PAY FOR THE SECOND-BEST SERVICE: A GAME-THEORETIC APPROACH AGAINST DISHONEST LLM PROVIDERS

Yuhan Cao[1,4]*, **Yu Wang**[5,1]*, **Sitong Liu**[4], **Miao Li**[6], **Yixin Tao**[7], **Tianxing He**[1,2,3]†
[1]Shanghai Qi Zhi Institute, [2]Tsinghua University,
[3]Xiongan AI Institute, [4]ShanghaiTech University,
[5]Institute of Information Engineering, Chinese Academy of Sciences
[6]Beijing Institure for General Artificial Intelligence, BIGAI
[7]Shanghai University of Finance and Economics

ABSTRACT

The widespread adoption of Large Language Models (LLMs) through Application Programming Interfaces (APIs) induces a critical vulnerability: the potential for dishonest manipulation by service providers. This manipulation can manifest in various forms, such as secretly substituting a proclaimed high-performance model with a low-cost alternative, or inflating responses with meaningless tokens to increase billing. This work tackles the issue through the lens of algorithmic game theory and mechanism design. We are the first to propose a formal economic model for a realistic user-provider ecosystem, where a user can iteratively delegate T queries to multiple model providers, and providers can engage in a range of strategic behaviors. As our central contribution, we prove that for a continuous strategy space and any $\epsilon \in (0, \frac{1}{2})$, there exists an approximate incentive-compatible mechanism with an additive approximation ratio of $O(T^{1-\epsilon} \log T)$, and a guaranteed quasi-linear second-best user utility. We also prove an impossibility result, stating that no mechanism can guarantee an expected user utility that is asymptotically better than our mechanism. Furthermore, we demonstrate the effectiveness of our mechanism in simulation experiments with real-world API settings.

1 INTRODUCTION

In recent years, large language models (LLMs) have demonstrated capabilities across a multitude of domains (Thirunavukarasu et al., 2023; Hong et al., 2024; Balunović et al., 2025). Following the release of ChatGPT (OpenAI, 2022), there has been a rapid and widespread adoption of LLMs among the general public, embedding them into a variety of commonplace activities. The inference processes of these models are predominantly controlled by companies, which provide access to individual users through interactive interfaces and Application Programming Interfaces (APIs). The growing user base has also triggered the development of a secondary market (e.g., TogetherAI (TogetherAI, 2023)), where providers aggregate resources to offer more affordable services.

This market expansion introduces a critical vulnerability: the potential for service provider dishonesty. Since users interact with LLMs in a "black-box" manner, a clear economic incentive exists for providers to substitute the advertised models with less capable alternatives to reduce their own operational costs. Such deceptive practices can manifest in various forms. For instance, when user is asking for a large model like Qwen3-235B-A22B (Yang et al., 2025), a provider might secretly switch to a smaller one, such as Qwen3-30B-A3B. It might also deploy a quantized version (e.g., using INT8 or FP8) (Gao et al., 2025). Additionally, providers might dynamically adjust a model's reasoning effort (as in reasoning models like OpenAI o1(OpenAI, 2024a)) without user consent.

*Equal contribution.
†Corresponding author.

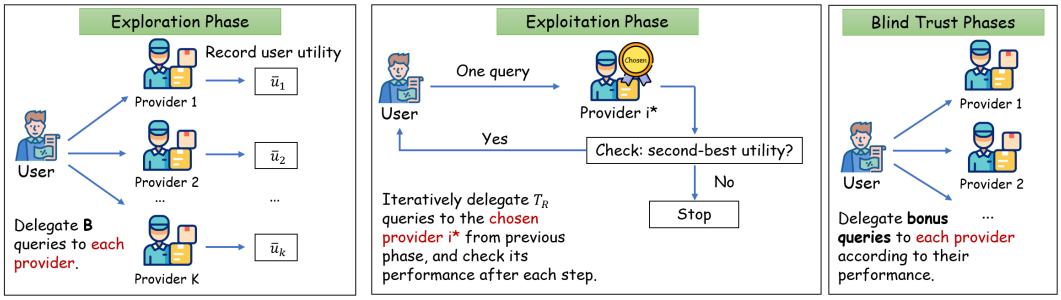

Figure 1: An illustration figure of our proposed mechanism, in which the provider's optimal strategy would guarantee the user a second-best overall utility. The mechanism splits the user's queries among four sequential phases. In the first phase (exploration), a batch of sample outputs from each service provider is collected. They need to compete to get selected in the second phase (exploitation), where the best-performing provider is asked to deliver the second-best user utility for the majority of queries. This is followed by the final two blind trust phases (detailed in Section 3), which serve as incentives for providers to behave well in the first two phases.

The pay-per-token API billing also introduces new risks, allowing providers to inflate charges by inserting superfluous tokens that are difficult for users to detect (Kumar et al., 2025).

These dishonest behaviors have profound negative implications. Firstly, they erode the trustworthiness of the service and promote vicious competition within the market. Secondly, for researchers dependent on APIs for scientific investigation, undisclosed behaviors severely undermine experimental reproducibility.

Existing research has approached this challenge from a technical detection standpoint. Efforts include auditing model outputs for semantic behavioral drift (Chen et al., 2023; Eyuboglu et al., 2024) or token-level consistency (Miao & Fang, 2025; Sun et al., 2025a), and examining the inherent discrepancy across different models (Gao et al., 2025; Sun et al., 2025b; Cai et al., 2025). In this work, we tackle the problem through the lens of algorithmic game theory and computational economics. In other words, we propose a mechanism in which the providers would compete with each other, thus protecting user utility.

1.1 Our Contributions

The User-Provider Delegation Game Model. We model an online user-provider delegation game within a market consisting of a single user and multiple candidate providers, each declaring a specific LLM for service. The user has T sequential queries. For each query, the user selects and delegates it to one provider. Upon accepting the delegation, a strategic provider could take action by choosing a model with a different actual cost or reporting an inflated output token sequence. After receiving the token sequence and making a payment, the user evaluates the utility gained, which informs their future delegation decisions.

We formulate a mechanism design problem in which providers determine their strategies based on the user's delegation mechanism. We characterize the providers' action space, which comprises cost and token sequence reporting. Our proposed model, which draws upon both mechanism design theory (Nisan & Ronen, 1999) and contract theory (Dütting et al., 2025), addresses a novel problem that has not been extensively studied in prior literature.

To better study the relationship between provider actions and outcomes, we also model the performance of LLMs. This specifies how cost control affects the expected reward and output sequence length.

Impossibility Result and Utility Guarantee Mechanism. We provide a theoretical guarantee under the continuous cost control model. Inspired by (Braverman et al., 2019) in the online learning domain, we first propose an impossibility theorem, proving that there is no $o(T)$-approximately

incentive compatible mechanism that can guarantee a first-best user utility (formally defined later) asymptotically. Based on this, our objective is to propose a mechanism that guarantees a utility close to the second-best user utility.

Concretely, for any $\epsilon \in (0, \frac{1}{2})$, we propose a mechanism that is $O(T^{1-\epsilon} \log T)$-approximately incentive compatible. This mechanism consists of four phases: an exploration phase, during which a batch of "test samples" from each service provider are collected; an exploitation phase, where the best-performing provider is selected and asked to deliver the second-best user utility for the majority of queries throughout the mechanism; this is followed by two subsequent blind trust phases, which serve as incentives for providers to behave well in the first two phases. This mechanism guarantees a utility of $u_{SB} - O(T^{1-\epsilon} \log T + T^{2\epsilon})$, where u_{SB} represents the second-best user utility.

Simulation experiments with real-world API settings. Furthermore, we conduct experiments in simulated environments with real-world API price and performance settings to demonstrate the effectiveness of our mechanism. We designed a total of six strategies, including our own proposed strategy for providers. Through experimental evaluation, we demonstrate that our proposed strategy achieves the highest utility for the provider among all tested strategies, while also delivering considerable utility to the user.

2 THE MODEL

2.1 THE USER-PROVIDER DELEGATION GAME

We model the interaction between the user and the service providers as a repeated Stackelberg game. The user acts as the principal, who commits to and announces a delegation mechanism at the beginning. The K service providers act as agents who observe the mechanism and strategically respond to maximize their own utility over T consecutive queries. Our goal is to design a mechanism in which the user is guaranteed to get the "second-best" service.

Model Components and Interaction Protocol. The game unfolds over T discrete time units. In each time unit $t \in [T]$, the user has a query prompt ω_t sampled from a dataset D. The user selects one provider, denoted by $d_t \in \{1, \ldots, K\}$, to handle the query.

Each provider i is characterized by a public price-per-token $p_i \in \mathbb{R}_+$. Upon being selected, provider d_t could secretly switch LLMs, leading to an actual cost-per-token[1] $c_t \in [0, p_{d_t}]$. This chosen cost leads to the generation of a cost-dependent output token sequence, τ_t. In addition, the provider may report a different, potentially manipulated sequence, τ'_t, to the user. All other providers $i \neq d_t$ do not take any action.

The user receives the reported sequence τ'_t and makes a payment of $p_{d_t} \cdot |\tau'_t|$. The user then evaluates the received response, obtaining a reward $v_t \in [0, R]$, where R is a known upper bound of reward. The delegation choice becomes public knowledge and is added to the history of the game.

All possible output token sequences are assumed to have a length bounded by a constant L.

Remark 1. *For simplicity, we ignore the price and cost associated with processing the input prompt. As the prompt is user-provided, its length cannot be misreported, and any strategic cost control for processing it can be modeled similarly to the cost control for generating the output.*

Provider's Strategy Space and Truthful Behavior. A dishonest provider can deviate from the *truthful* behavior to increase its utility. We consider two main types of strategies:

1. Cost control: For query t, the provider d_t can secretly set $c_t \in [0, p_{d_t}]$ to a value other than the truthful one. For example, the provider could choose a cheaper, lower-quality model or sub-model to generate the response, or resort to quantization (Gong et al., 2024).
2. Token sequence reporting: For query t, the provider d_t can report a token sequence τ'_t that is longer than the actual generated sequence τ_t to inflate its payment. This can be done by adding inconsequential reasoning tokens, or by using techniques like watermarking

[1] We will simplify it as *cost* later when there is no ambiguity.

(Zhao et al., 2024) to add tokens that are difficult for humans to discern. We assume such misreporting has negligible cost.[2]

In reality, the strategy space is discrete and finite (e.g., the provider can only choose from a limited number of models). In our theory, for mathematical convenience, **we assume a continuous strategy space**, where c_t and the reported length $|\tau'_t|$ are both real numbers.

Next, we define the truthful behavior for provider d_t as follows:

1. Truthful cost: The provider incurs a cost $c_t = p_{d_t}$.
2. Truthful reporting: The provider reports the exact generated token sequence, i.e., $\tau'_t = \tau_t$.

Objectives and Utilities. The objectives of the user and providers are defined by their respective utility functions. The total utility for both the user and the provider is determined by the mechanism M and strategies $(S_1, S_2, \ldots, S_K)$ chosen.

The user aims to maximize the cumulative expected reward minus payments, with its expected utility defined as:

$$U(M, S_1, S_2, \ldots, S_K) = \mathbb{E}\left[\sum_{t=1}^{T} (v_t - p_{d_t} \cdot |\tau'_t|)\right].$$

A strategic provider i aims to maximize the cumulative expected difference between the truthful cost and the actual incurred cost, with its expected utility defined as:

$$u_i^p(M, S_1, S_2, \ldots, S_K) = \mathbb{E}\left[\sum_{t=1}^{T} (p_i \cdot |\tau'_t| - c_t \cdot |\tau_t|) \cdot \ 1_{d_t=i}\right].$$

Note that if the provider behaves truthfully, its expected utility is 0. Thus intuitively, it directly represents the gains from a smart strategy.

The randomness in the expectation is sourced from the user's mechanism, the provider's strategy, the text generation process, or the data sampling.

2.2 Goals of Mechanism Design

Dominant Strategy and Incentive Compatibility. The user's objective is to design a mechanism that shapes provider behavior to protect the user's own utility. Without a well-designed mechanism, a dishonest provider would naturally adopt a highly exploitative strategy, such as incurring minimal cost while reporting maximum-length outputs.

The goal of the mechanism is not to force total truthfulness (we will prove it is impossible later), but to strategically incentivize providers to be truthful to a certain level. To measure the effectiveness of the mechanism, we need to introduce the concept of a *dominant strategy*. An ideal mechanism ensures that each provider has an (approximately) optimal strategy, irrespective of the strategies chosen by other providers. This allows the proposed mechanism to produce a guaranteed outcome, as all providers will act according to the user's expectation. If such a mechanism is achievable, we call the mechanism is (approximately) incentive compatible.

Definition 1 (Dominant Strategy)**.** *Given a mechanism M, a strategy S_i is a $o(T)$-dominant strategy for provider i if for any of its other strategies S'_i, and for any strategy*

$S_{-i} = (S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_K)$ of other providers, we have

$$u_i^p(M, S_i, S_{-i}) \geq u_i^p(M, S'_i, S_{-i}).$$

Definition 2 ($o(T)$-Dominant Strategy)**.** *Given a mechanism M, a strategy S_i is a $o(T)$-dominant strategy for provider i if for any of its other strategies S'_i, and for any strategy*

[2]Note that how such inflation is realized is out of scope for this work.

S_{-i} of other models, with a probability of $1 - o\left(\frac{1}{T}\right)$, we have

$$u_i^p(M, S_i, S_{-i}) \geq u_i^p(M, S_i', S_{-i}) - o(T).$$

By this definition, we say a mechanism is $o(T)$-approximately incentive compatible if under this mechanism, an $o(T)$-approximately dominant strategy exists for every provider.

First-Best and Second-Best User Utility. We can evaluate the performance of our mechanism against two clear benchmarks based on truthful provider behavior. As an optimal benchmark, the first-best user utility represents the user utility if the user could identify the single best provider and use them, assuming that the provider acts truthfully. It is defined as:

$$u_{FB} = T \cdot \max_{i \in K} \left\{ \mu_i^r - p_i \mu_i^l \right\}.$$

where $\mu_i^r \in [0, R]$ and $\mu_i^l \in [1, L]$ is the provider's underlying expected reward and length of token sequence when incurring the truthful cost, respectively.

Assume provider i^* can deliver u_{FB}. Similarly, we define the second-best user utility as follows:

$$u_{SB} = T \cdot \max_{i \in K \setminus \{i^*\}} \left\{ \mu_i^r - p_i \mu_i^l \right\}.$$

3 Second-Best Utility Guarantee Mechanism

In this section, we propose our mechanism to mitigate the dishonest provider problem. There are four phases in our mechanism. We show the pseudocode of our mechanism in Mechanism 1 and illustrate it in Figure 1.

Informally, the optimal strategy under our mechanism (stated in Section 4) is that the provider would be truthful in the exploration phase, give "second-best" service in the exploitation phase, and receive additional rewards in the blind trust phase. With this result in mind, we explain each phase below.

Exploration phase. This phase is designed for the elicitation of providers' performance information. It operates by delegating each provider (expected to behave truthfully) on B queries to estimate their performance. The best-performing provider is chosen for the next phase. The collected samples are also used to estimate the second-best utility.

Exploitation phase. This is the key phase for the mechanism's user utility guarantee. It requires the best-performing provider to consistently deliver a user utility equivalent to the **second-best** among all providers from the exploration phase, for a duration of $T_R = O(T)$ queries. Recognizing the stochasticity in query samples and LLM generation, the performance check after each query involves a relax range of $O(T^{-\epsilon} \log T)$, which ensures that as long as the provider acts meeting the expectations, the probability of failing to complete the full T_R queries is small ($o(\frac{1}{T})$).

Blind trust phase I. This phase primarily serves to reward the chosen provider for behavior that meets user expectations during the exploitation phase, while also compensating other providers. In this phase and the subsequent blind trust phase II, providers are permitted to engage in actions maximizing utilities, namely by incurring the minimum possible cost and reporting the maximum possible length of the token sequence. Note that, the exploitation phase requires an initial number of B delegations to mitigate the effects of randomness. Therefore, without a reward of B-queries for the best provider, there is a clear incentive for it to deviate.

Blind trust phase II. As stated above, this phase is designed to incentivize honest behavior from all providers during the exploration phase. The number of queries is calculated to ensure that providers will achieve maximum utility when they incur truthful cost and report the truthful token sequence in the exploration phase.

Mechanism 1 Second-Best Utility Guarantee Mechanism

Input: Total queries T, number of model providers K, max possible reward R, the price-per-token set $\{p_i\}$, $\epsilon \in (0, \frac{1}{2})$.
$B \leftarrow T^{2\epsilon}$, $M \leftarrow T^{-\epsilon} \ln(KT)$
▷ Exploration phase starts
for each model provider $i = 1, \ldots, K$ **do**
  Delegate B queries to provider i.
  Calculate its average reward $\bar{v}_i$, average length of output token sequence $\overline{|\tau_i|}$, and average user utility $\bar{u}_i$.
  $\delta_i \leftarrow \frac{2\bar{v}_i}{p_i L} - \frac{2\overline{|\tau_i'|}}{L}$

▷ Exploitation phase starts; break ties randomly
$i^* \leftarrow \arg\max_i \bar{u}_i$, $\bar{u}' \leftarrow \max_{i \neq i^*} \bar{u}_i$
Inform provider i^* of the value $\bar{u}'$.
$T_R \leftarrow \left\lfloor T - \left(5K + \frac{2R}{\min_i\{p_i\}L} + \sum_{i \neq i^*} \delta_i\right) B - K \right\rfloor$
`validated` ← **true**
for $j = 1, \ldots, T_R$ **do**
  Delegate 1 query to provider i^*.
  if $j > B$ **and** current average utility in this phase from i^* is $< \bar{u}' - \frac{(b+p_{i^*}L)M}{3}$ **then**
    `validated` ← **false**
    break

▷ Blind trust phase I starts
if `validated` **then**
  Delegate B queries to provider i^*.
for each service provider $i \neq i^*$ **do**
  Delegate B queries to provider i.
▷ Blind trust phase II starts
for each service provider $i = 1, \ldots, K$ **do**
  $\delta_i' \leftarrow \delta_i + 3$
  if $\delta_i' \geq 0$ **then**
    Delegate $\lfloor B\delta_i' \rfloor$ queries to provider i.
    With a probability $B\delta_i' - \lfloor B\delta_i' \rfloor$, delegate an additional query to provider i.

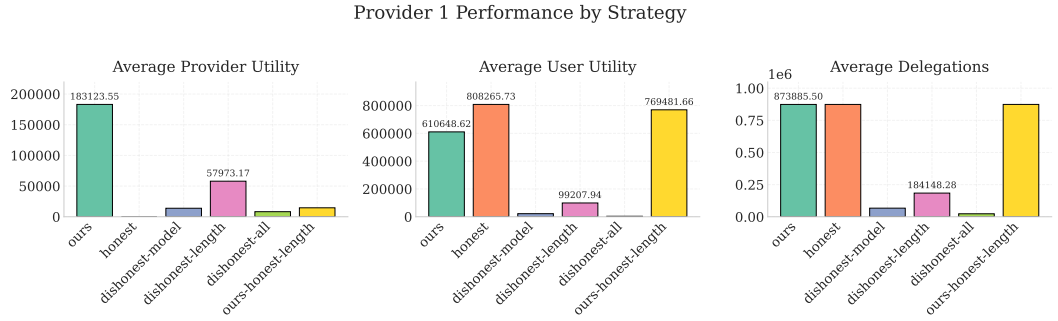

Figure 2: Simulation experiment where each provider enumerates strategies listed in Table 1. provider 1's expected provider utility, user utility, and number of delegations, averaged under permutations. Our proposed strategy gets the highest utility. The average delegations of `ours`, `honest`, `ours-honest-token` are the same.

4 Our Results

In this section, we present the main theoretical results. Firstly, we show that no $o(T)$-approximate incentive compatible mechanism can achieve the first-best user utility asymptotically. Consequently, we demonstrate that, given the assumption of certain model capabilities, the proposed mechanism is $O(T^{1-\epsilon} \log T)$-approximately incentive compatible; that is, it is an $O(T^{1-\epsilon} \log T)$-dominant strat-

Table 1: Overall provider strategies considered in our simulations. At each phase, the provider may choose a different strategy. Detailed descriptions of the strategy choices are provided in Table 2.

Provider Strategy	**Exploration**	**Exploitation**	**Blind Trust I**	**Blind Trust II**
Ours	Honest	Second-best	Worst	Worst
Honest	Honest	Honest	Honest	Honest
Dishonest-model	Honest	Worst model	Worst	Worst
Dishonest-length	Honest	Worst length	Worst	Worst
Dishonest-all	Worst	Worst	Worst	Worst
Ours-honest-length	Honest	Second-best with honest length	Worst	Worst

egy for all service providers. Concurrently, we prove that this mechanism yields a user utility close to the second-best user utility u_{SB}.

We provide theoretical guarantees for the continuous strategy space. **The complete proofs are detailed in Appendix B**.

4.1 Notations

To formally link a provider's actions to outcomes, we model the relationship between cost, output length, and reward.

For a random query $\omega \sim D$, provider i's incurring cost c induces a distribution $\mathcal{T}_{i,c}$ over the generated token sequence, denoted by the random variable $\tau_{i,c}$, and a distribution $\mathcal{V}_{i,c}$ over the user's reward, denoted by the random variable $v_{i,c}$. The distribution of the sequence length $|\tau_{i,c}|$ is consequently induced by $\mathcal{T}_{i,c}$.

We assume providers have an internal belief about their cost-performance trade-offs, modeled by two functions, $g_i(c)$ and $h_i(c)$, where $c \in [0, p_i]$, under the boundary condition $g_i(p_i) = \mu_i^l$ and $h_i(p_i) = \mu_i^r$ (both μ_i^l and μ_i^r are defined in Section 2). The expected length is defined such that $\mathbb{E}[|\tau_{i,c}|] = g_i(c)$. The expected reward is defined such that $\mathbb{E}[v_{i,c}] = h_i(c)$. We assume they are Lipschitz continuous.

4.2 Main Results

First, we state the impossibility of designing an $o(T)$-approximate incentive compatible mechanism that allows a user to achieve the first-best utility.

Theorem 4.1 (The impossibility of the first-best user utility)**.** *For any $\xi \in (0, 1)$, there is no $o(T)$-approximately incentive compatible mechanism can guarantee an expected user utility of $\xi u_{FB} + (1 - \xi)u_{SB}$, where u_{FB} and u_{SB} are the first-best and the second-best user utility (defined in Section 2.2).*

Proof sketch: We prove this theorem by contradiction. We construct a sufficiently large number of type sets (formally defined in Appendix B), among which the utility from the best provider in each type set exhibits certain variance, while the utility from the second-best providers remains constant. We derive that if a mechanism were to exist that could guarantee an expected utility of $\xi u_{FB} + (1 - \xi)u_{SB}$, it would necessitate the user to delegate tasks to the best provider more than T queries, which is an impossibility.

Next, we state the first positive result for our mechanism.

Theorem 4.2 ($O(T^{1-\epsilon} \log T)$-dominant strategy of all model providers)**.** *If for all providers $i \in [K]$ and choices of cost $c \in [0, p_i]$, the cost-performance functions $h_i(c), g_i(c)$ satisfies:*

- $\frac{\mathrm{d}h_i}{\mathrm{d}c}(c) - p_i \frac{\mathrm{d}g_i}{\mathrm{d}c}(c) \geq \gamma$ *for some* $\gamma > 0$,
- $\frac{\mathrm{d}h_i}{\mathrm{d}c}(c) - c\frac{\mathrm{d}g_i}{\mathrm{d}c}(c) - g_i(c) \geq 0$,

then the following strategy is an $O(T^{1-\epsilon} \log T)$ dominant strategy for provider i in Mechanism 1:

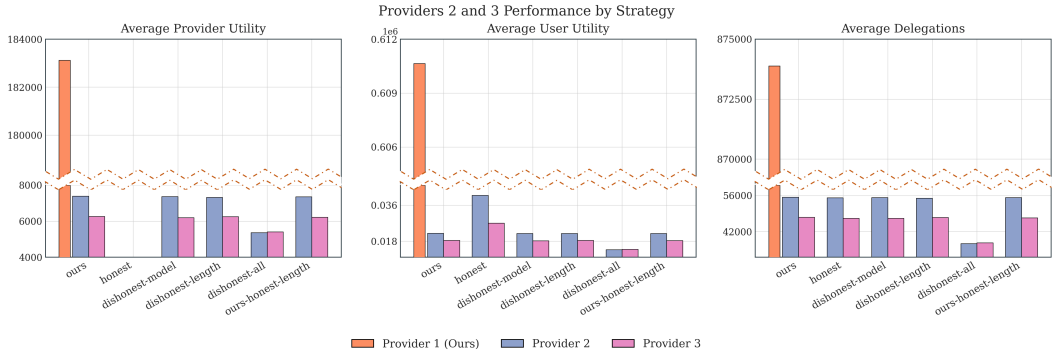

Figure 3: Average provider utility, user utility (from each provider), and number of delegations for provider 2 and provider 3 adopting different strategies when provider 1 adopts our proposed strategy. Since provider 1 has the best-performing LLM, they can not achieve a decent utility.

- *In the exploration phase, incur the truthful cost and report the truthful output token sequence for each query;*
- *If $i = i^*$, in the exploitation phase, for each query, calculate a pair (c', l')) s.t.*

$$(c', l') = \underset{\substack{c \in [0, p_i], l \in [g_i(c), L] \\ h_i(c) - p_i l \geq \bar{u}'}}{\arg\max} \{p_i l - cl\},$$

  where $i^*, \bar{u}'$ are chosen by the previous phase. Then incur a cost of c' and generate a corresponding token sequence τ. If $|\tau| \leq l'$, report an output token sequence of length l'; otherwise, report the truthful token sequence.
- *In the blind trust phases, incur a zero cost and report an output token sequence of length L for each query.*

Note that p_i is the price-per-token and the truthful cost-per-token of provider i, μ_i^r and μ_i^l are expected reward and length of output token sequence when incurring the truthful cost, and L is the max possible length of token sequence.

This theorem relies on two assumptions. First, we assume a positive correlation between a provider's incurred cost and the utility delivered to the user. This assumption is justifiable within our context, i.e., for each provider, in their own choices, LLMs of higher cost always yield better performance. The second assumption is that, as cost-per-token rises, the growth in reward to the user must outpace the increase in total cost.

By this theorem, we know that our mechanism is $O(T^{1-\epsilon} \log T)$-approximately incentive compatible.

Corollary 1. *Mechanism 1 is $O(T^{1-\epsilon} \log T)$-approximately incentive compatible.*

Finally, we prove that our mechanism guarantees a utility asymptotically close to the second-best user utility. The proof of this theorem is straightforward, as the user utility in our mechanism is primarily derived from the T_R queries during the exploitation phase.

Theorem 4.3 (Second-best user utility guarantee)**.** *If all providers action according to Theorem 4.2, the user could get an expected utility of at least $u_{SB} - O(T^{1-\epsilon} \log T + T^{2\epsilon})$.*

5 Experimental Results

To verify the correctness of our mechanism, we conduct simulation experiments. Due to lack of space, detailed settings are stated in Section C.

Table 2: Description of strategy choices considered in our simulations.

Strategy Choice	LLM Used	Output Length Reported
Honest	Largest cost	Truthfully
Second-best with honest length	Same as Theorem 4.2	Truthfully
Second-best	Same as Theorem 4.2	Same as Theorem 4.2
Worst model	Lowest cost	Truthfully
Worst length	Largest cost	Report the maximum limit
Worst	Lowest cost	Report the maximum limit

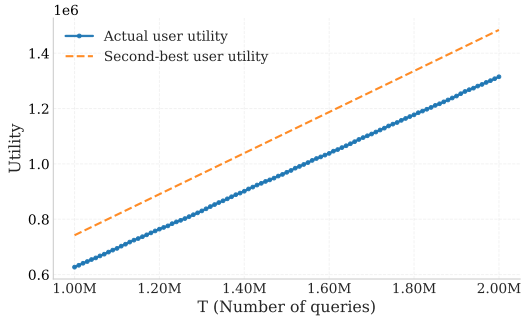

Figure 4: User utility of provider 1 as T increases from 1 million to 2 million. We also plot a theoretical reference, u_{SB} (defined in Section 2).

Metrics. For our main experiment, we calculate the average provider utility, user utility, and number of delegations for different strategy permutations (6^3 combinations in total). For provider 1 (Figure 2), we compute the average results under different strategy permutations adopted by provider 2 and provider 3. For provider 2 and provider 3 (Figure 3), we compute the average results under different strategy permutations when provider 1 adopts our proposed strategy in Theorem 4.2. For the supplementary experiment (Figure 4), we consider the situation that all providers follow our proposed strategy, as stated in Theorem 4.3.

Optimality of Our Proposed Strategy. We show our main experiment result in Figure 2 and Figure 3. Under all permutations of provider strategies, adhering to our proposed strategy yields the optimal average provider utility for the highest-performing provider (provider 1), while concurrently generating substantial user utility. Conversely, when provider 1 adopts our proposed strategy, providers 2 and 3 are precluded from being selected by the user during the exploitation phase, regardless of their chosen strategies. This outcome validates the robustness of our mechanism.

User Utility Guarantee. The results of our supplementary experiment are presented in Figure 4. As illustrated in the figure, the user utility exhibits a linear increase with T and close to the second-best user utility, which is consistent with the statement in Theorem 4.3. This demonstrates that our mechanism maintains stable effectiveness across different values of T.

6 Conclusion and Future Work

This work confronts the critical issue of service provider dishonesty within the black-box LLM API market, by shifting the focus from technical detection to the lens of algorithmic game theory. We formalize this challenge as a user-provider delegation game, for which we establish a foundational impossibility theorem. In response, we propose an approximately incentive-compatible mechanism that guarantees the user a near-optimal utility, the effectiveness of which is validated through comprehensive simulations grounded in real-world API parameters.

To the best of our knowledge, this work represents a foundational step towards ensuring a more transparent and trustworthy market for large language model services. Future research could extend our model to encompass multiple users, fostering a more competitive and fair market environment.

Another valuable problem is to assume the user have a budget constraint, which is a more realistic assumption and a challenge for our model. By establishing a framework grounded in economic principles, we hope to inspire further research at the intersection of artificial intelligence and mechanism design, fostering a healthier and more reliable ecosystem for all users of large language models.

7 LIMITATIONS

Our work has the following limitations:

First, our exploration of the model is limited. Although we have established a novel game model, there is considerable room for further investigation. Our theory does not delve into the mechanism design for scenarios where the token sequence length and the action space for cost control are discrete. Furthermore, our proposed mechanism requires the provider to have complete prior knowledge of the model's capabilities. Further, we do not consider potential collusion, which is also a challenge to mechanism design.

Second, the model itself has the potential for expansion to address more complex real-world situations. For instance, in practice, a provider could maliciously shrink the output token sequence. It is also possible for a provider to increase the length of the token sequence through prompt injection without affecting the LLM's performance, while still honestly reporting the token sequence.

REFERENCES

Introducing claude 4, 2025. URL https://www.anthropic.com/news/claude-4.

Introducing gpt-5, 2025. URL https://openai.com/index/introducing-gpt-5/.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2025.

Dirk Bergemann, Alessandro Bonatti, and Alex Smolin. The economics of large language models: Token allocation, fine-tuning, and optimal pricing, 2025. URL https://arxiv.org/abs/2502.07736.

Djallel Bouneffouf and Raphael Feraud. Survey: Multi-armed bandits meet large language models, 2025. URL https://arxiv.org/abs/2505.13355.

Mark Braverman, Jieming Mao, Jon Schneider, and S Matthew Weinberg. Multi-armed bandit problems with strategic arms. In *Conference on Learning Theory*, pp. 383–416. PMLR, 2019.

Will Cai, Tianneng Shi, Xuandong Zhao, and Dawn Song. Are you getting what you pay for? auditing model substitution in llm apis, 2025. URL https://arxiv.org/abs/2504.04715.

Lingjiao Chen, Matei Zaharia, and James Zou. How is chatgpt's behavior changing over time?, 2023. URL https://arxiv.org/abs/2307.09009.

Y Chen, L Li, and Y Zhang. Exploration–exploitation in llm decoding: An ϵ-greedy approach with regret guarantees. *arXiv preprint arXiv:2403.01168*, 2024.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Paul Dütting, Vahab Mirrokni, Renato Paes Leme, Haifeng Xu, and Song Zuo. Mechanism design for large language models. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 144–155, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701719. doi: 10.1145/3589334.3645511. URL https://doi.org/10.1145/3589334.3645511.

Paul Dütting, Michal Feldman, and Inbal Talgam-Cohen. Algorithmic contract theory: A survey. *Found. Trends Theor. Comput. Sci.*, 16(3–4):211–412, December 2025. ISSN 1551-305X. doi: 10.1561/0400000113. URL https://doi.org/10.1561/0400000113.

Sabri Eyuboglu, Karan Goel, Arjun Desai, Lingjiao Chen, Mathew Monfort, Chris Ré, and James Zou. Model changelists: Characterizing updates to ml models. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, pp. 2432–2453, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704505. doi: 10.1145/3630106.3659047. URL https://doi.org/10.1145/3630106.3659047.

Irena Gao, Percy Liang, and Carlos Guestrin. Model equality testing: Which model is this API serving? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QCDdI7X3f9.

Ruihao Gong, Yifu Ding, Zining Wang, Chengtao Lv, Xingyu Zheng, Jinyang Du, Haotong Qin, Jinyang Guo, Michele Magno, and Xianglong Liu. A survey of low-bit large language models: Basics, systems, and algorithms, 2024. URL https://arxiv.org/abs/2409.16694.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Abhinav Kumar, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. Overthink: Slowdown attacks on reasoning llms, 2025. URL https://arxiv.org/abs/2502.02542.

Benjamin Laufer, Jon Kleinberg, and Hoda Heidari. Fine-tuning games: Bargaining and adaptation for general-purpose models. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 66–76, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701719. doi: 10.1145/3589334.3645366. URL https://doi.org/10.1145/3589334.3645366.

Qijun Miao and Zhixuan Fang. User-side model consistency monitoring for open source large language models inference services. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11610–11622, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.569. URL https://aclanthology.org/2025.acl-long.569/.

Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pp. 129–140, 1999.

OpenAI. Introducing chatgpt, 2022. URL https://openai.com/index/chatgpt/.

OpenAI. Introducing openai o1, 2024a. URL https://openai.com/o1/.

OpenAI. Openai o3-mini, 2024b. URL https://openai.com/index/openai-o3-mini/.

Eden Saig, Ohad Einav, and Inbal Talgam-Cohen. Incentivizing quality text generation via statistical contracts. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=wZgw4CrxwK.

Guoheng Sun, Ziyao Wang, Bowei Tian, Meng Liu, Zheyu Shen, Shwai He, Yexiao He, Wanghao Ye, Yiting Wang, and Ang Li. Coin: Counting the invisible reasoning tokens in commercial opaque llm apis, 2025a. URL https://arxiv.org/abs/2505.13778.

Haoran Sun, Yurong Chen, Siwei Wang, Wei Chen, and Xiaotie Deng. Mechanism design for llm fine-tuning with multiple reward models, 2024. URL https://arxiv.org/abs/2405.16276.

Mingjie Sun, Yida Yin, Zhiqiu Xu, J. Zico Kolter, and Zhuang Liu. Idiosyncrasies in large language models, 2025b. URL https://arxiv.org/abs/2502.12150.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

TogetherAI. Announcing together inference engine – the fastest inference available, 2023. URL https://www.together.ai/blog/together-inference-engine-v1.

Ander Artola Velasco, Stratis Tsirtsis, and Manuel Gomez-Rodriguez. Auditing pay-per-token in large language models, 2025a. URL https://arxiv.org/abs/2510.05181.

Ander Artola Velasco, Stratis Tsirtsis, Nastaran Okati, and Manuel Gomez-Rodriguez. Is your llm overcharging you? tokenization, transparency, and incentives, 2025b. URL https://arxiv.org/abs/2505.21627.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36:30840–30861, 2023.

J Wang, H Liu, and M Zhang. Auditing llms with adversarial multi-armed bandits. In *ICLR*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

K Zhang, R Wang, and Z Liu. Contextual bandit rlhf: Reducing annotation cost via thompson sampling. In *NeurIPS*, 2023.

Xuandong Zhao, Sam Gunn, Miranda Christ, Jaiden Fairoze, Andres Fabrega, Nicholas Carlini, Sanjam Garg, Sanghyun Hong, Milad Nasr, Florian Tramer, Somesh Jha, Lei Li, Yu-Xiang Wang, and Dawn Song. Sok: Watermarking for ai-generated content, 2024. URL https://arxiv.org/abs/2411.18479.

A Additional Related Work

Auditing Large Language Models. Numerous studies have investigated methods for auditing the disparities between the expected and actual outputs of commercial LLMs from various perspectives. Chen et al. (2023); Eyuboglu et al. (2024) investigates how LLMs change their behavior over time. Sun et al. (2025b) explores the idiosyncrasies present in the text generated by different language models. Sun et al. (2025a) studies the auditing of invisible reasoning tokens in reasoning LLMs, specifically examining the existence of token count inflation with low-effort token injection. Cai et al. (2025) investigates the use of Trusted Execution Environments (TEEs), a hardware architecture, to provide cryptographic guarantees of model integrity. Gao et al. (2025) propose a concise and effective statistical testing scheme that quantifies changes in the output distribution due to quantization and watermarking by testing for the Maximum Mean Discrepancy between distributions. Recent works (Velasco et al., 2025b;a) systematically explore how LLM providers misreport token usage to gain additional utility under identical string representations, as well as how to audit such behaviors.

Game Theory Meets Large Language Models. The advancement of large language models brings new economic scenarios and problems. Dütting et al. (2024) introduce the token auction model, a mechanism design framework that allows multiple LLM agents to influence generated content through bids. Bergemann et al. (2025) provide economic frameworks for token pricing, and Laufer et al. (2024); Sun et al. (2024) analyse the game-theoretic structure of fine-tuning and RLHF respectively. Saig et al. (2024) investigates how to incentivize LLM service providers to generate high-quality texts via cost-robust contracts.

Online Learning in Large Language Model Scenario. Algorithms from online learning can be utilized to control LLMs under uncertainty. Zhang et al. (2023) frame RLHF as a contextual bandit and use Thompson sampling to select the most informative comparison queries. Chen et al. (2024) study the exploration–exploitation trade-off when an LLM generates candidate answers with an ϵ-greedy wrapper. Wang et al. (2025) treat safety auditing as an adversarial multi-armed bandit. Recent work (Bouneffouf & Feraud, 2025) systematically explore future directions for combining multi-armed bandits with LLMs. Drawing inspiration from mechanism design in (Braverman et al., 2019), we propose a novel model to address problems within the LLM scenario. This new context, in turn, presents new challenges for theoretical guarantees.

B PROOFS

B.1 IMPOSSIBILITY OF FIRST-BEST USER UTILITY

For clarity, we first provide definitions of types and type sets.

Definition 3 (Types and Type Sets)**.** *For any service provider i, we say its type consists of all private information (e.g. μ_i^r, u_i^p and g_i defined in Section 2 and Section 4),. A type set is the set of types for all providers.*

Note that although the price-per-token p_i for provider i is public knowledge, the utility function is private. For simplicity, we will also use p_i in the proof without further explanation.

Theorem 4.1 (The impossibility of the first-best user utility)**.** *For any $\xi \in (0, 1)$, there is no $o(T)$-approximately incentive compatible mechanism can guarantee an expected user utility of $\xi u_{FB} + (1 - \xi)u_{SB}$, where u_{FB} and u_{SB} are the first-best and the second-best user utility (defined in Section 2.2).*

Proof. We prove by contradiction.

Assume there exists an $o(T)$-approximate incentive-compatible mechanism that guarantees the user a worst-case utility of $\xi u_{FB}+(1-\xi)u_{SB}$. Denote i^* as the provider who delivers u_{FB}, and i' as the provider who delivers u_{SB}. We construct $N > \exp\left(\frac{1}{\xi-\lambda}\right)$ type sets, where $\lambda = \frac{\xi}{2}$, and for the j-th type set, we set $u_{FB} = T \cdot (\frac{1}{2} + \frac{j}{2N})$ and $u_{SB} = \frac{T}{2}$. we denote $u^{(j)} = \frac{1}{2} + \frac{j}{2N}$ and $u^\circ = \frac{1}{2}$. Within these type sets, we consistently use the notation (i^*, i'). The types of all other service providers are identical across all type sets. Let $F_j(c) = \frac{cu^{(j)}}{p_{i^*}}$, where $= 1$ and p_{i^*} keeps unchanged for all type sets.

For the j-th type set, let x_jT denote the expected number of delegations to provider i^* under the assumed mechanism, and let y_jT be the expected utility to the user. The total expected cost incurred by provider i^* could be represented by $F_j^{-1}(y_j)T$.

Because the mechanism is $o(T)$-approximately incentive compatible, all service providers will adopt their $o(T)$-approximate dominant strategies. During the mechanism running on the j-th type set, any provider k will prefer their dominant strategy for the j-th type set over the dominant strategy for any other type set j' (where $j \neq j'$), i.e., $x_jp_{i^*}T - F_j^{-1}(y_j)T \geq x_{j'}p_{i^*}T - F_j^{-1}(y_{j'})T - o(T)$. Hence, when T is sufficiently large, we have

$$
\begin{aligned}
x_jp_{i^*}T - F_j^{-1}(y_j)T &> x_{j'}p_{i^*}T - F_j^{-1}(y_j')T - \frac{\lambda p_{i^*}T}{u^{(j)}N} \\
\Leftrightarrow x_jp_{i^*}T - \frac{y_jp_{i^*}T}{u^{(j)}} &> x_{j'}p_{i^*}T - \frac{y_{j'}p_{i^*}T}{u^{(j)}} - \frac{\lambda p_{i^*}T}{u^{(j)}N} \\
\Leftrightarrow x_j - \frac{y_j}{u^{(j)}} &> x_{j'} - \frac{y_{j'}}{u^{(j)}} - \frac{\lambda}{u^{(j)}N} \\
\Leftrightarrow x_ju^{(j)} - y_j &> x_{j'}u^{(j)} - y_{j'} - \frac{\lambda}{N}.
\end{aligned}
\tag{1}
$$

Note that we only consider T to be non-constant in asymptotic notations. On the other hand, since the expected utility of i^* is not less than 0 (they can achieve it by incurring 0 cost at all times), we have

$$\begin{aligned} x_j p_{i^*} T - F^{-1}(y_j) T &\geq 0 \\ \Rightarrow x_j u^{(j)} - y_j &\geq 0. \end{aligned}$$

Therefore, by enumerating all j for inequality (1), we know that

$$\begin{cases} y_1 \leq x_1 u^{(1)} \\ y_2 \leq u^{(2)}(x_2 - x_1) + y_1 + \frac{\lambda}{N} \\ y_3 \leq u^{(3)}(x_3 - x_2) + y_2 + \frac{\lambda}{N} \\ \dots \\ y_N \leq u^{(N)}(x_N - x_{N-1}) + y_{N-1} + \frac{\lambda}{N}. \end{cases}$$

Thus for all $j \in [N]$ we have

$$\begin{aligned} y_j &\leq u^{(j)} x_j + x_{j-1}(u^{(j-1)} - u^{(j)}) + x_{j-2}(u^{(j-2)} - u^{(j-1)}) \\ &\quad + \dots + x_1(u^{(1)} - u^{(2)}) + \frac{j\lambda}{N} \\ &= u^{(j)} x_j + \sum_{k=1}^{j-1} x_k (u^{(k)} - u^{(k+1)}) + \frac{j\lambda}{N}. \end{aligned} \tag{2}$$

Consider an upper bound for the user's utility, that is, apart from delegating provider i^* for $x_j T$ queries, the user delegates provider i' for the remaining $(1 - x_j)T$ queries (as an upper bound). For provider i', it is impossible to bring more expected utility to the user, because they need to incur the truthful cost and report the truthful token sequences to bring such an expected utility to the user, and hence their own expected utility is already 0. At this point, the upper bound of the user's utility is $[y_j + (1 - x_j)u^\circ]\, T$. Therefore, $[y_j + (1 - x_j)u^\circ]\, T \geq (\xi u^{(j)} + (1 - \xi)u^\circ)T$.

By substituting y_j from the inequality (2), we know that

$$\begin{aligned} &\left[u^{(j)} x_j + \sum_{k=1}^{j-1} x_k (u^{(k)} - u^{(k+1)}) + (1 - x_j)u^\circ + \frac{j\lambda}{N} \right] T \\ \geq &(\xi u^{(j)} + (1 - \xi)u^\circ)T \\ \Rightarrow &u^{(j)} x_j + \sum_{k=1}^{j-1} x_k (u^{(k)} - u^{(k+1)}) + (1 - x_j)u^\circ \\ &\geq \xi u^{(j)} + (1 - \xi)u^\circ - \frac{j\lambda}{N} \\ \Rightarrow &(u^{(j)} - u^\circ) x_j \geq \xi(u^{(j)} - u^\circ) + \sum_{k=1}^{j-1} x_k (u^{(k+1)} - u^{(k)}) - \frac{j\lambda}{N} \\ \Rightarrow &x_j \geq \xi + \sum_{k=1}^{j-1} \frac{x_k (u^{(k+1)} - u^{(k)})}{(u^{(j)} - u^\circ)} - \lambda = \xi + \sum_{k=1}^{j-1} \frac{x_k}{j} - \frac{j\lambda}{N}. \end{aligned}$$

Next we use induction to prove $x_j \geq (\xi - \frac{j\lambda}{N}) \sum_{k=1}^{j} \frac{1}{k}$.

When $j = 1$, $x_1 = \xi - \frac{\lambda}{N}$. Assume (by induction) for $j^\circ \in [N]$, we have $x_j \geq (\xi - \frac{j\lambda}{N}) \sum_{k=1}^{j} \frac{1}{k}$ holds for all $1 \leq j \leq j^\circ$. For $j^\circ + 1$, we know that

$$
\begin{aligned}
x_j &\geq \xi + \sum_{k=1}^{j-1} \frac{x_k}{j} - \frac{j\lambda}{N} \\
&\geq \xi - \frac{j\lambda}{N} + \sum_{k=1}^{j-1} \frac{\left(\xi - \frac{k\lambda}{N}\right) \sum_{l=1}^{k} \frac{1}{l}}{j} \\
&\geq \left(\xi - \frac{j\lambda}{N}\right) \left(1 + \frac{1}{j} \sum_{k=1}^{j-1} \sum_{l=1}^{k} \frac{1}{l}\right) \\
&= \left(\xi - \frac{j\lambda}{N}\right) \left(1 + \frac{1}{j} \sum_{l=1}^{j-1} \frac{j-l}{l}\right) \\
&= \left(\xi - \frac{j\lambda}{N}\right) \left(1 + \sum_{l=1}^{j-1} \frac{j-l}{jl}\right) \\
&= \left(\xi - \frac{j\lambda}{N}\right) \left(1 + \sum_{l=1}^{j-1} \frac{1}{l} - \frac{j-1}{j}\right) \\
&= \left(\xi - \frac{j\lambda}{N}\right) \sum_{l=1}^{j} \frac{1}{l}.
\end{aligned}
$$

Therefore, $x_j \geq \left(\xi - \frac{j\lambda}{N}\right) \sum_{k=1}^{j} \frac{1}{k} > \left(\xi - \frac{j\lambda}{N}\right) \ln j$ for all $j \in [N]$ and we have

$$
\begin{aligned}
x_N &> (\xi - \lambda) \ln N \\
&= (\xi - \lambda) \ln \exp\left(\frac{1}{\xi - \lambda}\right) \\
&= 1.
\end{aligned}
$$

It implies that the user delegates $x_N T > T$ queries for i^*, which is a contradiction.

□

B.2 Properties of Our Mechanism

We first state all lemmas used in the following proofs.

Lemma B.1 (Hoeffding's inequality)**.** *Let $Z_1, \ldots, Z_n$ be independent bounded random variables with $Z_i \in [a, b]$ for all i, where $-\infty < a \leq b < \infty$. Then*

$$
\mathbb{P}\left(\frac{1}{n} \sum_{i=1}^{n} (Z_i - \mathbb{E}[Z_i]) \geq t\right) \leq \exp\left(-\frac{2nt^2}{(b-a)^2}\right)
$$

and

$$
\mathbb{P}\left(\frac{1}{n} \sum_{i=1}^{n} (Z_i - \mathbb{E}[Z_i]) \leq -t\right) \leq \exp\left(-\frac{2nt^2}{(b-a)^2}\right)
$$

for all $t \geq 0$.

Lemma B.2. *The total completed queries of mechanism 1 are at most T.*

Proof. The total completed queries

$$
\begin{aligned}
T' &= KB + T_R + KB + \sum_{i=1}^{K} [B(\delta_i + 3) + 1] \\
&\leq KB + T - \left[5K + \frac{2R}{\min_i\{p_i\}L} + \sum_{i \neq i^*} \delta_i \right] B + KB - K \\
&+ \sum_{i=1}^{K} [B(\delta_i + 3) + 1] \\
&= T - \left[\frac{2R}{\min_i\{p_i\}L} + \sum_{i \neq i^*} \delta_i \right] B + \sum_{i=1}^{K} B\delta_i \\
&= T - \frac{2BR}{\min_i\{p_i\}L} + B\delta_{i^*} \\
&\leq T - \frac{2BR}{\min_i\{p_i\}L} + \frac{2BR}{p_{i^*}L} \\
&\leq T.
\end{aligned}
$$

The second-to-last inequality is from the definition of δ_i in Mechanism 1.

□

Lemma B.3. *For all $i \in [K]$, if the cost-performance functions $h_i(c), g_i(c)$ satisfies that there exists a $\gamma > 0$ s.t. $\frac{\mathrm{d}h_i}{\mathrm{d}c}(c) - p_i \frac{\mathrm{d}g_i}{\mathrm{d}c}(c) \geq \gamma$, then for all pairs $(c', l'), (c', l'')$ s.t.*

$$
\begin{aligned}
(c', l') &= \underset{\substack{c \in [0, p_i], l \in [g_i(c), L] \\ h_i(c) - p_i l \geq \bar{u}'}}{\arg\max} \{p_i l - cl\}, \\
(c'', l'') &= \underset{\substack{c \in [0, p_i], l \in [g_i(c), L] \\ h_i(c) - p_i l \geq \bar{u}' - \frac{(b + p_i L)M}{3}}}{\arg\max} \{p_i l - cl\},
\end{aligned}
$$

we have $p_i(l'' - l') - c''l'' + c'l' = O(M)$.

Proof. Let $H(c, l) = p_i l - cl$, $\Delta_M = \frac{(b + p_i L)M}{3}$, and P_1, P_2 denote two programs in the statement respectively.

(c'', l'') may not be feasible for P_1, as the value of $h_i(c'') - p_i l''$ could be in the interval $[\bar{u}' - \Delta_M, \bar{u}')$. To construct a feasible solution for P_1, we need to increase the value of the expression $h_i(c) - p_i l$ by at most Δ_M.

By the assumption

$$
\frac{\mathrm{d}h_i}{\mathrm{d}c}(c) - p_i \frac{\mathrm{d}g_i}{\mathrm{d}c}(c) \geq \gamma > 0,
$$

we know that the user utility function $\phi(c) = h_i(c) - p_i g_i(c)$ is strictly increasing in c at a rate of at least γ. We will leverage this property to construct a feasible solution for P_1 starting from (c'', l'').

We construct a new point $(\tilde{c}, \tilde{l})$ by perturbing c'' by a small positive amount Δ_c. Let $\tilde{c} = c'' + \Delta_c$. To ensure feasibility while controlling the change, we set $\tilde{l}$ to its lower bound (constrained by both P_1 and P_2), i.e., $\tilde{l} = g_i(\tilde{c})$.

We know that $l'' \geq g_i(c'')$. Therefore:

$$
\phi(c'') = h_i(c'') - p_i g_i(c'') \geq h_i(c'') - p_i l'' \geq \bar{u}' - \Delta_M.
$$

We want to choose Δ_c such that $\phi(\tilde{c}) = \phi(c'' + \Delta_c) \geq \bar{u}'$. This requires an increase in the value of ϕ by at most Δ_M. The change in ϕ is:

$$\begin{aligned}\phi(c'' + \Delta_c) - \phi(c'') &= \int_{c''}^{c''+\Delta_c} \left(\frac{\mathrm{d}h_i}{\mathrm{d}c}(c) - \frac{\mathrm{d}g_i}{\mathrm{d}c}(c) p_i \right) \mathrm{d}c \\ &\geq \int_{c''}^{c''+\Delta_c} \gamma \, \mathrm{d}c = \gamma \Delta_c.\end{aligned}$$

To achieve the required increase, we set $\gamma\Delta_c = \Delta_M$, which gives $\Delta_c = \frac{\Delta_M}{\gamma}$. Since $\Delta_M = O(M)$ and γ is a positive constant, we have $\Delta_c = O(M)$.

We have now constructed a point $(\tilde{c}, \tilde{l}) = (c'' + \Delta_c, g_i(c'' + \Delta_c))$ that is feasible for P_1. The change in the c coordinate is $|\tilde{c} - c''| = \Delta_c = O(M)$. Since g_i is Lipschitz continuous, the change in the l coordinate is also bounded:

$$\left| \tilde{l} - g_i(c'') \right| = |g_i(c'' + \Delta_c) - g_i(c'')| \leq \kappa \Delta_c \mu_i^l = O(M),$$

where κ is the Lipschitz constant of g_i. The distance between (c'', l'') and $(\tilde{c}, \tilde{l})$ is thus of order $O(M)$.

On the other hand, the objective function $H(c, l) = p_i l - cl$ has bounded partial derivatives on the compact feasible domain, which implies it is also Lipschitz continuous. This property guarantees that a perturbation of order $O(M)$ in the input variables will result in a change of at most order $O(M)$ in the function's value, i.e.,

$$\begin{aligned}H(c'', l'') - H(\tilde{c}, \tilde{l}) &= O\left(\left\| (c'', l'') - (\tilde{c}, \tilde{l}) \right\|_\infty \right) \\ &= O(M).\end{aligned}$$

By construction, $(\tilde{c}, \tilde{l})$ is a feasible solution for P_1. By the definition of optimality, $H(c', l')$ must be greater than or equal to the value of the objective function at any feasible point, i.e., $H(c', l') \geq H(\tilde{c}, \tilde{l})$. Combining our results, we get:

$$\begin{aligned}p_i(l'' - l') - c''l'' + c'l' &= H(c'', l'') - H(c', l') \\ &\leq H(c'', l'') - H(\tilde{c}, \tilde{l}) \\ &= O(M).\end{aligned}$$

□

Theorem 4.2 ($O(T^{1-\epsilon} \log T)$-dominant strategy of all model providers)**.** *If for all providers $i \in [K]$ and choices of cost $c \in [0, p_i]$, the cost-performance functions $h_i(c), g_i(c)$ satisfies:*

- $\frac{\mathrm{d}h_i}{\mathrm{d}c}(c) - p_i \frac{\mathrm{d}g_i}{\mathrm{d}c}(c) \geq \gamma$ *for some* $\gamma > 0$,
- $\frac{\mathrm{d}h_i}{\mathrm{d}c}(c) - c\frac{\mathrm{d}g_i}{\mathrm{d}c}(c) - g_i(c) \geq 0$,

then the following strategy is an $O(T^{1-\epsilon} \log T)$ dominant strategy for provider i in Mechanism 1:

- *In the exploration phase, incur the truthful cost and report the truthful output token sequence for each query;*
- *If $i = i^*$, in the exploitation phase, for each query, calculate a pair (c', l')) s.t.*

$$(c', l') = \underset{\substack{c \in [0, p_i], l \in [g_i(c), L] \\ h_i(c) - p_i l \geq \bar{u}'}}{\arg\max} \{p_i l - cl\},$$

where $i^*, \bar{u}'$ are chosen by the previous phase. Then incur a cost of c' and generate a corresponding token sequence τ. If $|\tau| \leq l'$, report an output token sequence of length l'; otherwise, report the truthful token sequence.

- *In the blind trust phases, incur a zero cost and report an output token sequence of length L for each query.*

Note that p_i is the price-per-token and the truthful cost-per-token of provider i, μ_i^r and μ_i^l are expected reward and length of output token sequence when incurring the truthful cost, and L is the max possible length of token sequence.

Next we give the proof of Theorem 4.2, we do not repeat it here due to lack of space.

Proof. We prove that the strategy of any model provider at each step is approximately optimal under any historical conditions, using backward induction, a common method in sequential games.

First, we begin the proof with the analysis of the blind trust phases. Regardless of the previous delegation history, any action taken by a provider during this phase has no impact on the number of future delegations they or any other provider will receive. Therefore, in blind trust phases, a service provider will choose to incur a minimum cost, and misreport a token sequence with a length of L, which is optimal under any circumstances (i.e., a dominant strategy).

Next, we analyze the exploitation phase (with consideration of the blind trust phase I). It is important to note that in a sequential game, if all players are sufficiently rational, they will analyze the subsequent game situation after making a current choice and select the optimal decision for the entire game. We only need to consider the case where $i = i^*$, as when $i \neq i^*$, they have no room for action and receive the maximized utility by incurring a zero cost, and misreporting a token sequence with a length of L (which has been analyzed above).

Let $F(c, l) = h_{i^*}(c) - p_{i^*} l$ be the expected user utility when provider i^* incurs a cost of c and report the token sequence with a length of l. The history from the exploration phase leads to two cases: either $F(p_{i^*}, \mu_{i^*}^l) = \mu_{i^*}^r - p_{i^*}\mu_{i^*}^l < \bar{u}'$ or $F(p_{i^*}, \mu_{i^*}^l) \geq \bar{u}'$. If $F(p_{i^*}, \mu_{i^*}^l) < \bar{u}'$, the provider i^* should immediately incurs a zero cost and misreports a token sequence of length L (deviating), because they cannot obtain an expected utility greater than always deviating throughout the exploitation phase and the blind trust phase I.

If $F(p_{i^*}, \mu_{i^*}^l) \geq \bar{u}'$, let $\bar{v}_t$, $\overline{|\tau_t|}$ and $\overline{|\tau_t'|}$ be the average reward, average length of truthful and reported token sequences delivered by provider i^* after t time units in the exploitation phase. When KT is sufficiently large, by Hoeffding's inequality, we know that

$$
\begin{aligned}
\mathrm{P}\left[|\bar{v}_t - h_{i^*}(c')| \geq \frac{bM}{3}\right] &\leq 2\exp\left(-\frac{2B\left(\frac{bM}{3}\right)^2}{b^2}\right) \\
&= 2\exp\left(-\frac{2BM^2}{9}\right) \\
&= 2\exp\left(-\frac{2(\ln KT)^2}{9}\right) \\
&\leq 2\exp\left(-\frac{18\ln KT}{9}\right) \\
&\leq \frac{2}{(KT)^2}.
\end{aligned}
$$

Therefore, with a probability of $1 - o\left(\frac{1}{T}\right)$, the inequality $|\bar{v}_t - h_{i^*}(c')| \leq \frac{bM}{3}$ holds for all $t \geq B$. Similarly, $\left|\overline{|\tau_t|} - g_{i^*}(c')\right| \leq \frac{LM}{3}$ also holds for all $t \geq B$. Since the provider can only inflate the length of the reported token sequence, we have $\overline{|\tau_t'|} - l' \leq \frac{LM}{3}$.

Consequently, it can be inferred that with a probability of $1 - o\left(\frac{1}{T}\right)$,

$$\begin{aligned}\bar{v}_t - p_{i^*}\overline{|\tau_t'|} &\geq h_{i^*}(c') - p_{i^*}l' - \frac{(b+p_{i^*}L)M}{3} \\ &= \bar{u}' - \frac{(b+p_{i^*}L)M}{3}.\end{aligned}$$

This implies that the provider will not trigger the `validated` flag (turning it to false) with a probability of $1 - o\left(\frac{1}{T}\right)$. Therefore, the expected utility for provider i^* in the exploitation phase is then given by

$$U_2 \geq T_R(p_{i^*}l' - c'l'),$$

where "2" in the notation U_2 refers to the second phase.

Consider an ideal utility possibly achievable for provider i^*. The strategy is similar: For each query of the exploitation phase, it calculates a pair

$$(c'', l'') = \underset{\substack{c\in[0,p_{i^*}],l\in[g_{i^*}(c),L] \\ h_{i^*}(c)-p_{i^*}l\geq\bar{u}'-\frac{(b+p_{i^*}L)M}{3}}}{\arg\max} \{p_{i^*}l - cl\}.$$

Then it incurs a cost of c'' and generates a corresponding token sequence τ''. If $|\tau''| \leq l''$, it reports an output token sequence of length l''; otherwise, it reports the truthful token sequence. Should the provider be sufficiently fortunate, this strategy would also allow it to avoid triggering the `validated` flag and thereby achieve an ideal utility. Although the realization of such a scenario is probabilistically negligible, the resulting ideal utility serves as a definitive upper bound for the true optimal utility. Consequently, this ideal utility can be leveraged to compute a more strict additive approximation ratio. Hence, we calculate the ideal provider utility

$$U_2' = T_R(p_{i^*}l'' - c''l'').$$

and the additive approximation ratio

$$U_2' - U_2 = T_R[p_{i^*}(l'' - l') - (c''l'' - c'l')].$$

By Lemma B.3, we know that $U_2' - U_2$ is $O(T_R M) = O(T^{1-\epsilon}\log T)$.

Therefore, the strategy for provider i^* in the exploitation phase (with the consideration of the blind trust phase I) is an $O(T^{1-\epsilon}\log T)$-approximate dominant strategy.

Finally, we analyze the exploration phase. We will prove that incurring the truthful cost and output token sequence in this phase is a dominant strategy for provider i, regardless of the strategies of other providers.

Let provider i incurs an average cost $\bar{\alpha}_i$ and misreports an average extra token sequence length of $\bar{\beta}_i$. We know that $\bar{\alpha}_i \in [0, p_i]$, and $\bar{\beta}_i \in [0, L]$. When $i \neq i^*$, its total expected provider utility is

$$\begin{aligned}U_{\text{all}}\left(\bar{\alpha}_i, \bar{\beta}_i\right) = Bp_i\left(\bar{\beta}_i + g_i(\bar{\alpha}_i)\right) - B\bar{\alpha}_i g_i(\bar{\alpha}_i) + Bp_iL \\ + Bp_iL\left(3 + \frac{2h_i(\bar{\alpha}_i)}{p_iL} - \frac{2\left(\bar{\beta}_i + g_i(\bar{\alpha}_i)\right)}{L}\right)^+.\end{aligned}$$

We denote $G = 3 + \frac{2h_i(\bar{\alpha}_i)}{p_i L} - \frac{2\left(\bar{\beta}_i + g_i(\bar{\alpha}_i)\right)}{L}$. If $G \geq 0$, we first take the partial derivative with respect to $\bar{\alpha}_i$, and we have

$$\begin{aligned}
\frac{\partial U_{\text{all}}}{\partial \bar{\alpha}_i}\left(\bar{\alpha}_i, \bar{\beta}_i\right) &= Bp_i \frac{\mathrm{d}g_i}{\mathrm{d}\bar{\alpha}_i}(\bar{\alpha}_i) - B\bar{\alpha}_i \frac{\mathrm{d}g_i}{\mathrm{d}\bar{\alpha}_i}(\bar{\alpha}_i) - Bg_i(\bar{\alpha}_i) \\
&+ Bp_i L \cdot \left(\frac{1}{p_i L} \cdot \frac{\mathrm{d}h_i}{\mathrm{d}\bar{\alpha}_i}(\bar{\alpha}_i) \right. \\
&\left. + \frac{1}{p_i L}\left(\frac{\mathrm{d}h_i}{\mathrm{d}\bar{\alpha}_i}(\bar{\alpha}_i) - p_i \frac{\mathrm{d}g_i}{\mathrm{d}\bar{\alpha}_i}(\bar{\alpha}_i) \right) - \frac{\frac{\mathrm{d}g_i}{\mathrm{d}\bar{\alpha}_i}(\bar{\alpha}_i)}{L} \right) \\
&= B\left(\frac{\mathrm{d}h_i}{\mathrm{d}\bar{\alpha}_i}(\bar{\alpha}_i) - \bar{\alpha}_i \frac{\mathrm{d}g_i}{\mathrm{d}\bar{\alpha}_i}(\bar{\alpha}_i) - g_i(\bar{\alpha}_i) \right) \\
&+ B\left(\frac{\mathrm{d}h_i}{\mathrm{d}\bar{\alpha}_i}(\bar{\alpha}_i) - p_i \frac{\mathrm{d}g_i}{\mathrm{d}\bar{\alpha}_i}(\bar{\alpha}_i) \right) \\
&\geq 0.
\end{aligned}$$

On the other hand, the partial derivative with respect to $\bar{\beta}_i$

$$\frac{\partial U_{\text{all}}}{\partial \bar{\beta}_i}\left(\bar{\alpha}_i, \bar{\beta}_i\right) = Bp_i - 2Bp_i < 0.$$

Similarly, when $\bar{\beta}_i = 0$ (which means reporting truthful token sequences at all times), U_{all} takes its maximum value ($\bar{\alpha}_i$ fixed).

Since U_{all} is additively separable, when $\bar{\alpha}_i = p_i$ and $\bar{\beta}_i = 0$, U_{all} reaches its maximum value

$$Bp_i L + Bp_i L\left(3 + \frac{2\mu_i^r}{p_i L} - \frac{2\mu_i^l}{L}\right).$$

We know that $3 + \frac{2\mu_i^r}{p_i L} - \frac{2\mu_i^l}{L} \geq 1$, so $G \geq 0$ holds.

If $G < 0$, then

$$\frac{\partial U_{\text{all}}}{\partial \bar{\beta}_i}\left(\bar{\alpha}_i, \bar{\beta}_i\right) = Bp_i \geq 0.$$

Therefore, when $\bar{\beta}_i = L$, U_{all} reaches its (ideal) maximum value of

$$B(p_i - \bar{\alpha}_i)g_i(\bar{\alpha}_i) + Bp_i L \leq 2Bp_i L.$$

Since $3 + \frac{2\mu_i^r}{p_i L} - \frac{2\mu_i^l}{L} \geq 1$,

$$Bp_i L + Bp_i L\left(3 + \frac{2\mu_i^r}{p_i L} - \frac{2\mu_i^l}{L}\right) \geq 2Bp_i L.$$

Thus, the overall maximum value of U_{all} is attained at $\bar{\alpha}_i = 1$ and $\bar{\beta}_i = 0$.

When $i = i^*$, since the utility of the exploitation phase is independent of both $\bar{\alpha}_i$ and $\bar{\beta}_i$, the maximum value is also achieved when $\bar{\alpha}_i = 1$ and $\bar{\beta}_i = 0$. Thus, in either case, incurring the truthful cost and reporting the truthful token sequence is an $O(T^{1-\epsilon}\log T)$-dominant strategy for all providers.

□

Theorem 4.3 (Second-best user utility guarantee)**.** *If all providers action according to Theorem 4.2, the user could get an expected utility of at least $u_{SB} - O(T^{1-\epsilon}\log T + T^{2\epsilon})$.*

Proof.

$$\begin{aligned}\text{user utility} &= \mathbb{E}\left[\sum_{t=1}^{T}\left(v_{d_t,t} - p_{d_t}\cdot|\tau'_{d_t,t}|\right)\right]\\ &\geq \sum_{i=1}^{K}(\mu_i^r - p_i\mu_i^l)B + (\bar{u}' - \frac{(b+p_{i^*}L)M}{3})T_R\\ &+ \sum_{i=1}^{K}(\frac{2\mu_i^r}{p_iL} - \frac{2\mu_i^l}{L} + 3)p_iBL\\ &= u_{SB} - O(T^{1-\epsilon}\log T + T^{2\epsilon}).\end{aligned}$$

□

C Experimental Settings

Although our theoretical analysis is confined to the study of continuous action spaces, we conduct our experiments in a scenario where the action spaces for both cost control and token sequence length are discrete to demonstrate the practical effectiveness of our mechanism. Analogous to Theorem 4.2, in this context, we employ the upper difference as a proxy for the differential to construct a discrete version of assumptions (we omit the detailed formulation of this assumption due to lack of space). The proposed strategy in Theorem 4.2 remains the same in the subsequent experiments, with slight modifications for a discrete action space.

C.1 Setup

Simulation Scenarios. Our scenario involves three independent providers, each employing three models. We use the price of the best model as the provider's price, and the price of each model as the provider's different cost options. When a user delegates a query to a provider, the provider selects a model according to its strategy, returns the output token sequence to the user, and reports the number of tokens used (which may be inaccurate), thereby charging the user. In addition to our proposed strategy, we consider several other strategies, as shown in Table 1. Provider 1 employs three models: gpt-5-high (gpt, 2025), deepseek-r1 (DeepSeek-AI et al., 2025), and gpt-5-medium. Provider 2 employs o3-mini (OpenAI, 2024b), o1-mini (OpenAI, 2024a), and deepseek-r1. Provider 3 offers claude-4.0 (cla, 2025), o1-mini, and deepseek-r1. All model pricing information is obtained from the official websites of the respective models.

Dataset. We select the maximum flow task from NLGraph (Wang et al., 2023) as the dataset for our experiments for the following reasons: (1) the maximum flow task enables the generation of a large number of experimental samples; (2) current LLMs' performance on the task has not yet reached saturation. Following Wang et al. (2023), we use the "partial credit" as the evaluation metric for a single query (which is also the user's reward), ranging from 0 to 1, with higher values indicating better performance.

Hyperparameters. In our main experiment, we set T to 1 million. To reduce costs and accelerate the process, we run 2,000 results for each LLM on NLGraph and repeatedly sample these 2,000 results during the experiment. We set ϵ to 0.3, and L to 38,058 (the maximum output length in our samples). To verify the user utility guarantee, we also conduct a supplementary experiment by varying T from 1 million to 2 million.

We verify that the assumptions discussed at the beginning of this section are satisfied. And for this task, provider 1 has the best-performing LLM (gpt-5-high).

C.2 Strategies

In our experimental evaluation, we compare several different strategies (Table 1) to validate the efficacy of our proposed mechanism. As a baseline for comparison, the `honest` strategy consistently reports the model that is optimal for the user (based on the assumptions in Theorem 4.2, a higher cost

correlates with greater user utility). Consequently, the provider utility for this strategy is perpetually zero. In contrast, the `dishonest-all` strategy perpetually employs the worst LLM and bills for the maximum possible token length.

We further consider two more sophisticated dishonest strategies, `dishonest-token` and `dishonest-length`. These strategies behave honestly during the exploration phase and attempt to be selected for the subsequent exploitation phase. However, during the exploitation phase, they resort to deploying the worst LLM or billing for the maximum possible token length, respectively. When successful, these strategies could severely degrade user utility.

Furthermore, to demonstrate that misreporting the token sequence length is necessary for a strategic provider, we implement an additional strategy, `ours-honest-length`. This variant only attempts to substitute the model while reporting the truthful output length for billing.

D PROMPTS

D.1 PROMPTS IN NLGRAPH

Case: An Example Question for the Max-Flow Problem

In a directed graph, the nodes are numbered from 0 to 3, and the edges are:
an edge from node 1 to node 0 with capacity 10,
an edge from node 0 to node 2 with capacity 6,
an edge from node 2 to node 3 with capacity 4.
Q: What is the maximum flow from node 1 to node 3?
A:
<reason>
From the source (node 1), we can send 10 units of flow to node 0.
10 units of flow arrive at node 0, then we can send 6 units of flow from node 0 to node 2.
6 units of flow arrive at node 0, then we can send 4 units of flow from node 2 to node 3.
The total unit of flow arrives at node 3 is 4. So the maximum flow from node 1 to node 3 is 4.
</reason>
<answer>
4
</answer>

In a directed graph, the nodes are numbered from 0 to 4, and the edges are:
an edge from node 0 to node 3 with capacity 3,
an edge from node 0 to node 4 with capacity 10,
an edge from node 1 to node 3 with capacity 3,
an edge from node 1 to node 4 with capacity 5,
an edge from node 2 to node 1 with capacity 10,
an edge from node 3 to node 1 with capacity 10,
an edge from node 3 to node 4 with capacity 1,
an edge from node 4 to node 3 with capacity 2.
Q: What is the maximum flow from node 0 to node 4?
A:
<reason>
From the source (node 0), we can send 10 units of flow to node 4, and 3 units of flow to node 3.
3 units of flow arrive at node 3, then we can send 3 units of flow from node 3 to node 1.
3 units of flow arrive at node 1, then we can send 3 units of flow from node 1 to node 4.
The total unit of flow arrives at node 4 is 13. So the maximum flow from node 0 to node 4 is 13.
</reason>
<answer>
13
</answer>

In a directed graph, the nodes are numbered from 0 to 4, and the edges are:
an edge from node 0 to node 1 with capacity 10,
an edge from node 1 to node 2 with capacity 6,
an edge from node 0 to node 2 with capacity 2,
an edge from node 1 to node 3 with capacity 6,
an edge from node 3 to node 2 with capacity 3,
an edge from node 2 to node 4 with capacity 5,
an edge from node 3 to node 4 with capacity 8.
Q: What is the maximum flow from node 0 to node 4?
A:
<reason>
From the source (node 0), we can send 10 units of flow to node 1, and 2 units of flow to node 2.
10 units of flow arrive at node 1, then we can send 6 units of flow from node 3 to node 1, and 4 units of flow from node 3 to node 2.
6 units of flow arrive at node 2, then we can send 5 units of flow from node 2 to node 4.
6 units of flow arrive at node 3, then we can send 6 units of flow from node 3 to node 4.
The total unit of flow arrives at node 4 is 11. So the maximum flow from node 0 to node 4 is 11.
</reason>
<answer>
11
</answer>\n\n\nIn a directed graph, the nodes are numbered from 0 to 14, and the edges are:\nan edge from node 1 to node 3 with capacity 13,\nan edge from node 1 to node 0 with capacity 5,\nan edge from node 1 to node 13 with capacity 18,\nan edge from node 1 to node 10 with capacity 11,\nan edge from node 1 to node 5 with capacity 16,\nan edge from node 1 to node 2 with capacity 18,\nan edge from node 2 to node 8 with capacity 20,\nan edge from node 2 to node 1 with capacity 15,\nan edge from node 2 to node 5 with capacity 1,\nan edge from node 3 to node 6 with capacity 20,\nan edge from node 3 to node 11 with capacity 19,\nan edge from node 4 to node 12 with capacity 14,\nan edge from node 4 to node 1 with capacity 14,\nan edge from node 5 to node 14 with capacity 4,\nan edge from node 5 to node 9 with capacity 20,\nan edge from node 5 to node 7 with capacity 1,\nan edge from node 5 to node 0 with capacity 16,\nan edge from node 5 to node 2 with capacity 17,\nan edge from node 6 to node 9 with capacity 1,\nan edge from node 8 to node 7 with capacity 15,\nan edge from node 8 to node 3 with capacity 2,\nan edge from node 8 to node 0 with capacity 3,\nan edge from node 9 to node 12 with capacity 12,\nan edge from node 9 to node 5 with capacity 13,\nan edge from node 10 to node 6 with capacity 9,\nan edge from node 10 to node 0 with capacity 5,\nan edge from node 10 to node 12 with capacity 7,\nan edge from node 10 to node 11 with capacity 14,\nan edge from node 11 to node 6 with capacity 20,\nan edge from node 11 to node 12 with capacity 17,\nan edge from node 12 to node 7 with capacity 12,\nan edge from node 12 to node 0 with capacity 11,\nan edge from node 12 to node 13 with capacity 10,\nan edge from node 12 to node 1 with capacity 15,\nan edge from node 12 to node 4 with capacity 20,\nan edge from node 13 to node 6 with capacity 9,\nan edge from node 13 to node 8 with capacity 17,\nan edge from node 13 to node 0 with capacity 16,\nan edge from node 13 to node 10 with capacity 16,\nan edge from node 13 to node 2 with capacity 13,\nan edge from node 13 to node 11 with capacity 3,\nan edge from node 14 to node 3 with capacity 1,\nan edge from node 14 to node 4 with capacity 13.\nQ: What is the maximum flow from node 9 to node 7?\nA: