

# Supplementary Experiments and Discussions of

## Generative AI as a Service in 6G Edge-Cloud: Generation

### Task Offloading by In-context Learning

Hao Zhou, Chengming Hu, Dun Yuan, Ye Yuan, Di Wu, Xue Liu, Zhu Han, and Charlie Zhang

The following presents supplementary experiments and discussions of our work, including detailed input prompt design, LLM output analyses, and discussions of potential issues.

#### 1. Input Prompt Design

The textbox below shows our input prompt for the task description, including “task goal”, “task description”, and “extra rules”.

##### Prompt input for LLMs

Task goal: You need to select between two decisions: “local” and “offload”.

Task description: You have to consider the condition of each case, including 2 keywords: keyword 1: task size; keyword 2: quality\_requirement.

Following are some examples you can learn:.....

Now I will give you a new condition to solve, task\_size=middle, quality requirement=high.

Please reply “local” or “offload”. Please explain why you make this decision.

##### The replies of Llama3-8b for “offload” decisions.

Run= 0, timeslot=0

Based on the given condition:

\*task\_size = middle,

\*quality\_requirement = high

I decide to choose “offload”.

I make this decision because a high quality requirement suggests that the task *requires a high level of precision and accuracy*, which may be challenging to achieve on a local device with limited resources, Offloading the task to *a more powerful and resource-rich remote server can ensure that the task is completed* with the required level of quality. Additionally, since the task size is middle, it is not too small to be handled locally, but not too large to be handled remotely, making offloading a suitable option.

#### 2. Output Analyses of Llama3-8b and GPT-4o

Given the prompt input, the left textbox shows the decisions and replies of Llama3-8b for making “offload” decisions. It is worth noting that currently no example is provided to the LLM agent, which is indicated by “timeslot = 0”. In this case, Llama3-8b can still:

1) Correctly identify the keywords we defined in the input prompt, including “task\_size” and “quality\_requirement”, and make the decision “offload” or “local” as instructions;

2) Show a clear understanding of the task offloading concepts, e.g., “*a high quality requirement suggests that the task requires a high level of precision and accuracy*”, and considering “local devices with limited resources”, then “offloading the task to a more powerful and resource-rich remote server”.

Similarly, the following presents the replies of Llama3-8b for “local” decisions. It also demonstrates that Llama3-8b already has a clear concept of task offloading by considering “task size” and “quality requirements”, and understanding that “*task can be handled locally without compromising the quality requirement*”.

##### The replies of Llama3-8b for “local” decisions.

Based on the given condition:

task\_size=middle, quality\_requirement=middle

I decide: local.

I make this decision because the task size is middle, which is not too large or too small, and *the quality requirement is also middle*, which is not extremely high or low. In this case, it’s reasonable to assume that *the task can be handled locally without compromising the quality requirement*.

Llama3-8b is considered a small-scale model, and we also tested the reply of large-scale LLMs such as GPT-4o, which is a large-scale state-of-the-art LLM model. One can observe that GPT-4o presents a more detailed explanation in terms of “task size” and “quality requirement”. It clearly shows that GPT-4o can fully understand that “*a middle-sized task might be somehow manageable locally but could benefit from ...*”. Meanwhile, “*High quality requirements often imply a need for more computational power...*”. These explanations align very well with common knowledge in mobile edge computing.

The replies of GPT-4o for "offload" decisions.

Run= 0, timeslot=0  
Decision: "offload"

Explanation:

**\*\*Task Size = Middle\*\*:** A middle-sized task might be somewhat manageable locally but could benefit from additional computing resources when offloaded.

**\*\*Quality Requirement = High\*\*:** High-quality requirements often imply a need for more computational power and potentially specialized resources to ensure the task meets high standards.

Given that the task size is moderate and the quality requirement is high, it is generally preferable to offload the task. Offloading can provide access to more powerful and specialized computational resources, ensuring the high-quality requirement is met more efficiently and effectively than potentially straining local resources.

The above results and analyses demonstrate that LLMs already have some fundamental knowledge and understanding of task offloading and edge computing. It reveals the great potential of applying LLMs to network systems.

### 3. Potential Issues

We also find some potential issues when using LLMs. For example, we have changed the initial output rules by "Please reply 'local' or 'offload', no other words are allowed".

The unstable output of GPT-3.5 Turbo with changing reply forms.

Run= 0, timeslot= 1  
**GPT-3.5 reply:** offload

Run= 0, timeslot= 2  
**GPT-3.5 reply:** offload

Run= 0, timeslot= 3  
**GPT-3.5 reply:** Given the condition: task\_size=middle, quality \_requirement=high. Based on the examples provided, the decision would be: offload

Run= 0, timeslot= 4  
**GPT-3.5 reply:** Based on the condition provided, the decision should be 'offload'

Run= 0 timeslot= 5  
**GPT-3.5 reply:** offload

Run= 0, timeslot= 6  
**GPT-3.5 reply:** Based on the condition provided (task\_size=middle, quality\_requirement=middle),...

However, we find that GPT-3.5 Turbo will still produce some unstable output, which means the rule "no other words are allowed" is not well understood. The unstable output forms will further affect the algorithm's performance.

By contrast, both GPT-4o and Llama3-8b can follow the output instructions by replying "local" or "offload" only. This is because both LLMs are recently released models with state-of-the-art architecture and design, while GPT 3.5 was released in 2022 November as an early model.

Therefore, selecting appropriate LLMs is crucial when applying LLMs to network problems.

The stable output of GPT-4o.

Run= 0, timeslot= 1  
**GPT-4o reply:** offload

Run= 0, timeslot= 2  
**GPT-4o reply:** local

Run= 0, timeslot= 3  
**GPT-4o reply:** offload

Run= 0, timeslot= 4  
**GPT-4o reply:** local

Run= 0 timeslot= 5  
**GPT-4o reply:** local

Run= 0, timeslot= 6  
**GPT-4o reply:** offload

The stable output of Llama3-8b.

Run= 0, timeslot= 1  
**Llama3-8b reply:** offload

Run= 0, timeslot= 2  
**Llama3-8b reply:** local

Run= 0, timeslot= 3  
**Llama3-8b reply:** local

Run= 0, timeslot= 4  
**Llama3-8b reply:** offload

Run= 0 timeslot= 5  
**Llama3-8b reply:** local

Run= 0, timeslot= 6  
**Llama3-8b reply:** offload