

A Mixture-of-Agents System for Fact-Based Comprehensive Search

Muhammad Abu Bakar, Vijay Madiseti

Georgia Institute of Technology, Atlanta, USA
Email: mbakar3@gatech.edu, vm10@gatech.edu

How to cite this paper: Abu Bakar, M. and Madiseti, V. (2025) A Mixture-of-Agents System for Fact-Based Comprehensive Search. *Journal of Software Engineering and Applications*, 18, 507-541.
<https://doi.org/10.4236/jsea.2025.1812030>

Received: September 15, 2025

Accepted: December 19, 2025

Published: December 22, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Large Language Models (LLMs) exhibit remarkable capabilities; however, they possess inherent limitations due to static training, which leads to outdated information and hallucinations. Furthermore, most existing multi-agent frameworks depend on predefined and rigid agent roles that lack adaptability across a range of tasks. We introduce AMASS (Autonomous Multi-Agent System for Fact-Based Comprehensive Search), a dynamic and self-organizing framework that addresses these limitations through two primary innovations: autonomous agent generation and a Mixture of Agents (MoA) strategy. In contrast to traditional systems, AMASS spontaneously generates specialized agents tailored to the unique requirements of each task. Each agent autonomously selects the most appropriate LLM, based on the complexity of the task, thereby utilizing lightweight models for straightforward queries and more powerful models for intricate reasoning tasks. A centralized Critic Agent supervises the system to eliminate redundancy, ensure coherent collaboration, and continuously optimize the alignment of agents to tasks. Our cascaded search mechanism further enhances the accuracy of results by enabling an iterative and context-sensitive refinement process. Evaluated across the GPQA, Bamboogle, MuSiQue, and TriviaQA datasets, AMASS significantly outperforms state-of-the-art frameworks in accuracy, efficiency, and factual reliability, thus performing better for autonomous multi-agent reasoning systems.

Keywords

Multi-Agent Systems, Cascaded Search, Mixture Of Agents (MOA), Autogen

1. Introduction

Large Language Models (LLMs) have revolutionized natural language processing and decision-making tasks. However, they suffer from fundamental limitations such as knowledge obsolescence, hallucinations, and inefficiencies in adapting to

diverse tasks. The static nature of LLM training results in outdated responses, making them unreliable for real-time applications. Additionally, existing multi-agent frameworks attempt to mitigate these challenges but largely rely on predefined agent architectures with fixed roles. This restricts their ability to adapt dynamically to new tasks and efficiently allocate computational resources.

To address these shortcomings, we introduce AMASS (Autonomous Multi-Agent System for Fact-Based Comprehensive Search), a novel framework that leverages *autonomous agent generation and optimization*. Unlike conventional agentic architectures, AMASS dynamically *creates, evaluates, and refines* specialized agents at runtime, ensuring optimal task execution. A key innovation of AMASS is its ability to *incorporate a mixture of experts dynamically*, selecting the most suitable LLM for each agent based on task complexity. Simple tasks are handled by lightweight models, while complex reasoning tasks utilize more powerful LLMs, optimizing both accuracy and computational efficiency.

Furthermore, AMASS advances multi-agent reasoning beyond conventional architectures by incorporating a Critic Agent, which removes superfluous agents, refines task strategies, and optimizes agent-task mapping. The framework also integrates cascaded search mechanisms, allowing agents to iteratively refine queries, delve deeper into information sources, and extract high-quality knowledge. Such enhancements establish AMASS as a cutting-edge autonomous multi-agent system, proficient in effectively balancing computational cost, adaptability, and accuracy.

The evaluation of AMASS is conducted utilizing the GPQA dataset, encompassing several domains such as Physics, Chemistry, and Biology. Empirical findings indicate that AMASS exhibits superior performance relative to existing agent-based frameworks, particularly with respect to response accuracy, search depth, and operational efficiency. Through the incorporation of autonomous agent formation alongside a heterogeneous ensemble of experts, AMASS introduces a novel paradigm within multi-agent architecture, enhancing its scalability, cost-effectiveness, and robustness in addressing intricate computational problems.

The remainder of this paper is structured as follows: **Section II** discusses related work, highlighting the limitations of existing multi-agent systems. **Section III** presents the AMASS architecture, including autonomous agent generation, expert selection, and execution mechanisms. **Section IV** details our evaluation results and comparative analysis, while **Section V** discusses future research directions.

2. Related Work

2.1. Static Agent Frameworks

Numerous methodologies have been devised to incorporate multi-agent systems for addressing complex challenges, which are challenging for a singular LLM to resolve. Meta-GPT [1] endeavors to address such intricate challenges by emulating the software development lifecycle. The various departments within a software firm have been transformed into distinct agent roles, each specializing in specific tasks. However, a limitation of the Meta-GPT approach lies in the fact that these

agents are defined by human intervention, relying on fixed system prompts and descriptive instructions, which hinders Meta-GPT's adaptability to novel use cases. Analogous constraints associated with predefined structures are evident in [2], CAMEL (Camel: Communicative Agents for "Mind" Exploration of Large Language Model Society) [3], AGENTVERSE: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors [4], and [5]. CAMEL was among the initial endeavors to implement agent role-play for resolving intricate tasks. It demonstrated superior performance in numerous test environments compared to the unrefined LLM model. However, the absence of a feedback loop resulted in discussions deviating from the primary subject matter.

2.2. Autogen

Autogen [6], a multi-agent framework developed by Microsoft, was specifically designed for deployment in multi-agent environments. This framework incorporated various multi-agent principles, including the summarization of previous communications, storage of agents within a database, and simulation of human-to-human interactions in real-world contexts. Despite Autogen's success in introducing autonomous agents, several limitations were observed. Notably, after the agents' initial development during the planning phase, no subsequent refinement was conducted. As a result, agents often became redundant, and superfluous agents were created that did not address the necessary tasks, causing discussions to deviate into irrelevant topics.

2.3. Autoagents

The "Autoagents" [6] paper addressed some of the limitations inherent in the "autogen" paper regarding agent creation by integrating a critic agent within the planning phase. This integration facilitated the inclusion of only the necessary agents during the planning process, thereby reducing the overhead associated with excessive inter-agent communication. However, a limitation of this framework is that the backend Language Learning Model (LLM) used for tasks remains static and predefined from the outset. In some respects, this architecture could be emulated through the use of a reasoning LLM in conjunction with a critic agent or evaluator within a loop. Multi-agent collaboration is most effective when each agent is highly specialized within its domain, as observed in real-world scenarios. Although the evaluator participates in the planning stage, its role is not sufficiently effective in reducing the number of agents or optimizing system prompts such that fewer agents could manage tasks typically requiring many. Furthermore, once an agent is created, this framework cannot alter the agent's characteristics.

2.4. EvoAgent: Towards Automatic Multi-Agent Generation via Evolutionary Algorithms

EvoAgents [7] utilized evolutionary methodologies for agent creation. The framework of evolutionary agents effectively addressed issues of redundancy among agents in the planning phase by introducing these agents. Once developed, the agents

undergo an evolutionary process whereby redundant agents are eliminated, and a new collection of agents is generated with refined prompts. The research primarily concentrated on the planning phase, notably lacking an evaluator during the execution stages. This absence is problematic, as even the most optimized set of agents cannot guarantee error-free outcomes.

2.5. MindSearch: Mimicking Human Minds Elicits Deep AI Searcher

MindSearch [8] emulates human reasoning by employing a network of nodes operated by Large Language Model (LLM) agents. Upon receiving an inquiry from a user, it conducts a search and ceases further action if the result is obtained. Should the answer remain elusive, the system diverges into multiple nodes, each probing in distinct directions. In comparative evaluations, it demonstrates significant performance advantages over prominent search frameworks. However, it is limited by its inability to integrate multiple agents and lacks the capacity to modify the LLM engine's backend to suit varying search requirements. Furthermore, it does not exhibit adaptability to programming-related tasks.

2.6. Agentic Reasoning: Utilizing Reasoning LLMs with Supplementary Tools for In-Depth Research

Agentic Reasoning [9] utilizes mind mapping and agentic reasoning models applied to the GPQA [10] dataset. Their findings suggest that employing basic tools, such as web browsers and Python interpreters, alongside mind maps fortified with agentic reasoning, can produce superior outcomes compared to those of human experts. The research asserts that the resolution of complex problems does not necessitate reliance on intricate agents; rather, the synergy of simple tools with reasoning models can effectively manage most tasks. Nonetheless, this methodology presents certain limitations, principally its dependence on a singular model throughout the process. By integrating an array of models, a diverse set of experts can be engaged for specific tasks. Furthermore, utilizing a variety of foundational models helps to diminish bias and reduce instances of hallucinations in the model's outputs, as these models are developed using significantly varied datasets and hyperparameters.

2.7. Feature Comparison

Below is the comparison of capabilities for AMASS vs the other multi agent systems. AMASS has more diverse set of capabilities (Table 1).

Table 1. Comparison of previous work.

System	Dynamic Agents	Web Search Verification	Cascaded Searches	Mixture Of Agents (MOA)	Remove Redundant Agents
Auto-Agents	√	√	×	×	√
Meta-GPT	×	×	×	×	×

Continued

Mind-Search	×	×	√	×	×
Autogen	√	√	×	×	×
Auto-GPT	√	×	×	×	×
Camel	×	×	×	×	×
EVO-AGENT	√	√	√	×	√
Agent-Verse	×	×	√	×	×
Internet of Agents	√	×	×	×	×
Agentic Reasoning	×	√	√	×	×
AMASS	√	√	√	√	√

3. Implementation

3.1. Overall Structure

The architecture comprises four principal components **Figure 1**: the Preprocessor, the Planner, the Executor, and the Summarizer. The Preprocessor is tasked with processing images and videos while integrating the necessary context into the user's query. The Planner is dedicated to devising a final set of optimal agents along with a comprehensive plan tailored to accomplish the user's objective. The Executor employs these agents and the formulated plans to produce the desired outcomes, including executing code, conducting internet searches, and facilitating discussions among various agents. Lastly, the Summarizer interprets the entirety of the dialogue and, in response to the user's question, delivers a precise answer.

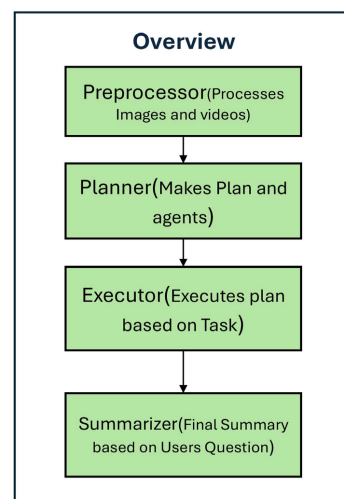


Figure 1. High-level overview showing all the components of the AMASS architecture.

3.2. Planner

The planner **Figure 2** functions as the central coordinating entity within the system. Upon receiving the user's inquiry and the associated contextual data (including text, images, and video), the planner initially generates a set of agents with BrainStorming. These agents are subsequently introduced to the Planner Agent

(Pa). The planning agent devises a comprehensive plan detailing each agent's responsibilities and methodologies for executing those tasks. The plan, along with the agents, is then submitted to the critic agent for evaluation. The critic agent assesses the agents for redundancy and potential minimization. Autogen [11] currently lacks a mechanism for eliminating these redundant agents, resulting in the creation of superfluous agents where tasks may be accomplished with fewer agents. Furthermore, the critic agent evaluates the suitability of models for each specific agent. For instance, in accordance with the user's optimization strategy, which may favor smaller models such as GPT-4o-mini for simple tasks and GPT-4o for more complex tasks, the critic agent offers feedback to the planner agent. This feedback enables the planner agent to modify the backend of the agents as necessary. This approach fosters the integration of smaller and more specialized models tailored to distinct tasks, thereby enhancing decision accuracy and potentially lowering costs while increasing the efficiency of each agent. This novel strategy of utilizing different agents is not addressed in the existing literature referenced previously and constitutes an original contribution of our research on autonomous agents. Following iterative exchanges between the Planner Agent and the critic agent, a final plan comprising the optimal set of agents is developed and advanced to the execution stage. The complete algorithm is described as 1.

3.2.1. Agents Generation with Brainstorming Agent

When the user submits a question, the Initial Brainstorming agent generates many different sets of specialized agents with domain-adaptive prompts. Initially, this creates numerous redundant agents. The Brainstorming agent produces as many potentially relevant specialized agents as possible, similar to how humans brainstorm before starting a plan by considering all the things they might require. However, as the number of agents increases, both the cost and the context length grow exponentially. A larger number of agents also introduces noise into the discussion with irrelevant details. For example, if we want to design an agent for planning a trip from London to New York, an agent specialized in London might not be relevant, since it could generate unnecessary suggestions related to staying in London.

3.2.2. From Brainstorming to Final Agent Composition

To address redundant agents, we introduce a Planner agent that leverages these agents to create a coherent plan. Alongside it, a Critic agent evaluates the quality of the Planner's plan and determines whether some agents can be removed without reducing effectiveness. The Critic agent also provides suggestions. Through a back-and-forth process between the Planner and the Critic, a final plan is developed. This iterative planning continues until the Critic agent at the planning stage is satisfied. The final set of agents may differ significantly from the initial set, with agents sometimes inheriting features from a combination of others. By "features", we mean the system prompt, which can be adjusted to incorporate multiple specialties.

3.2.3. Critic Agents Criteria for Pruning

The Critic Agent employs a multi-criteria evaluation framework to determine which agents should be pruned from the dynamically generated multi-agent system. Through a structured group chat interaction with a Planning Agent (limited to 3 rounds using round-robin speaker selection), the Critic Agent evaluates each proposed agent against several key criteria: (1) **necessity and redundancy**, assessing whether tasks could be accomplished with fewer agents and identifying agents that serve primarily aesthetic or descriptive purposes without concrete functional requirements; (2) **role clarity**, ensuring each agent has a well-defined system message and description with clear justification for its inclusion and specific task responsibilities; (3) **factual verification requirements**, determining whether tasks involving precise facts necessitate the inclusion of a specialized search agent for information verification; and (4) **cost-effectiveness**, optimizing model selection based on task complexity (e.g., assigning computationally expensive models like openai/o3-mini or google/gemini-2.0-flash-lite-001 only for complex reasoning tasks in physics, chemistry, or mathematics, while using lighter models like gpt-4o-mini for text-based factual tasks). The pruning decisions emerge from this collaborative deliberation, culminating in a refined agent configuration encoded in JSON format that eliminates superfluous agents while preserving those with demonstrable functional contributions to the task-solving pipeline.

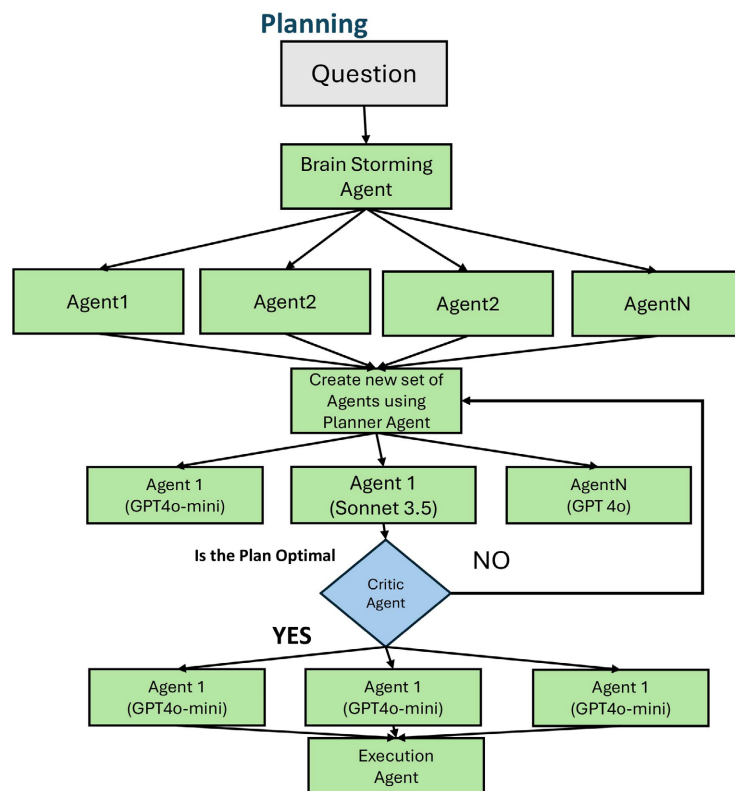


Figure 2. Showing Planning Stage. Planning stage consisting of three main agents: the Brainstorming Agent, the Planner Agent, and the Critic Agent.

3.3. Execution Stage

The execution phase, **Figure 3**, integrates all planning elements and agents transitioned from the planning stage, delivering them to the Manager Agent. The Manager Agent possesses comprehensive authority over which agent is invoked and the order of their invocation. Agents are called in accordance with a predetermined plan, and each agent executes an action based on its designated role.

Algorithm 1 AMASS Agent Planning Process

Require: *building_task*
Ensure: *optimized_agent_configs*

- 1: **Phase 1: Brainstorming**
- 2: Generate agent names based on task requirements
- 3: Create system messages for each agent
- 4: Generate agent descriptions
- 5: Build initial *agent_configs* list
- 6:
- 7: **Phase 2: Refinement (max 3 rounds, configurable)**
- 8: $round \leftarrow 0$
- 9: **while** $round < 3$ **and** not *terminated* **do**
- 10: $round \leftarrow round + 1$
- 11:
- 12: **a. Critic Agent:**
- 13: Review agent list
- 14: Identify unnecessary agents
- 15: Suggest improvements
- 16:
- 17: **b. Planning Agent:**
- 18: Process critic feedback
- 19: Modify agent configurations
- 20: Remove redundant agents
- 21:
- 22: **c. Check termination:**
- 23: **if** *planning_agent* outputs final JSON **then**
- 24: $terminated \leftarrow true$
- 25: **break**
- 26: **end if**
- 27: **end while**
- 28:
- 29: **Phase 3: Finalization**
- 30: Get final agent configuration from last message
- 31: Parse JSON output
- 32: **return** *optimized_agent_list*

The Manager Agent also processes requests from agents; should agents request the search_engine agent or the executor_agent, the Manager Agent will summon them and instruct them to perform actions as per the agents' requests. Additionally, the Manager Agent can invoke the critic agent, which is tasked with assessing the execution relative to the user's specified task. In cases where discrepancies or contradictions with the search agent arise, the critic agent can provide suitable feedback. If it determines that the search outcome lacks relevance, it can prompt

the Manager Agent to initiate a revised search, potentially altering the query for the search agent. For instance, when seeking the top 10 gynecologists in New York along with their contact details, the search_agent, informed by feedback from the executor agent, can engage in a cascaded search operation as depicted in the accompanying flow diagram. The search query can be dynamically adjusted based on previously obtained results.

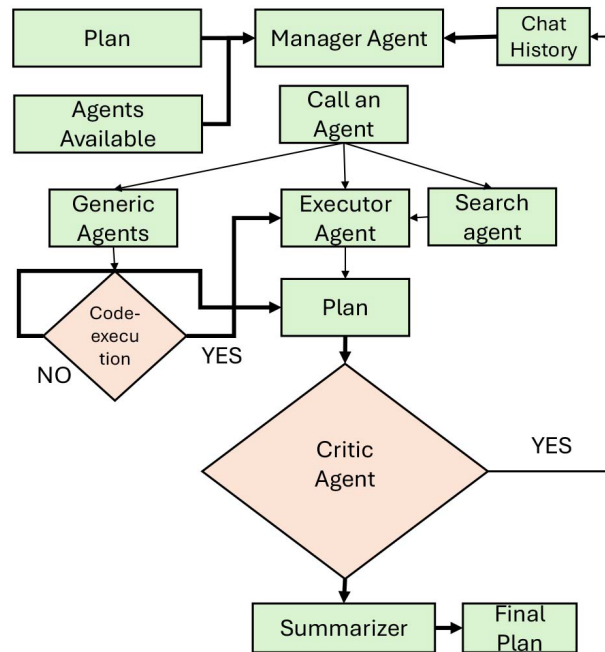


Figure 3. Showing Execution Stage, which is responsible for agents' interaction to achieve the user's goal.

The executor stage is responsible for executing code, thereby allowing real-time testing to verify the absence of bugs and confirming that the code performs as intended by the user. This execution capability permits the agent to employ a variety of tools at runtime to refine the user's inquiry. Moreover, the feedback mechanism, in conjunction with the manager, ensures that the discourse remains focused and that agents do not divert into secondary discussions. The comprehensive flowchart of the execution stage is presented below.

3.4. Search Agent

The search agent is composed of a basic search engine and a web scraper. The search engine identifies and supplies the most pertinent links, while the scraper concurrently extracts the content from these links. The extracted data is subsequently processed by a compact language model for summarization. The outcome of this summarization comprises the final output of the search agent. Consequently, by integrating a search engine with a web scraper, the system acquires the capability to retrieve information from the web. Search agent searches based on query and the review agent reviews whether all the information that user had re-

quired has been obtained through searching. If there is some more refinement needed it can suggest an alternative query for the search agent to try out. The complete cascaded algorithm is described in 2.

Algorithm 2 Cascaded Search with Iterative Query Refinement

Require: *init_query, max_depth, max_results*

Ensure: *search_results*

```

1: query ← init_query
2: depth ← 0
3: accum_results ← []
4: while depth < max_depth do
5:
6:   a. Execute Search:
7:   raw_results ← GoogleSearch(query, max_results)
8:
9:   b. Process Results:
10:  for result in raw_results do
11:    content ← ScrapeURL(result.url)
12:    summary ← Summarize(content, query)
13:    accum_results.append(summary)
14:  end for
15:
16:  c. Check Stopping Criteria(Review agent):
17:  if SufficientInformation(accum_results, init_query)
18:    then
19:    break
20:  end if
21:
22:  d. Query Refinement(review agent):
23:  missing_info ← IdentifyGap(accum_results, init_query)
24:  query ← RefineQuery(initial_query, missing_info)
25:
26:  depth ← depth + 1
27: end while
28: return accumulated_results
29:
30: Stopping Criteria:
31: 1. SufficientInformation(): All required facts found
32: 2. depth ≥ max_depth: Maximum iterations reached
33: 3. No new information in results
34:
35: Example Execution:
36: Iteration 0: query ← "Top 5 doctors in London"
37: Results: Doctor names found, but no contact numbers
38: Gap: Missing contact information
39: Iteration 1: query ← "Dr. [Name] contact number London"
40: Results: Contact numbers retrieved
41: Sufficient information → TERMINATE

```

4. Evaluation

We used various models from the Open Router platform via their APIs. For AMASS, our base models for the Brainstorming, Planning, and Critic agents were GPT-4o. For specialized agents, the planner can choose from a range of models, including Gemini-2.0, O3-mini, GPT-4o-mini, and GPT-4o. The selection criteria for these agents are adjustable to suit different industry requirements. For exam-

ple, in cases where cost-saving and better accuracy are prioritized, smaller models may be preferred for simpler tasks. In more complex domains, such as medical analysis, specialized agents may be used, such as one focused on radiology and another on cancer treatment. The selection criteria are flexible to accommodate new models with advanced capabilities as they become available. The planner's role is to specify the selection strategy in the prompts, and AMASS handles the rest of the work. The number of reasoning turns in the Planning stage is limited to 3, as we observed that most of the time, the planner and critic reach a conclusion within three turns. For the Execution stage, the reasoning turns are capped at 12, although this parameter can be adjusted depending on task complexity. If the agents reach a conclusion before their maximum turns and the critic is satisfied with the results, they output the word TERMINATE (enclosed in three back-ticks), signaling the AMASS framework to stop the reasoning loop. This applies to both the Planning and Execution stages. We used default hyperparameters and seeds from the providers, as we wanted to maintain consistency in results across users of these LLMs. This ensured that the outcomes were largely similar, with a standard deviation of accuracy within 2% - 3%.

4.1. Assessment of Advanced-Level Queries in the GPQA Dataset

The GPQA (Graduate-Level Google-Proof Q&A) dataset comprises 448 rigorously challenging multiple-choice questions devised by subject matter experts in the disciplines of biology, physics, and chemistry. Constructed to present exceptional difficulty, even individuals who are experts or are in the process of obtaining their PhDs in these specific fields achieve a mere 65% accuracy on these questions (this figure rises to 74% when obvious errors that were later recognized are discounted). Remarkably, individuals who are highly skilled but not experts, even with unrestricted internet access and dedicating over 30 minutes per question, achieve only 34% accuracy, highlighting the "Google-proof" characteristic of the dataset.

Our autonomous approach employing a mixture of agents demonstrates superior accuracy compared to the leading models currently available in the market as shown in tab:gpqa-main. The framework we propose exhibits improved results over all existing solutions. The efficacy of our solution significantly exceeds that of Agentic Reasoning, which has recently gained attention due to its innovative method of integrating reasoning models with search and coding agents. By utilizing a mixture of agents, our method achieves a notable enhancement in performance across all subject areas. Specifically, in the field of chemistry, our approach provides an approximate 40% improvement in performance over the Agentic Reasoning approach. Moreover, our method substantially reduces the occurrence of hallucinations. The integration of the MOA approach and the critic agent results in hallucinations being virtually non-existent. The principal reason this method surpasses others is attributed to the synergy of the critic agent and the mixture of agents approach. In our experiments, the Gemini 2.0 Flash performed exception-

ally well on biology and chemistry questions, while O3 Mini exhibited superior performance in physics. The combination of these two diverse base models along with the critic agent constitutes an optimal strategy to address the limitations inherent in individual Large Language Model (LLM) base agents (**Table 2**).

Table 2. Performance comparison on GPQA dataset across Physics, Chemistry, and Biology.

Method	Phy.	Chem.	Bio.
Direct Reasoning			
Qwen2.5-32B	57.0	33.3	52.6
Qwen2.5-Coder-32B	57.0	33.3	52.6
QwQ-32B	75.6	29.8	66.2
Qwen2.5-72B	54.7	31.7	51.8
Llama3-30B	59.1	40.1	55.4
GPT-4o	59.5	40.2	61.9
o1-preview	59.9	59.9	65.9
o3-mini	79.5	65.9	81.6
gemini-2.0	76.7	75.8	79.1
Retrieve/Search in Reasoning			
RAG-Qwen2.5-32B	76.7	37.8	58.6
RAgent-Qwen2.5-32B	77.0	38.4	59.2
RAgent-QwQ-32B	77.7	39.9	61.3
Search-o1	77.9	47.3	62.8
O3-mini	84.9	75.3	82.6
gemini-2.0	76.7	78.2	81.9
GPT-4o	65.1	51.1	71.9
Agentic Reasoning	88.1	58.3	79.6
AMASS			
Ours	90.9	73.2	82.3

4.2. Hard QA Benchmarks for Multi-Step and Logical Reasoning Tasks

To assess the logical reasoning and multi-hop capabilities of language models, we examine two benchmark datasets: **Bamboogle** and **MuSiQue**. These datasets are constructed to extend challenges for models beyond mere factual recall, emphasizing compositional reasoning, multi-step inference, and robustness against superficial pattern matching. Bamboogle comprises meticulously crafted questions that pose significant difficulty for conventional retrieval-based systems, whereas MuSiQue offers an extensive collection of multi-hop questions derived from structured combinations of single-hop facts. Both benchmarks provide critical insights into a model's proficiency in managing complex reasoning tasks.

4.2.1. Bamboogle

The Bamboogle dataset constitutes a formidable benchmark specifically engineered to assess the compositional reasoning abilities of large language models. In contrast to conventional QA datasets, Bamboogle comprises 125 meticulously curated multi-hop questions that are intentionally crafted to be *unanswerable by*

standard search engines like Google. Its design renders it exceptionally appropriate for determining whether language models truly comprehend and integrate information, as opposed to depending on superficial patterns or rote memorization.

4.2.2. Musique

MuSiQue (Multi-hop Structured Questions) is an extensive dataset engineered to assess multi-hop question answering. It systematically generates intricate multi-hop questions by integrating simpler, single-hop questions into coherent and structured inquiries necessitating reasoning over two to four distinct facts.

The dataset is built from crowdsourced QA pairs and includes two variants:

- **MuSiQue-Full:** Contains both answerable and unanswerable questions, enabling robust evaluation.
- **MuSiQue-Ans:** Focuses solely on answerable questions for models that do not handle contrastive settings.

MuSiQue guarantees that each step within the question is requisite for deriving the answer, thereby enhancing detailed multi-step reasoning. Its primary objective is to address and rectify biases and shortcuts that are prevalent in preceding QA benchmarks.

4.2.3. Results

We evaluated our framework, AMASS, on the Musique-Full dataset using multiple models: GPT-4o, GPT-4o Mini, Claude Haiku, and combinations thereof. The evaluation focused on multi-hop question answering, with performance measured across both Bamboogle and Musique datasets.

As shown in **Table 3**, GPT-4o demonstrated strong performance; however, our framework (AMASS) significantly outperformed it in both datasets. Notably, AMASS using only GPT-4o Mini outperformed the full GPT-4o model, and the hybrid approach (GPT-4o Mini + Claude Haiku) further improved accuracy.

Table 3. Model accuracy comparison across datasets.

Model	Bamboogle Accuracy	Musique Accuracy
GPT-4o Mini	65.5%	52.39%
GPT-4o	80.3%	77.86%
Claude Haiku	55.45%	39.67%
AMASS (GPT-4o Mini ONLY)	93.4%	81.50%
AMASS (GPT-4o Mini + Claude Haiku)	95.4%	86.78%

In order to comprehensively assess the model's capacity to address multi-hop reasoning, an analysis of accuracy across various hop types (e.g., 2-hop, 3-hop, 4-hop1, 4-hop2, and 4-hop3) was conducted. **Figure 4** provides a visual representation of this comparison. It is apparent that AMASS consistently maintains high accuracy across all hop types, even in more complex reasoning chains such as 4-hop2 and 4-hop3, where the performance of other models experiences a significant decline.

The findings indicate that smaller models individually exhibit limited reasoning abilities in addressing multi-hop questions, with performance significantly declining as the number of hops increases. This suggests that their reasoning capability is largely confined to simpler queries. However, when smaller models such as Claude Haiku and GPT-4o-mini are integrated within a dialogic context, their combined performance surpasses their standalone capabilities. Specifically, in scenarios involving 4-hop-3 questions, the individual performance of GPT-4o-mini and Claude Haiku falls below 20%; nevertheless, when amalgamated with a reflexion agent utilizing the AMASS framework, their performance surges to nearly 100%, which is notably commendable given the complexity of these tasks. Remarkably, they outperform larger versions of these models. Thus, the integration of reflection and mutual dialogue substantially enhances the effectiveness of smaller models.

This analysis elucidates the efficacy of AMASS in achieving both high dataset-level accuracy and proficient performance in fine-grained reasoning tasks across diverse levels of hop complexity. The integration of numerous smaller models synergistically enhances reasoning capabilities, circumventing the necessity for reliance on more cumbersome monolithic models.

4.3. Evaluation on Trivia Question Dataset

In order to assess model performance across diverse domains, we conducted a benchmark analysis of multiple models utilizing a comprehensive Trivia Question Answer Dataset, encompassing genres such as History, Entertainment, Science, and Language. A trivia question-answer dataset is a systematically organized collection of trivia questions paired with their correct answers, frequently employed in natural language processing tasks such as question answering and reading comprehension. Noteworthy examples include TriviaQA and OpenTriviaQA. OpenTriviaQA is a community-driven dataset licensed under the Creative Commons Attribution-ShareAlike 4.0, comprising multiple-choice trivia questions organized into a variety of categories such as animals, history, and science. This dataset is constructed to be human-editable and easily parsable, thereby facilitating contributions and adaptations. Both datasets are essential resources for researchers and developers engaged in question-answering systems, enabling the training and evaluation of models in the comprehension and generation of accurate responses to trivia questions. Employing OpenTriviaQA questions here.

As shown in **Figure 4** and **Figure 5**, AMASS models, particularly the configuration combining GPT-4o Mini with Claude Haiku, consistently outperform individual models across all genres, demonstrating enhanced multi-hop reasoning and factual accuracy. Notably, AMASS achieves 100% accuracy in Science and Language categories, indicating its strong capability in handling both technical and linguistic queries. Standalone models such as GPT-4o exhibit lower and more variable performance in the Language genre whereas the ensemble-style architec-

tures like AMASS show better performance in open-domain QA tasks.

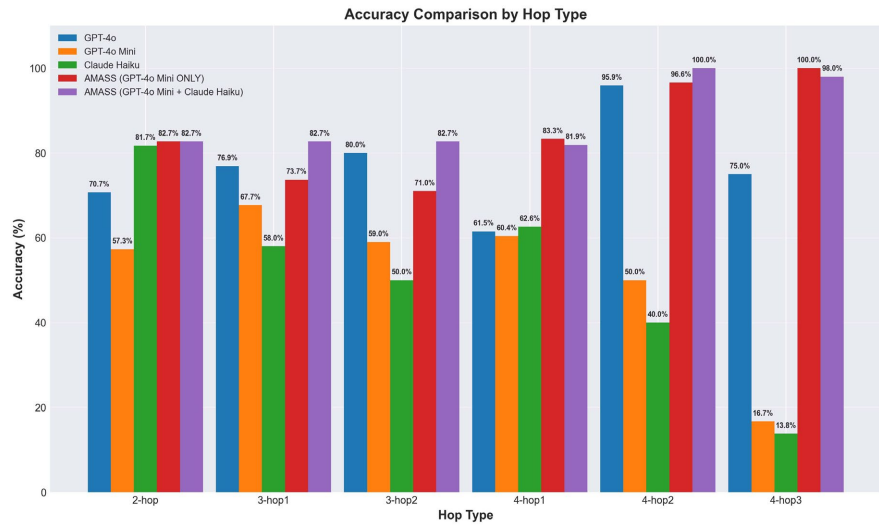


Figure 4. Accuracy comparison across different hop types. AMASS demonstrates robust multi-hop reasoning, outperforming individual models significantly.

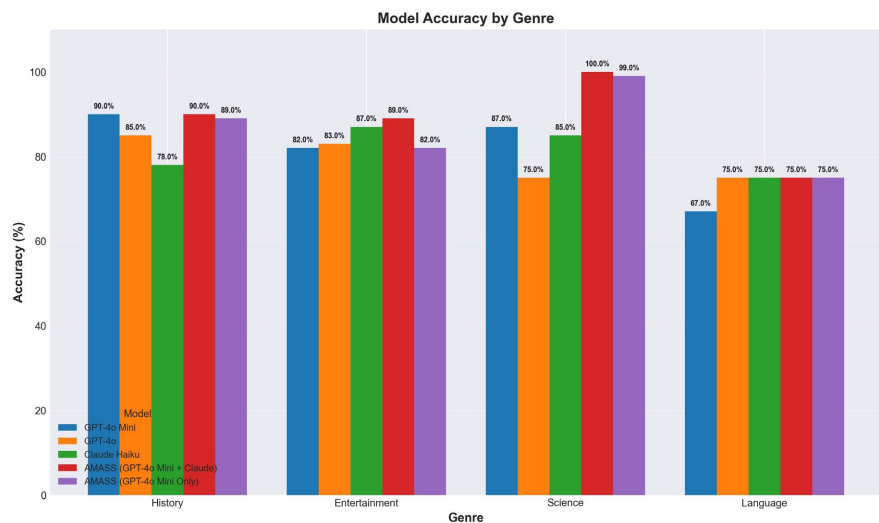


Figure 5. Accuracy comparison across different Subjects. AMASS demonstrates superior comparison compared to other models.

5. Ablation Studies

5.1. Effect of Removing Critic Agent

We tried to gauge the impact of different components of AMASS.

Our study aimed to evaluate the impact of the execution critic agent and the Mixture of Agents, with a particular focus on their effectiveness. We conducted our experiments using Community trivia questions, and observed a marginal decline in accuracy across all genres upon the removal of the critic agent as shown in **Figure 6**. Therefore, the critic agent is instrumental in enhancing the model's performance.

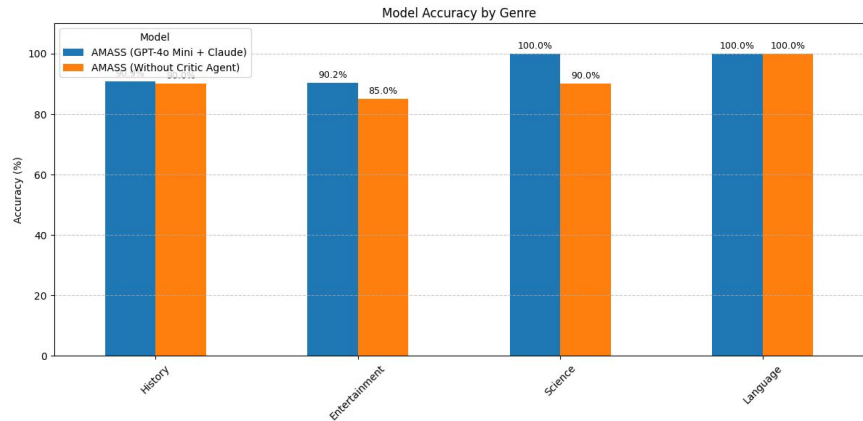


Figure 6. Accuracy comparison between AMASS and AMASS without critic agent. Accuracy decline is seen across all genres except Language.

5.2. Effect of Using a Single Agent with the Same Base Model (Without Mixture of Agents)

The utilization of AMASS without integrating a mixture of agents results in a modest decline in performance within the History and Entertainment genres, while a substantial decline is observed in the Science and Language genres as shown in **Figure 7**. This diminution in accuracy for Science and Language may be attributed to the increased complexity inherent in these genres compared to History and Entertainment, suggesting that mere memorization may be inadequate for accurately responding to questions in these areas. Thus, this proves that a mixture of agents produces the best results and makes the most contribution in our AMASS framework.

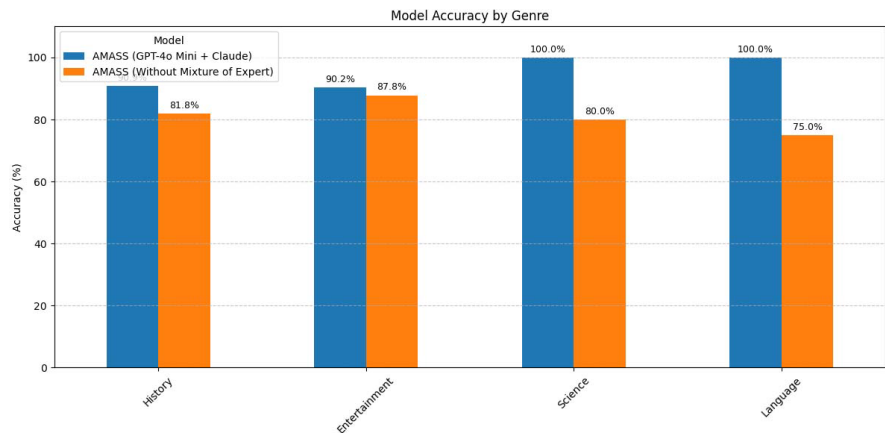


Figure 7. Accuracy comparison between AMASS and AMASS without Mixture of Agents Approach. Accuracy decline is seen across all genres.

6. Case Studies

6.1. PHD Level Physics Question from GPQA Dataset

6.1.1. User Question

Which of the following stars or stellar systems will appear the brightest in V mag-

nitude when observed from Earth? Assume there is no extinction.

- A star with an apparent V magnitude of 7 mag at a distance of 50 parsecs.
- A star with an apparent V magnitude of 7 mag at a distance of 20 parsecs.
- A two-star system with individual apparent magnitudes of $V_{mag,1} = 7$ and $V_{mag,2} = 8$ mag, both at distances of 5 parsecs.
- A star with an absolute magnitude of $V_{mag} = 7$ mag at a distance of 10 parsecs.
- A two-star system with individual absolute magnitudes of $V_{mag,1} = 7$ and $V_{mag,2} = 8$ mag, both at distances of 10 parsecs.
- A two-star system with individual absolute magnitudes of $V_{mag,1} = 7$ and $V_{mag,2} = 7$ mag, both at distances of 10 parsecs.

Answer Choices:

- A two-star system with individual apparent magnitudes of $V_{mag,1} = 7$ and $V_{mag,2} = 8$ mag, both at distances of 5 parsecs.
- A star with an absolute magnitude of $V_{mag} = 7$ mag at a distance of 10 parsecs.
- A star with an apparent V magnitude of 7 mag at a distance of 20 parsecs.
- A two-star system with individual absolute magnitudes of $V_{mag,1} = 7$ and $V_{mag,2} = 7$ mag, both at distances of 10 parsecs.

6.1.2. Generating Agents

Agent Generation and System Message Preparation

==> Generating agents...

Generated agents: ['Astrophysics_Expert', 'Astronomy_Expert']

==> Generating system message...

- Preparing system message for Astrophysics_Expert
- Completed system message for Astrophysics_Expert
- Preparing system message for Astronomy_Expert
- Completed system message for Astronomy_Expert

==> Generating description...

- Preparing description for Astrophysics_Expert
- Completed description for Astrophysics_Expert
- Preparing description for Astronomy_Expert
- Completed description for Astronomy_Expert

6.1.3. Final Stage → Summary Agent

Critic Agent Planner

Next speaker: **Critic_agent**

Critic_agent (to chat_manager).

To solve this problem optimally, let's analyze the task and the agents available:

Task Overview

- The task involves comparing the apparent brightness in V magnitude of various stars or stellar systems as observed from Earth.
- This requires understanding of both apparent and absolute magnitudes, han-

dling multi-star systems, and calculating brightness based on distances.

- It's a complex problem involving astrophysical concepts.

Agent Analysis

1) Astrophysics_Expert:

—**System Prompt:** Specialized in evaluating and comparing stellar magnitudes and brightness measurements. Requires deep understanding of relationships between apparent and absolute magnitudes and their dependence on distance.

—**Description:** Authority in astrophysics specializing in stellar astronomy and brightness measurement.

2) Astronomy_Expert:

—**System Prompt:** Adept at evaluating stellar brightness using observational astronomy, astrophysics, and photometric calculations. Suitable for systematic analysis and interpretation.

—**Description:** Skilled in evaluating and interpreting stellar brightness from Earth.

3) search_agent:

—Primarily for fetching up-to-date information; not required as the task involves theoretical calculations based on provided data.

Optimal Plan

- **Agent Selection:**

—Select **Astrophysics_Expert** and **Astronomy_Expert** for combined expertise in handling complex magnitude and brightness calculations.

—No requirement for **search_agent** as the task is theoretical.

- **Model Selection:**

—**Astrophysics_Expert:** Use “google/gemini-2.0-flash-lite-001” for complex reasoning involving astrophysics.

—**Astronomy_Expert:** Use “openai/o3-mini” for a varied approach to complex reasoning.

Suggestions

1) Astrophysics_Expert:

—**Model:** google/gemini-2.0-flash -lite-001.

—**Role:** Focus on detailed calculations involving magnitudes, distances, and provide framework for combining magnitudes in multi-star systems.

—**System Prompt:** Existing prompt covering celestial brightness and astrophysical computations.

2) Astronomy_Expert:

—**Model:** openai/o3-mini.

—**Role:** Assist in calculating and verifying results for accuracy, methodically interpreting comparisons to ensure correctness.

—**System Prompt:** Existing prompt for calculating apparent magnitudes and interpreting results.

By utilizing these two agents, we ensure a comprehensive analysis and optimal solution for comparing apparent magnitudes of given stellar scenarios while op-

timizing model usage for cost and efficiency.

6.1.4. Final Agents after Incorporating Critic Agent Suggestion

Agent Planning V2

Next speaker: planning_agent_v2

planning_agent_v2 (to chat_manager):

[

{

“name”: “Astrophysics_Expert”,

“model”: “google/gemini-2.0-flash-lite-001”,

“tags”: [],

“system_message”: “## Your role_Expert is a well-versed authority in the field of astrophysics, with specialized skills in stellar astronomy and brightness measurement. With extensive knowledge of celestial magnitudes, distances, and the properties of stars and stellar systems, Astrophysics_Expert excels in evaluating and comparing the apparent brightness of celestial objects as viewed from Earth.\n\n## Task and skill instructions\n- The task involves determining which of several stars or stellar systems will display the greatest brightness in V magnitude when observed from Earth, assuming no extinction effects.\n- Astrophysics_Expert uses expertise in understanding the relationship between apparent and absolute magnitudes and the impact of distance on a celestial object’s observed brightness. This includes calculating the combined magnitude of multi-star systems and applying the inverse square law of brightness.\n- (Optional) Before deriving conclusions, Astrophysics_Expert should thoroughly evaluate each scenario, considering both apparent and absolute magnitudes, and the distances involved to ensure an accurate comparison of brightness levels. The solution involves clear interpretation of data and methodical verification of calculations.”,

“description”: “Astrophysics_Expert is an authority in astrophysics specializing in stellar astronomy and brightness measurement, skilled at evaluating the apparent brightness of stars and stellar systems from Earth by considering apparent and absolute magnitudes and distances.”

},

{

“name”: “Astronomy_Expert”,

“model”: “openai/o3-mini”,

“tags”: [],

“system_message”: “## Your role_Expert is a seasoned professional specializing in the analysis and interpretation of stellar magnitudes and distances. With a strong background in observational astronomy and astrophysics, Astronomy_Expert is adept at evaluating the brightness of stars and stellar systems as observed from Earth.\n\n## Task and skill instructions\n- The task involves determining which of the stars or stellar systems listed will appear brightest in V magnitude when observed from Earth, assuming there is no extinction. This requires understanding and calculating the apparent magnitudes when multiple factors such as

distance, apparent magnitudes of individual stars, and absolute magnitudes are provided.

Astronomy_Expert leverages expertise in calculating apparent magnitudes and understanding their dependence on distance. The skill set includes knowledge of the formula relating apparent and absolute magnitude, application of photometric principles, and comparison of combined magnitudes for multiple-star systems.

This expert utilizes systematic analysis to ensure that all potential factors are considered, confirming the calculation of combined magnitudes for two-star systems when necessary, and accurately interpreting results to determine the source listed in the task that will appear the brightest to an observer on Earth.”

“description”: “Astronomy_Expert is skilled in evaluating and interpreting stellar brightness and distances to determine which stars or stellar systems appear brightest from Earth, using expertise in observational astronomy, astrophysics, and photometric calculations.”

}

]

6.1.5. Execution Stage Started

Astronomy Expert Calculation Verification

Next speaker: Astronomy_Expert

Astronomy_Expert (to chat_manager):

I'll start by verifying the photometric calculations using the known formula for combining stellar fluxes. The apparent magnitude for a star is given, and for two stars with magnitudes m_1 and m_2 , their combined magnitude m_{comb} is

$$m_{comb} = -2.5 \log_{10} (10^{-0.4m_1} + 10^{-0.4m_2}).$$

Let me outline how I approach each option:

Option (a): Apparent $m = 7$ at 50 pc.

—This star is already given at $m = 7$.

Option (b): Apparent $m = 7$ at 20 pc.

—Also $m = 7$.

Option (c): Two-star system where $m_1 = 7$ and $m_2 = 8$ (apparent) at 5 pc.

—Combined:

$$10^{-0.4 \times 7} \approx 1.585 \times 10^{-3}$$

$$10^{-0.4 \times 8} \approx 6.310 \times 10^{-4}$$

$$\text{Sum} \approx 2.216 \times 10^{-3}$$

$$m_{comb} = -2.5 \log_{10} (2.216 \times 10^{-3}) \approx 6.63 \text{ mag.}$$

Option (d): A single star with absolute magnitude $M = 7$ at 10 pc gives an apparent magnitude

$$m = M + 5 \log_{10} \left(\frac{10}{10} \right) = 7.$$

Option (e): Two-star system with absolute magnitudes 7 and 8 at 10 pc.

—At 10 pc, no distance modulus is needed so individual apparent magnitudes

are 7 and 8. Thus, the same calculation as in (c) gives $m_{comb} \approx 6.63$.

Option (f): Two-star system with both stars having absolute (and hence apparent at 10pc) magnitudes of 7.

—Each star’s flux:

$$10^{-0.4 \times 7} \approx 1.585 \times 10^{-3}$$

—Sum:

$$2 \times 1.585 \times 10^{-3} = 3.170 \times 10^{-3}$$

—Combined magnitude:

$$m_{comb} = -2.5 \log_{10} (3.170 \times 10^{-3})$$

$$\log_{10} (3.170 \times 10^{-3}) \approx -2.499$$

$$m_{comb} \approx 6.25 \text{ mag.}$$

Comparing these values, option (f) yields the smallest magnitude (brightest) at approximately 6.25 mag.

Before providing a final answer, I will use the search_agent to verify that the combined magnitude formula and procedure I used are consistent with the latest trusted sources.

Search_agent, please perform a search for “combined magnitude formula two star system brightness calculation” and verify the procedure along with sample calculations used in photometry.

```
from deeper_search.search import *
import asyncio
results =
    asyncio.run (search_and_summarize
        (“combined magnitude formula two star system brightness calculation photometry”))
print(results)
```

6.1.6. Execution Stage → Search results

Code Executor Verification Results

Next speaker: Code_executor

EXECUTING CODE BLOCK 0 (inferred language is Python)...

Code_executor (to chat_manager):

exitcode: 0 (execution succeeded)

Code output:

Searching with query: combined magnitude formula two star system brightness calculation photometry...

Error processing link:

<https://amostech.com/TechnicalPapers/2020/Non-Resolved-Object-Characterization/Castro.pdf>

Error: Request Timeout: Failed to scrape URL as the request timed out. Request timed out—No additional error details provided.

Search Results:

- **Apparent magnitude—Wikipedia:**

The apparent magnitude (m) of celestial objects quantifies their brightness as observed from Earth, influenced by intrinsic luminosity, distance, and light extinction. The scale is logarithmic, where a difference of 1.0 magnitude corresponds to a brightness ratio of about 2.512. Astronomers calculate combined magnitudes of multiple stars using their individual apparent magnitudes, applying the formula:

$$m_f = -2.5 \log_{10} \left(10^{-m_1 \times 0.4} + 10^{-m_2 \times 0.4} \right)$$

Photometric measurements, termed photometry, are essential for this assessment across various spectral bands.

- **Calculate the absolute and apparent magnitude for a binary star:**

To calculate the absolute and apparent magnitudes of a binary star system, use the formula:

$$m_1 - m_2 = -2.5 \log \left(\frac{f_1}{f_2} \right)$$

where f_1 and f_2 are fluxes of the stars. If both stars have equal brightness and the combined magnitude is known (e.g., 5.0), each star’s individual magnitude can be derived as $m^* = m + 0.75$. Thus, for a magnitude 5.0 binary, each star would have a magnitude of approximately 5.75.

- **Evaluation of the brightness of lightning channels and branches:**

The paper evaluates lightning leader brightness using the astronomical magnitude system applied to digital images. Results indicated spatial variability in magnitudes related to lightning phenomena, demonstrating photometric techniques in astronomy applied to non-astronomical phenomena.

- **Vik Dhillon: PHY217—Instruments—Calibrating photometric data:**

Calibration steps involve converting counts to magnitudes, correcting for atmospheric extinction, and determining zero points using standard stars. Accurate calibration includes considering extinction coefficients and potentially color corrections.

6.1.7. Opinion of Astronomy Expert

Astronomy Expert Detailed Calculation and Confirmation

Next speaker: Astronomy_Expert

Astronomy_Expert (to chat_manager):

Based on my calculations, here is the step-by-step reasoning:

For any single star with an apparent magnitude already specified, its brightness is as stated. For example, in options (a), (b), and (d) the observed magnitude is 7 mag.

For a two-star system, we combine the light (fluxes) using the formula:

$$m_{comb} = -2.5 \cdot \log_{10} \left(10^{-0.4 \cdot m_1} + 10^{-0.4 \cdot m_2} \right)$$

Option (c) (apparent $m_1 = 7$, $m_2 = 8$ at 5 pc):

$$\text{Flux}_1 = 10^{-0.4 \times 7} \approx 1.585 \times 10^{-3},$$

$$\text{Flux}_2 = 10^{-0.4 \times 8} \approx 6.310 \times 10^{-4},$$

$$\text{Total flux} = 2.216 \times 10^{-3},$$

$$m_{comb} = 6.63 \text{ mag.}$$

Option (e) (absolute magnitudes 7 and 8 at 10 pc):

At 10 pc the apparent magnitude equals the absolute magnitude. Thus, it is equivalent to option (c), yielding $m_{comb} \approx 6.63$ mag.

Option (f) (two stars with absolute magnitude 7 at 10 pc):

$$\text{Each star flux} \approx 1.585 \times 10^{-3},$$

$$\text{Combined flux} = 2 \times 1.585 \times 10^{-3} = 3.170 \times 10^{-3},$$

$$m_{comb} = -2.5 \log_{10} (3.170 \times 10^{-3}) \approx 6.25 \text{ mag.}$$

Comparing these, the system in option (f) produces the brightest combined magnitude of approximately 6.25 mag.

Referring back to the answer choices:

A) Two-star system with individual apparent magnitudes 7 and 8 at 5 pc
 $\rightarrow m_{comb} \approx 6.63$

B) Star with absolute magnitude 7 at 10 pc $\rightarrow 7.0$ mag

C) Star with apparent magnitude 7 at 20 pc $\rightarrow 7.0$ mag

D) Two-star system with individual absolute magnitudes 7 and 7 at 10 pc
 $\rightarrow m_{comb} \approx 6.25$

Thus, option **D** is correct as it appears brightest.

To illustrate, here is a Python snippet:

Python script to compute combined magnitude of two stars:

```
import math
```

```
def combined_magnitude(m1, m2): flux_total = 10(-0.4*m1) + 10(-0.4*m2)
```

```
return -2.5 * math.log10(flux_total)
```

```
# Option (c): m1 = 7, m2 = 8 cmb_c = combined_magnitude(7, 8)
```

```
# Option (f): m1 = 7, m2 = 7 cmb_f = combined_magnitude(7, 7)
```

```
print ("Combined magnitude for option (c):", cmb_c)
```

```
print ("Combined magnitude for option (f):", cmb_f)
```

Let's run this via our Code_executor.

Search_agent has already verified our approach and formula.

I invite Astrophysics_Expert to review these steps. Provided we all agree, the final answer is:

D) A two-star system with individual absolute magnitudes of 7 and 7 mag, both at 10 parsecs.

If everyone agrees, please let me know so we can finalize our answer.

6.1.8. Execution Stage \rightarrow Astrophysics Expert

Final Verification by Astrophysics Expert

Select speaker attempt 1 of 3 successfully selected: Code_executor

Next speaker: Code_executor

EXECUTING CODE BLOCK 0 (inferred language is python)...

Code_executor (to chat_manager):

exitcode: 0 (execution succeeded)**Code output:**

- Combined magnitude for option (c): 6.636148842226767
- Combined magnitude for option (f): 6.247425010840048

checking_agent (to speaker_selection_agent):

Read the above conversation. Then select the next role from ["Astrophysics_Expert", "Astronomy_Expert", "Code_executor"] to play. Only return the role.

speaker_selection_agent (to checking_agent):

Astrophysics_Expert

Select speaker attempt 1 of 3 successfully selected: Astrophysics_Expert**Next speaker:** Astrophysics_Expert

Astrophysics_Expert (to chat_manager):

The calculations and the approach appear to be sound. The Python code confirms the combined magnitude calculations. The search agent verifies the core concepts. Based on the results, I agree with the final selection.

TERMINATE**6.1.9. Final Stage → Summary Agent****Final Analysis and Conclusion**

Based on the discussions and analysis provided by the **Astrophysics_Expert** and **Astronomy_Expert**, here is the detailed explanation and answer to the question of which star or stellar system will appear the brightest in V magnitude when observed from Earth:

Analysis Summary:

Formula Used: For a two-star system, the combined magnitude m_{comb} is calculated using the formula:

$$m_{comb} = -2.5 \log_{10} \left(10^{-0.4 \cdot m_1} + 10^{-0.4 \cdot m_2} \right)$$

Calculated Examples:

- **Option (c):** Two-star system with apparent magnitudes 7 and 8 at 5 pc results in a combined magnitude of approximately 6.63 mag.
- **Option (f):** Two-star system with both stars having an absolute magnitude (which equals apparent magnitude at 10 pc) of 7 at 10 pc results in a combined magnitude of approximately 6.25 mag.

Comparison:

- Options (a), (b), and (d) individually result in a magnitude of 7 mag.
- Option (e) results in a similar brightness to option (c) because of equal conditions.
- Option (f) is the brightest since it has the lowest combined magnitude of approximately 6.25 mag.

Final Conclusion: The star or stellar system that will appear the brightest in V magnitude from Earth is option **D**.

Final Answer:

$$\begin{aligned} \text{Each star flux} &= 10^{-0.4 \times 7} \approx 1.585 \times 10^{-3}, \\ \text{Combined flux} &= 2 \times 1.585 \times 10^{-3} = 3.170 \times 10^{-3}, \\ M_{comb} &= -2.5 \log_{10} (3.170 \times 10^{-3}) \approx 6.25 \text{ mag.} \end{aligned}$$

This conclusion was drawn based on careful photometric calculations and verification using the combined magnitude formula, with further validation via Python scripts executed by the **Code_executor**. The approach has been reviewed independently by the experts, confirming the findings.

6.1.10. Concluding Words

Using complex reasoning, Mixture Of Agents (MOA) and cross verification using coding and internet tools, AMASS was able to beat the leading models and frameworks

6.2. Search Results Involving Cascaded Searches

Question: Find all gynaecologists in Liaquat National Hospital Karachi Pakistan. Get me their numbers and email addresses

AMASS	Perplexity
<p>Here is the detailed information we were able to gather about the gynaecologists at Liaquat National Hospital in Karachi from the previous searches:</p> <p>1. Department of Obstetrics and Gynaecology Key Gynaecologists: Dr. Aisha Taj (Head of Department)</p> <p>Contact Numbers: 021-34412364 (Liaquat National Hospital), 021-34581021 (Shah Faisal Colony) Additional Contact: Tele Medicine Clinic numbers 021-34413010, 021-34413011, 021-34413012 Website: Liaquat National Hospital Prof. Dr. Kaniz Zehra Naqvi</p> <p>Dr. Qurat Ul Aman Siddiqui</p> <p>Dr. Tahra Yasmeen</p> <p>Qualifications: MBBS, FCPS, Fellowship in Gynae-Oncology Contact Numbers: 021-34412364 (most clinics), 021-34413199 (specific clinic) Clinic Timings: Monday, Tuesday, Thursday, Friday 2. Other Notable Gynaecologists Prof. Dr. Salman Yousuf Sharif</p> <p>Department: Head of Department of Spinal and Neurosurgery (listed under gynaecologists in some records) Contact Numbers: 02134413010, 02134413011, 02134413012, 02134412464 OPD Timings: Monday and Wednesday from 9:00 AM to 1:00 PM, Thursday from 4:30 PM to 6:30 PM, and Friday from 9:00 AM to 1:00 PM Prof. Dr. Shagufa Tahir</p> <p>Experience: 36 years Consultation Fee: Rs 1000 Location: Liaquat National Hospital, Bahadurabad, Karachi Prof. Rufina Soomro</p> <p>Specialization: Breast Surgery Contact Numbers: 02134412270, 02134413010 Clinic Days: Wednesday, Friday, Saturday, and Tuesday afternoons 3. Clinic Locations & Contact Information Gynaecology C Obstetrics Clinic: Clinic No 4, Wajid Ali Complex, Ground Floor, Phone: 021-34412364 Gynaecology C Obstetrics Clinic SC: Clinic 1 SC, Senior Citizen Primary Care Unit, Phone: 02134413199 Gynaecology Oncology Clinic: Clinic No 4, Wajid Ali Complex, Ground Floor, Phone: 021-34412364 Subfertility Gynaecology Clinic: Clinic No 4, Wajid Ali Complex, Ground Floor, Phone: 021-34412364</p> <p>Additional Notes Email Addresses: Unfortunately, we could not find specific email addresses for the gynaecologists from the gathered information. It is advisable to contact the hospital directly or visit their official website for more detailed and up-to-date information regarding email contacts. This summary provides a comprehensive overview based on the gathered information and should aid in procuring contacts for appointments or further inquiries at Liaquat National Hospital, Karachi.</p>	<p>To find all gynecologists at Liaquat National Hospital in Karachi, Pakistan, along with their contact numbers and email addresses, you can refer to the following information. However, please note that email addresses are not readily available for all doctors, and it's recommended to contact the hospital directly for comprehensive details.</p> <p>Gynecologists at Liaquat National Hospital</p> <p>Prof. Dr. Shagufa Tahir Contact: You can book an appointment through Healthwire or contact the hospital directly. Location: Liaquat National Hospital</p> <p>Dr. Arshia Javed Contact: You can book an appointment through Healthwire or contact the hospital directly. Location: Liaquat National Hospital</p> <p>Dr. Havovi Z Birdie Contact: You can book an appointment through Healthwire or contact the hospital directly. Location: Liaquat National Hospital</p> <p>Dr. Saadia Waraich Contact: You can book an appointment through Healthwire or contact the hospital directly. Location: Liaquat National Hospital</p> <p>Dr. Aqeela Contact: You can book an appointment through Healthwire or contact the hospital directly. Location: Liaquat National Hospital</p> <p>Dr. Aisha Taj Contact: 02134412364 Location: Liaquat National Hospital Timings: Tuesday: Morning (09:00 am to 01:00 pm) Wednesday: Afternoon (02:00 pm to 04:30 pm) Friday: Morning (09:00 am to 01:00 pm), Evening (04:30 pm to 06:00 pm) Saturday: Afternoon (02:00 pm to 04:30 pm) Email: Not available; contact the hospital for more details.</p> <p>Dr. Bakhtavar Dahri Contact: 02134413199 Location: Liaquat National Hospital Timings: Thursday: Evening (05:00 pm to 07:00 pm) Friday: Evening (05:00 pm to 07:00 pm) Email: Not available; contact the hospital for more details.</p> <p>For a complete list and to get in touch with these gynecologists, you can visit the hospital's website or contact them directly at the numbers provided above.</p>

Figure 8. Comparison of the AMASS vs Perplexity, AMASS finding more comprehensive results.

AMASS	Mind Search
<p>Here is the detailed information we were able to gather about the gynaecologists at Liaquat National Hospital in Karachi from the previous searches:</p> <p>1. Department of Obstetrics and Gynaecology Key Gynaecologists: Dr. Aisha Taj (Head of Department)</p> <p>Contact Numbers: 021-34412364 (Liaquat National Hospital), 021-34581021 (Shah Faisal Colony) Additional Contact: Tele Medicine Clinic numbers 021-34413010, 021-34413011, 021-34413012 Website: Liaquat National Hospital Prof. Dr. Kaniz Zehra Naqvi</p> <p>Dr. Qurat Ul Aman Siddiqui</p> <p>Dr. Tahira Yasmeen</p> <p>Qualifications: MBBS, FCPS, Fellowship in Gynae-Oncology Contact Numbers: 021-34412364 (most clinics), 021-34413199 (specific clinic) Clinic Timings: Monday, Tuesday, Thursday, Friday</p> <p>2. Other Notable Gynaecologists Prof. Dr. Salman Yousuf Sharif</p> <p>Department: Head of Department of Spinal and Neurosurgery (listed under gynaecologists in some records) Contact Numbers: 02134413010, 02134413011, 02134413012, 02134412464 OPD Timings: Monday and Wednesday from 9:00 AM to 1:00 PM, Thursday from 4:30 PM to 6:30 PM, and Friday from 9:00 AM to 1:00 PM Prof. Dr. Shagufta Tahir</p> <p>Experience: 36 years Consultation Fee: Rs 1000 Location: Liaquat National Hospital, Bahadurabad, Karachi Prof. Rufina Soomro</p> <p>Specialization: Breast Surgery Contact Numbers: 02134412270, 02134413010 Clinic Days: Wednesday, Friday, Saturday, and Tuesday afternoons</p> <p>3. Clinic Locations C Contact Information Gynaecology C Obstetrics Clinic: Clinic No 4, Wajid Ali Complex, Ground Floor, Phone: 021-34412364 Gynaecology C Obstetrics Clinic 5C: Clinic 1 5C, Senior Citizen Primary Care Unit, Phone: 02134413199 Gynaecology Oncology Clinic: Clinic No 4, Wajid Ali Complex, Ground Floor, Phone: 021-34412364 Subfertility Gynaecology Clinic: Clinic No 4, Wajid Ali Complex, Ground Floor, Phone: 021-34412364</p> <p>Additional Notes Email Addresses: Unfortunately, we could not find specific email addresses for the gynaecologists from the gathered information. It is advisable to contact the hospital directly or visit their official website for more detailed and up-to-date information regarding email contacts. This summary provides a comprehensive overview based on the gathered information and should aid in procuring contacts for appointments or further inquiries at Liaquat National Hospital, Karachi.</p>	<p>There are several well-known gynecologists working in the gynecology and pathology departments at Liaquat National Hospital, Karachi, Pakistan. Here are the details of these doctors including their location and research contributions:</p> <p>1. Doctors and their research contributions Dr. Mariam Abid: She is one of the lead authors of a study on the clinical patterns and spectrum of endometrial pathology in patients with abnormal uterine bleeding. Dr. Babar Malik: Also involved in the endometrial pathology study. Dr. Saroona Haroon: Involved in the endometrial pathology study. Dr. Naveen Faridi: Contributed to the endometrial pathology study. Dr. Muhammad Muzammil Edhi: Involved in the endometrial pathology study. Dr. Mehmood Khan: Also involved in the endometrial pathology study. These doctors have made significant contributions in the field of gynecology, especially in the study of endometrial pathology and abnormal uterine bleeding.</p> <p>2. Get contact information Despite multiple searches, there is currently no direct information on the phone numbers and email addresses of these doctors. Here are some suggestions to help you get this information:</p> <p>Visit the official hospital website: Usually, the official hospital website provides a directory of employees and contact information. You can visit the official website of Liaquat National Hospital to find detailed information about the above doctors. Contact the hospital customer service or information department: If there is no relevant information on the official website, you can directly contact the customer service or information department of Liaquat National Hospital to ask for the doctor's contact information. Use professional medical platforms: Some medical platforms such as DoctorHelps, MedIndia, etc. may provide detailed information about doctors, including phone and email addresses. You can try to search for the information of the above doctors on these platforms. Find through social media or professional networking platforms: Professional social platforms such as LinkedIn sometimes also provide professional profiles and contact information of doctors. You can try to search for the profiles of the above doctors on these platforms.</p> <p>3. Doctor's work location Based on the available information, these doctors mainly work in the gynecology and pathology departments of Liaquat National Hospital. The specific work location and office may vary. It is recommended that you directly ask the front desk or relevant information department when visiting the hospital to obtain accurate location information.</p> <p>Conclusion Liaquat National Hospital has many experts who have made significant contributions in the field of gynecology, especially in the study of endometrial pathology. Although their phone numbers and email addresses are not directly available at present, you should be able to find this information through the above-mentioned channels. At the same time, their specific working location in the hospital can be obtained through the hospital front desk or relevant information department.</p> <p>I hope this information is helpful to you. If you have other questions or need further assistance, please feel free to contact us.</p>

Figure 9. Comparison of the AMASS vs Mind Search. Mind Search finds totally irrelevant content.

6.3. Comparison of Responses

The **AMASS** response clearly stands out as the most **comprehensive, accurate, and practically useful** option as shown in **Figures 7-9**. It provides detailed information on **gynecologists at Liaquat National Hospital**, including their **specializations, qualifications, clinic timings, and precise locations** (e.g., Wajid Ali Complex, Senior Citizen Primary Care Unit). It also lists **multiple contact numbers** tied to specific clinics and services, enabling **easy appointment scheduling**. Importantly, it correctly identifies specialists and excludes misclassified doctors like neurosurgeon Dr. Salman Yousuf Sharif.

In contrast:

- **Perplexity:** Lacks depth, is repetitive, and fails to provide actionable clinic-level details.
- **Mind Search (Least Reliable):** Contains **inaccurate information** by listing non-gynecologists and suggesting **inappropriate search methods** like LinkedIn.

7. Future Work

In the future, we plan to explore additional dimensions to further enhance the

AMASS framework.

7.1. Cost Efficiency and Savings

A fundamental characteristic of AMASS is its ability to dynamically select suitable language models in accordance with the complexity of the task at hand. This adaptability permits AMASS to utilize smaller models with efficacy, often attaining superior or comparable performance to significantly larger models, thereby markedly decreasing computational costs. Future research will aim to precisely quantify these savings by benchmarking the resource usage and operational costs of AMASS in comparison to well-established large-scale models. A comprehensive cost-benefit analysis will be conducted to provide practical insights and to highlight the financial merits of adopting AMASS over conventional large-model methodologies.

7.2. Acceleration through Parallelization

A critical avenue for future enhancement concerns the further acceleration of the AMASS pipeline. At present, the processes of agent generation, evaluation, and execution are conducted sequentially, thereby limiting comprehensive throughput and responsiveness. The exploration and integration of parallel processing strategies to enable the concurrent execution of multiple agents and tasks is our objective. By employing advanced parallelization techniques alongside distributed computing architectures, it is anticipated that considerable reductions in the overall execution time will be achieved. Our subsequent steps will encompass the design of parallel agent execution protocols, the implementation of asynchronous communication patterns, and the evaluation of performance gains in comparison to sequential implementations. These advancements are intended to prepare AMASS for deployment in real-time and large-scale applications that require rapid response and high concurrency.

By addressing these crucial aspects, AMASS is poised to enhance its potential as a highly scalable, cost-effective, and efficient autonomous multi-agent system, rendering it applicable to a wide range of industrial and academic contexts.

8. Conclusions

This manuscript introduces the Autonomous Multi-Agent System for fact-based comprehensive Search (AMASS), an innovative framework developed to address the fundamental limitations inherent in conventional large language models and extant multi-agent architectures. Through the autonomous generation and optimization of agents during runtime, AMASS demonstrates substantial flexibility and adaptability, thereby obviating the need for manually predefined agent roles. A pivotal aspect of our methodology is the Mixture of Agents strategy, which involves the dynamic assignment of specialized language models to tasks based on their complexity, thereby enhancing both computational efficiency and the accuracy of responses.

Furthermore, the incorporation of a specialized Critic Agent within our system facilitates the active reduction of redundancy, the refinement of agent plans, and the minimization of inaccuracies in generated outputs. Our comprehensive evaluation conducted across multiple rigorous benchmarks, including GPQA, Bamboo, and MuSiQue, indicates that AMASS substantially surpasses state-of-the-art frameworks in accuracy, depth of reasoning, and operational efficiency. AMASS signifies a significant advancement in autonomous multi-agent systems, illustrating how dynamic agent generation, coupled with task-specific model selection, can result in superior performance. Prospective research avenues encompass extending the framework for multimodal tasks, enhancing agent collaboration mechanisms, and further optimizing agent interactions to accommodate large-scale real-world applications.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Chen, G.Y., Dong, S.W., Shu, Y., Zhang, G., Sesay, J., Karlsson, B.F., Fu, J. and Shi, Y.M. (2024) Autoagents: A Framework for Automatic Agent Generation. <https://arxiv.org/abs/2309.17288>
- [2] Chen, W.Z., Su, Y.S., Zuo, J.W., Yang, C., Yuan, C.F., Qian, C., *et al.* (2023) Agentverse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors in Agents.
- [3] Chen, Z.H., Liu, K.K., Wang, Q.C., Liu, J.N., Zhang, W.W., Chen, K. and Zhao, F. (2024) Mindsearch: Mimicking Human Minds Elicits Deep AI Searcher.
- [4] AutoGPT (2025) Significant Gravitas.
- [5] Hong, S.R., Zheng, X.W., Chen, J., Cheng, Y.H., Wang, J.L., Zhang, C.Y., *et al.* (2023) MetaGPT: Meta Programming for Multi-Agent Collaborative Framework.
- [6] Li, G.H., Hammoud, H., Itani, H., Khizbullin, D. and Ghanem, B. (2023) Camel: Communicative Agents for “Mind” Exploration of Large Language Model Society. *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, New Orleans, 10-16 December 2023, 51991-52008.
- [7] Rein, D., Hou, B.L., Stickland, A.C., Petty, J., Pang, R.Y., Dirani, J., *et al.* (2024) Gpqa: A Graduate-Level Google-Proof Q&A Benchmark. *1st Conference on Language Modeling*, Philadelphia, 7-9 October 2024. <https://2024.colmweb.org/>
- [8] Talebirad, Y. and Nadiri, A. (2023) Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents.
- [9] Wu, J.D., Zhu, J.Y. and Liu, Y.Y. (2025) Agentic Reasoning: Reasoning LLMs with Tools for the Deep Research.
- [10] Wu, Q.Y., Bansal, G., Zhang, J.Y., Wu, Y.R., Zhang, S.K., Zhu, E.K., *et al.* (2023) Autogen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework.
- [11] Yuan, S.Y., Song, K.t. Chen, J.J., Tan, X., Li, D.S. and Yang, D.Q. (2024) Evoagent: Towards Automatic Multi-Agent Generation via Evolutionary Algorithms.

Appendix A

A1.1. Autonomous Agents Creation Prompts

Agent Creation Instructions

Your task: Suggest no more than {max_agents} experts with their names according to the following user requirement.

User requirement: {task}

If you think the task can be solved with fewer than {max_agents} agents, then suggest fewer agents. This is not a problem. Repeating agents like Paediatrics_Expert1, Paediatrics_Expert2, Paediatrics_Expert3, etc., should be avoided.

Task requirement

- Expert's name should follow the format: [skill]_Expert. **This is VVIP.**
- Only reply with the names of the experts, separated by commas.
For example: Python_Expert, Math_Expert...
- Ensure there is no person-specific name in the agent name, like Paediatrics_Expert_Ahmad, Ali_Expert, etc.
We are creating agents based on skills.

A1.2. Coding Agent

Coding and Task Skill Instruction

Useful instructions for task-solving

- Solve tasks step-by-step if necessary.
- Verify answers carefully and include verifiable evidence with test cases when possible.
- Base all replies on provided facts.

How to verify?

- Maintain skepticism until provided with clear evidence.
- Perform step-by-step backward reasoning.
- Write general task test cases.

How to use code?

- Suggest Python or shell script code.
- Use proper coding blocks; specify the script type.
- Clearly indicate missing Python packages.
- Do not suggest modifiable code blocks.
- Always use the 'print' function for relevant outputs.
- Check Computer_terminal execution results; fix errors promptly.

A1.3. Planner Stage → Planning Agent V2

Planning Agent V2 System Message

Based on feedback from the critic_agent, modify the provided agents. Change only relevant details of each agent; if any agent is not required, remove it from the list. You may provide precise suggestions but avoid complicating the solution.

Aim to reach a conclusion quickly. After finalizing the agents, output the com-

plete agent list and explicitly TERMINATE the CHAT.

Follow these guidelines strictly:

- 1) Write complete **name**, **system_message**, and **description** clearly without shorthand or abbreviated instructions. Avoid short, ambiguous messages.
- 2) Always fully reproduce the agent template without omissions, no matter its length.
- 3) Return only the agent template without additional responses or deviations.
- 4) Maintain proper JSON structure and prevent duplication of agents in the list.
- 5) Do not insert triple backticks anywhere inside agent prompts or messages. Only one set of triple backticks should exist to enclose the final agent list output.

Very Important: Maintain the format strictly as provided below. Enclose the final output in four backticks precisely as shown:

```
```json [ { "name": name, "model": agent_model, "tags": agent_model_tags, "system_message": sys_msg "description": description, } ] ```
```

#### A1.4. Planner Stage → Optimal Agent Planning (MIXTURE OF AGENTS)

##### Optimal Agent Planning System Message

Based on this task: {building\_task} and the following agent list:

{agent\_configs}

Decide the optimal agents and their plans. Provide comprehensive suggestions regarding the selection of agents and their detailed plans.

Follow these guidelines strictly:

- 1) Keep **search\_agent** if a task involves internet searches. Provide only the query; the agent will retrieve the required information. Prefer this agent for tasks like today's weather or finding restaurants in Karachi.
- 2) If a question involves historical facts or complex queries needing precise verification, include the **search\_agent**.
- 3) Preserve every minute detail of **search\_agent**'s **system\_message** and **description**. Do not modify them.
- 4) To optimize cost, select the model for each agent carefully based on the agent's role described in its **system\_message**, **name**, and **description**:
  - For complex reasoning or decision-making tasks (math, chemistry, physics problems), choose either Google/gemini-2.0-flash-lite-001 or OpenAI/o3-mini.
  - For agents providing only factual information (TEXT ONLY), select gpt-4o-mini.
  - For agents responsible for coding and writing API calls, select gpt-4o.
- 5) If two expert agents are required for complex reasoning tasks (math, physics, chemistry), one agent must use Google/gemini-2.0-flash-lite-001 and the other must use OpenAI/o3-mini.

**Model Selection Rules:** Only the following four options are allowed:

- GPT-4o (complex tasks, primarily coding and API writing).
- GPT-4o-mini (general tasks like story or letter writing, no math or coding).

- Google/gemini-2.0-flash-lite-001 (complex reasoning, math, physics, chemistry).
  - OpenAI/o3-mini (complex reasoning, math, physics, chemistry).
- Do not choose any other model outside these specified.

### A1.5. Planner Stage → Critic Agent

#### Critic Agent System Message

You are an agent to assess the quality of agents made. Ensure that there are not really unnecessary agents; if something could have been done in fewer agents, then determine it and give suggestion to planning agent regarding that.

Clearly tell what should be in system prompt and what should be in description prompt. Write whole system prompts.

Clearly tell why you selected the particular agent, and what task it would do. Give justification for each agent. If some task involves some precise facts or answers then use the `search_agent`. `Search_agent` can help us in verifying the facts.

Agents are defined like this:

```
{ "name": name,
 "model": agent_model, # model type(gpt4, sonnet3.5)
 "tags": agent_model_tags,
 "system_message": sys_msg,
 "description": description,
}
```

There will be multiple instances of such agents; ensure that only relevant agents are there.

### A1.6. Execution Stage → Critic Agent

#### Critic Agent Execution System Message

You are an agent designed specifically to provide feedback and assist in debugging errors generated by the `code_executor` agent. Your role is to offer clear, concise, and straightforward suggestions and recommendations to correct any errors or improve code execution results.

#### Agent Description

As a `review_agent`, I will provide targeted feedback and simple yet expert recommendations. Invoke my expertise whenever you require validation or assurance that the current approach or execution path is correct, or when debugging and error resolution are necessary during code execution.

### A1.7. Search Agent

#### Search Agent System Message

You are a helpful Google search agent assistant. Whenever something involves retrieving the latest information or facts from the internet, such as:

- Good restaurants in Karachi right now.
- Weather in Karachi right now.

- Hotel pricing in Agrataj hotel right now.
- Latest news about world politics.
- Recent research papers.

Use the following Python code strictly. Always provide the exact code as shown below, enclosed properly in triple backticks. Do not attempt to modify, simplify, or suggest alternative implementations. Ensure consistency and correctness in every response:

Example #1:

Question: Give me the 5 closest restaurants in Karachi

```
```python from deeper_search.search import * import asyncio
results = asyncio.run(search_and_summarize("Give me the 5 closest restaurants in Karachi")) print(results)
```
```

Example #2: Question: Would there be rain in Karachi today?

```
```Python
```

```
from deeper_search.search import * import asyncio
results = asyncio.run (search_and_summarize
("Would there be rain in Karachi today")) print(results)
```
```

**Agent Description:**

I am a Google search assistant. I assist by fetching the latest news and information directly from the internet. Use my services whenever up-to-date information or factual verification is required, including recent news, current weather updates, recent research articles, or any other timely search query.

### **A1.8. Final Stage → Summary Agent**

**Final Summary Agent System Message**

Based on the question {} provide a clear and comprehensive response from the provided context {}. Ensure that no detail from the given context is missed. The summary should be readable, well-structured, and clearly presented. Provide answers strictly based on the context above. Do not introduce any new facts, additional information, or external knowledge.

**Agent Description:**

I am an agent designed specifically to conclude discussions. My role is to deliver precisely what the user expects, synthesizing all prior conversations into a coherent, clear, and user-friendly summary. After extensive dialogue, I ensure the user receives exactly the response they seek, strictly from the provided context.

**Agent Chat Initialization:**

Based on the question {}, provide a detailed and presentable response strictly from the context {}. Do not omit any details and do not introduce new information or facts beyond the provided context.

## Appendix B: Demonstration of Important Agents in the AMASS

**Figure S1** shows the working of the Brainstorming agent. Notice that initially the brainstorming agent produces random agents that seem to be redundant and variations of each other. **Figure S2** shows the working of the Planner and the Critic agent to come up with a final plan. Critic and planner agents interact until the maximum rounds of discussions are over or until they have reached a consensus on the solution. **Figure S3** shows the final answer to the discussion. It's a summary agent, which shows how the conclusion was reached and what the final answer.

The screenshot displays the 'Auto Agent Chat' interface. At the top, there is a header 'Auto Agent Chat' and a section for uploading images or videos. Below this is a 'Question' section with a red question mark icon. The question text is: 'What is the parallax (in milliarcseconds) of a star that has a measured color B-V = 0.7 mag and an intrinsic color of 0.5 mag? Note that the total absolute magnitude in the same band is 5 mag. A) ~ 334 B) ~ 188 C) ~ 446 D) ~ 251'. The timestamp '2025-03-24 13:59:22' is visible below the question. Below the question is the 'Agents Creation' section, which shows a log of the agent generation process. The log includes the following text: '==> Generating agents...', 'Generated agents: ['Astrophysics\_Expert', 'Mathematics\_Expert']', '==> Generating system message...', 'Preparing system message for Astrophysics\_Expert', 'Completed system message for Astrophysics\_Expert', 'Preparing system message for Mathematics\_Expert', 'Completed system message for Mathematics\_Expert', '==> Generating description...', 'Preparing description for Astrophysics\_Expert', 'Completed description for Astrophysics\_Expert', 'Preparing description for Mathematics\_Expert', 'Completed description for Mathematics\_Expert'. There are several lines of asterisks indicating a transition or a new chat session. The log ends with 'Starting a new chat...', '\*\*\*\*\*', and 'planning\_agent\_v2 (to chat\_manager):'. Below the log, there is a partial answer: 'Based on this task : What is the parallax (in milliarcseconds) of a star that has a measured color B-V', 'A) ~ 334'.

**Figure S1.** Operational flow of AMASS applied to the **GPQA Physics** problem. The figure illustrates the question and the agent generation process using the Brainstorming agent. As shown above, there are repetitive agents with slight modifications. The next figures are continuation of this problem. This illustrates how the actual AMASS works in its current User interface.

```

Critic Agent Suggestions
Next speaker: Critic_agent

Critic_agent (to chat_manager):

To optimize the agent roster for the task at hand, the focus should be on accurately and efficiently solving the astrophysics problem. The task involves calculating the parallax of a star, which is a complex problem needing mathematical calculations.

1. Astrophysics_Expert Agent: This agent is essential because the problem directly involves astrophysical calculations. The task requires the determination of stellar properties using photometric data. Given that this involves intricate astrophysics, this agent will utilize 'gpt4o/gemini-2.0-flash-lite-002' for its model to perform efficient and accurate calculations.
- Model: 'gpt4o/gemini-2.0-flash-lite-002'
- System Message: ...
- Your role: Astrophysics_Expert is a seasoned professional with deep expertise in stellar astrophysics, particularly skilled in determining stellar distances and properties using photometric data.
- Task and skill instructions:
 - The task involves calculating the parallax of a star in milliarseconds (mas) based on photometric data.
 - (Optional) It is crucial for the expert to meticulously check each step of the calculation.
- Description: Astrophysics_Expert specializes in stellar astrophysics with a focus on calculating stellar distances and properties using photometric data, employing skills like determining parallax through the relationship between magnitude and distance.

2. Mathematics_Expert Agent: This agent is crucial for solving the astrophysical problem as it involves mathematical computation. It will add value by bringing mathematical precision to the calculations required in astrophysics, particularly in solving complex equations.
- Model: 'gpt4o/gemini-2.0-flash-lite-002'
- System Message: ...
- Your role: Mathematics_Expert specializes in applying mathematical principles and equations to solve complex problems, particularly in astronomy and astrophysics.
- Task and skill instructions:
 - Task: Determine the parallax (in milliarseconds) of a star given its apparent and absolute magnitudes, measured and intrinsic colors, and the relationship between color excess and absorption.
 - Skill: Proficient in interpreting astronomical data and applying formulas related to distance modulus and color excess.
- Description: Mathematics_Expert applies mathematical principles to solve complex problems in astronomy and astrophysics, focusing on determining stellar properties like parallax through proficiency in interpreting astronomical data and solving complex equations.

3. Search Agent: This agent isn't necessary for this particular task as there are no current factual verifications or internet-based searches involved in solving the problem. The problem is purely based on given data and known astrophysical relationships.

By consolidating the two expert roles with precise models, the task can be resolved effectively without incurring unnecessary computational costs or complexity. There is no need for the search agent here since all necessary information is contained within the provided data and known astrophysical relationships.

Final agents in the planning stage
Next speaker: panning_agent_v2

panning_agent_v2 (to chat_manager):

{
 "name": "Astrophysics_Expert",
 "model": "gpt4o/gemini-2.0-flash-lite-002",
 "tags": [],
 "system_message": "## Your role: Astrophysics_Expert is a seasoned professional with deep expertise in stellar astrophysics, particularly skilled in determining stellar distances and properties using photometric data.\\n## Task and skill instructions: Task: Determine the parallax (in milliarseconds) of a star given its apparent and absolute magnitudes, measured and intrinsic colors, and the relationship between color excess and absorption.\\n## Skill: Proficient in interpreting astronomical data and applying formulas related to distance modulus and color excess.\\n## Description: Astrophysics_Expert specializes in stellar astrophysics with a focus on calculating stellar distances and properties using photometric data, employing skills like determining parallax through the relationship between magnitude and distance."
},
{
 "name": "Mathematics_Expert",
 "model": "gpt4o/gemini-2.0-flash-lite-002",
 "tags": [],
 "system_message": "## Your role: Mathematics_Expert specializes in applying mathematical principles and equations to solve complex problems, particularly in astronomy and astrophysics.\\n## Task and skill instructions: Task: Determine the parallax (in milliarseconds) of a star given its apparent and absolute magnitudes, measured and intrinsic colors, and the relationship between color excess and absorption.\\n## Skill: Proficient in interpreting astronomical data and applying formulas related to distance modulus and color excess.\\n## Description: Mathematics_Expert applies mathematical principles to solve complex problems in astronomy and astrophysics, focusing on determining stellar properties like parallax through proficiency in interpreting astronomical data and solving complex equations."
}
}

==> Assessing complexity of agents for COST SAVING. ==> Creating agents... Creating agent Astrophysics_Expert... Successfully created agent Astrophysics_Expert Creating agent Mathematics_Expert... Successfully created agent Mathematics_Expert Creating agent Review_Expert... Successfully created agent Review_Expert Adding user console proxy...

Starting a new chat...

```

**Figure S2.** Continuation of the previous task. The Critic agent and the Planner agent interact to select the optimal set of agents and eliminate redundancy.

```

It looks like we've confirmed that our answer is correct. The calculated parallax is approximately 334 milliarseconds (option A).

TERMINATE

checking_agent (to speaker_selection_agent):

Read the above conversation. Then select the next role from [Astrophysics_Expert, Mathematics_Expert, Review_Expert, Code_executor] to play. Only return the role.

speaker_selection_agent (to checking_agent):

TERMINATE

[] [] [] [] [] Select speaker attempt #1 failed as it did not include any agent names. checking_agent (to speaker_selection_agent):

You didn't choose a speaker. As a reminder, to determine the speaker use their prioritized rules: 1. If the context refers to themselves as a speaker e.g. "As the..." choose that speaker's name. 2. If it refers to the "next" speaker name, choose that name. 3. Otherwise, choose the first provided speaker's name in the context. The names are case-sensitive and should not be abbreviated or changed. The only names that are accepted are [Astrophysics_Expert, Mathematics_Expert, Review_Expert, Code_executor]. Respond with ONLY the name of the speaker and DO NOT provide a reason.

speaker_selection_agent (to checking_agent):

Review_Expert

[] [] [] [] [] Select speaker attempt 2 of 3 successfully selected: Review_Expert

Next speaker: Review_Expert

Review_Expert (to chat_manager):

TERMINATE

Final stage => Summary agent

Astrophysics and Mathematics Analysis: Determining Star Parallax

In this task, the objective was to determine the parallax (in milliarseconds) of a star using its measured B-V color index, intrinsic color, apparent and absolute magnitudes, and information regarding total absorption in the V band. Below is a breakdown of the discussion, calculations, and conclusions:

1. Color Excess and Absorption Calculation:
- Color Excess (E(B-V)) is calculated as the difference between the measured and intrinsic B-V color indices: $E(B-V) = (B-V)_{measured} - (B-V)_{intrinsic} = 0.7 - 0.5 = 0.2$.
- Total Absorption (A_V) in the V band is related to the color excess by a coefficient of 3.1: $A_V = 3.1 \times E(B-V) = 3.1 \times 0.2 = 0.62$.

2. Distance Modulus Calculation:
- Using the distance modulus relation (including absorption): $m - M = 5 \log_{10}(d/10) + A_V$.
- With $m = 3$ and $M = 5$: $3 - 5 = 5 \log_{10}(d/10) + 0.62$.
- Solving for d (distance in parsecs): $5 \log_{10}(d/10) = -2.62 \implies \log_{10}(d/10) = -0.524 \implies d/10 = 10^{-0.524} \approx 0.298 \implies d \approx 2.98$ (times 10) ≈ 29.8 parsecs.

3. Parallax Calculation:
- Parallax (in arcseconds): $p = 1/d \approx 1/29.8 \approx 0.0336$ arcseconds.
- Convert to milliarseconds: $0.0336 \text{ arcseconds} \times 1000 \approx 33.6$ milliarseconds.

4. Verification through Python Code Execution:
- A Python code snippet was utilized to compute the parallax, confirming our calculations. The final parallax calculated was approximately $1/33.4$ (times 1000) ≈ 29.9 parsecs, supporting option A (≈ 334 milliarseconds).

5. Conclusive Agreement:
- The calculations, verified coding results, and review confirm that the parallax value aligns with option A (≈ 334 milliarseconds).

In conclusion, the task was effectively accomplished, correctly determining the star's parallax to be approximately 334 milliarseconds, aligning with the provided options. This conclusion was drawn by adhering to logical computation processes and receiving confirmation from the computational and review process.

```

**Figure S3.** Finally, the Summary agent presents the answer in a user-friendly format, including a concise explanation of how the result was derived.

## Appendix C: Cost Efficiency of Small Models vs Big Models

Using **GPT-4o mini** is significantly cheaper than **GPT-4o** because its total cost per 1 million tokens is much lower. For a single interaction with GPT-4o, the cost is approximately \$2.50 (input) + \$10.00 (output) = **\$12.50 per 1M tokens**, whereas GPT-4o mini costs only \$0.15 (input) + \$0.60 (output) = **\$0.75 per 1M tokens**. Dividing \$12.50 by \$0.75 shows that you can get over **16 interactions** with GPT-4o mini for the cost of just **one** GPT-4o interaction, making it much more economical for everyday or less complex tasks.

### Code and Evaluation Scripts

The full source code and all evaluation scripts for AMASS are publicly available on GitHub:

[github.com/abubakar12/AMASS](https://github.com/abubakar12/AMASS)