1 Introduction

This paper[1] studies the Distributed Mixture-of-Agents (MoA) framework for collaborative inference with Large Language Models (LLMs) on edge devices. The MoA paradigm enables multiple LLMs, each hosted on separate, user-associated edge devices, to work together by exchanging semantic information, thereby enhancing response quality beyond what any single LLM can achieve. The system employs a structure of multiple proposer models and an aggregator model: proposers generate candidate responses to user prompts, while the aggregator synthesizes these into a refined, high-quality answer.

In this paper, they considered 5 different LLMs to work together on a particular problem. They randomly select k devices out of n-1 to transmit the device for inference. When the response from K LLMs returns, then the original LLM acts as an aggregator to synthesize a refined output.

In this paper, they have talked about two specific experiments: (1) all devices use the same LLM, and (2) devices use a diverse set of LLMs. Results show that increasing the number of layers (M) and proposers (k) improves accuracy but also increases latency and queue size, illustrating a clear trade-off. For instance, with Qwen-1.5-72B, accuracy rises from 21.91% (no MoA) to 40.63% (M=2, k=3), but latency and queue size also grow. Using various LLMs further boosts the accuracy, reaching 55.55% with M = 2 and k = 3, but again at the cost of higher latency and queue size.

So, the main focus of this paper was –

- theoretical calculation of queuing stability;
- leveraging open-source LLMs for distributed MoA

2 Contribution of the Paper

Distributed Mixture-of-Agents is a promising architecture for using the collective intelligence of several LLMs directly at the edge devices. It provides a possible route to obtain high-performance inference while keeping decentralisation, improving robustness, and protecting data privacy. The work studied here shows the potential of such a system utilising gossip protocols and examines its basic capacity constraints using queuing theory, hence providing a significant first step.

The first paper to consider both semantic gossips and timeliness.

Distributed Edge Setting: The entire system operates across edge devices without reliance on a central server or cloud.

Decentralised Gossip Communication: It employs gossip algorithms for peer-to-peer information exchange, emphasising robustness and decentralisation.

Queuing Stability Analysis: It provides a formal analysis of the system's capacity and stability under the constraints of edge processing and the chosen MoA configuration.

Experimental evidence : It provides an interesting result stating that partitioning one large LLM across multiple resources (edge-edge or edge-cloud) using techniques like pipeline parallelism results in decent results.

3 Limitatons

Future research must address the critical limitations identified, primarily by integrating intelligent agent coordination strategies into the distributed edge framework. Developing lightweight, adaptive agent selection (routing) mechanisms that consider context (query, agent capabilities, system state) and exploring dynamic task specialization strategies are paramount.

Simplistic Agent Selection: The reliance on uniform random selection of proposer LLMs is a major limitation. This approach is inefficient and non-adaptive. It fails to consider agent capabilities (expertise), task requirements, or the current system state.

Lack of Task Specialization: The framework assumes generalist agents tied to users, acting as proposers or aggregators based purely on context. This overlooks the potential efficiency and accuracy benefits of task specialization, where agents could focus on specific domains or roles.

4 Related Work

The **Mixture of Agents (MoA)**[2] framework is inspired by the concept of **Mixture of Experts (MoE)**[3]. In MoE, multiple expert networks each specializing in different skill sets are selectively activated in each layer using a gating function. However, MoE architectures often suffer from high computational overhead and limited flexibility and scalability.

To address these challenges, the **Mixture of Agents (MoA)** architecture was introduced. Unlike Mixture of Experts (MoE), MoA leverages large language models (LLMs) that operate entirely through prompt interfaces, eliminating the need to modify internal activations or model weights. This design enhances flexibility and scalability.

However, the MoA framework still suffers from a high Time to First Token (TTFT), leading to latency issues during inference.

To mitigate this, the **Sparse Mixture-of-Agents (SMoA)**[4] framework was proposed. SMoA introduces sparsity in agent interactions through two key components:

- **The Judge** LLM, which selectively identifies and promotes high-quality responses for progression to subsequent rounds.

- **The Moderator** LLM, which manages the flow of information and decides when to terminate the interaction process.

Additionally, SMoA utilizes a function called **Aggrole(D, T, n)** to generate diverse, role-specific prompts. Each prompt establishes a distinct ”persona,” enabling the system to incorporate multiple perspectives into the final output while maintaining efficiency.

Some work is also focus on evolving the structural organization of agent interactions like

MacNET[5] : A Multi-Agent Collaborative Network that organizes multiple agents into a Directed Acyclic Graph (DAG). Agents interact in topologically sorted order, collaboratively and iteratively refining partial solutions at each step. The architecture is highly scalable supporting a seamless transition from just a few agents to over 1,000 without requiring any additional training.

GPTSWARM[6] :Described language agent system through graph representation. Here each node is dedicated to a specific task. Swarm graphs contain subgraphs representing agents. Apart from edge optimization, this framework allows each node in the graph to self-improve by adapting its prompts based on previous input and task feedback.

5 Research Idea

To design a more robust and intelligent distributed Mixture-of-Agents (MoA) system at the edge, this research proposes a hybrid framework that combines three complementary mechanisms: specialization, intelligent selection, and graph-based orchestration.

First, selecting or training individual edge agents to develop domain-specific expertise, forming a pool of specialized LLM-based agents. Next, selection strategies like Sparse Mixture-of-Agents (SMoA) are used to dynamically identify the most suitable neighbors for collaboration, based on task relevance and historical performance. Finally, the entire system is structured as a dynamic computational graph using a GPTSWARM-like architecture, where agents (nodes) interact through optimized pathways (edges). These edge connections are continuously refined to maximize task-specific utility, allowing the system to learn optimal collaboration strategies over time.

This integrated approach leverages the strengths of all three avenues to create a scalable, adaptive, and efficient framework for multi-agent coordination in edge environments.

References

[1] P. Mitra, P. Kaswan, and S. Ulukus, “Distributed mixture-of-agents for edge inference with large language models,” 2024.

[2] J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, and J. Zou, “Mixture-of-agents enhances large language model capabilities,” *arXiv preprint arXiv:2406.04692*, 2024.

[3] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” 2017.

[4] D. Li, Z. Tan, P. Qian, Y. Li, K. S. Chaudhary, L. Hu, and J. Shen, “Smoa: Improving multi-agent large language models with sparse mixture-of-agents,” 2024.

[5] C. Qian, Z. Xie, Y. Wang, W. Liu, K. Zhu, H. Xia, Y. Dang, Z. Du, W. Chen, C. Yang, Z. Liu, and M. Sun, “Scaling large language model-based multi-agent collaboration,” 2025.

[6] M. Zhuge, W. Wang, L. Kirsch, F. Faccio, D. Khizbullin, and J. Schmidhuber, “Gptswarm: Language agents as optimizable graphs,” in *Forty-first International Conference on Machine Learning*, 2024.