Cached Model-as-a-Resource: Provisioning Large Language Model Agents for Edge Intelligence in Space–Air–Ground Integrated Networks

Minrui Xu, *Member, IEEE*, Dusit Niyato, *Fellow, IEEE*, Hongliang Zhang, *Senior Member, IEEE*, Jiawen Kang, *Senior Member, IEEE*, Zehui Xiong, *Senior Member, IEEE*, Shiwen Mao, *Fellow, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*—Edge intelligence in space-air-ground integrated networks (SAGINs) can enable worldwide network coverage beyond geographical limitations for users to access ubiquitous and low-latency intelligence services. Facing global coverage and complex environments in SAGINs, edge intelligence can provision large language models (LLMs) agents for users via edge servers at ground base stations (BSs) or cloud data centers relayed by satellites. As LLMs with billions of parameters are pre-trained on vast datasets, LLM agents have few-shot learning capabilities, e.g., chain-of-thought (CoT) prompting for complex tasks, which raises a new trade-off between resource consumption and performance in SAGINs. In this paper, we propose a joint caching and inference framework for edge intelligence to provision sustainable and ubiquitous LLM agents in SAGINs. We introduce "cached model-as-a-resource" for offering LLMs with limited context windows and propose a novel optimization framework, i.e., joint model caching and inference, to utilize cached model resources for provisioning LLM agent services along with communication, computing, and storage resources. We design "age of thought" (AoT) considering the CoT prompting of LLMs, and propose a least AoT cached model replacement algorithm for optimizing the provisioning cost. We propose a deep Q-network-based modified second-bid (DQMSB) auction to incentivize satellite/ground network operators in real-time, which can enhance allocation efficiency by 23% while guaranteeing strategy-proofness and being free from adverse selection.**

***Index Terms*—Space-air-ground integrated networks (SAGINs), edge intelligence, large language model (LLM) agents, auction theory, deep reinforcement learning (DRL).**

I. Introduction

SPACE-AIR-GROUND integrated networks (SAGINs) provide global network coverage and enable real-time edge intelligence services such as image recognition and data analysis [1]. Beyond terrestrial communication systems constrained by limited network capacity and ground-based station (BS) coverage, satellites relay computing services from cloud data centers and supply seamless connectivity in harsh environments, e.g., oceans and mountains [2]. Edge intelligence in SAGINs supports smart-ocean applications that range from real-time aquaculture monitoring to AI assistants for passengers, crew, and fishermen [3]. Recent advances in large language models (LLMs) [4], [5], [6] greatly enhance the capabilities of edge intelligence in SAGINs, enabling AI agents based on LLMs, i.e., LLM agents, to tackle unseen and complex reasoning tasks with various data modalities [7]. Moreover, SAGINs can provision low-latency and privacy-preserving LLM agent services [7] at edge servers of ground BSs or at cloud data centers relayed by satellites, which act as autonomous assistants for daily life and work.

Built on billion-parameter models pre-trained on Internet-scale corpora, LLM agents support few-shot learning [8], [9], encompassing in-context learning (ICL) for unseen tasks, chain-of-thought (CoT) prompting for complex reasoning, and role-playing under specific instructions. Although training and inference demand substantial computing resources, users with low-end mobile devices in SAGINs can invoke LLM services from edge servers at ground BSs or via satellite-backhauled cloud data centers. Therefore, heterogeneous deployment of LLM services in SAGINs reduces latency and safeguards user privacy [10], [11]. However, resource-limited edge servers

Received 31 May 2024; revised 27 July 2025; accepted 8 November 2025; approved by IEEE Transactions on Networking Editor S. Kompella. Date of publication 14 January 2026; date of current version 23 January 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62572132 and Grant U22A2054; in part by Guangdong Provincial Natural Science Foundation General Program under Grant 2025A1515010137; in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A151514 0137; in part by the Seatrium New Energy Laboratory; in part by Singapore Ministry of Education (MOE) Tier 1 under Grant RT5/23 and Grant RG24/24; in part by Nanyang Technological University (NTU) Centre for Computational Technologies in Finance (NTU-CCTF); in part by the Research Innovation and Enterprise (RIE) 2025 Industry Alignment Fund–Industry Collaboration Projects (IAF-ICP), administered by the Agency for Science, Technology and Research (A*STAR) under Award I2301E0026; in part by NSF under Grant ECCS-2302469; in part by Amazon; and in part by Japan Science and Technology Agency (JST) Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE) under Grant JPMJAP2326. *(Corresponding author: Jiawen Kang.)*

Minrui Xu and Dusit Niyato are with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: minrui001@e.ntu.edu.sg; dniyato@ntu.edu.sg).

Hongliang Zhang is with the School of Electronics, Peking University, Beijing 100871, China (e-mail: hongliang.zhang92@gmail.com).

Jiawen Kang is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China (e-mail: kavinkang@gdut.edu.cn).

Zehui Xiong is with the School of Electronics, Electrical Engineering and Computer Science (EEECS), Queen's University Belfast, BT7 1NN Belfast, U.K. (e-mail: z.xiong@qub.ac.uk).

Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smao@ieee.org).

Zhu Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea (e-mail: hanzhu22@gmail.com).

Digital Object Identifier 10.1109/TON.2025.3649068

cannot host every model concurrently [5]. Furthermore, few-shot performance is bounded by the context-window size dictated by each model architecture [12]. During inference, accumulated tokens initially enhance output but, once the window is filled [13], the quality of LLM responses deteriorates markedly.

As the context windows of running LLMs can be depleted during provisioning LLM agent services, the cached models at edge servers of ground BSs should be considered as an unexplored type of resource analogous to conventional communication, computing, and storage resources [5]. To minimise provisioning cost, network operators must coordinate these local LLMs by accounting not only for hardware constraints but also for the context availability. Model caching, analogous to content caching, serves as an optimization framework for edge intelligence that lowers service latency and overall resource consumption. An effective framework must also incorporate the few-shot learning capability of LLMs [8], since each additional demonstration modifies both resource usage and inference quality. Furthermore, the economic value of service opportunities tends to be positively correlated across operators. Satellites, acting solely as relays between end users and cloud data centres, receive limited cost/performance feedback and therefore face significant information asymmetry relative to terrestrial BSs. This asymmetry can give rise to adverse selection [14] and result in inefficient operator allocation within real-time mechanisms. For instance, in maritime satellite communications, satellites typically offer flat-rate data plans without direct visibility into per-task GPU resource consumption, unlike coastal ground stations that precisely monitor real-time resource usage [1].

To address these challenges, in this paper, we propose a joint model-caching and inference framework that delivers sustainable, low-latency, and privacy-aware LLM-agent services throughout SAGINs. Depending on user location and resource conditions, LLMs execute on edge servers collocated with ground BSs or on cloud data centers reached via satellite or terrestrial relays. Ground BSs therefore furnish nearby users with prompt responses, whereas satellites extend coverage to remote oceans and mountains. To raise service quality, we elevate cached LLMs to first-class resources and formulate a joint caching-and-inference optimisation. Exploiting LLMs' few-shot capability, we introduce the age-of-thought (AoT) metric to gauge the freshness and coherence of intermediate reasoning states. A least-AoT (LAoT) cache-replacement policy then removes the model with the largest AoT, thereby minimising provisioning cost for operators. Finally, we integrate a modified second-bid auction with a deep Q-network (DQMSB) that adaptively tunes the price-scaling factor, remains strategy-proof, and eliminates adverse selection in satellite resource allocation.

Our main contributions can be summarized as follows.

- We formulate a novel optimization framework for edge intelligence, i.e., the joint model caching and inference framework, to provision sustainable and ubiquitous LLM agents with satellites and ground BSs in SAGINs.
- In this framework, for the first time, we propose the concept of "cached model-as-a-resource" to implement edge intelligence, where cached models are regarded as a type of resource similar to conventional communication, computing, and storage resources, at edge servers at ground BSs, and cloud datacenters in SAGINs.
- We formulate the LLM agent provisioning problem for ground BSs to minimize total system cost under resource and coverage constraints. To tackle this problem effectively, we design a novel least AoT model caching algorithm to schedule loading and eviction of LLMs using the AoT, evaluating the relevance and coherence of intermediate thoughts in context windows.
- To maximize the revenue of network operators in provisioning high-quality LLM agent services, we propose the DQMSB auction, which can guarantee free of adverse selection and be fully strategy-proof, by using DRL to select the optimal pricing scaling factor.

The remaining sections of this paper are organized as follows. In Section II, we present a review of related work. In Section III, we describe the system model for provisioning LLM agents in SAGINs. In Section IV, we formulate the problem, propose the model caching algorithm, and design the market. In Section V, we propose the DQMSB auction. In Section VI, we present the simulation experiments. Finally, we conclude this paper in Section VII.

II. Related Works

A. Edge Intelligence in Space-Air-Ground Integrated Networks

Provisioning AI services in SAGINs can significantly enhance the intelligent configuration and control of SAGINs to adapt to their environment, improving various performance metrics such as latency, energy usage, bandwidth, and real-time adaptability [1]. Xu et al. in [15] introduce a cloud-edge aggregated artificial intelligence architecture that leverages the on-orbit lightweight 5G core and edge computing platform provided by the Tiansuan constellation. For mission-critical 6G services, Hou et al. in [16] propose a three-layer architecture in SAGINs for ultra-reliable and low-latency edge intelligence that includes unikernel-based ultra-lightweight virtualization and microservice-based paradigms for prompt response and improved reliability. Considering the time-varying characteristics of content sources and the dynamic demands of users, Qin et al. in [17] propose a content service-oriented resource allocation algorithm that aims to achieve a stable matching based on users' preferences for SAGINs.

B. Large Language Models for Edge Intelligence

In literature, LLMs are an essential part of next-generation edge intelligence systems, which have been leveraged to design, analyze, and optimize edge intelligence [18], [19]. For instance, Du et al. in [20] investigate the potential of LLMs as a valuable tool for FPGA-based wireless system development. Furthermore, Cui et al. in [21] introduce LLMind, an AI framework that integrates LLMs with domain-specific AI modules and IoT devices for executing complex tasks. In multi-agent systems for 6G communications, Jiang et al. [22] demonstrated the effectiveness of LLMs in collaborative data retrieval, planning, and reflection through a semantic communication case study. Considering the issues that traditional deep offloading architectures are facing several issues, including heterogeneous constraints, partial perception, uncertain generalization, and lack of traceability, Dong et al. [23] propose an LLM-based offloading framework that utilizes LLMs for offloading decisions, addressing issues like heterogeneous constraints and uncertain generalization. Nevertheless, the execution of LLMs usually requires enormous computing resources, which are infeasible for edge environments. Therefore, considering efficient training and inference architecture in 6G networks, Lin et al. in [24] explore feasible techniques such as split learning/inference, parameter-efficient fine-tuning, quantization, and parameter-sharing inference for pushing LLMs to the edge. Nevertheless, existing studies seldom examine how the freshness of cached LLMs affects network-wide cost; none of them couple freshness with spectrum, computing, and storage constraints in a unified optimization.

C. Auction Design for SAGINs

Auctions are efficient and effective methods for real-time network resource allocation in SAGINs [25], [26], [27]. In civil aircraft augmented SAGINs, Chen et al. [28] propose a truthful double auction for device-to-device (D2D) communications and a reverse auction mechanism for spectrum sharing. For lightweight blockchain-based SAGINs, Yang et al. in [29] propose a secure sequential Vickrey auction mechanism to ensure secure and reliable spectrum sharing within the SAGINs. However, the current auctions treat communication and computing costs as exogenous and overlook the fact that valuation depends on model freshness and token-level usage statistics, which are observable only at ground stations and not at satellites. This is a gap that motivates our deep-Q-network modified-second-bid (DQMSB) design. Therefore, in this paper, we propose DQMSB that can mitigate adverse selection and increase market efficiency for SAGINs with DQN-based price scaling factor selection.

III. System Model

For edge intelligence in SAGINs, each group of users would like to utilize LLM agents based on one or several LLMs. In the system, each group of users can use one LLM agent as their active assistant because their attention is limited depending on their preferences and current tasks. In SAGINs, a group of users needs to select network providers, including satellites and ground BSs, to access LLM agent services. As shown in Fig. 1, the system consists of $N+1$ network operators, including one or several Low Earth Orbit (LEO) satellites in orbit and multiple ground BSs equipped with edge servers, all connected to the cloud data center via backhaul links. The set of network operators is represented by $\mathcal{N} = \{0, 1, \ldots, N\}$, where the LEO satellite is represented by 0 and the set of BSs is represented by $\{1, \ldots, N\}$. The edge servers at ground BSs can execute LLM agent services for users while the rest of the services can be offloaded to cloud data centers with the relay of satellites or ground BSs. We use the set $\mathcal{I} = \{1, 2, \ldots, I\}$ to denote the available LLM agent services based on the set of LLMs $\mathcal{M} = \{1, \ldots, M\}$. As LLMs are capable of performing multiple downstream tasks in LLM agent services simultaneously, it is considered that the number of LLM agent services is far greater than the number of LLMs [7], i.e., $I \gg M$. In the group of users $\mathcal{U}_n$ covered by network operator n, $R_n^t = \{R_{n,i,m}^t | i \in \mathcal{I}, m \in \mathcal{M}\}$ is used to represent the number of requests generated by LLM agent service i to execute LLM m for its specific functions, including planning, memory, tool-using, and embodied actions. Initially, the size of input data of LLM agent service i can be denoted as d_i. Additionally, the configuration of LLM m consists of the amount of runtime GPU memory, which is proportion to model size s_m, the computation required per token e_m, and the size of context window w_m.

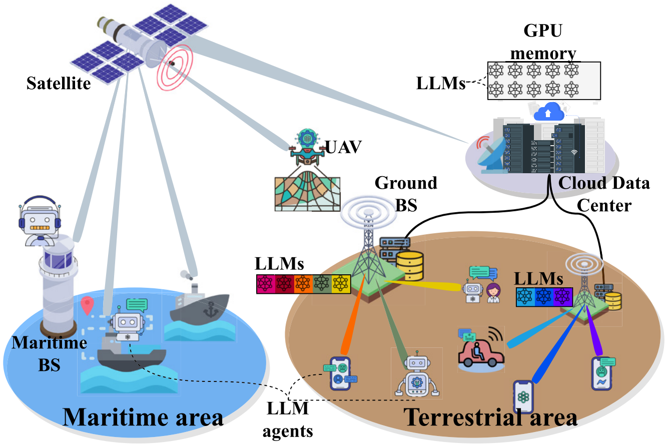


Fig. 1. Joint caching and inference framework for provisioning large language model (LLM) agents in SAGINs.

A. Coverage Time Model

The coverage time model for LEO satellite networks addresses the dynamic positioning of satellites concerning users. Unlike terrestrial networks, whose infrastructure remains stationary, LEO satellites exhibit constant motion, necessitating that users establish connections based on specific geometric metrics [2]. These metrics include the altitude l of the LEO satellite orbit above the mobile user, the Earth's radius E, and the slant distance s from users to the LEO satellites. The elevation angle θ^e, delineating the line of sight between a mobile user and an LEO satellite, is determined by $\theta^e = \arccos\left(\frac{E+l}{s}\right) \cdot \sin\theta^g$, where θ^g represents the geocentric angle covering the LEO satellite's service area, calculated as $\theta^g = \arccos\left(\frac{E}{E+l}\right) \cdot \cos\theta^e - \theta^e$. Let v^S denote the velocity of LEO satellite 0. The maximal communication duration T_0^S between a mobile user and the LEO satellite is given by

$$T_0^S = \frac{L}{v^S}, \tag{1}$$

where $L = 2\theta^g(E+l)$ is the arc length over which communication with the LEO satellite is available for users. Due to its intermediary role as a relay node, a LEO satellite has inherently limited observability regarding cloud-side resource consumption and service-level metrics. Conversely, ground base stations directly interact with computational resources, thus obtaining precise real-time measurements. Consequently,

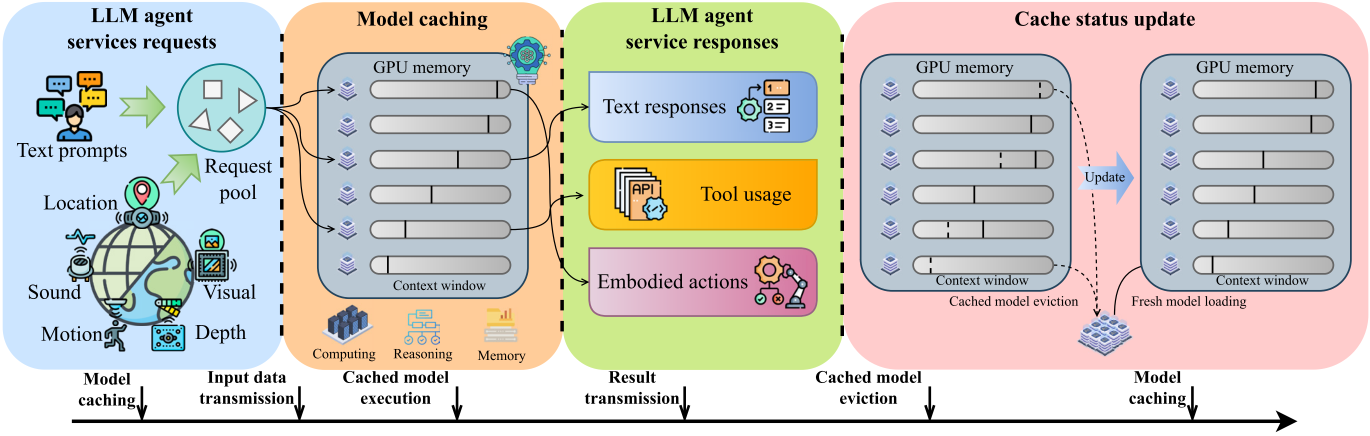


Fig. 2. The workflow of the joint caching and inference framework for provisioning LLM agents with cached models.

satellite operators' valuations become positively correlated but privately uncertain, increasing the risk of adverse selection in resource allocation.

B. Communication Model

To facilitate interaction with LLM agents, a group of users covered by network operator n, denoted by $\mathcal{U}_n$, can access services through network operator n for data transmission. These users share the same spectral resources, resulting in mutual interference among them [3]. The channel power gain from mobile user $u \in \mathcal{U}_n$ to LEO satellite 0, accounting for large-scale fading and shadowed-Rician fading, is represented by $g_{u,0}$ [30]. Similarly, $g_{u,n}$ represents the channel power gain from mobile user u to ground BS $n = 1, \ldots, N$, incorporating large-scale fading and Rayleigh fading for terrestrial communications. The bandwidth allocated by satellite 0 and ground BSs $n = 1, 2, \ldots, N$, is denoted as B_0 and B_n, respectively. Consequently, the uplink transmission rate for user $u \in \mathcal{U}_n$ to transmit input data of LLM agent services to network operators is given by

$$r_{u,n} = B_n \log_2 \left(1 + \frac{g_{u,n} p_u}{\sum_{j \in \mathcal{U}_n \setminus \{u\}} g_{j,n} p_j + \sigma^2} \right), \quad (2)$$

where p_u is the transmit power of user u and σ^2 is the power of the additive white Gaussian noise (AWGN). The satellite serves as an intermediary in providing LLM agent services between mobile users and cloud data centers via the satellite backbone network [31], with the transmission rate denoted by r_0^C. Moreover, ground BSs $n = 1, 2, \ldots, N$ connect to cloud data centers through the terrestrial core network with a fixed transmission rate r_n^C.

C. Model Caching for LLM Agent Services

To facilitate the provisioning of LLM agent services in SAGINs, we introduce a joint model caching and inference framework that enables edge servers located at ground BSs to cache LLMs and offload requests, optimizing the utilization of edge computing resources to provision LLM agent services for users, as shown in Fig. 2. Specifically, ground BSs $n = 1, \ldots, N$ are tasked with determining local caching and offloading strategies. Here, $a_{n,i,m}^t \in \{0, 1\}$ represents the binary variable that indicates whether model m for service i is cached at ground BS n during time slot t, and $b_{n,i,m}^t \in [0, 1]$ signifies the continuous variable reflecting the proportion of model m for service i being executed at ground BS n at time slot t. Let $\mathbf{a}_n^t = \{a_{n,1,1}^t, \ldots, a_{n,I,M}^t\}$ encapsulate the model caching decisions at ground BS n, with $\mathbf{a}^t = \{\mathbf{a}_1^t, \ldots, \mathbf{a}_N^t\}$ aggregating these decisions across all network operators. Furthermore, the request offloading decision for ground BS n is denoted by $\mathbf{b}_n^t = \{b_{n,1,1}^t, \ldots, b_{n,I,M}^t\}$, while $\mathbf{b}^t = \{\mathbf{b}_1^t, \ldots, \mathbf{b}_N^t\}$ represents the collective offloading decisions of ground BSs $n = 1, \ldots, N$. To extend the coverage of ground communication systems, satellite 0 with limited computing and energy resources acts as a relay between users and cloud data centers, whose model caching decisions are $\mathbf{a}_0^t = \mathbf{0}$ and request offloading decisions are $\mathbf{b}_0^t = \mathbf{0}$, i.e., all the LLM agent services are offloaded to cloud data centers for remote executing relayed by satellite 0.

For edge intelligence in SAGINs, LLM agent services requested by users can be executed at edge servers at ground BSs when the required LLMs are cached into GPUs. Let G_n denote the GPU computing capacity in terms of GPU memory of ground BS n. Then, the decision of model caching $\mathbf{a}_n^t$ should satisfy the following constraint at time slot t for ground BS $n = 1, \ldots, N$, as

$$\sum_{i \in \mathcal{I}} \sum_{m \in \mathcal{M}} a_{n,i,m}^t s_m \leq G_n, \quad (3)$$

where s_m is the running size of model m. This indicates that the edge servers cannot load all the LLMs into GPUs as the computing resources at edge servers are constrained. After the models are loaded into the GPUs of edge servers, LLM agent services can be executed at the ground BSs. Therefore, the constraint of LLM agent service provisioned at ground BS $n = 1, \ldots, N$, is represented as

$$b_{n,i,m}^t \mathbf{1}(R_{n,i,m}^t > 0) \leq a_{n,i,m}^t, \quad \forall i \in \mathcal{I}, m \in \mathcal{M}, \quad (4)$$

where $\mathbf{1}(\cdot)$ is the indicator function and $\mathbf{1}(R_{n,i,m}^t > 0)$ indicates that there are requests of LLM agent service i for model m at BS n at time slot t. Finally, the total computing power consumption of edge servers is constrained by the total computing capacity of GPUs at ground BS $n = 1, \ldots, N$, which can be represented as

$$\sum_{i \in \mathcal{I}} \sum_{m \in \mathcal{M}} e_m a^t_{n,i,m}(1 - b^t_{n,i,m}) R^t_{n,i,m} \leq E_n. \tag{5}$$

Nevertheless, in cloud data centers, it could be assumed that there is no GPU memory constraint or computing capacity constraint for executing LLMs.

D. Chain-of-Thought Inference Model

To improve the relevance and coherence of LLM agents, LLMs can leverage CoT prompting to perform step-by-step reasoning before obtaining the final response [32]. As an advanced inference approach to elicit the emerging abilities of LLMs, CoT prompting allows LLMs to generate a sequence of intermediate reasoning steps towards problem-solving or concluding, instead of attempting to solve the entire problem in merely zero-shot manner. During inference of LLMs, given any task description prompt d, LLM m can generate an answer by recursively predicting the sequence of next tokens from the learned distribution p_m conditioned on the concatenation of d and of the tokens sampled so far. For all sequences of messages in LLM agent service i, $D_i = \{d_{i,0}, \ldots, d_{i,k}\}$ of at most w_m tokens, the $p_m(D_i)$ follows the general product rule of probability [33], i.e.,

$$\begin{aligned} p_m(D_i) &= p_m(d_{i,0}, \ldots, d_{i,k}) \\ &= p_m(d_{i,0}) p_m(d_{i,1}|d_{i,0}) \ldots p_m(d_{i,k}|d_{i,0}, \ldots, d_{i,k-1}), \end{aligned} \tag{6}$$

which is a good approximation of the true distribution $\hat{q}(D_i)$.

For each CoT prompt in LLM agent service i, LLM m is provided with c_i varying length CoT examples $E_i = \{e_{i,0}, \ldots, e_{i,k}\}$ and each thought $e_{i,k}$ in E_i is a sequence of k_i tokens representing one reasoning step. Those examples are designed to aid the LLMs in producing correct answers via CoT generation and thus for service i, E_i are generated with true intentions $\theta^\star$ and true context $c^\star$. To serve LLM agent service i, for the given E_i and a task $d_{i,0}$, LLMs then generate $(d_{i,1}, \ldots, d_{i,k})$ messages. To evaluate the performance of the approximation of LLMs, we have the following definition.

Definition 1 (ϵ-ambiguity [33]): For CoT examples E_i of LLM service i generated based on true context $c^\star$ and true intention $\theta^\star$, the ambiguity of the chain $\epsilon(E_i)$ is defined as the complement of the likelihood of the context $c^\star$ and intentions $\theta^\star$ conditioned on CoT examples E_i, i.e.,

$$\hat{q}(c^\star, \theta^\star | E_i) = 1 - \epsilon(E_i). \tag{7}$$

In addition to the ambiguous definition of LLMs, we define the quality of contexts in training datasets as follows.

Definition 2: To account for the potentially non-uniform distribution of contexts in training datasets at network operators, we introduce a skewness parameter $\gamma_n(c^\star)$ for each network operator n, which is defined as

$$\gamma_n(c^\star) = \sup_{c \in C} \frac{\hat{q}(c^\star)}{\hat{q}(c_n)}, \tag{8}$$

where c_n is the context owned by network operator n.

Ground BSs can collect the context during their provisioning of LLM agent services, which satisfies the preference of their local users. Therefore, we have the following assumption.

Assumption 1: The prior distribution associated with true contexts $c^\star$ is uniform.

Based on the uniform context considered in Assumption 1, $\gamma_n(c^\star) = 1$ guarantees small values or provides CoT examples that should have small enough ambiguity, so the model with high certainty could guess the true context $c^\star$ from them [34]. Following Definition 2, we can estimate the difference of ambiguity measurements between the learned distribution p_m and the true distribution $\hat{q}$, conditioned on input messages D_i of service i, as follows.

Theorem 1: Considering a collection of c_i varying length CoT examples, which are generated from the intention $\theta^\star$ with the optimal context $c^\star$ sampled from $q_m(c)$ that satisfies Assumption 1. Furthermore, let $d_{i,0}$ be the input message or task sampled from $q(\cdot|\theta^\star_0)$, which is generated from $\theta^\star_0$ sampled from $q_m(\cdot|c^\star)$. Then, for any sequence of messages D_i, we have

$$|p_m(D_i|d_{i,0}, E_i) - \hat{q}(D_i|d_{i,0}, c^\star)| \leq \eta \prod_{y=1}^{c_i} \frac{\epsilon(E_{i,y})}{1 - \epsilon(E_{i,y})}, \tag{9}$$

where $\eta = 2\frac{\epsilon(d_{i,0})}{1-\epsilon(d_{i,0})}$ depends on the ambiguity of the input.

Proof: We provide a simplified proof to show how to obtain this bound with the necessary steps and the complete proof can be found in [34]. Starting from $p_m(D_i|d_{i,0}, E_i)$, we have

$$\begin{aligned} p_m(D_i|d_{i,0}, E_i) &= \frac{\hat{q}(D_i, E_i)}{\hat{q}(d_{i,0}, E_i)} \\ &= \frac{\hat{q}(D_i, E_i, c^\star) + \sum_{c \neq c^\star} \hat{q}(D_i, E_i, c)}{\hat{q}(d_{i,0}, E_i, c^\star) + \sum_{c \neq c^\star} \hat{q}(d_{i,0}, E_i, c)} \\ &= \frac{\hat{q}(D_i \backslash \{d_{i,0}\}, E_i, c^\star) + \frac{\sum_{c \neq c^\star} \hat{q}(D_i, E_i, c)}{\hat{q}(d_{i,0}, E_i, c^\star)}}{1 + \frac{\sum_{c \neq c^\star} \hat{q}(d_{i,0}, E_i, c)}{\hat{q}(d_{i,0}, E_i, c^\star)}} \\ &= \frac{\hat{q}(D_i \backslash \{d_{i,0}\}, E_i, c^\star) + \Lambda}{1 + \Upsilon}, \end{aligned} \tag{10}$$

where Λ and Υ are given by

$$\Lambda = \frac{\sum_{c \neq c^\star} \hat{q}(D_i, E_i, c)}{\hat{q}(d_{i,0}, E_i, c^\star)} \text{and } \Upsilon = \frac{\sum_{c \neq c^\star} \hat{q}(d_{i,0}, E_i, c)}{\hat{q}(d_{i,0}, E_i, c^\star)}. \tag{11}$$

By leveraging the definition of ambiguity measure for CoT example $E_{i,y}$, we can establish the following bounds on Λ and Υ as

$$\Lambda, \Upsilon \leq \frac{\gamma^{c_i}_n(c^\star)\epsilon(d_{i,0})}{1 - \epsilon(d_{i,0})} \prod_{y=1}^{c_i} \frac{\epsilon(E_{i,y})}{1 - \epsilon(E_{i,y})}. \tag{12}$$

Finally, combining the above components, we have

$$\begin{aligned} &|p_m(D_i|d_{i,0}, E_i) - \hat{q}(D_i|d_{i,0}, c^\star)| \\ &\quad = \frac{|\Lambda - \Upsilon \hat{q}(D_i/\{d_{i,0}\}|d_{i,0}, c^\star)|}{1 + \Upsilon} \\ &\quad \leq |\Lambda + \Upsilon \hat{q}(D_i/\{d_{i,0}\}|d_{i,0}, c^\star)| \\ &\quad \leq \Lambda + \Upsilon \\ &\quad \leq \eta \prod_{y=1}^{c_i} \frac{\epsilon(E_{i,y})}{1 - \epsilon(E_{i,y})}, \end{aligned} \tag{13}$$

where $\eta = 2\frac{\gamma_n^{c_i}(c^\star)\epsilon(d_{i,0})}{1-\epsilon(d_{i,0})}$, following Assumption 1 and indicating that $\gamma_n(c^\star) = 1$ and thus $\eta = 2\frac{\epsilon(d_{i,0})}{1-\epsilon(d_{i,0})}$. □

Theorem 1 indicates that the LLM prompted with CoT example E_i is capable of approximating the true natural language distribution equipped with true context and intentions.

Assumption 2: The CoT example E_i generated from $\theta^\star$ with a context $c^\star \sim q(c)$ is bounded by the ambiguity measure, i.e.,

$$\epsilon(E_i) = \hat{q}(c^\star, \theta^\star | E_i) \leq \sigma, \tag{14}$$

where $\sigma \in \left[0, \frac{1}{2}\right]$.

Assumption 2 implies that when carefully selected, CoT examples E_i and the true context $c^\star$ can be recovered from E_i with reasonably high certainty, i.e., the probability the $c^\star$ is behind E_i is strictly greater than on a half. Given such CoT examples, we can transform the bound in Theorem 1 into a geometrical convergence rate with the number of examples growing large as

$$|p_m(D_i|d_{i,0}, E_i) - \hat{q}(D_i|d_{i,0}, c^\star)| \leq \eta\beta^{c_i}, \tag{15}$$

where the CoT gain is $\beta = \frac{\sigma}{1-\sigma} \in [0, 1)$. Those examples described in Assumption 2 should be carefully selected to guarantee low ambiguity requirements. In practice, however, it can be challenging to collect such chain-of-thought examples, as there can be an assumption that allows us to measure ambiguity for a given sequence of thoughts as below.

Assumption 3: For the CoT examples E_i generated from true intentions $\theta^\star$ with the true context context $c^\star \sim q(c)$, the associated ambiguity measure $\epsilon(E_i)$ vanishes as the length of sequence grows large as

$$\lim_{l\to\infty} \epsilon(E_i) = 0. \tag{16}$$

Assumption 3 implies that uncertainty over true context $c^\star$ and true intentions $\theta^\star$ for a sequence of thoughts is diminishing when more of these thoughts are collected. Therefore, for long enough CoT examples, the asymptotic requirement is sufficient to guarantee a low ambiguity measure. Satisfying Assumption 3, we have the following lemma.

Lemma 1: Considering CoT examples E_i for any fixed $\sigma \in \left[0, \frac{1}{2}\right)$ there is a length threshold $k_{i,\sigma}^\star \in \mathbb{N}$. For any $k_i \geq k_{i,\sigma}^\star$, we have

$$\epsilon(E_i) \leq \sigma. \tag{17}$$

Proof: By selecting $\sigma \in \left[0, \frac{1}{2}\right)$, CoT examples E_i following Assumption 3 have the approximation that $\lim_{l\to\infty} \epsilon(E_i) = 0$. Then, there exists $k_{i,\sigma}^\star \in \mathbb{N}$ such that for any $k_i \geq k_{i,\sigma}^\star$, the inequality $\epsilon(E_i) \leq \sigma$ holds. □

Based on Lemma 1, the geometrical convergence rate in Eq. (15) can be established when the LLM is prompted with CoT examples E_i of sufficient length, i.e., $k_i \geq k_{i,\sigma}^\star$ following Assumption 3. In contrast to the low ambiguity requirement, CoT examples with low asymptotic ambiguity can be more attainable. Therefore, during the step-by-step inference of LLMs, the original CoT examples E_i satisfying Assumption 3 can be split or divided into different lengths and sizes of thoughts E_i' following the required threshold $\sigma \in \left[0, \frac{1}{2}\right)$, which can be more refined reasoning steps with predefined ambiguity.

LLMs, such as GPT-3, can perform CoT prompting which indicates that they can learn from past CoT examples for complex tasks presented to them. The intermediate thoughts can be used to enhance the performance of LLM agents, as LLMs can use meta-gradient learning during interaction to fit them [35]. However, depending on the relevance and coherence of intermediate thoughts, few-shot learning may have favorable or unfavorable impacts on the model performance. Based on the caching decision $a_{n,i,m}^t$ and offloading decision $b_{n,i,m}$, the batch of requests executed as ground BS n can be calculated as $\delta_{n,i,m}^t = a_{n,i,m}^t(1-b_{n,i,m}^t)R_{n,i,m}^t k_i$ for service i and model m, where k_i is the size of CoT examples for service i which can be estimated via Lemma 1. In general, the number of thoughts increases monotonically when the LLM is cached into the edge servers, which can be represented as

$$K_{n,i,m}^t = \begin{cases} 0, & t = 0, \\ a_{n,i,m}^t(K_{n,i,m}^{t-1} + \delta_{n,i,m}^t), & \text{otherwise.} \end{cases} \tag{18}$$

Similar to the definition of age of information (AoI), the AoT measures the freshness of intermediate thoughts within the cached LLMs for the current inference requests. With a vanishing factor $\Delta_{i,m}^t$ of thoughts, the AoT is adjusted by the non-increasing age utility function, which is represented as

$$\kappa_{n,i,m}^t = \begin{cases} 0, & t = 0, \\ a_{n,i,m}^t\{\kappa_{n,i,m}^{t-1} + \delta_{n,i,m}^t - \Delta_{i,m}^t\}^+, & \text{otherwise.} \end{cases} \tag{19}$$

According to the AoT, the weighted total of the number of examples in demonstrations may be used to determine the number of examples in context. The AoT metric is computationally lightweight and practically implementable without internal modifications to the LLM inference process. Calculation involves only maintaining two integer counters per cached model—the accumulated length of reasoning tokens and the current AoT score. Updating these counters upon inference completion incurs negligible computational overhead, as it only requires two integer additions and one comparison operation. This procedure leverages data (timestamps, token counts) inherently generated by the inference pipeline, thus avoiding any modifications to model architecture or inference mechanisms. Consequently, AoT computation is both efficient and practically viable for real-world deployments, introducing no meaningful latency increase.

Based on Eq. (15), the few-shot CoT reasoning performance $A_{i,m}$ of model m in service i can be defined as

$$A_{n,i,m}^t = \alpha_{i,m} \log(1/\beta_{i,m}^{\kappa_{n,i,m}^t}), \tag{20}$$

where $\alpha_{i,m}$ is the zero-shot accuracy of LLM m for service i, estimated on a held-out validation set [8]. The factor β represents the per-thought attenuation rate of CoT gain that obtained from the ambiguity upper-bound in Assumption 2, and $\log(1/\beta_{i,m}^{\kappa_{n,i,m}^t})$ is the performance gain of generated CoT examples of LLM m for service i [33], [34].

IV. Problem Formulation, Caching Algorithm Design, and Market Design

In this section, we first formulate the problem of provisioning LLM agents in SAGINs to maximize the quantity and quality of LLM agent services. To solve the formulated problem, we next propose a model caching algorithm for local cached model management for ground BSs. Furthermore, we design an LLM agent market for incentivizing network operators, i.e., satellites and ground BSs, to contribute their resources to execute LLM agents.

During the provisioning of LLM agent services, network operators, i.e., satellite or terrestrial BSs, need to provide communication, computation, storage, and cached model resources to run the LLM agent services. In detail, the user's request and returned results need to be transmitted over the wireless channel, which costs bandwidth to complete the interaction between the user and the LLM agent deployed at the edge. The reasoning process of LLM consumes a lot of computing resources, especially when LLMs perform CoT reasoning; they need a large amount of intermediate computation to obtain the final high-quality results. Since LLM agents use CoT reasoning to improve the quality of final answers, they need substantial storage resources to store past CoT examples locally for demonstration. Finally, since the context window of LLM is limited, when the model inference is deployed beyond a certain number of tokens, the model needs to be refreshed, i.e., to evict the old model and load another fresh model.

A. *Cost Structure*

As mentioned above, the LLM agent services provisioned at ground BSs can be handled by edge servers or offloaded to cloud data centers via the core network. Based on the decisions of model caching and request offloading, the overall cost of providing LLM agent services, including the cost at the edge and the cost at the cloud, can be expressed as follows.

1) Edge Inference Cost: Specifically, the cost of edge inference includes the cost of switching in GPUs, the cost of transmitting data across edges, the cost of performing computations on edges, and the cost of model accuracy. Based on decisions about model caching, each edge server must load models into the GPU memory before execution. During the model loading process, there is a cost associated with switching between models in GPUs, which includes the latency of loading the model and the cost of wear and tear on the hardware [36]. Therefore, the switching cost l_n^s of ground BS n to load and evict models can be calculated as

$$l_n^{switch}(\mathbf{a}^t) = \sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}} \lambda \mathbf{1}(a_{n,i,m}^t > a_{n,i,m}^{t-1}), \tag{21}$$

where λ denotes the coefficient for loading and evicting the model and $\mathbf{1}(\cdot)$ is the indicator function. When $a_{n,i,m}^t > a_{n,i,m}^{t-1}$, i.e., $a_{n,i,m}^t = 1$ and $a_{n,i,m}^{t-1} = 0$, $\mathbf{1}(a_{n,i,m}^t > a_{n,i,m}^{t-1})$ indicates that the loading of an uncached model. Otherwise, there is no switching cost incurred at edge servers.

When the requested models are cached into the GPU memory of edge servers, users communicate with the edge servers to request LLM agent services. Let l_n^{trans} denote the transmission cost of input prompts and inference results. The transmission cost of ground BS n can be calculated as

$$l_n^{trans}(\mathbf{a}^t, \mathbf{b}^t) = \sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}} R_{n,i,m}^t \left(l_{n,i} + \frac{d_i}{r_n^C} b_{n,i,m}^t \right), \tag{22}$$

where $l_{n,i} = d_i / \mathbb{E}_{u\in\mathcal{U}_n}[r_{u,n}]$ is the unit transmission cost per input and result for service i to transmit the input data with size d_i from users $\mathcal{U}_n$ to ground BS n.

Let f_n denote the computing capacity of ground BS n. The execution of LLM agent services at ground BSs incurs inference latency, which is denoted as l_n^{comp} for ground BS n. The edge computing cost can be calculated as

$$l_n^{comp}(\mathbf{a}^t, \mathbf{b}^t) = \sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}} \delta_{n,i,m}^t \frac{e_m}{f_n}, \tag{23}$$

where $\delta_{n,i,m}^t = a_{n,i,m}^t(1 - b_{n,i,m}^t)R_{n,i,m}^t k_i$ is the total computation token for a batch of LLM agent service i. Finally, as ground BSs might not have sufficient resources for executing the best-matched model requested by LLM agents, which might introduce a performance gap, the requests processed by other LLMs with the equivalent function incur accuracy cost l_n^{acc}, which can be represented as

$$l_n^{acc}(\mathbf{a}^t, \mathbf{b}^t) = \sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}} \bar{A}_{n,i,m}^t R_{n,i,m}^t a_{n,i,m}^t (1 - b_{n,i,m}^t), \tag{24}$$

where $\bar{A}_{n,i,m}^t = \frac{1-\alpha_{i,m}}{\kappa_{n,i,m}^t \log(1/\beta_{i,m})}$ is the unit accuracy cost following Eq. (20). By sacrificing some accuracy of LLM agent services, the system can reduce the model missing rate. Therefore, the total edge inference cost of ground BS n is

$$\begin{aligned} L_n^t(\mathbf{a}^t, \mathbf{b}^t) = {} & l_n^{switch}(\mathbf{a}_n^t) + l_n^{trans}(\mathbf{a}_n^t, \mathbf{b}_n^t) \\ & + l_n^{comp}(\mathbf{a}_n^t, \mathbf{b}_n^t) + l_n^{acc}(\mathbf{a}_n^t, \mathbf{b}_n^t). \end{aligned} \tag{25}$$

The edge inference cost is jointly determined by the caching decisions and offloading decisions of ground BSs. Nevertheless, the missed or offloaded requests are executed by cloud data centers. The fourth coordinate $l_{\text{acc}}(\kappa)$ makes our cost model qualitatively different from traditional hardware-only formulations because it ties monetary expenditure to LLM-specific freshness. Satellites, lacking direct access to token-level context-window statistics, can only estimate l_{acc} via delayed telemetry, whereas ground BSs observe it in real time. This information gap creates valuation asymmetry that the auction mechanism must address.

2) Cloud Inference Cost: The ground BSs are typically limited in resources and cannot serve all LLM agent service requests. There are two main reasons for this limitation. First, the ground BSs' computing capacities may be insufficient to load many LLMs into the GPU memory. Second, the ground BSs' energy capacity may not be enough to handle all requests. Therefore, some requests will be offloaded to cloud data centers for remote execution.

When the requested models are not available or the ground BS lacks sufficient resources, these user requests are transmitted to the cloud data center, which then allocates resources to serve them. According to [36], cloud data centers can provide serverless LLM agent services, charging users on a "pay-as-you-go" basis. This means that users pay based on the number of requests rather than the specific resources occupied. However, this cloud-based inference introduces additional latency due to data transmission in the core network, which is larger than the latency at ground BSs. Additionally, the accuracy cost of offloaded inference requests executed by the cloud data center is expected to be minimal, as the requests can be processed using the most accurate model with common CoT examples owned by the data center.

Based on the above analysis, we use l_n^C to represent the total cost of offloading requests to the cloud data center for remote execution. Then, the total cloud computing cost at time slot t is

$$L_C^t(\mathbf{a}^t, \mathbf{b}^t) = \sum_{n\in\mathcal{N}} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} l_{0,m} b_{n,i,m}^t R_{n,i,m}^t, \tag{26}$$

where $l_{0,m}$ is the unit process cost of model m at cloud data centers. Therefore, the total cost L_n^{total} for provisioning LLM agent services at network operator n can be calculated as

$$L_n^{total} = \frac{1}{T} \sum_{t\in\mathcal{T}} \left(L_C^t + L_n^t \right). \tag{27}$$

B. Problem Formulation of Ground Base Stations

To improve the efficiency of mobile edge intelligence, we take into account both the cost of edge inference and cloud inference. This includes considering the switching cost, accuracy cost, transmission cost, and inference cost over a specific period T. For ground BSs $n = 1, \ldots, N$, the problem of providing LLM agent services is formulated as

$$\begin{aligned} &\min_{\mathbf{a}_n^t, \mathbf{b}_n^t} \; L_n^{total} && \text{(28a)}\\ &\text{s.t.} \;\; (3), (4), (5) && \text{(28b)}\\ &\quad\;\; K_{n,i,m}^t \le w_m, \quad \forall i \in \mathcal{I}, \quad \forall m \in \mathcal{M} && \text{(28c)}\\ &\quad\;\; a_{n,i,m}^t \in \{0, 1\} && \text{(28d)}\\ &\quad\;\; b_{n,i,m}^t \in [0, 1]. && \text{(28e)} \end{aligned}$$

Constraint (28c) indicates that the context tokens cannot exceed the size of context windows. To address the optimization problem described above, we need to overcome the challenge of time-coupled elements, such as GPU memory and CoT examples, as it takes into account both future request dynamics and historical CoT examples. Furthermore, the problem is a mixed-integer programming problem, which is known to be NP-hard. To solve the problem efficiently, we require low-complexity algorithms to determine decisions of model caching and request offloading.

C. The Least Age-of-Thought Caching Algorithm

To effectively serve LLMs for provisioning LLM agent services, we propose the least AoT algorithm based on the proposed AoT metric. When additional GPU memory is required for loading an uncached requested LLM, the least AoT algorithm counts the value of CoT examples, calculates them, and removes the cached LLM with the lowest AoT. Therefore, at each time slot t, the model caching decisions can be obtained by solving the maximization problem of the number of CoT examples for the cached models, which can be represented as

$$\begin{aligned} &\max_{\mathbf{a}^t} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} \kappa_{n,i,m}^t && \text{(29a)}\\ &\text{s.t.} \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} a_{n,i,m}^t s_m \le G_n, \quad \forall n \in \mathcal{N} && \text{(29b)}\\ &\quad\;\; \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} (1 - b_{n,i,m}^t) e_m \le E_n, \quad \forall n \in \mathcal{N} && \text{(29c)}\\ &\quad\;\; a_{n,i,m}^t \in \{0, 1\}. && \text{(29d)} \end{aligned}$$

Algorithm 1 The Least Age-of-Thought Cached Model Replacement Algorithm

```
Input: Model caching status $\mathbf{a}_n^{t-1}$, model context status $\kappa_n^t$,
  and LLM agent service requests $R_n^t$, GPU memory
  capacity $G_n$, GPU computing capacity $E_n$.
Output: Model caching decision $\mathbf{a}_n^t$ and request offloading
  decision $\mathbf{b}_n^t$.
Initialize $\mathbf{a}_n^t = \mathbf{0}$, $\mathbf{b}_n^t = \mathbf{0}$,
  $G_n^t = G_n - \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} a_{n,i,m}^{t-1} s_m$, and $E_n^t = 0$.
for $R_{n,i,m}^t > 0$ in $R_n^t$ do
    if $a_{n,i,m}^{t-1} = 1$ then
        $a_{n,i,m}^t \leftarrow 1$;
    end
    else if $G_n^t + s_m \leqslant G_n$ then
        $a_{n,i,m}^t \leftarrow 1$;
        $G_n^t \leftarrow G_n^t + s_m$;
    end
    else
        while $G_n^t + s_m \geqslant G_n$ do
            $(\bar{\imath}, \bar{m}) \to \arg\min_{(\bar{\imath}, \bar{m})} \{\kappa_{n,\bar{\imath},\bar{m}}^t \in \kappa_n^t\}$;
            if $a_{n,\bar{\imath},\bar{m}}^{t-1} = 1$ and $a_{n,\bar{\imath},\bar{m}}^t = 0$ then
                $G_n^t = G_n^t - s_{\bar{m}}$;
            end
        end
        if $G_n^t + s_m \leqslant G_n$ then
            $a_{n,i,m}^t \leftarrow 1$;
            $G_n^t \leftarrow G_n^t + s_m$;
        end
    end
    if $a_{n,i,m}^t = 1$ and $E_n^t + e_m a_{n,i,m}^t R_{n,i,m}^t \leqslant E_n$ then
        $b_{n,i,m}^t \leftarrow 1$;
        $E_n^t \leftarrow E_n^t + e_m a_{n,i,m}^t b_{n,i,m}^t R_{n,i,m}^t$;
        Update context status $\kappa_{n,i,m}^t$ following Eq. (19);
    end
    if $K_{n,i,m}^t > w_m$ then
        $a_{n,i,m}^t \leftarrow 0$;
        $b_{n,i,m}^t \leftarrow 0$.
    end
end
```

The available capacity of GPU memory G_n^t of ground BS $n = 1, \ldots, N$ at time slot t can be calculated as $G_n^t = G_n - \sum_{i\in\mathcal{I}} \sum_{m\in\mathcal{M}} a_{n,i,m}^t s_m$. This algorithm allows the least important LLM to have a higher chance for eviction in the current inference task. The complexity of the algorithm increases linearly as the number of models increases. Therefore, it works well with a large number of LLMs on ground BSs with limited GPU memory. Using more intermediate reasoning steps during inference makes the LLMs perform more accurately. Based on caching decisions $\mathbf{a}_n^t$ by solving the optimization problem in (29a), offloading decisions $\mathbf{b}_n^t$ are obtained by solving the optimization problem in (28a). The detailed implementation of the Least AoT algorithm is provided in Algorithm 1. At each decision slot, the Least AoT scheduler updates Age-of-Thought scores and identifies the cached model with minimum score, resulting in a time and memory complexity of $\mathcal{O}(M)$, enabling millisecond-level execution at edge servers.

AoT-based caching decisions are triggered every few minutes as context windows deplete or user demand shifts, whereas the DQMSB auction is executed for each batch of incoming service requests with its DQN weights retrained offline and refreshed daily.

D. Market Design

To motivate LLM agent providers to construct and update LLM agents for users, we design an LLM agent market where sellers (LLM agent providers) can earn profits from provisioning LLM agent services, and bidders (network operators) are competing for provisioning LLM agent services to their users. In every decision slot, each LLM agent provides offers exactly one service opportunity to all network operators. The interaction proceeds in four explicit stages: (i) the seller publishes the slot description (model catalogue, AoT decay coefficient, slot duration) and a common random seed; (ii) each operator privately evaluates a valuation $v_n = c_n m_n$ (see Section IV-B) and submits a sealed bid x_n; (iii) the auctioneer applies the pricing rule to allocate the slot and collect payment; (iv) the winning operator executes the LLM agent, after which the provider reveals the realised AoT increments and resource consumption to all bidders, closing the feedback loop for the next slot. This message-exchange timeline eliminates ambiguity regarding what information is public, what remains private, and when updates occur. We consider that the network operators are risk-neutral bidders in the market whose surpluses are positively correlated with each other based on the revelation principle [14].

In SAGINs, users can obtain the services of a running LLM agent via a satellite or ground BS as their assistants to perform their local tasks. Each network operator $n \in \mathcal{N}$ has valuation $v_n = c_n m_n$ for the opportunity to serve the LLM agent, which is a production of the common value $c_n = \mathbb{E}_{t\in\mathcal{T}}[L_n^{total} - l_n^{acc}(\mathbf{a}^t, \mathbf{b}^t)]$ about physical resource consumption and the match performance gain $m_n = \mathbb{E}_{t\in\mathcal{T}}[\log(1/\beta_{i,m}^{\kappa_{n,i,m}^t})]$ obtained in Eq. (20). Specifically, the common value captures attributes of the resource consumption required to execute the LLMs, which depends on the communication, computing, and storage resources. Meanwhile, the match quality captures idiosyncratic components of cached models in network operators that affect the quality of LLM agents [14]. During the valuation of LLM agents, the common value is considered to be independent of match quality, i.e., the resource consumption of LLM agents is not relevant to the quality of LLM agents. We use $v_{(i)}$, $c_{(i)}$, and $m_{(i)}$ to denote the i-th highest valuation, common value, and match value factor, respectively.

In the market, a mechanism $\mathcal{M}(\mathbf{v}) = (\mathbf{z}, \mathbf{p})$ is required to map the privately held valuation $\mathbf{v}$ to allocation probabilities $\mathbf{z} = (z_0, z_1, \ldots, z_N)$ and payment $\mathbf{p} = (p_0, p_1, \ldots, p_N)$. The expected surplus, i.e., the realization of valuation, for satellite is $\mathbb{E}[v_0 z_0(\mathbf{v})]$. Meanwhile, the surplus from the LLM agent allocated to the ground BSs is given by $\mathbb{E}\left[\sum_{n=1}^{N} v_n z_n(v)\right]$. To maximize the total surplus of network operators to provision LLM agents, the problem for the mechanism can be formulated as

$$\max_{\mathcal{M}} \mathbb{E}\left[\sum_{n=0}^{N} v_i z_i(v)]\right] \tag{30a}$$

$$\text{s.t.} \quad \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}} \frac{d_i R_{0,i,m}^t}{\mathbb{E}_{u\in\mathcal{U}_0}\left[r_{u,0} + r_0^C\right]} \leqslant T_0^S, \tag{30b}$$

$$\sum_{n=0}^{N} z_n \leqslant 1, \tag{30c}$$

$$z_n \in \{0,1\}, \quad \forall n \in \mathcal{N}. \tag{30d}$$

Constraint (30b) indicates that the provisioning time of satellites cannot exceed their coverage time. Constraints (30c) and (30d) indicate that there is one and only one network operator that can obtain the opportunity to run LLM agents.

V. The Deep Q-Network-Based Modified Second-Bid Auction

A. Modified Section-Price Auction

In the LLM agent market, all network operators submit their bids $\mathbf{x} = (x_0, x_1, \ldots, x_N)$ to the auctioneers. Then, the auctioneer leverages MSB auction [14] to determine the winning bidder and payment, which can be formulated as follows.

Mechanism 1 (Modified Second-bid Auction): The MSB auction allocates the LLM agent to the highest BS if their bid exceeds the second-highest bid by a price scaling factor of ρ or more and prices the LLM agent with the second-highest bid scaling with ρ. When no performance bidders win, the opportunity is allocated to the relaying satellite, which offloads LLM agent services to cloud data centers, with the contracted price x_0 is chosen to maximize its expected profit as $x_0 = \max_x \mathbb{E}[(v_0 - v_{(1)}\mathbf{1}(v_{(1)\leq x_0}))]$ [14]. Formally, the allocation rule and the pricing rule can be represented as follows.

- Allocation rule: For ground BSs, namely, the performance bidders, $n = 1, \ldots, N$, the allocation probabilities $z_n \in \{0,1\}$ for the deterministic mechanism are determined by
$$z_n(\mathbf{x}) = \mathbf{1}(x_n > \rho \max\{\mathbf{x}_{-n}\}), \tag{31}$$
for $\rho \geq 1$. Then, the allocation probability for the satellite can be calculated based on the allocation probabilities of ground BSs, as $z_0(\mathbf{x}) \leq 1 - \sum_{n=1}^{N} z_n(\mathbf{x})$.
- Pricing rule: If the winner is ground BS $n = 1, \ldots, N$, the winning ground BS is charged with the product of the price scaling factor ρ the second highest bid, i.e., the payment p_n can be calculated as
$$p_n(\mathbf{x}) = z_n(\mathbf{x}) \cdot \rho \max\{\mathbf{x}_{-n}\}. \tag{32}$$
Furthermore, the payment of the satellite depends on their contract price x_0, i.e., $p_0(\mathbf{x}) = z_0(\mathbf{x}) x_0$.

While MSB is straightforward, its reliance on a fixed price-scaling factor ρ creates significant trade-offs. A lower ρ excessively favors ground stations, neglecting the satellite's hidden costs, whereas a higher ρ overly restricts ground station bids, undermining overall system efficiency. Existing solutions reliance on a fixed price-scaling factor $\rho = \max(1, \mathbb{E}[x_0]/\mathbb{E}[x_{(2)}])$ based on historical statistical information [14]. Given the inherently dynamic and non-stationary nature of LLM agent services in satellite–ground networks, where user demand, satellite coverage, and computational availability can shift rapidly, a single fixed price-scaling factor ρ is unlikely to remain optimal across time, making it necessary to determine ρ adaptively in real time.

B. DQN-Based Price Scaling Factor

To leverage DRL to determine the price scaling factor, we formulate the process of MSB as a Markov decision process

Algorithm 2 The DQMSB Auction

```
Input: Bids $\mathbf{x}$;
Output: Allocation probabilities and payments;
Initialize Q-function parameters $\phi$, target Q-function
  parameters $\phi'$, and replay buffer $\mathcal{B}$;
for episode in $1, \dots, T$ do
    for iteration $k$ in $1, \dots, K$ do
        Receive the bids $\mathbf{x}$ from bidders and observe the
          state $S_k$;
        Determine the price scaling factor $\rho = 10^{a_k/|\mathcal{A}|}$,
          following $a_k = \arg\max_a Q_\phi(S_k, a)$;
        Calculate the winning probabilities $\mathbf{z}$ and payments
          $\mathbf{p}$ obtained from the allocation rule in Eq. (31) and
          the pricing rule in Eq. (32);
        Observe the next states $S_{k+1}$ and reward $r_k$;
        Store transition $(S_k, a_k, r_k, S_{k+1})$ to $\mathcal{B}$;
        Sample a mini-batch of experiences
          $(S_k, A_k, r_k, S_{k+1})$ from $\mathcal{B}$;
        Calculate $y_k = r_k + \gamma \max_{a^{k+1}} Q_{\phi'}(S_{k+1}, a_{k+1})$
          using target network $Q_{\phi'}$;
        Update $\phi$ by performing gradient descent on the loss
          calculated in Eq. (34);
        Update target network $\phi'$.
    end
end
```

(MDP), consisting of states, actions, and rewards. At each time step, the DRL-based auctioneer observes the current state and selects an action, i.e., the price scaling factor from the feasible action space. Then, the auctioneer determines the winning bidder and the payment, and receives the total surplus as the reward. Formally, the MDP of MBS can be formulated as follows. During the auction process, multiple network operators submit their bids to the auctioneer. Therefore, the state space consists of the bidding information in MSB, i.e., $S_k \triangleq \{\mathbf{x}^k\}$ at decision slot k. To calculate the pricing rule and allocation rule, the auctioneer needs to determine the level of the price scaling factor, i.e., $a_k \in \mathcal{A} \subseteq \mathbb{N}$. Then, the price scaling factor is calculated as $\rho = 10^{a_k/|\mathcal{A}|}$, where $|\mathcal{A}|$ is the size of action space. The reward is calculated as the total surplus achieved in the market, i.e., $r_k = \mathbb{E}\left[\sum_{n=0}^{N} v_i z_i(v)\right]$ calculated in Eq. (30a).

Based on the MDP of MBS, the objective of the DRL-based auctioneer is to optimize a non-linear function approximation with parameters ϕ as a Q-network to maximize the future discounted return $R_k = \sum_{k'=k}^{K} \gamma^{k'-k} r_{k'}$. In this regard, the auctioneer needs to learn the optimal action-value function as

$$Q^\star(S,a) = \mathbb{E}_{S'}\left[r + \gamma \max_{a'} Q^\star(S',a')|S,a\right], \tag{33}$$

where the action a' is selected to maximize the expected value of $r + \gamma Q^\star(S',a')$. Therefore, the target for iteration k can be calculated as $y_k = r_k + \gamma \max_{a^{k+1}} Q_{\phi'}(S^{k+1}, a^{k+1})$, where ϕ' is the parameters of target network $Q_{\phi'}$. To minimize the performance gap between the current Q value and the target, the loss function can be defined as

$$L = \frac{1}{K}\sum_{k=1}^{K}\left(y_k - Q_\phi(S_k, a_k)\right)^2. \tag{34}$$

For computational efficiency and performance stability, DQN leverages stochastic gradient descent to optimize the parameters ϕ on the loss calculated in Eq. (34) and update the target network ϕ' in each period of iteration. Finally, the DQMSB auction is provided in Algorithm 2. During live auctions, the DQMSB mechanism evaluates each action using a trained Q-network and performs an $\arg\max$ operation over $|A|$ actions, followed by a linear scan of N bids to determine auction outcomes. Consequently, the overall complexity is $\mathcal{O}(|A|\,C_{\text{net}} + N)$, where C_{net} denotes the computation of a single forward pass through the fully-connected network.

C. Property Analysis

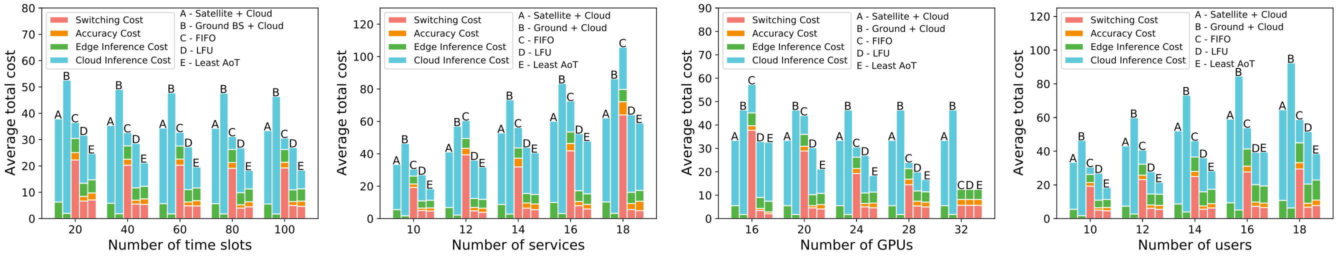
For an auction, strategy-proofness means that participants cannot achieve a higher utility by altering their honest bids. Adverse-selection-free means that the presence of market externalities and asymmetric information is unrelated to bidders' valuations. Therefore, it is important to note that the DQMSB auctions are fully strategy-proof and adverse-selection-free, as given in the following theorem.

Theorem 2: The DQMSB auction with the price scaling policy with fixed parameters $\bar{\phi}$ is anonymous, fully strategy-proof, and adverse-selection-free.

Proof: To prove that the proposed DQMSB auction is anonymous, fully strategy-proof, and adverse-selection-free, the auction should be characterized by a critical payment function χ conditioned on $\bar{\phi}$ such, for any competing bids $\mathbf{x}_{-n}$, ground BS bidder $n = 1, \dots, N$ wins if and only if its bid exceeds the critical payment $\chi(\mathbf{x}_{-n};\bar{\phi}) = \rho^{a_k/|\mathcal{A}|}\max\{\mathbf{x}^k_{-n}\}$, where $a_k = \arg\max_a Q(\mathbf{x}^k_{-n} \cup \{\mathbb{E}[x_n]\}, a; \bar{\phi})$. Then, when the ground BS bidder n is conditional on winning, it needs to pay the critical payment $\chi(\mathbf{x}_{-n};\bar{\phi})$. As $\rho^{a_k/|\mathcal{A}|} \geq 1$, only the ground BS with the highest bid bidder can win, which can satisfy the condition that $\chi(\mathbf{x}_{-n};\bar{\phi}) \geq \max\{\mathbf{x}_{-n}\}$. In addition, the critical payment function of DQMSB $\chi(\mathbf{x}_{-n};\bar{\phi}) = \rho^{a_k/|\mathcal{A}|}\max\{\mathbf{x^k_{-n}}\}$ satisfies

$$\begin{aligned}\chi(\max\{\mathbf{x}_{-n}\};\bar{\phi}) &= \rho^{a_k/|\mathcal{A}|}\max\{\max\{\mathbf{x}_{-n}\}\}\\ &= \rho^{a_k/|\mathcal{A}|}\max\{\mathbf{x}_{-n}\}\\ &= \chi(\mathbf{x}_{-n};\bar{\phi}).\end{aligned} \tag{35}$$

Therefore, considering there are two bidders in the market with one value higher than $\chi(\mathbf{x}_{-n};\bar{\phi})$ and the other one value $\max\{\mathbf{x}_{-n}\}$, which will cause $\chi(\max\{\mathbf{x}_{-n}\};\bar{\phi}) \neq \chi(\mathbf{x}_{-n};\bar{\phi})$, the DQMSB auction cannot satisfy false-name proof. Specifically, when $\chi(\mathbf{x}_{-n};\bar{\phi}) < \chi(\max\{\mathbf{x}_{-n}\};\bar{\phi})$, the first bidder can submit a lower bid while maintaining the other bids in the set of bids and thus the auction is not winner false-name proof. Otherwise, the auction is not loser false-name proof when $\chi(\mathbf{x}_{-n};\bar{\phi}) > \chi(\max\{\mathbf{x_{-n}}\};\bar{\phi})$, where the losing bidder in the market can submit a higher bid compared with the winner's bid while maintaining the other bids in the set of bids $\mathbf{x}_{-n}$. Furthermore, the critical payment function χ of the DQMSB auction is homogeneous of degree one, which indicates that the auction is adverse selection-free. Suppose that χ is not homogeneous of degree one, a bidder could manipulate the system by adjusting their bid in response to their private information $C \in \{1, c\}$, i.e., $\chi(\mathbf{m}_{-n};\bar{\phi}) < \chi(c\mathbf{m}_{-n};\bar{\phi})/c$, where $c \in \mathbb{R}_+, n \geq 2$, and $\mathbf{x}_{-n} \in \mathbb{R}_+^{n-1}$ and $C = 1$, $z_n(C\mathbf{m}) = z_n(\mathbf{m}) = \mathbf{1}_{\{m_n > \chi(m_{-n};\bar{\phi})\}} = 1$, so $z_0(C\mathbf{m}) = 0$. When $C \neq 1$, it indicates that the bidder can change its bid



(a) Average total cost versus time steps.

(b) Average total cost versus number of services.

(c) Average total cost versus number of GPUs.

(d) Average total cost versus number of users.

Fig. 3. Performance of model caching algorithms under different system settings.

from $C = c$ to influence the probability of winning the auction, i.e., $z_n(C\mathbf{m}) = z_n(c\mathbf{m}) = \mathbf{1}(cm_n > \chi(cm_{-n}; \bar{\phi})) = 0$. However, the ground BS bidder n with the highest bid cannot win the auction, and thus the satellite bidder 0 wins the auction, i.e., $z_0(C\mathbf{m}) = 1$.

Based on the above analysis, the proposed DQMSB can guarantee the anonymous, fully strategy-proof, and adverse-selection-free with the deterministic $Q_{\bar{\phi}}$. □

Theorem 2 describes MSB auctions as robust auctions that are anonymous, deterministic, not prone to adverse selection, and fully strategy-proof. It is important to note that the authors based on other DRL algorithms may possess any three of these characteristics. By allowing non-anonymity, the value of ρ determined by DRL algorithms can vary depending on the bids of the current bidders. Furthermore, as the DQMSB mechanism modifies only the selection of the price-scaling factor ρ, while fully preserving the standard second-price allocation and payment rules, the existing proof of strategy-proofness continues to hold. Consequently, because the DQN-based approach solely adjusts ρ without altering the underlying MSB structure, the mechanism's inherent robustness against adverse selection is likewise maintained.

VI. Experimental Results

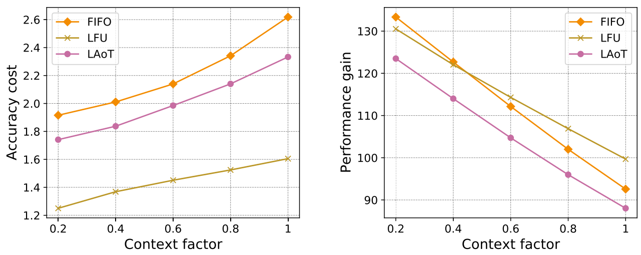
In this section, we validate the joint model caching and inference framework while evaluating the performance of the proposed least AoT cached model replacement algorithm and the DQMSB auction.

A. Parameter Settings

We consider the SAGINs with one satellite and multiple ground BSs to provision LLM agent services to users. We evaluate the proposed algorithm within 100 time steps. For each GPU in the edge servers of ground BSs, the memory is 80 GB, energy efficiency is 810 GFLOPS/W, and energy capacity is 300 W. The default number of services is set to 10, the default number of GPUs at ground BSs is set to 24, and the default number of users is set to 10. We leverage ImageBind [37] as the multimodal perception module of LLM agents, whose performance is 77.7% for images, 50.0% for videos, 63.4% for infrared, 54.0% for depth, 66.9% for audio, and 25.0% for IMU. We consider two types of LLMs for performing CoT reasoning, including the LLAMA-65B and GPT3-174B, whose context windows are 2k and 8k tokens, respectively. The total number of reasoning LLMs is set to 10. The runtime service data-traces is generated by assigning each active service (e.g., service = 10) to a fixed model using a random mapping, and at each time step, we simulate request counts using independent Poisson processes with a mean of number of users / 10 (default users = 10, so λ = 1). The default context vanishing factor Δ_n^t is set to 0.6. For each CoT reasoning example, the maximum size is set to 200 tokens. The transmit power of users is set to 0.2 W, and the allocated bandwidth is set to 20 MHz. The size of the input data of LLM agent services is uniformly selected from [100, 200] MB. The quantity of LLM service requests is generated from the Poisson point process, depending on the number of users. Due to the ratio of existing ground BSs and communication satellites, the edge access cost for satellites is set to 0.005 and 0.0001 for ground BSs. The cloud access cost is set to 0.04 for ground BSs and 0.025 for satellites. For the LLM agent service market, the number of BSs is set to 5 by default. The satellite-related settings follow [2] and the DRL-related parameter settings follow [38]. The evaluation adopts three layers of metrics. The first layer disaggregates total cost into switching, accuracy, edge inference, and cloud inference, enabling pinpoint diagnosis of which operations drive expenditure under diverse workload and hardware conditions. The second layer studies accuracy, cost, and normalised performance gain versus the context factor, revealing a trade-off between demonstration freshness and computational load. The third layer quantifies total surplus and its division between satellites and ground base stations under different auction baselines, thereby translating technical performance into economic welfare. Together, these metrics expose how the proposed AoT-guided caching and DQMSB auction jointly improve resource usage, user experience, and operator revenue.

B. Performance Evaluation of the Least AoT Algorithm

During the performance evaluation of the proposed model caching algorithm, we leverage several traditional caching baselines for comparison, including first-in-first-out (FIFO) and least frequently used (LFU) algorithms. As we can observe from Fig. 3, across the entire grid, the satellite-enabled architecture cuts total provisioning cost by an average of 41% and by no less than 20% in every individual configuration, chiefly because it shifts long-haul traffic away from congested ground gateways and, unlike local caching strategies,



(a) Accuracy cost versus vanishing factor.

(b) Performance gain versus vanishing factor.

Fig. 4. Model performance under different vanishing factors.

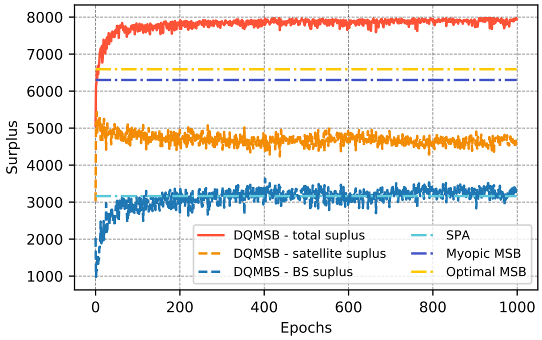


Fig. 5. The convergence of the proposed DQMSB auction.

introduces neither switching nor accuracy penalties. For instance, Fig 3(a) presents the average cost of different caching and offloading strategies (A-E) over a series of time slots. The costs are divided into four categories: switching cost, accuracy cost, edge inference cost, and cloud inference cost. For all strategies, there is a clear trend of decreasing average cost as the number of time slots increases. This indicates that over time, the systems may become more efficient, possibly due to improved caching optimization or better distribution of LLM agent tasks between the edge and the cloud. In addition, Fig. 3(b) demonstrates the increase of the average total cost for different caching and offloading schemes as the number of services increases. This suggests that as the system has to handle a larger variety of services, the associated costs rise, possibly due to increased complexity and demand for resources. The least AoT algorithm demonstrates a consistent performance advantage, maintaining the lowest average total cost across different numbers of services.

Furthermore, Fig. 3(c) illustrates how the average total cost changes for various caching and offloading schemes with the number of GPUs utilized. As we can observe, more available GPUs tend to lower the cost, likely due to the improved computational efficiency and reduced processing time at the edge. The relatively stable switching and accuracy costs across different GPU counts suggest these costs are more dependent on the efficiency of the algorithm itself rather than on hardware resources. Meanwhile, edge inference cost reductions point to the benefits of local processing power, highlighting the importance of edge capabilities in managing LLM caching. Overall, the average total cost decreases for most schemes as GPU resources increase. Finally, Fig. 3(d) demonstrates an upward trend in average total cost for all schemes as the number of users increases. This suggests that the system's costs escalate with the growing user base, likely due to increased demand for LLM services, which intensifies the load on caching and computation resources. The increase in average total cost across all schemes with more users suggests that user demand has a direct impact on the system's resource utilization and cost efficiency. The least AoT algorithm demonstrates scalability by maintaining the lowest increase in cost, indicating its potential for cost-effective expansion as user numbers grow. The relative stability of the switching cost across varying user counts may imply that the action of switching between cached LLMs does not contribute significantly to cost variations.

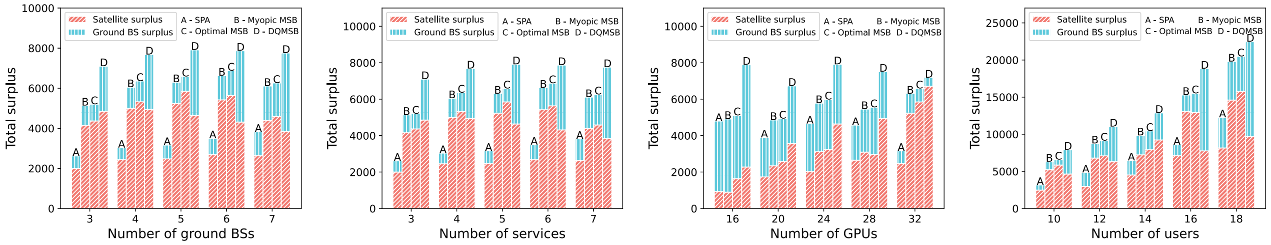
As shown in Fig. 4(a), as the context factor increases, the cost of accuracy also increases for all three algorithms (FIFO, LFU, and least AoT). This indicates that when the context becomes more influential, which can be seen as the importance of cache relevance, it becomes more challenging and expensive for all methods to maintain high accuracy in caching decisions. Meanwhile, in Fig. 3(b), there is a decrease in performance gain for all algorithms as the context factor increases. This suggests that as the importance of the context in caching decisions increases, the performance gain of these algorithms decreases. The overall decrease in performance gain with a higher context factor suggests that as the relevance of the context increases, the potential for any algorithm to outperform basic caching strategies diminishes. This might be because the decision-making process becomes more complex and the benefits of sophisticated strategies are less pronounced.

C. Convergence Analysis

Initially, we demonstrate the convergence performance of the DQMSB auction in Fig. 5. At the beginning of training, the total surplus achieved by the DQMSB auction starts with a sharp increase and then levels off, indicating that the mechanism quickly learns an effective strategy for maximizing surplus and then converges to a stable solution. At around 200 epochs, the total surplus stabilizes at a high level. Although there are minor fluctuations following this rise, the surplus remains relatively consistent, indicating that the system has reached a convergence in its learning phase. Interestingly, the performance of DQMBS in ground BS surplus can achieve similar performance to the SPA, at around 3,000. This indicates that the SPA can realize the surplus of satellites, which can reach around 4,500 for the optimal solution. Furthermore, the total surplus achieved by the DQMSB auction can outperform the MSB auction by around 20%. The convergence performance of the DQMSB auction can be considered constantly robust, as it achieves and maintains a higher total surplus compared to the benchmarks.

D. Performance Evaluation for the DQMSB Auction

To evaluate the performance of the DQMSB auction, we leverage several auction baselines, including the second-price auction, myopic MSB, and optimal MSB. Particularly, the price scaling factor of myopic MSB is set as $\rho = \max(1, x_0/x_{(2)})$ with current round information and the price scaling factor of optimal MSB can be set as $\rho = \max(1, \mathbb{E}[x_0]/\mathbb{E}[x_{(2)}])$ with historical statistic information [14]. Under different numbers of bidders, Fig. 6(a)



(a) Total surplus versus number of ground BSs.

(b) Total surplus versus number of services.

(c) Total surplus versus number of GPUs.

(d) Total surplus versus number of users.

Fig. 6. Performance of the proposed DQMSB auction under different system settings.

demonstrates the total surplus achieved by various auction mechanisms based on the number of ground BSs. Across all auction mechanisms, the total surplus increases as the number of ground BSs grows from 3 to 7. This suggests that a larger number of ground BSs enables better service provision and coverage, resulting in a higher overall value generated by the auctions. The DQMSB auction consistently yields the highest total surplus, regardless of the number of ground BSs. This indicates that the DQMSB auction is more efficient in resource allocation and surplus generation compared to other auction types. The Myopic MSB and Optimal MSB auctions outperform SPA but fall short of the performance achieved by the DQMSB auction. As we can observe from Fig. 6(b), the total surplus does not monotonically increase or decrease with the number of services. For instance, there is a drop in total surplus for all mechanisms when varying from 10 to 12 services, followed by an increase at 14 services, another decrease at 16, and an increase again at 18. In most cases, the surplus from ground BS services is greater than the surplus from satellite services. This non-linear relationship implies that simply increasing the number of services cannot guarantee a higher surplus.

In Fig. 6(c), there appears to be a general increase in the total surplus for all the auctions as the number of GPUs increases. The surplus for the other mechanisms also tends to increase, although not as consistently or significantly as the DQMSB auction. The ground BS surplus dominates the total surplus for all auction mechanisms and quantities of GPUs. However, as the number of GPUs increases, the satellite surplus also increases, suggesting a positive relationship between computing resources and the ability to generate surplus in satellite-based services. The increasing trend of total surplus with more GPUs implies that having more computing resources allows the auction mechanisms, particularly DQMSB, to perform better. Finally, Fig. 6(d) demonstrates an upward trend in the total surplus with an increasing number of users for all auction mechanisms, which suggests that more users contribute to a higher valuation and competition, thus increasing the total surplus. When the number of users increases, the DQMSB auction can yield a larger surplus from ground BSs, thereby limiting computing and communication resources that can be allocated effectively to maximize the total surplus.

VII. Conclusion

In this paper, we proposed a joint caching and inference framework for provisioning ubiquitous edge intelligence services in SAGINs. In SAGINs, satellites and ground BSs were utilized to provision global LLM agent services with edge servers at ground BSs or remote cloud data centers. Specifically, considering the unique few-shot learning capabilities of LLMs and new constraints on the size of context windows, we introduced a new concept, i.e., the cached model as a resource, beyond conventional communication, computing, and storage resources. For allocating cached model resources, we designed a new metric, namely, age of thought, to evaluate the relevance and consistency of thoughts/CoT examples in context windows during inferences and proposed the least AoT algorithm. Finally, we proposed the DQMSB for incentivizing network operators to provision LLM agents with high market efficiency through the DQN-based price scaling factor. Theoretically, we proved that the proposed DQMSB auction is anonymous, fully strategy-proof, and adverse-selection-free. Future work will involve building a prototype platform and implementing representative LLM-agent applications, enabling the continuous collection of communication and computing traces to support empirical validation in real-world scenarios. In addition, we will extend the metric set to include energy consumption, carbon emission, per-service fairness, and end-to-end latency distribution, leveraging a hardware-in-the-loop prototype for empirical validation at a larger scale.

References

[1] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Commun. Surveys Tut.*, vol. 20, no. 4, pp. 2714–2741, 4th Quart., 2018.

[2] Q. Tang, Z. Fei, B. Li, and Z. Han, "Computation offloading in LEO satellite networks with hybrid cloud and edge computing," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9164–9176, Jun. 2021.

[3] T. Wei, W. Feng, Y. Chen, C.-X. Wang, N. Ge, and J. Lu, "Hybrid satellite-terrestrial communication networks for the maritime Internet of Things: Key technologies, opportunities, and challenges," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8910–8934, Jun. 2021.

[4] B. Min et al., "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Comput. Surv.*, vol. 56, no. 2, pp. 1–40, Sep. 2023.

[5] M. Xu et al., "Sparks of generative pretrained transformers in edge intelligence for the metaverse: Caching and inference for mobile artificial intelligence-generated content services," *IEEE Veh. Technol. Mag.*, vol. 18, no. 4, pp. 35–44, Dec. 2023.

[6] M. Xu et al., "When large language model agents meet 6G networks: Perception, grounding, and alignment," 2024, *arXiv:2401.07764*.

[7] Z. Xi et al., "The rise and potential of large language model based agents: A survey," 2023, *arXiv:2309.07864*.

[8] T. B. Brown et al., "Language models are few-shot learners," in *Proc. NIPS*, vol. 33, 2020, pp. 1877–1901.

[9] H. Yang, M. Siew, and C. Joe-Wong, "An LLM-based digital twin for optimizing human-in–the loop systems," 2024, *arXiv:2403.16809*.

[10] S. Jiang and J. Wu, “Approaching an optimal Bitcoin mining overlay,” *IEEE/ACM Trans. Netw.*, vol. 31, no. 5, pp. 2013–2026, Oct. 2023.

[11] M. Xu et al., “Unleashing the power of edge-cloud generative AI in mobile networks: A survey of AIGC services,” *IEEE Commun. Surveys Tut.*, vol. 26, no. 2, pp. 1127–1170, 2nd Quart., 2024.

[12] C. Packer et al., “MemGPT: Towards LLMs as operating systems,” 2023, *arXiv:2310.08560*.

[13] J. Yang, “LongQLoRA: Efficient and effective method to extend context length of large language models,” 2023, *arXiv:2311.04879*.

[14] N. Arnosti, M. Beck, and P. Milgrom, “Adverse selection and auction design for internet display advertising,” *Amer. Econ. Rev.*, vol. 106, no. 10, pp. 2852–2866, Oct. 2016.

[15] X. Xu, Q. Wang, Y. Hou, and S. Wang, “AI-SPACE: A cloud-edge aggregated artificial intelligent architecture for tiansuan constellation-assisted space-terrestrial integrated networks,” *IEEE Netw.*, vol. 37, no. 2, pp. 22–28, Mar. 2023.

[16] X. Hou, J. Wang, Z. Fang, Y. Ren, K.-C. Chen, and L. Hanzo, “Edge intelligence for mission-critical 6G services in space-air-ground integrated networks,” *IEEE Netw.*, vol. 36, no. 2, pp. 181–189, Mar. 2022.

[17] P. Qin, M. Wang, X. Zhao, and S. Geng, “Content service oriented resource allocation for space–air–ground integrated 6G networks: A three-sided cyclic matching approach,” *IEEE Internet Things J.*, vol. 10, no. 1, pp. 828–839, Jan. 2023.

[18] Z. Chen, Z. Zhang, and Z. Yang, “Big AI models for 6G wireless networks: Opportunities, challenges, and research directions,” 2023, *arXiv:2308.06250*.

[19] Y. Shen et al., “Large language models empowered autonomous edge AI for connected intelligence,” *IEEE Commun. Mag.*, vol. 62, no. 10, pp. 1–7, Oct. 2024.

[20] Y. Du, H. Deng, S. Chang Liew, K. Chen, Y. Shao, and H. Chen, “The power of large language models for wireless communication system development: A case study on FPGA platforms,” 2023, *arXiv:2307.07319*.

[21] H. Cui, Y. Du, Q. Yang, Y. Shao, and S. Chang Liew, “LLMind: Orchestrating AI and IoT with LLM for complex task execution,” 2023, *arXiv:2312.09007*.

[22] F. Jiang et al., “Large language model enhanced multi-agent systems for 6G communications,” 2023, *arXiv:2312.07850*.

[23] L. Dong et al., “LAMBO: Large AI model empowered edge intelligence,” 2023, *arXiv:2308.15078*.

[24] Z. Lin, G. Qu, Q. Chen, X. Chen, Z. Chen, and K. Huang, “Pushing large language models to the 6G edge: Vision, challenges, and opportunities,” 2023, *arXiv:2309.16739*.

[25] M. Zhang, L. Yang, S. He, M. Li, and J. Zhang, “Privacy-preserving data aggregation for mobile crowdsensing with externality: An auction approach,” *IEEE/ACM Trans. Netw.*, vol. 29, no. 3, pp. 1046–1059, Jun. 2021.

[26] D. Niyato, N. C. Luong, P. Wang, and Z. Han, *Auction Theory for Computer Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[27] J. Du, C. Jiang, H. Zhang, Y. Ren, and M. Guizani, “Auction design and analysis for SDN-based traffic offloading in hybrid satellite-terrestrial networks,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2202–2217, Oct. 2018.

[28] Q. Chen, W. Meng, S. Han, and C. Li, “Service-oriented fair resource allocation and auction for civil aircrafts augmented space-air-ground integrated networks,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13658–13672, Nov. 2020.

[29] N. Yang, D. Guo, Y. Jiao, G. Ding, and T. Qu, “Lightweight blockchain-based secure spectrum sharing in space–air–ground-integrated IoT network,” *IEEE Internet Things J.*, vol. 10, no. 23, pp. 20511–20527, Dec. 2023.

[30] R. Deng, B. Di, S. Chen, S. Sun, and L. Song, “Ultra-dense LEO satellite offloading for terrestrial networks: How much to pay the satellite operator?,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6240–6254, Oct. 2020.

[31] N. Cheng et al., “Space/aerial-assisted computing offloading for IoT applications: A learning-based approach,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[32] Z. Zhang et al., “Igniting language intelligence: The Hitchhiker’s guide from chain-of-thought reasoning to language agents,” 2023, *arXiv:2311.11797*.

[33] H. Jiang, “A latent space theory for emergent abilities in large language models,” 2023, *arXiv:2304.09960*.

[34] R. Tutunov, A. Grosnit, J. Ziomek, J. Wang, and H. Bou-Ammar, “Why can large language models generate correct chain-of-thoughts?,” 2023, *arXiv:2310.13571*.

[35] D. Dai et al., “Why can GPT learn in-context? Language models secretly perform gradient descent as meta-optimizers,” in *Proc. Findings Assoc. Comput. Linguistics (ACL)*, Toronto, ON, Canada, Jul. 2023, pp. 4005–4019.

[36] K. Zhao et al., “EdgeAdaptor: Online configuration adaption, model selection and resource provisioning for edge DNN inference serving at scale,” *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5870–5886, Oct. 2023.

[37] R. Girdhar et al., “ImageBind one embedding space to bind them all,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos, CA, USA, Jun. 2023, pp. 15180–15190.

[38] J. Weng et al., “Tianshou: A highly modularized deep reinforcement learning library,” *J. Mach. Learn. Res.*, vol. 23, no. 1, pp. 12275–12280, Jan. 2022.

Minrui Xu (Member, IEEE) received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2021. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include large language models over networks, Metaverse, deep reinforcement learning, and mechanism design. He was a recipient of the IEEE VTS Daniel E. Noble Fellowship and the 2025 IEEE ICC Best Paper Award.

Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkut’s Institute of Technology Ladkrabang (KMITL), Thailand, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Canada. He is currently a Professor with the College of Computing and Data Science, Nanyang Technological University, Singapore. His research interests include mobile generative AI, edge general intelligence, quantum computing and networking, and incentive mechanism design.

Hongliang Zhang (Senior Member, IEEE) is currently an Endowed Boya Young Fellow Assistant Professor with the School of Electronics, Peking University. His current research interests include intelligent surfaces, aerial access networks, and the Internet of Things. He was a recipient of the IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award, the IEEE ComSoc Best Tutorial Paper Award, the IEEE Comsoc Heinrich Hertz Award for Best Communications Letters, the IEEE Neal Shepherd Memorial Best Propagation Paper Award, the IEEE ComSoc Asia–Pacific Outstanding Paper Award, the IEEE GLOBECOM Best Paper Award, and the IEEE/CIC ICCC Best Demo Award. He is an Editor of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS LETTERS, and *IET Communications*. He was an Exemplary Editor of IEEE COMMUNICATIONS LETTERS in 2023.

Jiawen Kang (Senior Member, IEEE) received the Ph.D. degree from Guangdong University of Technology, China, in 2018. He has been a Post-Doctoral Researcher at Nanyang Technological University, Singapore, from 2018 to 2021. He currently is a Full Professor with Guangdong University of Technology. His research interests include blockchain, security, and privacy protection in wireless communications and networking.

Zehui Xiong (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University (NTU). He is currently a Full Professor with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, U.K. Prior to that, he was with Singapore University of Technology and Design and NTU. He was a Visiting Scholar with Princeton University and the University of Waterloo. Recognized as a Clarivate Highly Cited Researcher, he has published over 250 peer-reviewed research articles in leading journals, with numerous Best Paper Awards from international flagship conferences. Featured in Forbes Asia 30U30, he serves as an editor for many leading journals and the chair for numerous international conferences. His honors include the IEEE Asia–Pacific Outstanding Young Researcher Award, the IEEE VTS Early Career Award, the IEEE Early Career Award for Excellence in Scalable Computing, the IEEE Technical Committee on Blockchain and Distributed Ledger Technologies Early Career Award, the IEEE Internet Technical Committee Early Achievement Award, the IEEE TCSVC Rising Star Award, the IEEE TCI Rising Star Award, the IEEE TCCLD Rising Star Award, the IEEE ComSoc Outstanding Paper Award, the IEEE Best Land Transport Paper Award, the IEEE Asia–Pacific Outstanding Paper Award, the IEEE CSIM Technical Committee Best Journal Paper Award, the IEEE SPCC Technical Committee Best Paper Award, and the IEEE Big Data Best Influential Conference Paper Award.

Shiwen Mao (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Polytechnic University in 2004. He is currently a Professor, an Earle C. Williams Eminent Scholar, and the Director of the Wireless Engineering Research and Education Center, Auburn University. His research interests include wireless networks, multimedia communications, and smart grid. He is the Editor-in-Chief of IEEE Transactions on Cognitive Communications and Networking, a Member-at-Large on the Board of Governors of the IEEE Communications Society, and the Vice President of Technical Activities of the IEEE Council on Radio Frequency Identification (CRFID). He received the IEEE ComSoc MMTC Outstanding Researcher Award in 2023, the SEC 2023 Faculty Achievement Award for Auburn, the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award in 2019, the Auburn University Creative Research and Scholarship Award in 2018, the NSF CAREER Award in 2010, and several service awards from IEEE ComSoc. He was a co-recipient of the 2022 Best Journal Paper Award of the IEEE ComSoc eHealth Technical Committee, the 2021 Best Paper Award of Elsevier/KeAi Digital Communications and Networks Journal, the 2021 IEEE Internet of Things Journal Best Paper Award, the 2021 IEEE Communications Society Outstanding Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the 2018 Best Journal Paper Award and the 2017 Best Conference Paper Award from IEEE ComSoc MMTC, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the field of communications systems. He was a co-recipient of the Best Paper/Demo Awards of 12 conferences.

Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University in 1997 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1999 and 2003, respectively. From 2000 to 2002, he was a Research and Development Engineer with JDSU, Germantown, MD. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, ID. Currently, he is a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department and the Computer Science Department, University of Houston, Houston, TX, USA. His main research targets on the novel game-theory related concepts critical to enabling efficient and distributive use of wireless networks with limited resources. His other research interests include wireless resource allocation and management, wireless communications and networking, quantum computing, data science, smart grid, carbon neutralization, and security and privacy. He has been an AAAS Fellow since 2019 and an ACM Fellow since 2024. He was the winner of the 2021 IEEE Kiyo Tomiyasu Award (an IEEE Field Award), for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks." He received the NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for *EURASIP Journal on Advances in Signal Processing* in 2015, the IEEE Leonard G. Abraham Prize in the field of communications systems (Best Paper Award in IEEE Journal on Selected Areas in Communications) in 2016, the IEEE Vehicular Technology Society 2022 Best Land Transportation Paper Award, and several best paper awards in IEEE conferences. He has been a 1% Highly Cited Researcher according to Web of Science since 2017. He was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018 and an ACM Distinguished Speaker from 2022 to 2025.