IEEE INFOCOM 2025 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 979-8-3315-4370-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOMWKSHPS65812.2025.11152975

Towards On-Demand Metaverse Service Deployment in 6G Vehicular Networks Using Multimodal LLMs

Gordon Owusu Boateng[1], Amine Kidane Ghebreziabiher[1], Rabeb Mizouni[2], Azzam Mourad[1,3], Hadi Otrok[2], Jamal Bentahar[1,4], and Sami Muhaidat[5,6]

[1] KU 6G Research Center, Department of Computer Science, Khalifa University, Abu Dhabi, UAE

[2] Center of Cyber-Physical Systems (C2PS), Department of Computer Science, Khalifa University, Abu Dhabi, UAE

[3] Artificial Intelligence Cyber Systems Research Center, Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon

[4] Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

[5] KU 6G Research Center, Department of Computer and Information Engineering, Khalifa University, Abu Dhabi, UAE

[6] Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada

***Abstract*—The Sixth-Generation (6G) technology is envisioned to revolutionize vehicular networks by enabling dynamic, on-demand service deployment, particularly within the immersive domains of the metaverse. However, existing solutions are limited by generalized services for specific traffic events, the immobility of Roadside Units (RSU), and limited context awareness in unimodal data inputs. This paper proposes a vehicular Multimodal Large Language Model (MLLM)-driven framework for context-aware and heterogeneous service deployment in 6G vehicular networks. Specifically, we capitalize on multimodal data (e.g., text, images, and videos) with rich contextual meanings to enhance the vehicular MLLM's inference accuracy for on-demand service deployment recommendation. Considering the fixed locations of RSUs, we introduce OBU clusters as alternative node options for hosting recommended on-demand services. Then, we develop an optimal node selection algorithm that selects the most suitable RSU or OBU cluster node for an application-specific metaverse service deployment, considering their resource allocations, expected Quality of Service (QoS), and Resource Utilization (RU) offerings, as well as resource constraints. Comprehensive simulation results reveal that the proposed vehicular MLLM improves perception and service recommendation accuracy by about 12.5% and 8.9%, respectively, compared with GPT-4. Moreover, the node selection algorithm selects the optimal node with the highest expected utility for on-demand service deployment.**

***Index Terms*—Multimodal LLMs, vehicular networks, on-demand service deployment, 6G**

I. Introduction

The advent of the upcoming Sixth-Generation (6G) mobile network technology promises a plethora of transformative advancements in vehicular networks, characterized by ultra-reliable connectivity and pervasive intelligence [1]. To achieve this feat, vehicular networks should be capable of dynamically deploying on-demand services tailored to the heterogeneous needs of the stakeholders (vehicles, pedestrians, etc.) while adapting to the dynamicity of the environment. Such services in the vehicular network paradigm span emergency response, real-time traffic management, and metaverse applications (e.g., augmented passenger entertainment).

Cloud computing has been primarily leveraged to deploy on-demand services, where the service is hosted on a centralized cloud server, and users are granted access on demand. For instance, Salahuddin *et al.* [2] proposed a Roadside Unit (RSU) cloud architecture that integrates traditional and specialized RSUs to enable dynamic service management in the Internet of Vehicles (IoV). While effective for many applications, this approach introduces unacceptable delays, considering that most vehicular network services are delay-sensitive. Moreover, the restricted physical position of RSUs and clouds constrains their flexibility to provide service continuity with variation in vehicle mobility. To bring cloud intelligence closer to the edge and provide everywhere, anytime service access, Sami *et al.* [3] proposed leveraging vehicular Onboard Units (OBUs) as fog nodes to deploy containerized microservices on demand. The OBUs were clustered to form volunteering vehicular fogs, which host the microservices. However, they mainly focused on container placement without intricate emphasis on service customization for specific traffic events. It is noteworthy that different vehicular network events may require deploying customized services to resolve the issue at hand.

Several existing works have proposed Machine Learning (ML) techniques to enhance on-demand service deployment in vehicular networks [4], [5]. Huang *et al.* [5] utilized Deep Reinforcement Learning (DRL) to dynamically forecast service demands and orchestrate strategic on-demand dynamic deployment in multiple edge clouds. However, they overlooked the criteria for selecting a specific service for deployment, further limiting their applicability. In summary, the above-mentioned works share the following common limitations: 1) they primarily rely on unimodal (textual or image metadata) data inputs, which limit the richness of deployment decision-making contexts; 2) they utilize mere heuristic or conventional ML approaches that struggle to capture the contextual relation-

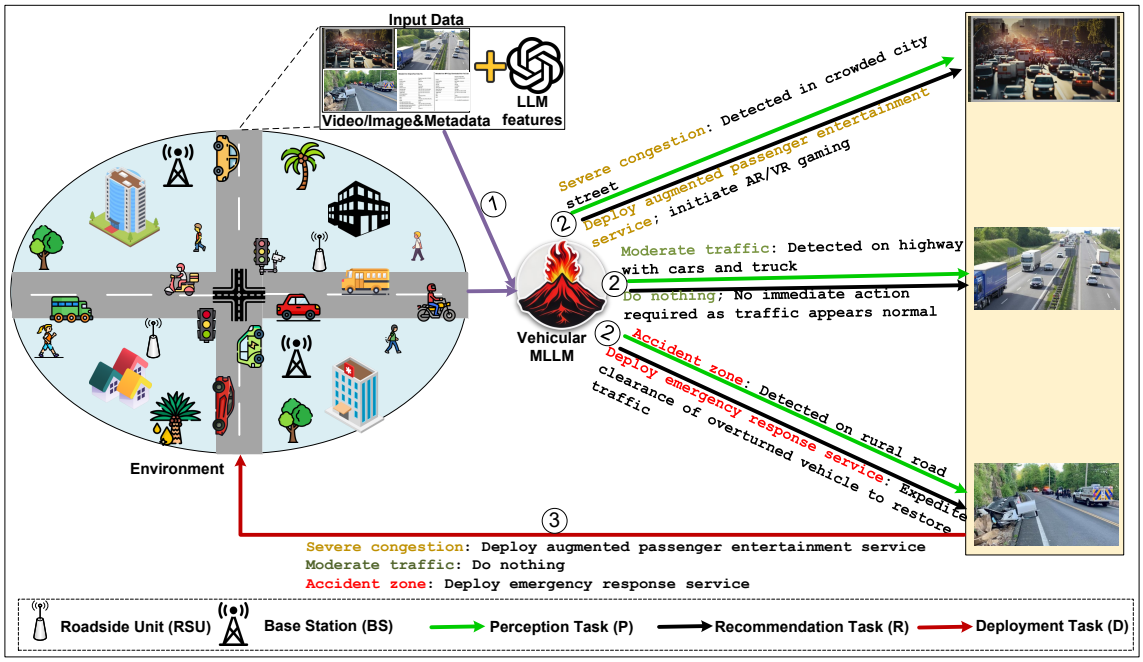

Fig. 1: MLLM-driven on-demand service deployment framework.

ships between data inputs and the dynamic environment.

Large Language Models (LLMs) have demonstrated exceptional learning capabilities across a wide range of tasks, and vehicular network tasks are not an exception [6]. Particularly, Multimodal LLMs (MLLMs) offer several distinct advantages by integrating multiple data modalities, such as text, image, and video data, to enhance semantic understanding and improve decision-making in autonomous vehicles [7], [8]. Liu *et al.* [8] utilized MLLM to propose a framework that employs bird's-eye view representation to optimize the spatial relationships between cooperative autonomous vehicles and passenger requests. Until now, the application of MLLMs for context-aware and heterogeneous on-demand service deployment in vehicular networks has been unexplored.

Motivated by the above-mentioned limitations, this paper proposes a novel MLLM-driven framework for on-demand heterogeneous service deployment in 6G vehicular networks. Unlike existing studies that often assume a one-size-fits-all service deployment, our framework acknowledges that the diversity in network traffic events warrants customized services to be deployed. *To the best of our knowledge, this is the first work to leverage MLLMs for context-aware and heterogeneous on-demand service deployment in 6G vehicular networks.* The main contributions of this paper are summarized as follows:

- We design a vehicular MLLM trained on multimodal data to provide more accurate, context-aware inference on real-time deployment of a specific on-demand service that matches a specific network traffic event.
- To address challenges such as high vehicular mobility and fixed positions of RSUs, we introduce an alternative deployment option where vehicular OBUs form dynamic clusters as deployment nodes to collaboratively host on-demand services. The OBU clusters aggregate their resources to deploy on-demand services, overcoming the resource limitation problem of individual OBUs.
- We propose an optimal node selection algorithm that determines the most suitable deployment option among OBU clusters and RSU nodes for a recommended metaverse (augmented passenger entertainment) service, considering their best response resource allocations, expected Quality of Service (QoS), and Resource Utilization (RU) offerings, as well as resource limitations.

The rest of the paper is organized as follows: Section II presents the system model, and Section III formulates the problem. Section IV presents the simulation results, and Section V concludes the paper.

II. System Model

A. System Framework

Consider a time-variant vehicular network with a set of $\mathcal{B} = \{1, 2, ..., B\}$ vehicles randomly distributed across the coverage area, as depicted in Fig. 1. The environment is equipped with cameras and sensors mounted on traffic lights and RSUs to facilitate real-time environment perception. Each vehicle b is equipped with onboard cameras to capture visual data from its surroundings and OBUs to enable local computation. A set of $\mathcal{S} = \{1, 2, ..., S\}$ services is available for on-demand deployment, tailored to specific traffic events such as congestion, moderate traffic, or accidents. The network constitutes a single BS, and a set of RSUs and OBUs, denoted

as $\mathcal{I} = \{1, 2, ..., I\}$, and $\mathcal{O} = \{1, 2, ..., O\}$, respectively. A number of OBUs form a cluster j to be in contention for hosting a service, i.e., $j \in \mathcal{O}$. Traditionally, an RSU will be selected for on-demand service deployment due to its high resource capacity and deployment stability. However, given the fixed position of RSUs and high vehicular mobility, OBUs can dynamically form clusters to collaboratively host services, ensuring flexibility in service deployment. To determine the most suitable infrastructure for deploying a given service, a node selection algorithm considers resource allocation and the corresponding expected utility (QoS and RU offering) of each RSU and OBU cluster node, as well as resource constraints, selecting the one with the optimal utility. In this work, resource allocation refers to assigning fractions of communication, computation, and storage resources for service hosting.

The mode of operation of the vehicular MLLM-driven framework involves the following primary tasks:

(1) **Perception task (P):** The cameras mounted on vehicles, RSUs, and traffic lights capture real-time images and videos of traffic flow and accidents. The sensors collect metadata such as vehicle speed, weather conditions, and traffic density. To enhance the contextual understanding of the multimodal data (camera feed and sensor metadata), each image and video data is analyzed by a Generative Pre-Trained Transformer (GPT) model that extracts feature representations. Then, the multimodal data and GPT-generated features are combined into a unified input dataset. This dataset is fed into a vehicular MLLM deployed in the cloud, which is trained and fine-tuned to detect traffic events such as congestion, moderate traffic, or accident zones.

(2) **Recommendation task (R):** Based on perception inference insights from **P**, the vehicular MLLM recommends an appropriate context-specific service for on-demand deployment. For instance, if the MLLM detects an accident on a rural road, it recommends emergency response services, such as clearing overturned vehicles and redirecting traffic.

(3) **Deployment task (D):** The deployment task ensures the efficient hosting of the recommended service via optimal node selection. Based on the recommended service, a node selection algorithm determines the optimal hosting node (either RSU or OBU cluster) based on their resource allocations and expected utility offerings, as well as the resource limitations.

B. Vehicular MLLM

The vehicular MLLM leverages the transformer architecture to recommend on-demand service s for deployment utilizing multimodal inputs of tokenized text, image, and video data. It consists of an LLM $f_\Phi(\cdot)$ parameterized by Φ, a vision encoder $g_\psi(\cdot)$ parameterized by ψ, and a projector $p_\theta(\cdot)$ parameterized by θ [9]. The inputs are embedded into dense feature vectors with modality-specific embeddings E and positional encodings PE, which can be simplified as

$$\begin{aligned} E &= E_{txt} + E_{img} + E_{vid}, \\ PE &= PE_{txt} + PE_{img} + PE_{vid}, \end{aligned} \tag{1}$$

where the input text X_{txt}, image data X_{img}, and video data X_{vid} are encoded into text feature $Z_{txt} = f(X_{txt})$ and visual features $Z_{img} = g(X_{img})$, and $Z_{vid} = g(X_{vid})$, respectively. The self-attention mechanism computes the relationships between tokens via query Q, key K, and value V matrices as

$$\begin{aligned} &Q = (E + PE) \times W^Q; K = (E + PE) \times W^K; \\ &V = (E + PE) \times W^V, \end{aligned} \tag{2}$$

where W is a learned weight matrix. Following [10], the attention and multi-head self-attention are given by

$$\begin{aligned} Attention(Q, K, V) &= softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \\ MultiHead(Q, K, V) &= Concat[head_1; \ldots; head_h]W^O, \end{aligned} \tag{3}$$

where $head_h = Attention(QW_h^Q, KW_h^K, VW_h^V)$, d_k is the dimensionality of K, and W^O is the learned weight matrix for combining multi-head attention outputs. The resulting self-attention layer output is $\mathcal{Z} = MultiHead(Q, K, V)$, where $\mathcal{Z} = \mathcal{Z}_{txt}, \mathcal{Z}_{img}, \mathcal{Z}_{vid}$. Then $\mathcal{Z}_{img}$ and $\mathcal{Z}_{vid}$ are projected into a shared embedding space for textual representation using the projector $p_\sigma(\cdot)$, yielding modality-specific embeddings as $H_{img} = p(\mathcal{Z}_{img})$, and $H_{vid} = p(\mathcal{Z}_{vid})$. These embeddings, together with $\mathcal{Z}_{txt}$, are passed through the MLP in $f_\Phi(\cdot)$, comprising stacked transformer layers of self-attention and Feed-Forward Networks (FFNs). The final layer recommends the best-fit service s for deployment by

$$s = MLLM(\mathcal{Z}). \tag{4}$$

C. Deployment Model

Node selection for hosting the recommended on-demand service s on RSU i or OBU cluster j is based on three factors: resource allocation, expected utility offering, and resource constraints. For simplicity, both RSU i and OBU cluster j are collectively referred to as node n, where $n \in \{i, j\}$, unless otherwise specified. For on-demand service deployment, a requesting user sends a service request $q_s = [s : \phi_s(r_s^{min}, d_s^{max})]$ to the BS to select a node for service hosting, where ϕ_s, r_s^{min}, and d_s^{max} denote the QoS, minimum data rate, and maximum delay requirements of service s, respectively. The BS broadcasts the request to RSUs and OBU clusters. Each RSU and OBU cluster responds with their resource allocations $\rho_{n,s}$, expected QoS offerings $\phi_{n,s}(r_{n,s}, d_{n,s})$, and related RU $\Psi_{n,s}$ for service s as $y_{n,s} = [n : \rho_{n,s}, \phi_{n,s}(r_{n,s}, d_{n,s}), \Psi_{n,s}]$, where $\phi_{n,s}$, $r_{n,s}$, and $d_{n,s}$ are the expected QoS, data rate, and delay of node n, respectively. Based on q_s and $y_{n,s}$, the BS runs a node selection algorithm to designate the most appropriate node that optimizes $\rho_{n,s}$, $\phi_{n,s}$, and $\Psi_{n,s}$, considering resource limitations. The total resource owned by node n includes bandwidth W_n (Hz) and computation C_n (CPU cycles). Therefore, the fraction of resource required to host service s is given by $\rho_{n,s} = \{w_{n,s}, c_{n,s}\}$.

According to the Shannon capacity theory, the data rate $r_{n,s}$ that node n can offer to host service s is given by

$$r_{n,s} = w_{n,s} \cdot \log_2(1 + \psi_{n,BS}), \tag{5}$$

where $\psi_{n,BS}$ denotes the Signal-to-Noise-Ratio (SNR). The data rate constraint is satisfied if $r_{n,s} \geq r_s^{min}$.

For hosting service s on node n, two forms of delay are considered: transmission delay $d_{n,s}^{trans}$, and computation delay $d_{n,s}^{comp}$. The transmission delay is expressed as $d_{n,s}^{trans} = \frac{l_s}{r_{n,s}}$, where l_s is the data size (in MB); the computation delay is expressed as $d_{n,s}^{comp} = \frac{\gamma_s}{c_{n,s}}$, where γ_s is the number of CPU cycles (in Megacycles). Therefore, the deployment delay is expressed as

$$d_{n,s} = d_{n,s}^{trans} + d_{n,s}^{comp}. \tag{6}$$

The delay constraint is satisfied if $d_{n,s} \leq d_s^{max}$.

Based on $r_{n,s}$ and $d_{n,s}$, the QoS is computed by [11]

$$\begin{aligned} \phi_{n,s}(r_{n,s}) &= \frac{1}{1+e^{-\eta(r_{n,s}-r_s^{min})}} \\ \phi_{n,s}(d_{n,s}) &= \frac{1}{1+e^{-\eta(d_s^{max}-d_{n,s})}}, \end{aligned} \tag{7}$$

where $\phi_{n,s}(r_{n,s}, d_{n,s}) = \frac{\phi_{n,s}(r_{n,s})+\phi_{n,s}(d_{n,s})}{2}$, with its value in the range $[0, 1]$, and η is a curve-fitting parameter.

Given the fraction of resource required to host service s, RU $\Psi_{n,s}$ is defined as the ratio of allocated resource to total resource, and is expressed as

$$\Psi_{n,s} = \frac{1}{2}\left(\frac{w_{n,s}}{W_n} + \frac{c_{n,s}}{C_n}\right), \tag{8}$$

whose value is in the range $[0, 1]$. The expected utility $\mathcal{U}_{n,s}(\phi_{n,s}, \Psi_{n,s})$ of node n for hosting service s is given by

$$\mathcal{U}_{n,s}(\phi_{n,s}, \Psi_{n,s}) = \alpha \cdot \phi_{n,s} - \beta \cdot \Psi_{n,s}, \tag{9}$$

where $\alpha + \beta = 1$; α and β are coefficients that denote the importance of $\phi_{n,s}$ and $\Psi_{n,s}$, respectively.

III. Problem Formulation

Based on the service recommended by the vehicular MLLM, the BS selects the most appropriate hosting node, considering $\rho_{n,s}$ and $\mathcal{U}_{n,s}$ of each node, as well as resource constraints. We formulate the node selection problem as an optimization problem that maximizes $\mathcal{U}_{n,s}$ with the appropriate $\rho_{n,s}$. To this end, the objective function is defined as $\mathcal{U}_{n,s}(\phi_{n,s}, \Psi_{n,s})$. Therefore, the optimization problem is expressed as

$$\begin{aligned} \underset{w_{n,s}, c_{n,s}}{\text{maximize}} \quad & \mathcal{U}_{n,s}(\phi_{n,s}, \Psi_{n,s}) \\ \text{s. t.:} \quad & C1\text{: } w_{n,s} \leq W_n,\ c_{n,s} \leq C_n, \\ & C2\text{: } r_{n,s} \geq r_s^{min},\ d_{n,s} \leq d_{n,s}^{max}. \end{aligned} \tag{10}$$

Constraint $C1$ satisfies the bandwidth and computation constraints, and Constraint $C2$ satisfies the data rate and delay constraints. We note that the outcome of (10) is the optimal node n^* selected to host service s. To achieve this, we design an optimal node selection algorithm that selects the node with the maximum $\mathcal{U}_{n,s}(\phi_{n,s}, \Psi_{n,s})$ to host the on-demand service. **Algorithm 1** presents the proposed node selection algorithm.

Algorithm 1 Optimal Node Selection Algorithm

Input: MLLM output s, Node set $\mathcal{N}$, service request q_s
Output: Selected optimal node n^*

```
for each node n ∈ N do
    Obtain request q_s = [s : φ_s(r_s^min, d_s^max)] from BS
    Select w_{n,s} and c_{n,s} to compute r_{n,s}, d_{n,s}, and U_{n,s} by (5), (6), and (9) that satisfy C1 and C2
    Compute U_{n,s}(φ_{n,s}, Ψ_{n,s}) by (10)
    Submit U_{n,s}(φ_{n,s}, Ψ_{n,s}) as part of the response y_{n,s} = [n : ρ_{n,s}, φ_{n,s}(r_{n,s}, d_{n,s}), Ψ_{n,s}]
end for
BS selects n* with max. U_{n,s}(φ_{n,s}, Ψ_{n,s}) to host s
return n*
```

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed vehicular MLLM framework through simulations. We build upon *LLaVA-OneVision* [9], an open-source MLLM, as the base model for developing our vehicular MLLM that performs **P** and **R** tasks. LLaVA-OneVision is ideal for our use case due to its versatility in computer vision tasks, including single-image, multi-image, and video analysis. Its architecture includes a *vision encoder*, an *LLM*, and a *projector*. We deploy Qwen2 [12] as the LLM and SigLIP [13] as the vision encoder. Two variants of the vehicular MLLM are used: one with 0.5 billion parameters (0.5B) and another with 7 billion parameters (7B). Unless explicitly specified, the training setup and approach apply to both variants.

A. Dataset Description

Our proposed vehicular MLLM framework is evaluated on the following publicly available datasets:

Highway accident detection and classification dataset [14]: Includes intersections, exits, and associated features such as accident type.

CCTV dataset [15]: Includes video footage of highways and associated features such as weather conditions, timestamps, and congestion status.

The video files in the datasets are aggregated and denoted as **V**. We refer to the original features in both datasets as *High-Level* ($\mathbf{F_{HL}}$) features. To enable our model to well identify correlations with $\mathbf{F_{HL}}$ features, we utilize GPT-4 to artificially extract environment information that may not have been explicitly provided in the datasets, e.g., visible injuries, pedestrian involvement, etc. This extracted information is referred to as *Low-Level* ($\mathbf{F_{LL}}$) features and describes the main observations in each image or video data frame. We provide three dataset types used in our simulations as follows:

- **Dataset 1 (D1):** Comprises **V**, $\mathbf{F_{HL}}$, and the expected output of **P**.
- **Dataset 2 (D2):** Comprises **V**, $\mathbf{F_{HL}}$, $\mathbf{F_{LL}}$, and the expected output of **P**.
- **Dataset 3 (D3):** Comprises the inferences of **P**, and the expected output of **R**.

TABLE I: Performance Comparison of Different Models

Tasks	Perception (P)				Recommendation (R)	
Models (0.5B)	**Model 1**	**Proposed-P**	**Model 2**	**Model GPT-4**	**Proposed-R**	**Model GPT-4**
F1 Score (f1)	0.791	0.829	0.820	0.743	0.851	0.798
GLEU (g)	0.229	0.269	0.243	0.082	0.259	0.194
Models (7B)	**Model 1**	**Proposed-P**	**Model 2**	**Model GPT-4**	**Proposed-R**	**Model GPT-4**
F1 Score (f1)	0.798	**0.836**	0.831	0.743	**0.869**	0.798
GLEU (g)	0.234	**0.273**	0.267	0.082	**0.313**	0.194

To ensure representative coverage across all datasets, we stratify the evaluation datasets as 20% random splits for each and remove 30% of $\mathbf{F_{HL}}$ from **D1** to mimic real-world scenarios.

B. Model Training and Fine-Tuning

Based on **D1**, **D2**, and **D3**, we compare our proposed model (**Proposed-P/Proposed-R**) with benchmarks (**Model 1**, **Model 2**, and **Model GPT-4**) for performance evaluation:

- **Model 1:** Trained on **D2**. **V** and $\mathbf{F_{HL}}$ are passed through GPT-4 to obtain $\mathbf{F_{LL}}$. Then, the model is trained on **V**, $\mathbf{F_{HL}}$, and $\mathbf{F_{LL}}$. Its output is the result of **P**.
- **Proposed-P:** First trained on **D2** (with Model 1 as base model). Then $\mathbf{F_{LL}}$ is removed and the model is trained on **D1**. It has knowledge of $\mathbf{F_{LL}}$ through initial training on **D2**. Its output is the result of **P**.
- **Model 2:** Directly trained on **D1** (with LLaVA-OneVision model as base model). It has no knowledge of $\mathbf{F_{LL}}$. Its output is the result of **P**.
- **Proposed-R:** Trained on **D3** (with Proposed-P as base model). Its output is the result of **R**.
- **Model GPT-4:** Proprietary GPT-4 model. Its output is the result of **P** or **R**.

Model training was performed on two Tesla V100 GPUs, each with 32GB of VRAM. Due to the substantial memory requirements of MLLMs, we utilized Low-Rank Adaptation (LoRA), allowing the training process to fit within the available GPU memory. Fine-tuning was performed with learning rate and warm-up ratio set to 1×10^{-6} and 0.03, respectively. A global batch size of 4 was used, and the training times varied as follows: approximately 2 hours for the 0.5B model, 5.5 hours for the 7B model when using video data, and 1 hour for datasets without video inputs.

C. Simulation Results

We first evaluate the performance of the different models, each with 0.5B and 7B parameters, in terms of their ability to perform perception task **P**. We use BERTScore and Google BLEU (GLEU) as the primary evaluation metrics. BERTScore evaluates text similarity with a range of [0,1] and comprises Precision (p), Recall (r), and F1-score (f1). GLEU measures the overlap of n-grams (sequences of words) between the generated and reference texts and comprises GLEU score (g). Columns 2-5 of Table I provide the performance of the various models in terms of perception task **P**. Since p and r are used to calculate f1, we omit their values in the table.

The results show that the 7B variants of the models outperform their 0.5B counterparts; however, similar trends and conclusions can be drawn for both. Therefore, we focus our analysis more on the 7B models. We observe that Proposed-P achieves the best f1 and g scores among the comparative models, followed by Model 2, Model 1, and Model GPT-4, respectively. This is because Proposed-P is first trained on dataset **D2**, which gives the model a more contextual understanding of the input data. Even though $\mathbf{F_{LL}}$ is removed, and the model is retrained on **D1**, it has some knowledge of $\mathbf{F_{LL}}$ to complement inference accuracy. Model 2 has no knowledge of $\mathbf{F_{LL}}$, which makes it heavily reliant on $\mathbf{F_{HL}}$. Since Model 1 relies fully on both $\mathbf{F_{HL}}$ and $\mathbf{F_{LL}}$ for model inference, it struggles to achieve more accurate results when $\mathbf{F_{LL}}$ is removed from its input. Finally, Model GPT-4 achieves the worst result due to the absence of a complementary contextual understanding of the datasets. We conclude that Proposed-P demonstrates superior generalization to datasets with missing features and, thus, is best for our scenario.

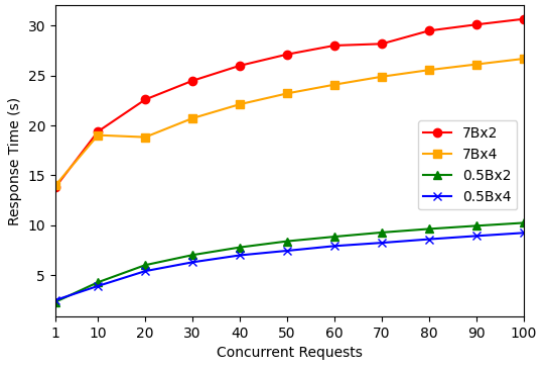

Fig. 2: Concurrent requests vs. response time

Since Proposed-P achieves the best results for perception task **P**, we use it as the base model to deploy Proposed-R and compare its performance with Model GPT-4 in terms of recommendation task **R**. Columns 6-7 of Table I provide the results of **R**. We observe that Proposed-R achieves superior results as compared to Model GPT-4. Specifically, Proposed-R and Model GPT-4 achieve f1 scores of 0.869 and 0.798, respectively, and g scores of 0.313 and 0.194, respectively. This demonstrates that our proposed model effectively analyzes multimodal inputs and provides actionable recommendations for on-demand service deployment based on the contextual understanding and inference from **P**.

Fig. 2 evaluates the performance of our proposed model in terms of response time as the number of concurrent requests increases. We submit 1-100 requests of video frames and their accompanying metadata to the model for inference. We consider both 0.5B and 7B models, each with 2 model replicas and 4 model replicas as follows: 0.5B (0.5B×2, 0.5B×4); 7B (7B×2, 7B×4). Each model replica is hosted on a GPU with 32GB VRAM, i.e., 2 Replicas require 2 GPUs, and 4 Replicas require 4 GPUs. From the figure, we observe that the response time increases with an increasing number of concurrent requests for all model versions. Specifically, the 7B models record higher response times than the 0.5B models due to the fact that the 7B models are larger than their 0.5B

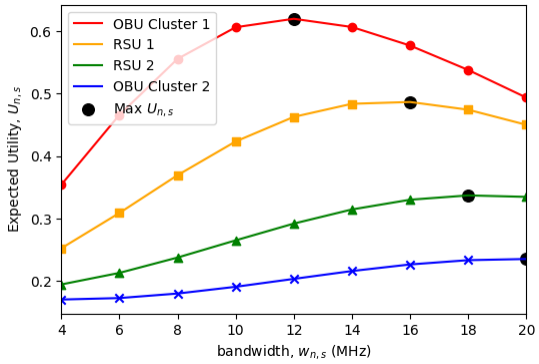

Fig. 3: Expected utility of nodes

counterparts. Moreover, both model variants with 4 replicas record lower response times compared with their 2 replicas counterparts. This is because the model with 4 replicas deploys four models to respond to requests unlike the model with 2 replicas that deploys 2 models to respond to requests. We can conclude that the 0.5B variants of our model achieve lower response time at the expense of subpar f1 and g scores compared with the 7B variants.

Suppose the vehicular MLLM recommends a metaverse (augmented passenger entertainment) service for on-demand deployment. Based on the output of **R**, we select an optimal node amongst RSUs and OBU clusters for deployment task **D**. In this simulation, 2 RSUs and 2 OBU clusters are randomly positioned with reference to the traffic event's location. We set $w_{n,s}$ to [1, 2,..., 20]MHz, r_s^{min} to 75Mbps, and d_s^{max} to 70ms. All other parameters are chosen according to [5]. Fig. 3 shows the performance of the proposed node selection algorithm in terms of normalized expected utility (defined in (9)) with increasing $w_{n,s}$. From the figure, we observe that the expected utility of each node increases with increasing $w_{n,s}$. Notably, OBU Cluster 1 achieves the highest expected utility with approximately 12MHz bandwidth, while RSU 1, RSU 2, and OBU Cluster 2 achieve their highest expected utilities with about 16MHz, 18MHz, and 20MHz bandwidth, respectively. This implies that OBU Cluster 1 utilizes its bandwidth resource more efficiently to balance QoS and RU while keeping bandwidth allocation at acceptable levels. Although the other nodes utilize more bandwidth resources, their utilities are lower due to their inability to achieve the best tradeoff between QoS and RU. We can conclude that OBU Cluster 1 is able to offer the best expected utility with reasonable resource allocation. Therefore, it is selected to deploy the on-demand service.

V. Conclusion

This paper proposed a novel MLLM-driven framework for context-aware and heterogeneous on-demand service deployment in 6G vehicular networks. Our framework leverages multimodal data to provide a more comprehensive and context-aware understanding of traffic events, enhancing the accuracy of on-demand service recommendations for specific traffic events. We introduced OBU clusters as alternatives to fixed RSUs for dynamic service hosting. Then, we developed a node selection algorithm that optimizes a metaverse service deployment considering resource allocation, expected utility, and resource limitations of each RSU and OBU cluster. Simulation results demonstrated the superiority of the proposed framework compared with GPT-4. Furthermore, the node selection algorithm consistently identified the optimal deployment node, ensuring efficient and reliable service delivery. Future work will explore simultaneous multiple service deployments.

References

[1] H. Guo, X. Zhou, J. Liu, and Y. Zhang, "Vehicular intelligence in 6g: Networking, communications, and computing," *Vehicular Commun.*, vol. 33, p. 100399, 2022.

[2] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-defined networking for rsu clouds in support of the internet of vehicles," *IEEE Internet of Things journal*, vol. 2, no. 2, pp. 133–144, 2014.

[3] H. Sami, A. Mourad, and W. El-Hajj, "Vehicular-obus-as-on-demand-fogs: Resource and context aware deployment of containerized micro-services," *IEEE/ACM trans. on networking*, vol. 28, no. 2, pp. 778–790, 2020.

[4] Q. Zhang, M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Machine learning for predictive on-demand deployment of uavs for wireless communications," in *2018 IEEE Global Commun. Conf. (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[5] Y. Huang, B. Feng, Y. Cao, Z. Guo, M. Zhang, and B. Zheng, "Collaborative on-demand dynamic deployment via deep reinforcement learning for iov service in multi edge clouds," *Journal of Cloud Comp.*, vol. 12, no. 1, p. 119, 2023.

[6] Y. Tang, X. Dai, C. Zhao, Q. Cheng, and Y. Lv, "Large language model-driven urban traffic signal control," in *2024 Australian & New Zealand Control Conf. (ANZCC)*. IEEE, 2024, pp. 67–71.

[7] G. O. Boateng, H. Sami, A. Alagha, H. Elmekki, A. Hammoud, R. Mizouni, A. Mourad, H. Otrok, J. Bentahar, S. Muhaidat, C. Talhi, Z. Dziong, and M. Guizani, "A survey on large language models for communication, network, and service management: Application insights, challenges, and future directions," 2024. [Online]. Available: https://arxiv.org/abs/2412.19823

[8] H. Liu, R. Yao, Z. Huang, S. Shen, and J. Ma, "Lmmcodrive: Cooperative driving with large multimodal model," 2024. [Online]. Available: https://arxiv.org/abs/2409.11981

[9] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, and C. Li, "Llava-onevision: Easy visual task transfer," 2024. [Online]. Available: https://arxiv.org/abs/2408.03326

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[11] C. Xu, T. Li, M. Sheng, and J. Li, "Self-organized dynamic caching space sharing in virtualized wireless networks," in *2016 IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1–6.

[12] A. Y. et al., "Qwen2 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2407.10671

[13] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, O. Hénaff, J. Harmsen, A. Steiner, and X. Zhai, "Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," 2025. [Online]. Available: https://arxiv.org/abs/2502.14786

[14] L. Kezebou, V. Oludare, K. Panetta, J. Intriligator, and S. Agaian, "Highway accident detection and classification from live traffic surveillance cameras: a comprehensive dataset and video action recognition benchmarking," in *Multimodal Image Exploitation and Learning 2022*, vol. 12100, May 2022, p. 121000M.

[15] A. Chan and N. Vasconcelos, "Probabilistic kernels for the classification of auto-regressive visual processes," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 846–851 vol. 1.