Research Paper

Demystifying Agentic AI and OpenAI’s Agent SDK

Prepared By: Abdul Basit Khalsai

Roll Number:00010435

Instructor Name: Sir Aneeq

Institution Name: GIAIC

Course Name: Agentic AI – Q4

Date of Submission: Friday May 30, 2025

Table of Contents

[**Why Read This Paper? (Introduction & Objective)** 3](#_Toc199513606)

[**What Is Agentic AI?** 3](#_Toc199513607)

[**What Are Large Language Models (LLMs)?** 3](#_Toc199513608)

[Definition: 3](#_Toc199513609)

[Capabilities: 4](#_Toc199513610)

[**How Do LLMs Work?** 4](#_Toc199513611)

[Training Process: How LLMs Learn 4](#_Toc199513612)

[Where Is the Knowledge Stored? 6](#_Toc199513613)

[What About Real-Time Data? 6](#_Toc199513614)

[How Do LLMs Search or Respond to Input? 6](#_Toc199513615)

[Tokenization = The Search Engine of LLMs 7](#_Toc199513616)

[**What Is Generative AI?** 7](#_Toc199513617)

[**Are You Confused Between Generative AI & Agentic AI? Let’s Compare** 7](#_Toc199513618)

[**Analogy to Understand** 7](#_Toc199513619)

[**How Can We Build Agentic AI?** 8](#_Toc199513620)

[**Spotlight: What Is OpenAI’s Agents SDK?** 8](#_Toc199513621)

[**Why Use OpenAI’s Agents SDK?** 8](#_Toc199513622)

[**Conclusion** 9](#_Toc199513623)

# **Why Read This Paper? (Introduction & Objective)**

As Artificial Intelligence rapidly evolves, we're moving from simple content generation to intelligent, goal-driven agents. This paper will help you:

* Understand the transition from Generative AI to Agentic AI
* Learn how LLMs (Large Language Models) power both
* Explore how developers can build robust agents using tools like OpenAI’s Agents SDK
* Gain clarity through analogies, visual diagrams, and beginner-friendly explanations

Outcome: By the end, you’ll understand what Agentic AI is, how it's different from traditional Generative AI, and how OpenAI’s SDK empowers developers to build these intelligent agents with ease.

# **What Is Agentic AI?**

Agentic AI refers to AI systems that don’t just respond – they act, plan, and make decisions. These AI agents are capable of:

* Understanding a goal
* Breaking it into steps
* Taking actions autonomously
* Learning or adapting from memory

LLMs (Large Language Models) play a central role here. They serve as the "brain" of Agentic AI—processing language, reasoning, and interpreting instructions.

# **What Are Large Language Models (LLMs)?**

Large Language Models (LLMs) are a type of artificial intelligence trained on massive amounts of text data. Their goal is to understand and generate human-like language.

## **Definition:**

LLMs are advanced AI systems that use machine learning to process, understand, and generate natural language. They are “large” because they are trained on billions of words and have billions (or even trillions) of internal parameters.

## **Capabilities:**

* Understand user instructions or questions
* Generate coherent and fluent text
* Translate between languages
* Summarize or explain information
* Perform logic-based reasoning

Think of LLMs as the "brain" behind both Generative AI and Agentic AI—they provide the language and reasoning ability.

# **How Do LLMs Work?**

(Internal Mechanism, Architecture, Data & Tokens)

LLMs (Large Language Models) don’t memorize answers — they learn patterns in language by training on massive datasets. Here’s how it all works:

## **Training Process: How LLMs Learn**

LLMs are trained on huge collections of text (books, articles, websites, conversations). This teaches the model to predict the next word (token) based on the context.

* Data Source: The training data is static — gathered once and not updated live unless the model is connected to real-time tools.
* Knowledge Storage: Instead of storing facts, LLMs compress patterns and relationships in their neural network parameters (like long-term memory made of math).
* **Model Architectures Used for LLMs**:

LLMs can be built using different neural architectures — each with their own learning mechanism:

* + **Transformer**:
    - **How It Learns**: Transformers look at **all the words in a sentence at once** and learn how each word relates to the others using a technique called **attention**. For example, when reading “The cat sat on the mat,” it learns that "cat" is connected to "sat" and "mat.".
    - **Training**: It is fed countless sentences and learns to predict the next word over and over. Over time, it gets better at recognizing grammar, tone, and meaning.
    - **Strengths**: Fast, great at long texts, understands context well.
    - **Weaknesses**: Needs a lot of memory and powerful hardware.
  + **2. RNN (Recurrent Neural Network)**
    - **How It Learns:** RNNs read text one word at a time from left to right, like a human reading a book line by line. It remembers past words using a kind of "memory" to understand what comes next.

**Training:** For every word, it updates its memory to learn what usually follows. But the memory fades quickly, so it struggles with longer sentences or remembering earlier context.

**Strengths:** Good for short, simple sequences like time series or short phrases.

**Weaknesses:** Can forget earlier parts of long text, and it’s slow to train since it reads sequentially.

* + **GAN (Generative Adversarial Network)**
    - **How It Learns:** GANs have two models playing a game. One (the Generator) tries to create realistic text (or images), and the other (the Discriminator) tries to spot if it’s fake. They improve by trying to outsmart each other.
    - **Training:** Over time, the generator gets better at “fooling” the discriminator, and the result is high-quality output.
    - **Strengths:** Good at generating content that looks realistic.
    - **Weaknesses:** Training can be unstable, and it’s not designed for understanding or generating complex language.
  + **Diffusion Models**
    - **How It Learns:** Think of this like “un-blurring” an image. These models learn to **reverse noise** — they start with a messy sentence or image and learn how to clean it up step by step.
    - **Training:** They are trained by adding noise to clean data, then learning how to undo it. Eventually, they can generate brand-new content from scratch.
    - **Strengths:** Great for creativity and generating high-quality content.
    - **Weaknesses:** Slower and computationally expensive.

**Bottom Line:** For LLMs, Transformers are the gold standard because they can understand context, handle long texts, and process information in parallel — which makes them ideal for real-world language understanding and generation.

## **Where Is the Knowledge Stored?**

* The model does not store facts like a database.
* Instead, it learns patterns and statistical associations in its parameters during training.
* Once trained, these patterns allow the model to make predictions based on context.

## **What About Real-Time Data?**

LLMs are trained up to a cutoff date (e.g., GPT-4's cutoff is 2023 unless tools like Browsing or APIs are enabled). They do not access live data unless connected to:

* External tools (via Agent SDK, LangChain, etc.)
* Plugins or APIs (e.g., weather, stock prices)

## **How Do LLMs Search or Respond to Input?**

It all comes down to tokens:

* A token is a small piece of text (e.g., "hello" = 1 token, "encyclopedia" = 3 tokens).
* When you ask a question, the model tokenizes it, processes it through neural layers, and predicts the most likely next token.
* It generates responses one token at a time, choosing the next word based on probability.

## **Tokenization = The Search Engine of LLMs**

* Input is broken into tokens.
* The model searches for its neural network patterns using attention mechanisms (especially in transformers) to find relevant associations.
* This lets it “respond” as if it understands—even though it’s pattern prediction.

# **What Is Generative AI?**

Generative AI is a category of AI that creates content—text, images, music, or code—based on patterns it has learned.

Real-World Examples:

* ChatGPT writing a blog post
* Midjourney generating digital art
* Copilot writing code in VS Code

But Here’s the Key: Generative AI responds to prompts. Agentic AI acts on goals.

# **Are You Confused Between Generative AI & Agentic AI? Let’s Compare**

**Key Differences:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Generative AI** | **Agentic AI** |
| Nature | Reactive (responds to input) | Proactive (acts toward a goal) |
| Task Scope | Single step (one response at a time) | Multi-step (plans and decides) |
| Memory | Typically, stateless | Has memory and context |
| Tools Integration | Limited or manual | Automatically uses tools (APIs, functions, etc.) |
| Example | Writing a poem | Planning and booking a trip |

# **Analogy to Understand**

Generative AI is like a pencil: You tell it what to write, and it writes.

Agentic AI is like a personal assistant: You tell it what you want, and it handles everything—including researching, deciding, and acting.

# **How Can We Build Agentic AI?**

Several frameworks exist to help developers create agentic behavior in AI systems:

|  |  |
| --- | --- |
| **Tool/Framework** | **Description** |
| **LangChain** | Python/JS framework to build LLM-powered agents |
| **CrewAI** | Team-based agent framework with roles/collaboration |
| **OpenAI Agent SDK** | SDK by OpenAI for building multi-tool intelligent agents |

# **Spotlight: What Is OpenAI’s Agents SDK?**

The OpenAI Agents SDK allows developers to build powerful AI agents that can:

* Accept goals instead of prompts
* Use multiple tools (e.g., APIs, file search, databases)
* Interact with sub-agents or other services
* Remember context and perform long tasks

Main Components:

* Main Agent: Central controller
* Tools: APIs, calculators, web browsers
* Memory: Persistent history or context
* Multi-Agent Collaboration: One agent can delegate tasks to others

# **Why Use OpenAI’s Agents SDK?**

* No Complex Coding: Simplified structure for agent orchestration
* Robust Multi-Tool Agents: Easily plug in various tools
* Readable Code: Clean and declarative syntax
* Self-Managing Agents: Agents can call others or delegate
* LLM-Agnostic: Works with various LLMs (e.g., GPT-4, Claude, Mistral)
* Open Source Compatibility: Easily extendable
* Memory Support: Track progress across sessions
* Goal-Driven Thinking: Not just text generation—real action

# **Conclusion**

We are entering a new era of AI—moving from reactive content creators to autonomous digital agents. With LLMs as the foundation, we built Generative AI. Now, with frameworks like OpenAI’s Agents SDK, we unlock Agentic AI—AI that reasons, plans, and acts.

This transition empowers developers to build intelligent, goal-driven applications without complex code. The OpenAI Agents SDK is a robust, future-ready tool that simplifies the creation of real AI assistants capable of making decisions and executing tasks—just like a human.