This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Deciding the Path: Leveraging Multi-Agent Systems for Solving Complex Tasks

Iman Abbasnejad Xuefeng Liu Atunu Roy
Amazon Web Services
{imanaba, liuxuefe, atanuroy}@amazon.com

Abstract

We present a multi-agent framework that enhances the capabilities of LLMs through intelligent task distribution and resource optimization for solving complex problems. The framework uses a dynamic routing mechanism that automatically delegates queries to specialized agents, complemented by an efficient tool selection system that reduces computational complexity. This autonomous architecture eliminates the need for human intervention while maintaining high performance across diverse tasks. Through comprehensive empirical evaluation on three challenging benchmarks, our multi-agent framework outperforms existing state-of-the-art LLMs and even specialized systems. Our method achieves 90.29% *accuracy on Math 401 (surpassing MathViz-E's* 89.53%*),* 91.3% *pass@1 on MBPP (exceeding QualityFlow's* 88.53%*), and obtains state-of-the-art valid efficiency and execution scores of* 56.28% *and* 54.39% *respectively on BIRD SQL (outperforming DeepSeek-V3's* 51.28% *and* 51.02%*). These findings validate the effectiveness of our collaborative multi-agent approach in enhancing LLM capabilities, demonstrating significant improvements in accuracy, reasoning depth, and computational efficiency across varied task domains. Our research establishes a foundation for more adaptable and sophisticated AI systems capable of addressing complex real-world problems.*

1. Introduction

Large Language Models (LLMs) have revolutionized natural language processing with their unprecedented capabilities across numerous domains. Despite these achievements, they encounter significant limitations in task-specific scenarios, particularly regarding computational efficiency, resource optimization, accuracy in specialized tasks, and the necessity for extensive fine-tuning. While LLMs excel in general knowledge applications due to their comprehensive pre-training on vast datasets, their performance in specialized tasks remains suboptimal.

The remarkable human ability to leverage general knowledge and appropriate tools for novel tasks has inspired a new paradigm in artificial intelligence. Recent developments in AI agents have shown promising results in enhancing response quality through multi-tool integration. Notable work by [20] introduced a groundbreaking agent framework enabling autonomous collaboration between communicative agents with minimal human oversight. Furthermore, [30] advanced the field by developing sophisticated generative agents capable of simulating complex human behavior in interactive environments.

A critical challenge in AI systems lies in the sophisticated understanding required for effective tool utilization. This encompasses not only tool awareness but also mastery of tool interfaces, parameter specifications, usage patterns, and capability boundaries. The consequences of improper tool usage can be severe, leading to system failures or erroneous outputs. Thus, developing systems with comprehensive understanding of tool APIs and their response patterns is paramount.

Our research presents a transformative approach to enhancing LLM performance through an innovative multi-agent collaborative framework. We conduct an in-depth investigation into the capabilities and limitations of autonomous communicative agents, with particular emphasis on the dynamics of inter-agent collaboration. This understanding is fundamental to advancing artificial intelligence and developing robust, future-proof AI systems.

We introduce a pioneering framework that orchestrates multiple cooperative agents to tackle complex queries with unprecedented efficiency. At its core, our method implements a sophisticated role-playing mechanism featuring an intelligent router that autonomously orchestrates communicative agents toward optimal task completion. Our research makes several significant contributions to the field: (i) Development of an innovative routing system that autonomously orchestrates inter-agent communications, representing a significant advancement in multi-agent coordination, (ii) Implementation of a fully automated framework that eliminates human intervention while maintaining high performance, marking a substantial improvement over existing semi-automated systems, (iii) Novel integration of

multiple LLMs in a collaborative environment, demonstrating significant performance enhancements through sophisticated agent coordination, (iv) Introduction of an advanced tool selection mechanism that optimizes resource utilization while maximizing task performance through intelligent tool invocation.

Our research not only addresses fundamental challenges in current LLM applications but establishes a new paradigm for AI systems. This breakthrough approach enables more efficient, adaptive, and powerful AI systems capable of handling complex, real-world scenarios through sophisticated multi-agent collaboration, setting new standards for artificial intelligence capabilities.

2. Related Work

Recent advancements in Large Language Models (LLMs) have introduced significant developments in LLM-based agents and their applications. Multiple studies have demonstrated the potential of LLMs in iterative interactions and the complementary benefits of combining multiple LLMs ([36], [40], [11], [6]). This growing body of research has established a foundation for more sophisticated AI systems capable of complex problem-solving.

The introduction of Chain of Thought (CoT) prompting by [34] marked a paradigm shift in LLM capabilities. This was followed by the groundbreaking ReAct framework [37], which revolutionized the field by integrating reasoning and acting capabilities in LLMs for general task solving. ReAct's methodology of systematically generating reasoning traces from contextual information prior to action execution has established a fundamental framework for contemporary LLM-based problem-solving approaches.

Recent research has made substantial progress in resource optimization for LLM deployments. Notably, [7] introduced an adaptive model selection framework that optimizes the trade-off between computational costs and model performance, demonstrating up to 40% cost reduction while maintaining accuracy. Building on this, the MultiTool-CoT framework[16] presented a methodology for dynamic tool orchestration, enabling LLMs to systematically invoke external tools through structured reasoning phases. This framework implements a novel few-shot learning paradigm that guides tool selection and execution, resulting in significant improvements in both solution accuracy and computational efficiency and tool calls. These advances demonstrate that precisely controlled tool utilization, when aligned with specific reasoning stages, can substantially enhance both solution quality and resource utilization.

While the advances in tool orchestration and resource optimization are promising, the increasing complexity of tasks and available tools present new challenges in the AI landscape. Particularly,[24] identified significant hurdles in maintaining computational accuracy during sophisticated reasoning processes, especially when tasks require non-linear reasoning paths or complex interactions between multiple tools. Their analysis revealed that accuracy degradation can increase up to 30% in scenarios involving tool dependencies. This challenge is further complicated by the technical requirements of tool integration, as shown by[22] and [42], where demonstrated that successful tool utilization requires not only surface-level API knowledge but also deep understanding of tool behaviors, parameter specifications, and potential failure modes. Their studies showed that improper tool interface implementation can lead to cascading errors and reducing overall system reliability.

Recent research has extensively explored the augmentation of LLMs with external tools. [25] and [29] established foundational frameworks enabling LLMs to acquire tool utilization skills through demonstration-based learning. This advancement was further developed by [31], who introduced innovative methods for automatically identifying beneficial tools during the pre-training phase, significantly improving model adaptability. [32] and [26] developing sophisticated frameworks capable of orchestrating multiple AI models and tools in unified environments. These developments were complemented by domain-specific contributions from [28] and [23], who introduced specialized approaches for optimizing tool selection in particular application domains.

In additionally, recent work in multi-agent systems has explored novel directions. [8], conducted extensive investigations into emergent behaviors within multi-agent LLM systems, revealing complex patterns of interaction and cooperation previously unobserved. [39] introduced sophisticated graph-based methodologies that revolutionized agent coordination by representing inter-agent relationships and communication patterns in structured network formats. The field further evolved [35] in autonomous decision-making, where established new paradigms for agent independence and collaborative problem-solving in dynamic environments. [41] introduced modular approaches to complex task resolution, enabling agents to decompose and tackle challenging problems through systematic collaboration. [12] demonstrated that enabling multiple LLM instances to propose, critique, and refine responses through structured debate significantly improved mathematical and strategic reasoning capabilities. This debate-driven approach established new benchmarks for accuracy in complex problem-solving tasks

Our research extends these foundations by introducing a sophisticated `tool_selector` component within an autonomous process. This novel approach not only leverages the collective strengths of multi-agent systems but also addresses fundamental challenges in efficient tool selection and utilization. By implementing intelligent query routing to specialized agents and optimizing tool selection, our

method enhances the problem-solving capabilities of multi-agent systems while maintaining efficient resource management. This advancement represents a significant step forward in creating more capable and resource-efficient AI systems.

3. Method

This section introduces our novel multi-agent framework, designed to address the challenges of complex task solving using LLMs. Our proposed model architecture comprises three primary components: Router, Multi-agent, and Grader.

Router: Denoted as $\mathcal{M}_R$, functions as the central intelligence of our system, efficiently directing conversations to the most appropriate tools or agents. It leverages a `tool_selector` to access and manage k specialized tools or agents, represented as $\mathcal{M}1:k$, each being an LLM multi-agent model equipped with its own set of tools. Given an input query x from the user, $\mathcal{M}_R$ employs the `tool_selector` to analyze the query's requirements and the agents' capabilities, subsequently directing x to the most suitable agent $\mathcal{M}_i$ based on its specialized domain and functionalities. This dynamic routing mechanism ensures optimal utilization of the system's diverse capabilities, enhancing overall performance and efficiency in handling complex, multi-faceted tasks.

Multi-agent: Each Multi-agent $\mathcal{M}_i$ is designed for a specific task domain. It can be represented as a tuple: $\mathcal{M}_i = (L_i, R_i, S_i, T_i)$, where, L_i is the language model utilized by the agent, R_i is the role description, defining the agent's purpose and responsibilities, S_i represents the agent's state, carrying information about its tasks and interactions, T_i is the set of tools accessible to the agent, which may include functions or other agents. Specialized tools are designed to cater to specific domains with functionalities that are not available in foundation models. As a result, they are better suited to address the needs of domain-specific tasks.

Tool Selector: For each tool in the tool set T_i, the LLM is provided with a usage protocol $d_i \in \mathcal{D}$. This protocol includes: (i) a description outlining the tool's purpose, (ii) argument requirements for invocation, (iii) response schema specifying expected execution results. This meta-information enables the LLM to automatically learn tool usage, master input-output schemas, and understand data flow relations among various tools. Each tool is associated with a set of prompts and examples that the agent utilizes during the tool selection process.

Grader: Denoted as $\mathcal{M}_G$, is a language model responsible for evaluating the generated answer $\mathcal{A}$. It assesses whether the user query has been adequately addressed in $\mathcal{A}$, providing a binary score of 'yes' or 'no'.

3.1. Graph representation

To model the interactions within our autonomous collaborative agent system, we employ a graph-like structure. This representation captures the multi-round textual message exchanges between agents and tools. Figure 1 visualizes schematic representation of the proposed multi-agent graph.

Node: Each node in the graph represents an agent at time step t. Nodes process input queries from other agents or tools from the previous time step and generate responses accordingly.

Edges: They represent the connections between nodes in the graph, which can be conditional or direct. These edges facilitate the flow of communication messages between nodes, forming the interaction architecture of the agent collaboration system.

Messages Guide the information flow through the graph. At each time step t, a message is represented as a tuple: $m_t = (c_t, a_t, k_t)$, where, c_t is the message content, a_t is the action taken by the agent at time step t and k_t is a metadata dictionary containing information such as timestamp, sender, receiver, and additional data from tools and agents.

Graph structure The proposed graph integrates the aforementioned components to create a dynamic, interconnected system of agents and tools. This structure enables efficient routing of queries, collaborative problem-solving, and optimized tool selection. The process flow can be described as follows:

Initialization $(t = 0)$**:** The process begins with a central router node that interfaces between the user and a collection of specialized agents and tools. Upon receiving a user query x, the router performs three primary functions, (i) comprehend the user's query, (ii), assess the capabilities of available tools and agents, (iii) determine the most appropriate next node for query direction.

Tool Selection: Before routing, a tool selector component analyzes the query and available tools. It populates a `tool_selector` key in the state with a subset of tools most relevant to the current query. This selection process enhances efficiency by reducing the tool set to only the most appropriate ones for subsequent processing.

Query Routing $(t = 1)$**:** The router forwards the query, along with the selected tools, metadata, and the action $m_{t=1}$ to the chosen agent node in the next layer.

Agent Processing: The selected agent initiates a ReAct (Reasoning and Acting) process, utilizing only the tools specified in the `tool_selector` subset. This focused approach potentially improves efficiency and accuracy. The agent extracts crucial information from observations, tracks progress, transitions between action plans, handles exceptions, and adjusts strategies as necessary.

Answer Generation: The agent generates an answer $\mathcal{A}$, which, combined with the original user query x, forms the

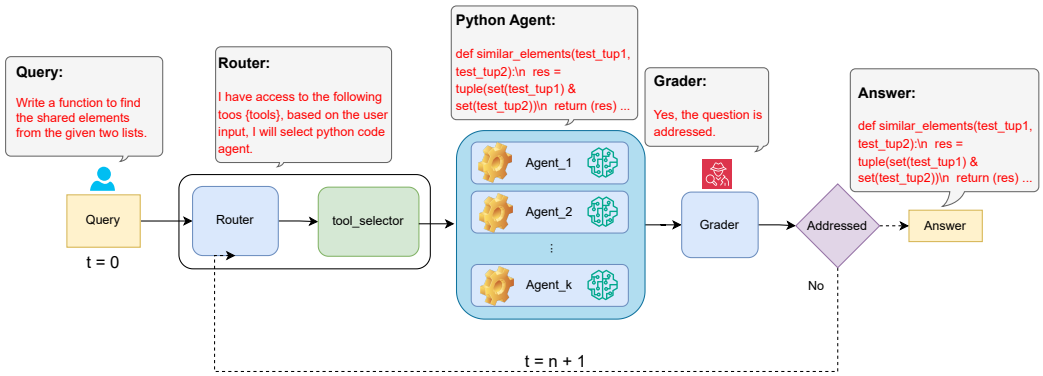


Figure 1. Schematic representation of our proposed system. The diagram illustrates the flow from user query input through the Router, `tool_selector`, Agents, and Grader, showing the system's dynamic routing and iterative problem-solving capabilities.

next message $m_{t=n}$.

Grading $(t = n + 1)$**:** The message $m_{t=n}$ is forwarded to a "grader" node. The grader employs chain-of-thought (CoT) prompting, decomposing the problem into multiple reasoning steps for thorough analysis. It outputs a binary score: "yes" if $\mathcal{A}_n$ successfully addresses x, or "no" if the question remains inadequately answered.

Process Conclusion or Iteration: If the grader outputs "yes", the system concludes and outputs $\mathcal{A}$ as the final response. If "no", the system may loop back to the `tool_selector` to refine tool selection based on the current state and understanding of the problem, before invoking additional agents or tools to continue the problem-solving process.

4. Experimental Setup

We demonstrate the efficiency of our proposed system in autonomous task execution. Specifically, we present experimental results across diverse text-based scenarios to evaluate the system's performance on various symbolic tasks. Our observations indicate that our proposed solution can effectively function as symbolic reasoners in different contexts.

4.1. Dataset:

To rigorously assess the proposed system, we conducted experiments on a range of complex problems, encompassing, arithmetic operations, Python programming challenges and Text-to-SQL conversion tasks.

To ensure a comprehensive evaluation of our method, we randomly interspersed the three task categories, requiring the agent to autonomously assign each task to the appropriate module. This approach allows us to assess not only the system's ability to solve individual tasks but also its capacity for task classification and routing. Our experimental design aims to test the system's versatility and robustness across different symbolic reasoning domains. By incorporating this diverse set of problems, we seek to validate the system's potential for autonomous operation in varied computational contexts.

Math 401: [38] This dataset is a diverse collection of mathematical operations and their results, ranging from basic arithmetic to more complex expressions. It includes various types of calculations such as addition, subtraction, multiplication, division, and exponentiation, covering integers, decimals, and negative numbers. Each entry consists of a "query" representing the mathematical expression and a "response" indicating the correct answer.

BIRD SQL: [21] Represents a significant advancement in text-to-SQL benchmarks, bridging the gap between academic research and real-world applications. This comprehensive dataset comprises $12,751$ text-to-SQL pairs across 95 databases from 37 professional domains. Unlike previous benchmarks, BIRD-SQL emphasizes the importance of database values, introducing unique challenges such as handling external knowledge, processing dirty data, and optimizing SQL efficiency in large databases. Its design requires models to not only perform semantic parsing but also comprehend and effectively utilize database contents. BIRD-SQL is a standard for evaluating text-to-SQL models.

Mostly Basic Python Problems (MBPP): [4] is a comprehensive benchmark consisting of approximately 1000 crowd-sourced Python programming problems, designed to evaluate code generation and program synthesis capabilities. Tailored for entry-level programmers, these problems cover a wide range of fundamental programming concepts and standard library functionalities. Each problem in the dataset is structured with three key components: a clear natural language task description, a reference Python code solution, and three automated test cases for objective evaluation. The MBPP's comprehensive coverage of various

Python programming aspects makes it an ideal tool for measuring AI models' proficiency in understanding and generating Python code across different difficulty levels and programming concepts.

4.2. Evaluation Metrics

As was discussed in the previous section, we evaluate our system across diverse tasks. This comprehensive evaluation approach allows us to assess the versatility and robustness of our proposed system in handling a wide range of computational challenges. By testing the system on multiple task types, we aim to demonstrate its ability to adapt to different problem domains and showcase its potential for real-world applications. We report the performance of our method on Math 401 dataset using Accuracy metric (Acc) , Relative Error (RE) and Non-Number Ratio (NNR) if the generated content does not contain any numbers. When evaluating on the BIRD SQL dataset, we adopt Execution Accuracy (EA) and Valid Efficiency Score (VES) metrics defined in [21] which capture both the correctness and efficiency of generated SQL queries. For the MBPP dataset, we use the $Pass@k$ metric, which evaluates the model's ability to generate correct Python code solutions on the k-th attempts. We assess the agent's environmental interaction capabilities using the "Repeat Action Ratio" (RAR) and its task completion awareness through the "Aware of Completion Ratio" (ACR) which measures the proportion of instances where the agent accurately recognizes task completion and initiates a finish action.

4.3. Model Setting

Our experimental framework employs two LLMs, Claude 3 Haiku [3] and Claude 3 Sonnet [3]. To optimize the balance between deterministic output and creative generation, we set the temperature parameter to 0.3 across all experiments. This configuration introduces a controlled level of stochastically to the generation process, enabling the models to explore diverse solution paths while maintaining coherence and relevance. The temperature setting also facilitates multiple generation attempts when required, allowing the system to refine its outputs iteratively. This approach ensures a robust evaluation of our multi-agent framework's performance across various complex problem-solving scenarios.

4.4. Experimental Results and Analysis

We evaluate our proposed method across three distinct tasks: mathematical problem-solving, text-to-SQL generation, and code generation over the current state-of-the-art models including, GPT-4 [1], GPT3.5-Turbo [27], GPT-o3-mini, LLaMA3 [2], Mistral [18], Mixtral [19], DeepkSeek-v2.4 [9], DeepkSeek-V3 [10], Claude-3.5 Haiki and Sonnet [3] models. We also compare our solution against the current task specific agents such as MathViz-E [5], Autogen [35], MetaGPT [14], QualityFlow [15], XiYan-SQL [13] and CHESS-SQL [33].

Mathematical Problem-Solving: Table 1 presents the comparative analysis on the Math 401 dataset. Our agent-enhanced Claude-3.7-Sonnet w/a achieves state-of-the-art performance with 90.29% accuracy, surpassing specialized mathematical reasoning systems like MathViz-E (89.53%), Autogen (84.83%), and MetaGPT (81.83%), as well as DeepSeek V3 (84.28%) and GPT-4 (83.54%). This improvement is particularly noteworthy given DeepSeek V3's strong baseline performance and GPT-4's capabilities in mathematical reasoning. The integration of our agent framework yields substantial improvements in precision metrics, with both Claude-3.7 variants achieving minimal Relative Error (RE=0.02) and Near-Number Rate (NNR=0.01), matching or exceeding even specialized mathematical systems.

Examining model performance across different architectures illustrates the significant impact of our agent framework. The progression from base Claude models (Claude-3.5-Sonnet: 64.32% accuracy, RE=0.31, NNR=1.04) to their agent-enhanced variants (Claude-3.7-Sonnet w/a: 90.29% accuracy, RE=0.02, NNR=0.01) demonstrates substantial improvements that exceed gains achievable through model scaling alone. While large-scale models like LLaMA 3.1 405B show competitive performance (77.42% accuracy), they still fall significantly behind our agentic approach. Even newer models like GPT-o3-mini (77.34% accuracy, RE=0.11, NNR=0.23) cannot match our agent-enhanced framework's capabilities in mathematical reasoning and precision.

Code Generation: Table 2 presents the comparative analysis on the MBPP dataset. Our agent-enhanced models demonstrate superior performance across all metrics. Claude-3.7-Sonnet w/a achieves state-of-the-art performance with 91.3% pass@1 accuracy, significantly outperforming both specialized code generation systems like QualityFlow (88.53%) and MapCoder (87.27%), as well as GPT-4 (84.1%) and DeepSeek V3 (82.91%).

The integration of our agent framework yields substantial improvements in code generation capabilities, creating a clear performance gap between our agent-enhanced Claude models and their base versions. While the standard Claude-3.5-Sonnet achieves 73.3% pass@1, its agent-enhanced variant reaches to 91.3%. This dramatic improvement margin exceeds that observed in other model families and specialized systems, highlighting our agent framework's unique ability to enhance structured reasoning and code generation tasks. The consistent performance advantage holds even when compared to newer models like GPT-o3-mini (82.1%), demonstrating that our approach provides benefits beyond what's achievable through model scaling

Model	Acc ↑	RE ↓	NNR ↓
GPT-4	83.54	0.07	0.00
GPT-o3-mini	77.34	0.11	0.23
GPT-3.5-Turbo	75.06	0.14	0.40
LLaMa 3.1 405B	77.42	0.14	0.28
LLaMa 3 65B	28.43	1.61	4.74
LLaMa 3 8B	22.71	0.23	0.42
Mistral-2402	32.62	1.13	4.02
Mixtral-8x7B	41.23	1.26	3.22
Claude-3.5-Haiku	65.31	0.18	0.57
Claude-3.5-Sonnet	64.32	0.31	1.04
Claude-3.7-Sonnet	71.38	0.29	0.8
DeepSeek V3	84.28	0.03	0.01
DeepSeek V2.5	66.34	0.12	0.98
MathViz-E [5]	89.53	0.03	0.01
Autogen [35]	84.83	0.06	0.02
MetaGPT [14]	81.83	0.08	0.03
Claude-3.5-Haiku w/a	85.17	0.04	0.01
Claude-3.5-Sonnet w/a	86.12	0.03	0.01
Claude-3.7-Sonnet w/a	**90.29**	**0.02**	**0.01**

Table 1. Model performance comparison on Math 401 dataset.

Model	pass@1 ↑
GPT-4	84.1
GPT-o3-mini	82.1
GPT-3.5-Turbo	76.2
LLaMA 3.1 405B	73.6
LLaMA 3 65B	62.7
LLaMA 3 8B	39.3
Mistral-2402	39.1
Mixtral-8x7B	38.2
Claude-3.5-Haiku	69.2
Claude-3.5-Sonnet	73.3
Claude-3.7-Sonnet	79.2
DeepSeek V3	82.91
DeepSeek V2.5	72.06
QualityFlow [15]	88.53
MapCoder [17]	87.27
Autogen [35]	87.73
MetaGPT [14]	83.18
Claude-3.5-Haiku w/a	82.3
Claude-3.5-Sonnet w/a	89.2
Claude-3.7-Sonnet w/a	**91.3**

Table 2. Model Performance Comparison on MBPP dataset.

alone.

Text-to-SQL: Tables 3 and 4 demonstrate our method's effectiveness in semantic parsing and database query generation. The comprehensive evaluation reveals significant performance variations across models, with our Claude-3.7-Sonnet w/a achieving state-of-the-art results. With knowl-

Model	Development Data	
	wok	wk
GPT-4	34.60	49.77
GPT-o3-mini	31.37	47.28
GPT-3.5-Turbo	27.97	43.81
LLaMA 3.1 405B	33.27	44.63
LLaMA 3 65B	25.28	35.02
LLaMA 3 8B	19.91	23.89
Mistral-2402	22.45	30.21
Mixtral-8x7B	20.36	23.28
Claude-3.5-Haiku	32.75	45.28
Claude-3.5-Sonnet	34.75	47.28
DeepSeek V3	35.31	51.28
DeepSeek V2.5	22.12	30.07
Autogen [35]	31.18	40.17
MetaGPT [14]	31.39	41.92
XiYan-SQL [13]	36.28	43.90
CHESS-SQL [33]	34.12	47.38
Claude-3.5-Haiku w/a	34.39	51.80
Claude-3.5-Sonnet w/a	35.86	53.45
Claude-3.7-Sonnet w/a	**37.61**	**56.28**

Table 3. The Valid Efficiency Score (VES) of advanced text-to-SQL models in BIRD (wok and wo represent without knowledge and with knowledge respectively).

Model	Development Data	
	wok	wk
GPT-4	24.05	37.22
GPT-o3-mini	25.02	48.42
GPT-3.5-Turbo	27.29	42.70
LLaMA 3.1 405B	24.89	41.82
LLaMA 3 65B	19.91	27.42
LLaMA 3 8B	16.02	20.72
Mistral-2402	36.92	45.02
Mixtral-8x7B	33.62	39.32
Claude-3.5-Haiku	30.90	46.35
Claude-3.5-Sonnet	32.12	47.83
DeepSeek V3	38.29	51.02
DeepSeek V2.5	29.02	36.02
Autogen [35]	30.62	38.20
MetaGPT [14]	30.92	40.87
XiYan-SQL [13]	34.17	46.04
CHESS-SQL [33]	33.62	44.81
Claude-3.5-Haiku w/a	36.62	45.92
Claude-3.5-Sonnet w/a	38.39	52.13
Claude-3.7-Sonnet w/a	**41.15**	**54.39**

Table 4. The Execution Accuracy (EX) of advanced text-to-SQL models in BIRD (wok and wo represent without knowledge and with knowledge respectively).

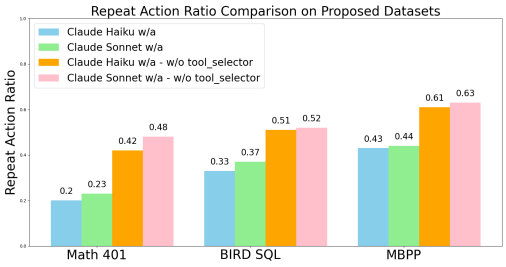


Figure 2. (a) Repeat Action Ratio ↓

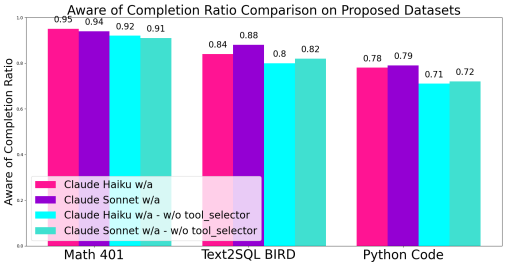


Figure 3. (b) Aware of Completion Ratio ↑

Figure 4. Quantitative analysis of agent performance metrics: (a) Repeat Action Ratio (RAR) showing minimal redundant operations across different problem domains, and (b) Aware of Completion Ratio (ACR) demonstrating high task completion awareness. These metrics indicate superior efficiency in problem-solving strategy selection and execution compared to baseline approaches.

edge incorporation, our agent-enhanced model reaches a VES of 56.28% and EX of 54.39%, substantially outperforming proprietary models like GPT-4 (VES: 49.77%, EX: 37.22%), GPT-o3-mini (VES: 47.28%, EX: 48.42%), DeepSeek V3 (VES: 51.28%, EX: 51.02%), and agentic SQL systems including XiYan-SQL (VES: 43.90%, EX: 46.04%) and CHESS-SQL (VES: 47.38%, EX: 44.81%).

The knowledge integration impact is particularly pronounced in our agent-enhanced architecture, with incremental improvements visible across model versions. Claude-3.5-Sonnet w/a demonstrates substantial gains when knowledge is incorporated (VES: from 35.86% to 53.45%, EX: from 38.39% to 52.13%), while our most advanced Claude-3.7-Sonnet w/a pushes performance even further (VES: 56.28%, EX: 54.39%). This consistent improvement pattern suggests our agent framework uniquely leverages external knowledge for query generation, outperforming even specialized SQL systems and establishing a new benchmark for text-to-SQL performance.

4.5. Efficiency Analysis

As are shown in Figures 2 and 3, using `tool_selector` significantly reduces action repetition (RAR). This efficiency gain is particularly important when compared to traditional approaches like ReAct and Chain of Thoughts integrated with models like GPT-4 and DeepSeek V3, which typically require multiple attempts and long tokens to solve complex problems.

The task completion awareness (ACR) metrics demonstrate our agentic models achieve high awareness rates (0.78-0.95 ACR), with `tool_selector` integration further improving completion recognition. This high awareness rate translates to more efficient resource utilization and better task management compared to traditional approaches. The consistent performance across different problem domains demonstrates the robustness of our framework in handling diverse computational challenges while maintaining operational efficiency.

These comprehensive results validate our framework's effectiveness across multiple domains, showing consistent improvements over both traditional and state-of-the-art models like DeepSeek V3 and GPT-4. The framework's ability to enhance performance while maintaining computational efficiency positions it as a significant advancement in AI system design.

4.6. Discussion

Our experimental results demonstrate the effectiveness of our autonomous multi-agent framework across diverse computational tasks. The framework's dynamic routing mechanism and `tool_selection` system achieved state-of-the-art performance metrics, with our latest Claude-3.7-Sonnet w/a implementation reaching 90.29% accuracy in mathematical problem-solving, 91.3% pass@1 rate in code generation, and 56.28% VES and 54.39% EX scores in text-to-SQL generation. These results surpass not only powerful general models like DeepSeek V3 and GPT-4, but also specialized systems like MathViz-E (89.53%), QualityFlow (88.53%), and traditional agent frameworks. The implementation of our `tool_selector` significantly reduced computational overhead while maintaining exceptional precision (RE=0.02, NNR=0.01 on mathematical tasks).

The comprehensive performance improvements across Math 401, BIRD SQL, and MBPP benchmarks validate our framework's key innovations in autonomous orchestration and resource optimization. For example, in text-to-SQL generation, our framework demonstrates consistent improvement from Claude-3.5-Haiku w/a (51.80% VES) to Claude-3.5-Sonnet w/a (53.45% VES) to Claude-3.7-Sonnet w/a (56.28% VES), outperforming even specialized SQL systems like XiYan-SQL (43.90%) and CHESS-SQL (47.38%). These results conclusively demonstrate that sophisticated agent coordination can dramatically enhance LLM capabilities across diverse domains while maintaining operational efficiency.

5. Limitations and Risks

The integration of multiple autonomous agents in AI systems has it's own pros and cons. While it enhances problem-solving capabilities across diverse tasks, as evidenced by our experiments, it also introduces new complexities. This section delves into the limitations inherent in multi-agent systems and the associated risks that need careful consideration.

5.1. Limitations

The integration of multiple agents enhances the system's ability to handle diverse and complex tasks, as evidenced by the performance improvements of Claude-Haiku [3] w/a and Claude-Sonnet [3] w/a across various datasets (Tables 1, 2, 3 and 4). However, this increased capability comes at the cost of heightened system sensitivity and operational complexity. Our findings indicate that as the number of agents and complexity of the tasks increase, the frequency of recall actions required to successfully complete tasks will rise. This suggests that while agents provide more diverse capabilities, it also necessitates more sophisticated coordination and decision-making processes. The implementation of a `tool_selector` has proven instrumental in mitigating some of these challenges by optimizing resource allocation.

5.2. Risks and Ethics Statement

While our proposed method of integrating multiple agents to handle complex tasks aims to overcome limitations of single LLM-based solutions, it is not without ethical considerations. The primary concern lies in ensuring that these agent systems do not violate human values or ethical norms during their operation and potential self-evolution. It is crucial to implement robust supervision and regulatory mechanisms to guide the development and deployment of multi-agent systems.

6. Conclusion and Future work

Our research introduces an innovative multi-agent framework that significantly advances the capabilities of Large Language Models through sophisticated agent coordination and resource optimization. The framework's effectiveness is demonstrated through comprehensive empirical evaluation across three challenging benchmarks—Math 401, BIRD SQL, and MBPP—where it consistently outperformed both state-of-the-art models and specialized systems. Key innovations include our dynamic routing mechanism for autonomous inter-agent communications and an advanced tool selection system that optimizes resource utilization. The framework's success is particularly evident in its achievement of 90.29% accuracy in mathematical problem-solving, 91.3% pass@1 in code generation, and 56.28% VES in text-to-SQL generation, all while maintaining high operational efficiency. These results establish a new benchmark for multi-agent LLM systems and provide a foundation for more adaptable AI solutions to complex real-world problems.

Future work can explore on adapting the agentic framework to more dynamic and complex environments, potentially incorporating multi-modal reasoning capabilities and developing more sophisticated knowledge integration mechanisms. Of particular interest is the development of enhanced memory architectures that could enable more efficient knowledge sharing and utilization across diverse computational tasks. While our current results establish a strong foundation for autonomous multi-agent systems, continued research in these directions will be crucial for addressing increasingly complex real-world challenges.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] AI@Meta. Llama 3 model card. 2024.

[3] Anthropic. Anthropic. *The Claude 3 Model Family: Opus, Sonnet, Haiku.*, 2024.

[4] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[5] Arya Bulusu, Brandon Man, Ashish Jagmohan, Aditya Vempaty, Jennifer Mari-Wyka, and Deepak Akkil. Mathviz-e: A case-study in domain-specialized tool-using agents. *arXiv preprint arXiv:2407.17544*, 2024.

[6] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

[7] Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

[8] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*, 2024.

[9] DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024.

[10] DeepSeek-AI. Deepseek-v3. *https://github.com/deepseek-ai/DeepSeek-V3/blob/main/DeepSeek$_V$3.pdf*, 2024.

[11] David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A Saurous, Jascha Sohl-Dickstein,

et al. Language model cascades. *arXiv preprint arXiv:2207.10342*, 2022.

[12] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

[13] Yingqi Gao, Yifu Liu, Xiaoxia Li, Xiaorong Shi, Yin Zhu, Yiming Wang, Shiqi Li, Wei Li, Yuntao Hong, Zhiling Luo, et al. Xiyan-sql: A multi-generator ensemble framework for text-to-sql. *arXiv preprint arXiv:2411.08599*, 2024.

[14] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

[15] Yaojie Hu, Qiang Zhou, Qihong Chen, Xiaopeng Li, Linbo Liu, Dejiao Zhang, Amit Kachroo, Talha Oz, and Omer Tripp. Qualityflow: An agentic workflow for program synthesis controlled by llm quality checks. *arXiv preprint arXiv:2501.17167*, 2025.

[16] Tatsuro Inaba, Hirokazu Kiyomaru, Fei Cheng, and Sadao Kurohashi. Multitool-cot: Gpt-3 can use multiple external tools with chain of thought prompting. *arXiv preprint arXiv:2305.16896*, 2023.

[17] Md Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. Mapcoder: Multi-agent code generation for competitive problem solving. *arXiv preprint arXiv:2405.11403*, 2024.

[18] Sablayrolles Alexandre Mensch Arthur Bamford Chris Chaplot Devendra Singh de las Casas Diego Bressand Florian Lengyel Gianna Lample Guillaume Saulnier Lucile Lavaud Lelio Renard Lachaux ´ Marie-Anne Stock Pierre Scao Teven Le Lavril Thibaut Wang Thomas Lacroix Timothee Jiang, Albert Q. and ´ William El. . Sayed. Mistral. *https://mistral.ai/news/announcing-mistral-7b/*, 2023.

[19] Sablayrolles Alexandre Mensch Arthur Bamford Chris Chaplot Devendra Singh de las Casas Diego Bressand Florian Lengyel Gianna Lample Guillaume Saulnier Lucile Lavaud Lelio Renard Lachaux ´ Marie-Anne Stock Pierre Scao Teven Le Lavril Thibaut Wang Thomas Lacroix Timothee Jiang, Albert Q. and ´ William El. . Sayed. Mixtral. *https://mistral.ai/news/mixtral-of-experts/*, 2023.

[20] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36: 51991–52008, 2023.

[21] Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms. *arXiv preprint arXiv:2304.08244*, 2023.

[23] Yaobo Liang, Chenfei Wu, Ting Song, Wenshan Wu, Yan Xia, Yu Liu, Yang Ou, Shuai Lu, Lei Ji, Shaoguang Mao, et al. Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis. *Intelligent Computing*, 3:0063, 2024.

[24] Yuan Ling, Fanyou Wu, Shujing Dong, Yarong Feng, George Karypis, and Chandan K Reddy. International workshop on multimodal learning-2023 theme: Multimodal learning with foundation models. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5868–5869, 2023.

[25] Ruibo Liu, Jason Wei, Shixiang Shane Gu, Te-Yen Wu, Soroush Vosoughi, Claire Cui, Denny Zhou, and Andrew M Dai. Mind's eye: Grounded language model reasoning through simulation. *arXiv preprint arXiv:2210.05359*, 2022.

[26] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[27] OpenAI. chatgpt: Optimizing language models for dialogue. *https://openai.com/blog/chatgpt.*, 2022.

[28] Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*, 2023.

[29] Aaron Parisi, Yao Zhao, and Noah Fiedel. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255*, 2022.

[30] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[31] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.

[32] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024.

[33] Shayan Talaei, Mohammadreza Pourreza, Yu-Chen Chang, Azalia Mirhoseini, and Amin Saberi. Chess: Contextual harnessing for efficient sql synthesis. *arXiv preprint arXiv:2405.16755*, 2024.

[34] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[35] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

[36] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. Ai chains: Transparent and controllable human-ai interaction

by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–22, 2022.

[37] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[38] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. How well do large language models perform in arithmetic tasks?, 2023.

[39] Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, and Dawei Cheng. G-designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*, 2024.

[40] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

[41] Li Zhou, Xinfeng Deng, Zhe Wang, Xiaoying Zhang, Yanjie Dong, Xiping Hu, Zhaolong Ning, and Jibo Wei. Semantic information extraction and multi-agent communication optimization based on generative pre-trained transformer. *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[42] Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36:50117–50143, 2023.