Synergizing Logical Reasoning, Knowledge Management and Collaboration in Multi-Agent LLM System

Adam Kostka, Jaroslaw Chudziak

Synergizing Logical Reasoning, Knowledge Management and Collaboration in Multi-Agent LLM System

Adam Kostka and **Jarosław A. Chudziak**

Faculty of Electronics and Information Technology
Warsaw University of Technology, Poland
adam.kostka.stud@pw.edu.pl and jaroslaw.chudziak@pw.edu.pl

Abstract

This paper explores the integration of advanced Multi-Agent Systems (MAS) techniques to develop a team of agents with enhanced logical reasoning, long-term knowledge retention, and Theory of Mind (ToM) capabilities. By uniting these core components with optimized communication protocols, we create a novel framework called SynergyMAS, which fosters collaborative teamwork and superior problem-solving skills. The system's effectiveness is demonstrated through a product development team case study, where our approach significantly enhances performance and adaptability. These findings highlight SynergyMAS's potential to tackle complex, real-world challenges.

1 Introduction

Large Language Models (LLMs) have seen significant advancements in recent years, particularly in answer accuracy, increased context windows, and the ability to tackle complex tasks. However, despite these improvements, LLMs continue to face challenges such as hallucinations, knowledge gaps due to incomplete training data, and difficulties in managing long-term dependencies (Naveed et al., 2024). Enhancing LLM performance remains complex, often requiring enormous resources for training and domain-specific data integration, which may not always yield the desired outcomes (Raschka, 2024).

A promising approach to overcoming these limitations is the development of Multi-Agent Systems (MAS) that incorporate LLMs at their core. MAS can be tailored for specific use cases, allowing more sophisticated solutions by refining communication protocols and integrating external tools and databases.

This study aims to broaden the capabilities of LLMs by creating a MAS framework that integrates three critical components: logical reasoning, Retrieval-Augmented Generation (RAG), and
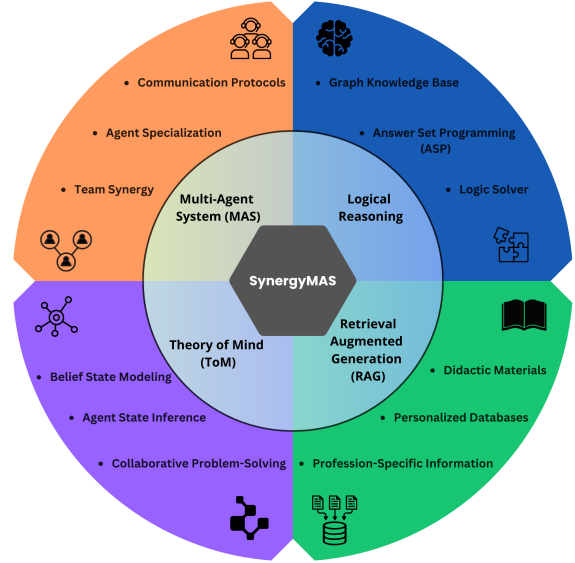

Figure 1: SynergyMAS: Integrating Logical Reasoning, RAG, and Theory of Mind in a Multi-Agent System to enhance LLM capabilities for complex tasks (based on Sun et al. (2024)).

Theory of Mind (ToM). By combining these elements, we propose a new framework named SynergyMAS, which enhances LLM capabilities for complex tasks. The framework was tested on multiple LLMs, including Claude and Gemini, to evaluate its versatility and effectiveness.

LLMs often struggle with complex reasoning tasks that require logical thinking, frequently leading to hallucinations (Chen et al., 2023). Previous research has shown that incorporating logical reasoning, such as a logical solver, can mitigate these issues on logic testing datasets (Pan et al., 2023). In this study, we implement a graph knowledge base integrated with a logic solver using Answer Set Programming (ASP), enhancing the system's ability to engage in extended, logical discussions.

Moreover, MAS members will have access to a personalized RAG database with didactic materials

specific to their profession. This approach aims to improve specialists' performance by providing them with relevant information without overburdening others.

As LLMs evolve, their ToM capabilities also improve (Kosinski, 2024). Given the importance of ToM in collaborative problem-solving, enhancing these capabilities should boost system effectiveness. Including a belief state in agents' communications allows for better inference of each other's states, promoting better teamwork.

This paper contributes to Artificial Intelligence, Machine Learning, and Dialogue Systems by:

1. Proposing a novel architecture for multi-agent LLM systems, integrating logical solvers with graph databases and new communication protocols that enhance agent synergy.

2. Demonstrating the effectiveness of combining logical reasoning, RAG, and ToM in improving agent capabilities for complex conversations requiring external knowledge and domain-specific expertise.

3. Offering insights into practical applications of such systems in collaborative problem-solving scenarios, identifying real-world contexts where SynergyMAS can significantly enhance system performance.

The main components of SynergyMAS are illustrated in Figure 1.

2 Related Work

SynergyMAS integrates several ideas discussed in various studies, each playing a critical role in enhancing MAS performance. This section explores the current research on these components.

2.1 Background on Multi-Agent LLM Systems

The topic of MAS has been ongoing for many years, with significant contributions documented (Wooldridge, 2009). Recently, MAS has gained popularity in AI due to its superior ability to solve complex problems through the collaborative efforts of autonomous agents (Lei, 2024). MAS architectures typically involve a network of intelligent agents with specialized knowledge working towards common objectives (Russell and Norvig, 2016). A hierarchical structure, as showcased in Figure 2, is one such structure utilized in this article.
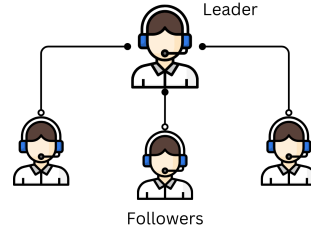

Figure 2: Hierarchical team structure.

Recent improvements in LLMs continue to enhance the potential of MAS, allowing for more refined communication and reasoning among agents (Wu et al., 2023). However, as these systems grow in complexity, they face coordination, scalability, and consistency challenges, driving ongoing research to improve MAS functionality.

2.2 Overview of Logical Reasoning in AI

The complex and opaque nature of LLMs makes it challenging to directly analyze and improve their behavior and outputs (Calegari et al., 2020). This has made it difficult for LLMs to effectively handle convoluted logical reasoning tasks (Chen et al., 2024a). Incorporating logical solver tools into MAS can address this limitation (Chen et al., 2024b). Many studies have attempted to integrate logical thinking into models using single or multiple methods (Yang et al., 2023). The typical approach involves converting natural language into a logical format (Gelfond and Kahl, 2014), followed by applying a logic solver to resolve the query. This approach has shown promising results, improving performance on logic testing datasets like ProofWriter (Tafjord et al., 2021). However, the use of these solvers to enhance reasoning capabilities in multi-agent systems still requires further exploration.

2.3 Graph Databases in AI Systems

Graph databases have gained prominence in AI for their ability to efficiently manage and query complex relationships between data points. Unlike traditional databases, graph databases use nodes and edges to represent and traverse connections, making them well-suited for interconnected data applications (Lane et al., 2019). Graph databases like neo4j are increasingly used in AI to enhance reasoning and decision-making by providing structured, context-aware data retrieval (Besta et al., 2023). Their integration into AI systems supports advanced tasks such as knowledge management

(Liang et al., 2024) and logical inference, essential for tackling complex problems.

2.4 RAG (Retrieval-Augmented Generation)

RAG has emerged as a critical enhancement in the LLM landscape, addressing challenges related to context expansion and the inclusion of real-world knowledge in responses while reducing hallucinations (Lewis et al., 2021). This approach allows models to analyze vast amounts of information beyond their training data, leading to the creation of specialized agents with domain-specific expertise. Recent innovations like Corrective RAG (CRAG) (Yan et al., 2024) and GraphRAG (Edge et al., 2024) demonstrate the field's dynamic nature and progress in implementing long-term knowledge into models. In our study, we employ CRAG, which includes a web search tool that provides agents with relevant real-time data. This synergy expands the agents' knowledge base and improves their expertise. By introducing agents to trade knowledge, we aim to closely mirror the behavior and insights of real-world specialists.

2.5 Theory of Mind

ToM enables agents to reason about others' mental states (McLaughlin et al., 2011), facilitating effective collaboration and communication. In MAS, ToM allows agents to infer others' beliefs, intentions, and goals (Shoham and Leyton-Brown, 2012). While LLMs have shown promising results in ToM tasks (Li et al., 2023), further tests are needed in text-based problem-solving scenarios. ToM is crucial for agent collaboration, enabling them to understand each other's perspectives and make collective decisions. By incorporating ToM capabilities into our system, we aim to improve agents' social intelligence, making them more effective collaborators.

3 System Architecture: SynergyMAS

SynergyMAS is designed to leverage multiple agents working together to solve complex, domain-specific problems. Figure 3 illustrates the architecture of a single agent within this system, highlighting key components such as memory management, planning capabilities, knowledge retrieval and reasoning tools, and action execution. This agent structure forms the foundation of SynergyMAS and underpins the functionality discussed in the following subsections.
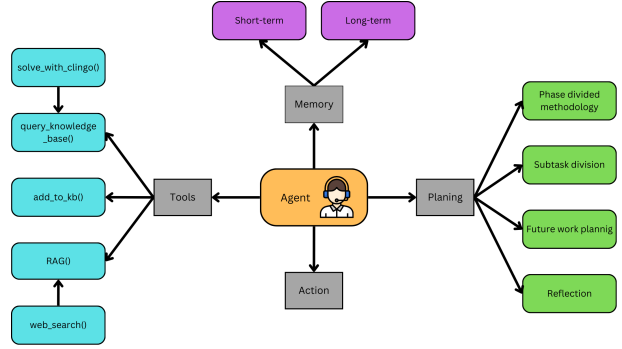

Figure 3: Agents architecture: illustrates the Synergy-MAS architecture, showcasing the key components and interactions of an agent within the multi-agent system (based on Weng (2023)).

3.1 System Philosophy and Objectives

The development of SynergyMAS is guided by a central question: How can a multi-agent system powered by Large Language Models (LLMs) effectively navigate and solve complex, domain-specific problems? The solution is to build a system capable of managing long, detailed conversations while addressing the unique characteristics of specialized domains.

To achieve this, SynergyMAS was designed with its core components of logical reasoning, knowledge retrieval, and Theory of Mind working together in synergy. This synergy is also about employing communication protocols crafted to unify individual agents' efforts, amplifying each approach's strengths. The system's hierarchical structure, with a central "boss" agent coordinating tasks, ensures all agents operate together in a constructive fashion.

The success of SynergyMAS is determined by its ability to maintain logical consistency, efficiently retrieve and synthesize knowledge, and adapt dynamically to evolving problem-solving scenarios. The true synergy in SynergyMAS lies in this orchestrated union, where each element enhances the others, resulting in a system with superior capabilities.

3.2 Overall System Design

The overall design of SynergyMAS revolves around a hierarchical structure, with a "boss" agent coordinating the workflow and ensuring that all agents align with the system's objectives. This structure integrates three key components: Logical Reasoning, RAG-Based Knowledge Management, and ToM Capabilities.

Logical Reasoning component is powered by a Neo4j graph knowledge base and a logic solver. Conversation data is continuously added to the graph, enabling agents to retrieve relevant information, which is then translated into an ASP query. The logic solver processes this query and returns a logical response in natural language.

RAG-Based Knowledge Management utilizes a modified version of Corrective RAG (CRAG). This component first retrieves information from a Chroma vector base containing domain-specific data. The query is forwarded to an external web search using the Tavily Search framework if relevant data is not found.

ToM capabilities capabilities are integrated through the "My Beliefs" section in each agent's response. This feature enables agents to infer and reason about the beliefs and strategies of others, promoting effective collaboration and coordination.

3.3 Agent Interactions

Agent interactions in SynergyMAS are governed by structured communication protocols, ensuring efficient problem-solving. After each task, control returns to the "boss" agent, who evaluates progress, tracks the conversation stage, and assigns new tasks. This process prevents redundancy and maintains focus on the current objectives.

Each agent's response is divided into three parts: My Beliefs, Response, and Future Work.

- **My Beliefs** reflects the agent's understanding of the task, incorporating insights from other agents and applying ToM to align with the team's overall strategy.

- **Response** section delivers task-specific solutions, leveraging logical reasoning and RAG for accuracy and relevance.

- **Future Work** outlines potential next steps and challenges, guiding the boss agent in steering the conversation forward without unnecessary deviations.

This structured approach ensures clarity, cohesion, and focus throughout the problem-solving process.

4 Logical Reasoning Component

The logical reasoning component in SynergyMAS integrates a Neo4j graph knowledge base with a logic solver, specifically Clingo, to manage and process complex queries. A logic solver is a tool
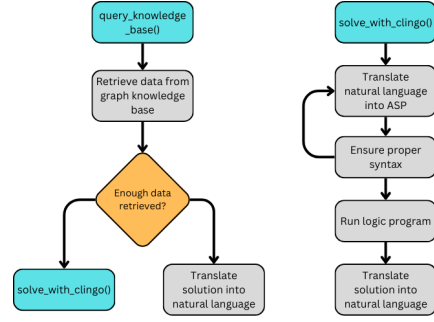

Figure 4: Logical Reasoning Functions: Shows the core logical functions of SynergyMAS, including knowledge base expansion and query solving.

that processes queries based on formal logic, allowing the system to infer conclusions from a set of given facts and rules. Clingo, a widely-used logic solver, leverages ASP to solve complex problems by generating potential solutions that satisfy all the logical constraints (Wang et al., 2024c), as illustrated in Figure 4.

4.1 Adding and Retrieving Data from the Knowledge Base

After each agent's response, the `add_to_kb` function is called to process the agent's answer and add relevant data to the Neo4j graph knowledge base.

When an agent needs to retrieve data, the `query_knowledge_base` function is invoked with the agent's question. The question is first translated into a Cypher query by an LLM, which retrieves the relevant data from the graph. If the amount of retrieved data is below a predefined threshold, the query is further translated into an ASP format and processed by the Clingo solver. If the retrieved data exceeds the threshold, it is directly translated into natural language and returned to the agent.

4.2 Implementation Using Logic Solver

When data retrieval from the graph is insufficient, the LLM translates the natural language question into a logical representation suitable for processing by the Clingo solver. For example, consider a scenario where the knowledge base includes facts about software development tasks:

$$task(ImplementFeatureX)$$

$$\forall x(task(x) \land assigned(x, Alice) \rightarrow completed(x))$$

$$assigned(ImplementFeatureX, Alice)$$

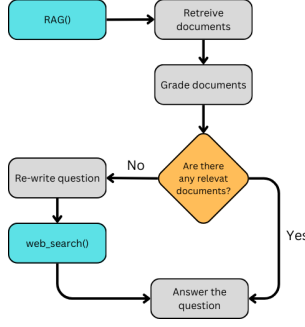The query might be: "Is the task ImplementFeatureX completed?" The logic solver would process this and infer:

Figure 5: Corrective-RAG flowchart: Document retrieval, grading, and adaptive question-answering process.

$$completed(ImplementFeatureX)$$

Safety mechanisms validate the ASP syntax to ensure logical consistency and correctness, allowing the LLM up to three attempts to produce valid code (McGinness and Baumgartner). During testing, the GPT-4o model demonstrated superior performance compared to the GPT-3.5-Turbo model, making it the preferred choice.

After processing by the Clingo solver, the ASP output is translated back into natural language, providing the agent with a clear and accurate response.

5 RAG Knowledge Base

The RAG knowledge base in SynergyMAS is designed to provide agents with access to domain-specific knowledge, making their knowledge and actions more similar to those of a domain specialist. The system leverages a modified version of CRAG, which enhances standard RAG by incorporating a document grading layer. This layer classifies the relevance of documents stored in a Chroma vector base, ensuring that the most relevant information is retrieved. If the internal database lacks relevant data, the query is forwarded to an external search using the Tavily web search tool.

Figure 5 illustrates the workflow of the modified CRAG system used in SynergyMAS. The retrieval process begins with query analysis, where the agent formulates a query based on the assigned task. CRAG then grades the relevance of documents within the database, retrieving information from the Chroma vector base if relevant data is available. The query is forwarded to the Tavily Search tool for external data if necessary. The query is optimized for internet search, and retrieved information is synthesized to generate a coherent and accurate response.

The vector-based storage system in Synergy-MAS enhances efficiency by enabling quick retrieval and reuse of processed data, allowing integration of documents. This scalable framework, supported by CRAG, ensures agents can access and utilize up-to-date, domain-specific knowledge. By combining internal databases with external web search capabilities, SynergyMAS provides a robust and flexible environment for knowledge retrieval, empowering agents to perform their specialized roles effectively and improving overall system performance.

6 Theory of Mind Implementation

In the SynergyMAS system, ToM capabilities enhance collaborative efficiency by enabling agents to reason about their own and others' mental states. ToM is implemented through explicit belief state representations, allowing agents to attribute beliefs, intents, and knowledge to themselves and their teammates.

Inspired by recent studies, the belief state system in SynergyMAS uses prompt engineering to represent and update belief states dynamically. Each agent's belief state evolves with new information and interactions, facilitating a deeper understanding of team dynamics and improving task execution. Figure 6 demonstrates how the 'My Beliefs' sections enable agents to integrate knowledge from different team members, fostering informed decision-making.

ToM capabilities are integrated into the *My Beliefs* section of agent responses, which builds upon two ideas:

- **Introspection:** Agents summarize their own mental states and recent actions, enhancing self-awareness.

- **First-Order ToM:** Agents infer the mental states of other agents based on shared information and observations.

By incorporating ToM, SynergyMAS significantly improves collaborative efficiency and overall system performance. Explicit belief state representations allow for aligned strategies and coherent problem-solving, resulting in better decision-making, more transparent communication, reduced errors, and increased efficiency. This ToM-enhanced approach demonstrates the potential for
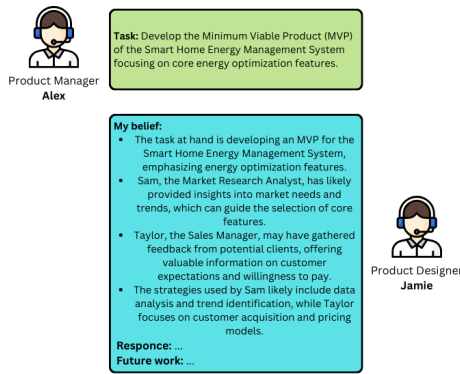
Figure 6: Belief Prompts for Collaboration: Demonstrates how 'My Beliefs' sections facilitate agent collaboration and informed decision-making.

creating more sophisticated, human-like AI systems that advance multi-agent collaboration across various domains.

7 Product Development Team Case Study (Lean Startup Methodology)

This case study demonstrates SynergyMAS's application in product development using the Lean Startup methodology. While this example focuses on a specific approach, the SynergyMAS framework is versatile and applicable to various problem-solving scenarios across different domains. Figure 7 illustrates the Lean Startup process used in this case study, highlighting the iterative Build-Measure-Learn cycles that guide product development.

7.1 Defining the Case Study

This case study explores how SynergyMAS can be used to develop a Smart Home Energy Management System, employing the Lean Startup methodology (Ries, 2017). The goal is to show how a team of specialized agents can collaboratively guide product development through iterative Build-Measure-Learn cycles. The primary objectives are to validate product-market fit, optimize energy management algorithms, and ensure the system meets market demands and regulatory standards. The sequential collaboration between Alex, Jamie, and Sam is depicted in Figure 8.

7.2 Team Structure and Roles

The product development team comprises a Product Manager (PM), a Market Research Analyst (MRA), a Product Designer (PD), and a Sales Manager (SM). Each agent specializes in a distinct do-

main, contributing their expertise to product development and refinement.

Product Manager (PM) - Alex

Role: Oversees the Lean Startup process, making key decisions about product direction and ensuring alignment with company goals.

Responsibilities: Alex defines the product vision and strategy, coordinates the Build-Measure-Learn cycles, synthesizes feedback from other agents, and ensures the project stays on track.

Market Research Analyst (MRA) - Sam

Role: Conducts market analysis and identifies customer needs and trends to inform the product development process.

Responsibilities: Sam provides data-driven insights, analyzes user data to identify emerging trends, collaborates with Alex to align market insights with the product vision, and works with Jamie to integrate market trends into the design process.

Product Designer (PD) - Jamie

Role: Responsible for designing the product, ensuring its usability and aesthetics.

Responsibilities: Jamie creates user-friendly and visually appealing designs, incorporates market trends and customer preferences, collaborates with Sam, and works with Taylor to align product design with sales strategies.

Sales Manager (SM) - Taylor

Role: Develops sales strategies and manages customer relationships.

Responsibilities: Taylor provides insights on customer preferences, sales potential, and go-to-market strategies, collaborates with Alex to ensure alignment with the product vision, works with Jamie to meet customer expectations, and gathers feedback to refine the product.

7.3 Analysis of Agent Interactions and Decision-Making Processes

This section explores how agents interact and make decisions during product development, covering key aspects such as:

- **Communication Protocol:**
  - The PM (Alex) oversees interactions, ensuring each agent contributes based on their expertise.

- Control returns to the PM after each agent's response, who then assesses the task and assigns the next one.
- The structured format ensures efficient information flow and collaborative decision-making.

- **Iterative Development:**
  - During the Build phase, Jamie develops the MVP, focusing on core features.
  - Sam and Taylor provide input to ensure the MVP meets market demands and sales potential.
  - Feedback is collected and analyzed to guide subsequent iterations.

- **Decision-Making:**
  - Decisions are based on quantitative data and qualitative feedback.
  - Alex synthesizes insights from all agents to decide whether to persevere or pivot.
  - Documentation of each phase ensures transparency and informed decision-making.

- **Conflict Resolution:**
  - Differences in opinions are addressed through structured discussions.
  - The PM ensures all perspectives are considered, selecting the best course of action.

8 Evaluation

Comparing SynergyMAS [1] directly with traditional systems like ChatGPT-4o, ChatGPT-4o with Chain of Thought (CoT), and ChatGPT-4o with Tree of Thoughts (ToT) is challenging due to their design differences (Wei et al., 2023; Yao et al., 2023; Guo et al., 2024). SynergyMAS excels in handling longer conversations and collaborative problem-solving, while the other models vary in their ability to maintain context and provide depth over extended interactions. To gain valuable insights, a controlled test scenario evaluated each system's performance in analyzing a Smart Home Energy Management System, a critical component in the Lean Startup methodology. The quality of each response was assessed to draw conclusions from the results.

---

[1] Code available at https://github.com/feilaz/SynergyMAS-Evaluation
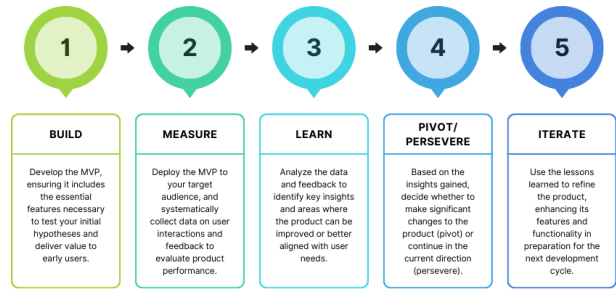

Figure 7: Diagram illustrating the Lean Startup methodology (Ries, 2017) steps used in our team case study.

8.1 Comparative Analysis with Existing Multi-Agent Systems

Structural Analysis: ChatGPT-4o provides a clear, concise structure but lacks detail in areas like energy savings variability. CoT offers a more comprehensive analysis but with some repetition. ToT enhances structure and provides future-oriented thinking with clear next steps. SynergyMAS balances structure and detail, offering specific data points and calculations, though with some inconsistencies in structuring.

Content Analysis: ChatGPT-4o focuses on key insights and next steps but lacks depth in areas like energy savings variability. CoT provides detailed analysis of specific data points, though with less emphasis on competitive analysis. ToT excels in a comprehensive, future-oriented analysis. SynergyMAS provides a thorough competitive analysis and justified priorities despite some redundancy in conclusions.

Analytical Depth: ChatGPT-4o offers a good overview but lacks depth in exploring causes of energy savings variability. CoT explores multiple factors in detail, providing AI improvement suggestions. ToT delivers the most in-depth analysis among single-model approaches, integrating future strategies. SynergyMAS offers the deepest analysis, exploring factors from multiple perspectives and providing a comprehensive strategy.

Unique Contributions: ChatGPT-4o emphasizes climate advantages and AI personalization. CoT analyzes energy savings variability with precise suggestions. ToT integrates a comprehensive, future-oriented perspective. SynergyMAS provides the most exhaustive competitive position analysis and detailed strategies for gamification, emphasizing user education for AI adoption.
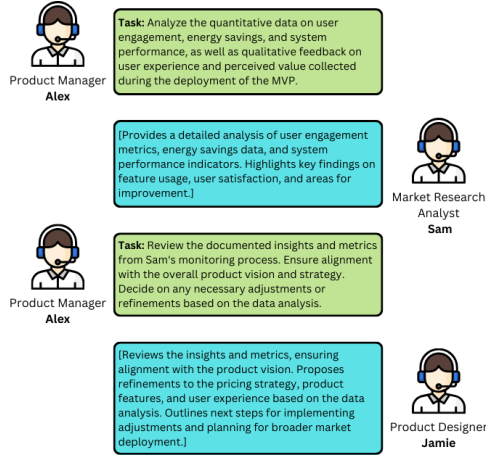
Figure 8: Multi-Agent Task Progression: Illustrates the sequential collaboration among agents to advance the Smart Home Energy Management System from concept to deployment.

8.2 Discussion of Strengths and Limitations

SynergyMAS excels in multi-perspective analyses with transparent reasoning processes. Its multi-agent design enhances collaborative problem-solving, effectively combining specialized knowledge (Wang et al., 2024b). The iterative improvement process and the ability to retrieve external information give SynergyMAS an advantage in complex scenarios. While ToT offers a highly structured, in-depth approach with a strong future-oriented perspective, SynergyMAS's ability to integrate diverse viewpoints and provide iterative improvements remains superior in complex scenarios. However, SynergyMAS could reduce redundancy in agent responses and improve analysis techniques. Its multi-agent nature may introduce variability (Chen et al., 2024c), raising efficiency concerns compared to single-model approaches like ChatGPT-4o, CoT, and ToT.

9 Future Work

Scalability is vital for SynergyMAS's future. As tasks grow in complexity, the system must manage more agents and handle longer, more intricate conversations. Future efforts will focus on optimizing the architecture to meet these challenges while staying goal-oriented. Potential enhancements include refining the hierarchical structure or adopting more flexible nested designs (Han et al., 2024), alongside advanced memory management techniques (Xie et al., 2024) and improved information synthesis across threads, making the framework more adaptable to evolving tasks (Wang et al., 2024a). These

advancements will equip SynergyMAS to tackle increasingly complex problems while maintaining coherence and efficiency.

While SynergyMAS has been demonstrated with specific use cases, such as a Smart Home Energy Management System, future work will also involve testing the framework on open-source models like LLaMA, Gemma, and PHI-3. This will help assess its adaptability and performance across a broader range of LLM architectures. Additionally, the framework's principles can be adapted to various other domains, such as healthcare for collaborative diagnostics and finance for market analysis. Similar systems have shown promise in software development for project management (Cinkusz and Chudziak, 2024a,b), and SynergyMAS could also support personalized learning in education. With the rapid emergence of new LLMs, expanding SynergyMAS to integrate with a wider variety of models will further enhance its applicability and effectiveness across diverse tasks and domains.

10 Conclusion

This paper has presented SynergyMAS, a multi-agent system designed to enhance collaborative problem-solving by integrating logical reasoning, RAG-based knowledge management, and Theory of Mind capabilities. The system's performance was evaluated in the context of developing a Smart Home Energy Management System, demonstrating significant improvements in analysis depth, data-driven insights, and actionable recommendations. SynergyMAS offers an ordered and iterative approach, using specialized agents to provide exhaustive and well-justified responses.

The development and evaluation of Synergy-MAS contribute to the field of multi-agent LLM research by showcasing the potential benefits of integrating advanced reasoning and collaboration capabilities. The system's proficiency in handling complex tasks suggests that similar approaches could be beneficial in other domains requiring refined problem-solving and decision-making. By emphasizing the importance of scalability and versatility, this work lays the groundwork for future research and development in multi-agent systems, ultimately aiming to create more intelligent and effective AI solutions for various applications.

References

Maciej Besta, Robert Gerstenberger, Emanuel Peter, Marc Fischer, Michał Podstawski, Claude Barthels, Gustavo Alonso, and Torsten Hoefler. 2023. Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries. *Preprint*, arXiv:1910.09017.

Roberta Calegari, Giovanni Ciatto, Viviana Mascardi, and Andrea Omicini. 2020. Logic-based technologies for multi-agent systems: A systematic literature review - autonomous agents and multi-agent systems.

Meiqi Chen, Yubo Ma, Kaitao Song, Yixin Cao, Yan Zhang, and Dongsheng Li. 2023. Learning to teach large language models logical reasoning. *Preprint*, arXiv:2310.09158.

Meiqi Chen, Yubo Ma, Kaitao Song, Yixin Cao, Yan Zhang, and Dongsheng Li. 2024a. Improving large language models in event relation logical prediction. *Preprint*, arXiv:2310.09158.

Minyu Chen, Guoqiang Li, Ling-I Wu, Ruibang Liu, Yuxin Su, Xi Chang, and Jianxin Xue. 2024b. Can language models pretend solvers? logic code simulation with llms. *Preprint*, arXiv:2403.16097.

Pei Chen, Boran Han, and Shuai Zhang. 2024c. Comm: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving. *Preprint*, arXiv:2404.17729.

Konrad Cinkusz and Jaroslaw A. Chudziak. 2024a. Communicative agents for software project management and system development. In *Proceedings of the 21st International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2024)*.

Konrad Cinkusz and Jaroslaw A. Chudziak. 2024b. Towards llm-augmented multiagent systems for agile software engineering. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, ASE '24, page 2476–2477, New York, NY, USA. Association for Computing Machinery.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Michael Gelfond and Yulia Kahl. 2014. *Knowledge representation, reasoning, and the design of Intelligent Agents: The answer-set programming approach*. Cambridge University Press.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *Preprint*, arXiv:2402.01680.

Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, Zhaozhuo Xu, and Chaoyang He. 2024. Llm multi-agent systems: Challenges and open problems. *Preprint*, arXiv:2402.03578.

Michal Kosinski. 2024. Evaluating large language models in theory of mind tasks. *Preprint*, arXiv:2302.02083.

Hobson Lane, Hannes Hapke, and Cole Howard. 2019. *Natural language processing in action understanding, analyzing, and generating text with python*. Manning Publications: distributed by Skillsoft Books.

Bin Lei. 2024. Macm: Utilizing a multi-agent system for condition mining in solving complex mathematical problems. *Preprint*, arXiv:2404.04735.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Huao Li, Yu Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Lewis, and Katia Sycara. 2023. Theory of mind for multi-agent collaboration via large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yuanyuan Liang, Keren Tan, Tingyu Xie, Wenbiao Tao, Siyuan Wang, Yunshi Lan, and Weining Qian. 2024. Aligning large language models to a domain-specific graph database. *Preprint*, arXiv:2402.16567.

Lachlan McGinness and Peter Baumgartner. Automated theorem provers help improve large language model reasoning. In *EPiC Series in Computing*. EasyChair.

Brian P. McLaughlin, Ansgar Beckermann, and Sven Walter. 2011. *The Oxford Handbook of Philosophy of Mind*. Oxford University Press.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models. *Preprint*, arXiv:2307.06435.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Sebastian Raschka. 2024. *Build a large language model*. Manning Publications.

Eric Ries. 2017. *The Lean Startup: How Today's entrepreneurs use continuous innovation to create radically successful businesses*. Currency.

Stuart J. Russell and Peter Norvig. 2016. *Artificial Intelligence: A modern approach*. Pearson.

Yoav Shoham and Kevin Leyton-Brown. 2012. *Multiagent Systems Algorithmic, game-theoretic, and logical foundations Yoav Shoham; Kevin Leyton-Brown*. Cambridge Univ. Press.

Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, Yue Wu, Wenhai Wang, Junsong Chen, Zhangyue Yin, Xiaozhe Ren, Jie Fu, Junxian He, Wu Yuan, Qi Liu, Xihui Liu, Yu Li, Hao Dong, Yu Cheng, Ming Zhang, Pheng Ann Heng, Jifeng Dai, Ping Luo, Jingdong Wang, Ji-Rong Wen, Xipeng Qiu, Yike Guo, Hui Xiong, Qun Liu, and Zhenguo Li. 2024. A survey of reasoning with foundation models. *Preprint*, arXiv:2312.11562.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2021. Proofwriter: Generating implications, proofs, and abductive statements over natural language. *Preprint*, arXiv:2012.13048.

Qian Wang, Tianyu Wang, Qinbin Li, Jingsheng Liang, and Bingsheng He. 2024a. Megaagent: A practical framework for autonomous cooperation in large-scale llm agent systems. *Preprint*, arXiv:2408.09955.

Yulong Wang, Tianhao Shen, Lifeng Liu, and Jian Xie. 2024b. Sibyl: Simple yet effective agent framework for complex real-world reasoning. *Preprint*, arXiv:2407.10718.

Zhongsheng Wang, Jiamou Liu, Qiming Bao, Hongfei Rong, and Jingfeng Zhang. 2024c. Chatlogic: Integrating logic programming with large language models for multi-step reasoning. *Preprint*, arXiv:2407.10162.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Lilian Weng. 2023. Llm-powered autonomous agents. *lilianweng.github.io*.

Michael J. Wooldridge. 2009. *An introduction to multiagent systems*. John Wiley Sons.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *Preprint*, arXiv:2308.08155.

Weijian Xie, Xuefeng Liang, Yuhui Liu, Kaihua Ni, Hong Cheng, and Zetian Hu. 2024. Weknow-rag: An adaptive approach for retrieval-augmented generation integrating web search and knowledge graphs. *Preprint*, arXiv:2408.07611.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *Preprint*, arXiv:2401.15884.

Zhun Yang, Adam Ishay, and Joohyung Lee. 2023. Coupling large language models with logic programming for robust and general reasoning from text. *Preprint*, arXiv:2307.07696.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.