The details of the architecture in *Exploring Multi-Agent Dynamics for Generative AI and Large Language Models in Mobile Edge Networks* are as follows:

(1) *System Evaluation:* In mobile edge networks, when a user send a request to the edge server, the system performs a comprehensive evaluation to assess the current network conditions and predict task execution feasibility as well as efficiency. Specifically, when a user generates a specific request, it sends the observed data, along with the detailed request, to the edge server based on a particular protocol, such as user datagram protocol (UDP) [R1] and micro air vehicle link (MAVLink) [R2]. Once upon a request is received from the user, the edge server evaluates the broader network environment in response to the requirement of the user. *For instance, when a UAV generates a demand for post-disaster ground user data acquisition, the UAV would send its location, battery life, velocity, the proximity of ground users, other relevant factors along with the requirement to the edge server, while the edge server evaluates the available bandwidth, channel condition, and signal strength in the post-disaster area.*

(2) *Data Integration, Analysis, and Normalization:* The GenAI-LLM interface module in the mobile edge server aggregates and formalizes the data required by the GenAI-LLM.

First, the GenAI-LLM interface module retrieves the user-submitted data from the message queue, including observations, requests, and network environment data from other server modules.

Second, this module employs natural language processing (NLP) and Transformer techniques to analyze user requests and extract key information [R3]. *For instance, suppose that the request text is "The data from devices in post-disaster area A needs to be collected urgently! Prioritize searching for data from devices in the southeast corner! Please plan the optimal route!", and the text of data collected by the UAV is as follows:*

```
{
  "UAV_id": "UAV001",
  "timestamp": "2024-01-19T17:30:00Z",
  "location": {"latitude": 37.7749, "longitude": -122.4194, "altitude": 100},
  "battery_level": 75%,
  "maximum_speed": 40 m/s,
  "minimum_speed": 2 m/s,
  "X-axis range of region A": [0, 1000],
  "Y-axis range of region A": [0, 1000],
  "the number of ground user": 100,
  "location of ground users": ...,
}.
```

In such case, key information, such as the target area and velocity constraints, can be obtained.

Third, sentiment analysis tools like TextBlob and VADER (valence aware dictionary and sEntiment reasoner) are also employed [R4] [R5]. *For example, by analyzing the request above, VADER might detect an urgent tone, leading the system to prioritize task completion time in the data provided to the GenAI-LLM.* This nuanced approach ensures more comprehensive data analysis.

Finally, NLP techniques are used to aggregate and polish all the data analysis. This data is then structured to provide the GenAI-LLM with a comprehensive understanding of user requirements for formulating optimization problems.

(3) *Metric Verification and Optimization Problem Formulation:* Based on the structured data, GenAI-LLM identifies the necessary metrics and constructs the optimization problem.

First, the GenAI-LLM interface module regards the structured data as a prompt and uses few-shot and chain-of-thought promptings to enrich the structured data, mitigating the errors and hallucinations in the output of the GenAI-LLM. *For instance, the details of an enriched prompt is as follows:*

```
{
  "instruction": "output the optimization objectives, constraints, and decision
      variables of an optimization problem based on the following data",
  "user request": {
        "content": "Please plan an optimal trajectory to collect data from
            devices in post-disaster area A, prioritizing the southeast corner",
        "timestamp": "2024-01-19T17:30:00Z",
        "emergency degree": "high",
        ...
  },
  "user data":{
        "UAV_id": "UAV001",
        "battery_level": 75%,
        "maximum_speed": 40 m/s,
        "minimum_speed": 2 m/s,
        "location": {"latitude": 37.7749, "longitude": -122.4194, "altitude":
            100},
        ...
  },
  "environment data":{
        "bandwidth": 20E6,
        "X-axis range of region A": [0, 1000],
        "Y-axis range of region A": [0, 1000],
        "channel condition": "line-of-sight",
        "the number of ground users": 100,
        "location of ground users": ...,
        ...
  },
  "expected output": {
        "content": "the optimization objectives, constraints, and decision
            variables of an optimization problem",
        "reference format":{
                "objectives functions": "f1, the transmission rate; ...",
                "constraints": "c1, the velocity constraint; ...",
                "decision variables": "d1, the trajectory; ...",
        }
  },
  "reference examples": {
        "example 1": {
          "prompt": {
            "instruction": "output the optimization objectives, constraints, and
                decision variables of an optimization problem based on the
                following data",
            "user request": {
              "content": "Plan an optimal trajectory for a UAV to collect data
                  from devices in a disaster area, prioritizing the southeast
                  corner.",
              "emergency degree": "high"
            },
            "user data": {
              "UAV_id": "UAV002",
              "battery_level": 90 %,
              "maximum_speed": 50 m/s,
              "minimum_speed": 5 m/s,
              "location": {"latitude": 34.0522, "longitude": -118.2437, "altitude
```

```
                  ":150 },
                ...
              },
              "environment data": {
                "bandwidth": 15E6,
                "X-axis range of region A": [0,2000],
                "Y-axis range of region A": [0,2000],
                "channel condition": "line-of-sight",
                "the number of ground users": 50,
                "location of ground users": [
                  {"latitude": 34.06, "longitude": -118.25},
                  {"latitude": 34.07, "longitude": -118.26},
                  // ... more user locations
                  {"latitude": 34.09, "longitude": -118.28}
                ]
              }
            },
            "output": {
              "objectives functions": "f1, Maximize the total data collected from
                  all devices; f2, Minimize the flight time.",
              "constraints": "c1, The UAV's speed must be between 5 m/s and 50 m/s;
                   c2, The UAV's battery level must remain above a certain threshold
                   (e.g., 20%); c3, The trajectory must stay within the boundaries
                  of the disaster area; c4, Prioritize visiting devices in the
                  southeast corner.",
              "decision variables": "d1, The sequence of locations (waypoints) the
                  UAV will visit; d2, The time spent at each location."
            }
          },
          "example 2": {
                  "prompt": { ...
                  },
                  "output": { ...
                  }
          },
          ...
  },
  "suggested reasoning steps": {
          1: "Identify the scenario for the optimization problem."
          2: "Deduce optimization objectives and decision variables by analyzing
              user requirements."
          3: "Derive constraints by analyzing user and environment data."
          4: "Generate output based on the provided format."
  },
}.
```

Apparently, the model leverages both few-shot prompting, by providing specific reference examples, and chain-of-thought prompting, by attaching suggested reasoning steps.

Second, the enriched prompt is fed to the GenAI-LLM, which, after processing, outputs the necessary metrics such as the optimization objectives, decision variables, and constraints of the optimization problem with the assistance of retrieval-augmented generation (RAG) [R6]. *For instance, the output of GenAI-LLM can be as follows:*

$$\text{optimization objectives: } f_1 : \text{the total transmission rate; } f_2 : \text{the data collection time;}$$
$$f_3 : \text{the energy consumption,}$$

decision variables: $\mathbf{q}_\mathrm{u}$: the trajectory of the UAV; $\mathbf{v}_\mathrm{u}$: the velocity of the UAV,

constraints: $C1$: the location constraint of UAV; $C2$: the velocity constraint of UAV; ...

Note that this is a just simplified example, and the actual output is more structured and specific.

Third, the GenAI-LLM interface module generates a new prompt, assisting the GenAI-LLM finishes the problem formulation. Similarly, few-shot and chain-of-thought prompting are also used to enrich the prompt.

Finally, the enriched prompt is input into the GenAI-LLM and the model formulates an optimization problem accordingly. *For instance, based on the aforementioned process, the output optimization problem is as follows:*

$$\begin{aligned}
\min_{\{\mathbf{q}_\mathrm{u},\mathbf{v}_\mathrm{u}\}} \ & F = \{-f_1, f_2, f_3\}, && \text{(1a)}\\
\text{s.t. } & C1: 0 \le x_\mathrm{u}[t] \le 1000, 0 \le t \in T, && \text{(1b)}\\
& C2: 0 \le y_\mathrm{u}[t] \le 1000, 0 \le t \in T, && \text{(1c)}\\
& C3: 2 \le v_\mathrm{u}[t] \le 40, 0 \le t \in T, && \text{(1d)}\\
& C4: \mathbf{q}_\mathrm{u}[0] = \mathbf{q}_\mathrm{start}, \mathbf{q}_\mathrm{u}[T] = \mathbf{q}_\mathrm{end}, && \text{(1e)}\\
& \dots
\end{aligned}$$

(4) *Optimization Problem Solving:* The process for solving the formulated problem are divided as follows:

First, the optimization problem is decomposed into subproblems based on the intrinsic characteristics via common decomposition methods, which are objective-based, variable-based, and space-based decompositions, among other strategies. *For instance, the multi-objective optimization problem in Eq.* (1) *can be decomposed based on decision variables, yielding subproblems for trajectory optimization (*$\mathbf{q}_\mathrm{u}$*) and velocity optimization (*$\mathbf{v}_\mathrm{u}$*).*

Second, the interaction paradigm is selected among multiple agents based on the dominant properties of the formulated problem. *For example, "efficiency-driven" problems, characterized by the limited time and other factors associate with efficiency, are best addressed with a cooperative paradigm where agents collaborate to find a globally optimal solution.*

Third, the interaction structure among multiple agents is determined based on the features of the optimization problem. *For instance, a message shared structure is preferred when a multi-agent GenAI-LLMs system use the block coodinate descent (BCD) to solve problems [R7]. In such cases, the first agent updates a block (i.e., a part of decision variables) and puts it into the repository. The second agent updates the second block based on the first block in the shared repository and stores it there. Subsequently, the third agent retrieves the results of the first two blocks, updates the third block, and so on, until all agents have completed their respective update tasks.*

Fourth, the interaction strategy is chosen based on the selected interaction paradigm, structure and the trade-off between resource consumption and performance. *For instance, with the aforementioned cooperative paradigm and message shared structure, message passing strategy is the most appropriate since agents can collaborate by publishing messages (such as the updated decision variable) in a shared space, while other agents can then read these messages, update their own data. Moreover, direct communication might be preferred if high interaction performance is critical, since it can minimize delay and reduce overhead.*

Fifth, subproblems are automatically assigned to specialized agents for solution based on characteristics such as dimensionality, constraints, and objective function properties, ensuring optimal agent-task matching. *For instance, the trajectory optimization ($\mathbf{q}_{\mathrm{u}}$) subproblem is assigned to an agent equipped with deep reinforcement learning (DRL), while the velocity optimization ($\mathbf{v}_{\mathrm{u}}$) subproblem is assigned to an agent with swarm intelligence algorithms. In this way, the former iteratively optimizes $\mathbf{q}_{\mathrm{u}}$ based on the current $\mathbf{v}_{\mathrm{u}}$ from the repository, and the latter optimizes the $\mathbf{v}_{\mathrm{u}}$ while keeping the trajectory $\mathbf{q}_{\mathrm{u}}$ fixed.*

Finally, if the latency approaches a predefined threshold (e.g., 90% of the maximum) before interaction completes, each agent submits its partially content to a central point for consistency checks, conflict resolution, and aggregation before forwarding to the GenAI-LLM. Otherwise, if latency constraints are not violated, the agents continue interacting until a complete solution is obtained, and the consolidated content is then forwarded GenAI-LLM. Subsequently, the content is then reviewed by both the GenAI-LLM and human, if any errors or hallucinations are found, the content is regenerated. Otherwise, it is sent to the user.

(5) *Continuous Improvement:* Successful strategies and learned behaviors are shared among agents to enhance the overall system performance. *For instance, if an agent discovers a particularly efficient UAV trajectory for a specific user density, this pattern, along with its associated context (e.g., user location), is stored in a repository to share with other agents.* Moreover, the user provides a feedback to the server, rating both the satisfaction with the proposed solution and its feasibility as well as efficiency. GenAI-LLM then regards the user feedback as a reward signal and uses DRL techniques, such as policy gradients and value function approximation [R8], to update its parameters. These updates fine-tune the internal neural network weights of GenAI-LLM, helping it generate contents that better align with user needs and produce more accurate results. In this way, GenAI-LLMs can be updated incrementally with each new batch of data, without requiring a complete retraining cycle.

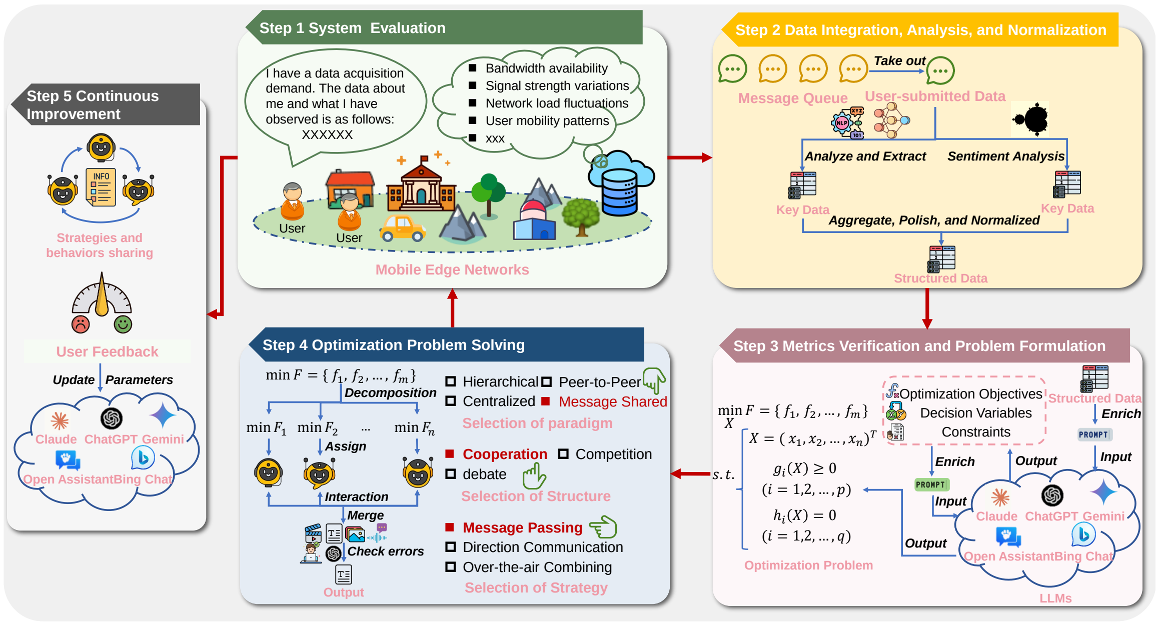

Fig. 1: The framework of the proposed architecture. In step 1, a comprehensive evaluation is conducted to assess the current network conditions, predict task execution feasibility and efficiency, and optimize accordingly. In step 2, the GenAI-LLM interface module in the mobile edge server aggregates, analyze, and normalizes the data required by the GenAI-LLM. In step 3, the GenAI-LLM constructs the optimization problem based on the structured data. In step 4, interaction paradigm, structure, and strategy are determined. Moreover, the optimization problem is decomposed into several subproblems, each of which is assigned to an agent for execution. When the interaction among multiple agents ends, the output without errors and hallucinations is forwarded to the original user. In step 5, continuous improvement is conducted via knowledge sharing among agents and GenAI-LLM will update the its parameter based on user feedback and reinforcement learning techniques.

References

[R1] Y. Gu, and L. G. Robert, ”UDT: UDP-based data transfer for high-speed wide area networks,” *Comp. Netw.*, vol. 51, no. 7, pp. 1777-1799, 2007.

[R2] S. Atoev, K. R. Kwon, S. H. Lee, and K. S. Moon, ”Data analysis of the MAVLink communication protocol,” *Proc. IEEE ICISCT*, 2017.

[R3] A. Vaswani, ”Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, 2017.

[R4] W. Aljedaani, F. Rustam, M.W. Mkaouer, A. Ghallab, V. Rupapara, P.B. Washington, E. Lee, and I. Ashraf, ”Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry,” *Knowl.-Based Syst.*, vol. 255, pp. 109780.

[R5] C. Hutto, and G. Eric, ”Vader: A parsimonious rule-based model for sentiment analysis of social media text,” *Proc. AAAI*, vol. 8. no. 1, 2014.

[R6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.T. Yih, T. Rocktäschel, and S. Riedel, ”Retrieval-augmented generation for knowledge-intensive nlp tasks,” *Adv. Neural Inf. Process. Syst.*, vol. 33, pp.9459-9474.

[R7] P. Tseng, ”Convergence of a block coordinate descent method for nondifferentiable minimization,” *J. Optim. Theory Appl.*, vol. 109, pp.475-494.

[R8] J. Peters, and S. Schaal, ”Reinforcement learning of motor skills with policy gradients,” *Neural Netw.*, vol. 21, no. 4, pp.682-697, 2008.