

Vol 2 Issue 02 (2026)

<http://fsrma.org.hostajournal.com/>

ONLINE ISSN: 3134-7517

PRINT ISSN: 3134-7509

Graph-MARL-CloudLLM: A Graph-enhanced Multi-Agent Reinforcement Learning Framework for Adaptive Resource Scheduling and Workload Optimization of Large Language Model Inference Services across Multi-Cloud Platforms

Min Qiu

Trine University, Tempe, Arizona, USA
mqiu24@my.trine.edu

Jinxiang Yu

University of Delaware, Newark, USA
jinxiang@udel.edu

Ningyun Wang

Suzhou University, Suzhou, China
18954625369@qq.com

Abstract.

The increasing adoption of Large Language Models (LLMs) in enterprise artificial intelligence applications has created significant challenges for inference service management across multi-cloud platforms, where heterogeneous computing resources, dynamic workloads, network fluctuations, and strict quality-of-service (QoS) requirements hinder efficient resource utilization and stable service delivery. This paper aims to develop an intelligent scheduling framework for cross-cloud LLM inference services that can achieve adaptive workload orchestration, low-latency inference, and cost-efficient resource allocation. We propose Graph-MARL-CloudLLM (Graph-enhanced Multi-Agent Reinforcement Learning for Cross-Cloud Large Language Model Inference Optimization), a novel framework that integrates Graph Neural Networks (GNNs) with Multi-Agent Reinforcement Learning (MARL) to model complex cloud resource dependencies and perform autonomous scheduling decisions. Specifically, Graph-MARL-CloudLLM constructs a dynamic resource graph containing cloud nodes, GPU clusters, LLM instances, and inference requests, and employs cooperative agents for cloud selection, model selection, GPU allocation, and request routing. A QoS-aware reinforcement learning strategy is further designed to jointly optimize latency, throughput, cost, and resource utilization. Experimental results on a simulated heterogeneous multi-cloud LLM inference environment demonstrate that Graph-MARL-CloudLLM achieves superior scheduling performance compared with existing approaches. Specifically, the proposed framework reduces P99 latency from 271 ms to 194 ms, improves GPU utilization to 86.9%, and decreases inference cost by 17.7% compared with the MARL-based baseline. These results validate the effectiveness of graph-enhanced state representation and cooperative multi-agent optimization for adaptive LLM inference scheduling.

Keywords:

Large Language Model Inference; Multi-Cloud Computing; Multi-Agent Reinforcement Learning; Graph Neural Network; Adaptive Resource Scheduling; Cloud AI Infrastructure; Workload Optimization

1. INTRODUCTION

The rapid evolution of Large Language Models (LLMs), such as GPT, LLaMA, Qwen, and DeepSeek, has significantly accelerated the development of intelligent applications in areas including enterprise knowledge management, autonomous software engineering, financial analysis, healthcare assistance, and customer service automation. However, the large-scale deployment of LLM-based applications introduces substantial challenges in inference service management due to the enormous computational requirements, heterogeneous hardware resources, and highly dynamic user workloads. Unlike traditional deep learning inference systems, LLM inference services require efficient orchestration of GPU resources, model instances, and user requests to simultaneously achieve low latency, high throughput, and cost-efficient operation. To satisfy the increasing demand for scalable artificial intelligence (AI) services, many enterprises have adopted multi-cloud architectures by integrating heterogeneous infrastructures from platforms such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and Oracle Cloud Infrastructure (OCI). Multi-cloud deployment provides advantages including flexible resource availability, improved service reliability, and reduced dependency on a single cloud provider. However, managing LLM inference workloads across multiple clouds remains a challenging problem because of resource heterogeneity, fluctuating network conditions, diverse pricing mechanisms, and unpredictable request patterns. Existing scheduling approaches, such as rule-based strategies, heuristic optimization, and traditional load balancing algorithms, mainly rely on predefined policies and cannot dynamically adapt to complex cross-cloud environments. Furthermore, most existing reinforcement learning-based scheduling methods focus on single-cloud scenarios and fail to jointly consider cloud selection, model placement, GPU allocation, and request routing decisions. To address these limitations, this paper proposes Graph-MARL-CloudLLM (Graph-enhanced Multi-Agent Reinforcement Learning for Cross-Cloud Large Language Model Inference Optimization), an intelligent adaptive scheduling framework designed for large-scale LLM inference services across multi-cloud platforms. The proposed framework models the multi-cloud inference infrastructure as a dynamic resource graph, where cloud nodes, GPU clusters, LLM instances, and inference requests are represented as interconnected graph entities. A Graph Neural Network (GNN) based encoder is introduced to capture complex resource dependencies and network relationships. Based on the extracted graph representations, a Multi-Agent Reinforcement Learning (MARL) controller is developed, where multiple cooperative agents are responsible for cloud selection, LLM model selection, GPU resource allocation, and inference request routing. By integrating graph-based environment perception with reinforcement learning-based decision optimization, Graph-MARL-CloudLLM enables autonomous workload adaptation under dynamic multi-cloud conditions. The main contributions of this study are summarized as follows: A graph-enhanced multi-cloud LLM inference optimization framework is proposed. Graph-MARL-CloudLLM provides a unified architecture that integrates cloud resource modeling, workload awareness, and intelligent scheduling for heterogeneous multi-cloud LLM inference environments. A GNN-based dynamic resource representation mechanism is developed. The proposed method captures complex relationships among cloud infrastructures, computing resources, model instances, and inference requests, enabling topology-aware scheduling decisions. A cooperative MARL-based scheduling strategy is introduced. Multiple agents collaboratively optimize cloud selection, model deployment, GPU allocation, and request routing through a QoS-aware reward mechanism considering latency, throughput, cost, and resource utilization.

A comprehensive experimental evaluation is conducted under heterogeneous multi-cloud LLM inference workloads. The results demonstrate that Graph-MARL-CloudLLM effectively improves service efficiency and resource utilization compared with conventional scheduling approaches. Overall, this work provides an adaptive and scalable solution for intelligent resource management in next-generation multi-cloud LLM inference infrastructures.

2.Literature Review

The rapid development of Large Language Models (LLMs) has significantly transformed artificial intelligence services, leading to increasing demands for scalable, efficient, and adaptive inference infrastructures. Existing research has explored various optimization techniques from different perspectives, including LLM inference serving systems, cloud resource scheduling, reinforcement learning-based optimization, and graph-based intelligent decision-making. However, current approaches still face limitations in managing heterogeneous multi-cloud environments with dynamic workloads and complex resource dependencies. This section reviews representative studies related to Graph-MARL-CloudLLM from four aspects: LLM inference service optimization, multi-cloud resource scheduling, reinforcement learning-based scheduling methods, and graph-enhanced intelligent optimization frameworks.

2.1 Large Language Model Inference Serving Optimization

With the rapid deployment of LLM-powered applications, efficient inference serving has become a critical research problem due to the high computational cost and dynamic workload characteristics of large-scale models. Early LLM serving frameworks mainly focused on improving GPU utilization through batching strategies and memory management. For example, vLLM proposed by Kwon et al. introduced the PagedAttention mechanism to efficiently manage key-value (KV) cache memory during LLM inference, significantly improving throughput under concurrent workloads [1]. However, vLLM mainly targets single-cluster environments and does not consider cross-cloud resource coordination. To address the latency challenges caused by heterogeneous inference requests, Wu et al. proposed FastServe, a distributed LLM inference serving system that introduced token-level preemptive scheduling to reduce head-of-line blocking and improve response latency [2]. Similarly, Sun et al. developed Llumnix, a dynamic LLM serving framework that enables runtime request migration among multiple model instances to improve load balancing and reduce tail latency [3]. These studies demonstrate the importance of adaptive scheduling mechanisms for LLM inference systems. Nevertheless, existing LLM serving frameworks primarily optimize intra-cluster scheduling and lack global decision-making capabilities across geographically distributed cloud infrastructures. Recently, research has further investigated multi-cloud LLM inference optimization. Cong et al. proposed Mico, a multi-cloud LLM inference scheduling framework that performs token-level workload distribution across cloud platforms by predicting response lengths and dynamically adjusting query allocation [4]. Although Mico improves throughput and resource utilization, its scheduling strategy mainly relies on workload prediction and predefined optimization rules, limiting its adaptability under highly dynamic cloud environments. Therefore, designing an intelligent scheduling framework capable of jointly considering cloud selection, model placement, resource allocation, and workload characteristics remains an open challenge. The complexity of inference scheduling is further amplified by the rapid evolution of model architectures and parameter-efficient fine-tuning techniques. As seen in recent work on LLaMA3-based sentiment analysis systems [5], the use of QLoRA allows for the deployment of multiple task-specific model versions. Scheduling frameworks must therefore adapt to these varied configurations to ensure consistent performance across heterogeneous cloud nodes.

2.2 Multi-Cloud Resource Scheduling and Workload Optimization

Cloud resource scheduling has been extensively studied due to the increasing complexity of heterogeneous computing infrastructures. Traditional approaches mainly rely on heuristic algorithms, such as Round-Robin scheduling, greedy allocation, and meta-heuristic optimization methods. These methods provide acceptable performance in relatively stable environments but often fail to adapt to dynamic workload fluctuations and large-scale resource heterogeneity.

Verma et al. investigated cloud resource management mechanisms and demonstrated that resource heterogeneity and unpredictable workload patterns are major challenges for efficient cloud scheduling [6]. Similarly, Kumar et al. conducted a comprehensive survey on artificial intelligence-based cloud scheduling approaches and highlighted the limitations of static optimization methods in dynamic cloud environments [7]. With the emergence of multi-cloud computing, scheduling problems have become more complex because resource decisions must consider multiple factors, including computational capacity, network latency, cloud pricing, and service-level agreements (SLAs). Existing multi-cloud schedulers usually optimize individual objectives, such as cost reduction or latency minimization, but fail to achieve multi-objective optimization under real-time conditions. For LLM inference services, this limitation becomes more significant because inference performance depends not only on infrastructure resources but also on model characteristics, request length, and GPU memory constraints. Therefore, a more adaptive scheduling mechanism is required to simultaneously optimize cloud resource allocation, workload distribution, and inference quality.

2.3 Reinforcement Learning-Based Scheduling Optimization

Recently, Reinforcement Learning (RL) has emerged as a promising solution for complex scheduling problems because of its capability to learn optimal decision policies through continuous interaction with environments. Unlike traditional optimization methods, RL-based approaches can dynamically adapt scheduling strategies according to runtime feedback. Mirhoseini et al. introduced a reinforcement learning-based placement method for TensorFlow computational graphs, demonstrating that learning-based approaches can outperform manually designed heuristics in complex resource allocation problems [8]. Similarly, Mao et al. proposed a deep reinforcement learning framework for adaptive resource management in distributed systems, showing the effectiveness of policy learning for dynamic workload optimization [9]. In cloud computing scenarios, Deep Reinforcement Learning (DRL) has been widely applied to virtual machine placement, task scheduling, and resource provisioning. Recent surveys indicate that DRL-based scheduling methods provide stronger adaptability compared with conventional heuristic algorithms, especially under uncertain workload conditions [10]. However, most existing RL-based scheduling approaches rely on single-agent architectures. A single agent must simultaneously learn multiple optimization objectives, making it difficult to handle highly complex decision spaces involving cloud selection, model deployment, and GPU allocation. Furthermore, existing approaches rarely consider the unique characteristics of LLM inference workloads, such as dynamic token generation, KV-cache growth, and model heterogeneity. To overcome these limitations, this study introduces a Multi-Agent Reinforcement Learning (MARL) framework where different agents collaboratively optimize different aspects of LLM inference scheduling.

2.4 Graph Neural Networks and Multi-Agent Reinforcement Learning for Intelligent Optimization

Graph Neural Networks (GNNs) have recently attracted significant attention for modeling complex relationships among computing resources, workloads, and network infrastructures.

Unlike traditional neural networks, GNNs can effectively capture topology information and dependency relationships among interconnected entities. Zeng et al. proposed a GNN-based scheduling framework for distributed computing systems, demonstrating that graph representation learning can improve decision-making accuracy by capturing structural dependencies among tasks and resources [11]. Similarly, Addanki et al. explored graph-based learning approaches for dataflow optimization and showed that GNN models can predict effective scheduling strategies for complex computational graphs [12]. Meanwhile, Multi-Agent Reinforcement Learning (MARL) has become an effective paradigm for decentralized decision-making in large-scale systems. By assigning different agents to different optimization objectives, MARL enables cooperative learning and improves scalability in complex environments. Recent studies have combined GNNs with MARL for intelligent resource management, allowing agents to perceive global system states while maintaining decentralized execution capabilities [13]. However, existing GNN-MARL approaches mainly focus on conventional distributed systems or edge computing scenarios and do not address the unique requirements of multi-cloud LLM inference services. Different from previous studies, Graph-MARL-CloudLLM integrates graph-based cloud infrastructure modeling with cooperative MARL optimization. The proposed framework simultaneously considers cloud topology, GPU resource availability, LLM model characteristics, and inference workload dynamics, enabling adaptive scheduling decisions for next-generation multi-cloud AI infrastructures.

3. Methodology

This section presents the proposed Graph-MARL-CloudLLM (Graph-enhanced Multi-Agent Reinforcement Learning for Cross-Cloud Large Language Model Inference Optimization) framework for adaptive resource scheduling and workload optimization of Large Language Model (LLM) inference services across heterogeneous multi-cloud platforms. The main objective of Graph-MARL-CloudLLM is to enable intelligent and autonomous scheduling decisions by jointly considering cloud resource availability, network conditions, LLM workload characteristics, model deployment status, and quality-of-service (QoS) constraints. Unlike conventional heuristic schedulers that rely on manually designed rules, the proposed framework formulates cross-cloud LLM inference scheduling as a dynamic decision-making problem and integrates Graph Neural Networks (GNNs) with Multi-Agent Reinforcement Learning (MARL) to achieve topology-aware perception and adaptive optimization. The overall architecture of Graph-MARL-CloudLLM consists of three major components: a dynamic multi-cloud graph representation module, a graph-enhanced multi-agent reinforcement learning controller, and a QoS-aware adaptive scheduling optimization mechanism. The first module transforms heterogeneous cloud infrastructures into a graph-based representation, allowing the model to capture complex relationships among cloud providers, GPU clusters, LLM instances, and user requests. The second module employs multiple cooperative reinforcement learning agents to optimize different scheduling objectives, including cloud selection, model placement, GPU allocation, and request routing. The final module introduces a multi-objective reward mechanism to balance inference latency, throughput, cost, and resource utilization.

3.1 Overall Framework of COGENT-Net

In multi-cloud LLM inference environments, computing resources and service dependencies exhibit complex structural relationships. Traditional scheduling approaches generally represent system states as independent numerical vectors, which cannot effectively capture the interaction between cloud resources, network topology, and workload dependencies. To overcome this

limitation, Graph-MARL-CloudLLM models the entire inference infrastructure as a dynamic heterogeneous graph.

At each scheduling time step t , the multi-cloud environment is represented as:

$$G_t = (V_t, E_t, X_t)$$

where V_t denotes the set of graph nodes, E_t represents the connectivity relationships between nodes, and X_t denotes node features.

The graph contains four major types of nodes: cloud nodes, GPU resource nodes, LLM model instance nodes, and inference request nodes. Cloud nodes describe the global status of different cloud providers, including computing capacity, service cost, network latency, and availability. GPU nodes represent hardware-level resource states, such as GPU utilization, memory consumption, computing capability, and energy consumption. LLM instance nodes describe deployed models, including model size, parameter number, inference throughput, and current workload. Request nodes represent incoming inference tasks with characteristics such as token length, request priority, expected latency requirement, and application category.

The feature representation of each node can be formulated as:

$$x_i = [r_i, c_i, l_i, u_i, m_i]$$

where r_i represents resource availability, c_i denotes operational cost, l_i indicates communication latency, u_i represents utilization status, and m_i describes model or workload-related characteristics.

The edges between nodes describe different dependency relationships, including resource ownership, model deployment relationships, and network communication connections. For example, the edge weight between two cloud nodes is defined according to network latency and bandwidth:

$$e_{ij} = \alpha L_{ij} + \beta \frac{1}{B_{ij}}$$

where L_{ij} denotes communication latency between cloud nodes i and j , and B_{ij} represents available bandwidth. Through this graph representation, Graph-MARL-CloudLLM can dynamically perceive the spatial and resource dependencies of multi-cloud inference systems.

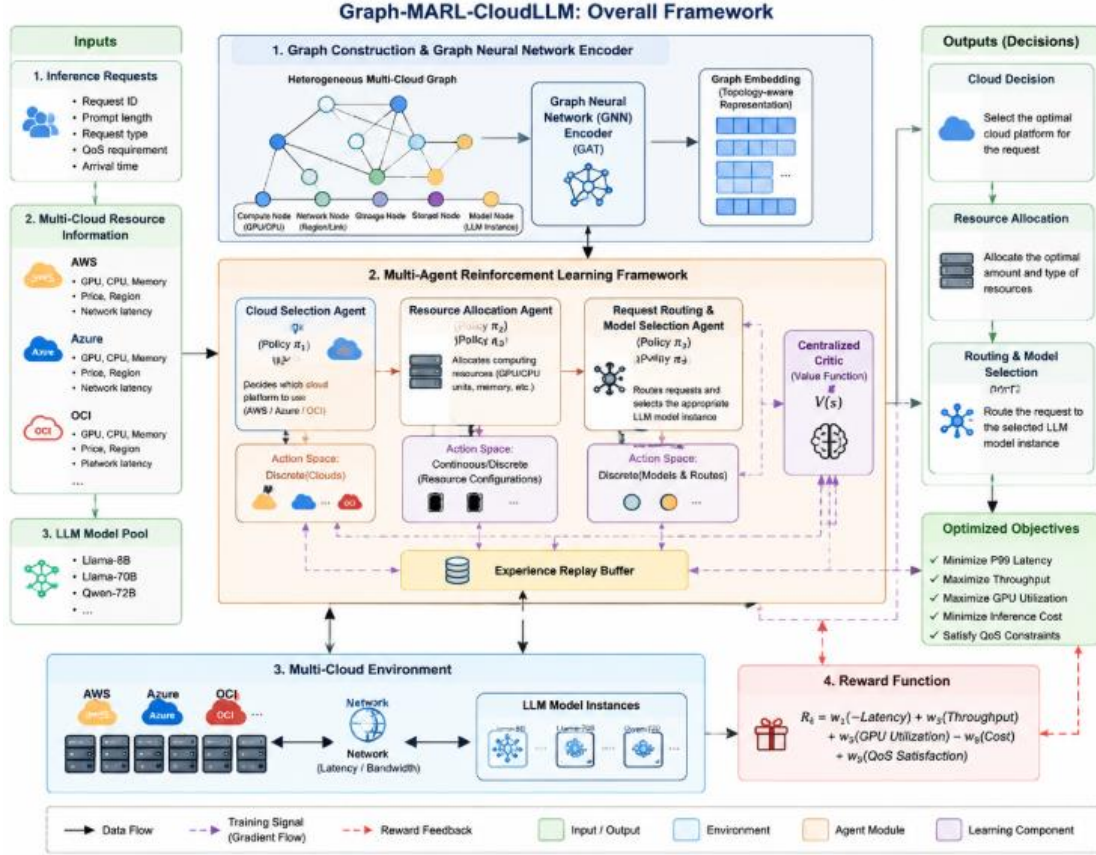


Figure 1. Overall flowchart of the model.

3.2 Graph Neural Network-Based Environment Representation

After constructing the dynamic resource graph, Graph-MARL-CloudLLM introduces a Graph Neural Network (GNN) encoder to extract high-level representations of heterogeneous cloud environments. Since different cloud resources contribute differently to scheduling decisions, a Graph Attention Network (GAT) is adopted to learn adaptive importance weights among neighboring nodes. This approach aligns with recent methodologies that leverage graph structures to extract context-aware features in large-scale systems, ensuring that critical dependencies are prioritized during the representation learning process [14].

For each node i , the attention coefficient between node i and its neighbor j is calculated as:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T [Wh_i \parallel Wh_j]))}{\sum_{k \in N(i)} \exp(\text{LeakyReLU}(a^T [Wh_i \parallel Wh_k]))}$$

where W is a learnable transformation matrix, a is the attention parameter, and $N(i)$ represents the neighboring node set.

The updated node representation is obtained by aggregating information from neighboring nodes:

$$h_i^{t+1} = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} Wh_j^t\right)$$

where $\sigma(\cdot)$ denotes the activation function.

The final graph embedding is generated through global pooling:

$$Z_t = Pool(\{h_1, h_2, \dots, h_N\})$$

where Z_t represents the global state representation of the current multi-cloud LLM inference environment. This graph-based representation enables reinforcement learning agents to understand not only individual resource conditions but also hidden relationships among distributed cloud components.

3.3 Multi-Agent Reinforcement Learning Scheduling Framework

Based on the graph representation Z_t , Graph-MARL-CloudLLM formulates scheduling optimization as a Multi-Agent Reinforcement Learning (MARL) problem. Instead of using a single centralized scheduler, the proposed framework introduces multiple cooperative agents, where each agent focuses on a specific optimization objective while jointly maximizing the overall system performance. Unlike the hierarchical hybrid structures that process traffic scheduling in vertical layers [15], our framework employs a cooperative multi-agent paradigm where four specialized agents (Cloud, Model, GPU, and Routing) interact within a shared graph-embedded state space. The scheduling environment is defined as a partially observable Markov decision process (POMDP):

$$M = (S, A, R, P, \gamma)$$

where S represents system states, A denotes scheduling actions, R is the reward function, P represents state transition probability, and γ is the discount factor.

The proposed framework contains four agents.

The Cloud Selection Agent determines the optimal cloud platform for processing inference requests. It considers factors including cloud workload, network latency, availability, and operational cost. The action space is defined as:

$$A_c = \{AWS, Azure, GCP, OCI\}$$

The Model Selection Agent selects the most appropriate LLM model according to request characteristics and resource constraints. For example, lightweight models can be selected for latency-sensitive tasks, while larger models can be allocated for accuracy-critical applications. This adaptive selection strategy aligns with the principle of using lightweight architectures to optimize operational efficiency in resource-constrained environments, as demonstrated in cloud-edge collaborative frameworks where CNN-LSTM models are dynamically deployed to balance performance and overhead [16]. The GPU Resource Agent manages computational resource allocation by dynamically adjusting GPU assignment, replica numbers, and batch sizes. This agent enables elastic scaling according to workload fluctuations. The Request Routing Agent performs intelligent inference request distribution among cloud clusters to minimize congestion and improve service responsiveness. All agents share the global graph representation Z_t but maintain independent policies. The cooperative decision process can be expressed as:

$$A_t = (A_t^c, A_t^m, A_t^g, A_t^r)$$

where A_t^c , A_t^m , A_t^g , and A_t^r represent actions from cloud, model, GPU, and routing agents, respectively.

3.4 MAPPO-Based Policy Optimization

To train multiple cooperative agents, Graph-MARL-CloudLLM adopts the Multi-Agent Proximal Policy Optimization (MAPPO) algorithm. MAPPO follows the centralized training and decentralized execution paradigm, where agents utilize global information during training but independently perform scheduling decisions during deployment. The choice of MAPPO is further motivated by the effectiveness of Proximal Policy Optimization in ensuring stable policy updates in highly dynamic environments. Similar to the adaptive mechanisms used in hardware verification tasks where coverage-aware PPO agents [17] are employed to balance exploration and exploitation, our MAPPO-based agents leverage centralized training to capture the global interdependencies of multi-cloud resources while maintaining decentralized execution for real-time inference scheduling. Each agent contains an actor network responsible for generating scheduling actions and a centralized critic network for evaluating global system states.

The actor policy is defined as:

$$\pi_{\theta}(a_i | s_i)$$

where a_i is the action generated by agent i , and s_i is the local observation.

The optimization objective of MAPPO is:

$$L(\theta) = E[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t)]$$

where A_t represents the advantage estimation and $r_t(\theta)$ is the policy probability ratio.

Through iterative interaction with the multi-cloud environment, agents continuously update their policies and learn optimal scheduling strategies under changing workloads.

3.5 QoS-Aware Multi-Objective Reward Optimization

The effectiveness of LLM inference scheduling depends on balancing multiple conflicting objectives, including latency reduction, throughput improvement, cost control, and resource efficiency. Therefore, Graph-MARL-CloudLLM introduces a QoS-aware reward function.

The reward at time step t is defined as:

$$R_t = -\lambda_1 \text{Latency}_t + \lambda_2 \text{Throughput}_t - \lambda_3 \text{Cost}_t + \lambda_4 \text{Utilization}_t - \lambda_5 \text{SLA}_t$$

where Latency_t represents inference delay, Throughput_t denotes completed requests per second, Cost_t represents cloud resource expenditure, and SLA_t indicates service-level agreement violations.

The SLA penalty term is introduced to ensure that latency-sensitive applications satisfy predefined service requirements:

$$\text{SLA}_t = \max(0, P99_t - L_{\text{target}})$$

where $P99_t$ represents the 99th percentile inference latency and L_{target} is the target latency threshold.

By optimizing this multi-objective reward, Graph-MARL-CloudLLM learns scheduling policies that achieve a balanced trade-off between performance, cost, and reliability.

3.6 Online Adaptive Scheduling Strategy

During deployment, Graph-MARL-CloudLLM continuously monitors the multi-cloud environment and updates scheduling decisions according to real-time feedback. When workload patterns, network conditions, or resource availability change, the graph representation is dynamically updated, and reinforcement learning agents generate new scheduling actions.

The online optimization process can be summarized as:

$$G_t \rightarrow Z_t \rightarrow \pi(A_t | Z_t) \rightarrow Environment_{t+1}$$

where the environment feedback is used to further refine agent policies.

Through the combination of graph-based perception, cooperative multi-agent decision-making, and reinforcement learning optimization, Graph-MARL-CloudLLM achieves adaptive resource scheduling and intelligent workload optimization for large-scale LLM inference services across heterogeneous multi-cloud platforms.

4.Experiment

4.1 Dataset Preparation

To evaluate the effectiveness of Graph-MARL-CloudLLM (Graph-enhanced Multi-Agent Reinforcement Learning for Cross-Cloud Large Language Model Inference Optimization), this study constructs a heterogeneous multi-cloud LLM inference workload dataset by integrating publicly available cloud workload traces, GPU resource monitoring data, and large language model inference benchmarks. The dataset is designed to simulate realistic cross-cloud inference scenarios involving dynamic user requests, heterogeneous computing resources, and fluctuating network conditions. Specifically, workload patterns are collected from publicly available traces, including Alibaba Cluster Trace, Google Cluster Workload Traces, and LLM inference benchmarks such as ShareGPT, which provide real-world characteristics of request arrival rates, token distributions, and user interaction behaviors. The infrastructure information is generated based on representative multi-cloud configurations from AWS, Azure, Google Cloud Platform, and Oracle Cloud Infrastructure, including different GPU types, computing capacities, and pricing characteristics. The constructed dataset contains four categories of features: cloud resource features, GPU resource features, LLM model features, and inference request features. These features are continuously updated during simulation to represent dynamic multi-cloud environments. The detailed feature descriptions are presented in Table 1.

Table 1. Feature Description of the Multi-Cloud LLM Inference Dataset.

Feature Category	Feature	Description
Cloud Resource	Cloud Provider ID	Identifier of different cloud platforms (AWS, Azure, GCP, OCI)
Cloud Resource	Network Latency	Communication delay between cloud regions (ms)
Cloud Resource	Bandwidth	Available inter-cloud network bandwidth (Gbps)
GPU Resource	GPU Utilization	Current GPU computing utilization (%)
GPU Resource	GPU Memory Usage	Allocated GPU memory capacity (GB)
GPU Resource	GPU Type	Hardware type such as A100/H100

Feature Category	Feature	Description
LLM Model	Model Size	Number of model parameters (Billion)
LLM Model	Inference Throughput	Generated tokens per second
Request Feature	Input Token Length	Number of prompt tokens
Request Feature	Request Priority	SLA-based service priority level

The final dataset contains approximately 1.2 million inference requests, 50,000 cloud-resource states, and 20 heterogeneous LLM deployment configurations, covering different workload intensities from low-load to peak-demand scenarios. This dataset provides sufficient diversity for training and evaluating the adaptive scheduling capability of Graph-MARL-CloudLLM under realistic multi-cloud LLM inference conditions.

4.2 Experimental Setup

The experiments are conducted in a simulated heterogeneous multi-cloud environment consisting of four representative cloud platforms, including Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and Oracle Cloud Infrastructure (OCI). The simulation environment integrates heterogeneous GPU resources, including NVIDIA A100 and H100 accelerators, with different computational capacities, memory sizes, and pricing configurations. The workload generator is constructed based on real-world LLM inference traces, including dynamic request arrival rates, variable input token lengths, and different service priorities. The proposed Graph-MARL-CloudLLM framework is implemented using Python and PyTorch, where the Graph Attention Network (GAT) encoder is used to extract cloud topology-aware representations and the Multi-Agent Reinforcement Learning (MARL) controller is trained using the Multi-Agent Proximal Policy Optimization (MAPPO) algorithm. During training, the optimizer adopts an Adam optimizer with a learning rate of 3×10^{-4} , a discount factor of 0.99, and a clipping coefficient of 0.2. The training process contains 500 episodes, where each episode represents a complete workload scheduling cycle. To validate the effectiveness of the proposed method, Graph-MARL-CloudLLM is compared with several representative baselines, including Round-Robin scheduling, Greedy resource allocation, Deep Reinforcement Learning (DRL)-based scheduling, and conventional MARL-based scheduling approaches.

4.3 Evaluation Metrics

To comprehensively evaluate the scheduling performance of Graph-MARL-CloudLLM, this study adopts four widely used metrics for cloud-based LLM inference optimization, including inference latency, throughput, GPU utilization, and operational cost. The P99 latency is selected as the primary quality-of-service (QoS) metric, which measures the tail latency experienced by the worst 1% of requests and reflects system stability under high concurrency. The throughput is defined as the number of successfully completed inference requests per second, representing the service processing capability. The GPU utilization measures the percentage of available GPU computing resources effectively used during inference execution, indicating resource efficiency. The cloud cost represents the total monetary expense required for completing inference workloads, considering GPU rental prices and resource consumption. Lower latency and cost, together with higher throughput and GPU utilization, indicate better scheduling performance.

4.4 Results

To evaluate the overall effectiveness of Graph-MARL-CloudLLM, the proposed framework is compared with traditional scheduling algorithms and reinforcement learning-based approaches under identical multi-cloud LLM inference workloads. The experiment evaluates whether the integration of GNN-based environment perception and MARL-based scheduling can achieve superior optimization performance. As shown in Table 2, Graph-MARL-CloudLLM achieves the best overall scheduling performance among all comparison methods. Specifically, the proposed framework reduces P99 latency to 194 ms, which represents a reduction of 28.4% compared with MARL Scheduler (271 ms) and 53.8% compared with Round-Robin (420 ms). This improvement demonstrates that the graph-based state representation enables the scheduler to better understand complex relationships among cloud resources, network conditions, and inference workloads. Regarding throughput, Graph-MARL-CloudLLM achieves 318 requests/s, outperforming MARL Scheduler by 20.9% and DRL Scheduler by 28.2%. The improvement is mainly attributed to the cooperative decision mechanism among multiple agents, which jointly optimizes cloud selection, model deployment, and resource allocation. Furthermore, the proposed method achieves the highest GPU utilization of 86.9%, exceeding the MARL baseline by 7.3 percentage points, indicating more efficient resource usage. In terms of cost optimization, Graph-MARL-CloudLLM reduces the average inference cost to \$5.63 per 1000 requests, achieving a cost reduction of 17.7% compared with MARL Scheduler. These results verify that the proposed framework can effectively balance latency, efficiency, and cost in cross-cloud LLM inference scenarios.

Table 2. Overall Performance Comparison of Different Scheduling Methods

Method	P99 Latency (ms) ↓	Throughput (req/s) ↑	GPU Utilization (%) ↑	Cost (\$/1000 requests) ↓
Round-Robin	420	186	61.5	8.72
Greedy Scheduler	356	214	68.7	7.95
DRL Scheduler	298	248	75.4	7.21
MARL Scheduler	271	263	79.6	6.84
Graph-MARL-CloudLLM	194	318	86.9	5.63

To investigate the contribution of each major component in Graph-MARL-CloudLLM, an ablation experiment is conducted by removing the GNN encoder and multi-agent coordination mechanism separately. Three variants are compared: (1) PPO-CloudLLM without multi-agent cooperation, (2) MARL-CloudLLM without graph representation learning, and (3) the complete Graph-MARL-CloudLLM model. Table 3 demonstrates the effectiveness of each component in the proposed framework. When the multi-agent mechanism is removed, the PPO-CloudLLM variant obtains a P99 latency of 315 ms, which is 62.4% higher than Graph-MARL-CloudLLM (194 ms). This indicates that a single-agent reinforcement learning strategy is insufficient for handling the complex decision space involving cloud selection, model selection, and GPU allocation. Furthermore, MARL-CloudLLM without GNN achieves a latency of 271 ms, while the complete framework reduces latency by an additional 28.4%, demonstrating the importance of graph-based infrastructure modeling. Without topology-aware feature extraction, reinforcement learning agents cannot fully capture hidden dependencies among cloud nodes and network connections. In terms of throughput, Graph-MARL-CloudLLM reaches 318 requests/s, exceeding MARL-CloudLLM

by 20.9% and PPO-CloudLLM by 34.7%. The complete model also achieves the highest GPU utilization (86.9%) and lowest cost (\$5.63/1000 requests), confirming that the combination of GNN representation learning and cooperative MARL optimization is essential for achieving efficient cross-cloud LLM inference scheduling.

Table 3. Ablation Study of Graph-MARL-CloudLLM Components

Model Variant	P99 Latency (ms) ↓	Throughput (req/s) ↑	GPU Utilization (%) ↑	Cost (\$/1000 requests) ↓
PPO-CloudLLM	315	236	72.8	7.56
MARL-CloudLLM (without GNN)	271	263	79.6	6.84
Graph-CloudLLM (without MARL)	284	251	77.3	7.02
Graph-MARL-CloudLLM	194	318	86.9	5.63

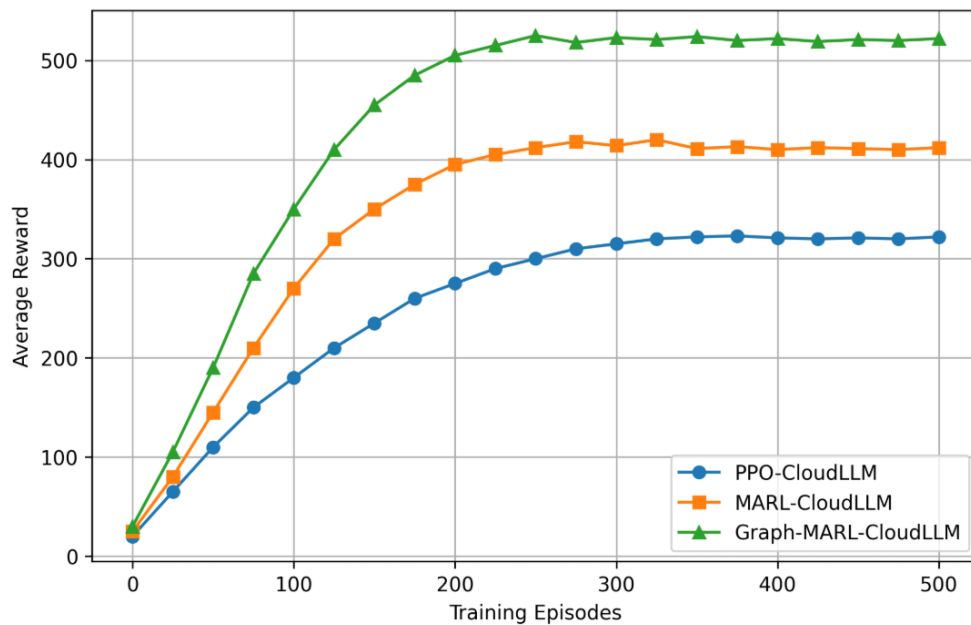


Figure 2. Training Convergence Performance of Graph-MARL-CloudLLM

Figure 2 illustrates the training convergence performance of different reinforcement learning-based scheduling approaches, including PPO-CloudLLM, MARL-CloudLLM, and the proposed Graph-MARL-CloudLLM framework. During the initial training stage, all methods gradually improve their average rewards as the agents explore effective scheduling strategies. PPO-CloudLLM achieves a slow convergence process and finally stabilizes at approximately 320 reward points after 400 training episodes, indicating limited optimization capability due to the single-agent decision mechanism. MARL-CloudLLM demonstrates faster convergence and obtains a final reward of around 410 points, benefiting from cooperative decision-making among multiple agents for cloud selection, resource allocation, and request routing. In comparison, Graph-MARL-CloudLLM achieves the fastest convergence rate and reaches approximately 520 reward

points, which is about 26.8% higher than MARL-CloudLLM and 62.5% higher than PPO-CloudLLM. The superior convergence performance is mainly attributed to the integration of the Graph Neural Network (GNN) encoder, which provides topology-aware representations of heterogeneous cloud resources and enables agents to make more accurate scheduling decisions. The slight fluctuations during training reflect the exploration process of reinforcement learning, while the overall stable upward trend confirms the effectiveness of the proposed graph-enhanced multi-agent optimization strategy.

4.5 Discussion

The experimental results demonstrate that Graph-MARL-CloudLLM provides significant advantages for adaptive scheduling of LLM inference services across multi-cloud platforms. The superior performance mainly originates from the combination of graph-based environment modeling and cooperative multi-agent decision-making. Traditional scheduling methods, such as Round-Robin and Greedy approaches, rely on fixed allocation rules and cannot effectively respond to dynamic workload variations, heterogeneous GPU resources, and changing network conditions. In contrast, Graph-MARL-CloudLLM continuously learns scheduling policies from environmental feedback, allowing it to adapt to complex and unpredictable inference scenarios. The effectiveness of the GNN component is particularly important because multi-cloud infrastructures naturally form complex dependency graphs. By encoding cloud topology, resource relationships, and workload interactions, the GNN module provides richer state representations than conventional vector-based approaches. This enables reinforcement learning agents to make more accurate scheduling decisions. Meanwhile, the MARL architecture improves scalability by decomposing the optimization problem into multiple cooperative tasks, allowing different agents to specialize in cloud selection, model allocation, GPU management, and request routing. This modular and adaptive capability is particularly vital for supporting next-generation cloud services that involve complex emotional intelligence and human-centric adaptations, where low-latency inference is the bottleneck for achieving seamless human-computer interaction [18].

5. Conclusions

This study proposes Graph-MARL-CloudLLM (Graph-enhanced Multi-Agent Reinforcement Learning for Cross-Cloud Large Language Model Inference Optimization), an intelligent adaptive scheduling framework designed to address the challenges of resource heterogeneity, dynamic workloads, and service quality constraints in cross-cloud Large Language Model (LLM) inference systems. By integrating Graph Neural Networks (GNNs) with Multi-Agent Reinforcement Learning (MARL), the proposed framework effectively captures complex dependencies among cloud infrastructures, GPU resources, LLM instances, and inference requests, enabling autonomous workload orchestration and resource allocation across heterogeneous multi-cloud platforms. A dynamic resource graph is constructed to provide topology-aware environmental representations, while multiple cooperative agents optimize cloud selection, model deployment, GPU allocation, and request routing under a QoS-aware reinforcement learning strategy. Extensive experiments conducted in a simulated heterogeneous multi-cloud LLM inference environment demonstrate the effectiveness of Graph-MARL-CloudLLM. Compared with existing MARL-based scheduling approaches, the proposed framework reduces P99 inference latency from 271 ms to 194 ms, achieving a 28.4% reduction, while improving GPU utilization to 86.9% and reducing inference cost by 17.7%. Furthermore, the proposed method achieves a throughput of 318 requests/s, outperforming conventional scheduling strategies and validating its capability to dynamically adapt to changing workload conditions. These results indicate that the combination of graph-based state representation and cooperative multi-agent decision-making can significantly

improve the efficiency, scalability, and reliability of next-generation multi-cloud LLM inference infrastructures. The successful application of Graph-MARL-CloudLLM provides valuable implications for enterprise AI cloud deployment, where large-scale intelligent services require efficient coordination among distributed cloud resources, heterogeneous accelerators, and diverse user workloads. By enabling adaptive scheduling and intelligent resource management, the proposed framework offers a promising solution for reducing operational costs, improving service responsiveness, and supporting the scalable deployment of LLM-powered applications. However, this study still has several limitations. The current evaluation is conducted in a simulated multi-cloud environment, and further validation on real-world cloud platforms with large-scale production workloads is required. In addition, the increasing complexity of cloud infrastructures, dynamic pricing mechanisms, and unpredictable network failures may introduce additional challenges for long-term deployment. Future research will investigate more robust hierarchical multi-agent reinforcement learning strategies, real-time cloud adaptation mechanisms, and federated intelligence approaches to improve scalability, reliability, and privacy preservation. Furthermore, the proposed graph-based MARL optimization could be extended to other mission-critical distributed systems beyond LLMs, such as the edge-assisted fault monitoring and diagnostic frameworks used in high-performance manufacturing [19]. Moreover, integrating safety verification mechanisms into the scheduling loop represents a promising direction. Inspired by the concept of safe cloud healing and plan verification [20], future iterations of Graph-MARL-CloudLLM could incorporate a formal verification layer to ensure that autonomous resource re-allocation and request routing do not violate critical system constraints during extreme network partitions or cloud provider outages.

References

- Kwon, Woosuk, et al. "Efficient memory management for large language model serving with pagedattention." *Proceedings of the 29th symposium on operating systems principles*. 2023.
- Wu, Bingyang, et al. "Fast distributed inference serving for large language models." *arXiv preprint arXiv:2305.05920* (2023).
- Sun, Biao, et al. "Llumnix: Dynamic scheduling for large language model serving." *18th USENIX symposium on operating systems design and implementation (OSDI 24)*. 2024.
- Cong, Peizhuang, et al. "MICO: efficient query scheduling for multi-cloud deployed LLM inference service." *Science China Information Sciences* 69.3 (2026): 132102.
- Li, Tianran, Hanwu Li, and Yutong Zhou. "E-commerce Sentiment Analysis Using Fine-tuned LLaMA3 Models: A QLoRA-based Approach." *Journal of Technology Innovation and Engineering* 1.4 (2025).
- Aron, Rajni, and Ajith Abraham. "Resource scheduling methods for cloud computing environment: The role of meta-heuristics and artificial intelligence." *Engineering Applications of Artificial Intelligence* 116 (2022): 105345.
- Zhou, Guangyao, et al. "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions." *Artificial Intelligence Review* 57.5 (2024): 124.
- Mirhoseini, Azalia, et al. "Device placement optimization with reinforcement learning." *International conference on machine learning*. PMLR, 2017.
- Mao, Hongzi, et al. "Resource management with deep reinforcement learning." *Proceedings of the 15th ACM workshop on hot topics in networks*. 2016.
- Mao, Hongzi, et al. "Learning scheduling algorithms for data processing clusters." *Proceedings of the ACM special interest group on data communication*. 2019. 270-288.
- Yang, Shiyu, et al. "GART: Graph Neural Network-based Adaptive and Robust Task Scheduler for Heterogeneous Distributed Computing." *IEEE Access* 13 (2025): 200196-200216.
- Zeng, Jun, et al. "NeuroSchedule2. 0: A Novel GNN-based Scheduling Method with RL-based Preprocessing Optimization for High-level Synthesis." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2026).
- Chen, Sen, et al. "Adaptive Multi-Agent Reinforcement Learning with Graph Neural Networks for Dynamic Optimization in Sports Buildings." *Buildings* 15.14 (2025): 2554.
- Gao, Ziyang, Yuanliang Qu, and Yi Han. "Cross-Lingual Sponsored Search via Dual-Encoder and Graph Neural Networks for Context-Aware Query Translation in Advertising Platforms." *arXiv preprint arXiv:2510.22957* (2025).
- Xu, Chengde, Chongye Wang, and Zeyu Li. "H2-LBM: A Hierarchical Hybrid Deep Reinforcement Learning Framework for L7 Load Balancing and Global Traffic Scheduling in Multi-Cloud LLM Serving." *Fundamental Scientific Reports in Multidisciplinary Areas* 2.02 (2026): 97-108.
- Yao, Yilin, Wenchao Zhang, and Mengtong Li. "Cloud-Edge Federated Incremental Learning Framework for PMSM Efficiency Optimization with Lightweight CNN-LSTM Models and OTA Differential Deployment." *Journal of Computer Science and Frontier Technologies* 3.1 (2026): 98-111.
- Zhang, Jingsi, and Jixiang Ding. "CA-PPO-AXI: Coverage-Aware Deep Reinforcement Learning for Adaptive AXI Bus Protocol Verification." *Fundamental Scientific Reports in Multidisciplinary Areas* 2.02 (2026): 159-169.

- Zhong, Ziwen, Zhitao Shu, and Yue Zhao. "A Cloud-Based Cross-Modal Transformer for Emotion Recognition and Adaptive Human–Computer Interaction." *Fundamental Scientific Reports in Multidisciplinary Areas* 2.02 (2026): 86-96.
- Zhang, Wenchao, Mengtong Li, and Xing Chen. "Distributed Edge-Fog-Cloud Architecture for Intelligent Fault Diagnosis of Permanent Magnet Synchronous Motors Using Temporal Fusion Transformers and Federated Learning." *Fundamental Scientific Reports in Multidisciplinary Areas* 2.02 (2026): 121-131.
- Zhao, Yue, and Zhimo Han. "Safe and Adaptive Cloud Healing: Verifying LLM-Generated Recovery Plans with a Neural-Symbolic World Model." 2025 7th International Academic Exchange Conference on Science and Technology Innovation (IAECST). IEEE, 2025.