Exploring behavior of LLM's as part of Models of Consciousness

CS 297 Report

Presented to:

Dr. Chris Pollett

Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements for the Class

CS 297

Presented By:

Adithya Harish Kumar

December, 2024

ABSTRACT

This semester's work explores approaches to artificial intelligence (AI) and cognitive modeling, focusing on the implementation and experimentation on different models of consciousness using CrewAI and Large Language Models (LLMs). The research encompasses the development of an LLM Topology Manager, a sophisticated framework for orchestrating complex LLM interactions, and its application in query answering systems.

The LLM Topology Manager introduces a framework designed for advanced AI interactions, incorporating Infrastructure as Code principles and Agentic Retrieval-Augmented Generation (RAG) to enhance configurability, reproducibility, and performance. This system enables the creation of user-defined topologies for LLM interactions, offering a flexible and powerful solution for a wide range of AI-driven applications and research projects.

Additionally, the project delves into housefly brain processes and sensorimotor pathways, creating a detailed image-caption dataset and implementing a comprehensive housefly sensorimotor pathway using the Topology Manager framework. The research extends its scope to human brain processes, exploring the Default Mode Network (DMN) and setting the stage for future research in human visual pathways. By bridging the gap between insect and human brain functions, the project offers a comparative perspective on neural networks across species.

Keywords – Multiple Drafts model, CrewAI, LLM Topology Manager, Infrastructure as Code, Agentic RAG, housefly sensorimotor pathways, Default Mode Network

TABLE OF CONTENTS

I. INTRODUCTION

Large Language Models (LLMs) can simulate multi-stage real-world processes by representing each stage as a modular task performed by specialized instances of the model. By chaining tasks, the LLM can mimic the entire workflow, ensuring data consistency between stages and adapting dynamically to new constraints or inputs. This approach enables automation of complex, interdependent processes with clear delineation of responsibilities at each step.

Models of consciousness aim to explain the mechanisms underlying conscious thought and perception and can be implemented using Large Language Models (LLMs) to simulate these processes. Dennett's Multiple Drafts model suggests that consciousness arises from parallel processes in the brain that create multiple competing narratives [2], with no central observer deciding which is the "true" perception, a concept that can be mirrored by LLMs operating in distributed, narrative-focused roles. The Dehaene-Changeux model, which proposes a Global Neuronal Workspace for broadcasting and integrating information [1], can be simulated by orchestrating LLMs in a hierarchical architecture where key outputs are shared across interconnected agents. These implementations provide computational tools to explore and validate theories of consciousness.

Agentic Retrieval-Augmented Generation (RAG) combines retrieval systems with LLMs to create task-specific agents that can query external knowledge bases [4], process retrieved information and generate context-aware outputs. Frameworks like Crew AI use this approach to coordinate multiple agents, each focused on a distinct aspect of a complex problem, enabling efficient division of labor and iterative refinement of tasks. We improve upon this approach by building a framework for orchestrating complex LLMs as agents and their interactions by providing a mechanism to configure the interaction topology of these agents through declarative definitions.

This research then extends its scope to implementing biological systems using the topology manager, focusing on housefly brain processes and sensorimotor pathways. By creating a detailed image-caption dataset and implementing a housefly sensorimotor pathway using the Topology Manager framework, the project aims to bridge the gap between artificial and biological cognitive systems. The study explores the Default Mode Network in human brains, setting the stage for future research in human visual pathways and providing a comparative perspective between insect and human brain functions.

This report is divided into 4 sections representing the progress on different deliverables created in this semester. In section 2 we create an implementation of Dennett's Multiple Drafts model of consciousness using CrewAI to demonstrate the feasibility of using an agentic LLM framework to realize theoretical models of consciousness. In section 3 contains the design for an LLM topology manager which we built to configure and support agentic interactions. In section 4 we implement Dennett's Multiple Drafts model using the topology manager we built and compare it with the system in section 2. In the last section, we map the brain processes of a housefly onto different LLMs arranged in a logical representation of their brain regions. We implement this on our topology manager and demonstrate reasonable similarity of output actions as to what a real housefly would do. Section 6 concludes the work completed in this report.

II. 2 LLM Multiple Drafts model

This deliverable demonstrates a CrewAI application to implement Daniel Dennett's Multiple Drafts model of consciousness. This will act as a comparative system to the LLM topology manager which we will build in deliverable 3 to demonstrate how multiple LLMs can be used in a collaborative manner. The implementation utilizes three Large Language Models (LLMs) in an agentic framework, working in concert to process and interpret information, mirroring the parallel processing and integration of multiple cognitive streams proposed by Dennett's theory.

CrewAI is an agentic LLM framework designed to facilitate the creation and coordination of multiple AI agents working together to solve complex tasks like LLM based data retrieval as in [4]. CrewAI's architecture is built on the concept of a "crew" – a group of AI agents that collaborate, share information, and perform tasks in a coordinated manner. Each agent within a CrewAI system can be equipped with specific tools, knowledge bases, and decision-making capabilities, allowing for a diverse and complementary set of skills within the crew.

The system architecture consists of two constrained LLMs and one primary LLM:

1. **LLM 1 (Quantitative Focus)**: This agent is tasked with summarizing any quantities described in the input question.
2. **LLM 2 (Logical Relationships Focus)**: This agent identifies and summarizes logical relationships or interactions between entities in the input question.
3. **Primary LLM (Integrator)**: This agent integrates the outputs from LLM 1 and LLM 2 to formulate a comprehensive answer to the original question.

The CrewAI framework orchestrates the interaction between these agents, allowing for a structured flow of information that emulates the Multiple Drafts model's concept of parallel processing and integration of cognitive content.

The outputs from each LLM for these questions provide insight into how the system breaks down and processes complex queries, showcasing the potential of this CrewAI-based implementation of the Multiple Drafts model.

The system processes input questions through the following steps:

1. The input question is simultaneously presented to LLM 1 and LLM 2.
2. LLM 1 and LLM 2 process the question according to their specific constraints and generate summaries.
3. The Primary LLM receives the summaries from LLM 1 and LLM 2, along with the original question.
4. The Primary LLM integrates this information to produce a definitive answer.

Table. 1 Experimental Outputs for Dennett's Model

Question	LLM 1	LLM 2	Primary	Answer
I have 100 apples. I gave 20 apples to my sister and 10 apples to my brother. How many apples do I have left?	Before giving any apples away, I had 100 apples. After giving 20 apples to my sister and 10 apples to my brother, I am left with 70 apples.	Before giving any apples away, I had 100 apples. After giving 20 apples to my sister and 10 apples to my brother, I am left with 70 apples.	After giving 20 apples to my sister and 10 apples to my brother, I am left with 70 apples. So, I have 70 apples left.	70 apples
There are 10 crows on an electric line. If a hunter shoots the	The quantities described in the question are: 10 crows initially on the electric line. 1	The logical relationships and interactions between entities in the question	If we only consider the quantities and interactions mentioned, it seems like 9 crows should be left on the	0 crows

leftmost crow and it falls to the ground, how many crows will be left on the electric line?	crow shot by the hunter.	are: There is an initial number of crows sitting on an electric line. A hunter shoots one of the crows, specifically the leftmost one. The shot crow falls to the ground. The question asks to determine the number of crows remaining on the electric line after the shooting incident	line. However, typically, if a hunter shoots, the sound would scare the remaining crows away, which would mean no crows are left on the line. Without additional context, and considering only the quantities explicitly stated, the answer could be interpreted as 9 crows. But if we consider a realistic scenario where the noise would scare away the rest, the answer would be 0 crows left on the electric line.	
Statements: All trucks fly. Some scooters fly. Conclusions: All trucks are scooters. Some scooters do not fly. Which of the conclusions are correct according to the statements?	Analyzing the statements, we can see that all trucks are stated to fly, indicating that the entire set of trucks possess the attribute of flying. On the other hand, only some scooters are mentioned to fly, implying that there is a subset of scooters that can fly, but not necessarily all of them. Therefore, the correct conclusion according to the statements is that all trucks are not scooters, as there is no direct correlation between the flying attribute	In conclusion, the statements do not support the conclusions that all trucks are scooters and that some scooters do not fly.	Based on the statements provided, the correct conclusion is that all trucks are not scooters, as there is no direct correlation between trucks and scooters in terms of the flying attribute. The statements do not provide enough evidence to conclude that some scooters do not fly, as it only mentions that some scooters fly without specifying the proportion that do not. Therefore, the conclusions that all trucks are scooters and that some scooters do not fly are not supported by the statements.	Conclusion 2 is correct and Conclusion 1 is incorrect

	and the type of vehicle.			

This approach not only demonstrates the practical application of Dennett's theoretical model but also highlights the power of CrewAI in creating sophisticated, multi-agent AI systems capable of tackling complex cognitive tasks through collaborative information processing but has the drawback of not being able to control the topology of the LLM interactions and suffers from a unified memory problem where all the LLMs have information about each other’s interaction which deviates from biological brain processes where specific neural processes are active for different tasks.

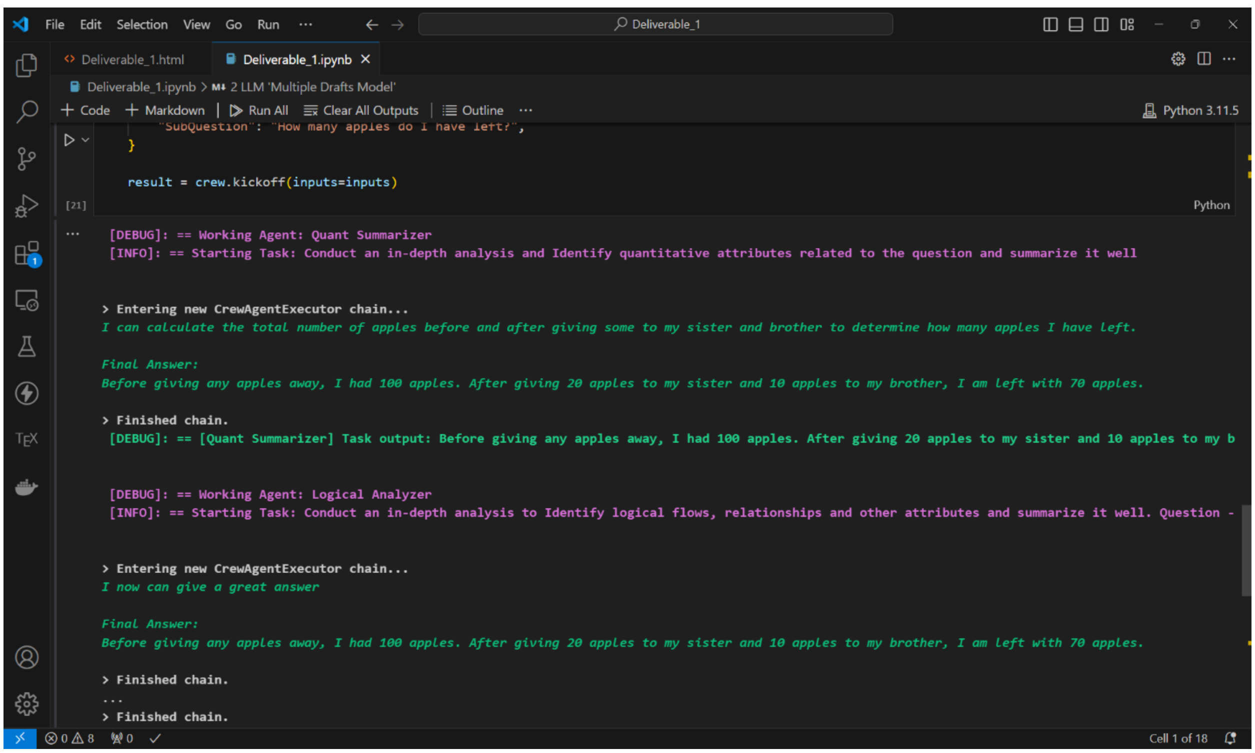

Fig. 1 CrewAI Agent solving Question 1

III. An LLM Topology Manager

In deliverable 2, we design and develop LLM Topology Manager. It is a framework designed and built for advanced AI interactions. This framework provides us with a way to configure how the LLMs interact with each other, which LLMs they interact with and what data is shared between them. This is a feature absent in existing systems like Crew AI. The LLM Topology Manager is a framework designed to orchestrate complex interactions with Large Language Models (LLMs) in a user-defined topology.

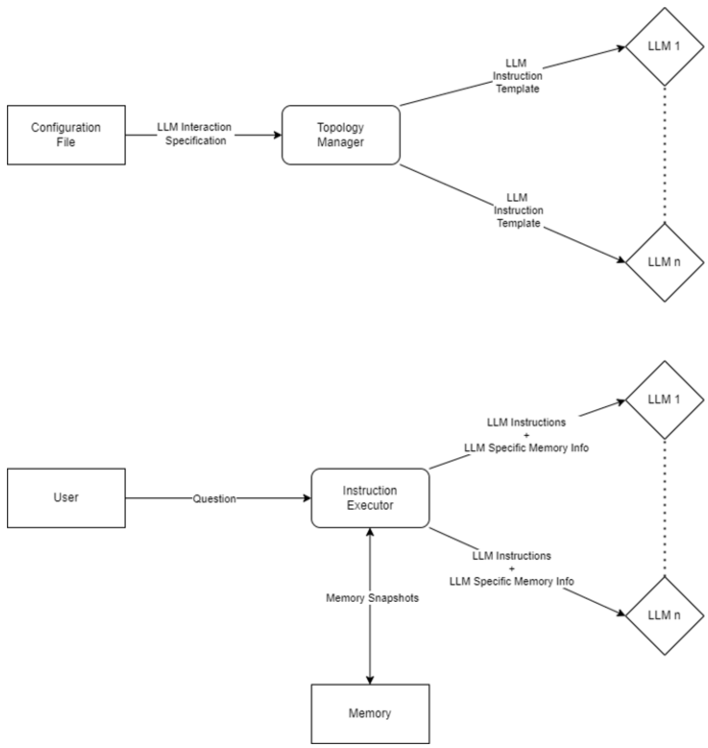

Fig. 2 System Design for the Topology Manager

The LLM Topology Manager comprises several key components that work in harmony to process user queries and generate intelligent responses:

1. **Configuration File:** Acts as the blueprint for the LLM environment, defining interaction parameters and infrastructure details.

2. **Topology Manager:** Transforms the configuration into LLM instruction templates, ensuring a Directed Acyclic Graph (DAG) structure for optimal execution flow.
3. **Instruction Executor:** Serves as the central processing unit, managing query dispatch and leveraging memory snapshots for contextual understanding.
4. **Memory Module:** Maintains contextual information through memory snapshots, enhancing the relevance and accuracy of LLM responses.
5. **LLM Modules:** Individual units that process specific instructions and generate tailored responses based on user queries and memory context.

The system has a modular architecture, with each component designed for specific functionalities:

1. **Config Module**: Validates infrastructure files and ensures proper graph structure.
2. **Core Module**: Manages the flow of interactions between user queries and LLM responses.
3. **DB Module**: Implements web scraping and Retrieval-Augmented Generation (RAG) using LanceDB.
4. **Manager Module**: Orchestrates model initialization and verifies graph acyclicity.
5. **Models Module**: Provides an abstraction layer for LLM querying via OpenRouter APIs OpenRouter and OpenAI and local execution with LMStudio.

The LLM Topology Manager leverages Infrastructure as Code principles to enhance configurability and reproducibility. IaC allows for:

1. Declarative Configuration: LLM topologies and interactions are defined in machine-readable files, typically in JSON or YAML format.
2. Version Control: Configuration files can be versioned, enabling easy tracking of changes and rollbacks.
3. Automated Deployment: The system can automatically set up and configure LLM environments based on the IaC specifications.

4. Consistency: IaC ensures that the LLM topology is consistently deployed across different environments, reducing configuration drift.

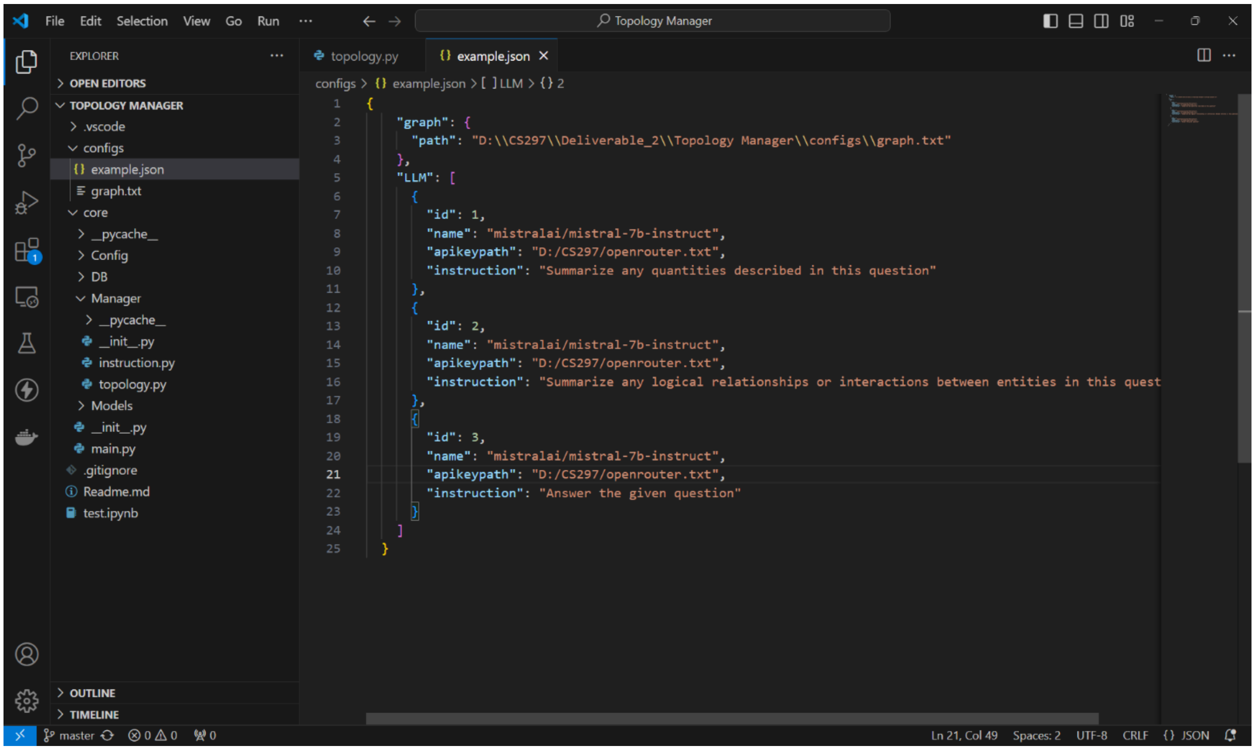

Fig. 3 A JSON based IaC Configuration of LLMs

The system incorporates Agentic Retrieval-Augmented Generation (Agentic RAG), a powerful technique that significantly improves LLM performance:

1. Dynamic Information Retrieval: Agentic RAG allows LLMs to actively seek and incorporate relevant information from external sources during the generation process.
2. Contextual Understanding: By leveraging vector search capabilities of LanceDB, the system can efficiently retrieve and utilize context-specific information.
3. Improved Accuracy: The combination of web scraping and RAG enhances the LLM's ability to provide up-to-date and factually accurate responses.
4. Adaptive Learning: Agentic RAG enables the system to continuously improve its knowledge base, adapting to new information and user interactions.

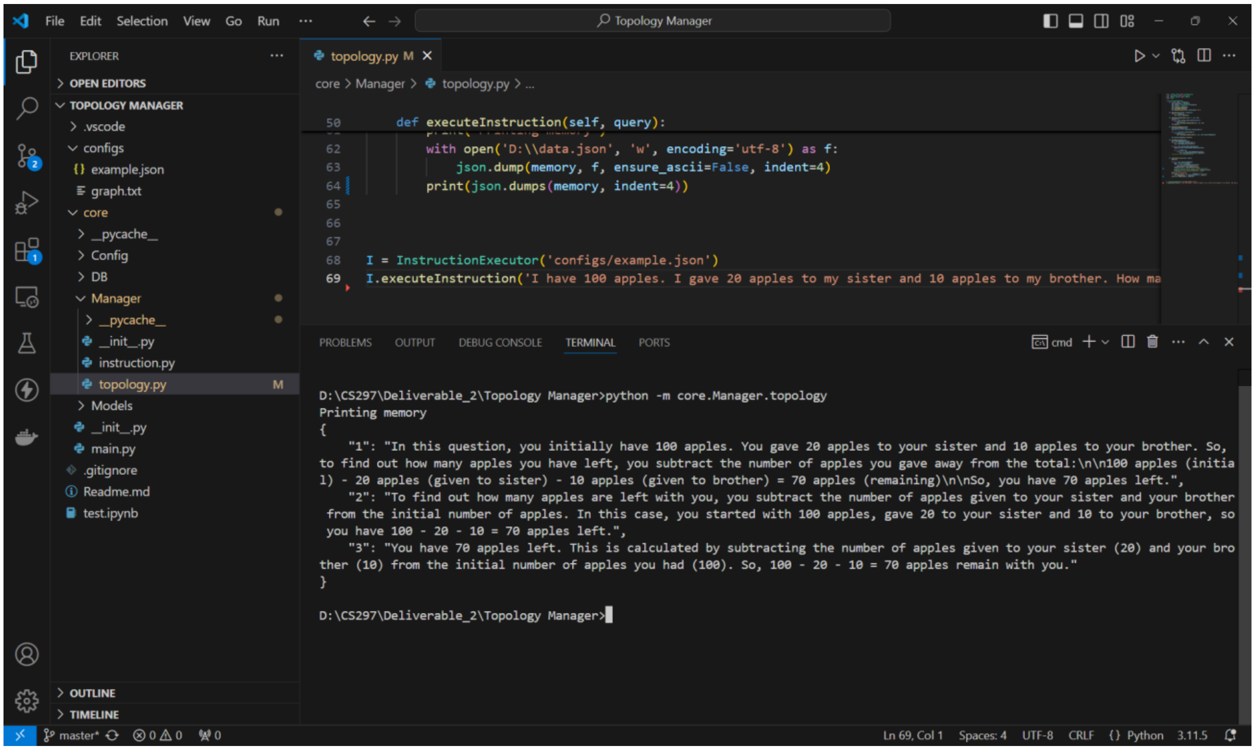

Fig. 4 Topology Manager in action

The LLM Topology Manager through its LLM modularity and network flow based design offers a flexible, powerful, and efficient framework for complex LLM interactions. By integrating IaC principles and Agentic RAG capabilities, it provides a robust solution for a wide range of AI-driven applications and research projects.

IV. LLM Topology Configuration for Query Answering

Deliverable 3 presents an effort to implement Daniel Dennett's Multiple Drafts model using the LLM (Large Language Model) topology manager we created as part of deliverable 2. The primary objective is to use this implementation to demonstrate the topology manager's capabilities by performing a task similar to Crew AI from deliverable 1. This approach aims to demonstrate the capabilities of our Topology Manager with the added benefit of being able to configure agent topology and a localized memory for each LLM agent.

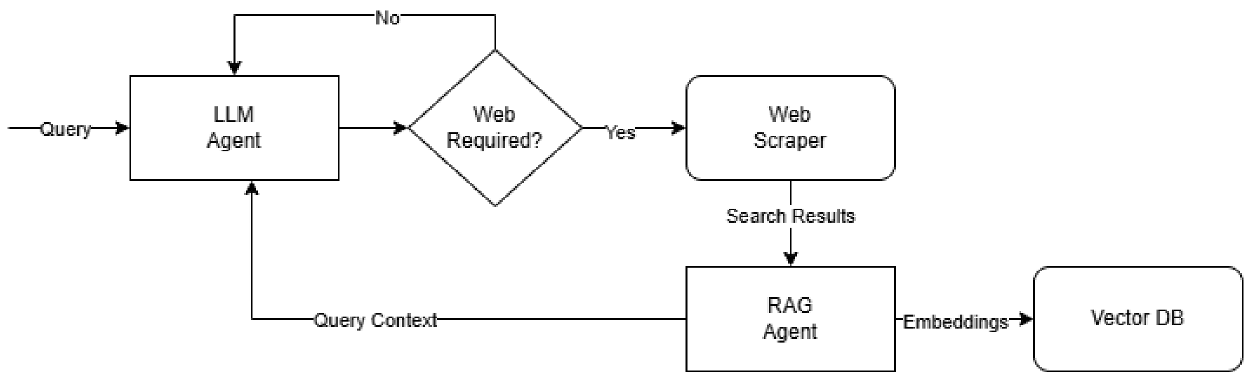

Fig. 5 RAG Pipeline

In addition to the hierarchical LLM structure, another agentic RAG (Retrieval-Augmented Generation) pipeline was created to enhance the system's capabilities. This pipeline incorporates the following components:

1. **Web Scraping**: A script was developed to scrape results from the web using the query as parameters. This allows the system to gather up-to-date and relevant information from online sources.
2. **On-Device LLM Solution**: The scraped results were used as context to answer questions using a completely on-device LLM solution via LMStudio. This approach enables efficient processing without relying on external API calls.

3. **Vector Database**: The results were stored in LanceDB, a vector database that allows for efficient storage and retrieval of high-dimensional data.
4. **Embedding Model**: The nomic embed v1.5 model was used for generating embeddings. This model helps in creating meaningful vector representations of the text data, enabling efficient similarity searches and retrieval.

This additional pipeline complements the hierarchical LLM structure by providing real-time, contextually relevant information from the web, potentially improving the accuracy and relevance of the system's responses.

The architecture consists of a hierarchical structure of LLMs, designed to process and synthesize information in stages:

1. LLM 1: Summarizes quantitative information within the input question.
2. LLM 2: Identifies and summarizes logical relationships or interactions in the question.
3. Primary LLM: Combines the summaries from LLM 1 and LLM 2, interpreting the results to answer the question.

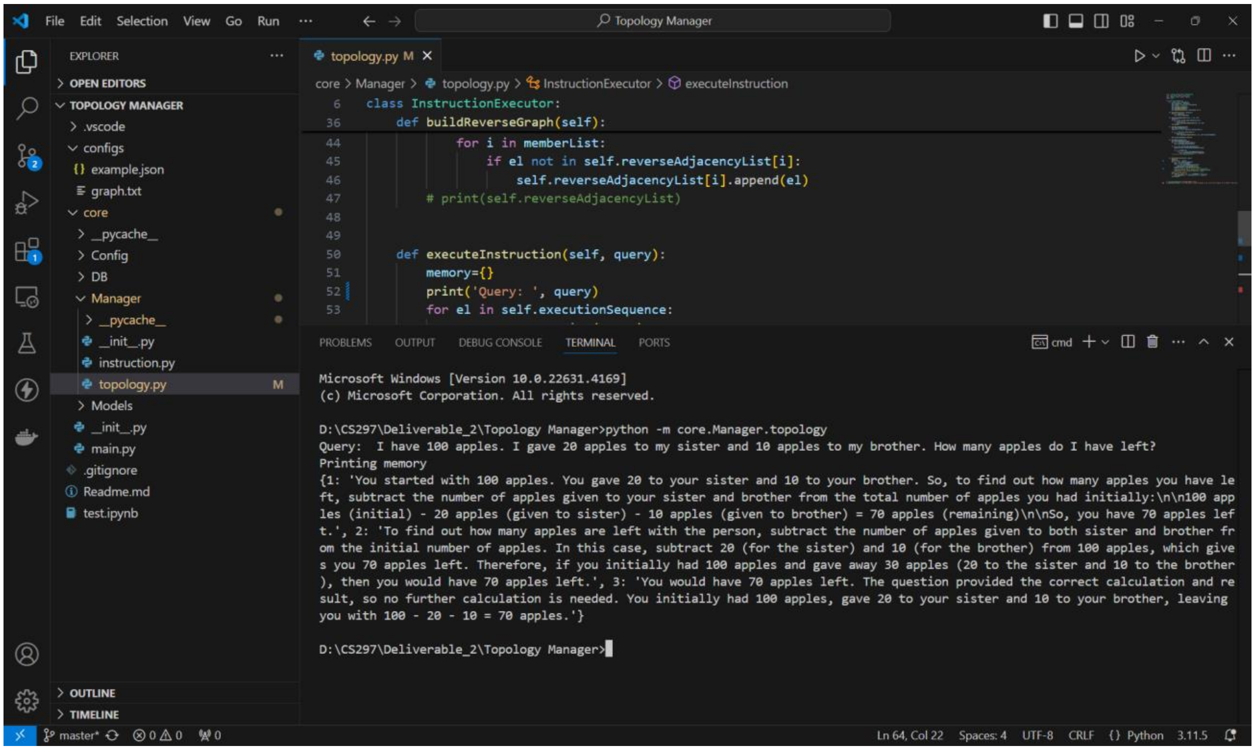

Fig. 6 Outputs of different LLMs for a Logical Reasoning Question

The question set is the same one used in deliverable 1 and this deliverable also acts as a contrast between the 2 systems. This hierarchical approach allows for a more nuanced and comprehensive analysis of the input question, potentially leading to more accurate and insightful responses.

Table. 2 Topology Manager Outputs for Dennett's Model

Question	**LLM 1**	**LLM 2**	**Primary**	**Answer**
I have 100 apples. I gave 20 apples to my sister and 10 apples to my brother. How many apples do I have left?	You started with 100 apples. You gave 20 to your sister and 10 to your brother. So, to find out how many apples you have left, subtract the number of apples given to	To find out how many apples are left with the person, subtract the number of apples given to both sister and brother from the initial number of apples. In this	You would have 70 apples left. The question provided the correct calculation and result, so no further calculation is needed. You initially had 100 apples, gave 20 to your sister and 10 to your brother, leaving you	70 apples

	your sister and brother from the total number of apples you had initially: 100 apples (initial) - 20 apples (given to sister) - 10 apples (given to brother) = 70 apples (remaining) So, you have 70 apples left.	case, subtract 20 (for the sister) and 10 (for the brother) from 100 apples, which gives you 70 apples left. Therefore, if you initially had 100 apples and gave away 30 apples (20 to the sister and 10 to the brother), then you would have 70 apples left.	with 100 - 20 - 10 = 70 apples.	
There are 10 crows on an electric line. If a hunter shoots the leftmost crow and it falls to the ground, how many crows will be left on the electric line?	The question describes a quantity of 10 crows that were on an electric line. After the hunter shoots and kills the leftmost crow, there will be 9 crows remaining on the electric line. The information about the fallen crow or the ground is not relevant to the remaining crows on the line, hence not included in the summary.	The logical relationship here is that if the leftmost crow is shot and falls to the ground, the number of crows remaining on the electric line will decrease by 1, as the fallen crow is no longer part of the group on the line. Initially, there are 10 crows on the electric line, and after the leftmost crow is shot and falls to the ground, there will be 9 crows left on the electric line.	There will be 9 crows left on the electric line. This is because when the leftmost crow is shot and falls to the ground, it is no longer part of the group on the line, resulting in a decrease of 1 from the initial number of 10 crows.	0 crows
Statements: All trucks fly. Some scooters fly. Conclusions: All trucks are scooters.	The conclusion that "All trucks are scooters" is not correct according to the given statements. While some scooters can	Given the two statements "All trucks fly" and "Some scooters fly", one can conclude that "Some scooters do	Based on the given information and reasoning, it can be asserted that the conclusion "Some scooters do not fly" is a correct statement as	Conclusion 2 is correct and Conclusion 1 is incorrect

Some scooters do not fly. Which of the conclusions are correct according to the statements?	fly (as per the second statement), the first statement states that all trucks fly, which contradicts the general understanding that trucks do not fly like scooters. The conclusion that "Some scooters do not fly" is correct, as it is directly stated in the second statement.	not fly" is a correct conclusion since the second statement explicitly states some scooters can fly, but doesn't say all scooters fly. However, the conclusion that "All trucks are scooters" is not a correct conclusion because, per the general understanding, trucks are not commonly associated with flying capabilities.	it does not contradict either of the provided statements. By contrast, the conclusion "All trucks are scooters" is not a supportable conclusion since it neither agrees with nor follows logically from both of the initial statements.	

V. Brain Processes

This deliverable aims at implementing a sensorimotor pathway of a housefly using the Topology Manager. The goal of this deliverable is to implement a neurobiological process with LLMs modelling their subsystems, the inputs being stimuli (audio, visual, text) and the outputs fed to subsequent subsystems. This work builds a foundation and acts as a proof-of-concept for exploring the possibility of LLMs modelling more complex visual pathways of the human brain in future work. We again make use of Topology Manager developed in deliverable 2.

We created a comprehensive image-caption pair dataset for housefly behavior analysis. This dataset, comprising 64 samples from 4 diverse housefly videos, involved frame-by-frame extraction and detailed captioning of observed behaviors. This meticulously prepared dataset serves as a crucial resource for training and evaluating machine learning models in housefly behavior recognition and analysis, laying the groundwork for more advanced studies in insect neurobiology and behavior. This dataset can be used for multimodal LLM interaction.

We then implemented a housefly sensorimotor pathway using the Topology Manager framework. This implementation includes seven key subsystems: Stimulus Selection, Head Direction, Internal Drives, Circadian Rhythm Simulation, Multisensory Integration Simulation, Self-Motion Processing, and Optic Flow Simulation. Each subsystem was designed with specific inputs, outputs, and duties, collectively modeling the complex sensory processing and decision-making capabilities of a housefly. This comprehensive model provides insights into how houseflies navigate their environment, respond to stimuli, and maintain internal homeostasis.

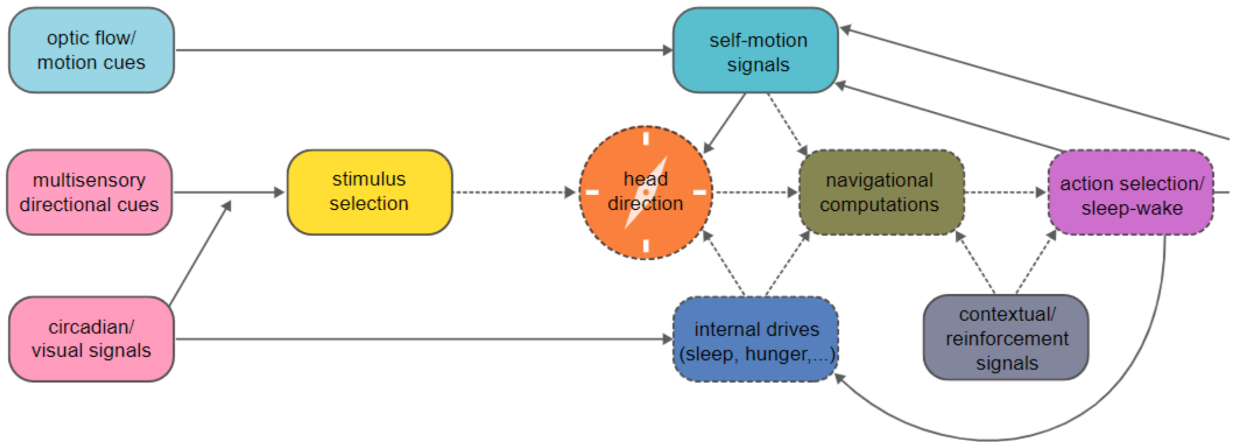

Fig. 7 Housefly Sensorimotor Pathways

To improve the functionality of our model, we enhanced the Topology Manager to maintain a history of previous outputs. This modification involved storing past outputs in memory and feeding them back to the agent, enabling more context-aware and historically informed decision-making processes. This improvement significantly enhances the model's ability to simulate the continuous, adaptive nature of insect behavior, providing a more realistic representation of how houseflies process and respond to their environment over time.


```
{
    "query": "Behave like you are this subsystem and generate the outputs as specified and provide one of the outputs you are created to
    "memory": {
        "4": "Circadian Signals: prepare for sleep",
        "7": "Internal Drives: seek sleep.",
        "3": "Multisensory Cues: head towards the sound, \nMultisensory Cues: orient to light source.",
        "5": "Stimulus Selection: orient to light source.",
        "1": "Optic Flow: turning right",
        "2": "Self-Motion: adjust heading slightly left",
        "6": "Output: Maintain a stable heading, adjust heading slightly left."
    }
}
```


Fig. 8 Output of Topology Manager for a particular timestep

Expanding the scope of our research, we conducted preliminary tests using the LLaVa-llama3 vision-language model (VLM) for simple instruction tasks. This exploration into multimodal agent capabilities lays the foundation for future integration of visual and linguistic processing in our human neurobiological system.

As a step towards extending our research to human brain processes, we explored the Default Mode Network (DMN), a large-scale brain network crucial in human cognition. The DMN, composed of regions including the dorsal medial prefrontal cortex and posterior cingulate cortex, is most active during states of wakeful rest and internal thought processes. This exploration provides a comparative perspective between insect and human brain functions, highlighting the complexity of neural networks across species and setting the stage for future research in human visual pathways as planned for the CS298 project.

VI. CONCLUSION

This project helped me make significant strides in understanding both artificial and biological cognitive systems. The successful implementation of Dennett's Multiple Drafts model using CrewAI demonstrated the potential for multi-agent AI systems to tackle complex cognitive tasks through collaborative information processing.

The development of the LLM Topology Manager represents a significant advancement in generative AI system design, offering a flexible, powerful, and efficient framework for complex LLM interactions. The exploration of housefly brain processes and sensorimotor pathways using this topology manager has yielded valuable insights into modelling insect neurobiology using generative AI. The creation of a comprehensive image-caption dataset and the implementation of a detailed sensorimotor pathway model lay the groundwork for more advanced studies in this field. Further, the preliminary exploration of the Default Mode Network (DMN) in human brains opens up new avenues for comparative studies between insect and human cognitive processes, setting the stage for research in modelling human visual pathways using LLMs.

In the upcoming semester, I will build upon this research and update the topology manager for processing text, audio and video inputs. Such a system with multimodal input capabilities would be desirable for modelling human visual pathways and implement human brain processes like the default model network (DMN).

REFERENCES

[1] [Stanislas1998] A neuronal model of a global workspace in effortful cognitive tasks. Stanislas Dehaene; Michel Kerszberg; Jean-Pierre Changeux;. Proc. Natl. Acad. Sci. USA Vol. 95, pp. 14529-14534. November 1998.

[2] [Sattin2021] Sattin, D.; Theoretical Models of Consciousness: A Scoping Review. Magnani, F.G.; Bartesaghi, L.; Caputo, M.; Fittipaldo, A.V.; Cacciatore, M.; Picozzi, M.; Leonardi, M. Brain Sci. 2021, 11, 535. https://doi.org/10.3390/brainsci11050535

[3] [Jeong2024] Domain-specialized LLM: Financial fine-tuning and utilization method using Mistral 7B. Journal of Intelligence and Information Systems, Jeong, C. (2024). Korea Intelligent Information System Society. 2024

[4] [Ravuru2024] Agentic Retrieval-Augmented Generation for Time Series Analysis. Chidaksh Ravuru, Sagar Srinivas Sakhinana, and Venkataramana Runkana. arXiv. 2024.