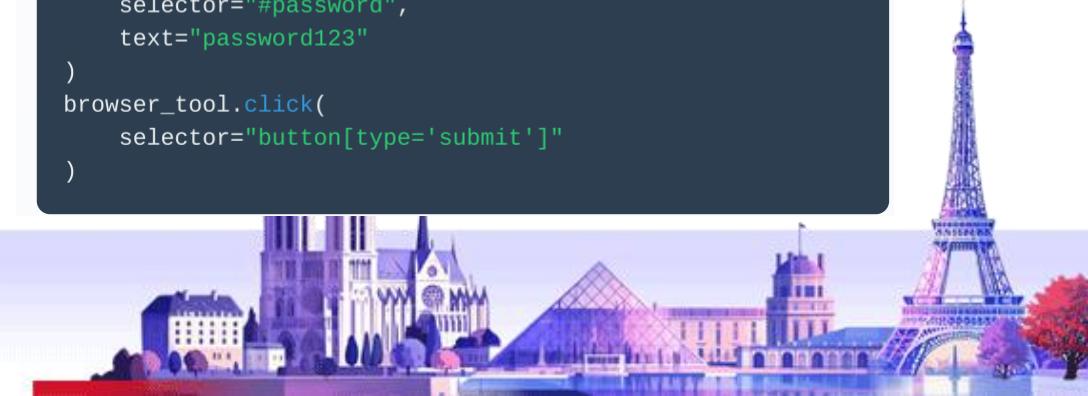
# Process search result
for result in results:
    print(result.title,
```

Browser-Use Example

```
# Using Browser-Use tool to automate website login
from openmanus.tools import BrowserUseTool

browser_tool = BrowserUseTool()
browser_tool.visit("https://example.com/login")

# Perform login operation
browser_tool.fill(
    selector="#username",
    text="user@example.com"
)
browser_tool.fill(
    selector="#password",
    text="password123"
)
browser_tool.click(
    selector="button[type='submit']"
)
```



OpenManus: Challenges & Future Prospects

GOSIM



Multi-modality & Tool Integration

- Cross-modal understanding for intelligent tool selection
- Unified representation learning for seamless integration
- Development of specialized multi-modal tools
- Multi-modal feedback loops for tool usage optimization



Long-term Planning & Execution

- Recursive planning with dynamic self-correction
- Memory-enhanced planning for consistent execution
- Hierarchical planning architecture (strategic to operational)
- Error detection and recovery in long-running tasks



Social Interaction & Collaboration

- Adaptive social interaction models for diverse scenarios
- Negotiation and consensus-building mechanisms
- Dynamic role assignment based on task requirements
- Trust and reputation systems for optimal collaboration



Open vs. Closed Source Integration

- Hybrid model architectures leveraging best of both worlds
- Complementary capabilities: open for basic, closed for complex
- Modular interfaces for seamless model substitution
- Federated learning and knowledge distillation strategies



THANK YOU

