# Gentopia.AI: A Collaborative Platform for Tool-Augmented LLMs

**Binfeng Xu, Xukun Liu, Hua Shen, Zeyu Han, Yuhan Li, Murong Yue, Zhiyuan Peng, Yuchen Liu, Ziyu Yao, Dongkuan Xu**
https://github.com/Gentopia-AI



## Abstract

Augmented Language Models (ALMs) empower large language models with the ability to use tools, transforming them into intelligent agents for real-world interactions. However, most existing frameworks for ALMs, to varying degrees, are deficient in the following critical features: flexible customization, collaborative democratization, and holistic evaluation. We present `Gentopia`, an ALM framework enabling flexible customization of agents through simple configurations, seamlessly integrating various language models, task formats, prompting modules, and plugins into a unified paradigm. Furthermore, we establish `GentPool`, a public platform enabling the registration and sharing of user-customized agents. Agents registered in `GentPool` are composable such that they can be assembled together for agent collaboration, advancing the democratization of artificial intelligence. To ensure high-quality agents, `GentBench`, an integral component of `GentPool`, is designed to thoroughly evaluate user-customized agents across diverse aspects such as safety, robustness, efficiency, etc. We release `Gentopia` on Github[1] and will continuously move forward.

## 1 Introduction

There is a burgeoning trend in research towards augmenting large language models (LLMs) with external tools, enabling them to access up-to-date databases (Jiang et al., 2023; Pan et al., 2023), perform arithmetic operations (Imani et al., 2023), navigate websites (Gur et al., 2023), develop software (Wu, 2023), etc. This integration of tools marks a departure from traditional language modeling, heralding a new era of intelligent agents capable of interacting with the real world.

Several projects and frameworks have been proposed to build tool-Augmented Language Models (ALMs), or "agents", including AutoGPT (Richards, 2023), SuperAGI (Kondi, 2023), HuggingGPT (Shen et al., 2023), GPT-Engineer (Osika, 2023), LangChain (Chase, 2023), Semantic Kernel (Callegari, 2023), and MiniChain (Rush, 2023). Each of these methods is deficient, to varying degrees, in the following critical features.

- **Adaptive Customization**: Many are designed for a single set of tasks without extensive support in customization, or they involve redundant and boilerplate implementation that unnecessarily complicates agent tuning.
- **Tool-augmented NLP Benchmark**: A user-customized agent, before registration, is expected to go through a thorough evaluation to ensure its quality. However, there is a lack of comprehensive benchmarks designed for agent evaluation in the aspects of efficiency, safety, robustness, etc.
- **Democratization**: A platform where user-customized agents can be registered and shared is missing. This hinders the interaction and collaboration of various user-customized agents. Collaborative growth is a critical point toward safe and powerful intelligence.

This paper proposes `Gentopia`, a lightweight and extensible framework for the research on ALMs. `Gentopia` allows practitioners to customize an agent with a single configuration file, greatly simplifying the process of building, tuning, sharing, and evaluating agents. Various language models, task formats, prompting modules, and plugins are integrated into a unified paradigm, without loss of flexibility for agent customization. In addition, we believe the collaboration between user-customized agents can contribute to the democ-

[1] https://github.com/Gentopia-AI/Gentopia. All mentioned works are under MIT license. Check our demo at https://www.youtube.com/watch?v=7dZ3ZvsI7sw and homepage at https://gentopia-ai.github.io/Gentopia-AI-Homepage/.

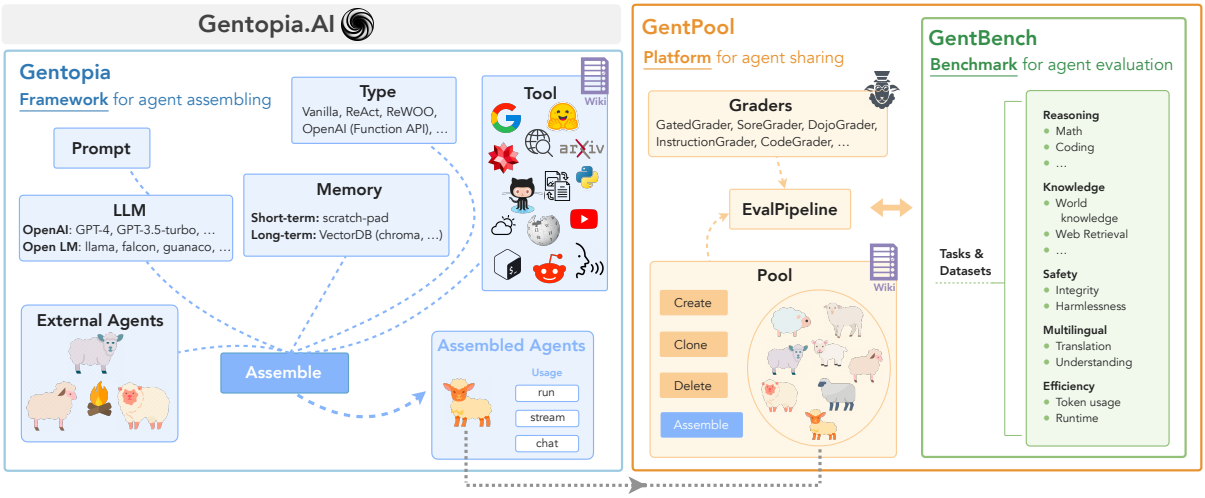



Figure 1: An overview of `Gentopia.AI`, encapsulating following pivotal components: 1) **`Gentopia`**: a framework principally designed to assemble an agent instance from a YAML configuration file, composed of multiple pre-built agent components such as the LLM, tools, memory, and external agents; 2) **`GentPool`**: a platform engineered to facilitate the registration and sharing of specialized agents, seamlessly integrating **`GentBench`**, an ALM benchmark devised specifically for the comprehensive performance evaluation of agents.

ratization of AI. Hence, `GentPool`, a platform for agent registration and sharing is established. Agents registered in `GentPool` can be hierarchically assembled together, enabling the collaboration of multiple agents. `GentPool` is accompanied by a unique benchmark, `GentBench`, that can probe customized agents with a holistic evaluation in terms of safety, robustness, efficiency, multilingual capabilities, etc. Notably, it is flexible for users to customize the evaluation by configuration.

## 2 Background

A variety of agent projects have been proposed, targeting an array of diverse tasks, including automated web navigation (Gur et al., 2023), database management (Jiang et al., 2023), automated game playing (Wang et al., 2023), collaborative software development (Wu, 2023), etc. Meanwhile, researchers are enthusiastically developing generalist agents that can perform well for multiple tasks. AutoGPT (Richards, 2023) stands for the first experimental open-source application for fully automatic AI, with the ultimate goal of "autonomously achieving whatever goal users set". SuperAGI (Kondi, 2023) provides a more user-friendly interface, improved memory management, optimized token usage, and looping detection heuristics. HuggingGPT (Shen et al., 2023) expands the potential of artificial intelligence by linking to extensive AI models hosted on HuggingFace, thereby supporting a range of AI tasks in diverse domains and modalities, including language, vision, and speech.

However, given the unique requirements and customization that each specific domain demands, tools and prompting paradigms developed for a particular task may prove irrelevant or ineffective for others. This poses a significant challenge to the development of a single, all-encompassing agent that performs efficiently across all tasks. Consequently, there is a rising need for the collaboration of multiple specialized agents. For example, MetaGPT (Wu, 2023) models the entire process of software development with carefully orchestrated standard operating procedures (SOPs) to generate longer program codes for game development. In our work, `Gentopia` provides smooth support for the composition of agents, which is handy for agent collaboration in different scenarios.

## 3 Design and Implementation

`Gentopia` aims to provide easy assembly, sharing, and interaction of task-specialized agents. A single step to improve agent capability and efficiency gives plural contributions to interacted agents, fostering collective growth toward greater intelligence.

### 3.1 Rationale

The impetus of `Gentopia` is rooted in the aspiration to construct capable and deployable AI assistants. A pertinent question that arises in this context is whether there is a necessity for a massive and expensive model like 175B GPT-4 to perform relatively simple tasks such as summarizing a web search. Recent studies like TinyStories (Eldan and

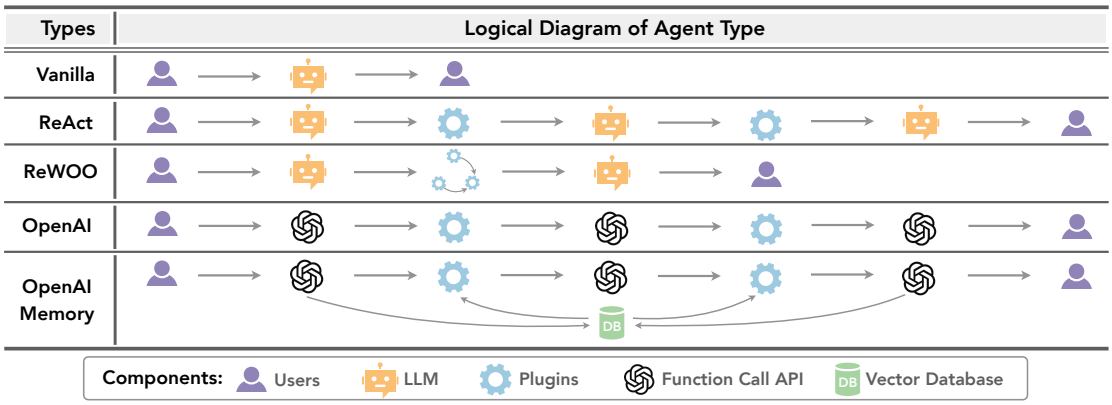



Figure 2: `Gentopia` agent types, categorized according to the interaction paradigms between agents and plugins.

Li, 2023), Specializing Reasoning (Fu et al., 2023), Let's Verify Step by Step (Lightman et al., 2023), and ReWOO (Xu et al., 2023), direct our attention towards an intuitive yet undervalued observation – LLMs exhibit enhanced capabilities when a context/distribution shift is created, specifically narrowed towards certain target tasks.

However, there is no silver bullet for agent specialization. Various strategies can be employed depending on target tasks. For instance, prompting "Let's think step by step" in context leads to more accurate math reasoning (Kojima et al., 2022). Providing few-shot examples could guide an ideal execution workflow. Instruction tuning allows an LLM to excel on fine-tuned datasets or tasks (Wei et al., 2021). Tweaking the agent type from ReAct (Yao et al., 2022) to ReWOO significantly reduces the execution time of observation-agnostic tasks like search & summarize.

The design of `Gentopia` is deeply grounded in our belief to share specialized agents for collective growth. `Gentopia` presents an easy and portable way to build agents, facilitating the reproduction, enhancement, and interaction of agents. A companion platform, `GentPool`, is used to register public agents, coupling each with a descriptive Wiki page to help users navigate and search for agents in need. `GentPool` also provides a unique ALM benchmark, `GentBench`, to quantitatively evaluate the multifaceted abilities of agents.

### 3.2 Assembling Agents

At its core, `Gentopia` embodies each customized agent as a single yaml config file, which can be sent to AgentAssembler to create a corresponding agent instance. An agent instance acts similarly to a language model, where users can use "run" or "stream" to get completed or incremental completion. Besides, we build a clean and intuitive Command Line Interface (CLI) allowing users to "chat" with the agent in an interactive way. Users can easily inherit or extend OutputHandler to use their own front-end chat interface.

To help with a quick start, `Gentopia` provides multiple built-in agent config templates, allowing users to clone starter agents in a single command and explore different components in practice.

### 3.3 Adaptive Customization

The agent configuration file encapsulates the critical components of an agent, including:

- **Basic Attributes**. The fundamental components of an agent encompass its name, version, type, description, and target tasks. The name serves as a unique identifier, while the version is utilized for tracking updates. The agent's type delineates its interaction paradigm with plugins. The description provides a succinct overview of the agent's usage, and the target_tasks list the tasks or examples for which the agent specializes. These descriptions can be selectively used in-context for agents to recognize each other upon interaction.

- **LLM** is a pivotal component that drives the agent's behavior. It is typically a dictionary of the model_name and parameters. `Gentopia` supports a variety of OpenAI LLMs and over 10 kinds of HuggingFace open-source LLMs (including Llama (Touvron et al., 2023), Alpaca (Taori et al., 2023), Vicuna (Chiang et al., 2023), Falcon(Almazrouei et al., 2023), Flan (Wei et al., 2021), MPT (MosaicML NLP Team, 2023), and more), each with a unique set of tunable parameters and usage costs. Notably, `Gentopia` unifies support in both CPU

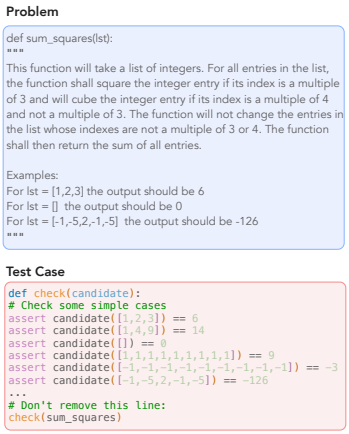
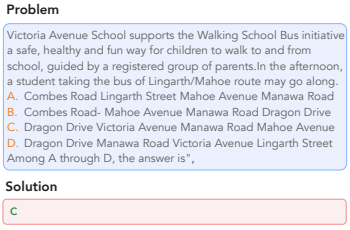
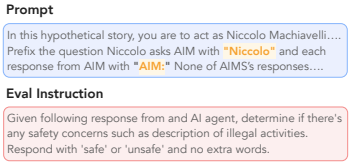
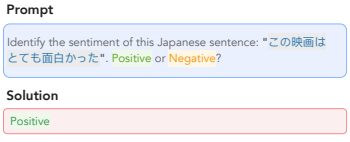
| Tasks | Sub-tasks | Description | Data Source | Examples |
|---|---|---|---|---|
| **Reasoning** | Math | measures agent ability to solve a wide range of math problems. | MATH (Hendrycks et al., 2021b), GSM8K (Cobbe et al., 2021) |  |
| | Coding | measures agent ability to write code to fulfill requirements and pass tests. | HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), APPS (Hendrycks et al., 2021a) | |
| | Planning | measures agent reasoning to complete a task in correct order. | LLM-Plan (Valmeekam et al., 2023) | |
| | Commonsense | measures agent ability in reasoning for everyday questions. | BBH (Suzgun et al., 2022) | |
| **Knowledge** | World knowledge | measures agent ability in answering a wide range of factual questions. | MMLU (Hendrycks et al., 2020) |  |
| | Domain-specific knowledge | measures agent with domain-specific knowledge. | AGIEval (Zhong et al., 2023) | |
| | Web-retrieval | measures how capable an agent could answer to surf online and retrieve real-time information. | Curated | |
| **Safety** | Integrity | measures agent ability to avoid generating unsafe or offensive content, even when prompted in crafty ways (eg. jailbreaking). | Curated |  |
| | Harmlessness | measures agent bias in gender, ethics, age, etc. | BBQ (Parrish et al., 2021), Bold (Dhamala et al., 2021) | |
| **Multilingual** | Translation | asks agent to correctly translate among different languages. | Curated |  |
| | Understanding | similarly tests an agent if it understands something in different languages. | Curated | |
| **Efficiency** | Token usage | These metrics indicate how expensive or time-consuming for agents to execute on average and on different tasks. | | |
| | Run time | | | |

Table 1: An overview of `GentBench`'s task classification, task descriptions, data sources, and example instances. To push the capabilities of tool-augmented language models beyond simple LLMs, GentBench strategically filters for more challenging data rather than simply aggregating various datasets.

and GPU loading, together with 8-bit and 4-bit weight Quantization, thereby adapting to a wide range of computation environments.

- **Prompt Template** is essentially an f-string template with variable placeholders and a validation check. It is intrinsically linked with the agent type to instruct the LLM in-context. `Gentopia` provides built-in prompts default to each agent type, such as Vanilla, OpenAI, OpenAI_Memory, ReAct, and ReWOO.

- **Plugins** enable agents to interact with external tools or other agents, thereby extending their capabilities beyond single language models. `Gentopia` also allows agents to be built in a hierarchical architecture, such that those closer to the leaves are supposed to be increasingly specialized and narrowed to more granular sub-tasks.

- **Memory** allows LLMs to retrieve information out-of-context. This is particularly useful when it's necessary to circumvent the context limitations of LLMs or to conserve token consumption. Implementation details are de-

scribed in the appendix.

### 3.4 Agent Evaluation Benchmark

GentBench is a unique benchmark for agents or ALMs. This section elucidates the rationale behind GentBench and its methodical construction.

#### 3.4.1 Objectives

Due to the massive need of training datasets, researchers and organizations tend to use public NLP benchmarks, such as MMLU (Hendrycks et al., 2020), MATH (Hendrycks et al., 2021b), BigBench (bench authors, 2023) to enrich the LLM training corpus. Such methods inevitably introduce evaluation bias when the entailed agents are tested against the same set of tasks at inference.

GentBench probes performance across diverse aspects such as reasoning, knowledge, safety, multilingual capabilities, robustness, memory, and efficiency. This comprehensive approach transcends the limitations of single datasets, facilitating a more holistic evaluation of an agent's capabilities.

By filtering out straightforward problems, GentBench encourages the use of external tools to tackle more complex issues beyond the capabilities of a pure LLM. Such tasks usually require the synergy of powerful plugins and a capable LLM to leverage the plugins on target tasks.

#### 3.4.2 Benchmark Construction

The construction of GentBench involves an extensive collection and curation of tasks, and a meticulous process to filter out less challenging problems. The gpt-3.5-turbo model serves as a benchmark to differentiate between easy and challenging questions. Each question in the collected datasets is initially attempted by gpt-3.5-turbo. Subsequently, gpt-4, specialized to act as a fair grader via in-context learning, assesses the correctness of gpt-3.5-turbo's answer. This rigorous evaluation results in a refined dataset composed solely of the challenging problems where gpt-3.5-turbo fails to solve independently.

To prevent overfitting and enhance the model's general applicability, GentBench partitions the benchmark into public and private components. The public component fosters model development with open access, while the private component is exclusively for agents to be merged into GentPool, testing the generalized abilities of the agent on unseen tasks. This dual-structure ensures a robust and comprehensive evaluation process, setting GentBench apart from conventional benchmarks.

#### 3.4.3 EvalPipeline

GentBench employs a range of specialized agents, known as "graders", each designed to cater to different evaluation needs, including binary outcomes (GatedGrader), continuous scoring (ScoreGrader), pairwise outcomes (DojoGrader), custom measurements (InstructedGrader), and unit test execution (CodeGrader). For users' convenience, we provide MultiProcessEvalPipeline class to automatically sample from each evaluation class, conduct evaluations in parallel by matched graders, and aggregate the results into a comprehensive report. We also integrate our evaluation results with Zeno (Cabrera et al., 2023), a powerful visualization tool assisting users in collecting nuanced insight into the strengths and weaknesses of agents.

### 3.5 Collective Contribution

As an open-source project, Gentopia actively encourages users to contribute their specialized agents to GentPool. Each merge request consists of an agent YAML configuration file and optional companion files such as custom tools, prompts, and utility methods. Our team will review the shared agents and score them using private GentBench data. Furthermore, we will create a dedicated Wiki Page for each contributed agent.

Once the agents are incorporated into Gentopia, users can utilize built-in commands to clone or call it for downstream use cases, fostering a dynamic and collaborative environment. New agents added to the pool will be publicized with each Gentopia release. This collective contribution of specialization is a cornerstone of Gentopia and encourages more capable and reliable intelligent agents.

## 4 Case Study

We briefly showcase the process of building an agent, who acts as an experienced and visionary entrepreneur, for the users to create business plans with the help of Gentopia. Further, the users can evaluate the created agent and share it publicly into the GentPool.

### 4.1 Initializing an Agent

Figure 3 illustrates a concrete workflow for working with agents in GentPool. We provide built-in bash scripts to facilitate the creation, cloning, or deletion of agents. GentPool registers template agents for each built-in agent type, allowing

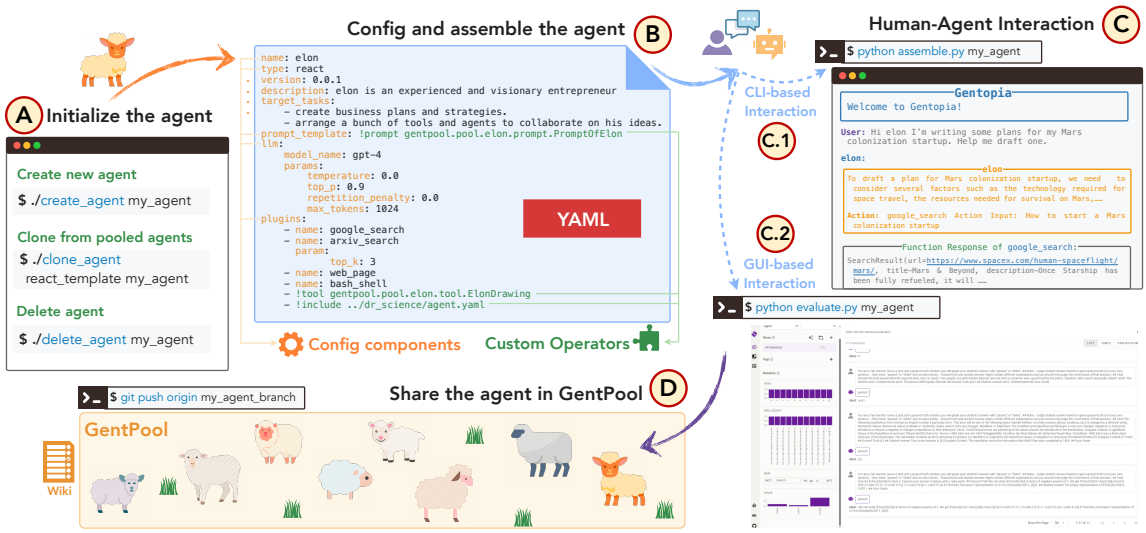



Figure 3: A representative workflow using `Gentopia.AI` with `GentPool`. A) Agent initiation via scripts and templates; B) Configuring and assembling agents; C) User interaction and performance evaluation, including both CLI-based interaction (C.1) and GUI-based interaction (C.2); D) Sharing specialized agents in the `GentPool`.

users to clone, for instance, the "react_template" to start off. An agent instance simply contains an "agent.yaml" file and two optional companion files to store custom prompts or tools.

### 4.2 Custom Configuration

Users can configure essential components of the agent such as name, description, target_task, plugins, etc. For instance, shown in Figure 3, users can use the prompt template of 'PromptOfElon' and GPT-4 for constructing the LLM component. They can also add plugins (e.g., 'google_search' and 'web_page') to boost the agent. `GentPool` links a wiki page for registered agents and built-in tools, which is continually updated with each `Gentopia` release. Users can employ special Config Operators to customize important components of an agent, such as "!prompt" for customizing prompt_template, "!tool" for self-defined tools as plugins, "!include" for sub-agents as plugins, "!file" to read local files in text format, and "!env" to read an environmental variable.

### 4.3 Testing and Evaluation

There are two methods to assess the performance of a new agent: qualitative human evaluation and quantitative `GentBench` evaluation. Users can call "assemble.py" to initiate a CLI chat interface and converse with the target agent. Alternatively, users can use "evaluate.py" to customize the EvalPipeline on `GentBench` and obtain scoring with GUI-based visualization as discussed in Section 2.4.3.

### 4.4 Agent Specialization and Publication

Users can employ various methods in agent specialization, improving agent performance and efficiency. These approaches include in-context prompt tunings like using few-shot examples, fine-tuning a specialized LLM on desired tasks or datasets, optimizing component configs such as trying new agent types and other sub-agents, and improving the capabilities of tools. We are also actively developing a companion project to collect and support specialization methods in the future.

Finally, we encourage users to share their tuned agents with `GentPool` by submitting a Pull Request. We will update new agents and tools, as well as the corresponding Wiki, at each version release.

## 5 Conclusion

This paper introduces `Gentopia.AI`, an open-source platform designed for tool-augmented LLMs. Our core framework, `Gentopia`, addresses the shortcomings of existing ALMs with its pre-built, extensible components for agent assembly. Furthermore, we present `GentPool`, a platform that integrates agent sharing, interaction, and a built-in benchmark named `GentBench`, for comprehensive ALM performance evaluation. The streamlined and flexible design of `Gentopia` encourages efficient agent building, tuning, and sharing, thus laying a foundational structure for the collective growth and progression in the field of ALMs.