2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC) | 979-8-3503-6013-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCWC60891.2024.10427646

State-Based Dynamic Graph with Breadth First Progression For Autonomous Robots

Tushar Chugh*, Kanishka Tyagi†, Pranesh Srinivasan‡ and Jeshwanth Challagundla§
School of Computer Science, Carnegie Mellon University, Pittsburgh, USA
Email: *tusharchugh19@gmail.com
Department of Electrical Engineering, The University of Texas, Arlington, Texas
Email: †kanishka.tyagi@mavs.uta.edu,
Google, Mountain View, USA
Email: ‡spranesh@gmail.com, §jeshwanth.challagundla@mavs.uta.edu

***Abstract*—This paper introduces a novel method for enhancing robotic systems using Large Language Models (LLMs). We focus on leveraging LLMs to substantially improve robots' decision-making and interaction with their environment. Our proposed framework employs an agent-based approach, where robots utilize LLMs for sophisticated pattern recognition, environmental understanding, and autonomous decision-making. The main innovation of this research is the integration of LLMs into a robotic system, enabling robots to process large volumes of unstructured data, recognize complex patterns, and make informed decisions with increased precision and efficiency. This integration marks a significant leap in robotic cognitive abilities, surpassing the constraints of traditional programming. Our methodology transforms LLMs into dynamic elements within robotic systems, fostering enhanced and responsive interactions with the environment. Robots equipped with LLMs are thus capable of advanced autonomous operations, including navigating complex environments, solving intricate problems, and interacting more naturally with humans. The primary contribution of this work is the creation of an agent-based graph framework, designed to facilitate collaborative problem-solving in robotic systems. This framework consists of multiple agents working collaboratively, spanning from data ingestion to comprehensive world understanding and decision-making. These agents include modules responsible for various operational aspects, such as environmental analysis, data processing, and specialized LLMs for data interpretation and summarization. Positioning LLMs as proactive, inquisitive agents, our approach enables them to actively seek information and efficiently collaborate with other agents to complete tasks. The dynamic nature of this graph search and inter-agent communication model is a considerable innovation in robotics, offering a more integrated and effective approach for robots to tackle complex tasks, thereby enhancing their ability to operate autonomously and intelligently in diverse environments.**

***Keywords*—large Language Models, agent-based approach, autonomous operations, graph framework, mixture of experts.**

I. Introduction

In the rapidly evolving field of robotics, the implementation of data-driven decision-making is becoming increasingly essential. With industries across the globe transitioning into the digital era, they confront the significant challenge of navigating through vast quantities of structured and unstructured data. The complexity of extracting valuable insights from multimodal data sources is a critical barrier in this regard.

Enter Large Language Models (LLMs) based on transformer architectures [1], [2], which are revolutionizing paradigms across various domains, including robotics. These models, notable for their expansive parameters, excel in identifying subtle patterns and contextual information in data, often surpassing traditional analytical methods. LLMs find applications ranging from content analysis to complex tasks such as trend prediction and anomaly detection. In robotics, they are invaluable for tasks such as environmental understanding and decision-making.

However, the integration of LLMs into robotics is not devoid of challenges. A primary concern is the potential for these models to generate inaccurate or 'hallucinated' information. Additionally, applying them in complex, multi-step robotic procedures presents significant difficulties.

This paper proposes an innovative approach to augment the capabilities of LLMs within robotics. Central to this approach is an agent-based graph where multiple agents, including LLMs, collaborate to address challenges specific to robotics. These agents range from a data source providing accurate information to Python libraries for data visualization, or even other LLMs for data processing and summarization. This methodology positions LLMs as proactive, inquisitive entities within robotic systems, seeking clarity and collaborating with other agents for effective task execution. This dynamic inter-agent communication framework promises a more comprehensive and innovative approach to solving complex robotic tasks. [3] discusses the usage of LLMs in databases and it's pros and cons for research and education.

Additionally, the utilization of LLMs in robotic systems parallels the Mixture of Experts concept, where each agent specializes in a particular aspect of robotic operation. This paper further explores this connection by establishing a mathematical framework that integrates the mixture of experts paradigm with LLMs in robotics. This effort in combining LLMs with mixture of experts concepts signifies a significant advancement in the field of robotics, paving the way for more intelligent and autonomous robotic systems.

II. Framework

In the core framework of this research paper, we present an integrated assembly of intelligent agents specifically designed to collaboratively execute tasks and respond to the unique requirements of a personal robot in post-workout care. These agents, while versatile, are endowed with specialized capabilities tailored to the task given to the robot. The primary agent is the Principal Supervisor that always runs and it can spin off other agents to accomplish the tasks. Below are more details about the principal supervisor and the example of other agents:

1) **Principal Supervisor**: This overarching agent is responsible for orchestrating the activities of the other agents and ensuring seamless interactions among them. It dynamically coordinates the task flow, adapting in real-time to the user's immediate needs and the available resources.
2) **Sensing-Based Information Gatherer** : Specialized in utilizing the robot's sensory data to identify available snacks and drinks at home, this agent ensures that the most up-to-date and relevant inventory information is always used in decision-making.
3) **Online Research Agent**: This agent is tasked with retrieving and analyzing online information through search for a given task. Retrieval Augmented Learning(RAG) is based on this concept [4] [5].
   User Preference Analyst:
   a) Profile Data Reviewer: Focuses on examining the user's preferences.
   b) Feedback Processor: Analyzes feedback from previous interactions to refine future recommendations, improving personalization over time.
4) **Decision-Making Agent**: Integrates insights from the other agents to select the most suitable choices based on the given preferences, data and the task.
5) **Path Planning Coordinator**: Plans and executes the optimal path for the robot to complete the task. This agent ensures efficient navigation and handling within the given environment.
   To better understand this consider a request made from the user to the robot: *"I've just finished exercising. Could you please get me a drink and a snack to help me recover?"*

As an example, the framework could spin off these agents to accomplish the task. Each of these tasks has an objective and a method that they follow:

1) **Information Search by Perception**: *Objective*: Identify available snacks and drinks at home.
   Method: The robot uses its perception data (like cameras, sensors) to scan the home environment. It cross-references this real-time data with the previously stored inventory data in its database to determine what items are currently available.
2) **Online Information Retrieval**: *Objective*: Gather information on the health benefits and nutritional value of the available items.
   Method: The robot accesses online databases and nutrition websites to retrieve up-to-date information about the healthiness of each available food item. This step ensures that the recommendations are based on the latest nutritional research and data.
3) **User Preference and Allergy Considerations**: *Objective*: Take into account the user's preferences and any allergy considerations.
   Method: The robot consults the user's profile, which contains information about dietary preferences, allergies, and past choices. This helps in tailoring the snack and drink selection to the user's specific health requirements and taste preferences.
4) **Decision Making:** *Objective*: Select the most suitable snack and drink for the user.
   Method: Based on the information gathered from the previous steps, the robot uses its decision-making algorithms to choose the best snack and drink. This decision is based on nutritional value, user preferences, and any specific post-workout recovery needs identified from the workout data.
5) **Planning the Path and Delivery:** *Objective* : Efficiently retrieve and deliver the selected items to the user.
   Method: The robot plans the optimal path to navigate the home environment, considering obstacles and the shortest route. It then retrieves the selected snack and drink and delivers them to the user. This step involves physical navigation and manipulation skills to ensure safe and efficient delivery.

   Each of these subtasks leverages the robot's capabilities in perception, data processing, user interaction, and physical navigation, ensuring a personalized and efficient post-workout care experience.

III. State based dynamic graph with breadth first progression

We introduce the State-Based Dynamic Graph with Breadth-First Progression, a framework centered around the Principal Supervisor. This system exemplifies advanced coordination and decision-making in robotic tasks. It dynamically adapts to real-time feedback and operational states to efficiently manage and deploy a range of specialized agents for comprehensive task execution. The Principal Supervisor functions as an asynchronous coordinator for auxiliary agents, with its decision-making process directly linked to its evolving state, which changes based on feedback and outputs from subordinate agents.

A. Dynamic Agent Operation Framework

The core of this system is the Principal Supervisor, functioning as an asynchronous coordinator for auxiliary agents. Its decision-making process is directly linked to its evolving state, which changes based on feedback and outputs from subordinate agents. Starting with the initial task of providing post-workout care, the Principal Supervisor strategically invokes specific agents to gather necessary information and execute tasks. This includes agents for perception-based information gathering, online nutrition research, user preference analysis, decision-making, and path planning. The agents to be engaged are selected and ranked based on their relevance and efficacy for the task, ensuring an optimized and iterative approach until the objective is achieved.

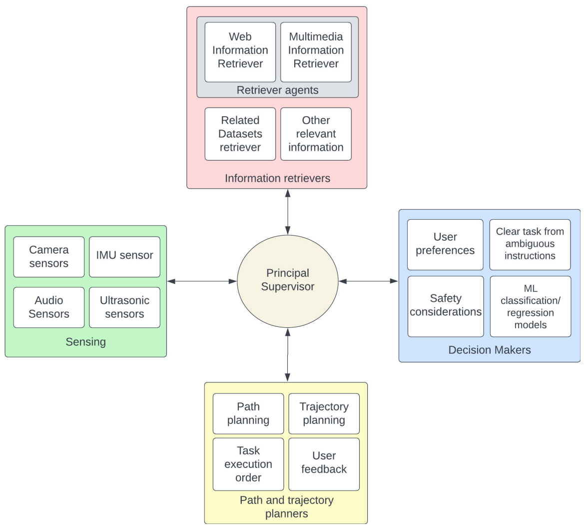

Fig. 1. Proposed algorithms for LLM: each agent works in a breadth first search approach to perform each tasks

B. Example Execution: Providing Post-Workout Care

1) Part 1: Initial Data Gathering and Analysis:

1) Principal Supervisor Initiation: Receives the task of providing post-workout care, including snack and drink selection.
2) Perception-Based Information Gathering: Activates to scan and identify available items using sensory data.
3) Online Nutrition Research Agent: Invoked to gather health and nutritional information about the available items.
4) User Preference Analysis: Engages to review user profile data for preferences and allergies.

2) Part 2: Decision-Making and Path Planning:

1) Decision-Making: With the gathered data, the Principal Supervisor analyzes and selects the most suitable snack and drink.
2) Path Planning Agent: Plans the optimal route for item retrieval and delivery within the home environment.

3) Part 3: Execution and Delivery:

1) Execution of Retrieval: The principal supervisor commands the robot to retrieve the chosen items efficiently.
2) Delivery: Ensures the safe and timely delivery of the items to the user.
3) Feedback Integration: Collects user feedback post-delivery to refine future recommendations.

In this adapted methodology, the principal supervisor dynamically coordinates a team of specialized agents, each playing a pivotal role in ensuring the robot effectively provides personalized post-workout care. This approach showcases the potential of a state-based dynamic graph and breadth-first progression in managing complex, multifaceted tasks within the realm of robotics.

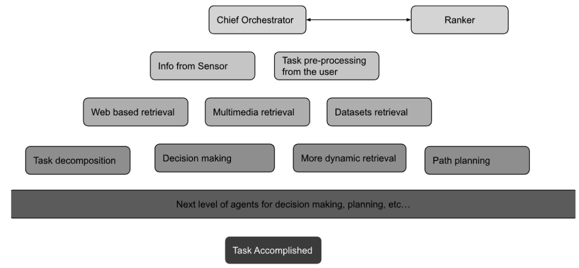

Fig. 2. Flow diagram to visualize the dynamic graph

IV. Mixture of Expert Agents

The mixture of local experts model, as described in [6], functions as a successful algorithm that employs a divide-and-conquer approach rooted in conditional computation. This model comprises a gating model, multiple expert models, and a training methodology geared towards learning these models' parameters. The proposed collaborative LLM agent approach closely resembles the Mixture of experts model, which we will call as Mixture of expert agents (MoEA model). In the proposed MoEA model Each expert LLM agent specializes in a specific section of the input space which includes different subtasks like sensing, retrieval etc. The principal supervisor can be approximated as the trainable probabilistic Gating Network (GN) assigning regions to different expert agents.

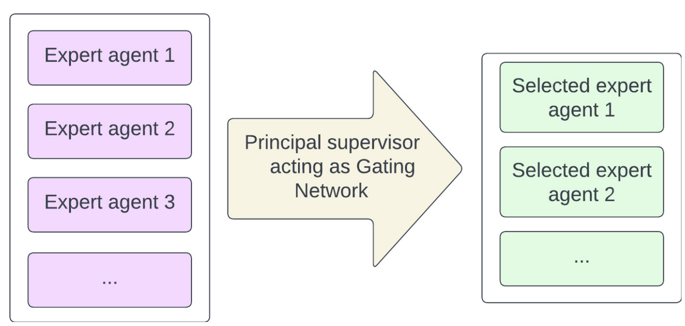

Fig. 3. Mixture-of-expert agents architecture.

A. Mathematical Framework

Enhanced performance in machine learning is often achievable by integrating multiple models, an approach encapsulated in the concept of *committee machines* [6], [7]. This method involves either averaging the predictions from a set of models, known as *boosting*, or selecting a single model for making predictions, akin to *decision trees*. When implemented in a probabilistic setting, this approach facilitates a more nuanced decision process through the distribution of input space among various models.

Consider an input vector $\mathbf{x}$, a target $\mathbf{t}$, and a parameter set $\boldsymbol{G} = \{\boldsymbol{G}_g, \boldsymbol{G}_e\}$, where G_g and G_e are the gating and expert parameters, respectively. Within this framework, a group of N experts forms a mixture model that yields the output $\mathbf{t}$ with a probability $P(\mathbf{t}|\mathbf{x}, \boldsymbol{G})$, represented as:

$$P(\mathbf{t}|\mathbf{x}, \boldsymbol{G}) = \sum_{n=1}^{N} P(t, n|\mathbf{x}, \boldsymbol{G}) = \sum_{n=1}^{N} P(n|\mathbf{x}, \boldsymbol{G}_g) P(\mathbf{t}|n, \mathbf{x}, \boldsymbol{G}_e). \tag{1}$$

Equation 1 delineates the mixture distribution, which may be either discrete or continuous. A basic MoEA architecture, illustrated in Figure 3, comprises N trainable experts. The outputs from these experts are weighted by the gating network and aggregated by the pooling system. This architecture is trained so that each expert models a specific aspect of the mixture model, while the gating system determines the mixing parameters $P(i|\mathbf{x}, \boldsymbol{G}_g)$.

The primary objective in training the MoEA architecture is to optimize the parameters that maximize the log-likelihood of a given dataset D, comprising N_v training patterns $\mathbf{x}_1, \ldots, \mathbf{x}_{N_v}$. The log-likelihood function $l(D, \boldsymbol{G})$ is formulated as:

$$l(D, \boldsymbol{G}) = \sum_{i=1}^{N_v} \ln \left(\sum_{n=1}^{N} P(n|\mathbf{x}, \boldsymbol{G}_g) P(\mathbf{t}|n, \mathbf{x}, \boldsymbol{G}_e) \right). \tag{2}$$

This function is maximized, as discussed in [8], under the assumption of a Gaussian mixture model, where each expert is responsible for modeling a component of the mixture.

B. Training of Mixture of Expert Agents

The training mechanism of a Mixture of Expert Agents (MoEA) represents a sophisticated and structured learning approach, where the Principal Supervisor, often an advanced Large Language Model (LLM) specifically tailored through initial prompt engineering [9] [10] [11], plays a central role. This supervisor's primary responsibility lies in accurately determining the most suitable expert LLM for a given task, taking into account the specific context of the input. This selection process is iterative and dynamic. At each step of a task, the supervisor assesses the present situation and selects an expert LLM best suited to address that particular phase of the task. Following the selection, the chosen expert LLM executes its assigned role, and the results of this action are fed back to the supervisor. This continuous feedback loop is critical as it allows the supervisor to make well-informed decisions regarding the deployment of expert agents in future steps.

A notable feature of this training method is the implementation of a high-temperature setting when operating the supervisor LLM repetitively. Within the realm of LLMs, high temperature typically translates to outputs that are less predictable and more diverse [12]. Such a setting is invaluable for probing a broad array of potential solutions and strategies the model might adopt. The system generates numerous pathways using a breadth-first strategy. These pathways are then meticulously evaluated based on their effectiveness in accomplishing the given task.

To promote and reinforce effective strategies, a reward-based system is integrated into the training process. Pathways that successfully complete tasks are acknowledged with positive rewards. Importantly, the system is designed to favor efficiency – pathways achieving task completion in fewer steps are rewarded more generously than those requiring more steps. This approach encourages the model not merely to find solutions but to seek the most resource-efficient solutions. Such an incentive structure fosters the development of answers that are accurate and also resource-conscious.

Training extends to both the supervisor LLM and the expert agent LLMs, guided by a reinforcement learning framework [13] where rewards play a pivotal role in shaping behavior. Over time, this training regimen refines the supervisor's ability to judiciously select the appropriate expert for each task, based on timing and requirements. This gradual refinement enhances the overall performance and efficiency of the MoEA system. The ultimate objective is to cultivate a robust, adaptive model capable of skillfully navigating a variety of tasks, effectively utilizing the specialized expertise of different agents to achieve optimal outcomes.

V. Conclusion

In our upcoming endeavors, our primary focus will revolve around the expansion and fine-tuning of state-based dynamic graph methodologies, aiming to open up fresh horizons in the realm of robotic adaptability and intelligence. Several promising directions await exploration. One avenue involves applying state-based dynamic graph techniques to intricate robotic systems like collaborative robots or self-driving vehicles. This application aims to bolster real-time decision-making in dynamic settings, enhancing their capabilities significantly. Another area of interest pertains to devising training algorithms for MoEA (Mixture of Expert Agents) within LLMs (Large Language Models) through the utilization of state-based dynamic graphs. This development aims to refine the decision-making processes in robots by factoring in environmental states and task requirements. Furthermore, implementing adaptive expert selection mechanisms using graph models in robotics represents an intriguing endeavor. This approach empowers robots to judiciously select the most pertinent expert(s) for each state or task, thereby optimizing their performance. Moreover, we are delving into the realm of automatic expert discovery and integration within MoEA frameworks. By leveraging state-based graphs, this research aims to equip robots with the ability to autonomously discover and seamlessly integrate new experts, enhancing their capabilities progressively. Additionally, a key focus area involves addressing forgetfulness in MoEA models. We aim to strategize approaches that mitigate forgetting in these models, ensuring sustained robotic learning and adaptation over extended periods through the application of state-based dynamic graph methodologies. By concentrating efforts on these pivotal areas, our future research endeavors aim to propel substantial advancements in robotics, fostering the evolution of more intelligent and adaptable robotic systems.

REFERENCES

[1] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al., "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] Sihem Amer-Yahia, Angela Bonifati, Lei Chen, Guoliang Li, Kyuseok Shim, Jianliang Xu, and Xiaochun Yang, "From large language models to databases and back: A discussion on research and education," *arXiv preprint arXiv:2306.01388*, 2023.

[4] Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and Ji-Rong Wen, "Reta-llm: A retrieval-augmented large language model toolkit," *2306.05212*, 2023.

[5] Retrieval-Augmented Generation for Large Language Models: A Survey, "Yunfan gao, yun xiong, xinyu gao, kangxiang jia, jinliu pan, yuxi bi, yi dai, jiawei sun, haofen wang," *2312.10997*, 2023.

[6] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[7] Kanishka Tyagi, Chinmay Rane, and Michael Manry, "Supervised learning," in *Artificial Intelligence and Machine Learning for EDGE Computing*, pp. 3–22. Elsevier, 2022.

[8] Matthew N Dailey and Garrison W Cottrell, "Organization of face and object recognition in modular neural network models," *Neural networks*, vol. 12, no. 7, pp. 1053–1074, 1999.

[9] Deepak Pathak Igor Mordatch Wenlong Huang, Pieter Abbeel, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," *arXiv preprint arXiv:2201.07207*, 2022.

[10] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu, "Unleashing the potential of prompt engineering in large language models: a comprehensive review," *arXiv:2310.14735*, 2023.

[11] Prompting Frameworks for Large Language Models: A Survey, "Xiaoxia liu, jingyi wang, jun sun, xiaohan yuan, guoliang dong, peng di, wenhai wang, dongxia wang," *arXiv:2311.12785*, 2023.

[12] Shuyin Ouyang, Jie M. Zhang, Mark Harman, and Meng Wang, "Llm is like a box of chocolates: the non-determinism of chatgpt in code generation," *arXiv:2308.02828*, 2023.

[13] Hao Sun, "Reinforcement learning in the era of llms: What is essential? what is needed? an rl perspective on rlhf, prompting, and beyond," *arXiv:2310.06147*, 2023.