Optimizing AIGC Services by Prompt Engineering and Edge Computing: A Generative Diffusion Model-Based Contract Theory Approach

Dongdong Ye, Shuting Cai, Hongyang Du, Jiawen Kang, Yinqiu Liu, Rong Yu, *Member*, *IEEE*, and Dusit Niyato, *Fellow*, *IEEE*

***Abstract*—The development of Generative AI (GAI) and AI-generated content (AIGC) has been significantly improved by pretrained foundation models and prompt-based methods. To boost the quality and reduce the latency of AIGC generation, prompt engineering and edge computing are introduced, demanding a multi-dimensional resource allocation approach. Thus, we use the generative diffusion model (GDM) and contract theory to design a two-stage, multi-dimensional resource allocation framework. In the first stage, we employ an approximation approach to quantitatively assess the relationship between the level of prompt optimization, the number of diffusion denoising steps, and the quality of AIGC generation. Based on the quality function, we formulate models for the utilities of an AI-generated content Service Provider (ASP) and users, leading to a non-convex quality-based contract problem optimizing the level of prompt optimization and the number of diffusion denoising steps. To address the time-consuming process of solving the non-convex problem due to variable cost of the ASP and gain preferences of the users, a GDM-based scheme is proposed to optimize quality-based contract items. In the second stage, for each group of users who choose the same quality-based contract items, a non-convex latency-based contract problem optimizing the CPU cycle frequency and network transmission rate is formulated, then the GDM-based scheme is also applied to find the optimal latency-based contract items. Numerical results show that the proposed GDM-based contract generation scheme is very advantageous in improving the quality of AIGC generation and decreasing the latency of AIGC generation, compared to other standard schemes.**

***Index Terms*—Edge computing, prompt engineering, AI-generated content, contract theory, generative diffusion model.**

This work was supported by the Key Area R & D Program of Guangdong Province (No. 2022B0701180001); the National Natural Science Foundation of China (No. 62102099); the Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515140137); the National Natural Science Foundation of China (No. U22A2054); the National Research Foundation Singapore and Infocomm Media Development Authority under its Future Communications Research & Development Programme; the Defence Science Organisation National Laboratories under the AI Singapore Programme (FCP-NTU-RG-2022-010 and FCP-ASTAR-TG-2022-003); the Singapore Ministry of Education Tier 1 (RG87/22); and the NTU Centre for Computational Technologies in Finance (NTU-CCTF).

Shuting Cai, Dongdong Ye, Jiawen Kang and Rong Yu are with School of Automation, Guangdong University of Technology, Guangzhou, China, 510006 (e-mail: shutingcai@gdut.edu.cn; dongdongye8@163.com; kavinkang@gdut.edu.cn; yurong@ieee.org).

Yinqiu Liu and Dusit Niyato are with the College of Computing and Data Science, Nanyang Technological University, Singapore (e-mail: yinqiu001@e.ntu.edu.sg, dniyato@ntu.edu.sg).

Hongyang Du is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pok Fu Lam, Hong Kong (e-mail: duhy@eee.hku.hk).

Corresponding author: Rong Yu.

I. Introduction

As a cornerstone of Artificial Intelligence Generated Content (AIGC), Generative Artificial Intelligence (GAI) is projected to contribute approximately 7 trillion to the global economy, enhancing the overall economic impact of Artificial Intelligence (AI) by around 50% [1]. Specifically, in natural language processing, GAI could generate nearly 2 trillion dollars in value through advanced applications such as chatbots and text summarization. For example, ChatGPT exemplifies this with its sophisticated conversational capabilities. In computer vision, GAI facilitates image editing and virtual reality, also expected to add nearly 2 trillion dollars in value. For example, DALL-E 3 illustrates this by converting textual descriptions into images, blending linguistic comprehension with visual creativity, underscoring the expanding role of machines in creative domains traditionally dominated by humans.

For many years, GAI has been the subject of research and has gone through multiple iterations. In particular, the introduction of Pretrained Foundation Models (PFMs) and prompt-based techniques has made it much easier to create GAI and AIGC. The "PFM + prompt" paradigm expands its applications beyond multimedia creation, including channel coding [1], network design [2], and defenses [3]. However, the paradigm faces challenges such as resource limitations and low-quality prompts.

- **Resource Limitations**: PFMs, densely packed with an extensive array of parameters, are notably resource-intensive. For example, deploying models such as GPT-3 requires at least one NVIDIA Ampere or a newer GPU, equipped with no less than eight gigabytes of GPU memory. Additionally, each cycle of generative inference consumes a considerable amount of computing power. This significant resource consumption undeniably poses a formidable barrier for numerous mobile users constrained by limited resources [4].
- **Low-Quality Prompts**: Users without proper training often find it challenging to create effective professional prompts for PFMs, especially when dealing with complex downstream tasks or when the PFM has hidden requirements. Poor-quality prompts can degrade the generation quality of PFMs and result in more frequent regenerations, leading to increased service delays [5].

However, there is hope in the potential of edge computing and prompt engineering to tackle these challenges. Edge

servers enable the local deployment of PFMs to act as AI-generated Content Service Providers (ASPs), offering AIGC services to mobile users [6]. The effectiveness of these mobile-edge AIGC services hinges on the strategic use of prompt engineering [5]. In this context, prompt engineering addresses the challenges of network resource optimization by treating prompts as critical variables. These prompts are carefully chosen, designed, and optimized to meet user needs while complying with network limitations. The benefits of this approach in prompt engineering are multifold.

- **Improving Quality of AIGC Generation**: The usefulness of PFMs may increase by providing the most appropriate prompts. The results in [7] found that optimizing the prompts can increase user satisfaction with produced images by 380%.
- **Reducing Latency of AIGC Generation**: Reducing the number of generation attempts directly decreases service latency, which, in turn, enhances user satisfaction, as satisfaction is inversely related to latency [4].
- **Reducing Energy Consumption**: Mobile edge networks are heavily dependent on resource efficiency. Reducing the number of re-generations may help save bandwidth and computing resources.

Despite these advantages, the following challenges are faced. Firstly, how do we quantify the relationship between the level of prompt optimization and the quality of AIGC generation, and identify the optimal level to meet the quality needs of users for AIGC services? Secondly, to improve the Quality of Experience (QoE) for users, multi-dimensional resource optimization strategies such as the level of prompt optimization, the number of diffusion denoising steps, CPU cycle frequency, and network transmission rate, must be implemented in resource-sparse edge networks. Lastly, ASPs that provide AIGC services require users to make payments to access these services. Since users are driven to maximize their own benefits, it is unrealistic to expect them to unconditionally comply with the ASPs' instructions.

Thus, we propose a two-stage, multi-dimensional resource allocation framework that utilizes a Generative Diffusion Model (GDM) and contract theory to enhance the quality and reduce the latency of AIGC generation. Although this paper primarily focuses on text-generated image services, the framework is adaptable to other types of AIGC services as well. In the first stage, a neural image assessment model [8] is used to assess the quality of image generation. Subsequently, an approximation approach is used to quantify the relationship between the level of prompt optimization, the number of diffusion denoising steps, and the quality of image generation. This method is a common practice in the literature and has been adopted in other studies, such as [9], [10]. Based on the approximation relationship, we establish models for the utilities of an ASP and users, leading to a non-convex quality-based contract problem optimizing the level of prompt optimization and the number of diffusion denoising steps. Variable gain per quality of image generation and variable cost of the ASP in mobile environments require the continual re-solution of the non-convex problem, making it more time-consuming to obtain the optimal quality-based contract items using conventional mathematical techniques. Fortunately, a GDM-based scheme is capable of handling the above issue [1]. It has been applied in various areas, such as blockchain, vehicular networks, vehicular metaverses, and information sharing in full-duplex semantic communications [1]. Thus, we employ the GDM-based scheme for optimal quality-based contract items. In the second stage, users first select a quality-based contract item that aligns with their type of gain per quality. Then, for each group of users who choose the same quality-based contract items, we formulate a non-convex latency-based contract problem optimizing the CPU cycle frequency and network transmission rate. The GDM-based scheme is also applied to find the optimal solution for the latency-based contract problem. The main contributions of this paper are summarized as follows:

- A curve approximation approach is employed to model users' QoE including the quality of AIGC generation and the latency reduction of AIGC generation. Based on the QoE, a quality-based contract problem and a latency-based contract problem between the ASP and users are formulated to maximize the utility of the ASP sequentially.
- Due to the users' variable gain and the ASP' variable cost, non-convex quality-based and latency-based contract problems must be solved repeatedly, which takes longer using traditional mathematical methods. To efficiently find the optimal quality-latency-based contract items, we propose a novel two-stage GDM-based scheme.
- Numerical results show that the proposed two-stage GDM-based contract generation scheme is very advantageous in improving the quality of AIGC generation and decreasing the latency of AIGC generation, compared to other baseline schemes. The effectiveness of the proposed scheme has also been confirmed.

The rest of this paper is structured as follows. The related work is presented in Section II. The system model is presented in Section III. GDM-based quality contract design is introduced in Section IV. GDM-based latency contract design is introduced in Section V. Section VI shows the performance evaluation. Section VII concludes this paper. Table I lists the notation frequently used in the paper.

II. Related Work

Recently, much attention has been paid to AIGC services in edge computing, including improving their performance and implementing incentive mechanisms. This section will focus on two of the most pertinent aspects of our research.

A. Performance Enhancement for AIGC services

Enhancing the performance of AIGC services within edge networks necessitates the strategic optimization of wireless resource allocation. The authors in [11] and [12] introduced a system for efficient model management and resource allocation to meet user needs, proposed a metric called 'age of context' for task relevance, and optimized edge server caching considering latency, energy, and accuracy. The authors

TABLE I: Summary of main notations.

Notation	Definition	Notation	Definition
M	the number of users	A	the quality of image generation
D	the latency reduction of image generation	g	re-generate the image for the g-th time until the image generation quality is met
ζ	the probability of achieving a certain threshold $\overline{A}$ for the quality of image generation	$\mathbb{E}[D]$	the expected latency reduction with the g-th generation of successful result
I	the number of types of gain per quality of image generation	θ_i^{A}	the type of the ith gain per quality of image generation
q_i^{A}	the probability that a user's type belongs to the type of the ith gain per quality	l_i^{A}	the level of prompt optimization for type-θ_i^{A} user
s_i^{A}	the number of diffusion denoising steps for type-θ_i^{A} user	p_i^{A}	the reward paid to the ASP for type-θ_i^{A} user
$\sigma_{1,i}$	the cost per level of prompt optimization for type-θ_i^{A} user	$\sigma_{2,i}$	the cost per number of diffusion denoising steps for type-θ_i^{A} user
$\boldsymbol{\rho}$	the parameter vector fitted by experiments	M_i	the number of the users choosing the quality-based contract item $\Phi_i^{\mathrm{A}} = (l_i^{\mathrm{A}}, s_i^{\mathrm{A}}, p_i^{\mathrm{A}})$
m_i	a user choosing the quality-based contract item $\Phi_i^{\mathrm{A}} = (l_i^{\mathrm{A}}, s_i^{\mathrm{A}}, p_i^{\mathrm{A}})$	$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$	the type of the jth gain per expected latency reduction with the type of the ith gain per quality
J	the number of types of gain per expected latency reduction for M_i users	$q_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$	the probability that a user's type of gain per expected latency reduction belongs to type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$
$t_j^{\max}(\theta_i^{\mathrm{A}})$	the maximum latency for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user	$b_{1,j}(\theta_i^{\mathrm{A}})$	the cost per computation energy consumption of optimizing the prompt for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user
$b_{2,j}(\theta_i^{\mathrm{A}})$	the cost per computation energy consumption of executing diffusion denoising for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user	$b_{3,j}(\theta_i^{\mathrm{A}})$	the cost per communication energy consumption for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user
$\delta_j(\theta_i^{\mathrm{A}})$	the CPU frequency per level of prompt optimization for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user	$\eta_j(\theta_i^{\mathrm{A}})$	the CPU frequency per number of diffusion denoising steps for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user
$d_j(\theta_i^{\mathrm{A}})$	the size of the diffusion denoising result for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user	$h_j(\theta_i^{\mathrm{A}})$	the status of wireless connection for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user
$\kappa_j(\theta_i^{\mathrm{A}})$	the effective switched capacitance for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user	$x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$	the CPU frequency for optimizing prompt for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user
$y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$	the CPU frequency for diffusion denoising for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user	$r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$	the network transmission rate for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user
K^{A} or K^{T}	the number of iterations for adding noise in the quality-based or latency-based contract generation model	$\boldsymbol{\phi}_k^{\mathrm{A}}$ or $\boldsymbol{\phi}_k^{\mathrm{T}}$	the features of sample after iteratively adding k times of noise the features of sample in the quality-based or latency-based contract generation model
$\pi_{\omega^{\mathrm{A}}}^{\mathrm{A}}(\boldsymbol{\phi}^{\mathrm{A}}\|\boldsymbol{e}^{\mathrm{A}})$ or $\pi_{\omega^{\mathrm{T}}}^{\mathrm{T}}(\boldsymbol{\phi}^{\mathrm{T}}(\theta_i^{\mathrm{A}})\|\boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}))$	the quality-based or latency-based contract design policy	$\mathcal{N}^{\mathrm{A}}$ or $\mathcal{N}^{\mathrm{T}}$	Gaussian distribution in the quality-based or latency-based contract generation model
$\mu_{\omega^{\mathrm{A}}}^{\mathrm{A}}$ or $\mu_{\omega^{\mathrm{T}}}^{\mathrm{T}}$	the mean in the quality-based or latency-based contract generation model	$\boldsymbol{\Sigma}_{\omega^{\mathrm{A}}}$ or $\boldsymbol{\Sigma}_{\omega^{\mathrm{T}}}$	the covariance matrix in the quality-based or latency-based contract generation model
$\varepsilon_{\omega^{\mathrm{A}}}^{\mathrm{A}}$ or $\varepsilon_{\omega^{\mathrm{T}}}^{\mathrm{T}}$	the quality-based or latency-based contract design network	ω^{A} or ω^{T}	the weights of the quality-based or latency-based contract design network
$\varepsilon_{\omega^{\mathrm{A}}}^{\mathrm{A}}$ or $\varepsilon_{\omega^{\mathrm{T}}}^{\mathrm{T}}$	the quality-based or latency-based contract design network	H_v^{A} or H_v^{T}	the weights of the quality-based or latency-based contract evaluation network
$\mathcal{L}^{\mathrm{A}}(\omega^{\mathrm{A}})$ or $\mathcal{L}^{\mathrm{T}}(\omega^{\mathrm{T}})$	the loss function in the quality-based or latency-based contract generation model	N^{A} or N^{T}	the batch size in the quality-based or latency-based contract generation model
γ^{A} or γ^{T}	the discount factor in the quality-based or latency-based contract generation model	τ^{A} or τ^{T}	the soft target update parameter in the quality-based or latency-based contract generation model
ϵ^{A} or ϵ^{T}	the exploration noise in the quality-based or latency-based contract generation model	$\mathcal{B}^{\mathrm{A}}$ or $\mathcal{B}^{\mathrm{T}}$	the replay buffer in the quality-based or latency-based contract generation model

proposed a model linking computational resources with user quality metrics and recommend a deep reinforcement learning algorithm for the optimal selection of ASPs in wireless edge networks [15]. The authors in [13] proposed a novel deep q-network-based algorithm to address the challenge of selecting an ASP in healthcare consumer electronics, optimizing service provision and energy consumption through a markov decision process model. The authors initially proposed an AI-generated optimal decision algorithm using diffusion models for a better selection of ASPs. Furthermore, they improved it by integrating deep reinforcement learning, creating the soft actor-critic algorithm for deep diffusion for more efficient ASP selection [14]. The authors of [16] developed an algorithm using multi-agent reinforcement learning and soft actor critic methods to efficiently schedule AIGC workloads across multiple, distant data centers, excelling in resource utilization, cost-efficiency and reduction of carbon emissions.

B. Incentive Mechanism for AIGC services

Previous studies have taken an optimistic view that all edge servers owned by ASPs will contribute their resources without any conditions, which is not realistic in the real world due to the costs associated with running AIGC services. Therefore, in [17], the authors suggested a novel multiscale sequential perception approach to predict user skeletons from wireless signals and applied game theory to create a pricing strategy for service provisioning. The above work assumes that ASPs fully understand users' preferences of QoE, such as service

TABLE II: Performance Enhancement and Incentive Mechanism for AIGC Services in Edge Network.

Ref.	Optimization strategy	Optimization goal	Optimization approach
[11], [12]	Caching and offloading decisions	Minimize the system cost including the switching cost, the accuracy cost, the transmission cost, and the inference cost	Least context algorithm
[13]	ASP selection decision	Maximize quality and minimize energy of AIGC services	DQN-based algorithm
[14]	Number of diffusion steps	Maximize human-aware content quality of AIGC services	Diffusion model-based AI-generated optimal decision algorithm
[15]	ASP selection decision	Maximize the quality of generated content reward and a congestion penalty	Deep reinforcement learning-enabled algorithm
[16]	Duration of each task execution	Maximize the utility of the system related to revenue and energy cost	Algorithm based on multi-agent reinforcement learning and actor-critic methods
[17]	Basic fee and unit price, computing resources	Maximize user's utility	Generative AI-aided game theory
[18]	Latency of AIGC service	Security-latency metric	Generative AI-aided contract
[19]	Data update frequency	Maximize satisfaction function of AIGC services	Contract theory
[20]	Offloading decisions	maximize the task completion rate and minimize the average response time of AIGC services	Vision language model empowered contract theory
This paper	Level of prompt optimization, CPU cycle frequency, network transmission rate, number of diffusion denoising steps	Quality and latency of prompt engineering based AIGC services	Two-stage GDM-based contract design

delay, which is unrealistic and complicates the alignment of rewards with service quality. Thus, in [18], the authors used contract theory to create flexible contracts that motivate ASPs to provide their resources for AIGC mobile services. In [19], the authors use Age of Information (AoI) as a metric to measure the freshness of the data for AIGC fine-tuning. Subsequently, a contract theory model based on AoI is proposed to motivate UAVs to contribute fresh data. The aforementioned work mainly addresses service latency, overlooking the crucial aspect of service accuracy, which is vital for AIGC services. Thus, in [4], a framework is presented to improve user QoE and lower energy consumption in AIGC services, using edge devices and prompt engineering in a mobile edge environment. The authors in [20] integrated diffusion-based AIGC models for low-light image enhancement in night time teleoperation, utilizing the vision language model empowered contract theory for automated difficulty assessment and differential pricing under information asymmetry.

In order to further improve the efficiency of resource utilization, it is essential to design multi-dimensional resource allocation strategies, such as level of prompt optimization, number of diffusion denoising steps, CPU cycle frequency, and network transmission rate, yet existing research has not explored this. Thus, we merge a diffusion model and contract theory to propose a two-tiered, multi-dimensional resource allocation framework with a focus on the users' QoE, including result quality and latency efficiency. A comparison of our work with what already exists is shown in Table II.

III. System Model

To illustrate the workflow of the GDM-based contract theory framework, we initially focus on a text-generated image service, subsequently introducing the concept of Quality of Experience (QoE). This workflow is also adaptable to various other AIGC services.

A. Workflow of Framework

We consider an ASP and M users. The workflow of the GDM-based contract theory framework can be accomplished by taking the following steps.

Step 1. Training of Quality-based and Latency-based Contract Generation Models: The ASP uses history records to train a quality-based contract generation model and a latency-based contract generation model based on the GDM, respectively. More details are given in Sections IV and V.

Step 2. Quality-based Contract Generation and Selection: Taking the environmental parameters as input, the ASP uses the quality-based contract generation model to generate quality-based contract items denoted as Φ^{A}. The input environmental parameters for the quality-based contract generation model are denoted as a vector $\boldsymbol{e}^{\mathrm{A}}$ and include the number of types of gain per quality of image generation, the type vector of gain per quality, the probability vector, the cost vector per level of prompt optimization, and the cost vector per number of diffusion denoising steps. Those parameters are denoted as I, $\boldsymbol{\theta}^{\mathrm{A}} = [\theta_1^{\mathrm{A}}, \ldots, \theta_i^{\mathrm{A}}, \ldots, \theta_I^{\mathrm{A}}]$, $\boldsymbol{q}^{\mathrm{A}} = [q_1^{\mathrm{A}}, \ldots, q_i^{\mathrm{A}}, \ldots, q_I^{\mathrm{A}}]$, $\boldsymbol{\sigma}_1 = [\sigma_{1,1}, \ldots, \sigma_{1,i}, \ldots, \sigma_{1,I}]$, and $\boldsymbol{\sigma}_2 = [\sigma_{2,1}, \ldots, \sigma_{2,i}, \ldots, \sigma_{2,I}]$. Here, θ_i^{A} is the type of the i-th gain per quality, q_i^{A} is the probability that a user's type belongs to the type of the i-th gain per quality, $\sigma_{1,i}$ is the cost per level of prompt optimization for the type of the i-th gain per quality, and $\sigma_{2,i}$ is the cost per number of diffusion denoising steps for the type of the i-th gain per quality. Each quality-based contract item is denoted as $\Phi_i^{\mathrm{A}} = (l_i^{\mathrm{A}}, s_i^{\mathrm{A}}, p_i^{\mathrm{A}}), i \in \{1, \ldots, I\}$ where l_i^{A} is the level of prompt optimization, s_i^{A} is the number of diffusion denoising steps, and p_i^{A} is the reward paid to the ASP. Then, the users choose the quality-based contract item that suits their gain types per quality.

Step 3. Latency-based Contract Generation and Selection: The ASP counts the number of the users that

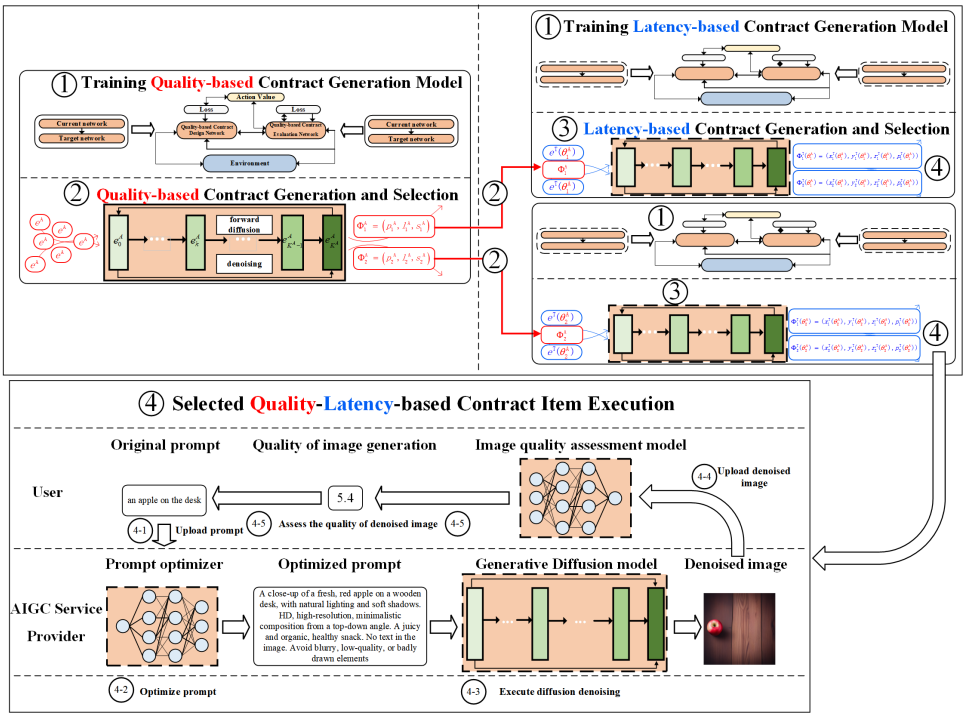


Fig. 1: Workflow of GDM-based contract theory framework. Step 1. The ASP trains two GDM-based models for quality-based and latency-based contract generation; Step 2. Based on the input environmental parameters $\boldsymbol{e}^{\mathrm{A}}$, the quality-based contract generation model generates quality-based contract items i.e., $\Phi^{\mathrm{A}} = \left\{\Phi_i^{\mathrm{A}}, i = \{1, \ldots, I\}\right\}$; Step 3. The number of type-θ_i^{A} users who select the same quality-based contract item is counted, each quality-based contract item Φ_i^{A}, and the environmental parameters $\boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ are then used as inputs in a latency-based contract generation model to generate corresponding latency-focused contract items i.e., $\Phi^{\mathrm{T}}(\theta_i^{\mathrm{A}}) = \left\{\Phi_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), j = \{1, \ldots, J\}\right\}$; Step 4. Execute the selected quality-latency-based contract.

have chosen the same quality-based contract item Φ_i^{A}, which is denoted as M_i. The latency-based contract generation model takes as input each quality-based contract item Φ_i^{A} and the number M_i, along with the input environmental parameters denoted as a vector $\boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}})$, and produces the latency-based contract items denoted as $\Phi^{\mathrm{T}}(\theta_i^{\mathrm{A}})$. The vector $\boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ includes the number of types of gain per expected latency reduction, the probability vector, the maximum latency vector, the cost vector per computation energy consumption of optimizing the prompt, the cost vector per computation energy consumption of executing diffusion denoising, the cost vector per communication energy consumption, the type vector of gain per expected latency reduction, the CPU frequency vector per level of prompt optimization, the CPU frequency vector per number of diffusion denoising steps, the size vector of the diffusion denoising result, and the status vector of wireless connection, the effective switched capacitance. Those parameters are denoted as J, $\boldsymbol{q}^{\mathrm{T}}(\theta_i^{\mathrm{A}}) = [q_1^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \ldots, q_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \ldots, q_J^{\mathrm{T}}(\theta_i^{\mathrm{A}})]$, $\boldsymbol{t}^{\max} = [t_1^{\max}, \ldots, t_j^{\max}, \ldots, t_J^{\max}]$, $\boldsymbol{b}_1 = [b_{1,1}, \ldots, b_{1,j}, \ldots, b_{1,J}]$, $\boldsymbol{b}_2 = [b_{2,1}, \ldots, b_{2,j}, \ldots, b_{2,J}]$, $\boldsymbol{b}_3 = [b_{3,1}, \ldots, b_{3,j}, \ldots, b_{3,J}]$, $\boldsymbol{\theta}^{\mathrm{T}}(\theta_i^{\mathrm{A}}) = [\theta_1^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \ldots, \theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \ldots, \theta_J^{\mathrm{T}}(\theta_i^{\mathrm{A}})]$, $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_j, \ldots, \delta_J]$, $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_j, \ldots, \eta_J]$, $\boldsymbol{d} = [d_1, \ldots, d_j, \ldots, d_J]$, $\boldsymbol{h} = [h_1, \ldots, h_j, \ldots, h_J]$, and $\boldsymbol{\kappa} = [\kappa_1, \ldots, \kappa_j, \ldots, \kappa_J]$. A more detailed explanation for those parameters refers to Table I. Each latency-based contract item is denoted as $\Phi_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}) = (x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), p_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})), j \in \{1, \ldots, J\}$ where $x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ is the CPU frequency for optimizing prompt, $y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ is the CPU frequency for diffusion denoising, $r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ is the network transmission rate, and $p_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}))$ is the reward paid to the ASP. Then, the users choose the latency-based contract item that suits their types of gain per expected latency reduction.

Step 4. Selected Quality-Latency-based Contract Execution: For each selected contract items $\Phi_i^{\mathrm{A}}, i \in \{1, \ldots, I\}$ and $\Phi_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), i \in \{1, \ldots, I\}, j \in \{1, \ldots, J\}$, the contract execution includes four stages. Step 4-1: Each user uploads its original prompt to the ASP. Step 4-2: The ASP is capable of optimizing the original prompt of image generation, with the level of prompt optimization l_i^{A} and the CPU frequency $x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$. Step4-3: Based on the optimized prompt, the ASP performs the diffusion denoising steps according to the number of diffusion denoising steps s_i^{A} and the CPU frequency $y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$. Step4-4: After the diffusion denoising steps have been completed, the denoised images are sent to the users with the network transmission rate $r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$. Step4-5: The users use several metrics to assess the quality of the generated result, such as neural image assessment [8]. If the quality of image generation meets the user's requirement, the image generation service is considered to be successful and the user will send the rewards p_i^{A} and $p_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ to the ASP.

B. Quality of Experience

The QoE has two components: the quality of image generation and the latency reduction of image generation denoted as A and D.

1) Quality of Image Generation: As the number of diffusion denoising steps increases, the quality of image generation improves [15], [21]. As the level of prompt optimization increases, the quality of image generation increases [4]. We have also verified the above result through the results of our experiments in Section VI. The relationship between the level of prompt optimization, the number of diffusion denoising steps, and the quality of image generation is defined as follows:

$$A = A(l^{\mathrm{A}}, s^{\mathrm{A}}, \boldsymbol{\rho}) \tag{1}$$

where $\boldsymbol{\rho}$ is the parameter vector fitted by experiments, $\boldsymbol{\rho} \geq \mathbf{0}$, l^{A} and s^{A} are positive integers.

2) Latency Reduction of Image generation: For each user, when l^{A} and s^{A} are both fixed, the total latency of obtaining the generated result includes three parts. The first part is the latency of optimizing the prompt. Motivated by [15], [21], [22], the latency for optimizing the prompt is defined as $\frac{\delta l^{\mathrm{A}}}{x^{\mathrm{T}}}$, where δ is the CPU frequency per level of prompt optimization, x^{T} is the CPU frequency for optimizing the prompt. Referring to [15], [21], [22], the second part is the latency of diffusion denoising denoted as $\frac{\eta s^{\mathrm{A}}}{y^{\mathrm{T}}}$, where η is the CPU frequency per number of diffusion denoisin steps, and y^{T} is the CPU frequency for diffusion denoising. Furthermore, the third part is the transmission latency denoted as $\frac{d}{r^{\mathrm{T}}}$, where d is the size of the diffusion denoising result and r^{T} is the network transmission rate [23]. Thus, the total latency is $\frac{\delta l^{\mathrm{A}}}{x^{\mathrm{T}}} + \frac{\eta s^{\mathrm{A}}}{y^{\mathrm{T}}} + \frac{d}{r^{\mathrm{T}}}$. Based on the total latency, we obtain the total latency reduction as follows:

$$D = t^{\max} - \frac{\delta l^{\mathrm{A}}}{x^{\mathrm{T}}} - \frac{\eta s^{\mathrm{A}}}{y^{\mathrm{T}}} - \frac{d}{r^{\mathrm{T}}}, \tag{2}$$

where $t^{\max}$ is the maximum latency.

The experimental results in [4] showed that the probability of achieving a specific quality threshold $\overline{A}$ in image generation increases with higher levels of prompt optimization and an increased number of diffusion denoising steps. We define the probability as $\zeta\left(A\left(l^{\mathrm{A}}, s^{\mathrm{A}}\right) > \overline{A}\right)$ and related to l, s, and A. To simplify the representation, the notation of the probability $\zeta\left(A\left(l^{\mathrm{A}}, s^{\mathrm{A}}\right) > \overline{A}\right)$ is reduced to $\zeta_{l^{\mathrm{A}}, s^{\mathrm{A}}}$. If the generated image fails to satisfy the user's quality requirements, it requires regeneration. This cycle continues until the desired quality is attained, at which point the service ends. The aforementioned process can be modeled mathematically to ascertain the expected latency reduction for the g-th iteration of image generation to meet the user's standards, expressed as:

$$\mathbb{E}[D] = \zeta_{l^{\mathrm{A}}, s^{\mathrm{A}}}(1 - \zeta_{l^{\mathrm{A}}, s^{\mathrm{A}}})^{g-1}\left[t^{\max} - g\left(\frac{\delta l^{\mathrm{A}}}{x^{\mathrm{T}}} + \frac{\eta s^{\mathrm{A}}}{y^{\mathrm{T}}} + \frac{d}{r^{\mathrm{T}}}\right)\right]. \tag{3}$$

It should be noted that we consider $g = 1$ in the paper. In future work, we will explore $g > 1$.

IV. Generative Diffusion Model for Quality-based Contract Design

Based on the quality of image generation, the utilities of the users and ASP are modeled. Then, a quality-based contract problem is formulated. Continuously, a GDM-based scheme is used to solve optimal quality-based contract items in a more efficient way. Finally, we analyze the complexity of the GDM-based scheme.

A. Utilities of User and AIGC Service Provider

The higher the quality of image generation, the higher the gain for the user. Referring to [24], the gain of user m is $\theta_m^{\mathrm{A}} A(l_m, s_m, \boldsymbol{\rho})$ where θ_m is the gain per quality of image generation. The user m must pay a reward p_m^{A} to the ASP. Thus, the utility of the user m is

$$u_m^{\mathrm{A}} = \theta_m^{\mathrm{A}} A(l_m^{\mathrm{A}}, s_m^{\mathrm{A}}, \boldsymbol{\rho}) - p_m^{\mathrm{A}}. \tag{4}$$

However, due to self-interest, user m is reluctant to disclose information about its gain per quality to the ASP. Without knowledge of the user m's gain per quality, it becomes challenging for the ASP to determine the optimal level of prompt optimization and the number of diffusion denoising steps needed to maximize its own payoffs while also setting an appropriate fee for user m. In such cases, many studies, such as the authors in [25] and [26] , assume that the ASP possesses knowledge of the probability distribution of gain per quality types based on statistical data while knowing the total number of users across all types,i.e., M. Additionally, the probability that a user's gain type per quality is of type θ_i^{A} is represented by q_i^{A}. To determine the amount of a particular type, we apply the method discussed in [27] and then multiply its probability by the total number of users across all types. Consequently, we have the quantity of users whose gain type per quality falls into type-θ_i^{A} is $Mq_i^{\mathrm{A}} = M_i$. Based on statistical information from the mobile data market, the ASP can classify the users into different types to characterize their heterogeneity, using some well-known data mining methods, e.g., k-means. According to their heterogeneity for a given gain per quality, we classify the users into I types and sorted in ascending order $\theta_1^{\mathrm{A}} \leq \cdots \leq \theta_i^{\mathrm{A}} \leq \cdots \leq \theta_I^{\mathrm{A}}$. Specifically, the user m whose gain per quality falls into i-th gain per quality is denoted as type-θ_i^{A} user. Thus, the utilities of these users belonging to type-θ_i^{A} can be defined as

$$u_i^{\mathrm{A}} = \theta_i^{\mathrm{A}} A(l_i^{\mathrm{A}}, s_i^{\mathrm{A}}, \boldsymbol{\rho}) - p_i^{\mathrm{A}}. \tag{5}$$

The cost required for the ASP to provide service to a type-θ_i^{A} user is defined as $\sigma_{1,i} l_i^{\mathrm{A}} + \sigma_{2,i} s_i^{\mathrm{A}}$, where $\sigma_{1,i}$ is the cost per level of prompt optimization and $\sigma_{2,i}$ is the cost per number of diffusion denoising steps. For all the types, the utility of the ASP is defined as

$$U_{\mathrm{sp}}^{\mathrm{A}} = \sum_{i=1}^{I} Mq_i^{\mathrm{A}}(p_i^{\mathrm{A}} - \sigma_{1,i} l_i^{\mathrm{A}} - \sigma_{2,i} s_i^{\mathrm{A}}). \tag{6}$$

B. Quality-based Contract Formulation

The lack of awareness of the ASP regarding the users' specific gain per quality, which pertains to their privacy, leads to an imbalance of information. This information asymmetry can be addressed by applying contract theory to determine the most suitable contract items for the ASP's consumers. In this context, the ASP acts as the main entity responsible for designing quality-based contracts, while the users are considered agents who select the contract item that aligns with their respective type. The quality-based contract item can be denoted as $\Phi^{\mathrm{A}} = \left\{\Phi_i^{\mathrm{A}} = (l_i^{\mathrm{A}}, s_i^{\mathrm{A}}, p_i^{\mathrm{A}}), i = \{1, \ldots, I\}\right\}$, where Φ_i^{A} is made for a type-θ_i^{A} user. In order to establish a feasible quality-based contract with asymmetric information, we introduce the following conditions for Individual Rationality (IR) and Incentive Compatibility (IC). The IR condition encourages user engagement and guarantees a non-negative utility. The mathematical expression for the IR conditions, applicable to a type-θ_i^{A} user, can be represented as follows:

$$\theta_i^{\mathrm{A}} A(l_i^{\mathrm{A}}, s_i^{\mathrm{A}}, \boldsymbol{\rho}) - p_i^{\mathrm{A}} \geq 0, i \in \{1, \ldots, I\}. \tag{7}$$

The IC conditions ensure that each type-θ_i^{A} user can achieve its maximum utility when selecting the quality-based contract item based on its own corresponding type. The mathematical expression for the IC conditions, applicable to a type-θ_i^{A} user, can be represented as follows:

$$\begin{aligned} \theta_i^{\mathrm{A}} A(l_i^{\mathrm{A}}, s_i^{\mathrm{A}}, \boldsymbol{\rho}) - p_i^{\mathrm{A}} \geq & \theta_i^{\mathrm{A}} A(l_{i'}^{\mathrm{A}}, s_{i'}^{\mathrm{A}}, \boldsymbol{\rho}) - p_{i'}^{\mathrm{A}}, \\ & \forall i, i^{'} \in \{1, \ldots, I\}. \end{aligned} \tag{8}$$

To maximize the utility of the ASP under the IR and IC conditions, a quality-based contract problem is formulated as follows:

$$\begin{aligned} \textbf{Problem 1:} \quad & \max_{l_i^{\mathrm{A}}, s_i^{\mathrm{A}}, p_i^{\mathrm{A}}} U_{\mathrm{sp}}^{\mathrm{A}} \\ \text{s.t.} \quad & (7), \text{and} \quad (8), i, i^{'} \in \{1, \ldots, I\}, \\ & l_i^{A,\min} \leq l_i^{\mathrm{A}} \leq l_i^{A,\max}, l_i \in \mathbb{Z}^+, i \in \{1, \ldots, I\}, \\ & s_i^{A,\min} \leq s_i^{\mathrm{A}} \leq s_i^{A,\max}, s_i \in \mathbb{Z}^+, i \in \{1, \ldots, I\}, \\ & p_i^{A,\min} \leq p_i^{\mathrm{A}} \leq p_i^{A,\max}, i \in \{1, \ldots, I\}, \end{aligned} \tag{9}$$

where $l_i^{A,\min}$, $s_i^{A,\min}$ and $p_i^{A,\min}$ are the minimum value of the optimization variables, $l_i^{A,\max}$, $s_i^{A,\max}$ and $p_i^{A,\max}$ are the maximum value of the optimization variables. In **Problem 1**, since the objective function is non-convex and the constraints are non-convex sets, it is difficult to solve **Problem 1**. The ASP's variable cost expenses and the users' variable gain per quality require solving the non-convex quality-based contract problem repeatedly, which takes longer delay using traditional mathematical methods. Fortunately, a GDM-based scheme is capable of handling this issue [1].

C. GDM-based Scheme for Quality-based Contract Problem

1) Generative Diffusion Model: GDM, a pioneering deep-generative model, operates by progressively modifying the data distribution in its forward diffusion phase through the incremental addition of Gaussian noise. In this phase, Gaussian noise is systematically added to an initial sample, denoted as $\boldsymbol{\phi}_0$, over K iterations, resulting in a sequence of samples $(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_K)$. As the iteration count K increases, the distinct characteristics of the original sample $\boldsymbol{\phi}_0$ are gradually obliterated, ultimately transforming into pure Gaussian noise. This process can be succinctly described as follows:

$$Q\left(\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K \mid \boldsymbol{\phi}_0\right) = \prod_{k=1}^{K} Q\left(\boldsymbol{\phi}_k \mid \boldsymbol{\phi}_{k-1}\right), \tag{10}$$

$$Q\left(\boldsymbol{\phi}_k \mid \boldsymbol{\phi}_{k-1}\right) := \mathcal{N}\left(\boldsymbol{\phi}_k; \sqrt{1-\beta_k}\boldsymbol{\phi}_{k-1}, \beta_k \mathbf{I}\right), \tag{11}$$

where β_k is a parameter that controls the influence of noise on the progress. Equation (11) suggests that, when provided with the sample $\boldsymbol{\phi}_{k-1}$, the sample $\boldsymbol{\phi}k$ at the k-th step follows a Gaussian distribution with a mean of $\sqrt{1-\beta_k}\boldsymbol{\phi}_{k-1}$ and a variance of $\beta_k \mathbf{I}$. The dependence of these parameters solely on the previous sample $\boldsymbol{\phi}_{k-1}$ indicates that the diffusion process qualifies as a Markov process.

In the reverse diffusion process $Q\left(\boldsymbol{\phi}_{k-1} \mid \boldsymbol{\phi}k, \boldsymbol{\phi}_0\right)$, when β_k is sufficiently small, it aligns with the forward diffusion process's posterior probability distribution $Q\left(\boldsymbol{\phi}_k \mid \boldsymbol{\phi}_{k-1}\right)$. For the generation of authentic samples, the model $P\omega\left(\boldsymbol{\phi}_{0:K}\right)$ must iteratively sample from Gaussian noise $\boldsymbol{\phi}_K$ and learn the precise parameters ω based on training data. This procedure can be depicted as follows:

$$P_\omega\left(\boldsymbol{\phi}_{0:K}\right) = P\left(\boldsymbol{\phi}_K\right) \prod_{k=1}^{K} P_\omega\left(\boldsymbol{\phi}_{k-1} \mid \boldsymbol{\phi}_k\right), \tag{12}$$

$$P_\omega\left(\boldsymbol{\phi}_{k-1} \mid \boldsymbol{\phi}_k\right) = \mathcal{N}\left(\boldsymbol{\phi}_{k-1}; \mu_\omega\left(\boldsymbol{\phi}_k, k\right), \Sigma_\omega\left(\boldsymbol{\phi}_k, k\right)\right), \tag{13}$$

where $P\left(\boldsymbol{\phi}_K\right) = \mathcal{N}\left(\boldsymbol{\phi}_K; 0, \mathbf{I}\right)$. Ultimately, the process of reverse diffusion can be accomplished by employing a highly trained $P_\theta\left(\boldsymbol{\phi}_{k-1} \mid \boldsymbol{\phi}_k\right)$ to estimate $Q\left(\boldsymbol{\phi}_{k-1} \mid \boldsymbol{\phi}_k, \boldsymbol{\phi}_0\right)$.

2) Training Phase: We first define the environment, quality-based contract design networks, and quality-based contract evaluation networks. The environment is defined by a vector $\boldsymbol{e}^{\mathrm{A}}$, which is the set of all variables that impact the optimal design of a quality-based contract, i.e.,

$$\boldsymbol{e}^{\mathrm{A}} = \left\{\boldsymbol{q}^{\mathrm{A}}, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \boldsymbol{\theta}^{\mathrm{A}}, M, I\right\}. \tag{14}$$

The diffusion model network known as the quality-based contract design policy, symbolized by $\pi_{\omega^{\mathrm{A}}}^{\mathrm{A}}(\boldsymbol{\phi}^{\mathrm{A}}|\boldsymbol{e}^{\mathrm{A}})$, assigns environment states to quality-based contract designs using the weights ω^{A}. Its primary objective, through the policy $\pi_{\omega^{\mathrm{A}}}^{\mathrm{A}}(\boldsymbol{\phi}^{\mathrm{A}}|\boldsymbol{e}^{\mathrm{A}})$, is to generate a deterministic quality-based contract design aimed at optimizing the expected total reward over a series of time steps. This policy $\pi_{\omega^{\mathrm{A}}}^{\mathrm{A}}(\boldsymbol{\phi}^{\mathrm{A}}|\boldsymbol{e}^{\mathrm{A}})$, utilizes the reverse mechanism of a conditional diffusion model, as shown below:

$$\begin{aligned} \pi_{\omega^{\mathrm{A}}}^{\mathrm{A}}(\boldsymbol{\phi}^{\mathrm{A}} \mid \boldsymbol{e}^{\mathrm{A}}) &= P_{\omega^{\mathrm{A}}}^{\mathrm{A}}\left(\boldsymbol{\phi}^{0:K^{\mathrm{A}}} \mid \boldsymbol{e}^{\mathrm{A}}\right) \\ &= \mathcal{N}^{\mathrm{A}}\left(\boldsymbol{\phi}^{K^{\mathrm{A}}}; \mathbf{0}, \mathbf{I}^{\mathrm{A}}\right) \prod_{k=1}^{K^{\mathrm{A}}} P_{\omega^{\mathrm{A}}}^{\mathrm{A}}\left(\boldsymbol{\phi}^{k-1,\mathrm{A}} \mid \boldsymbol{\phi}^{k,\mathrm{A}}, \boldsymbol{e}^{\mathrm{A}}\right), \end{aligned} \tag{15}$$

where $P_{\omega^{\mathrm{A}}}^{\mathrm{A}}(\boldsymbol{\phi}^{k-1,\mathrm{A}}|\boldsymbol{\phi}^{k,\mathrm{A}}, \boldsymbol{e}^{\mathrm{A}})$ can be modeled as a Gaussian distribution

$\mathcal{N}^{\mathrm{A}}\left(\boldsymbol{\phi}^{k-1,\mathrm{A}}; \mu^{\mathrm{A}}_{\omega^{\mathrm{A}}}\left(\boldsymbol{\phi}^{k,\mathrm{A}}, \mathbf{e}^{\mathrm{A}}, k\right), \boldsymbol{\Sigma}_{\omega^{\mathrm{A}}}\left(\boldsymbol{\phi}^{k,\mathrm{A}}, \mathbf{e}^{\mathrm{A}}, k\right)\right)$.
According to [28], $P^{\mathrm{A}}_{\omega^{\mathrm{A}}}(\boldsymbol{\phi}^{k-1,\mathrm{A}}|\boldsymbol{\phi}^{k,\mathrm{A}}, \boldsymbol{e}^{\mathrm{A}})$ can be modeled as a noise prediction model, with the covariance matrix fixed as follows:

$$\boldsymbol{\Sigma}_{\omega^{\mathrm{A}}}\left(\boldsymbol{\phi}^{k,\mathrm{A}}, \boldsymbol{e}^{\mathrm{A}}, k\right) = \beta^{\mathrm{A}}_k \mathbf{I}^{\mathrm{A}}, \tag{16}$$

and the mean constructed as:

$$\begin{aligned}&\mu^{\mathrm{A}}_{\omega^{\mathrm{A}}}\left(\boldsymbol{\phi}^{k,\mathrm{A}}, \mathbf{e}^{\mathrm{A}}, k\right)\\ &= \frac{1}{\sqrt{\alpha^{\mathrm{A}}_k}}\left(\boldsymbol{\phi}^{k,\mathrm{A}} - \frac{\beta^{\mathrm{A}}_k}{\sqrt{1-\bar{\alpha}^{\mathrm{A}}_k}}\varepsilon^{\mathrm{A}}_{\omega}\left(\boldsymbol{\phi}^{k,\mathrm{A}}, \mathbf{e}^{\mathrm{A}}, k\right)\right).\end{aligned} \tag{17}$$

We commence by sampling $\boldsymbol{\phi}^{K^{\mathrm{A}}} \sim \mathcal{N}^{\mathrm{A}}(\mathbf{0}, \mathbf{I}^{\mathrm{A}})$ and then proceed with the reverse diffusion chain, parameterized by ω^{A} as

$$\begin{aligned}&\boldsymbol{\phi}^{k-1,\mathrm{A}} \mid \boldsymbol{\phi}^{k,\mathrm{A}}\\ &= \frac{\boldsymbol{\phi}^{k,\mathrm{A}}}{\sqrt{\alpha^{\mathrm{A}}_k}} - \frac{\beta^{\mathrm{A}}_k}{\sqrt{\alpha^{\mathrm{A}}_k\left(1-\bar{\alpha}^{\mathrm{A}}_k\right)}}\varepsilon^{\mathrm{A}}_{\omega}\left(\boldsymbol{\phi}^{k,\mathrm{A}}, \mathbf{e}^{\mathrm{A}}, k\right) + \sqrt{\beta^{\mathrm{A}}_k}\varepsilon^{\mathrm{A}}.\end{aligned} \tag{18}$$

Effective training of the quality-based contract design policy $\pi^{\mathrm{A}}_{\omega^{\mathrm{A}}}$ within the vector $\boldsymbol{e}^{\mathrm{A}}$ involves the development of a quality-based contract design network $\varepsilon^{\mathrm{A}}_{\omega^{\mathrm{A}}}$. Following DDPM's guidelines [28], we set ε^{A} to 0 when $k = 1$ to improve sample quality. For the training of the quality-based contract design network $\varepsilon^{\mathrm{A}}_{\omega^{\mathrm{A}}}$, the Q-function from deep reinforcement learning (DRL) serves as inspiration, leading to the establishment of the quality-based contract evaluation network H^{A}_v. This network associates an environment-contract pair, $\left\{\boldsymbol{e}^{\mathrm{A}}, \Phi^{\mathrm{A}}\right\}$, with a value indicative of the anticipated cumulative reward for adhering to a quality-based contract design policy from the current state. By minimizing the loss function $\mathcal{L}^{\mathrm{A}}(\omega^{\mathrm{A}})$ through double Q-learning, the most effective quality-based contract design policy can be determined. The loss function is defined as follows:

$$\pi^{\mathrm{A}} = \arg\min_{\pi^{\mathrm{A}}_{\omega^{\mathrm{A}}}} \mathcal{L}^{\mathrm{A}}(\omega^{\mathrm{A}}) = -\mathbb{E}_{\boldsymbol{\phi}^{0,\mathrm{A}}\sim\pi^{\mathrm{A}}_{\omega^{\mathrm{A}}}}\left[H^{\mathrm{A}}_{\upsilon}\left(\mathbf{e}^{\mathrm{A}}, \boldsymbol{\phi}^{0,\mathrm{A}}\right)\right]. \tag{19}$$

The network of evaluating quality-based contracts employs the double Q-learning method for its training [29]. It involves the formulation of two primary networks, designated as $H^{\mathrm{A}}_{\upsilon^{\mathrm{A}}_1}$ and $H^{\mathrm{A}}_{\upsilon^{\mathrm{A}}_2}$, and their corresponding target counterparts, named $H^{\mathrm{A},\prime}_{\upsilon^{\mathrm{A}}_1}$ and $H^{\mathrm{A},\prime}_{\upsilon^{\mathrm{A}}_2}$, along with $\pi^{\mathrm{A},\prime}_{\omega^{\mathrm{A}}}$. The goal is to optimize υ^{A}_n for $n = 1, 2$ through minimization of the objective

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\phi}^{0,A}_{k+1}\sim\pi^{A}_{\omega^{A,\prime}}}\Bigg[\Bigg|\Bigg|\left(r(\mathbf{e}^{A}, \boldsymbol{\phi}^{A}_k) + \gamma^{A}\min_{n=1,2} H^{A}_{\upsilon^{A,\prime}_n}(\mathbf{e}^{A}, \boldsymbol{\phi}^{0,A}_{k+1})\right)\\ - H^{A}_{\upsilon^{A}_n}(\mathbf{e}^{A}, \boldsymbol{\phi}^{A}_k)\Bigg|\Bigg|^2\Bigg].\end{aligned} \tag{20}$$

3) Inference Stage: During the inference stage, the trained quality-based contract design network is used to generate efficient quality-based contract items based on current environmental circumstances. The quality-based contract items generated maximize the utility of the ASP while satisfying the IC and IR constraints of the users.

The detailed algorithm for the GDM-based optimal quality-based contract is shown in **Algorithm 1**. In the analysis of the complexity of Algorithm 1, the weights in the quality-based contract design and evaluation networks are denoted ψ^{A}_a and ψ^{A}_c, respectively. The initialization complexity stands at $\mathcal{O}\left(2\psi^{\mathrm{A}}_a + 2\psi^{\mathrm{A}}_c\right)$. The complexity of action generation increases to $\mathcal{O}\left(K^{\mathrm{A}}\psi^{\mathrm{A}}_a\right)$. Replay buffer activities maintain a storage complexity of $\mathcal{O}(1)$ and minibatch sampling complexity of $\mathcal{O}(N^{\mathrm{A}})$. Each update to quality-based contract design and evaluation networks incurs complexities $\mathcal{O}\left(\psi^{\mathrm{A}}_c\right)$ and $\mathcal{O}\left(\psi^{\mathrm{A}}_a\right)$, respectively. Updates to the target network have linear complexity in relation to parameter numbers. Consequently, the computational complexity in the training phase is adjusted to $\mathcal{O}\left(Z^{\mathrm{A}}_e Z^{\mathrm{A}}_s\left(K^{\mathrm{A}}\psi^{\mathrm{A}}_a + \psi^{\mathrm{A}}_c\right)\right)$. In the inference phase, to generate optimal quality-based contract items via the trained network, the complexity is $\mathcal{O}\left(\psi^{\mathrm{A}}_a\right)$, assuming that reward observation and exploration noise generation are constant-time operations. Thus, combining the training phase complexity and the inference phase complexity, the algorithm's total complexity is $\mathcal{O}\left(Z^{\mathrm{A}}_e Z^{\mathrm{A}}_s\left(K^{\mathrm{A}}\psi^{\mathrm{A}}_a + \psi^{\mathrm{A}}_c\right)\right)$.

V. Generative Diffusion Model for Latency-based Contract Design

After M users select the quality-based contract items, the ASP counts the number of the users that have chosen the same quaility-based contract item $(l^{\mathrm{A}}_i, s^{\mathrm{A}}_i, p^{\mathrm{A}}_i), i \in \{1, \ldots, I\}$. The number of the users is denoted as $M_i, i \in \{1, \ldots, I\}$ and we obtain $\sum_{i=1}^{I} M_i = M$. Here, the user choosing the quaility-based contract item $(l^{\mathrm{A}}_i, s^{\mathrm{A}}_i, p^{\mathrm{A}}_i)$ is $m_i \in \{1, \ldots, N_i\}$. For each $(l^{\mathrm{A}}_i, s^{\mathrm{A}}_i, p^{\mathrm{A}}_i)$ and N_i, $i \in \{1, \ldots, I\}$, based on the expected latency reduction of image generation, the utilities of the users and ASP are modeled. Then, a latency-based contract problem is formulated. Finally, a GDM-based scheme is also used to solve the optimal latency-based contract items.

A. Utilities of User and AIGC Service Provider

The higher the expected latency reduction of image generation, the higher the gain of the user n_i. Referring to [30], the gain of the user m_i is $\theta^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i)\mathbb{E}[D](x^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i), y^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i), r^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i))$ where $\theta^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i)$ is the type of gain per expected latency reduction with the type of gain per quality θ^{A}_i. The user m_i must pay a reward $p^{\mathrm{T}}_n(\theta^{\mathrm{A}}_i)$ to the ASP. Thus, the utility of user m_i is

$$u^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i) = \theta^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i)\mathbb{E}[D](x^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i), y^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i), r^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i)) - p^{\mathrm{T}}_m(\theta^{\mathrm{A}}_i). \tag{21}$$

However, M_i self-interest users may not provide information about their types of gain per expected latency reduction to the ASP. According to historical records, M_i users with different types of gain per expected latency reduction are classified into J types and sorted in ascending order $\theta^{\mathrm{T}}_1(\theta^{\mathrm{A}}_i) \leq \cdots \leq \theta^{\mathrm{T}}_j(\theta^{\mathrm{A}}_i) \leq \cdots \leq \theta^{\mathrm{T}}_J(\theta^{\mathrm{A}}_i)$. A user with a gain per quality of type-θ^{A}_i and gain per expected latency reduction of type-θ^{T}_j is referred to as type-$\theta^{\mathrm{T}}_j(\theta^{\mathrm{A}}_i)$ user for the sake of simplicity. Thus, the utilities of the users belonging to type-$\theta^{\mathrm{T}}_j(\theta^{\mathrm{A}}_i)$ can be defined as

$$u^{\mathrm{T}}_j(\theta^{\mathrm{A}}_i) = \theta^{\mathrm{T}}_j(\theta^{\mathrm{A}}_i)\mathbb{E}[D](x^{\mathrm{T}}_j(\theta^{\mathrm{A}}_i), y^{\mathrm{T}}_j(\theta^{\mathrm{A}}_i), r^{\mathrm{T}}_j(\theta^{\mathrm{A}}_i)) - p^{\mathrm{T}}_j(\theta^{\mathrm{A}}_i). \tag{22}$$

Algorithm 1: Algorithm for GDM-based Optimal Quality-based Contract

Training Phase:
Input hyper-parameters: number of iterations to add noise K^{A}, batch size N^{A}, discount factor γ^{A}, soft target update parameter τ^{A}, exploration noise ϵ^{A}.
Initialize replay buffer $\mathcal{B}^{\mathrm{A}}$, quality-based contract design network $\varepsilon_{\omega}^{\mathrm{A}}$ with weights ω^{A}, quality-based contract evaluation network $H_{\upsilon^{\mathrm{A}}}^{\mathrm{A}}$ with weights υ^{A}, target quality-based contract design network $\varepsilon_{\omega^{\mathrm{A},'}}^{\mathrm{A}}$ with weights $\omega^{\mathrm{A},'}$, target quality-based contract evaluation network $H_{\upsilon^{\mathrm{A},'}}^{\mathrm{A}}$ with weights $\upsilon^{\mathrm{A},'}$.
for Episode = 1 to Max episode Z_e^{A} **do**
  Initialize a random process $\mathcal{N}^{\mathrm{A}}$ for quality-based contract design exploration
  for Step = 1 to Max step Z_s^{A} **do**
    Observe the current environment $\boldsymbol{e}_k^{\mathrm{A}}$
    Set $\boldsymbol{\phi}_k^{K^{\mathrm{A}}}$ as Gaussian noise. Generate a quality-based contract design $\boldsymbol{\phi}_k^{0,\mathrm{A}}$ by denoising $\boldsymbol{\phi}_k^{K^{\mathrm{A}}}$ using $\epsilon_{\omega^{\mathrm{A}}}^{\mathrm{A}}$ according to (32)
    Add the exploration noise ϵ^{A} to $\boldsymbol{\phi}_k^{0,\mathrm{A}}$
    Execute quality-based contract design $\boldsymbol{\phi}_k^{0,\mathrm{A}}$ and observe the reward defined as

$$\begin{aligned}\lambda_k^{\mathrm{A}} = U_{\mathbf{sp},k}^{\mathrm{A}} &+ \sum_{i=1}^{I} \mathcal{P}^{\mathrm{A}}\left[\zeta_i^{\mathrm{A}}\theta_{i,k}^{\mathrm{A}} A(l_{i,k}^{\mathrm{A}}, s_{i,k}^{\mathrm{A}}, \boldsymbol{\rho}) - r_{i,k}^{\mathrm{A}}\right] \\ &+ \sum_{i=1}^{I}\sum_{i'=1,i'\neq i}^{I} \mathcal{P}^{\mathrm{A}}\left[\zeta_i^{\mathrm{A}}\theta_{i,k}^{\mathrm{A}} A(l_{i,k}^{\mathrm{A}}, s_{i,k}^{\mathrm{A}}, \boldsymbol{\rho}) - r_{i,k}^{\mathrm{A}}\right. \\ &\left. -\zeta_i^{\mathrm{A}}\theta_{i,k}^{\mathrm{A}} A(l_{i',k}^{\mathrm{A}}, s_{i',k}^{\mathrm{A}}, \boldsymbol{\rho}) + r_{i',k}^{\mathrm{A}}\right],\end{aligned}$$

    where $\mathcal{P}^{\mathrm{A}}(\cdot)$ is a penalty function. It implements a certain penalty when the IC and IR constraints are not satisfied. The penalty is denoted as ξ^{A}.
    Store the record $(\boldsymbol{e}_k^{\mathrm{A}}, \boldsymbol{\phi}_k^{0,\mathrm{A}}, \lambda_k^{\mathrm{A}})$ in replay buffer $\mathcal{B}^{\mathrm{A}}$
    Sample a random minibatch of N^{A} records $(\boldsymbol{e}_z^{\mathrm{A}}, \boldsymbol{\phi}_z^{0,\mathrm{A}}, \lambda_z^{\mathrm{A}})$ from $\mathcal{B}^{\mathrm{A}}$
    Set $y_z^{\mathrm{A}} = \lambda_z^{\mathrm{A}} + \gamma^{\mathrm{A}} H_{\varepsilon^{\mathrm{A},'}}^{'}(\boldsymbol{e}_z^{\mathrm{A}}, \boldsymbol{\phi}_k^{'0,\mathrm{A}})$, where $\boldsymbol{\phi}_k^{'0,\mathrm{A}}$ is obtained using $\varepsilon_{\omega^{\mathrm{A},'}}^{\mathrm{A}}$
    Update the quality-based contract evaluation network by minimizing the loss $\mathcal{L}^{\mathrm{A}} = \frac{1}{N^{\mathrm{A}}}\sum_z (y_z^{\mathrm{A}} - H_{\upsilon^{\mathrm{A}}}(\boldsymbol{e}_z^{\mathrm{A}}, \boldsymbol{\phi}_z^{\mathrm{A}}))$
    Update the quality-based contract design network by computing the policy gradient $\bigtriangledown_{\omega}\varepsilon_{\omega} \approx \frac{1}{N^{\mathrm{A}}}\sum_k \bigtriangledown_{\boldsymbol{\phi}^{0,\mathrm{A}}} H_{\upsilon^{\mathrm{A}}}(\boldsymbol{e}^{\mathrm{A}}, \boldsymbol{\phi}^{0,\mathrm{A}})|_{\boldsymbol{e}^{\mathrm{A}}=\boldsymbol{e}_z^{\mathrm{A}}} \bigtriangledown_{\omega^{\mathrm{A}}} \varepsilon_{\omega^{\mathrm{A}}}^{\mathrm{A}}|\boldsymbol{e}_z^{\mathrm{A}}$
    Update the target networks: $\omega^{\mathrm{A},'} \leftarrow \tau^{\mathrm{A}}\omega^{\mathrm{A}} + (1-\tau^{\mathrm{A}})\omega^{\mathrm{A},'}$ and $\upsilon^{\mathrm{A},'} \leftarrow \tau^{\mathrm{A}}\upsilon^{\mathrm{A}} + (1-\tau^{\mathrm{A}})\upsilon^{\mathrm{A},'}$
  end for
end for
return The trained quality-based contract design network $\varepsilon_{\omega}^{\mathrm{A}}$
Inference Phase:
Input the environment vector $\boldsymbol{e}^{\mathrm{A}}$
Generate the optimal quality-based contract design $\boldsymbol{\phi}^{0,\mathrm{A}}$ by denoising Gaussian noise using $\varepsilon_{\omega^{\mathrm{A}}}^{\mathrm{A}}$ according to (32)
The optimal quality-based contract design $\boldsymbol{\phi}^{0,\mathrm{A}}$

In addition, providing the image generation service consumes certain computational and communication resources. Referring to [23], [31], the cost of the computation energy consumption of optimizing a prompt is defined as $gb_{1,j}(\theta_i^{\mathrm{A}})\delta_j(\theta_i^{\mathrm{A}})\kappa_{1,j}(\theta_i^{\mathrm{A}})l_i^{\mathrm{A}}(x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}))^2$ where $b_{1,j}(\theta_i^{\mathrm{A}})$ is the cost per computation energy consumption of optimizing the prompt and $\kappa_{1,j}(\theta_i^{\mathrm{A}})$ is the effective switched capacitance. Similarly, the cost of the energy consumption of executing diffusion denoising is defined as $gb_{2,j}(\theta_i^{\mathrm{A}})\eta_j(\theta_i^{\mathrm{A}})\kappa_{2,j}(\theta_i^{\mathrm{A}})s_i^{\mathrm{A}}(y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}))^2$ where $b_{2,j}(\theta_i^{\mathrm{A}})$ is the cost per computation energy consumption of executing diffusion denoising and $\kappa_{2,j}(\theta_i^{\mathrm{A}})$ is the effective switched capacitance. According to [31], the cost of the communication energy consumption is denoted as $\frac{gb_{3,j}(\theta_i^{\mathrm{A}})d_j(\theta_i^{\mathrm{A}})r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})(\theta_i^{\mathrm{A}})}{(h_j(\theta_i^{\mathrm{A}}))^2}$ where $b_{3,j}(\theta_i^{\mathrm{A}})$ is the cost per communication energy consumption and $h_j(\theta_i^{\mathrm{A}})$ is the status of wireless connection. The ASP provides the image generation services and receives rewards from the users. The reward from the type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user is $p_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$. The ASP receives utility is the difference between the reward gained from the type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user and the total energy consumption, which is given as

$$\begin{aligned}U_{\mathrm{sp},j}^{\mathrm{T}}(\theta_i^{\mathrm{A}}) = p_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}) &- gb_{1,j}(\theta_i^{\mathrm{A}})\delta_j(\theta_i^{\mathrm{A}})\kappa_{1,j}(\theta_i^{\mathrm{A}})l_i^{\mathrm{A}}(x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}))^2 \\ &- gb_{2,j}(\theta_i^{\mathrm{A}})\eta_j(\theta_i^{\mathrm{A}})\kappa_{2,j}s_i^{\mathrm{A}}(y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}))^2 \\ &- \frac{gb_{3,j}(\theta_i^{\mathrm{A}})d_j(\theta_i^{\mathrm{A}})r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})(\theta_i^{\mathrm{A}})}{(h_j(\theta_i^{\mathrm{A}}))^2}.\end{aligned} \tag{23}$$

For all the types, the utility of the ASP is defined as

$$U_{\mathrm{sp}}^{\mathrm{T}}(\theta_i^{\mathrm{A}}) = \sum_{j=1}^{J} M_i q_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}) U_{\mathrm{sp},j}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \tag{24}$$

where $q_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ is the probability that a user's type of gain per expected latency reduction belongs to type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$.

B. Latency-based Contract Formulation

Similarly, the latency-based contract item can be denoted as $\Phi^{\mathrm{T}}(\theta_i^{\mathrm{A}}) = \left\{\Phi_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}) = (x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), p_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})), j \in \{1,\ldots,J\}\right\}$ where $\Phi_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ is made for type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user. The mathematical expression for the IR conditions, applicable to a type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user, can be represented as follows:

$$\begin{aligned}&\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})\mathbb{E}[D](x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})) - p_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}) \geq 0, \\ &j \in \{1,\ldots,J\}.\end{aligned} \tag{25}$$

The mathematical expression for the IC conditions, applicable to a type-$\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ user, can be represented as follows:

$$\begin{aligned}&\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})\mathbb{E}[D](x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})) - p_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}) \geq \\ &\theta_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})\mathbb{E}[D](x_{j'}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), y_{j'}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), r_{j'}^{\mathrm{T}}(\theta_i^{\mathrm{A}})) - p_{j'}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \\ &\forall j, j' \in \{1,\ldots,J\}.\end{aligned} \tag{26}$$

To maximize the utility of the ASP under the IR and IC conditions, a latency-based contract problem is also formulated as follows:

$$\begin{aligned}\textbf{Problem 2:}\quad & \max_{x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}),y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}),r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}),p_j^{\mathrm{T}}(\theta_i^{\mathrm{A}})} U_{\mathrm{sp}}^{\mathrm{T}}(\theta_i^{\mathrm{A}}) \\ \text{s.t.}\quad & (25), \text{and}\quad (26), j, j^{'} \in \{1,\ldots,J\}, \\ & x_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), y_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), r_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}), p_j^{\mathrm{T}}(\theta_i^{\mathrm{A}}) \geq 0, \\ & j \in \{1,\ldots,J\}.\end{aligned} \tag{27}$$

Since the objective function and constraints are not concave functions in **Problem 2**, it is difficult to use traditional methods to solve directly **Problem 2**.

C. GDM-based Scheme for Latency-based Contract Problem

The GDM-based scheme is also used to find the optimal latency-based contract items.

1) Training Phase: We first define the environment, latency-based contract design networks, and latency-based contract evaluation networks. The environment is represented as a vector $\boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}})$, which includes all factors that impact the optimal design of a latency-based contract.

$$\begin{aligned}\boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}) = \big\{ & \boldsymbol{t}^{\max}(\theta_i^{\mathrm{A}}), \boldsymbol{b}_1(\theta_i^{\mathrm{A}}), \boldsymbol{b}_2(\theta_i^{\mathrm{A}}), \boldsymbol{b}_3(\theta_i^{\mathrm{A}}), \boldsymbol{\theta}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \boldsymbol{q}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \\ & \boldsymbol{\delta}(\theta_i^{\mathrm{A}}), \boldsymbol{\eta}, \boldsymbol{d}(\theta_i^{\mathrm{A}}), \boldsymbol{h}(\theta_i^{\mathrm{A}}), \boldsymbol{\kappa}_1(\theta_i^{\mathrm{A}}), \boldsymbol{\kappa}_2(\theta_i^{\mathrm{A}}), l_i^{\mathrm{A}}, s_i^{\mathrm{A}}, M_i, J \big\}.\end{aligned} \tag{28}$$

A latency-based contract design policy, denoted by $\pi_{\omega^{\mathrm{T}}}^{\mathrm{T}}(\boldsymbol{\phi}^{\mathrm{T}}(\theta_i^{\mathrm{A}})|\boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}))$. Then, we use the reverse process of a conditional diffusion model to represent the latency-based contract design policy as follows:

$$\begin{aligned}&\pi_{\omega^{\mathrm{T}}}^{\mathrm{T}}(\boldsymbol{\phi}^{\mathrm{T}}(\theta_i^{\mathrm{A}}) \mid \boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}})) = P_{\omega^{\mathrm{T}}}^{\mathrm{T}}\left(\boldsymbol{\phi}^{0:K^{\mathrm{T}}}(\theta_i^{\mathrm{A}}) \mid \boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}})\right) \\ &= \mathcal{N}^{\mathrm{T}}\left(\boldsymbol{\phi}^{K^{\mathrm{T}}}(\theta_i^{\mathrm{A}}); \mathbf{0}, \mathbf{I}^{\mathrm{T}}\right) \prod_{k=1}^{K^{\mathrm{T}}} P_{\omega^{\mathrm{T}}}^{\mathrm{T}}(\theta_i^{\mathrm{A}}),\end{aligned} \tag{29}$$

where $P_{\omega^{\mathrm{T}}}^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ is a simplified form of $P_{\omega^{\mathrm{T}}}^{\mathrm{T}}(\boldsymbol{\phi}^{k-1,\mathrm{T}}(\theta_i^{\mathrm{A}})|\boldsymbol{\phi}^{k,\mathrm{T}}(\theta_i^{\mathrm{A}}), \boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}))$. According to [28], $P_{\omega^{\mathrm{T}}}^{\mathrm{T}}(\theta_i^{\mathrm{A}})$ can be also modeled as a noise prediction model, with the covariance matrix fixed as:

$$\boldsymbol{\Sigma}_{\omega^{\mathrm{T}}}\left(\boldsymbol{\phi}^{k,\mathrm{T}}(\theta_i^{\mathrm{A}}), \boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), k\right) = \beta_k^{\mathrm{T}} \mathbf{I}^{\mathrm{T}}, \tag{30}$$

and the mean is defined as follows:

$$\begin{aligned}&\mu_{\omega^{\mathrm{T}}}^{\mathrm{T}}\left(\boldsymbol{\phi}^{k,\mathrm{T}}(\theta_i^{\mathrm{A}}), \mathbf{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), k\right) \\ &= \frac{1}{\sqrt{\alpha_k^{\mathrm{T}}}}\left(\boldsymbol{\phi}^{k,\mathrm{T}}(\theta_i^{\mathrm{A}}) - \frac{\beta_k^{\mathrm{T}}}{\sqrt{1-\bar{\alpha}_k^{\mathrm{T}}}} \varepsilon_{\omega^{\mathrm{T}}}^{\mathrm{T}}\left(\boldsymbol{\phi}^{k,\mathrm{T}}(\theta_i^{\mathrm{A}}), \mathbf{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), k\right)\right).\end{aligned} \tag{31}$$

We begin by sampling $\boldsymbol{\phi}^{K^{\mathrm{T}}} \sim \mathcal{N}^{\mathrm{T}}(\mathbf{0}, \mathbf{I}^{\mathrm{T}})$ and then proceed with the reverse diffusion chain, parameterized by ω^{T}:

$$\begin{aligned}&\boldsymbol{\phi}^{k-1,\mathrm{T}}(\theta_i^{\mathrm{A}}) \mid \boldsymbol{\phi}^{k,\mathrm{T}}(\theta_i^{\mathrm{A}}) = \frac{\boldsymbol{\phi}^{k,\mathrm{T}}(\theta_i^{\mathrm{A}})}{\sqrt{\alpha_k^{\mathrm{T}}}} \\ &- \frac{\beta_k^{\mathrm{T}}}{\sqrt{\alpha_k^{\mathrm{T}}\left(1-\bar{\alpha}_k^{\mathrm{T}}\right)}} \varepsilon_{\omega^{\mathrm{T}}}^{\mathrm{T}}\left(\boldsymbol{\phi}^{k,\mathrm{T}}(\theta_i^{\mathrm{A}}), \mathbf{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), k\right) + \sqrt{\beta_k^{\mathrm{T}}}\varepsilon^{\mathrm{T}}.\end{aligned} \tag{32}$$

We train a network, denoted as $\varepsilon_{\omega^{\mathrm{T}}}^{\mathrm{T}}$, to generate latency-based contracts. This network is then used to train a latency-based contract design policy, denoted as $\pi_{\omega^{\mathrm{T}}}^{\mathrm{T}}$ in complex and high-dimensional environments, denoted as $\boldsymbol{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}})$. In the same way, we can also obtain the optimal latency-based contract design policy by minimizing the loss function $\mathcal{L}^{\mathrm{T}}(\omega^{\mathrm{T}})$ using double Q-learning in the following manner:

$$\begin{aligned}\pi^{\mathrm{T}} &= \arg\min_{\pi_{\omega^{\mathrm{T}}}^{\mathrm{T}}} \mathcal{L}^{\mathrm{T}}(\omega^{\mathrm{T}}) \\ &= -\mathbb{E}_{\boldsymbol{\phi}^{0,\mathrm{T}}(\theta_i^{\mathrm{A}}) \sim \pi_{\omega^{\mathrm{T}}}^{\mathrm{T}}}\left[H_{\upsilon^{\mathrm{T}}}^{\mathrm{T}}\left(\mathbf{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \boldsymbol{\phi}^{0,\mathrm{T}}(\theta_i^{\mathrm{A}})\right)\right].\end{aligned} \tag{33}$$

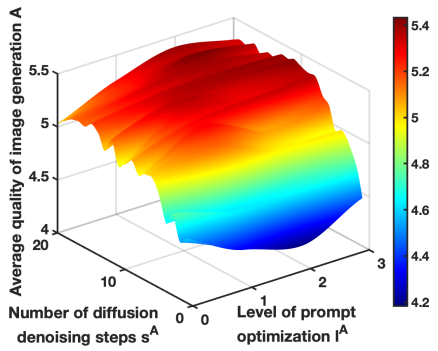
The network of evaluating latency-based contracts also employs the double Q-learning method for its training. It involves the formulation of two primary networks, designated as $H_{\upsilon_1^{\mathrm{T}}}^{\mathrm{T}}$ and $H_{\upsilon_2^{\mathrm{T}}}^{\mathrm{T}}$, and their corresponding target counterparts, named $H_{\upsilon_1^{\mathrm{T},\prime}}^{\mathrm{T}}$, $H_{\upsilon_2^{\mathrm{T},\prime}}^{\mathrm{T}}$ and $\pi_{\omega^{\mathrm{T},\prime}}^{\mathrm{T}}$. The goal is to optimize $\upsilon_{i,n}^{\mathrm{T}}$ for $n = 1, 2$ through minimization of the objective

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\phi}_{k+1}^{0,\mathrm{T}}(\theta_i^{\mathrm{A}}) \sim \pi_{\omega_i^{\mathrm{T},\prime}}^{\mathrm{T}}}\Bigg[\Big|\Big|\left(r(\mathbf{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \boldsymbol{\phi}_k^{\mathrm{T}}(\theta_i^{\mathrm{A}})) + \right. \\ &\left.\gamma^{\mathrm{T}} \min_{n=1,2} H_{\upsilon_n^{\mathrm{T},\prime}}^{\mathrm{T}}(\mathbf{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \boldsymbol{\phi}_{k+1}^{0,\mathrm{T}}(\theta_i^{\mathrm{A}}))\right) - H_{\upsilon_n^{\mathrm{T},\prime}}^{\mathrm{T}}(\mathbf{e}^{\mathrm{T}}(\theta_i^{\mathrm{A}}), \boldsymbol{\phi}_k^{\mathrm{T}}(\theta_i^{\mathrm{A}}))\Big|\Big|^2\Bigg].\end{aligned} \tag{34}$$

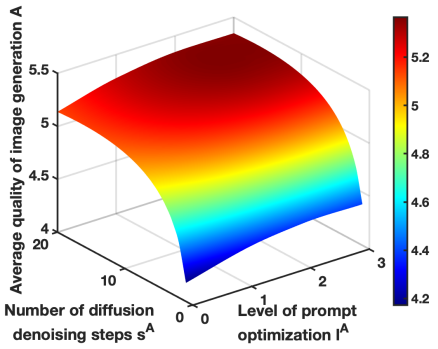
2) Inference Stage: The trained latency-based contract design network is used during the inference phase to generate efficient latency-based contract items based on current environmental parameters.

VI. Simulation Results

First, we employ an approximation approach to quantitatively evaluate the relationship between the level of prompt optimization, the number of diffusion denoising steps, and the quality of image generation, which is a common practice in the literature and has been adopted in other works, such as [9], [10]. Second, the approximation approach is also used to quantitatively assess the relationship between the level of prompt optimization, the number of diffusion denoising steps, and the probability that the quality of image generation exceeds the threshold $\overline{A}$. According to the data shown in Fig. 4, certain generated images do not meet the production criteria for user prompt word requests when the image quality falls below $\overline{A} = 4.5$; for example, $(s^{\mathrm{A}}, s^{\mathrm{T}}) = (5, 3)$ or $(s^{\mathrm{A}}, s^{\mathrm{T}}) = (5, 2)$. Furthermore, other images do not meet the criteria if the quality is less than $\overline{A} = 5.0$, such as $(s^{\mathrm{A}}, s^{\mathrm{T}}) = (7, 1)$ in our dataset. To obtain more consistent results in the simulation experiment, we established the quality threshold at $\overline{A} = 5.0$. Note that this threshold might differ for various datasets. However, our analysis method is still applicable to other datasets. Third, we introduce the setting of the GDM. Fourth, we evaluate the two-stage GDM-based contract generation scheme and demonstrate its superior performance compared to an existing DRL-based contract generation scheme. Continuously, the validity of the generated quality-latency contract is verified. Finally, we analyze the impact of prompt optimization on performance.



(a) Real experimental data.



(b) Fitted function.

Fig. 2: Real experimental data and fitted function.

A. Quantity of Quality of Image Generation

We employ an approximation approach to determine the relationship between the level of prompt optimization, the number of diffusion denoising steps, and the quality of image generation. The steps are as follows. In the first step, we define an original prompt, for instance, *an apple on the desk*. In the second step, referring to [32], we use a fixed learning algorithm to adjust different level l to optimize the original prompt. In the third step, the optimized prompt is inputted into the Stable Diffusion XL model [33], and the number of diffusion denoising steps is varied to obtain different output images. In the fourth step, the neural image assessment model [8] is used to access the quality of each image. These steps are performed $L \times S$ times to obtain the set $\{A_{l,s} | l \in [1, L], s \in [1, S]\}$, where L is the maximum level and S is the maximum number of diffusion denoising steps. In the fifth step, repeating the above steps 100 times to obtain the average experimental result, which is shown in Fig. 2(a). As the level of prompt optimization level and the number of diffusion denoising steps increase, the average quality of image generation improves. To numerically analyze the experimental results, we define A as follows:

$$A = \rho_1 \ln(\rho_2 l + 1) - \rho_3 l + \rho_4 \ln(\rho_5 s + 1) - \rho_6 s. \quad (35)$$

The algorithm for non-linear least squares modifies the values of $\boldsymbol{\rho}$ in order to minimize the sum of squared errors. The specific values for $\rho_1 = 9.7417, \rho_2 = 0.0978, \rho_3 = 0.7647, \rho_4 = 0.5158, \rho_5 = 3497.8463$, and $\rho_6 = 0.0307$ are used in this optimization process. The results of the fitted function are shown in Fig. 2(b). The above approximation approach can be extended to a wide variety of AIGC services.

B. Quantity of Probability of Image Generation Quality Exceeding a Threshold

We then use the frequency to approximate the probability $\zeta\left(A\left(l^{\mathrm{A}}, s^{\mathrm{A}}\right) > \overline{A}\right)$, as illustrated in Fig. 3. In Fig. 3, as the level of prompt optimization and the number of diffusion denoising steps increase, the probability $\zeta\left(A\left(l^{\mathrm{A}}, s^{\mathrm{A}}\right) > \overline{A}\right)$ increases. Note that most of the results generated are invalid when the number of inference steps is less than or equal to 3, as shown in Fig. 4. Therefore, the lower bound of the diffusion denoising step is set to $s^{A,\min} = 4$.

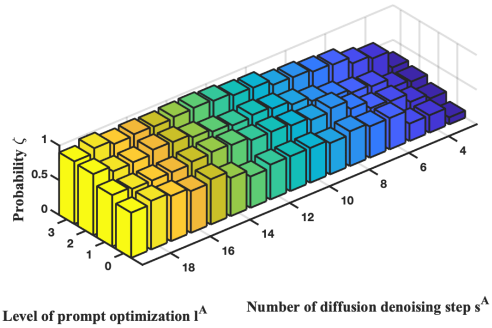


Fig. 3: Probability $\zeta\left(A\left(l^{\mathrm{A}}, s^{\mathrm{A}}\right) > \overline{A}\right)$ with different combinations of l^{A} and s^{A}.

C. Setting of GDM

Experimental Platform. Our algorithms are tested on a platform featuring Ubuntu 20.04 as the operating system, powered by an AMD Ryzen Threadripper PRO 3975WX with 32 cores CPU and complemented by an NVIDIA RTX A5000 GPU for enhanced performance.

GDM Design We utilize the diffusion model as the basis of the contract design network and two contract evaluation networks with the same structure to reduce the issue of overestimation, as reported in [14]. The configurations of the contract design and evaluation networks are described in Table III. For the quality-based contract generation model and the latency-based contract generation model, Table IV summarizes the detailed settings for other training hyperparameters in our experiments. According to [18], for the quality-based contract generation, we set $M = 20$ and $I = 2$; θ_1^{A} and θ_2^{A} are randomly sampled within $[1, 200]$ and $[200, 400]$ respectively; q_1^{A} and q_2^{A} are generated randomly; σ_1 and σ_2 are randomly sampled within $[1, 10]$. According to [15], [21], [23], for the latency-based contract generation, we set $M_1 = M_2 = 10$ and $J_1 = J_2 = 2$; $\theta_{1,1}^{\mathrm{T}}$ and $q_{2,1}^{\mathrm{T}}$ are randomly sampled within $[1, 25]$ while $\theta_{1,2}^{\mathrm{T}}$ and $q_{2,2}^{\mathrm{T}}$ are randomly sampled within $[25, 50]$; $q_{1,1}^{\mathrm{T}}$, $q_{1,2}^{\mathrm{T}}$, $q_{2,1}^{\mathrm{T}}$ and $q_{2,2}^{\mathrm{T}}$ are generated randomly; l and s are randomly sampled within $[1, 20]$ and $[0, 3]$ respectively; d is randomly sampled within $[5, 8] \times 10^5$ bit; $t^{\max}$ is randomly

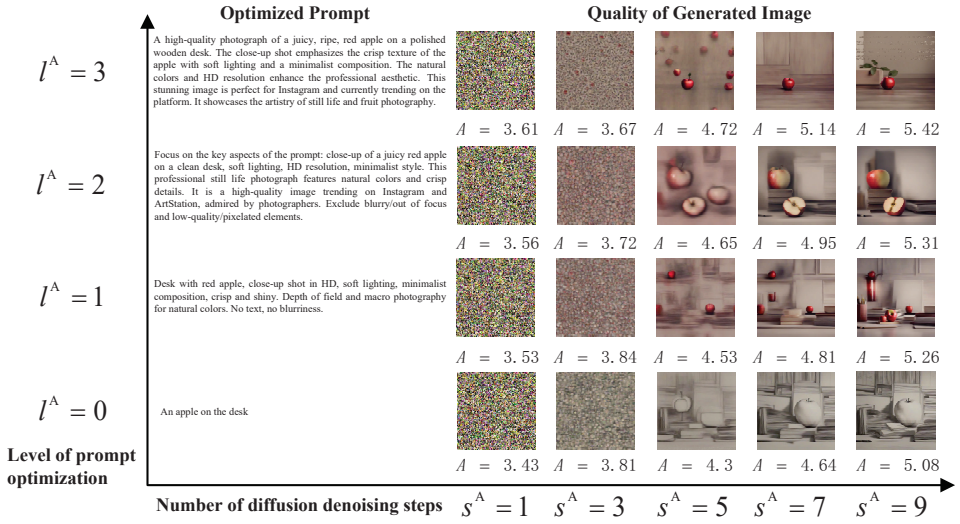


Fig. 4: Quality of generated image with different combinations of l^{A} and s^{A}.

TABLE III: Structure of Contract Design and Evaluation Networks

Networks	Layer	Activation	Units
Design	SinusoidalPosEmb	-	16
	FullyConnect	Tanh	32
	FullyConnect	-	16
	Concatenation	-	-
	FullyConnect	Tanh	256
	FullyConnect	Tanh	256
	FullyConnect	Tanh	12
Evaluation	FullyConnect	Mish	256
	FullyConnect	Mish	256
	FullyConnect	Mish	256
	FullyConnect	-	1

TABLE IV: Summary of Training Hyperparameter.

Hyperparameter	Setting in Quality-based Contract Generation Model	Setting in Latency-based Contract Generation Model
Learning rate of the contract design network	8×10^{-9}	10^{-6}
Learning rate of the contract evaluation network	8×10^{-9}	10^{-6}
Soft target update parameter	$\tau^{\mathrm{A}}=0.005$	$\tau^{\mathrm{T}}=0.005$
Batch size	$N^{\mathrm{A}}=10^6$	$N^{\mathrm{T}}=10^6$
Discount factor	$\gamma^{\mathrm{A}}=0.95$	$\gamma^{\mathrm{T}}=0.95$
Number of iterations for adding noise	$K^{\mathrm{A}}=3$	$K^{\mathrm{T}}=3$
Maximum capacity of the replay buffer	$\mathcal{B}^{\mathrm{A}}=10^6$	$\mathcal{B}^{\mathrm{T}}=10^6$
Exploration Noise	$\epsilon^{\mathrm{A}}=0.01$	$\epsilon^{\mathrm{T}}=0.01$
Max episode	$Z_e^{\mathrm{A}}=1000$	$Z_e^{\mathrm{T}}=1000$
Max step	$Z_s^{\mathrm{A}}=1$	$Z_s^{\mathrm{T}}=1$
Penalty	$\xi^{\mathrm{A}}=-300$	$\xi^{\mathrm{T}}=-200$

sampled within $[1,4]$ s; b_1 and b_2 are randomly sampled within $[8,10]\times10^7$; b_3 is randomly sampled within $[3,5]\times10^{-4}$; h is randomly sampled within $[3,5]\times10^6$; κ_1 and κ_2 are randomly sampled within $[1,4]\times10^{-28}$; η_1 and η_2 are randomly sampled within $[3000,5000]$ cycles/bit.

D. Efficiency of Two-stage GDM-based Contract Generation Scheme

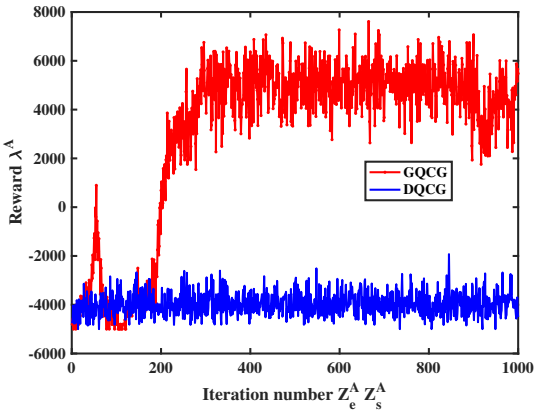


Fig. 5: Training process of GDM-based and DRL-based quality contract generation schemes.

1) GDM-based quality contract design: Fig. 5 shows the test reward curves of our GDM-based quality contract generation (GQCG) scheme and the DRL-based quality contract generation (DQCG) scheme. Our proposed GQCG scheme consistently outperforms the DQCG scheme when the same parameters are used. This is because the quality contract generation policy in our scheme is fine-tuned by the diffusion process, which reduces the effect of randomness and noise [1].

For a given environment state, we verify the validity of the generated quality contract items. Fig. 6 shows the validation of the IC and IR constraints in the proposed GDM-based quality contract design. We evaluate the utilities of different users with various types of gain per quality when selecting different quality-based contract items from the ASP. From Fig. 6, we validate that our quality-based contract design satisfies the IR and IC constraints. A user with an arbitrary type achieves the maximal utility with a non-negative value only when accepting the quality contract item matched with its type. The selection

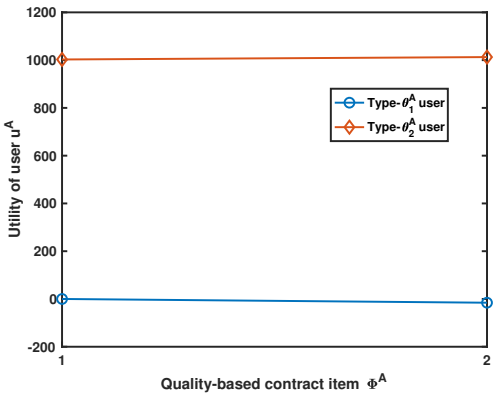


Fig. 6: Utility of user versus types of quality contract item.

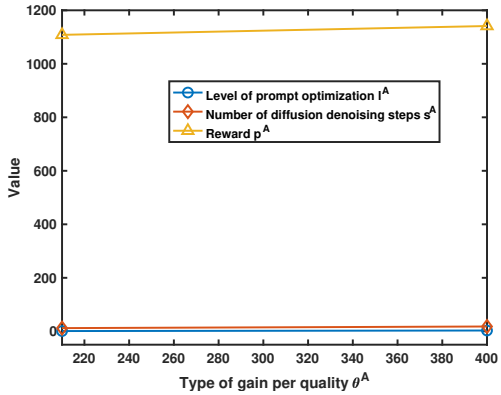


Fig. 7: Quality-based contract value under different types.

process of the quality contract item enables the types of user to be indirectly revealed to the ASP. This means that quality-based contract design is effective in solving the information asymmetry problem for the ASP. Fig. 7 shows the number of diffusion denoising steps, the level of prompt optimization, and the reward for ASP with respect to different types of gain per quality. To increase the quality of the inferred results, users with the higher types need to give more rewards to increase the number of diffusion denoising steps and the level of prompt optimization.

2) GDM-based latency contract design: The curves in Fig. 8 illustrate that our GDM-based latency contract generation (GLCG) scheme is more effective than the conventional DRL-based latency contract generation (DLCG) scheme when the same parameters are employed. The reason is similar to the reason for the results in Fig. 5.

For a given environment state, we will verify the validity of the generated latency-based contract items. After 20 users select the quality-based contract items, the ASP implements a latency-based contract design for these users selecting the same quality-based contract item. 10 users choose $(l_1^{\mathrm{A}}, s_1^{\mathrm{A}}, p_1^{\mathrm{A}})$, their maximum requested time is $t^{\max} = 3$s. 10 users choose $(l_2^{\mathrm{A}}, s_2^{\mathrm{A}}, p_2^{\mathrm{A}})$, their maximum requested latency is $t^{\max} = 4$s. The reason is similar to the reason for the results in Fig. 6. Fig. 9 validates the IC and IR constraints in the proposed

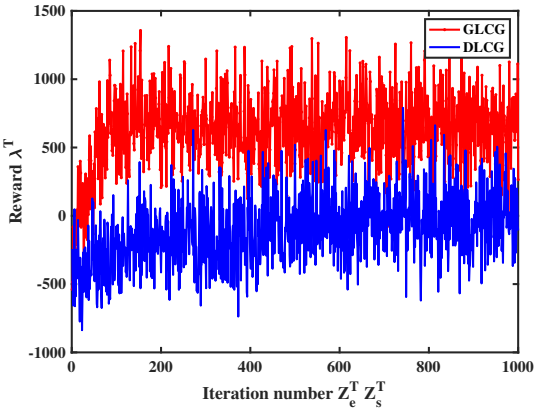
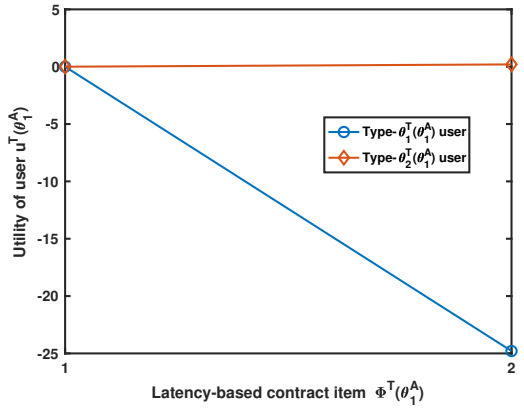
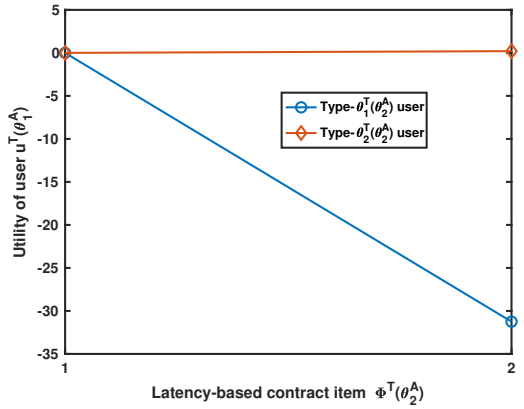


Fig. 8: Training process of GDM-based and DRL-based latency contract generation schemes.



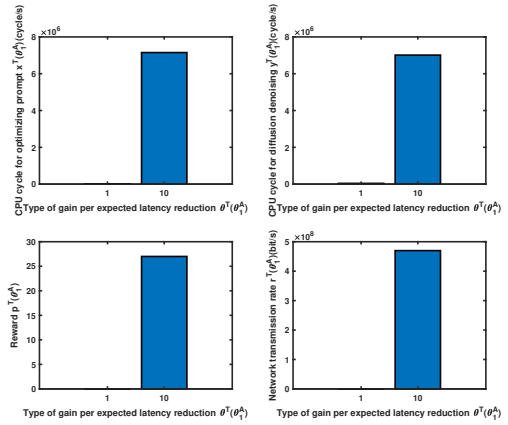
(a) $t^{\max} = 3$s, $l_1^{\mathrm{A}} = 1$ and $s_1^{\mathrm{A}} = 13$.



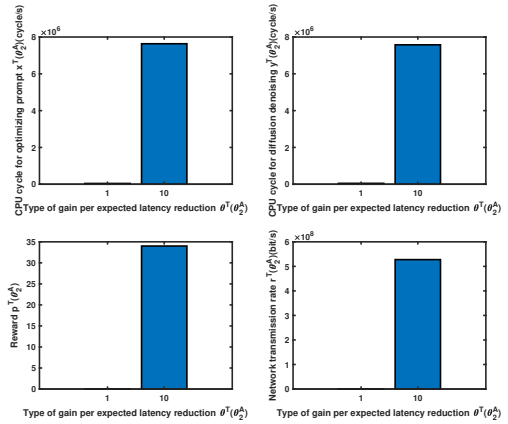
(b) $t^{\max} = 4$s, $l_2^{\mathrm{A}} = 2$ and $s_2^{\mathrm{A}} = 17$.

Fig. 9: Verification of latency-based contract design under different quality-based contract items.

latency-based contract design with various quality-based contract items, such as $t^{\max}$, l^{A}, and s^{A}. Fig. 10 shows CPU cycle for optimizing prompt and diffusion denoising, network transmission rate, and the reward for the ASP under different quality-based contract items.



(a) $t^{\max} = 3$s, $l_1 = 1$ and $s_1 = 13$.



(b) $t^{\max} = 4$s, $l_2 = 2$ and $s_2 = 17$.

Fig. 10: Latency-based contract value under different quality-based contract items.

E. Impact of prompt optimization on performance

Fig. 11 shows the impact of prompt optimization on quality-based contract design. The first approach is not to optimize the prompt. The second approach is to use prompt optimization, which is the approach proposed in this paper. Fig. 11 illustrates that the use of prompt optimization can be beneficial to improve both the ASP utility in Fig. 11 (a), the users' utilities in Fig. 11(b), and the quality of the diffusion denoising result in Fig. 11(c). In addition, as the type of gain per quality increases, so does the ASP utility, users' utilities, and the quality of the diffusion denoising result. Particularly, for type-θ_1^{A} and type-θ_2^{A} users, the quality of the diffusion denoising result is improved by 8% and 2%, respectively. The causes are summarized below. Due to the lack of prompt optimization, the quality of the generated images decreases, resulting in a significant drop in user satisfaction. However, the reduction in the amount users are willing to pay a reward may only decrease linearly, creating an asymmetry that directly impacts user utility. For example, a user who expects to generate a high-quality landscape image for use as wallpaper may receive a blurry, low-detail image due to the lack of prompt optimization. Although the user experiences significant disappointment, they can only reduce their payment from 20 dollars to 15 dollars rather than refuse to pay entirely. This linear reduction in payment fails to fully capture the user's strong dissatisfaction, ultimately leading to a substantial decrease in overall utility and perceived value.

Once users have chosen the same quality-based contract item, e.g. $l^{\mathrm{A}} = 2, s^{\mathrm{A}} = 17$, Fig. 12 displays the impact of prompt optimization on latency-based contract design. Those who employed prompt optimization selected a high-quality contract item, i.e. $l^{\mathrm{A}} = 2, s^{\mathrm{A}} = 17$, while those who did not use prompt optimization chose a contract item of lesser quality, i.e. $l^{\mathrm{A}} = 0, s^{\mathrm{A}} = 17$. The results illustrate that prompt optimization can be beneficial for enhancing the ASP utility in Fig. 12 (a), as well as for the users' utilities in Fig. 12 (b), and the expected latency reduction in Fig 12 (c). The explanation for the results shown in Fig. 12 (a) and Fig. 12 (b) is analogous to the reasoning behind the results in Fig. 11. To explain the results in Fig. 12 (c), when the number of diffusion denoising iterations is constant, employing prompt optimization boosts the likelihood of producing an image that meets the user's quality requirements. This, in turn, reduces the probability of needing to regenerate the image, thereby enhancing the total expected latency reduction. Additionally, as the gain per quality increases, the ASP utility, user utilities, and expected latency reduction all improve. For instance, for type-$\theta_2^{\mathrm{T}}(\theta_2^{\mathrm{A}})$ users, the expected latency reduction is increased by 22%.

VII. Conclusion

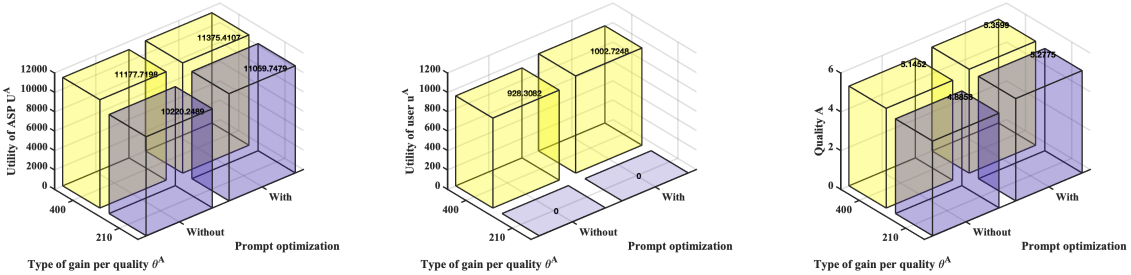
In this paper, we propose a two-stage, multi-dimensional resource allocation framework based on a GDM and contract theory. First, based on the quality of AIGC generation, we establish a model for the user and ASP utilities, leading to a quality contract problem. Its objective is to maximize the utility of the ASP. Then, a GDM-based scheme optimizes quality-based contract items. Users choose quality-based contract items based on their types of gain per quality, and then a non-convex latency-based contract problem is formulated for each group of users selecting identical quality-based contract items. The optimal latency-based contract items are again resolved using the GDM-based scheme. The numerical results show that the proposed GDM-based scheme is very advantageous to improve the quality of AIGC generation and decrease the latency of AIGC generation, compared to other standard schemes. Future work will focus on the design of a multitask incentive mechanism considering the effects of the irrational behavior of mobile terminals on the behavioral decisions of mobile terminals and ASPs.

References

[1] H. Du, R. Zhang, Y. Liu, J. Wang, Y. Lin, Z. Li, D. Niyato, J. Kang, Z. Xiong, S. Cui, *et al.*, "Beyond deep reinforcement learning: A tutorial on generative diffusion models in network optimization," *arXiv preprint arXiv:2308.05384*, 2023.

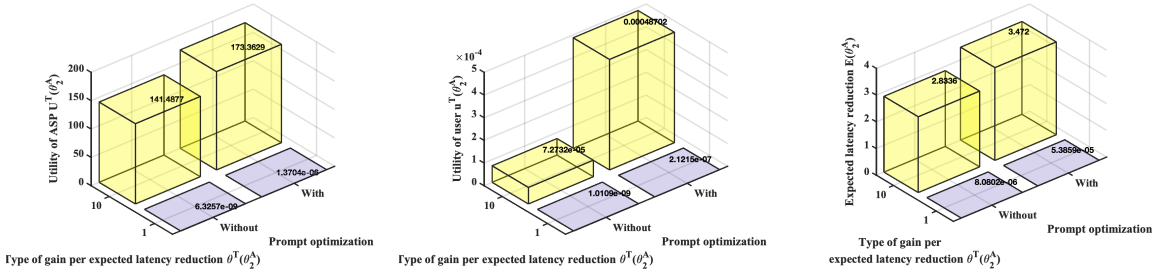
[2] H. Zou, Q. Zhao, L. Bariah, M. Bennis, and M. Debbah, "Wireless multi-agent generative ai: From connected intelligence to collective intelligence," *arXiv preprint arXiv:2307.02757*, 2023.

[3] M. A. Ferrag, M. Ndhlovu, N. Tihanyi, L. C. Cordeiro, M. Debbah, and T. Lestable, "Revolutionizing cyber threat detection with large language models," *arXiv preprint arXiv:2306.14263*, 2023.



(a) Utility of ASP. (b) Utility of user. (c) Quality

Fig. 11: Impact of prompt optimization on quality-based contract design.



(a) Utility of ASP. (b) Utility of user. (c) Expected latency reduction

Fig. 12: Impact of prompt optimization on latency-based contract design.

[4] Y. Liu, H. Du, D. Niyato, J. Kang, S. Cui, X. Shen, and P. Zhang, "Optimizing mobile-edge ai-generated everything (aigx) services by prompt engineering: Fundamental, framework, and case study," *arXiv preprint arXiv:2309.01065*, 2023.

[5] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[6] Y. Liu, H. Du, D. Niyato, J. Kang, Z. Xiong, C. Miao, A. Jamalipour, *et al.*, "Blockchain-empowered lifecycle management for ai-generated content (aigc) products in edge networks," *arXiv preprint arXiv:2303.02836*, 2023.

[7] Y. Hao, Z. Chi, L. Dong, and F. Wei, "Optimizing prompts for text-to-image generation," *arXiv preprint arXiv:2212.09611*, 2022.

[8] H. Talebi and P. Milanfar, "Nima: Neural image assessment," *IEEE transactions on image processing*, vol. 27, no. 8, pp. 3998–4011, 2018.

[9] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020.

[10] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Transactions on Mobile Computing*, vol. 20, no. 10, pp. 3034–3048, 2020.

[11] M. Xu, D. Niyato, H. Zhang, J. Kang, Z. Xiong, S. Mao, and Z. Han, "Sparks of gpts in edge intelligence for metaverse: Caching and inference for mobile aigc services," *arXiv preprint arXiv:2304.08782*, 2023.

[12] M. Xu, D. Niyato, H. Zhang, J. Kang, Z. Xiong, S. Mao, and Z. Han, "Joint foundation model caching and inference of generative ai services for edge intelligence," in *2023 IEEE Global Communications Conference*, pp. 3548–3553, IEEE, 2023.

[13] X. Lyu, S. Rani, and Y. Feng, "Optimizing aigc service provider selection based on deep q-network for edge-enabled healthcare consumer electronics systems," *IEEE Transactions on Consumer Electronics*, 2024.

[14] H. Du, Z. Li, D. Niyato, J. Kang, Z. Xiong, H. Huang, and S. Mao, "Generative ai-aided optimization for ai-generated content (aigc) services in edge networks," *arXiv preprint arXiv:2303.13052*, 2023.

[15] H. Du, Z. Li, D. Niyato, J. Kang, Z. Xiong, D. I. Kim, *et al.*, "Enabling ai-generated content (aigc) services in wireless edge networks," *arXiv preprint arXiv:2301.03220*, 2023.

[16] S. Zhang, M. Xu, W. Y. B. Lim, and D. Niyato, "Sustainable aigc workload scheduling of geo-distributed data centers: A multi-agent reinforcement learning approach," *arXiv preprint arXiv:2304.07948*, 2023.

[17] J. Wang, H. Du, D. Niyato, J. Kang, Z. Xiong, D. Rajan, S. Mao, *et al.*, "A unified framework for guiding generative ai with wireless perception in resource constrained mobile edge networks," *arXiv preprint arXiv:2309.01426*, 2023.

[18] Y. Liu, H. Du, D. Niyato, J. Kang, Z. Xiong, D. I. Kim, and A. Jamalipour, "Deep generative model and its applications in efficient wireless network management: A tutorial and case study," *arXiv preprint arXiv:2303.17114*, 2023.

[19] J. Wen, J. Kang, M. Xu, H. Du, Z. Xiong, Y. Zhang, and D. Niyato, "Freshness-aware incentive mechanism for mobile ai-generated content (aigc) networks," in *2023 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 1–6, IEEE, 2023.

[20] Z. Zhan, Y. Dong, Y. Hu, S. Li, S. Cao, and Z. Han, "Vision language model-empowered contract theory for aigc task allocation in teleoperation," *arXiv preprint arXiv:2407.17428*, 2024.

[21] Y. Wang, C. Liu, and J. Zhao, "Offloading and quality control for ai generated content services in edge computing networks," *arXiv preprint arXiv:2312.06203*, 2023.

[22] H. Chung, B. Sim, and J. C. Ye, "Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12413–12422, 2022.

[23] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle selection and resource optimization for federated learning in vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11073–11087, 2021.

[24] D. Ye, X. Huang, Y. Wu, and R. Yu, "Incentivizing semisupervised vehicular federated learning: A multidimensional contract approach with bounded rationality," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18573–18588, 2022.

[25] B. Zhang, L. Wang, and Z. Han, "Contracts for joint downlink and uplink traffic offloading with asymmetric information," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 4, pp. 723–735, 2020.

[26] L. Gao, X. Wang, Y. Xu, and Q. Zhang, "Spectrum trading in cognitive radio networks: A contract-theoretic modeling approach," *IEEE Journal*

on Selected Areas in Communications, vol. 29, no. 4, pp. 843–855, 2011.
[27] A. Rényi, *Probability theory*. Courier Corporation, 2007.
[28] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
[29] H. Hasselt, "Double q-learning," *Advances in neural information processing systems*, vol. 23, 2010.
[30] Z. Sun and G. Chen, "Contract-optimization approach (coa): A new approach for optimizing service caching, computation offloading, and resource allocation in mobile edge computing network," *Sensors*, vol. 23, no. 10, p. 4806, 2023.
[31] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019-IEEE conference on computer communications*, pp. 1387–1395, IEEE, 2019.
[32] "Promptperfect elevate your prompts to perfection." https://promptperfect.jina.ai/prompts.
[33] C. Mou, X. Wang, L. Xie, Y. Wu, J. Zhang, Z. Qi, Y. Shan, and X. Qie, "T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models," *arXiv preprint arXiv:2302.08453*, 2023.

Dongdong Ye received the Ph.D. degree in control science and engineering from the Guangdong University of Technology, Guangzhou, China, in 2021. He is currently a Postdoctoral Fellow with the Guangdong University of Technology. His research interests include game theory, resource management in wireless communications, and networking.

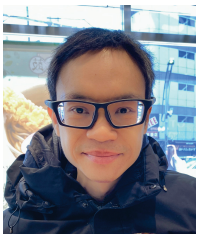
Shuting Cai received the B.Sc. and M.Sc. degrees in computer science from Central South University, Changsha, China, in 2001 and 2004, respectively, and the Ph.D. degree in control science and engineering from the Guangdong University of Technology, Guangzhou, China, in 2011. He is currently a Professor with Guangdong University of Technology. His current research interests include hardware architectures, multimedia signal processing, and computer vision.

Hongyang Du is an assistant professor at the Department of Electrical and Electronic Engineering, The University of Hong Kong. He received the BEng degree from the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, in 2021, and the PhD degree from the Interdisciplinary Graduate Program at the College of Computing and Data Science, Energy Research Institute @ NTU, Nanyang Technological University, Singapore, in 2024. He serves as the Editor-in-Chief assistant of IEEE Communications Surveys & Tutorials (2022-2024), and the Guest Editor for IEEE VTM. He is the recipient of the IEEE Daniel E. Noble Fellowship Award from the IEEE Vehicular Technology Society in 2022, the IEEE Signal Processing Society Scholarship from the IEEE Signal Processing Society in 2023, the Singapore Data Science Consortium (SDSC) Dissertation Research Fellowship in 2023, and NTU Graduate College's Research Excellence Award in 2024. He was recognized as an exemplary reviewer of the IEEE Transactions on Communications and IEEE Communications Letters in 2021. His research interests include edge intelligence, generative AI, semantic communications, and network management.

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

Yinqiu Liu received B.Eng. degree from Nanjing University of Posts and Telecommunications, China in 2020 and M.Sc degree from the University of California, Los Angeles in 2022. He is currently a Ph.D. student at the College of Computing and Data Science, Nanyang Technological University, Singapore. His current research interests include wireless communications, mobile AIGC, and generative AI.

Rong Yu (Member, IEEE) received his B.S. degree in communication engineering from the Beijing University of Posts and Telecommunications, China, in 2002, and Ph.D. degree in electronic engineering from Tsinghua University, China, in 2007. After that, he was with the School of Electronic and Information Engineering, South China University of Technology. In 2010, he joined the School of Automation, Guangdong University of Technology, where he is currently a professor. His research interests mainly focus on wireless networking and mobile computing, such as edge computing, federated learning, blockchain, digital twin, connected vehicles, and smart grid.



Dusit Niyato (Fellow, IEEE) is a professor in the College of Computing and Data Science, at Nanyang Technological University, Singapore. He received B.Eng. from King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand and Ph.D. in Electrical and Computer Engineering from the University of Manitoba, Canada. His research interests are in the areas of mobile generative AI, edge intelligence, decentralized machine learning, and incentive mechanism design.