LLM-xApp: A Large Language Model Empowered Radio Resource Management xApp for 5G O-RAN

Xingqi Wu
University of Michigan-Dearborn, Dearborn MI, USA
xingqiwu@umich.edu

Junaid Farooq
University of Michigan-Dearborn, Dearborn MI, USA
mjfarooq@umich.edu

Yuhui Wang
University of Michigan-Dearborn, Dearborn MI, USA
ywangdq@umich.edu

Juntao Chen
Fordham University
New York NY, USA
jchen504@fordham.edu

***Abstract*—The decentralized and modular architecture of open radio access networks (O-RAN) enhances flexibility and interoperability but introduces significant challenges in efficiently managing resource allocation. The disaggregation of network functions across distributed unit, centralized unit, and RAN intelligent controller (RIC) creates complexities in coordinating resources across multiple network slices, each with distinct and dynamic quality of service (QoS) requirements. Traditional machine learning (ML) approaches for resource management often rely on extensive offline training, which is impractical in the highly variable and real-time environments of O-RAN systems. This paper presents LLM-xApp, a novel large language model (LLM)-powered xApp framework for adaptive radio resource management in O-RAN systems. The proposed framework is based on intelligently prompting LLM agents to dynamically optimize resource allocation to different network slices. Experimental evaluations are conducted on the OpenAI Cellular (OAIC) platform showcasing significant improvements in average data rates as well as the reliability of the slices, demonstrating the potential of LLMs to enhance real-time decision-making in next-generation wireless networks.**

***Index Terms*—Open radio access network (O-RAN), large language models (LLM), xApp, network slicing.**

I. Introduction

The evolution of telecommunications networks from 5G to Next-Generation (NextG) systems has introduced unprecedented challenges in managing highly dynamic and heterogeneous environments. These networks must cater to diverse service requirements such as ultra-reliable low-latency communication (URLLC), massive machine-type communications (mMTC), and enhanced mobile broadband (eMBB), all while maintaining exceptional efficiency, scalability, and adaptability [1], [2]. This increasing complexity necessitates intelligent automation to ensure optimal resource allocation and seamless operation across these varied use cases. Machine Learning (ML) and Artificial Intelligence (AI) have emerged as foundational technologies for addressing these challenges. Their ability to process vast amounts of real-time data and make adaptive decisions has proven invaluable for network management, optimization, and orchestration [3]–[5]. In particular, open radio access network (O-RAN) architectures, with their disaggregated and modular design, provide a fertile ground for integrating AI-driven solutions [6]–[8]. The RAN intelligent controller (RIC), a key component in O-RAN, allows for the deployment of third-party *xApps* that can monitor, control, and optimize RAN operations in near-real time. Existing xApps, such as those implemented in platforms like OpenAI Cellular (OAIC) [9], have shown promise in tasks like secure slicing [10] and RAN management [11]. However, leveraging xApps for dynamically adapting RAN resources in O-RAN slices, particularly in response to dynamic user equipment (UE) needs [12], remains a nascent area of research.

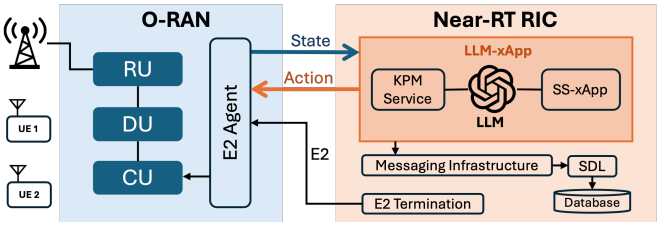


Fig. 1: Architecture of LLM-xApp and interface with O-RAN.

The concept of intent-based networking is crucial in this context. Intents represent the desired outcomes or service requirements of UEs, such as reduced latency, increased throughput, or optimized energy efficiency. Translating these high-level, dynamic intents into actionable network configurations is complex, particularly in scenarios with diverse and competing service demands. Large Language Models (LLMs) have recently emerged as a transformative technology in addressing such challenges [13], [14]. Their capabilities in understanding, reasoning, and generating human-like responses from complex inputs make them well-suited for interpreting and managing UE requirements [15], [16]. Unlike traditional rule-based or model-specific approaches, LLMs generalize across tasks, making them particularly effective in environments with high variability and incomplete information. For intent-driven resource allocation, LLMs excel at: (i) understanding diverse and ambiguous intent expressions from UEs; (ii) mapping these intents to actionable network configurations; and (iii) optimizing resource allocation dynamically while considering network constraints and priorities [17]. In this paper, we present the first xApp powered by an LLM for intent-driven resource allocation in O-RAN slices. The high-level architecture is shown in Fig. 1. Our approach focuses

Workshop on Security and Privacy of Next-Generation Networks (FutureG) 2025
24 February 2025, San Diego, CA, USA
ISBN 979-8-9919276-7-3
https://dx.doi.org/10.14722/futureg.2025.23057
www.ndss-symposium.org

on understanding UE QoS requirements expressed in real time, translating them into precise network configurations, and optimizing resource allocations dynamically to meet service requirements. This framework is implemented and tested on the open-source OAIC platform, demonstrating its feasibility and effectiveness in managing O-RAN slice resources under real-world conditions. The main contributions of our work are as follows:

- We introduce a novel xApp that leverages LLMs for real-time, intent-driven resource allocation in O-RAN slices.
- The developed LLM-xApp converts dynamic UE QoS requirements into actionable configurations to optimize resource utilization.
- The proposed framework is implemented and validated on the OAIC testbed, showcasing its capability in realistic O-RAN scenarios.

The rest of this paper is organized as follows. Section II provides an overview of the system model and problem formulation. Section III describes the design and implementation of our LLM-powered xApp on the OAIC testbed. Section IV presents experimental results, demonstrating the efficacy of the proposed framework. Finally, Section V concludes the paper.

II. System Model and Problem Formulation

We consider a 5G O-RAN system comprising two types of UEs, each connected to a specific network slice. The RAN system is considered to have two slices as follows: S_1, a high-priority slice with stringent QoS requirements, and S_2, a low-priority slice with more relaxed QoS demands. The available radio resources are partitioned into physical resource blocks (PRBs), which are dynamically allocated over discrete reconfiguration slots, indexed by $k = 1, 2, ...$, during which resource allocations are updated. Each reconfiguration slot consists of multiple sampling periods, indexed by $\mathcal{T} = \{1, 2, 3, \cdots, T\}$. Let $\mathcal{S}$ denote the set of operational network slices, where each slice $s \in \mathcal{S}$ is allocated $r_k^s \in \mathbb{Z}^+$ PRBs at reconfiguration slot k. The total PRB resource budget R is finite and shared among all slices, i.e., $\sum_{s\in S} r_k^s \leq R$. The number of users connected to each slice can vary during reconfiguration slots, affecting the experienced QoS. The QoS for slice s in reconfiguration slot k is represented by the utility vector: $\boldsymbol{u}_k^s = [u_k^s(t_k), u_k^s(t_k+1), \cdots, u_k^s(t_k+\tau_k)] \in \mathbb{R}_{\geq 0}^{\tau_k}$, where t_k is the starting sampling period of reconfiguration slot k, and τ_k denotes the number of sampling periods in slot k. If $u_k^s(t) < u_{\text{th}}^s$ for any t, the slice experiences reliability degradation.

The utility of slice s at any sampling period t is modeled by the QoS-aware utility function $U_s^t(\hat{\sigma}_s^t)$, where $\hat{\sigma}_s^t$ and σ_s^t represent the measured and requested data rates, respectively. For the high-priority slice (S_1), the utility is defined as

$$U_s^t(\hat{\sigma}_s^t) = \frac{1}{1+\exp\left(-a\left(\hat{\sigma}_s^t - \sigma_s^t + b\right)\right)}, \tag{1}$$

where a and b are parameters that shape the sigmoid curve. For low-priority slice (S_2), the utility is defined as follows:

$$U_s^t(\hat{\sigma}_s^t) = \frac{\log\left(\hat{\sigma}_s^t + c\right)}{\log\left(\sigma_s^t + c\right)}, \tag{2}$$

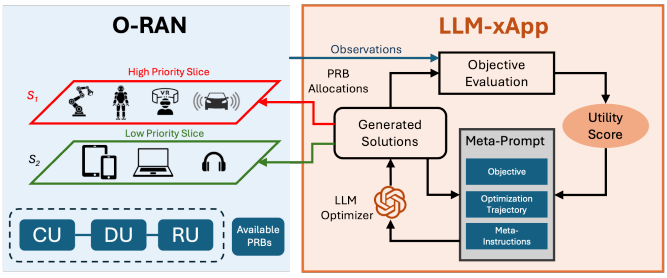


Fig. 2: LLM-driven optimization of resource provisioning in O-RAN.

where c controls the scale of the utility. The reliability of slice s at sampling period t, denoted by θ_s^t, is computed as the fraction of time windows where the utility falls below the threshold u_{th}^s. The reliability is defined as

$$\theta_s^t = \frac{\sum_{\tau=t-T_w/2}^{t+T_w/2} \mathbb{1}_{\left[u_k^s(\tau) \leq u_{\text{th}}^s\right]}}{T_w},$$

where T_w is the size of the measurement window, and $\mathbb{1}_{[.]}$ is the indicator function.

The objective of the proposed xApp in the near-realtime RIC is to maximize the time-averaged reliability across all slices and sampling periods. This can be formulated as

$$\max_{r_k^s \geq 0} \sum_{s\in S} \sum_{t\in\mathcal{T}} \frac{1}{|S||\mathcal{T}|} \theta_s^t, \tag{3}$$

$$\text{s.t.} \sum_{s\in S} r_k^s \leq R. \tag{4}$$

The relationship between the allocated resources r_k^s and the measured data rate $\hat{\sigma}_s^t$ is unknown but positively correlated. Additionally, the number of sampling periods τ_k within each reconfiguration slot k can vary dynamically. These characteristics result in a complex and dynamic mixed-integer optimization problem, where the optimization is constrained by both the discrete nature of the resource allocations and the time-varying dynamics of the system. In the following section, we propose a novel LLM-driven approach to address these challenges effectively.

III. Methodology

Our approach builds upon the optimization by prompting (OPRO) methodology proposed in [18], [19], leveraging LLMs as dynamic optimizers to address the challenge of resource allocation in O-RAN systems. The proposed LLM-xApp iteratively refines resource allocation strategies through structured meta-prompts, enabling the system to converge on solutions that maximize QoS metrics for network slices.

A. Agent Observation, Action, and Evaluation Function

The optimization process begins with the agent observing the system state at each reconfiguration slot k, represented by an observation vector

$$o_k = [\hat{\sigma}_1^{t_k}, \ldots, \hat{\sigma}_{|S|}^{t_k}, \sigma_1^{t_k}, \ldots, \sigma_{|S|}^{t_k}] \in \mathbb{R}_{\geq 0}^{2|S|}. \tag{5}$$

Here, $\hat{\sigma}_s^{t_k}$ represents the measured data rate for slice s over the previous observation window, and $\sigma_s^{t_k}$ is the requested data

rate. Based on this observation, the agent takes an action

$$\mathcal{A}_k = [a_1^k, \ldots, a_{|S|}^k] \in \mathbb{R}_{\geq 0}^{|S|}, \tag{6}$$

where a_s^k indicates the resource allocation proportion for slice s. The actual resource allocation r_s^k is computed as

$$r_s^k = \lceil R \cdot a_s^k / \sum_{s \in S} a_s^k \rceil, \tag{7}$$

where R is the total available resource. The agent's evaluation function ensures fair and reliable resource distribution while accommodating slices with higher QoS demands. The evaluation function is defined as:

$$V_k(o_k, A_k) = \sum_{s \in S} \beta_s \cdot g\left(\frac{2}{T_w} \sum_{t=t_k - T_w/2}^{t_k} \hat{\sigma}_s^t - \sigma_s^{t_k}\right) + \gamma_s, \tag{8}$$

where β_s is the priority weight of slice s, γ_s is a bias term, and $g(x)$ is a convex function peaking at $x = 0$.

B. Resource Allocation Framework for LLM-xApp

The proposed framework, illustrated in Fig. 2, iteratively allocates resources by translating the agent's historical observations into a meta-prompt. This meta-prompt, constructed with task descriptions and evaluation values, guides the LLM to generate optimal resource allocation decisions. After each iteration, the optimization history is updated in ascending order of evaluation values to prioritize better-performing solutions. To avoid local minima and ensure robust learning, a decaying temperature mechanism gradually reduces the randomness of LLM responses over iterations. High temperatures enable diverse exploration but may reduce accuracy, while lower temperatures yield more deterministic outputs. Algorithm 1 outlines the detailed LLM-driven orchestration process. The meta-prompt encapsulates three key components: an objective description, optimization history, and formatting instructions. Fig. 3 provides an example of this design for dynamic resource provisioning. The meta-prompt starts with a task description, detailing the LLM's role in maximizing utility through resource allocation. It includes optimization histories with evaluation scores sorted in ascending order, enabling the LLM to identify patterns from in-context examples. Finally, detailed output formatting instructions specify the required parameters and structure, ensuring the LLM's responses are interpretable and actionable.

Algorithm 1 LLM-Driven Dynamic Resource Management

Initialization:
Initialize system parameters: $t \leftarrow 0, k \leftarrow 0$, $Tem_0 \leftarrow Tem_{\text{max}}$, Tem_{Δ}, Tem_{min} , T.
while $t < T$ **do**
  if $t = t_k$ **then**
    Construct the prompt by incorporating the historical observations of the agent and their respective evaluation values.
    Extract action $\mathcal{A}^k$ based on LLM response.
    Map actions $\mathcal{A}^k$ to resource allocations $\mathbf{r}_s^k$.
    Compute the evaluation score using (8).
    Update optimization history based on o^k, $\mathcal{A}^k$, and $V^k\left(o^k, \mathcal{A}^k\right)$
    $Tem_{k+1} \leftarrow \max\left(Tem_{\text{min}}, Tem_k - Tem_{\Delta}\right)$
    $k \leftarrow k + 1$
    Update observation o^k based on the previous system response.
  end if
  $t \leftarrow t + 1$
end while

IV. Implementation and Analysis

A. Experiment Setup and workflow

The experiments were conducted using the OAIC testbed, an open-source platform designed to adhere to the O-RAN architecture. The implementation framework is built on the secure slicing xApp (SS-xApp) API [10], with modifications to handle the creation of NodeB, UEs, and slices. For LLM models, we use the U-M GPT [20], an API based service that allows access to the most popular LLM models such as GPT4. The workflow begins by initializing UEs, environment variables, the base station (BS), and the E2 interface between the RAN and OAIC's near-RT RIC. Once the 5G network setup is operational, UEs initiate communication sessions, and the xApp is deployed. Unlike traditional setups where a UE is bound to a single slice, we created multiple slices and assigned each UE to a separate slice.

For the experiment, two UEs were connected to srsRAN and assigned to two distinct slices with an initial equal allocation of PRBs. Traffic exchange was initiated using Iperf3 [21]. One of the UEs was programmed to request a higher data rate with an altered instruction set, significantly increasing its resource request frequency. After the deployment of the xApp, the system began collecting data, and the LLM agent made periodic resource allocation decisions at each reconfiguration slot, using historical data and environmental context. These decisions were mapped to share values via the ssxApp API, which determined resource allocation proportions and were transmitted to the NodeB via E2 control messages. The reconfiguration interval τ_k was determined by the response time of the LLM and system feedback. GPT-4 was employed as the LLM model for this experiment, with the function $g(x) = -x^2$. Parameter details are provided in Table I.

Parameters	Value	Parameters	Value
γ_1, γ_2	2000, 2000	Tem_{min}	0.3
Tem_{Δ}	0.05	β_1, β_2	2.5, 1
a_k^z	$\{1, 2, ..., 128\}$	σ_k^1, σ_k^2	40, 10
a, b, c	0.9, 6.5, 5	u_{th}^1, u_{th}^2	0.6, 0.96

TABLE I: Experiment Parameters.

B. Results and Comparative Analysis

To evaluate the performance of the LLM-driven resource orchestration framework, we compared it with three baseline approaches: random allocation, equal provisioning (equal distribution of resources between slices), and proportional provisioning (allocation based on the UEs' data rate requests). Figure 4 shows the data rate evolution for both UEs in the OAIC testbed system, starting from the initialization of the

Prompt

Objective:

```
You will help me optimize the resource allocation policy of an O-RAN system with
two slices. The resource levels of the slices are a and b, while the required
data rates are 40 and 10 Mbps respectively.
```

Optimization Trajectory:

```
Below are some previous (a,b) pairs along with their achieved data rates and
corresponding scores. The examples are arranged in descending order on the score,
where higher score values are better.

Input: a=115.0, b=13.0], Data rates: [31.79, 8.39], Score: 68.87
Input: a=108.0, b=20.0; Data rates: [27.99, 9.75], Score: 68.29
Input: a=100.0, b=28.0; Data rates: [21.35, 10.00], Score: 66.39
Input: a=85, b=34; Data rates: [32.20, 10.50]; Score: 61.30
```

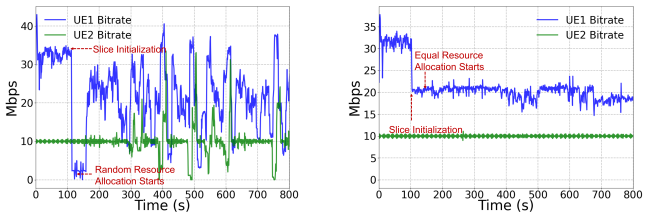
Meta Instructions:

```
Give me a new (a,b) pair of that is different from all pairs above, and has a
score higher than any of the above. Do not write text or code.
Output should in the form a=<value>, b=<value> where <value> can be an integer
between 0 and 128.
```

LLM Response

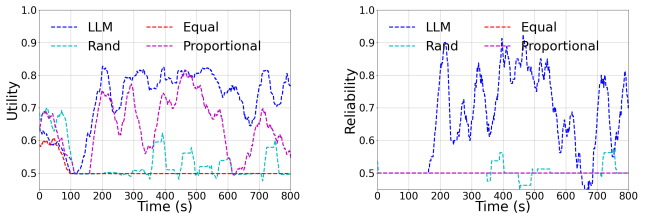
```
a=115, b=13
```

Fig. 3: An example of meta-prompt and LLM response for resource allocation in O-RAN.

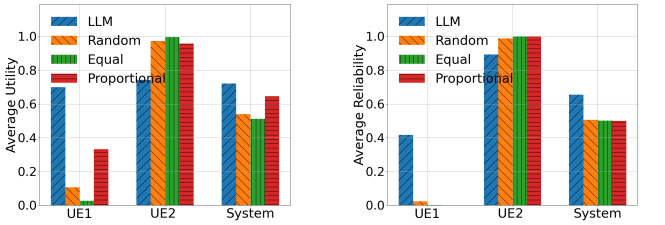


(a) (b) (c) (d)

Fig. 4: Comparative results of resource allocation scheme – (a) random, (b) equal, (c) proportional, and (d) LLM Driven.



(a) Smoothed System Utility (b) Smoothed System Reliability



(c) Time Averaged Utility (d) Time Averaged Reliability

Fig. 5: Comparative analysis of utility and reliability.

Iperf test. Initially, with limited total resources, UE1 achieved a data rate of approximately 30 Mbps. Around the 100-second mark, slices were created, and resources were evenly split between them, reducing UE1's data rate to 20 Mbps. Upon activation of the LLM-driven resource provisioning method, the agent adaptively adjusted resource allocations based on performance evaluations, as shown in Figure 4d. This highlights the method's ability to manage resource distribution effectively. Figures 5a and 5b present the window-smoothed utility and reliability metrics for UE1, UE2, and the overall system, averaged across both UEs. Figures 5c and 5d summarize the metrics over the entire measurement period. The results demonstrate that the proposed LLM-driven approach outperforms the baseline methods in terms of utility and reliability for both individual UEs and the overall system, affirming its efficiency in dynamic resource management.

V. Conclusions

This paper presented a novel framework leveraging LLMs for dynamic resource allocation in O-RAN systems. The proposed LLM-xApp effectively integrates contextual observations and historical optimization data into meta-prompts, enabling adaptive and efficient resource orchestration. Our approach employs an adaptive temperature mechanism to balance exploration and exploitation, ensuring robust convergence to high-quality solutions. The implementation and evaluation on the OAIC testbed demonstrate the superiority of the LLM-driven framework over traditional methods. The results highlight significant improvements in utility and reliability metrics for both individual UEs and the overall system. These findings underline the potential of LLMs as dynamic optimizers in complex, real-time resource management scenarios.

REFERENCES

[1] F. Rezazadeh, L. Zanzi, F. Devoti, H. Chergui, X. Costa-Pérez, and C. Verikoukis, "On the specialization of FDRL agents for scalable and distributed 6G RAN slicing orchestration," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 3473–3487, 2023.

[2] V.-P. Bui, S. R. Pandey, A. Casparsen, F. Chiariotti, and P. Popovski, "The role of game networking in the fusion of physical and digital worlds through 6G wireless networks," *IEEE Communications Magazine*, pp. 1–8, 2024.

[3] W. Alqwider, A. S. Abdalla, T. F. Rahman, and V. Marojevic, "Intelligent dynamic resource allocation and puncturing for next-generation wireless networks," *IEEE Internet of Things Journal*, vol. 11, no. 19, pp. 31 438–31 452, 2024.

[4] S. Zeb, M. A. Rathore, S. A. Hassan, S. Raza, K. Dev, and G. Fortino, "Toward AI-enabled NextG networks with edge intelligence-assisted microservice orchestration," *IEEE Wireless Communications*, vol. 30, no. 3, pp. 148–156, 2023.

[5] X. Wu, J. Farooq, and J. Chen, "Multi-agent distributed decentralized dynamic resource orchestration in 5G edge-cloud networks," *IEEE International Conference on Cloud Networking (Cloudnet 2024)*, Rio de Janeiro, Brazil, Nov. 2024.

[6] Z. A. E. Houda, H. Moudoud, and B. Brik, "Federated deep reinforcement learning for efficient jamming attack mitigation in O-RAN," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 7, pp. 9334–9343, 2024.

[7] N. Ghafouri, J. S. Vardakas, K. Ramantas, and C. Verikoukis, "A multi-level deep RL-based network slicing and resource management for O-RAN-based 6G cell-free networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 11, pp. 17 472–17 484, 2024.

[8] Y. A. Ergu, V.-L. Nguyen, R.-H. Hwang, Y.-D. Lin, C. Yu Cho, H.-K. Yang, H. Shin, and T. Q. Duong, "Efficient adversarial attacks against DRL-based resource allocation in intelligent O-RAN for V2X," *IEEE Transactions on Vehicular Technology*, pp. 1–13, 2024.

[9] P. S. Upadhyaya, N. Tripathi, J. Gaeddert, and J. H. Reed, "Open AI cellular (OAIC): An open source 5G O-RAN testbed for design and testing of AI-based RAN management algorithms," *IEEE Network*, vol. 37, no. 5, pp. 7–15, 2023.

[10] J. Moore, A. S. Abdalla, M. Zhang, and V. Marojevic, "Demo: SSxApp: Secure slicing for O-RAN deployments," in *IEEE Military Communications Conference (MILCOM 2023)*, Boston, MA USA, Nov. 2023, pp. 251–252.

[11] M. Kouchaki, S. B. H. Natanzi, M. Zhang, B. Tang, and V. Marojevic, "O-RAN performance analyzer: Platform design, development, and deployment," *IEEE Communications Magazine*, pp. 1–8, 2024.

[12] X. Wu, J. Farooq, and J. Chen, "Joint admission control and resource provisioning for URLLC traffic in O-RAN: A constrained multi-agent reinforcement learning approach," *IEEE International Conference on Communications (ICC 2025)*, Montreal, Canada, Jun. 2025.

[13] H. Zhou, C. Hu, Y. Yuan, Y. Cui, Y. Jin, C. Chen, H. Wu, D. Yuan, L. Jiang, D. Wu, X. Liu, C. Zhang, X. Wang, and J. Liu, "Large language model (LLM) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024.

[14] A. Maatouk, N. Piovesan, F. Ayed, A. De Domenico, and M. Debbah, "Large language models for telecom: Forthcoming impact on the industry," *IEEE Communications Magazine*, vol. 63, no. 1, pp. 62–68, 2025.

[15] O. G. Lira, O. M. Caicedo, and N. L. S. d. Fonseca, "Large language models for zero touch network configuration management," *IEEE Communications Magazine*, pp. 1–8, 2024.

[16] A. Mekrache, M. Mekki, A. Ksentini, B. Brik, and C. Verikoukis, "On combining XAI and LLMs for trustworthy zero-touch network and service management in 6G," *IEEE Communications Magazine*, pp. 1–7, 2024.

[17] M. A. Habib, P. E. I. Rivera, Y. Ozcan, M. Elsayed, M. Bavand, R. Gaigalas, and M. Erol-Kantarci, "LLM-based intent processing and network optimization using attention-based hierarchical reinforcement learning," 2024. [Online]. Available: https://arxiv.org/abs/2406.06059

[18] C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, and X. Chen, "Large language models as optimizers," *International Conference on Learning Representations (ICLR)*, Vienna, Austria, May 2024.

[19] K. A. Yuksel and H. Sawaf, "A multi-AI agent system for autonomous optimization of agentic AI solutions via iterative refinement and LLM-driven feedback loops," *arXiv preprint arXiv:2412.17149*, 2024.

[20] "U-M GPT Toolkit," https://its.umich.edu/computing/ai.

[21] *iPerf - The Ultimate Speed Test Tool for TCP, UDP, and SCTP*, 2003. [Online]. Available: https://iperf.fr/