LLM-Enhanced Contract Design for Edge AIGC Service Provisioning in Teleoperation

Zijun Zhan*, Yaxian Dong†, Daniel Mawunyo Doe‡, Yuqing Hu†, Shuai Li§, Shaohua Cao¶, and Zhu Han*
*Electrical and Computer Engineering Department, University of Houston, Houston, TX, USA
†Department of Architectural Engineering, The Pennsylvania State University, University Park, PA, USA
‡Electrical and Computer Engineering Department, Prairie View A&M University, Prairie View, TX, USA
§Department of Civil & Coastal Engineering, University of Florida, Gainesville, FL, USA
¶Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, China

***Abstract*—With the prevailing of AI-generated content (AIGC) services, incentive mechanism design is an emerging field in AIGC networks. The incentive mechanism design under information asymmetry of hidden action remains underexplored. In this paper, we research the bonus design for a teleoperator that incentivizes the edge AIGC service provider (ASP) to provide high-quality AIGC services when the teleoperator cannot observe the ASP's private settings and chosen actions (diffusion steps). We formulate the problem as an online learning contract design problem and propose a large language model (LLM)-empowered framework to solve it efficiently. Specifically, we utilize the LLM to tackle the ASP's settings approximation problem, NP-hard and with unknown size optimization variables, via the LLM's expertise in various domains. Upon the ASP's approximated settings, we address the contract derivation problem using convex optimization techniques. Simulation results on our Unity-based teleoperation platform showcase that our proposed method augments the utility of the teleoperator by $5 \sim 40\%$ in comparison with benchmarks.**

***Index Terms*—large language model (LLM), edge AI-generated content (AIGC), incentive mechanism design, moral hazard.**

I. Introduction

The advanced capabilities of AI-generated content (AIGC) services have provided significant momentum across various sectors [1]–[3]. For example, diffusion-based low-light image enhancement supports seamless teleoperation in visually degraded environments, including tunnels, drainage systems, and nighttime construction sites. To meet the strict low-latency requirements of teleoperation [4], AIGC services are increasingly deployed at the network edge. Similar to classical task-offloading scenarios, the design of incentive mechanisms between AIGC service users (teleoperators) and edge AIGC service providers (edge ASPs) is critical for reliable collaboration. The objective of incentive mechanism design is to align the interests of different parties by structuring payments, rewards, or service terms, thereby encouraging self-interested agents (teleoperator and edge ASP in this context) to act in ways that enhance overall system performance.

However, this interaction is often complicated by information asymmetry, which is a fundamental challenge in economic and communication systems [5]. Concretely, teleoperators typically lack insight into the service provider's private information, such as the actual complexity of the AIGC model, the size of its training dataset, or the precise number of diffusion steps performed during inference. This lack of transparency can result in misaligned incentives, inefficient resource allocation, or strategic behavior. To this end, this paper investigates the design of incentive mechanisms that remain effective under information asymmetry between teleoperators and edge ASPs.

To design an effective incentive mechanism in AIGC networks under information asymmetry, the authors in [3], [6], [7] proposed to utilize contract theory to aid the incentive mechanism design. The authors in [3] observed that the AIGC tasks' difficulty distribution is hidden type information to the edge ASP. They proposed a differentiated pricing scheme for AIGC tasks offloading based on contract theory. The authors in [6] devised a Wasserstein distributionally robust optimization (DRO) contract to derive robust latency-reward contract bundles for teleoperators under the information asymmetry that the AIGC services quality is hidden information to teleoperators. The authors in [7] declared that the edge ASP's AIGC model complexity is hidden type information to teleoperators; they utilized contract theory to derive latency-reward contract bundles.

The aforementioned works are based on incentive mechanism design in AIGC networks under hidden type information asymmetry; the incentive mechanism design in AIGC networks under hidden action remains underexplored. The incentive mechanism design under hidden action is termed as *moral hazard*. The authors in [8]–[10] utilized contract theory to tackle it under various scenarios. The authors in [8] modeled the resource devoted by validators in blockchain as a moral hazard problem and utilized contract theory to design a bonus incentive mechanism. The authors in [9] considered the training effort exerted by federated learning (FL) clients as a hidden action to the FL server; they utilized contract theory to derive the optimal incentive mechanism. The authors in [10] stated that virtual service providers might reduce service quality by devoting less computational resources after receiving compensation from users; they also utilized contract theory to design a bonus incentive mechanism.

In this paper, we consider modeling the incentive mechanism design in AIGC networks under hidden action as a moral hazard problem. The hidden action in this paper indicates that

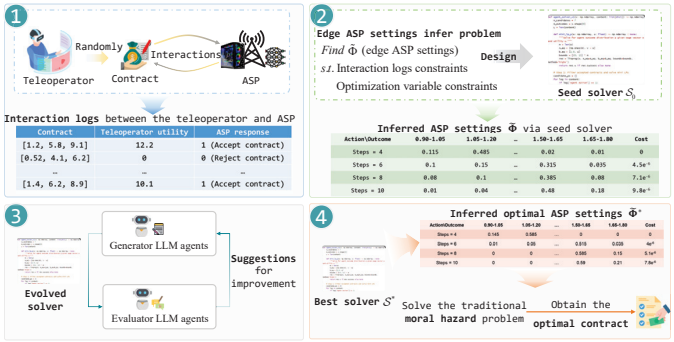

Fig. 1: Framework of LLM empowered online learning contract design.

the diffusion steps of the AIGC model undertaken by the edge ASP are concealed from the teleoperator. As depicted in the bottom table of Part 2 in Fig. 1, different diffusion steps of the AIGC model will render varying performance score received by the teleoperator. Regarding the moral hazard problem, previous works [8]–[10] assume the principal (teleoperators) can observe the settings of the agent (edge ASP) in advance. However, in practice, this assumption does not hold. We model the moral hazard problem as an online learning contract problem [11], [12]. In this way, we need to learn and infer the edge ASP's setting via multiple rounds of contract interaction between the teleoperator and edge ASP prior to tackling the moral hazard problem. The edge ASP's setting includes the mapping matrix between edge ASP's actions and teleoperator's outcomes, as well as the utility function of the edge ASP. Therefore, we formulate the research question in this paper as follows: *How can we design an effective incentive mechanism that maximizes the teleoperator utility under the hidden action of edge ASP?*

In light of the research question being APX-hard, which is difficult to solve via traditional solvers [13]–[15], we proposed to unleash the power of a large language model (LLM) to aid us in designing the incentive mechanism. We present the framework of our proposed LLM-empowered online learning contract design in Fig. 1, which includes four steps. Firstly, we need to collect the historical contract interaction logs between the teleoperator and the edge ASP. Secondly, we construct a seed solver for inferring the edge ASP's setting via contract interaction logs. Thirdly, we leverage the LLM to evolve the seed solver via generator LLM agents and evaluator LLM agents. Finally, we utilize the evolved seed solver to infer the near-optimal edge ASP setting, and we design the incentive mechanism upon it. In summary, we conclude the contributions of this paper as follows:

1. We propose a framework of LLM-empowered online learning contract design. In this framework, we utilize LLM generator agents and evaluator agents to evolve a seed solver for inferring the edge ASP's setting.
2. We extend the traditional moral hazard problem as an online learning contract design problem by eliminating the assumption that the principal (teleoperator) can observe the agent's (edge ASP) setting in advance.
3. We conduct numerous experiments to assess the effectiveness of our proposed method. The experimental results demonstrate that our proposed method can augment the teleoperator utility by around $5 \sim 40\%$ in comparison with benchmarks.

We organize the rest of this paper as follows: We present the system model and framework in Section II. We formulate the mathematic problem and introduce the LLM-empowered solution in Section III. We evaluate our proposed method with benchmarks in Section IV. Lastly, we conclude our paper in Section V.

II. System Model and Framework

A. System Model

1) Notations: We describe the online learning contract design problem with elements $\mathcal{C} = (\mathcal{D}, \mathcal{R}, \mathcal{O}, \mathcal{A}, \Phi, \mathbf{q})$ [15]. $\mathcal{D}$ indicates the historical interaction logs between the teleoperator and the edge ASP. We define $\mathcal{D} = \{(\mathbf{r}_k, \pi_T(\mathbf{r}_k), \mathbb{I}(\mathbf{r}_k))|k \in [K]\}$, in which $[K] = \{1, \cdots, k, \cdots, K\}$. $\mathbf{r}_k$ is the contract at k-th interaction log. We define it as $\mathbf{r}_k = \{r_m|m \in [M]\} \in \mathcal{R} \subset \mathbb{R}^M_{\geq 0}$ over the finite and discrete outcome space $\mathcal{O} = \{o_m|m \in [M]\}$. The edge ASP will select a diffusion step (action) a_n from the action space $\mathcal{A}$ as per $\mathbf{r}_k$, in which $\mathcal{A} = \{a_n|n \in [N]\}$. Action a_n leads to an AIGC service quality distribution $\mathbf{p}_n = p(\cdot|a_n)$ over $\mathcal{O}$ and incurs an additional processing cost $c_n \in \mathbb{R}_{\geq 0}$ to the edge ASP. Subsequently, the teleoperator perceives an outcome o_m and obtains the utility of $\pi_T(\mathbf{r}_k)$. Finally, given that the teleoperator cannot observe a_n, we utilize $\mathbb{I}(\mathbf{r}_k) = \{-1, 1\}$ to indicate whether the edge ASP rejects or accepts the contract $\mathbf{r}_k$. For clarity, we consider $\mathbf{P} \in \mathbb{R}^{N \times M}$ and $\mathbf{c} \in \mathbb{R}^N$. We assume o_m is an AIGC service quality range, defined as $o_m = [l_m, u_m)$. We use the median value of o_m, i.e., $\delta_m = (l_m + u_m)/2$, to design the utility function for the teleoperator. We utilize $\mathbf{q} = \{q_m|m \in [M]\}$ to represent the utility of teleoperator over $\mathcal{O}$. By referring to [6], [7], we set $q_m = ln(1 + \alpha\delta_m)$.

2) Utility Model of Edge ASP: We define the utility function of the edge ASP as:

$$\pi_A(a_i; \mathbf{r}, \Phi) = \mathbb{E}_{o_m \sim p(\cdot|a_n)}[r_m] - c_n + r_s - c_t. \quad (1)$$

Here, r_s is the AIGC services subscription fee, and c_t is the model training cost of the edge ASP. AIGC services subscription will bundle a default diffusion step a_1, and we set $c_1 = 0$.

In each contract interaction, the edge ASP selects the action $a^*(\mathbf{r}, \Phi)$ that maximizes its utility. We define the utility of the edge ASP in each contract interaction as:

$$\pi_A = \max_a \pi_A(a; \mathbf{r}, \Phi). \quad (2)$$

We term (2) as the incentive compatibility (IC) constraint. Additionally, the edge ASP will only accept the contract if

$$\pi_A \geq \Delta_s = r_s - c_t. \quad (3)$$

We refer to (3) as the incentive rationality (IR) constraint. We set $a^*(\mathbf{r}, \Phi) = a_1$ if the edge ASP rejects $\mathbf{r}$.

3) Utility Model of Teleoperator: Similarly, we model the utility function of the teleoperator as:

$$\pi_T(\mathbf{r}) = \mathbb{E}_{o_m \sim p(\cdot|a^*)}\left[q_m - r_m\right] - r_s, \tag{4}$$

where α is the coefficient mapping the AIGC services quality to the teleoperator's gain.

B. System Framework

We present the framework of LLM-empowered online learning contract in Fig. 1, which includes interaction logs generation, seed solver generation, seed solver evolution, and near-optimal contract design.

Interaction logs generation: The teleoperator randomly generates a contract and the edge ASP response to the contract. After multiple iterations, we can collect the interaction logs as shown in Part 1 of Fig. 1.

Seed solver generation: Upon interaction logs, the teleoperator designs a naive seed solver used to infer the edge ASP setting $\tilde{\Phi}$. We present the detailed process of seed solver design in Section III-B1.

Seed solver evolution: We use generator LLM agents and evaluator LLM agents to evolve the seed solver. We introduce the detailed seed solver evolution process in Section III-B2.

Near-optimal contract design: Upon the refined seed solver, we can infer a near-optimal edge ASP's setting. Next, we can derive the near-optimal contract by tackling the moral hazard problem via traditional convex optimization methods.

III. Problem Formulation and Solution

A. Problem Formulation

The optimization objective in this paper is maximizing the teleoperator utility via optimization contract $\mathbf{r}$, which is a moral hazard problem. However, given that the teleoperator cannot assess the edge ASP's setting Φ, we formulate a Φ approximation problem via $\mathcal{D}$ before solving the moral hazard problem and define it as follows:

$$\begin{aligned}
P1: \ find \quad & (\mathbf{X}, \tilde{\Phi}), && \text{(5a)}\\
s.t. \quad & \sum_{n=1}^{N} x_{n,k} = 1, \forall\ k, && \text{(5b)}\\
& \sum_{m=1}^{M} \tilde{p}_{n,m} = 1, \forall\ n, && \text{(5c)}\\
& \pi_T(\mathbf{r}_k) \le \tilde{\mathbf{p}}_n^T(\mathbf{q} - \mathbf{r}_k) - r_s + L(1 - x_{n,k}), \\
& \forall\ n, k, && \text{(5d)}\\
& \pi_T(\mathbf{r}_k) \ge \tilde{\mathbf{p}}_n^T(\mathbf{q} - \mathbf{r}_k) - r_s - L(1 - x_{n,k}), \\
& \forall\ n, k, && \text{(5e)}\\
& \pi_A(a_n; \mathbf{r}_k, \tilde{\Phi}) \ge \Delta_s - L(1 - x_{n,k}), \forall\ n, k, && \text{(5f)}\\
& \pi_A(a_n; \mathbf{r}_k, \tilde{\Phi}) \ge \pi_A(a_{n'}; \mathbf{r}_k) - L(1 - x_{n,k}), \\
& \forall\ n \ne n', \forall\ n, k, && \text{(5g)}\\
& \pi_A(a_n; \mathbf{r}_k) \le \Delta_s, \forall\ \mathbb{I}(\mathbf{r}_k) = 0, && \text{(5h)}\\
& x_{n,k} \in \{0, 1\}, \tilde{p}_{n,m} \in [0, 1], \tilde{c}_n \in [0, L]. && \text{(5i)}
\end{aligned}$$

Here, Eq. (5b) indicates the edge ASP selects action a_n at the k-th interaction round regarding the contract $\mathbf{r}_k$. Eq. (5c) is the probability constraint. Eqs. (5d) and (5e) is the utility matching constraint, indicating that the teleoperator utility under a_n, $\mathbf{r}_k$, and $\mathbf{p}_n$ should match the real teleoperator utility $\pi_T(\mathbf{r}_k)$. Eqs. (5f) and (5g) is the IR and IC constraints when edge ASP accepts $\mathbf{r}_k$. Eq. (5h) is the contract rejection constraint, which means the IR constraint is violated when the teleoperator formulates the contract $\mathbf{r}_k$. Eq. (5i) is the range constraint of variables $\mathbf{X}$, $\tilde{\mathbf{P}}$, and $\tilde{\mathbf{c}}$. L is a large constant used to aid the mathematical formulation.

Remarkably, we formulate P1 as a feasibility problem for two main reasons. First, the actual edge ASP's setting, Φ, is hidden from the teleoperator, so no metric or objective can effectively assess $\tilde{\Phi}$'s performance. Second, P1's formulation relies on $\mathcal{D}$, which does not guarantee we will find the actual setting Φ. We still refine $\tilde{\Phi}$ based on later interaction logs between the teleoperator and the edge ASP.

After solving P1, we can acquire an approximated edge ASP's setting $\tilde{\Phi}$. Subsequently, we can design the optimization problem as a standard moral hazard problem with $\tilde{\Phi}$.

$$\begin{aligned}
P2: \ \max_{\mathbf{r}} \quad & \mathbb{E}_{o_m \sim p(\cdot|a^*)}\left[q_m - r_m\right] - r_s, && \text{(6a)}\\
s.t. \quad & a^*(\mathbf{r}, \tilde{\Phi}) = \arg\max_{a} \pi_A(a; \mathbf{r}, \tilde{\Phi}) && \text{(6b)}\\
& \pi_A(\mathbf{r}, \tilde{\Phi}) \ge \Delta_s. && \text{(6c)}
\end{aligned}$$

B. LLM-Empowered Solution

P2 is a nonlinear convex optimization problem. If we can solve P1 efficiently, it can be solved with ease. However, P1 is a feasibility problem; the constraints form it as a mixed integer nonlinear programming (MINLP) problem, and the size of the optimization variable is unknown. It is difficult for us to directly solve P1 and obtain $\tilde{\Phi}$. Inspired by recent advances in utilizing LLM to solve the NP-hard problems [16]–[18], we consider utilizing the LLM to evolve an effective P1 solver. Our proposed LLM-empowered P1 solution consists of two stages: P1 solver initialization and P1 solver evolution, which we introduce in Sections III-B1 and III-B2, respectively.

1) P1 Solver Initialization: We consider naively constructing the P1 seed solver and present it in Algorithm 1. Notably, we define the P3 in line 6 as:

$$\begin{aligned}
P3: \ \min_{\mathbf{p}_k} \quad & {\mathbf{p}_k}^T \mathbf{r}_k, && \text{(7a)}\\
s.t. \quad & \sum_{m=1}^{M} p_{k,m} = 1, && \text{(7b)}\\
& {\mathbf{p}_k}^T(\mathbf{q} - \mathbf{r}_k) - r_s = \pi_T(\mathbf{r}_k), && \text{(7c)}\\
& p_{k,m} \in [0, 1]. && \text{(7d)}
\end{aligned}$$

Eq. (7b) is the probability constraint, which stems from Eq. (5c). Eq. (7c) is the teleoperator utility matching constraint, stems from (5d) and (5e). Eq. (7d) is the range constraint, partial of (5i). We designate the P1 seed solver as $\mathcal{S}_0$.

Upon the P1 seed solver $\mathcal{S}_0$, we design the initialization prompt and utilize an LLM generator to generate N_i P1

Algorithm 1: P1 Seed Solver

Input: Teleoperator's profit regarding outcomes $\mathbf{q}$ and historical interaction logs $\mathcal{D}$
Output: Inferred edge ASP's setting $\tilde{\Phi} = (\tilde{\mathbf{P}}, \tilde{\mathbf{c}})$

Assume the size of action space $N = \tilde{N}$
Initialize candidate probability vector set Ψ_p
Initialize edge ASP's cost vector $\tilde{\mathbf{c}} = 0$
foreach $(\mathbf{r}_k, \pi_T(\mathbf{r}_k), \mathbb{I}(\mathbf{r}_k)) \in \mathcal{D}$ **do**
  if $\mathbb{I}(\mathbf{r}_k) == 1$ **then**
    $\mathbf{p}_k \leftarrow$ Solve P3 via $\mathbf{r}_k$ and $\pi_T(\mathbf{r}_k)$
    $\Psi_p \leftarrow \mathbf{p}_k$
  end
end
$\tilde{\mathbf{P}} \leftarrow KMeans(\tilde{N}, \Psi_p)$
foreach $(\mathbf{r}_k, \pi_T(\mathbf{r}_k), \mathbb{I}(\mathbf{r}_k)) \in \mathcal{D}$ **do**
  if $\mathbb{I}(\mathbf{r}_k) == 1$ **then**
    $n_k \leftarrow argmax(\tilde{\mathbf{P}}\mathbf{r}_k)$
    if $\tilde{\mathbf{c}}[n_k] == 0$ **then**
      $\tilde{\mathbf{c}}[n_k] = \tilde{\mathbf{P}}[n_k]\mathbf{r}_k$
    end
    else
      $\tilde{\mathbf{c}}[n_k] = min(\tilde{\mathbf{c}}[n_k], \tilde{\mathbf{P}}[n_k]\mathbf{r}_k)$
    end
  end
  else
    $\tilde{\mathbf{c}} = max(\tilde{\mathbf{c}}, \tilde{\mathbf{P}}\mathbf{r}_k)$
  end
end
return $\tilde{\Phi} = (\tilde{\mathbf{P}}, \tilde{\mathbf{c}})$

Algorithm 2: LLM-Empowered P1 Solver Evolution

Input: Seed solver $\mathcal{S}_0$ and iteration rounds I
Output: Elitist P1 solver $\mathcal{S}^*$

/* Initialize and evaluate P1 solvers */
$\{\mathcal{S}_{n_i} | n_i \in [N_i]\} \leftarrow GeneratorLLM(\mathcal{S}_0, Init\ prompt)$
$\{\tilde{\Phi}_{n_i} | n_i \in [N_i]\} \leftarrow$ Invoke $\{\mathcal{S}_{n_i} | n_i \in [N_i]\}$
$\{\pi_T(\tilde{\mathbf{r}}_{n_i}) | n_i \in [N_i]\} \leftarrow$ Tackle P3 via $\tilde{\Phi}_{n_i}$ and evaluate $\tilde{\mathbf{r}}_{n_i}$
$\Omega_o \leftarrow \{\mathcal{S}_{n_i} | n_i \in [N_i]\}, i = N_i$
$\mathcal{S}^* \leftarrow$ Opt the best P1 solver from Ω_o
while $i \leq I$ **do**
  /* Begin the epoch of P1 solver evolution */
  $\Omega_n \leftarrow \emptyset$
  Rank $\mathcal{S}_s \in \Omega_o$ in a descending order as per $\pi_T(\tilde{\mathbf{r}}_s)$
  foreach $n_s \in [N_s/2]$ **do**
    Opt $\mathcal{S}_b$ and $\mathcal{S}_w$ from Ω_o with probability proportional to their rank
    $\Omega_n \leftarrow \{\mathcal{S}_b, \mathcal{S}_w\}$
    $\theta_{n_s} \leftarrow ShortReflectorLLM(\mathcal{S}_b, \mathcal{S}_w)$
    $\mathcal{S}_{n_s} \leftarrow CrossoverLLM(\mathcal{S}_b, \mathcal{S}_w, \theta_{n_s})$
    $\pi_T(\tilde{\mathbf{r}}_{n_s}) \leftarrow$ Invoke $\mathcal{S}_{n_s}$, tackle P3, and evaluate $\tilde{\mathbf{r}}_{n_s}$
    $\Omega_n \leftarrow \{\mathcal{S}_{n_s}\}$
    $i+ = 1$
  end
  $\Theta \leftarrow ShortReflectorLLM(\{\theta_{n_s} | n_s \in [N_s/2]\}, \Theta')$
  foreach $n_m \in [N_m]$ **do**
    $\mathcal{S}_{n_m} \leftarrow MutationLLM(\Theta, \mathcal{S}^*)$
    $\pi_T(\tilde{\mathbf{r}}_{n_m}) \leftarrow$ Invoke $\mathcal{S}_{n_m}$, tackle P3, and evaluate $\tilde{\mathbf{r}}_{n_m}$
    $\Omega_n \leftarrow \{\mathcal{S}_{n_m}\}$
    $i+ = 1$
  end
  $\mathcal{S}^* \leftarrow$ Opt the best P1 solver from $\Omega_n \cup \{\mathcal{S}^*\}$
  $\Omega_o \leftarrow \Omega_n$
end
return $\mathcal{S}^*$

solvers, which are used to evolve the better P1 solver in the next stage. The initialization prompt includes problem description, function description, input instance, and seed function (P1 seed solver).

Problem Description: We instruct the LLM agent to deduce a valid edge ASP setting that aligns with all interaction logs in $\mathcal{D}$ for an online learning contract design problem.

Function Description: We specify the input format for the P1 solver and outline its requirements, detailing the structure of a valid $\tilde{\Phi}$.

Input Instance: We illustrate the input of the P1 solver, i.e., $\mathbf{q}$ and $\mathcal{D}$.

Seed Function: We provide a runnable Python code, i.e., Algorithm 1, to the LLM agent for reference.

Lastly, we need to evaluate N_i initialized P1 solvers, $\{\mathcal{S}_{n_i} | n_i \in [N_i]\}$, before proceeding to the next stage. Specifically, we can acquire an edge ASP's setting $\tilde{\Phi}_{n_i}$ via $\mathcal{S}_{n_i}$, and then solve P1. Therefore, we utilize the teleoperator utility under $\mathcal{S}_{n_i}$ as its fitness score.

2) P1 Solver Evolution: The P1 solver evolution includes the following five steps: premise, short-term reflection, crossover, long-term reflection, and mutation. Notably, the P1 solver evolves over I iterations and outputs the elitist P1 solver $\mathcal{S}^*$ as the final output. We define an epoch as including a complete execution of the five-step process, encompassing multiple iterations.

Premise Step: At the beginning of each epoch, we sort all solvers in the old population Ω_o by fitness score and iteratively select two P1 solvers, $\mathcal{S}_b$ and $\mathcal{S}_w$, with probability proportional to their rank. Here, $\mathcal{S}_b$ and $\mathcal{S}_w$ denote the better and worse P1 solvers, respectively. Both are appended to the new population Ω_n, and the selection process continues until Ω_n reaches size N_s. Notably, we ensure that $\mathcal{S}_b$ and $\mathcal{S}_w$ have distinct fitness scores to facilitate the generation of the "short verbal gradient"

in the subsequent step.

Short-Term Reflection: For each pair $(\mathcal{S}_b, \mathcal{S}_w)$ selected in the previous step, we employ a short-term reflector LLM agent to generate a "short verbal gradient" to guide P1 solver refinement. Specifically, the reflector LLM compares $\mathcal{S}_b$ and $\mathcal{S}_w$ and outputs a concise insight, limited to fewer than 20 words, denoted as θ_{n_s}. For example: "*Incorporate LP for p inference, adaptive clustering with silhouette, simplex projection, and strict IR/IC cost adjustments.*" Since N_s P1 solvers are selected, we obtain $N_s/2$ short verbal gradients, denoted as $\{\theta_{n_s} \mid n_s \in [N_s/2]\}$.

Crossover: Given $\mathcal{S}b$, $\mathcal{S}w$, and the corresponding "short verbal gradient" θ_{n_s}, the crossover LLM agent generates an offspring $\mathcal{S}n_s$. This crossover step produces $N_s/2$ offspring, denoted as $\{\mathcal{S}_{n_s} \mid n_s \in [N_s/2]\}$. After evaluating their fitness scores, these offspring are appended to Ω_n.

Long-Term Reflection: After generating $\{\theta_{n_s} \mid n_s \in [N_s/2]\}$, the long-term reflector LLM agent aggregates these short verbal gradients with the previous long verbal gradient Θ' to form the current long verbal gradient Θ. For example, we present a long verbal gradient as follows: "*Combine adaptive density-based clustering (e.g., DBSCAN) on inferred distributions with LP-enforced strict IR/IC constraints. Refine costs upward using rejection margins, normalize outcome probabilities, and prioritize feasibility. Employ silhouette scores and hierarchical methods to ensure robust, realistic agent settings matching all logs.*"

Mutation: The mutation LLM agent utilizes $\mathcal{S}^*$ and Θ to generate N_m mutated P1 solvers, denoted as $\{\mathcal{S}_{n_m} \mid n_m \in [N_m]\}$. After assessing their fitness score, we append them to Ω_n. Lastly, we update the elitist P1 solver $\mathcal{S}^*$, set the old population $\Omega_o = \Omega_n$, and clean Ω_n before proceeding to the next epoch. For clarity, we present the overall process of P1 solver evolution in Algorithm 2.

Upon $\mathcal{S}^*$, we can acquire the near-optimal edge ASP's setting $\tilde{\Phi}^*$. We can use it to solve P2 via convex optimization methods and derive the near-optimal contract, thereby addressing the research question proposed in Section I.

IV. Experimental Evaluation

A. Configurations and Benchmarks

We set the number of historical interaction logs to $|\mathcal{D}| = K = 100$, with outcome space sizes $M = \{2, 4, 6, 8, 10, 12\}$ and edge ASP action space sizes $N = \{2, 3, 4, 5, 6, 7\}$. Each outcome o_m is defined as an even interval within $[0.9, 1.8]$; for example, with $M = 2$, we have $\{o_m \mid m \in [2]\} = \{[0.9, 1.35), [1.35, 1.8)\}$. The ASP's action a_n corresponds to the diffusion step of the AIGC model, where $a_n \in \{4, 5, 6, 7, 8, 9, 10\}$. The AIGC model, implemented as a conditional diffusion model, follows the configuration in [3]. For the ASP's setting $\Phi = (\mathbf{P}, \mathbf{c})$, we use the statistical distribution from 200 AIGC tasks [3], [6] under each diffusion step a_n for $\mathbf{p}_n$. The additional computational energy costs are set as $\mathbf{c} = \{0, 1.5^{-6}, 4^{-6}, 5.6^{-6}, 5.1^{-6}, 8.1^{-6}, 7.8^{-6}\}$ per image, based on electricity data from the Energy Information Administration (EIA) and CodeCarbon. The subscription fee per image is $r_s = 1.6^{-4}$, following Anthropic's pricing policy. Assuming a net profit margin of approximately 30%, the training cost allocation per image is $c_t = 1.2^{-4}$. The mapping coefficient from AIGC service quality to teleoperator utility is set as $\alpha \in \{5^{-4}, 1^{-3}, 5^{-3}\}$. All LLM agents used are instances of gpt-4.1-mini-2025-04-14, and the number of default iteration rounds is $I = 200$.

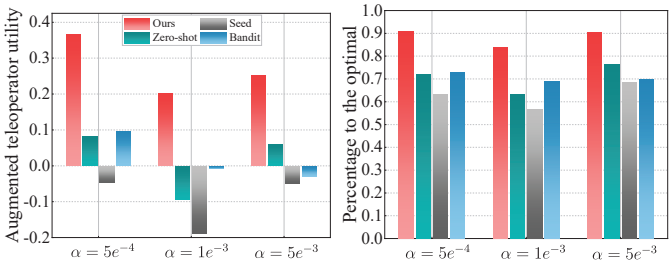

Fig. 2: Comparison results by varying α with $M = 12$ and $N = 7$.

To evaluate the effectiveness of our proposed method, we compare it with the following benchmarks:

1) **Seed**: The seed solver (Algorithm 1) is used to infer $\tilde{\Phi}$, followed by contract derivation.
2) **Zero-shot**: Our framework is run in a simulated scenario, and the resulting $\hat{\mathcal{S}}^*$ serves as the P1 solver for the teleoperator.
3) **Bandit**: Following [11], a multi-armed bandit approach is employed for direct contract derivation.

In this paper, we evaluate our proposed solution using two metrics: augmented teleoperator utility $\pi_T^{\%}$ and percentage of the optimal η. In the absence of a bonus incentive mechanism (contract), the edge ASP defaults to $a_1 = 4$ diffusion steps. Accordingly, we define $\pi_T^{\%}$ as:

$$\pi_T^{\%} = (\pi_T(\tilde{\mathbf{r}}) - \pi_T(\mathbf{0}))/\pi_T(\mathbf{0}). \tag{8}$$

For the metric of percentage to the optimal, η, we define it as:

$$\eta = \pi_T(\tilde{\mathbf{r}})/\pi_T(\mathbf{r}^*), \tag{9}$$

where $\mathbf{r}^*$ denotes the optimal contract, assuming the teleoperator has complete knowledge of the edge ASP's setting Φ.

B. Effectiveness of Proposed Method

In this section, we evaluate the effectiveness of our proposed method by comparing it with benchmarks under varying settings of α, M, and N.

Observing Fig. 2, our proposed method consistently surpasses the benchmarks in terms of $\pi_T^{\%}$ under varying α, which demonstrates the effectiveness of our proposed method. Compared to the seed solver, our proposed method can augment $\pi_T^{\%}$ by around 40%, further demonstrating the effectiveness of our proposed method. Compared with the traditional bandit algorithm, our proposed method can augment $\pi_T^{\%}$ around $20 \sim 25\%$. Moreover, in terms of the metric of η, our proposed method outperforms the benchmarks and maintains η at around 90% under varying α.

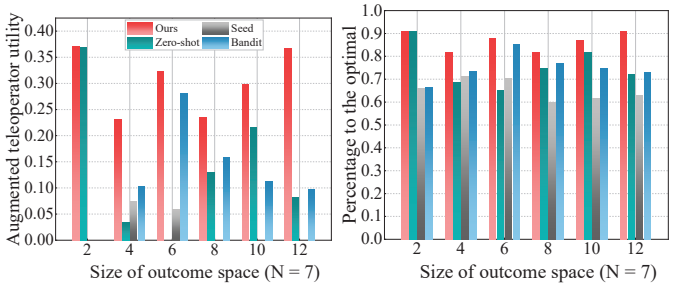

Fig. 3: Comparison results by varying M with $\alpha = 5e^{-4}$ and $N = 7$.

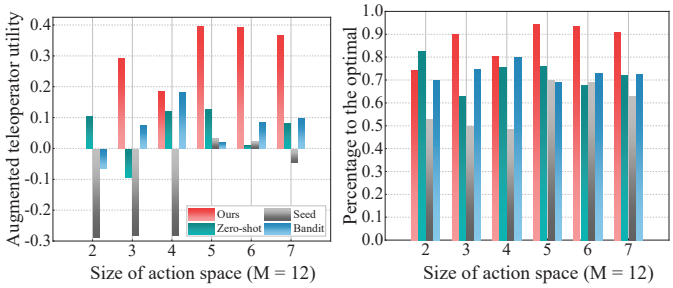

Fig. 4: Comparison results by varying N with $\alpha = 5e^{-4}$ and $M = 12$.

The results presented in Fig. 3 are analogous to Fig. 2. Concretely, observing Fig. 3, our proposed method can augment $\pi_T^{\%}$ in around $16 \sim 37\%$ in comparison with the seed solver under varying M. Comparing with the bandit algorithm, our proposed method can improve $\pi_T^{\%}$ in around $5 \sim 36\%$ under varying M. Notably, the zero-shot method can achieve the same $\pi_T^{\%}$ as our proposed method when $M = 2$. Similarly, we observed that our proposed method maintains η at around 90% under varying M.

Observing Fig. 4, one exception occurs, where our proposed method is inferior to the zero-shot when $N = 2$ and only achieves the second-best performance in $\pi_T^{\%}$. However, our proposed method can enlarge the improvement in $\pi_T^{\%}$ to around $35 \sim 58\%$ in comparison with the seed solver under varying N. In comparison with the bandit algorithm, our proposed method can improve $\pi_T^{\%}$ by at most 38%. Our proposed method can guarantee η above 90% under the setting of $N = \{3, 5, 6, 7\}$, which are align with the results in Figs. 2 and 3.

Moreover, to further assess our proposed method, we present the training time, token consumption, and monetary cost of our proposed method under different LLM models in Table I. The LLM models used in Table I are OpenAI (GPT-4.1-mini-2025-04-14), DeepSeek (DeepSeek-V3.1), Gemini (Gemini-2.0-flash), Ollama (Llama-3.3-70B-Instruct-Turbo), and Qwen (Qwen3-Coder-480B-A35B-Instruct-FP8), respectively. The results depicted that the training time (from 13.8 to 30.8 minutes) and monetary overhead (from \$0.78 to \$2.27) of the proposed method are affordable under different LLM models.

TABLE I: Comparison results among different LLM models

	OpenAI	DeepSeek	Gemini	Ollama	Qwen
Training time (min)	28.5	16.2	13.8	15.4	30.8
Token consumption	1.62M	0.92M	0.95M	0.98M	1.13M
Monetary cost (\$)	1.12	0.78	1.39	0.86	2.27

C. Scalability of Proposed Method

In light of the action space available to the edge AIGC service provider (ASP) is relatively small, it cannot adequately reflect the scalability of our proposed method. Therefore, following the setup in [15], we construct a synthetic online learning contract problem to evaluate scalability. Specifically, we generate the outcome distributions $\mathbf{p}_n$ for each action a_n by applying the SoftMax operator to a Gaussian random vector in $\mathbb{R}^M$. The outcome values q_m are sampled uniformly from $[0, 10]$. The action cost is defined as a mixture $c_n = (1 - \beta_p)c_r(a_n) + \beta_p c_i(a_n)$, where $c_r(a_n) = \beta_c \mathbb{E}_{o_m \sim \mathbf{p}_n}[q_m]$ represents a correlated cost proportional to the expected outcome, and $c_i(a_n)$ is an independent cost sampled uniformly from $[0, 1]$. Unless otherwise specified, we set $\beta_c = 0.7$ and $\beta_p = 0.3$. To assess scalability, we conduct experiments under varying sizes of the outcome space $M \in \{2, 3, 4, 5, 10, 20\}$, the action space $N \in \{10, 100, 1000\}$, and the number of historical interaction logs $K \in \{25, 50, 100\}$.

Since the purpose of this synthetic dataset is to evaluate the scalability of the proposed method, we focus exclusively on the performance metric η and report the results in Fig. 5. Observing Figs. 5a, 5b, and 5c, we find that the performance of our method improves steadily as the number of interaction logs increases. At the same time, larger outcome spaces (M) and action spaces (N) introduce additional complexity, leading to a gradual performance decline.

Remarkably, Fig. 5c shows that when $K = 100$, the proposed method consistently maintains $\eta \approx 75\%$ across all tested values of M and N. Furthermore, when $M = 2$ and $K \in \{50, 100\}$, the method achieves $\eta \approx 95\%$ even as the size of the action space increases. These results demonstrate that our method possesses strong scalability with respect to the action space size N, even under a limited number of interaction rounds. Furthermore, because the underlying optimization structure remains consistent across problem settings, these findings can be readily transferred to other domains and scenarios.

V. Conclusion

In this paper, we proposed an LLM-empowered online learning contract design for maximizing the teleoperator's utility under the hidden action of edge ASP. Specifically, given that the teleoperator cannot access the edge ASP's setting, we modeled the moral hazard as an online learning contract design problem. Next, we proposed an LLM-empowered P1 solver to approximate the edge ASP setting. Lastly, we applied the approximated edge ASP setting to solve the moral hazard problem and derive a near-optimal contract. Compared to the Bandit and Seed benchmarks, we boosted teleoperator utility by $5 \sim 40\%$. In most settings, we achieved an efficiency

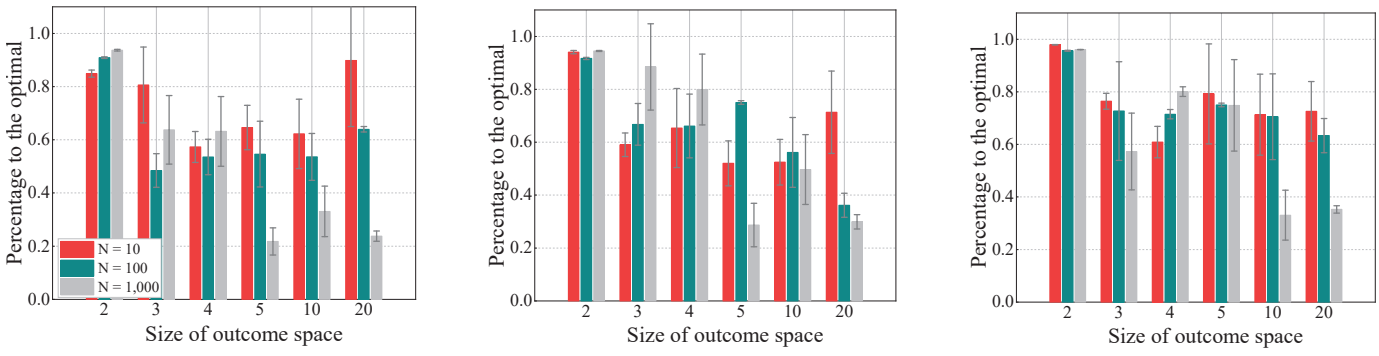

Fig. 5: Evaluate the scalability of the proposed method by varying the size of the outcome space M, the number of actions N, and the number of historical interaction logs K, in which $K = \{25, 50, 100\}$ in Figs. 5a, 5b, and 5c.

rate of $\eta \geq 90\%$, thereby demonstrating the robustness of our proposed method. Additionally, the training time and monetary overhead of our proposed method are affordable, with a maximum training time and cost of 30.8 minutes and $2.27. We also demonstrated the scalability of our proposed method in a synthetic dataset under the maximum size of outcome and action space of 20 and 1000.

Acknowledgment

This research is funded by the U.S. National Science Foundation (NSF) via Grants 2222730. In addition, this work is partially supported by NSF ECCS-2302469, Amazon, and Japan Science and Technology Agency (JST), Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE) JPMJAP2326.

References

[1] X. Chen, Z. Hu, and C. Wang, "Empowering education development through aigc: A systematic literature review," *Educ. Inf. Technol.*, vol. 29, no. 13, pp. 17 485–17 537, Feb. 2024.

[2] J. Chen, C. Yi, H. Du, D. Niyato, J. Kang, J. Cai, and X. Shen, "A revolution of personalized healthcare: Enabling human digital twin with mobile aigc," *IEEE Netw.*, vol. 38, no. 6, pp. 234–242, Nov. 2024.

[3] 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," *IEEE Trans. Mob. Comput.*, vol. 24, no. 8, pp. 7742–7756, Aug. 2025.

[4] S. B. Kamtam, Q. Lu, F. Bouali, O. C. Haas, and S. Birrell, "Network latency in teleoperation of connected and autonomous vehicles: A review of trends, challenges, and mitigation strategies," *Sensors*, vol. 24, no. 12, p. 3957, Jun. 2024.

[5] J. Li, D. Niyato, and Z. Han, *Cryptoeconomics: Economic Mechanisms behind Blockchains*. Cambridge University Press, 2023.

[6] Z. Zhan, Y. Dong, D. M. Doe, Y. Hu, S. Li, S. Cao, L. Fan, and Z. Han, "Distributionally robust contract theory for edge aigc services in teleoperation," *arXiv preprint arXiv:2505.06678*, May. 2025.

[7] J. Wen, J. Nie, Y. Zhong, C. Yi, X. Li, J. Jin, Y. Zhang, and D. Niyato, "Diffusion model-based incentive mechanism with prospect theory for edge aigc services in 6g iot," *IEEE Internet Things J.*, vol. 11, no. 21, pp. 34 187–34 201, Nov. 2024.

[8] J. Li, T. Liu, D. Niyato, P. Wang, J. Li, and Z. Han, "Contract-theoretic pricing for security deposits in sharded blockchain with internet of things (iot)," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 10 052–10 070, Jun. 2021.

[9] M. Tian, Y. Chen, Y. Liu, Z. Xiong, C. Leung, and C. Miao, "A contract theory based incentive mechanism for federated learning," *arXiv preprint arXiv:2108.05568*, Aug. 2021.

[10] L. Ismail, M. Qaraqe, A. Ghrayeb, and D. Niyato, "Enhancing trust and security in the vehicular metaverse: A reputation-based mechanism for participants with moral hazard," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Dubai, United Arab Emirates, Apr 2024.

[11] C.-J. Ho, A. Slivkins, and J. W. Vaughan, "Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems," in *Proceedings of the fifteenth ACM conference on Economics and computation*, Palo Alto, CA, Jun 2014.

[12] P. Dütting, T. Roughgarden, and I. Talgam-Cohen, "Simple versus optimal contracts," in *Proceedings of the ACM Conference on Economics and Computation*, Phoenix, AZ, Jun 2019.

[13] F. Bacchiocchi, M. Castiglioni, A. Marchesi, and N. Gatti, "Learning optimal contracts: How to exploit small action spaces," in *International Conference on Learning Representations (ICLR)*, Vienna, Austria, May 2024.

[14] G. Guruganesh, Y. Kolumbus, J. Schneider, I. Talgam-Cohen, E.-V. Vlatakis-Gkaragkounis, J. Wang, and S. Weinberg, "Contracting with a learning agent," in *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, Dec 2024.

[15] T. Wang, P. Dütting, D. Ivanov, I. Talgam-Cohen, and D. C. Parkes, "Deep contract design via discontinuous networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, Dec 2023.

[16] F. Liu, X. Tong, M. Yuan, X. Lin, F. Luo, Z. Wang, Z. Lu, and Q. Zhang, "Evolution of heuristics: Towards efficient automatic algorithm design using large language model," in *Proceedings of the 41st International Conference on Machine Learning (ICML)*, Vienna, Austria, Jul 2024.

[17] H. Ye, J. Wang, Z. Cao, F. Berto, C. Hua, H. Kim, J. Park, and G. Song, "Reevo: Large language models as hyper-heuristics with reflective evolution," in *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, Dec 2024.

[18] S. Yao, F. Liu, X. Lin, Z. Lu, Z. Wang, and Q. Zhang, "Multi-objective evolution of heuristic using large language model," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Philadelphia, PA, Feb-Mar 2025.