International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

AI Agents: Agent GPT

SHIVA SUMANTH REDDY
Assistant Prof
sumanthdsatm@gmail.com

REVANSIDDA 1DT22CS123
vbsiddu2003@gmail.com

KISHAN G
1DT22CS077
kishangirish9036@gmail.com

KUSHAL S
1DT22CS079
kush110202@gmail.com

Abstract - Agent GPT is an advanced autonomous AI system designed to simulate human-like reasoning and task execution through the deployment of AI agents. Unlike traditional large language models that respond passively to user prompts, Agent GPT can plan, iterate, and execute multi-step goals with minimal human intervention.

Each agent operates based on a defined objective, breaking it down into smaller tasks, leveraging APIs, tools, or internet access to gather information, and adapting dynamically to changing conditions. The system is often built on top of language models such as GPT-4, and incorporates features like memory, tool use, and recursive task execution to complete complex workflows.

Agent GPT is used in applications ranging from automated research and customer support to software development and marketing strategy generation. Its core innovation lies in enabling models to act not just as conversational tools, but as autonomous problem solvers capable of taking initiative, learning from context, and optimizing toward defined goals.

Key Features:

- AI-Driven Code Generation: Utilizes cutting- edge large language models to interpret natural language prompts and generate complete, production-ready code for a variety of applications.
- In-Browser Development Environment: Employs containerized technology to provide a full-stack environment entirely in the browser, allowing real-time editing, execution, and debugging.
- Seamless Deployment: Offers one-click deployment integrations with platforms like Netlify and Cloudflare for effortless publishing.
- Framework Support: Compatible with popular frameworks such as React, Next.js, Vue, Angular, and Node.js, accommodating diverse developer needs.
- Collaboration Capabilities: Supports real-time collaboration, enabling teams to build and iterate together directly within the browser.

INTRODUCTION

In recent years, the evolution of large language models (LLMs) such as OpenAI's GPT series has significantly advanced the field of natural language processing (NLP). These models have demonstrated impressive abilities to understand and generate human-like text, enabling a wide range of applications—from chatbots and content creation to coding assistance and data analysis.

However, most traditional LLMs operate in a reactive manner—responding to prompts without autonomy, memory, or long-term goal management. To address these limitations, Agent GPT introduces a new paradigm: autonomous AI agents capable of independently reasoning, planning, and executing complex tasks.

Agent GPT builds upon the foundation of LLMs by integrating advanced features such as:

- Self-directed task decomposition
- Iterative problem-solving
- Contextual memory
- Dynamic tool use

It simulates a human-like approach to task management: given a high-level objective, the agent autonomously

breaks it down into subtasks, prioritizes actions, gathers relevant information, and adjusts strategies based on real- time feedback. This transforms Agent GPT from a simple interactive model into a fully autonomous system— capable of initiating actions, making decisions, and learning from outcomes.

Step Toward General-Purpose AI
Context Awareness and Memory

The agent retains contextual information during execution to maintain coherent, goal-aligned behavior. In some implementations, it supports long-term memory, enabling it to remember user preferences, past actions, or conversations.

The development of Agent GPT marks a significant milestone in the journey toward general-purpose AI systems. By enabling persistent, goal-driven agents can interact with:

- Digital environments
- External APIs
- Other autonomous agents

Agent GPT opens doors to numerous applications— ranging from automated business workflows and research to software project management and creative assistance. As these agents evolve, Agent GPT not only pushes technological boundaries but also serves as a testbed for exploring the limits of autonomous artificial intelligence.

Key Characteristics of Agent GPT

Autonomous Task Execution

Agent GPT operates independently after receiving a high-level goal. It decomposes the objective into actionable subtasks and executes them sequentially, minimizing the need for human intervention.

Recursive Reasoning and Planning

The system continuously evaluates outcomes, refining its plan and strategy in real time. This allows it to adapt to dynamic or complex tasks as they unfold.

Tool and API Integration

Agent GPT can interface with external tools, APIs, and databases—allowing it to:

- Search the web
- Analyze documents
- Perform calculations
- Control or interact with external software systems

I. Literature survey

The emergence of large language models (LLMs) like GPT-3 and GPT-4 has revolutionized natural language processing (NLP), enabling machines to understand, generate, and interact in human-like language. However, early generations of LLMs operated reactively, requiring explicit user input for each interaction, lacking autonomy, memory, and the capacity for long-term planning. This limitation led to the exploration of autonomous AI agents, designed to extend the capabilities of LLMs by enabling proactive, goal-driven behavior.

1. Large Language Models (LLMs) and Their Limitations
Early studies, such as Brown et al. (2020) with GPT-3, demonstrated that transformer-based models could generalize across diverse tasks through few-shot and zero- shot learning. Despite these advancements, models remained stateless and non-autonomous, unable to independently decompose tasks or persist contextual understanding across sessions.
- Reference: Brown, T. et al. (2020). *Language Models are Few-Shot Learners*. arXiv preprint arXiv:2005.14165.

2. Towards Agent-Based LLMs
In response to these limitations, researchers began developing agent-based systems where LLMs act as cognitive cores within autonomous frameworks. Systems like AutoGPT, BabyAGI, and AgentGPT emerged in 2023 as prototypes demonstrating how LLMs could plan, execute, and adapt based on long-term goals.
- AutoGPT (Toran Bruce Richards, 2023) introduced the concept of self-prompting and recursive reasoning, allowing the LLM to refine its tasks continuously.
- BabyAGI built lightweight task managers for LLMs to operate as cognitive agents with prioritization and memory.
- AgentGPT extended this by integrating tool use, persistent memory, and web interaction for autonomous task execution.
- Reference: Richards, T. (2023). *AutoGPT GitHub Repository*. https://github.com/Torantulino/Auto- GPT

3. Memory and Contextual Awareness

Studies on long-term memory integration (e.g., LangChain, MemGPT) address the challenge of statelessness in LLMs. These architectures allow agents to retain context across sessions, enabling deeper task understanding, personalization, and sustained multi-turn interactions.
- Reference: Xie, S. et al. (2023). *LangChain: A Framework for Developing LLM-Powered Applications*. LangChain Documentation.
- Reference: Gurnee, M. et al. (2023). *MemGPT: Towards a Memory-Augmented LLM*. arXiv preprint arXiv:2310.08572.

4. Tool Use and API Interaction

Another key area of research is LLM-enabled tool use. Models such as ReAct (Yao et al., 2022) combine reasoning and acting by alternating between thinking (via natural language) and invoking tools (e.g., search APIs, calculators). This architecture underpins many autonomous agents today.
- Reference: Yao, S. et al. (2022). *ReAct: Synergizing Reasoning and Acting in Language Models*. arXiv preprint arXiv:2210.03629.

5. Planning and Recursive Task Management

Recursive task management enables an agent to evaluate outcomes, generate follow-up steps, and refine strategy—crucial for long-horizon goals. Planning techniques, inspired by Hierarchical Task Networks (HTNs) and reinforced through prompt engineering, allow LLMs to simulate complex workflows.
- Reference: Nye, M. et al. (2021). *Show Your Work: Scratchpads for Intermediate Computation with Language Models*. arXiv preprint arXiv:2112.00114.

III. Methodology

The methodology for understanding and evaluating Agent GPT involves both **architectural analysis** and **functional experimentation**.

1. System Architecture Design
Agent GPT is built on top of large language models (LLMs) such as GPT-4. The architecture includes several integrated modules:
- **Task Manager**: Accepts a high-level objective and decomposes it into actionable subtasks.
- **Planner Module**: Uses recursive reasoning to sequence tasks based on priorities, dependencies, and real-time outcomes.
- **Memory System**: Maintains contextual and long- term

memory to preserve task continuity and personalization.
• **Tool/Plugin Interface**: Connects with APIs, web tools, and external software to perform actions beyond language generation.
• **Executor Agent**: Executes each task autonomously while monitoring progress and adjusting strategies as needed.

2. Implementation Tools
The system can be implemented using:
• **OpenAI GPT-4 API / Claude / Gemini models**
• **LangChain / AutoGPT frameworks**

• **Python** for scripting task logic and API integration

• **Vector databases** (e.g., Pinecone, FAISS) for memory management
• **Webhooks/APIs** for external tool interaction

3. Workflow Execution
The process followed by Agent GPT typically involves:
1. **Input Handling**: Receive a high-level goal from the user.
2. **Task Decomposition**: Break the goal into structured subtasks.
3. **Planning and Prioritization**: Organize the subtasks based on logical sequence and dependencies.
4. **Tool Invocation**: If a task requires web search, data analysis, or code execution, the agent uses the appropriate tools.
5. **Contextual Adjustment**: Based on results, the agent refines its plan, repeats steps if needed, or explores alternatives.
6. **Memory Update**: Contextual memory is updated to retain key learnings, results, and user preferences.

IV. Conclusion

The development of **Agent GPT** marks a transformative shift in the use of large language models—from passive, reactive systems to **proactive, autonomous agents** capable of handling complex tasks with minimal supervision. By integrating planning, memory, tool usage, and recursive reasoning, Agent GPT transcends traditional LLM capabilities, paving the way for **general-purpose AI assistants**.

This advancement holds significant promise across diverse sectors, including business automation, research, education, software development, and customer support.

However, challenges remain—particularly in ensuring **reliability**, **ethical decision-making**, **explainability**, and **handling ambiguous or sensitive tasks**.
Future research will likely focus on enhancing **multi-agent collaboration**, improving **long-term memory reliability**, and ensuring **safe autonomy** in high-stakes domains. As autonomous AI agents continue to evolve, Agent GPT provides a robust foundation and a glimpse into the future of intelligent, self-directed systems.

In addition to technical advancements, the adoption of Agent GPT also raises important **societal and ethical considerations**. Autonomous agents acting on behalf of users require safeguards to prevent misuse, protect privacy, and ensure alignment with human values. Misinterpretation of goals, unchecked access to external systems, or manipulation of data could result in unintended consequences. As such, designing responsible agent behaviors, incorporating human-in-the-loop systems, and enforcing strict control over permissions and actions will be critical as these systems scale and become more widely integrated into daily workflows.\

Moreover, the evolution of Agent GPT signifies a foundational shift toward **human-AI collaboration**. These agents are not intended to replace human intelligence but rather to augment human capability by offloading repetitive or complex tasks, accelerating problem-solving, and enabling users to focus on higher-level strategic thinking. As interfaces improve and trust in such systems grows, Agent GPT and similar frameworks could play a central role in democratizing access to knowledge, automating innovation pipelines, and reshaping how individuals and organizations approach work and creativity in the AI era.

REFERENCES

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, *et al.*, "Language Models are Few- Shot Learners," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[2] OpenAI, "GPT-4 Technical Report," 2023. [Online]. Available: https://openai.com/research/gpt-4

[3] Y. Nakajima, "BabyAGI: A Simplified AI Task Management System Powered by GPT," *GitHub*, 2023. [Online]. Available: https://github.com/yoheinakajima/babyagi

[4] Torantulino, "Auto-GPT: An Experimental Open- Source Attempt to Make GPT-4 Fully Autonomous," *GitHub*, 2023. [Online]. Available: https://github.com/Torantulino/Auto-GPT

[5] J. Mouro, "CrewAI: A Multi-Agent Orchestration Framework for Autonomous AI Agents," *GitHub*, 2024. [Online]. Available: https://github.com/joaomdmoura/crewAI

[6] LangChain, "LangChain Documentation," 2023. [Online]. Available: https://docs.langchain.com/

[7] Pinecone, "Pinecone Vector Database Documentation," 2023. [Online]. Available: https://www.pinecone.io/docs/

[8] Chroma, "Chroma: The Open-Source Embedding Database," *GitHub*, 2023. [Online]. Available: https://github.com/chroma-core/chroma

[9] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, *et al.*, "Sparks of Artificial General Intelligence: Early Experiments with GPT-4," *Microsoft Research*, 2023. [Online]. Available: https://arxiv.org/abs/2303.12712

[10] SerpAPI, "Real-Time Google Search API for Scraping Search Results," 2023. [Online]. Available: https://serpapi.com/