TEST-TIME COMPUTE GAMES

Ander Artola Velasco[§] **Dimitrios Rontogiannis**[§] **Stratis Tsirtsis**[†] **Manuel Gomez-Rodriguez**[§]

[§]Max Planck Institute for Software Systems, Kaiserslautern, Germany
{avelasco, drontogi, manuel}@mpi-sws.org

[†]Hasso Plattner Institute, Potsdam, Germany
stratis.tsirtsis@hpi.de

ABSTRACT

Test-time compute has emerged as a promising strategy to enhance the reasoning abilities of large language models (LLMs). However, this strategy has in turn increased how much users pay cloud-based providers offering LLM-as-a-service, since providers charge users for the amount of test-time compute they use to generate an output. In our work, we show that the market of LLM-as-a-service is socially inefficient—providers have a financial incentive to increase the amount of test-time compute, even if this increase contributes little to the quality of the outputs. To address this inefficiency, we introduce a reverse second-price auction mechanism where providers bid their offered price and (expected) quality for the opportunity to serve a user, and users pay proportionally to the marginal value generated by the winning provider relative to the second-highest bidder. To illustrate and complement our theoretical results, we conduct experiments with multiple instruct models from the `Llama` and `Qwen` families, as well as reasoning models distilled from `DeepSeek-R1`, on math and science benchmark datasets.

1 INTRODUCTION

The massive computational resources required to run large language models (LLMs) have led many users to rely on a growing market of cloud-based service providers that offer LLM-as-a-service. A key driver of this computational demand is the use of test-time compute (TTC)—additional computations performed by an LLM at inference time to improve its performance using techniques such as chain-of-thought (Wei et al., 2022), tree-of-thought (Yao et al., 2023b), and best-of-n sampling (Chow et al., 2025). Importantly, in the LLM-as-a-service market, the cost of these additional computations ultimately falls on the users, who pay for all tokens generated by a model during inference, including intermediate tokens that are not visible in the final outputs observed by the users.

In this context, providers have the flexibility to (dynamically) adjust the amount of test-time compute an LLM uses to respond to a user's query.[1] However, in a competitive market, this flexibility creates a new strategic dimension beyond how providers price their services. In particular, providers can strategically increase the amount of test-time compute allocated to a user's query to maximize profit, even if the additional test-time compute contributes little to the quality of the response. Consider a simple illustrative example where, for a given set of queries with verifiable ground truth (*e.g.*, diagnosing patients based on their medical records), two different providers can run their LLMs in either a low-TTC mode (*e.g.*, standard generation) or a high-TTC mode (*e.g.*, chain-of-thought), with average accuracies and generation costs for the providers given by:

Provider 1

	Avg. acc.	Avg. gen. cost
Low TTC	70%	\$0.25
High TTC	90%	\$1

Provider 2

	Avg. acc.	Avg. gen. cost
Low TTC	50%	\$0.5
High TTC	95%	\$10

[1]For example, GPT-5 uses real-time routing to adjust reasoning depth, and thus test-time compute. See `https://openai.com/index/introducing-gpt-5/`.

If both providers price their models with a 25% profit margin over their generation costs, it is easy to see that a user who values each percentage point of accuracy as $\$0.02$ would always select the first provider, who offers them higher value (*i.e.*, $\$0.02 \times$ accuracy $-$ price) independently of the TTC mode chosen by the second provider. However, to increase their profit, the first provider is financially incentivized to choose the high-TTC mode, even though the low-TTC mode would maximize the sum of the user value and provider profit, and would therefore be socially optimal. In our work, we formally characterize the above gap and show that, when providers act to maximize profit in a competitive market, they cannot, in general, be expected to use the socially optimal amount of test-time compute.

Our contributions. We start by modeling the current LLM-as-a-service market as a normal-form game (Nisan et al., 2007a) between N providers who choose their test-time compute allocations to maximize profit, and users pay according to the price per compute set by their selected provider. Building on this characterization, we show that the LLM-as-a-service market admits a pure Nash equilibrium and that, under natural assumptions, the market providers are guaranteed to reach this equilibrium in finite time. Then, we characterize the test-time compute used by providers in an LLM-as-a-service market at the Nash equilibrium and show that, in general, it is not socially optimal. To address this, we conceptualize and analyze a forward-looking market based on a reverse second-price auction mechanism where providers bid with their offered price and (expected) quality for the opportunity to serve a user, and the user's price is determined by the marginal value generated by the winning provider relative to the second-highest bidder. We show that this auction mechanism is socially optimal, and it guarantees that users always obtain at least their second-best achievable value for the task and providers secure non-negative profits.

To complement our theoretical results, we conduct experiments with multiple instruct models from the `Llama` and `Qwen` families, as well as reasoning models distilled from `DeepSeek-R1`, on math and science benchmark datasets. Our results show that the existing pay-for-compute market is up to 19% (socially) inefficient, as measured by the Price of Anarchy.[2]

Further related work. Our work connects to the rapidly growing literature on the economic and strategic aspects of machine learning systems (Kleinberg & Raghavan, 2021; Tsirtsis et al., 2024; Einav & Rosenfeld, 2025; Saig & Rosenfeld, 2025; Rosenfeld & Xu, 2025), and more specifically, LLMs-as-a-service (Duetting et al., 2024; La Malfa et al., 2024; Mahmood, 2024; Laufer et al., 2024; Saig et al., 2024; Bergemann et al., 2025; Velasco et al., 2025; 2026; Cao et al., 2025). Therein, most closely related to ours is a strand of recent works that analyzes the incentives of providers of LLM-as-a-service when they act strategically. Specifically, Velasco et al. (2025; 2026) study how providers can overcharge users by misreporting the number of tokens used by the LLM to encode a given text, and introduce an auditing method to detect such behavior. In a similar vein, Sun et al. (2025) and Cao et al. (2025) consider a setting where a provider injects additional reasoning tokens to inflate the user's payment, while Saig et al. (2024) and Cai et al. (2025) study a scenario in which providers have a financial incentive to be unfaithful to users by deploying cheaper-to-run models to maximize profit.

Further, our work relates to a line of work that analyzes the capabilities enabled by test-time compute (Wen et al., 2025; Wu et al., 2025; Bi et al., 2025; Liu et al., 2025; Snell et al., 2025). While most of the literature has focused primarily on performance aspects, there is increasing interest in analyzing the economic implications of test-time compute. In particular, recent works by Wang et al. (2024), Zellinger & Thomson (2025) and Erol et al. (2025) argue for incorporating the substantial costs of test-time compute into LLM evaluation and ranking, while Wan et al. (2025), Komiyama et al. (2025) and Kalayci et al. (2025) develop methods for selecting test-time compute resources once performance gains become marginal under majority voting and best-of-n. However, our work is the first to show that, in the LLM-as-a-service market, providers are incentivized to strategically decide about the amount of test-time compute used by the LLMs they serve, and to examine the resulting effects on the social welfare.

Finally, our work also draws on the classic literature in auctions and mechanism design (Krishna, 2009; Che, 1993; Nisan et al., 2007b), which has studied optimal auction design (Kersten, 2014; Myerson, 1981; Bhawalkar & Roughgarden, 2011), mechanisms to incentivize truthful bidding (Ledyard,

[2]The code for our experiments is publicly available at `https://github.com/Human-Centric-Machine-Learning/strategic-ttc`.

1987; Vickrey, 1961; Clarke, 1971; Groves, 1973), and the (in)efficiency of various markets when players act in their own interest (Koutsoupias & Papadimitriou, 1999; Roughgarden & Tardos, 2002; Roughgarden, 2015). Within this literature, the mechanism we propose for test-time compute markets is closely related to scoring auctions studied in the context of procurement contracts and supplier selection (Che, 1993; Roughgarden et al., 2017; Cai & Liao, 2026), where bidders submit offers specifying not only a price but also additional attributes of their offer (*e.g.*, quality), and the seller uses a scoring rule to determine the winner. In many settings, such scoring auctions have been shown to outperform first-price auctions, achieving higher efficiency (Asker & Cantillon, 2008; Awaya et al., 2025); yet, to the best of our knowledge, they have not been considered in the context of LLM-as-a-service.

2 A Game-Theoretic Model of Test-Time Compute

We model the LLM-as-a-service market as a normal-form (test-time compute) game $\mathcal{G}$ (Nisan et al., 2007a) between N providers who simultaneously deploy their own pre-trained model and compete for user demand by strategically selecting the level of test-time compute they use. In practice, each provider $i \in [N]$ can select different levels of test-time compute for different types of tasks (*e.g.*, low for fact retrieval and high for mathematical reasoning or coding), and their choice of compute for one task does not restrict their choices for other tasks. As a result, providers compete to serve each task independently, and each task defines an independent game between providers.

More formally, given a task $\mathcal{T}$ characterized by a set of queries $\mathcal{Q}$, each provider can select a test-time compute level $\theta \in \Theta$ for the model they serve from a finite set Θ of test-time compute levels.[3] For instance, in test-time compute methods such as majority voting (Wang et al., 2023) and best-of-n sampling (Chow et al., 2025), each level of test-time compute θ naturally corresponds to the number of generated samples and, in chain-of-thought (Wei et al., 2022), it may correspond to different "reasoning effort" levels. Then, given a test-time compute level θ, the model served by provider i produces an average output quality $q_i(\theta) \in \mathbb{R}_+$ for the task $\mathcal{T}$, which may scale differently across providers, *i.e.*, $q_i \neq q_j$ for $i \neq j$. In practice, the average output quality q_i may correspond to average accuracy across queries if the task $\mathcal{T}$ has a verifiable ground truth, or to an average user satisfaction score if the task $\mathcal{T}$ is open-ended. Moreover, there is extensive empirical evidence (Wen et al., 2025; Wu et al., 2025; Bi et al., 2025) that the quality q_i typically increases with the level of test-time compute, *i.e.*,

$$q_i(\theta) \leq q_i(\theta') \quad \text{for} \quad \theta \prec \theta', \tag{1}$$

where $\prec$ denotes a total order in the space Θ describing whether θ corresponds to a level of compute that is on average higher than θ' (*e.g.*, as measured by the average number of tokens used by the model per query).

Although provider i can increase the average quality $q_i(\theta)$ they offer to users by selecting a higher level of test-time compute θ, this comes at the expense of a higher (average) generation cost $c_i(\theta) \in \mathbb{R}_+$ for the provider. As a result, to compensate for higher generation costs, the price $p_i(\theta)$ charged by each provider i increases with the level of test-time compute θ, *i.e.*, $p_i(\theta) < p_i(\theta')$ for $\theta \prec \theta'$. Here, since each provider i is unlikely to set a price per unit of test-time compute lower than the cost,[4] we assume any increase in test-time compute leads to a positive marginal profit for the provider, *i.e.*, for $\theta \prec \theta'$, it holds that $p_i(\theta) - c_i(\theta) < p_i(\theta') - c_i(\theta')$.

Further, given a test-time compute level θ, each provider i offers a certain average value $V_i(\theta)$ to users for the task $\mathcal{T}$, which is given by the difference between the average output quality $q_i(\theta)$ produced by their model and the average price $p_i(\theta)$ charged by the provider for task $\mathcal{T}$. That is,

$$V_i(\theta) = q_i(\theta) - p_i(\theta), \tag{2}$$

where, for ease of exposition, we assume no ties in the values offered by different providers, *i.e.*, $V_i(\theta) \neq V_j(\theta')$ for all $i \neq j$ and all $\theta, \theta' \in \Theta$, since in practice both q_i and p_i refer to average quantities and thus take real values.

Given the test-time compute levels selected by all providers, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N) \in \Theta^N$, users allocate their demand across providers based on the value $V_i(\theta_i)$ offered by each provider, but they also

[3] For ease of exposition, we consider that all providers share the same action space Θ.

[4] In practice, a provider's generation cost and price for a model's output is proportional to the number of tokens it used to generate the output.

Table 1: Market share function, $s_i(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i}))$

Perfect rationality	Bounded rationality
$\mathbb{1}\{V_i(\theta_i) > \max\{\boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})\}\}$	$\frac{e^{\beta V_i(\theta_i)}}{e^{\beta V_0} + \sum_{j=1}^{N} e^{\beta V_j(\theta_j)}}$

have the option to abstain from using any provider if none of them offers a value higher than an abstention threshold value $V_0 \in \mathbb{R}$. More formally, we characterize the allocation of user demand across providers through a market share function $s : \mathbb{R}^{N+1} \to [0,1]^{N+1}$ that maps the values offered by all providers and the abstention value to the fraction of queries served by each provider and the fraction of queries where users abstain. Moreover, as is typical in game theory, we use the notation $s_i(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i}))$ to denote the market share of provider i as a function of their offered value $V_i(\theta_i)$ while keeping fixed all other values $\boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})$.

In the remainder of the paper, we will focus on the two canonical market share functions shown in Table 1. The first function corresponds to a Bertrand market (Tirole, 1988) where users are perfectly rational and always select the provider offering them the highest value. The second function corresponds to a market where users are boundedly rational (McKelvey & Palfrey, 1995) and select providers probabilistically based on their offered values.[5] Here, the parameter $\beta > 0$ controls the degree of rationality; as β increases, users become more rational and, in the limit $\beta \to \infty$, users become perfectly rational.

Given the choices of test-time compute levels made by all providers and the resulting market share they attain, each provider i obtains a utility $U_i(\theta_i; \boldsymbol{\theta}_{-i})$ given by

$$U_i(\theta_i; \boldsymbol{\theta}_{-i}) = \underbrace{s_i(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i}))}_{\text{Provider's market share}} \cdot \underbrace{(p_i(\theta_i) - c_i(\theta_i))}_{\text{Provider's profit}}. \tag{3}$$

In the next section, we will analyze the (social) welfare $\mathrm{W}(\boldsymbol{\theta})$ generated by an LLM-as-a-service market, which naturally corresponds to the sum of the total value obtained by users and the total utility obtained by all providers. More formally:

$$\begin{aligned} \mathrm{W}(\boldsymbol{\theta}) &\coloneqq \sum_{i=1}^{N} U_i(\boldsymbol{\theta}_i; \boldsymbol{\theta}_{-i}) + \sum_{i=1}^{N} V_i(\boldsymbol{\theta}_i) \cdot s_i(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) + V_0 \cdot s_i(V_0; \boldsymbol{V}(\boldsymbol{\theta})) \\ &= \sum_{i=1}^{N} s_i(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) \cdot \underbrace{(q_i(\theta_i) - c_i(\theta_i))}_{\coloneqq \mathrm{W}_i(\theta_i)} + V_0 \cdot s_i(V_0; \boldsymbol{V}(\boldsymbol{\theta})) \end{aligned} \tag{4}$$

where prices $p_i(\theta)$ cancel out, and $\mathrm{W}_i(\theta_i)$ denotes each provider's contribution to social welfare, reflecting the quality they offer relative to the generation cost incurred by their chosen level of test-time compute.

3 Social Welfare of Test-Time Compute Games

In this section, we will show that, in the LLM-as-a-service market, we can expect the test-time compute levels selected by the market providers to be suboptimal in terms of social welfare. To this end, as is typical in game theory, we will analyze the (pure) Nash equilibria of the test-time compute game $\mathcal{G}$ introduced in Section 2.

A choice of test-time compute levels $\boldsymbol{\theta}^\dagger \in \Theta^N$ is a pure Nash equilibrium of $\mathcal{G}$ if no provider can improve their utility through a unilateral change in test-time compute, *i.e.*,

$$U_i(\theta_i^\dagger; \boldsymbol{\theta}_{-i}^\dagger) \geq U_i(\theta_i'; \boldsymbol{\theta}_{-i}^\dagger) \;\text{ for all }\; i \in [N], \theta_i' \in \Theta. \tag{5}$$

To determine the existence of pure Nash equilibria, we resort to the theory of potential games (Monderer & Shapley, 1996). Specifically, we start by showing that every test-time compute game is a

[5] In the context of random utility models (RUMs), boundedly rational users are those who only have access to noisy estimations of the value offered by each provider (Yao et al., 2023a).

generalized ordinal potential game, that is, there exists a function $\Phi : \Theta^N \to \mathbb{R}$, called a potential, such that any unilateral deviation of some provider i that strictly increases their utility also strictly increases the potential. More specifically, let

$$\Phi(\boldsymbol{\theta}) = \begin{cases} \log\left(p_{\pi(1)}(\theta_{\pi(1)}) - c_{\pi(1)}(\theta_{\pi(1)})\right) + C \cdot V_{\pi(2)}(\theta_{\pi(2)}) & \text{(perf. rat.)} \\ \displaystyle\sum_{i=1}^{N} \log\left(p_i(\theta_i) - c_i(\theta_i)\right) + \beta \cdot \sum_{i=1}^{N} V_i(\theta_i) - \log\left(e^{\beta V_0} + \sum_{i=1}^{N} e^{\beta V_i(\theta_i)}\right) & \text{(bound. rat.)} \end{cases} \tag{6}$$

where $C > 0$ is a constant whose value is explicitly given in Appendix C.1 and π is the ordering of providers in decreasing order of the value they offer under $\boldsymbol{\theta}$. Then, we have the following theorem:

Theorem 1. *The function Φ defined in Eq. 6 is a potential for the test-time compute game $\mathcal{G}$,* i.e.*, for all $\boldsymbol{\theta} \in \Theta^N$, $i \in [N]$, and $\theta_i' \in \Theta$, it holds that $U_i\left(\theta_i'; \boldsymbol{\theta}_{-i}\right) > U_i\left(\theta_i; \boldsymbol{\theta}_{-i}\right) \Rightarrow \Phi\left(\theta_i'; \boldsymbol{\theta}_{-i}\right) > \Phi\left(\theta_i; \boldsymbol{\theta}_{-i}\right)$.*

Intuitively, Theorem 1 reveals that, although each provider acts selfishly to increase their own utility, the structure of the market is such that their actions jointly optimize the potential $\Phi(\boldsymbol{\theta})$, which captures the overall state of the market. In this context, it is worth noting that competition between providers drives up both the value offered to the users and (some of) the providers' profits.

Following well-known results in game theory Nisan et al. (2007a), the characterization of the test-time compute game $\mathcal{G}$ as an ordinal potential game readily implies that the game admits a (not necessarily unique) pure Nash equilibrium (Monderer & Shapley, 1996).To understand why this holds, consider that the levels of test-time compute selected by the providers start at some arbitrary point $\boldsymbol{\theta}^1 \in \Theta^N$ and follow *better-response dynamics* over time. Formally, this means that, at each time step t, the levels of test-time compute $\boldsymbol{\theta}^t$ are such that either (i) there exists a single provider $i \in [N]$ who changes their level of test-time compute to $\theta_i^{t+1} \neq \theta_i^t$ with $U_i(\theta_i^{t+1}; \boldsymbol{\theta}_{-i}^t) > U_i(\theta_i^t; \boldsymbol{\theta}_{-i}^t)$ or (ii) $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t$ if no such change is possible by any provider. Then, since the domain Θ^N is finite, it is guaranteed that the sequence $\boldsymbol{\theta}^t$ eventually becomes constant—otherwise, the better-response dynamics would keep increasing the potential function Φ indefinitely—at which point no provider can further increase their utility and the market reaches a pure Nash equilibrium. In this context, note that better-response dynamics are particularly natural in practice. One can think of the better-response of a provider as the (minor) release of a model that has not undergone additional pre-training or post-training, but whose performance has changed due to optimizations in test-time compute methods.

Further, we identify properties that hold in the equilibrium of a test-time compute game $\mathcal{G}$, starting from an explicit characterization of the levels of test-time compute selected by providers under perfect rationality. When users are perfectly rational, they always opt for the provider that offers them the highest value (see Table 1). Thus, each provider's incentive is to select a level of test-time compute that offers higher value than what their competitors are offering. This observation allows us to determine the equilibrium reached by the providers using the maximum value $V_i^* := \max_{\theta \in \Theta} \{q_i(\theta) - p_i(\theta)\}$ that each provider can offer, where, without loss of generality, we assume that providers are indexed such that $V_1^* \geq V_2^* \geq \cdots \geq V_N^*$ and we will refer to the provider who can offer the highest value V_1^* as the *dominant* provider. In particular, we have the following theorem:

Theorem 2. *In any pure Nash equilibrium $\boldsymbol{\theta}^\dagger$ of a test-time compute game $\mathcal{G}$ with perfectly rational users, (i) the dominant provider serves all queries with $\theta_1^\dagger = \max_{\theta_1 \in \Theta} \{\theta_1 \mid V_1(\theta_1) > V_2^*\}$, and (ii) there exists at least one provider $i \neq 1$ such that*

$$V_i(\theta_i^\dagger) > \max_{\theta_1 \in \Theta} \left\{ V_1(\theta_1) \;\middle|\; V_1(\theta_1) < V_2^* \quad \text{and} \quad p_1(\theta_1) - c_1(\theta_1) > p_1(\theta_1^\dagger) - c_1(\theta_1^\dagger) \right\},$$

where the right-hand side of the inequality is $-\infty$ if the set is empty.

The above result states that, in equilibrium, the dominant provider captures the market by selecting a level of test-time compute that offers higher value to the users than what their competitors can possibly offer and, once they achieve that, they increase their test-time compute as much as possible to maximize their utility. At the same time, at least one competitor offers a sufficiently high-value alternative while gaining zero utility—otherwise, the dominant provider could also lower the value they offer, and the situation would not be an equilibrium.

For games $\mathcal{G}$ with boundedly rational users ($\beta < \infty$), the levels of test-time compute in equilibrium cannot be characterized explicitly, since all providers serve a positive fraction of user queries and

their chosen levels of test-time compute depend on the costs, prices, and qualities $\{c_i, p_i, q_i\}_{i=1}^N$ of all providers. Nonetheless, we next show that, as long as users are sufficiently rational, any Nash equilibrium of the game $\mathcal{G}$ with $\beta < \infty$ is also an equilibrium of the game $\mathcal{G}$ with perfectly rational users and, hence, also satisfies the properties of Theorem 2. Formally, we have the following theorem:

Theorem 3. *Consider a set of providers specified by $\{c_i, p_i, q_i\}_{i=1}^N$. Then, there exists a (finite) level of user rationality $\beta_0 > 0$ such that, for any $\beta > \beta_0$, any pure Nash equilibrium $\boldsymbol{\theta}^\dagger$ of the game $\mathcal{G}$ with boundedly rational users is also a Nash equilibrium of the game $\mathcal{G}$ with perfectly rational users.*

Taken together, the above results provide some intuition regarding the provider utilities and user value in equilibrium. However, they do not elucidate to what extent we can expect the actions of providers in equilibrium to be optimal in terms of social welfare. To shed light on this question, in what follows, we will analyze the *price of anarchy* (PoA), a standard game-theoretic measure (Koutsoupias & Papadimitriou, 1999; Nisan et al., 2007b) that compares the social welfare achieved by the test-time compute levels $\boldsymbol{\theta}^\dagger$ selected by the providers in equilibrium against the test-time compute levels that maximize the social welfare of the market. More formally, the price of anarchy of a test-time compute game $\mathcal{G}$ is given by

$$\text{PoA}\left(\mathcal{G}\right) \coloneqq \frac{\max_{\theta \in \Theta^N} \text{W}(\boldsymbol{\theta})}{\text{W}(\boldsymbol{\theta}^\dagger)}. \tag{7}$$

By definition, the price of anarchy is always at least 1, and it equals 1 only when the compute levels $\boldsymbol{\theta}^\dagger$ selected by providers at equilibrium also optimize the social welfare. However, this is unlikely to happen, because rational providers act to maximize their individual utilities rather than coordinating to maximize the social welfare of the market. To better understand what conditions can lead to $\text{PoA}(\mathcal{G}) > 1$, and since explicitly characterizing the test-time compute level $\arg\max_{\theta \in \Theta^N} \text{W}(\boldsymbol{\theta})$ in an arbitrary game $\mathcal{G}$ is generally intractable, we compare the test-time compute $\boldsymbol{\theta}^\dagger$ selected by the providers at equilibrium against the test-time compute $\boldsymbol{\theta}^*$ selected by the providers in an idealized situation in which they maximize their individual contribution $\text{W}_i(\theta_i^*)$ to the welfare. By doing so, we obtain the following lower-bound on the price of anarchy:

Theorem 4. *The price of anarchy of the test-time compute game $\mathcal{G}$ is lower-bounded, up to higher order terms, as*

$$\text{PoA}\left(\mathcal{G}\right) \geq 1 + \frac{\text{W}_{\pi^*(1)}(\theta^*_{\pi^*(1)}) - \text{W}_{\pi(1)}(\theta^\dagger_{\pi(1)})}{\text{W}_{\pi^*(1)}(\theta^*_{\pi^*(1)})} + f(\beta),$$

where $f(\beta) = \mathcal{O}\left(e^{-\beta \Delta V}\right)$, $\Delta V > 0$ is a constant depending on the game instance, and π, π^* are permutations of providers in decreasing order of the value they offer at $\boldsymbol{\theta}^\dagger$ and $\boldsymbol{\theta}^*$, respectively. Refer to the proof in Appendix C.4 for the exact expression of f, ΔV, and the higher order corrections to the above lower bound.

The first term in the lower-bound of the price of anarchy given by Theorem 4 quantifies the difference between the contributions to the social welfare by the providers who offer the highest value in an idealized situation in which they select a test-time compute level that maximizes their individual contributions to social welfare (*i.e.*, $\boldsymbol{\theta}^*$) and in equilibrium (*i.e.*, $\boldsymbol{\theta}^\dagger$), while the term $f(\beta)$ quantifies the effect of users' bounded rationality, which vanishes exponentially fast as users become fully rational ($\beta \to \infty$).

Although Theorem 4 illustrates the fact that competition between providers can lead to levels of test-time compute that are inefficient in terms of social welfare, it is worth highlighting that, even in the absence of such inefficiencies, there is still room for the social welfare to improve further. Specifically, as can be seen in Eq. 4, the social welfare depends not only on the provider's test-time compute choices, but also on the way market shares are allocated across providers, which is determined by the providers' prices and the way users select which provider serves their queries (see Table 1). In an ideal scenario, each provider would select the level of test-time compute θ_i^* that maximizes their individual contribution $W_i(\theta_i^*)$ to social welfare, and users would select the provider delivering the highest such contribution. However, even under perfect rationality, this ideal scenario may not be achievable because the provider delivering the highest contribution $W_i(\theta_i^*)$ may not be the one delivering the highest value to the users. In the next section, we conceptualize and analyze an alternative forward-looking market that realizes such an ideal scenario.

4 AN AUCTION MECHANISM FOR TEST-TIME COMPUTE

To design a market in which social welfare is maximized, we draw inspiration from mechanism design (Nisan & Ronen, 2001; Krishna, 2009) and propose a reverse second-price auction that, by design, incentivizes each provider to choose the level of test-time compute θ_i^* that maximizes their individual contribution $W_i(\theta_i^*)$ to social welfare. Within this auction, a user takes the role of a seller who provides queries representative of a task they would like to be served and providers take the role of bidders who declare a quality and price they can offer for serving the user's task based on the representative queries. Moreover, the auction is conducted by a third-party platform that allocates the user's task to the provider who offers the best quality versus price trade-off and determines the payment of the provider based on the stated qualities and prices of all providers.

Formally, we denote the game induced by the aforementioned auction as $\widetilde{\mathcal{G}}$ and, for simplicity of notation, we overload the notation used in Sections 2 and 3, with the understanding that the corresponding definitions may actually differ. The process starts with a user submitting a set of queries $\mathcal{Q}$ representative of a task $\mathcal{T}$ they wish to solve to the third-party platform conducting the auction. Then, all N providers bid simultaneously for the opportunity to serve the task $\mathcal{T}$. Specifically, each provider i selects a level of test-time compute θ_i, and submits a bid $(q_i(\theta_i), p_i)$, where $q_i(\theta_i) \in \mathbb{R}_+$ is the average output quality achieved by the model they serve with the test-time compute level θ_i and $p_i \in \mathbb{R}_+$ is the price, as is common in scoring auctions (Che, 1993; Asker & Cantillon, 2008). Here, since the average quality of the responses generated by a provider can be verified, we assume that providers are truthful regarding the quality achieved by the model they serve. Finally, the platform assigns the task $\mathcal{T}$ to the provider who offers the highest value $V_i(\theta_i, p_i) \coloneqq q_i(\theta_i) - p_i$ where, similarly to Section 3, we assume there are no ties between providers.

To align the incentives of the providers with social welfare, the third-party platform sets the payment made by the user to the winning provider using a second-price payment rule—the user pays an amount that does not depend on the price $p_{\pi(1)}$ bid by the winning provider, but instead equals the difference between the quality delivered by the winning provider and the value offered by the second-best provider. Formally, the payment is given by

$$P(\boldsymbol{\theta}, \mathbf{p}) = q_{\pi(1)}(\theta_{\pi(1)}) - V_{\pi(2)}(\theta_{\pi(2)}, p_{\pi(2)}), \tag{8}$$

where π is the ordering of providers in decreasing order of the value they offer, and $\boldsymbol{\theta}$ and $\mathbf{p}$ denote the vectors of test-time compute levels and prices selected by all providers, respectively. Based on the payment in Eq. 8, the utility obtained by each provider is then given by

$$U_i(\theta_i, p_i; \boldsymbol{\theta}_{-i}, \mathbf{p}_{-i}) = \left(P(\boldsymbol{\theta}, \mathbf{p}) - c_i(\theta_i)\right) \cdot \mathbb{1}\left\{V_i(\theta_i, p_i) > \max_{j \neq i} V_j(\theta_j, p_j)\right\}, \tag{9}$$

where $c_i(\theta) \in \mathbb{R}_+$ is the generation cost, and the utility of all providers except for the winning one equals zero. Similar to the competitive market setting in Section 2, we define social welfare as the sum of the total utility obtained by all providers and the net value obtained by the user, which is the difference between the quality offered by the winning provider and the payment set by the third-party platform. More concretely, since the only provider with non-zero utility is the winning one, social welfare is given by

$$\mathrm{W}(\boldsymbol{\theta}, \mathbf{p}) = \underbrace{U_{\pi(1)}(\theta_{\pi(1)}, p_{\pi(1)}; \boldsymbol{\theta}_{-\pi(1)}, \mathbf{p}_{-\pi(1)})}_{\text{Winning provider's utility}} + \underbrace{q_{\pi(1)}(\theta_{\pi(1)}) - P(\boldsymbol{\theta}, \mathbf{p})}_{\text{Net user value}} = W_{\pi(1)}\left(\theta_{\pi(1)}\right),$$

where the second equality follows from Eqs. 8 and 9 and highlights that, due to the structure of the payment rule, the social welfare depends entirely on the quality, generation cost, and level of test-time compute of the winning provider $\pi(1)$.

Next, we proceed to analyze the equilibrium properties of the game $\widetilde{G}$ induced by the auction. Our starting point is the observation that, under the payment rule given by Eq. 8, each provider i has a dominant strategy, that is, a combination of test-time compute $\theta_i^\dagger$ and price $p_i^\dagger$ that maximizes their utility regardless of how the other providers act. Formally, a dominant strategy for provider i satisfies for all $(\theta_i, \boldsymbol{\theta}_{-i}) \in \Theta^N$ and $(p_i, \mathbf{p}_{-i}) \in \mathbb{R}_+^N$:

$$U_i(\theta_i^\dagger, p_i^\dagger; \boldsymbol{\theta}_{-i}, \mathbf{p}_{-i}) \geq U_i(\theta_i, p_i; \boldsymbol{\theta}_{-i}, \mathbf{p}_{-i}), \tag{10}$$

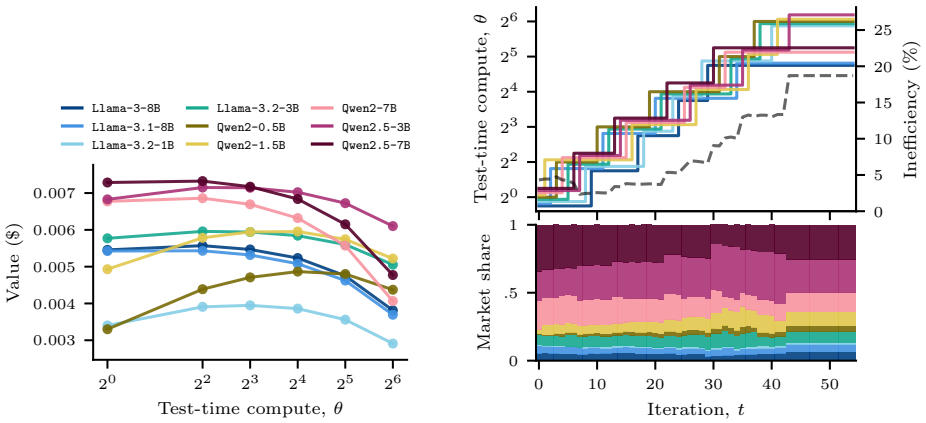


Figure 1: **Outcome of a test-time compute game.** The figure shows the outcome of a test-time compute game with $N = 9$ providers using best-of-n to serve queries from the GSM8K dataset. The left panel shows the value $V_i(\theta)$ that each provider offers to users as a function of their test-time compute. The right panels represent the better-response dynamics of the game and show the test-time compute $\boldsymbol{\theta}^t$ of each provider (with a small vertical jitter for visibility), together with the inefficiency of the game $\max_{\boldsymbol{\theta}} \mathrm{W}(\boldsymbol{\theta})/\mathrm{W}(\boldsymbol{\theta}^t) - 1$ (dashed black curve), and to the market share of each provider.

and, consequently, the choice of test-time compute levels $\boldsymbol{\theta}^\dagger$ and prices $\mathbf{p}^\dagger$ is a pure Nash equilibrium of $\widetilde{\mathcal{G}}$. The following theorem shows that such strategies (and hence, equilibria) always exist and characterizes their associated level of test-time compute and price:

Theorem 5. *In the game $\widetilde{\mathcal{G}}$, the choice of test-time compute level $\theta_i^\dagger = \theta_i^* = \arg\max_{\theta \in \Theta} W_i(\theta)$ and price $p_i^\dagger = c_i\left(\theta_i^*\right)$ is a dominant strategy for provider i.*

The above holds because, conditioned on winning the auction, the payment defined in Eq. 8 does not depend on the price $p_{\pi(1)}$. Thus, a provider can simultaneously increase their utility and the value they offer to the user by selecting the compute level θ_i^* to maximize $V_i(\theta_i, c_i(\theta_i)) = \mathrm{W}_i(\theta_i)$, and bidding the corresponding quality $q_i(\theta_i^*)$ and the lowest possible price—their generation cost c_i.

As an immediate consequence, in the Nash equilibrium $(\boldsymbol{\theta}^\dagger, \mathbf{p}^\dagger)$ of the game $\widetilde{\mathcal{G}}$, the value $V_i(\theta_i, p_i)$ offered by each provider i is equal to the maximum individual contribution to social welfare $\mathrm{W}_i(\theta_i^*)$ they can make. Since the platform selects the provider who offers the highest value to serve the task $\mathcal{T}$, in equilibrium, the winning provider is the one who can make the highest individual contribution to social welfare, *i.e.*, $\pi_{(1)} = \arg\max_{i \in [N]} \mathrm{W}_i(\theta_i^*)$. Then, the price of anarchy of the game $\widetilde{\mathcal{G}}$ satisfies

$$\mathrm{PoA}\left(\widetilde{\mathcal{G}}\right) := \frac{\max_{\boldsymbol{\theta}, \mathbf{p}} \mathrm{W}(\boldsymbol{\theta}, \mathbf{p})}{\mathrm{W}(\boldsymbol{\theta}^\dagger, \mathbf{p}^\dagger)} \overset{(*)}{=} \frac{\max_{i, \theta_i} \mathrm{W}_i(\theta_i)}{\mathrm{W}_{\pi(1)}(\theta^*_{\pi(1)})} = 1, \tag{11}$$

where $(*)$ holds because it is possible to set the prices $\mathbf{p}$ such that any provider wins the auction. Together, the above results show that, at equilibrium, the game $\widetilde{\mathcal{G}}$ achieves the maximum possible value of social welfare.

In this context, it is worth noting that, due to the choice of payment rule in Eq. 8, the game $\widetilde{\mathcal{G}}$ enjoys several additional desirable properties. First, one can easily verify that the winning provider receives a payment strictly higher than the price they bid (*i.e.*, $P(\boldsymbol{\theta}, \mathbf{p}) > p_{\pi(1)}$), and therefore always obtains positive utility. Second, the value received by the user matches the value offered by the second-best provider, *i.e.*, $q_{\pi(1)}(\theta_{\pi(1)}) - P(\boldsymbol{\theta}, \mathbf{p}) = V_{\pi(2)}(\theta_{\pi(2)}, p_{\pi(2)})$. We further explore these properties, as well as their implications for users and providers, in our experiments in Section 5.

5 Experiments

In this section, we conduct experiments to analyze the outcome of test-time compute games. We begin by briefly describing our experimental setup and refer the reader to Appendix D for additional details.

Experimental setup. For concreteness, here we focus on a particular game instance. Refer to Appendix E.2 for similar results for games with different levels of user rationality, and across other test-time compute methods (majority voting and chain-of-thought) and tasks.

We consider a test-time compute game, as defined in Section 2, in which each provider serves a different LLM, which we use to identify providers. More concretely, we study a game $\mathcal{G}$ with $N = 9$ providers, each serving a non-reasoning model selected from one of two families: (i) four models from the `Llama` family, `Llama-{3-8B,3.1-8B,3.2-1B,3.2-3B}-Instruct`, and (ii) five models from the `Qwen` family, `Qwen-{2-0.5B,2-1.5B,2-7B,2.5-3B,2.5-7B}-Instruct`. In the game $\mathcal{G}$, the task $\mathcal{T}$ that providers compete to serve consists of queries drawn from `GSM8K` (Cobbe et al., 2021), which contains mathematical questions with verifiable ground-truth answers. To serve this task, all providers use the same test-time compute method, namely best-of-n (Chow et al., 2025), in which providers can generate θ independent model outputs for a given query, and as the final response, they select the highest-scored one according to a reward model.

Lastly, we need to specify how the providers' compute choices determine both the value $V_i(\theta)$ they offer to users, and their own utilities $U_i(\boldsymbol{\theta})$. To this end, as we report in Appendix E.1, we first measure the average accuracy and the average number of generated tokens of all models for all test-time compute levels. Based on these measurements, we compute, for each provider and compute level θ, the prices $p_i(\theta)$ by multiplying the average number of generated tokens per query by the average per-token price. These token prices are obtained from the Hugging Face list of providers (see Appendix D). Then, we infer the provider costs $c_i(\theta)$ by assuming a fixed profit margin per token of 25% for all providers. Finally, we linearly map model accuracies into user values by considering that each (average) percentage point of accuracy is worth $\$0.008$ and set $\beta = 1000$.

Results. Figure 1 shows the outcome of the game $\mathcal{G}$ described above. In the upper panel, we show that the values $V_i(\theta)$ offered by providers are approximately concave with respect to their compute: users initially benefit from a slight increase in compute; however, as shown in Figure 3 in Appendix E.1.1, the number of tokens—and thus the price paid by the user—increases rapidly with compute, eventually diminishing their value.

To visualize how these differences in offered value translate into specific market outcomes, we show in the middle and lower panels of Figure 1 the dynamics of a test-time compute game where providers sequentially adjust their compute levels to increase their utility. In accordance with Theorem 1, since $\mathcal{G}$ is a potential game, the better-response dynamics converge to a pure Nash equilibrium. Importantly, we find that, once providers have reached an equilibrium, the market is socially inefficient: the price of anarchy is 1.19, representing a 16% loss in social welfare. Altogether, the above results suggest that competition in test-time compute markets does not naturally lead to the socially optimum outcome.

Lastly, in Table 3 (Appendix E.3), we show that compared to the test-time compute game $\mathcal{G}$, the auction mechanism $\widetilde{\mathcal{G}}$ increases social welfare by 25% and user value by 30%, while reducing the provider's utility by 25%. For other game instances, however, both users and providers can benefit simultaneously from the auction mechanism (see Table 4), while still guaranteeing that the mechanism is socially efficient (Eq. 11).

6 Conclusions

In this work, we introduced *test-time compute games*, a game-theoretic model in which LLM providers in a market of LLMs-as-a-service strategically select the level of test-time compute used by their model to compete for user queries and maximize their profit. Based on this model, we have demonstrated, first theoretically and then empirically, that current pricing approaches in markets of LLMs-as-a-service incentivize providers to set their level of test-time compute in a way that is not aligned with social welfare, as it does not optimally balance the tradeoff between generation cost and output quality. To address this, we proposed a forward-looking market based on a reverse second-price auction that provably aligns provider incentives with social welfare. We hope that our work inspires users, LLM providers, and online platforms to explore alternative market structures for generative AI that place social welfare at the center of their design choices.

Acknowledgements. Gomez-Rodriguez acknowledges support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101169607). Tsirtsis acknowledges support from the Alexander von Humboldt Foundation in the framework of the Alexander von Humboldt Professorship (Humboldt Professor of Technology and Regulation awarded to Sandra Wachter) endowed by the Federal Ministry of Education and Research via the Hasso Plattner Institute.

REFERENCES

John Asker and Estelle Cantillon. Properties of scoring auctions. *The RAND Journal of Economics*, 39(1):69–85, 2008. ISSN 07416261. URL http://www.jstor.org/stable/25046364.

Yu Awaya, Naoki Fujiwara, and Marton Szabo. Quality and price in scoring auctions. *Journal of Mathematical Economics*, 116:103083, 2025.

Dirk Bergemann, Alessandro Bonatti, and Alex Smolin. The economics of large language models: Token allocation, fine-tuning, and optimal pricing. EC '25, pp. 786, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400719431. doi: 10.1145/3736252.3742625. URL https://doi.org/10.1145/3736252.3742625.

Kshipra Bhawalkar and Tim Roughgarden. Welfare guarantees for combinatorial auctions with item bidding. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pp. 700–709. SIAM, 2011.

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. Forest-of-thought: Scaling test-time compute for enhancing LLM reasoning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4253–4267. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/bi25a.html.

Will Cai, Tianneng Shi, Xuandong Zhao, and Dawn Song. Are you getting what you pay for? auditing model substitution in llm apis. *arXiv preprint arXiv:2504.04715*, 2025.

Yuhang Cai and Huchang Liao. An overview of multi-attribute auctions: bibliometrics, methodologies, applications and future directions. *Expert Systems with Applications*, 299:130083, 2026. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2025.130083. URL https://www.sciencedirect.com/science/article/pii/S0957417425036991.

Yuhan Cao, Yu Wang, Sitong Liu, Miao Li, Yixin Tao, and Tianxing He. Pay for the second-best service: A game-theoretic approach against dishonest llm providers. *arXiv preprint arXiv:2511.00847*, 2025.

Yeon-Koo Che. Design competition through multidimensional auctions. *The RAND Journal of Economics*, 24(4):668–680, 1993. ISSN 07416261. URL http://www.jstor.org/stable/2555752.

Yeon-Koo Che, Daniele Condorelli, and Jinwoo Kim. Weak cartels and collusion-proof auctions. *Journal of Economic Theory*, 178:398–435, 2018. ISSN 0022-0531. doi: https://doi.org/10.1016/j.jet.2018.09.005. URL https://www.sciencedirect.com/science/article/pii/S0022053118305842.

Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Aviral Kumar, Rishabh Agarwal, Sridhar Thiagarajan, Craig Boutilier, and Aleksandra Faust. Inference-aware fine-tuning for best-of-n sampling in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=77gQUdQhE7.

Edward H Clarke. Multipart pricing of public goods. *Public choice*, pp. 17–33, 1971.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Paul Duetting, Vahab Mirrokni, Renato Paes Leme, Haifeng Xu, and Song Zuo. Mechanism design for large language models. In *Proceedings of the ACM Web Conference 2024*, pp. 144–155, 2024.

Ohad Einav and Nir Rosenfeld. A market for accuracy: Classification under competition. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=RPPBhhRddB.

Mehmet Hamza Erol, Batu El, Mirac Suzgun, Mert Yuksekgonul, and James Zou. Cost-of-pass: An economic framework for evaluating language models. *arXiv preprint arXiv:2504.13359*, 2025.

Theodore Groves. Incentives in teams. *Econometrica: Journal of the Econometric Society*, pp. 617–631, 1973.

Kenneth Hendricks and Robert H. Porter. Collusion in auctions. *Annales d'Économie et de Statistique*, (15/16):217–230, 1989. ISSN 0769489X, 22726497. URL http://www.jstor.org/stable/20075758.

Yusuf Kalayci, Vinod Raman, and Shaddin Dughmi. Optimal stopping vs best-of-n for inference time optimization. *arXiv preprint arXiv:2510.01394*, 2025.

Gregory E Kersten. Multiattribute procurement auctions: Efficiency and social welfare in theory and practice. *Decision Analysis*, 11(4):215–232, 2014.

Jon Kleinberg and Manish Raghavan. Algorithmic monoculture and social welfare. *Proceedings of the National Academy of Sciences*, 118(22):e2018340118, 2021. doi: 10.1073/pnas.2018340118. URL https://www.pnas.org/doi/abs/10.1073/pnas.2018340118.

Junpei Komiyama, Daisuke Oba, and Masafumi Oyamada. Best-of-infinity–asymptotic performance of test-time compute. *arXiv preprint arXiv:2509.21091*, 2025.

Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Annual symposium on theoretical aspects of computer science*, pp. 404–413. Springer, 1999.

V. Krishna. *Auction Theory*. Academic Press, 2009. ISBN 9780080922935. URL https://books.google.de/books?id=qW1128ktG1gC.

Emanuele La Malfa, Aleksandar Petrov, Simon Frieder, Christoph Weinhuber, Ryan Burnell, Raza Nazar, Anthony Cohn, Nigel Shadbolt, and Michael Wooldridge. Language-models-as-a-service: Overview of a new paradigm and its challenges. *Journal of Artificial Intelligence Research*, 80: 1497–1523, 2024.

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*, pp. 66–76, 2024.

John O Ledyard. Incentive compatibility. In *The New Palgrave dictionary of economics*, pp. 1–9. Springer, 1987.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling, 2025. URL https://arxiv.org/abs/2502.06703.

Rafid Mahmood. Pricing and competition for generative AI. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=8LbJfEjIrT.

George J Mailath and Peter Zemsky. Collusion in second price auctions with heterogeneous bidders. *Games and Economic Behavior*, 3(4):467–486, 1991. ISSN 0899-8256. doi: https://doi.org/10.1016/0899-8256(91)90016-8. URL https://www.sciencedirect.com/science/article/pii/0899825691900168.

Richard D. McKelvey and Thomas R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10(1):6–38, 1995. ISSN 0899-8256. doi: https://doi.org/10.1006/game.1995.1023. URL https://www.sciencedirect.com/science/article/pii/S0899825685710238.

Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1): 124–143, 1996. ISSN 0899-8256. doi: https://doi.org/10.1006/game.1996.0044. URL https://www.sciencedirect.com/science/article/pii/S0899825696900445.

Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35 (1):166–196, 2001. ISSN 0899-8256. doi: https://doi.org/10.1006/game.1999.0790. URL https://www.sciencedirect.com/science/article/pii/S089982569990790X.

Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, USA, 2007a. ISBN 0521872820.

Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani (eds.). *Algorithmic game theory*. Cambridge University Press, Cambridge, England, September 2007b.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Nir Rosenfeld and Haifeng Xu. Machine learning should maximize welfare, but not by (only) maximizing accuracy, 2025. URL https://arxiv.org/abs/2502.11981.

Tim Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM (JACM)*, 62(5): 1–42, 2015.

Tim Roughgarden and Éva Tardos. How bad is selfish routing? *Journal of the ACM (JACM)*, 49(2): 236–259, 2002.

Tim Roughgarden, Vasilis Syrgkanis, and Eva Tardos. The price of anarchy in auctions. *Journal of Artificial Intelligence Research*, 59:59–101, 2017.

Eden Saig and Nir Rosenfeld. Evolutionary prediction games. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=qsYbytjmQK.

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.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

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, 2025. URL https://arxiv.org/abs/2505.13778.

Jean Tirole. *The theory of industrial organization*. The MIT Press. MIT Press, London, England, January 1988.

Stratis Tsirtsis, Behzad Tabibian, Moein Khajehnejad, Adish Singla, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Optimal decision making under strategic behavior. *Management Science*, 70 (12):8506–8519, 2024. doi: 10.1287/mnsc.2021.02567. URL https://doi.org/10.1287/mnsc.2021.02567.

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

Ander Artola Velasco, Stratis Tsirtsis, and Manuel Gomez-Rodriguez. Auditing pay-per-token in large language models. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026.

William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

Guangya Wan, Zixin Stephen Xu, Sasa Zorc, Manel Baucells, Mengxuan Hu, Hao Wang, and Sheng Li. Beacon: Bayesian optimal stopping for efficient llm sampling. *arXiv preprint arXiv:2510.15945*, 2025.

Junlin Wang, Siddhartha Jain, Dejiao Zhang, Baishakhi Ray, Varun Kumar, and Ben Athiwaratkun. Reasoning in token economies: Budget-aware evaluation of LLM reasoning strategies. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19916–19939, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1112. URL https://aclanthology.org/2024.emnlp-main.1112/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Hao Wen, Yifan Su, Feifei Zhang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li. Parathinker: Native parallel thinking as a new paradigm to scale llm test-time compute, 2025. URL https://arxiv.org/abs/2509.04475.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms, 2025. URL https://arxiv.org/abs/2502.07266.

Fan Yao, Chuanhao Li, Karthik Abinav Sankararaman, Yiming Liao, Yan Zhu, Qifan Wang, Hongning Wang, and Haifeng Xu. Rethinking incentives in recommender systems: are monotone rewards always beneficial? *Advances in Neural Information Processing Systems*, 36:74582–74601, 2023a.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023b.

Michael J. Zellinger and Matt Thomson. Economic evaluation of llms, 2025. URL https://arxiv.org/abs/2507.03834.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

A Discussion and Limitations

In this section, we highlight several limitations of our work and discuss avenues for future research.

Model. Our definition of test-time compute games considers only factors related to the quality of LLM outputs and financial aspects, such as the pricing of LLMs by providers and their generation costs. In practice, however, other factors may influence users' choices among providers, leading to more complex behavior than described by the canonical market share functions in Table 1. These could include, for example, a provider's popularity or the support and reliability of their API service. Consequently, it would be interesting to extend our definition of test-time compute games to account for these additional factors and develop auction mechanisms for such (complex) games. Moreover, in our work, we assume that all users derive the same (average) value from a given LLM and test-time compute level; however, in practice, different users may value output quality differently, even for the same query and output. Extending test-time compute games to heterogeneous users is an interesting direction for future work.

Evaluation. We have studied the equilibria of test-time compute games on simulated markets using several state-of-the-art LLMs, including both reasoning and non-reasoning models, and test-time compute methods. In each of these games, a set of simulated providers compete to serve queries drawn from a benchmarking dataset with verifiable ground truth. However, it would be interesting to study the equilibria of test-time compute games where providers compete to serve open-ended queries without a verifiable ground truth that do not fall squarely into a single benchmarking dataset. Further, it would be important to analyze the dynamics of the LLM-as-a-service market using real-world data and deploy and evaluate our proposed auction mechanism in a real-world deployment.

Practical considerations. We have assumed that, in the auction mechanism described in Section 4, providers report truthfully the quality achieved by the models they serve. This is a reasonable assumption, since if a (dishonest) provider were selected to serve the task, the third-party platform could, in principle, detect discrepancies between the reported and realized quality. In practice, however, implementing such a mechanism would require the provider, the user, and the platform to agree ex ante on a contract specifying how quality is defined and measured. For tasks with verifiable ground truth, quality metrics are often well-defined. Examples include accuracy for fact retrieval or medical diagnosis, and `pass@k` for coding tasks. In contrast, open-ended tasks without verifiable ground truth are substantially more challenging to evaluate. In these settings, output quality is inherently subjective and may vary significantly across users depending on their preferences and intended use. In such cases, as recently argued by Saig et al. (2024), a practical approach is to automate quality evaluation using an agreed-upon evaluator, such as a designated LLM. Moreover, our second-price auction requires that providers accurately estimate the (average) quality that the model they serve can deliver for a given task. This requirement may be realistic in domains with established benchmarking datasets (*e.g.*, coding); however, it may be questionable in novel domains in which LLMs have not been extensively tested. Lastly, we note that, while second-price auctions possess many desirable properties, they also make a provider's payment dependent on the bids of other providers, leaving the mechanism potentially vulnerable to manipulation or collusion (Hendricks & Porter, 1989; Mailath & Zemsky, 1991; Che et al., 2018).

B Additional Details for the Introductory Example

Here, we provide additional details regarding the introductory example in Section 1. Therein, two LLM providers compete to serve a user on a fixed task and may each choose between a low- and a high-TTC mode for their respective models. The resulting average output accuracy and generation cost associated with each mode are given by:

Provider 1

	Avg. accuracy	Avg. gen. cost
Low TTC	70%	\$0.25
High TTC	90%	\$1

Provider 2

	Avg. accuracy	Avg. gen. cost
Low TTC	50%	\$0.5
High TTC	95%	\$10

Suppose that providers have fixed their prices to obtain a margin of 25% over their generation costs for both low and high TTC modes, and that the user values each percentage of accuracy as $\$0.02$. Then, using the notation in Section 2, for each compute mode $\theta \in \{\text{Low}, \text{High}\}$ used by each provider $i = 1, 2$, the value they offer to the user is

$$V_i(\theta) = \underbrace{\$0.02 \times a_i(\theta)}_{q_i(\theta)} - \underbrace{1.25 \times c_i(\theta)}_{p_i(\theta)}, \tag{12}$$

where $a_i(\theta) \in [0, 100]$ are the average accuracy when selecting compute θ, and c_i are the generation costs. Using the specific values in the above table, we obtain:

$$\begin{cases} V_1(\text{Low}) = \$1.0875 \\ V_1(\text{High}) = \$0.55 \end{cases} \quad \text{and} \quad \begin{cases} V_2(\text{Low}) = \$0.375 \\ V_2(\text{High}) = -\$10.6. \end{cases} \tag{13}$$

Consequently, a perfectly rational user would select the first provider to serve the queries, no matter which compute level they use, and the market (for this particular task) is monopolistically dominated by the first provider. The total user value and provider utility of this market (*i.e.*, the social welfare W in Section 3) is, depending on the compute level selected by the first provider:

$$\begin{cases} \text{W}(\theta_1 = \text{Low}, \theta_2) = V_1(\text{Low}) + p_1(\text{Low}) - c_1(\text{Low}) = \$1.15 \\ \text{W}(\theta_1 = \text{High}, \theta_2) = V_1(\text{High}) + p_1(\text{High}) - c_1(\text{High}) = \$0.8, \end{cases} \tag{14}$$

irrespectively of the compute level θ_2 of the second provider. That is, the low compute mode is socially optimal. However, the utility obtained by the first provider is

$$\begin{cases} U_1(\text{Low}; \theta_2) = 0.25 \times \$0.25 = \$0.0625 \\ U_1(\text{High}; \theta_2) = 0.25 \times \$1 = \$0.25, \end{cases} \tag{15}$$

which strictly incentivizes the provider to select the high-TTC mode despite its lower social welfare. In this stylized example, the price of anarchy is:

$$\text{PoA} = \frac{\$1.15}{\$0.8} = 1.4375. \tag{16}$$

C PROOFS

C.1 PROOF OF THEOREM 1

We prove Theorem 1 for the case of perfectly rational ($\beta = \infty$) and boundedly rational ($\beta < \infty$) separately.

C.1.1 POTENTIAL GAME (PERFECT RATIONALITY)

We will show that, in the case where users are perfectly rational, the game $\mathcal{G}$ is a generalized ordinal potential game, *i.e.*, there exists a function $\Phi : \Theta^N \to \mathbb{R}$ such that for any provider i:

$$U_i\left(\theta_i'; \boldsymbol{\theta}_{-i}\right) - U_i\left(\theta_i; \boldsymbol{\theta}_{-i}\right) > 0 \Rightarrow \Phi\left(\theta_i'; \boldsymbol{\theta}_{-i}\right) - \Phi\left(\theta_i; \boldsymbol{\theta}_{-i}\right) > 0 \quad \text{for all } \boldsymbol{\theta} \in \Theta^N, \theta_i' \in \Theta.$$

Given the test-time compute levels of all providers $\boldsymbol{\theta}$, let π be the permutation ordering providers by the value they offer to users, *i.e.*,

$$V_{\pi(1)}(\theta_{\pi(1)}) > V_{\pi(2)}(\theta_{\pi(2)}) > \cdots > V_{\pi(N)}(\theta_{\pi(N)}).$$

and denote, for conciseness, $V_{\pi(1)} = V_{\pi(1)}(\theta_{\pi(1)})$ and $V_{\pi(2)} = V_{\pi(2)}(\theta_{\pi(2)})$. We define the potential function as a weighted sum of the second largest value and the utility gained by the provider $\pi(1)$, *i.e.*,

$$\begin{aligned} \Phi(\boldsymbol{\theta}) &= C \cdot V_{\pi(2)} + \log U_{\pi(1)}(\theta_{\pi(1)}; \boldsymbol{\theta}_{-\pi(1)}) \\ &= C \cdot V_{\pi(2)} + \log\left(p_{\pi(1)}(\theta_{\pi(1)}) - c_{\pi(1)}(\theta_{\pi(1)})\right), \end{aligned} \tag{17}$$

where $C > 0$ is a constant whose value we will specify later on.

To prove that the function given by Eq. 17 is a generalized ordinal potential function for the game $\mathcal{G}$, we distinguish two cases, depending on whether the provider i who unilaterally changes their test-time compute level to increase their utility (i) is the winning provider under $\boldsymbol{\theta}$, that is, $i = \pi(1)$, or (ii) is not the winning provider under $\boldsymbol{\theta}$.

In the first case, based on Eq. 3 and the fact that the provider i strictly increased their utility, it has to hold that $s_i(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) = s_i(V_i(\theta_i'); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) = 1$ and $\theta_i \prec \theta_i'$. This implies that the ordering of providers in terms of the value they offer to the users remains the same before and after the change of test-time compute level by provider i, and the second highest value under $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ coincide. Thus,

$$\begin{aligned} \Phi\left(\theta_i'; \boldsymbol{\theta}_{-i}\right) - \Phi\left(\theta_i; \boldsymbol{\theta}_{-i}\right) &= \log U_{\pi(1)}(\theta'_{\pi(1)}; \boldsymbol{\theta}_{-\pi(1)}) - \log U_{\pi(1)}(\theta_{\pi(1)}; \boldsymbol{\theta}_{-\pi(1)}) \\ &= \log U_i(\theta_i'; \boldsymbol{\theta}_{-i}) - \log U_i(\theta_i; \boldsymbol{\theta}_{-i}) > 0. \end{aligned}$$

In the second case, it holds that $s_i(V_i(\theta_i'); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) = 0$, therefore, the only way for provider i to strictly increase their utility is by offering a higher value to the users than the previous winning provider and becoming the winning provider themselves. Therefore, under $\boldsymbol{\theta}'$, the winning provider is i and the provider who offers the second highest value to the users is the previous winning provider, which implies that $V'_{\pi'(2)} = V_{\pi(1)}$, where π' orders providers according to the values they offer under $\boldsymbol{\theta}'$. As a result, the difference in the function Φ is

$$\Phi\left(\theta_i'; \boldsymbol{\theta}_{-i}\right) - \Phi\left(\theta_i; \boldsymbol{\theta}_{-i}\right) = C \cdot (V_{\pi(1)} - V_{\pi(2)}) + \log U_i(\theta_i'; \boldsymbol{\theta}_{-i}) - \log U_{\pi(1)}(\theta_{\pi(1)}; \boldsymbol{\theta}_{-\pi(1)}), \tag{18}$$

where $V_{\pi(1)} - V_{\pi(2)} > 0$ because we have assumed no ties in the values providers can offer to the users. Moreover, since the action space Θ is finite, there has to exist a value $U_{max} > 0$ such that, for all j and $\boldsymbol{\theta}$, it holds that $U_j(\theta_j; \boldsymbol{\theta}_j) \leq U_{max}$. Combining this with the fact that provider utilities are non-negative, Eq. 18 implies that

$$\Phi\left(\theta_i'; \boldsymbol{\theta}_{-i}\right) - \Phi\left(\theta_i; \boldsymbol{\theta}_{-i}\right) \geq C \cdot (V_{\pi(1)} - V_{\pi(2)}) - \log U_{max},$$

and it suffices to find a value for the constant C such that $C > \frac{\log U_{max}}{V_{\pi(1)} - V_{\pi(2)}}$. Since we have assumed no ties in the values that providers can offer to the users, there has to exist a δ_{min} such that $V_i(\theta_i) - V_j(\theta_j) > \delta_{min}$, uniformly across providers i, j and compute levels θ_i, θ_j. Setting $C = \frac{\log U_{max}}{\delta_{min}}$ concludes the proof.

C.1.2 POTENTIAL GAME (BOUNDED RATIONALITY)

We will show that, in the case of boundedly rational users, the game $\mathcal{G}$ with $\beta < \infty$ is also a generalized ordinal potential game. To this end, we extend the potential in Eq. 17 under perfect rationality to a potential Φ under bounded rationality by observing that: (i) for $\beta < \infty$, all providers can have a non-zero market share (and hence non-zero utilities), (ii) each provider partially aims to improve their profit $p_i(\theta_i) - c_i(\theta_i)$ while still offering a sufficiently high value $V_i(\theta_i)$ relative to others, and (iii) in the log domain, utilities decompose as:

$$\log U_i(\theta_i; \boldsymbol{\theta}_{-i}) = \log\left(s_i(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) \cdot (p_i(\theta_i) - c_i(\theta_i))\right) \tag{19}$$

$$= \log\left(p_i(\theta_i) - c_i(\theta_i)\right) + \beta V_i(\theta_i) - \log\left(\sum_{j=0}^{N} \exp(\beta \cdot V_j(\theta_j))\right), \tag{20}$$

for $\boldsymbol{\theta} \in \Theta^N$ and any provider i, and we denote $V_0(\theta_0) \coloneqq V_0$ for simplicity. Based on the three observations above, let us define:

$$\Phi(\boldsymbol{\theta}) = \sum_{j=1}^{N} \log(p_j(\theta_j) - c_j(\theta_j)) + \beta \cdot \sum_{j=1}^{N} V_j(\theta_j) - \log\left(\sum_{j=0}^{N} \exp(\beta \cdot V_j(\theta_j))\right), \quad \forall \boldsymbol{\theta} \in \Theta^N, \tag{21}$$

where note that the last term is not summed over all providers, since it is a shared normalization constant related to the softmax allocation s under bounded rationality. To verify that Φ is indeed an ordinal potential function, consider a deviation θ_i' for provider i. Then:

$$U_i(\theta_i'; \boldsymbol{\theta}_{-i}) > U_i(\theta_i; \boldsymbol{\theta}_{-i}) \tag{22}$$

$$\iff \log U_i(\theta_i'; \boldsymbol{\theta}_{-i}) > \log U_i(\theta_i; \boldsymbol{\theta}_{-i}) \tag{23}$$

$$\iff \log(p_i(\theta_i') - c_i(\theta_i')) + \beta V_i(\theta_i') - \log\left(\sum_{j \neq i} e^{\beta V_j(\theta_j)} + e^{\beta V_i(\theta_i')}\right)$$

$$> \log(p_i(\theta_i) - c_i(\theta_i)) + \beta V_i(\theta_i) - \log\left(\sum_{j=0}^{N} e^{\beta V_j(\theta_j)}\right) \tag{24}$$

$$\iff \sum_{j \neq i,0} \left[\log(p_j(\theta_j) - c_j(\theta_j)) + \beta V_j(\theta_j)\right] + \log(p_i(\theta_i') - c_i(\theta_i')) + \beta V_i(\theta_i') - \log\left(\sum_{j \neq i} e^{\beta V_j(\theta_j)} + e^{\beta V_i(\theta_i')}\right)$$

$$> \sum_{j \neq i,0} \left[\log(p_j(\theta_j) - c_j(\theta_j)) + \beta V_j(\theta_j)\right] + \log(p_i(\theta_i) - c_i(\theta_i)) + \beta V_i(\theta_i) - \log\left(\sum_{j=0}^{N} e^{\beta V_j(\theta_j)}\right), \tag{25}$$

$$\tag{26}$$

and we can conclude that:

$$U_i(\theta_i'; \boldsymbol{\theta}_{-i}) > U_i(\theta_i; \boldsymbol{\theta}_{-i}) \tag{27}$$

$$\iff \underbrace{\sum_{j \neq i,0} [\log(p_j(\theta_j) - c_j(\theta_j)) + \beta V_j(\theta_j)] + \log(p_i(\theta_i') - c_i(\theta_i')) + \beta V_i(\theta_i') - \log\left(\sum_{j \neq i} e^{\beta V_j(\theta_j)} + e^{\beta V_i(\theta_i')}\right)}_{\Phi(\theta_i'; \boldsymbol{\theta}_{-i})}$$

$$> \underbrace{\sum_{j=1}^{N} \log(p_j(\theta_j) - c_j(\theta_j)) + \sum_{j=1}^{N} \beta V_j(\theta_j) - \log\left(\sum_{j=0}^{N} e^{\beta V_j(\theta_j)}\right)}_{\Phi(\boldsymbol{\theta})} \tag{28}$$

$$\iff \Phi(\theta_i'; \boldsymbol{\theta}_{-i}) > \Phi(\boldsymbol{\theta}). \tag{29}$$

C.2 PROOF OF THEOREM 2

We fix a test-time compute game $\mathcal{G}$ under perfect rationality and any pure Nash equilibrium $\boldsymbol{\theta}^\dagger \in \Theta^N$. Firstly, observe that it must be

$$s_1\left(V_1(\theta_1^\dagger); \boldsymbol{V}_{-1}(\boldsymbol{\theta}_{-1}^\dagger)\right) = 1. \tag{30}$$

Indeed, if the above were false, by definition of s under perfect rationality, there would exist a provider $i \neq 1$ such that $V_i(\theta_i^\dagger) > V_1(\theta_1^\dagger)$. Then, given that providers are indexed in decreasing order of their maximum possible offered values, we have that $V_1^* > V_i^*$ and hence provider 1 could best respond and strictly increase their utility, meaning that $\boldsymbol{\theta}^\dagger$ would not be a Nash equilibrium. Hence, in $\boldsymbol{\theta}^\dagger$, provider 1 offers a strictly higher value than any other provider. Furthermore, if $V_1(\boldsymbol{\theta}_1^\dagger) < V_2^*$, then, at least provider 2 could best respond by increasing the value they offer to V_2^*, serving all queries and obtaining strictly positive utility. Thus, in $\boldsymbol{\theta}^\dagger$ it holds that $V_1(\theta_1) > V_2^*$, and since provider 1 is playing a best response conditional on serving all queries, it must be that:

$$\theta_1^\dagger = \arg\max_{\boldsymbol{\theta}_1 \in \Theta} \{p_1(\theta_1) - c_1(\theta_1) \mid V_1(\theta_1) > V_2^*\} = \max\{\theta_1 | V_1(\theta_1) > V_2^*\}, \tag{31}$$

where the second equality holds because providers have profits that are increasing with the compute level. Next, we derive conditions on the compute of providers $i \neq 1$ such that $\boldsymbol{\theta}^\dagger$ is a Nash equilibrium. To this end, observe that since $V_1(\theta_1^\dagger) > V_2^*$, any provider $i \neq 1$ is unable to serve any queries, and hence has utility 0 independently of their deviations in $\boldsymbol{\theta}^\dagger$. We then distinguish two cases:

- If for any $\theta_1 \in \Theta$ we have that $V_1(\theta_1) > V_2^*$, then any provider $i \neq 1$ cannot increase their utilities, and provider 1 selecting their compute as in Eq. 31 is not able to improve their utility. We conclude that in this case, providers $i \neq 1$ selecting their compute arbitrarily leads to a pure Nash equilibrium (although the equilibrium is not strict).
- Suppose there exists $\theta_1 \in \Theta$ such that that $V_1(\theta_1) < V_2^*$ and that $p_1(\theta_1) - c_1(\theta_1) \geq p_1(\theta_1^\dagger) - c_1(\theta_1^\dagger)$ (note that we recover the previous case if $p_1(\theta_1) - c_1(\theta_1) < p_1(\theta_1^\dagger) - c_1(\theta_1^\dagger)$), with $\theta_1^\dagger$ as in Eq. 31. Then, provider 1 can strictly increase their utility by choosing θ_1 unless there exists another provider i with compute $\theta_i^\dagger$ such that $V_i(\theta_i^\dagger) > V_1(\theta_1)$.

We can summarize the above two cases by stating that at least one provider must offer a value higher than:

$$\max_{\theta_1 \in \Theta} \left\{ V_1(\theta_1) \middle| V_1(\theta_1) < V_2^* \quad \text{and} \quad p_1(\theta_1) - c_1(\theta_1) > p_1(\theta_1^\dagger) - c_1(\theta_1^\dagger) \right\} \tag{32}$$

with the constraint being lifted if the above set is empty. This concludes the proof.

C.3 PROOF OF THEOREM 3

The intuition for the proof is that the softmax converges to an indicator function for the maximum offered value as $\beta \to \infty$, and hence, the equilibrium computes should also converge. We now make this intuition precise, and assume $V_0 = 0$ in what follows for simplicity. We will leverage the following well-known fact about the softmax function, which we prove here for completeness:

Lemma 1. *Let $\boldsymbol{X} = (x_1, \dots, x_N) \in \mathbb{R}^N$ such that $x_1 > x_2 > \cdots > x_N$ and $\sigma^\beta(\boldsymbol{X})$ denote the softmax function with inverse temperature β. Then:*

$$|\sigma^\beta(\boldsymbol{X})_i - \mathbb{1}\{i = 1\}| \leq (N-1) \cdot \exp(-\beta \cdot (x_1 - x_2)), \quad \forall i = 1, \dots, N. \tag{33}$$

Proof. From the softmax definition, we have:

$$\begin{cases} \sigma^\beta(\boldsymbol{X})_1 = \dfrac{1}{1 + \sum_{j=2}^N \exp(-\beta(x_1 - x_j))} \\ \sigma^\beta(\boldsymbol{X})_i = \dfrac{\exp(-\beta(x_1 - x_i))}{1 + \sum_{j=2}^N \exp(-\beta(x_1 - x_j))}, i \geq 2. \end{cases} \tag{34}$$

Then, for $i \neq 1$ we have:

$$|\sigma^\beta(\boldsymbol{X})_i - \mathbb{1}\{i = 1\}| = \sigma^\beta(\boldsymbol{X})_i \leq \exp(-\beta(x_1 - x_i)) \leq (N-1) \cdot \exp(-\beta \cdot (x_1 - x_2)), \tag{35}$$

while for $i = 1$ we have:

$$|\sigma^\beta(\boldsymbol{X})_i - \mathbb{1}\{i = 1\}| = 1 - \sigma^\beta(\boldsymbol{X})_1 \leq \sum_{j \geq 2}^N \exp(-\beta \cdot (x_1 - x_j)) \leq (N-1) \cdot \exp(-\beta \cdot (x_1 - x_2)). \tag{36}$$

□

In our setting, we can use the above lemma with $\boldsymbol{X}$ corresponding to the vector of offered values $\boldsymbol{V}(\boldsymbol{\theta})$. More precisely, let us denote by s^β and s^∞ the market share functions under bounded rationality and perfect rrationality respectively (see Table 1). Then, for any $\boldsymbol{\theta} \in \Theta^N$, and any provider i, we have:

$$\left| s_i^\beta(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) - s_i^\infty(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) \right| \leq N \cdot \exp\left(-\beta \cdot (V_{\pi(1)}(\theta_{\pi(1)}) - V_{\pi(2)}(\theta_{\pi(2)})\right), \tag{37}$$

where π is the permutation ordering providers by their offered value. Taking the maximum among providers i and the (finitely many) compute levels they can select in the above expression, and letting $\delta_{min} > 0$ be the minimum gap in value offered by any two different providers (recall that we are assuming that providers always offer different values):

$$\max_{\boldsymbol{\theta}} \max_i \left| s_i^\beta(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) - s_i^\infty(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) \right| \leq N \cdot \exp\left(-\beta \cdot \delta_{min}\right). \tag{38}$$

Thus, we have that $s^\beta \to s^\infty$ with $\beta \to \infty$, where the convergence is uniform over Θ^N and the provider index i. Then, since for each provider i, the mapping $\theta_i \mapsto p_i(\theta_i) - c_i(\theta_i)$ takes finitely many values, we have that $\boldsymbol{U}^\beta \to \boldsymbol{U}^\infty$ with $\beta \to \infty$ uniformly over Θ^N and i, where $\boldsymbol{U}^\beta \colon \Theta^N \to \mathbb{R}^N$ denotes the function that maps a joint compute choice $\boldsymbol{\theta}$ to the provider's utilities as defined in Eq. 3 using the market allocation function s^β, and $\boldsymbol{U}^\infty$ is defined analogously.

To use the above convergence result, we let $\varepsilon > 0$ such that:

$$\varepsilon < \frac{1}{2} \min_{\boldsymbol{\theta}, \boldsymbol{\theta}'} \min_i \left\{ |U_i^\infty(\theta_i; \boldsymbol{\theta}_{-i}) - U_i^\infty(\theta_i'; \boldsymbol{\theta}'_{-i})| : \ |U_i^\infty(\theta_i; \boldsymbol{\theta}_{-i}) - U_i^\infty(\theta_i'; \boldsymbol{\theta}'_{-i})| > 0 \right\} \tag{39}$$

be at most half the minimum positive change in utility a provider can achieve in the game $\mathcal{G}$ under perfect rationality. Then, let β_0 be such that for any $\beta \geq \beta_0$, it holds that

$$\max_{\boldsymbol{\theta}} \max_i \left| s_i^\beta(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) - s_i^\infty(V_i(\theta_i); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i})) \right| \leq \varepsilon \tag{40}$$

and fix any pure Nash equilibrium $\boldsymbol{\theta}^\dagger$ of $\mathcal{G}$ for $\beta \geq \beta_0$. To show that $\boldsymbol{\theta}^\dagger$ is also an equilibrium of $\mathcal{G}$ under perfect rationality, we argue by contradiction, and suppose this premise is false. Then, there exists a provider i that can change their compute in the game $\mathcal{G}$ under perfect rationality and strictly increase their utility, that is, there exists $\theta_i' \in \Theta$ such that:

$$2\varepsilon \overset{(*)}{<} \left| U_i^\infty(\theta_i^\dagger; \boldsymbol{\theta}_{-i}^\dagger) - U_i^\infty(\theta_i'; \boldsymbol{\theta}_{-i}^\dagger) \right| \tag{41}$$

$$\leq \left| U_i^\infty(\theta_i^\dagger; \boldsymbol{\theta}_{-i}^\dagger) - U_i^\beta(\theta_i^\dagger; \boldsymbol{\theta}_{-i}^\dagger) \right| + \left| U_i^\beta(\theta_i^\dagger; \boldsymbol{\theta}_{-i}^\dagger) - U_i^\infty(\theta_i'; \boldsymbol{\theta}_{-i}^\dagger) \right| \tag{42}$$

$$\overset{(**)}{\leq} \varepsilon + \varepsilon \tag{43}$$

where $(*)$ follows from Eq. 39 and $(**)$ follows from Eq. 40. This is a contradiction, which proves that $\boldsymbol{\theta}^\dagger$ is also an equilibrium of $\mathcal{G}$ under perfect rationality.

C.4 PROOF OF THEOREM 4

The following is an auxiliary result that will be used to prove Theorem 4:

Lemma 2. *Let $x_1 > x_2 > \dots x_N$ and $y_1, \dots, y_N$ be real positive numbers, and denote both vectors by $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. Denote by $\sigma^\beta(\boldsymbol{X})$ the softmax with inverse temperature β. Then,*

$$\begin{cases} \sum_{i=1}^N y_i \cdot \sigma^\beta(\boldsymbol{X})_i \leq y_1 + e^{-\beta(x_1-x_2)} \cdot \sum_{i=2}^N (y_i - y_1) + \mathcal{O}\left(e^{-2\beta(x_1-x_2)}\right) \\ \sum_{i=1}^N y_i \cdot \sigma^\beta(\boldsymbol{X})_i \geq y_1 - e^{-\beta(x_1-x_2)} \cdot \sum_{i=2}^N (y_i + y_1) \end{cases} \tag{44}$$

Proof. From the softmax definition, we have:

$$\begin{cases} \sigma^\beta(\boldsymbol{X})_1 = \dfrac{1}{1 + \sum_{j=2}^N e^{-\beta(x_1-x_j)}} \\ \sigma^\beta(\boldsymbol{X})_i = \dfrac{e^{-\beta(x_1-x_i)}}{1 + \sum_{j=2}^N e^{-\beta(x_1-x_j)}} \leq e^{-\beta(x_1-x_i)}, \quad i \geq 2, \end{cases} \tag{45}$$

and,

$$\sum_{i=1}^N y_i \sigma^\beta(\boldsymbol{X})_i = y_1 + \sum_{i=2}^N (y_i - y_1) \cdot \sigma^\beta(\boldsymbol{X})_i. \tag{46}$$

We begin by showing that the first inequality in 44 holds. To this end, note that for $i \geq 2$, if $y_i - y_1 \geq 0$, then:

$$(y_i - y_1) \cdot \sigma^\beta(\boldsymbol{X})_i \leq (y_i - y_1) \cdot e^{-\beta(x_1-x_i)} \leq (y_i - y_1) \cdot e^{-\beta(x_1-x_2)}, \tag{47}$$

while if $y_i - y_1 < 0$, we have:

$$\begin{aligned} (y_i - y_1) \cdot \sigma^\beta(\boldsymbol{X})_i &= (y_i - y_1) \cdot \frac{e^{-\beta(x_1-x_i)}}{1 + \sum_{j=2}^N e^{-\beta(x_1-x_j)}} && (48) \\ &\leq (y_i - y_1) \cdot e^{-\beta(x_1-x_i)} \cdot \left(1 - \sum_{j=2}^N e^{-\beta(x_1-x_j)}\right) && (49) \\ &\leq (y_i - y_1) \cdot e^{-\beta(x_1-x_i)} + |y_i - y_1| \cdot e^{-\beta(x_1-x_i)} \cdot \left(\sum_{j=2}^N e^{-\beta(x_1-x_j)}\right) && (50) \\ &\leq (y_i - y_1) \cdot e^{-\beta(x_1-x_2)} + \underbrace{|y_i - y_1| \cdot e^{-\beta(x_1-x_i)} \cdot \left(\sum_{j=2}^N e^{-\beta(x_1-x_j)}\right)}_{\lambda^\beta(\boldsymbol{X},\boldsymbol{Y})} && (51) \end{aligned}$$

$$\tag{52}$$

where we have used that $1/(1+t) \geq 1 - t$ for any $t \geq 0$. Summing over all $i = 2, \dots, N$, and noting that $\lambda^\beta(\boldsymbol{X}, \boldsymbol{Y}) = \mathcal{O}\left(e^{-2\beta(x_1-x_2)}\right)$, we obtain:

$$\sum_{i=1}^N y_i \sigma^\beta(\boldsymbol{X})_i \leq y_1 + \sum_{i=2}^N (y_i - y_1) \cdot e^{-\beta(x_1-x_i)} + \mathcal{O}\left(e^{-2\beta(x_1-x_2)}\right), \tag{53}$$

which proves the first inequality in 44.

Similarly, we can show that the second inequality in 44 also holds. Indeed, from Eq. 46, we obtain:

$$\sum_{i=1}^{N} y_i \sigma^\beta(\boldsymbol{X})_i = y_1 - \sum_{i=2}^{N} (y_1 - y_i) \cdot \sigma^\beta(\boldsymbol{X})_i \tag{54}$$

$$\geq y_1 - \sum_{i=2}^{N} (y_1 + y_i) \cdot \sigma^\beta(\boldsymbol{X})_i \tag{55}$$

$$\geq y_1 - \sum_{i=2}^{N} (y_1 + y_i) \cdot e^{-\beta(x_1 - x_i)} \tag{56}$$

$$\geq y_1 - \sum_{i=2}^{N} (y_1 + y_i) \cdot e^{-\beta(x_1 - x_2)} \tag{57}$$

$$\tag{58}$$

□

We now prove Theorem 4. To this end, fix a test-time compute game $\mathcal{G}$ with $0 < \beta \leq \infty$ and a Nash equilibrium $\boldsymbol{\theta}^\dagger$. Further, for each provider i, denote by $\theta_i^* = \arg\max_\theta \{q_i(\theta) - c_i(\theta_i)\}$ their socially optimal compute, and let $\boldsymbol{\theta}^* = (\theta_i^*)_i$. Recall the definition of the price of anarchy in Eq. 4:

$$\mathrm{PoA}(\mathcal{G}) = \frac{\max_{\theta \in \Theta^N} \mathrm{W}(\boldsymbol{\theta})}{\mathrm{W}(\boldsymbol{\theta}^\dagger)} \geq \frac{\mathrm{W}(\boldsymbol{\theta}^*)}{\mathrm{W}(\boldsymbol{\theta}^\dagger)} = \frac{\overbrace{\sum_{i=1}^{N} s_i(V_i(\theta_i^*); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i}^*)) \cdot \mathrm{W}_i(\theta_i^*)}^{(\diamond)}}{\underbrace{\sum_{i=1}^{N} s_i(V_i(\theta_i^\dagger); \boldsymbol{V}_{-i}(\boldsymbol{\theta}_{-i}^\dagger)) \cdot \mathrm{W}_i(\theta_i^\dagger)}_{(\bullet)}}. \tag{59}$$

Denote by π and π^* the permutations ordering providers by their value at the equilibrium $\boldsymbol{\theta}^\dagger$ and at $\boldsymbol{\theta}^*$, respectively. Then, we can use Lemma 2 to lower bound the numerator and upper bound the denominator in the above.

Starting with the numerator $(\diamond)$, we take in Lemma 2 $\boldsymbol{X}$ to be the vector $(\boldsymbol{V}(\boldsymbol{\theta}^*), V_0)$ with $N+1$ components ordered by π^* (assuming without loss of generality that all providers can offer a value higher than the abstention value) and $\boldsymbol{Y}$ to be the vector $(\mathrm{W}_i(\theta_i^*), 0)$ with $N+1$ components ordered by π^*. Then, Lemma 2 implies:

$$(\diamond) \geq \mathrm{W}_{\pi^*(1)}(\theta^*_{\pi^*(1)}) \cdot \left(1 - e^{-\beta \Delta V^*}\right) - e^{-\beta \Delta V^*} \cdot \sum_{i \geq 2}^{N} \left(W_{\pi^*(i)}(\theta^*_{\pi^*(1)}) + W_{\pi^*(1)}(\theta^*_{\pi^*(i)})\right), \tag{60}$$

where $\Delta V^* = V_{\pi^*(1)}(\theta^*_{\pi^*(1)}) - V_{\pi^*(2)}(\theta^*_{\pi^*(2)})$ is the difference in values at $\boldsymbol{\theta}^*$ between the first and second providers.

Similarly, the term $(\bullet)$ can be upper-bounded by taking in Lemma 2 $\boldsymbol{X}$ to be the vector $(\boldsymbol{V}(\boldsymbol{\theta}^\dagger), 0)$ ordered by π (assuming without loss of generality that all providers can offer a value higher than the abstention value V_0 and taking $V_0 = 0$) and $\boldsymbol{Y}$ to be the vector $(\mathrm{W}_i(\theta_i^\dagger), 0)$ ordered by π. Then, we obtain:

$$(\bullet) \leq \mathrm{W}_{\pi(1)}(\theta^\dagger_{\pi(1)}) \cdot \left(1 - e^{-\beta \Delta V^\dagger}\right) + e^{-\beta \Delta V^\dagger} \cdot \sum_{i \geq 2}^{N} \left(W_{\pi(i)}(\theta^\dagger_{\pi(i)}) - W_{\pi(1)}(\theta^\dagger_{\pi(i)})\right) + \lambda^\beta, \tag{61}$$

where $\Delta V^\dagger = V_{\pi(1)}(\theta^\dagger_{\pi(1)}) - V_{\pi(2)}(\theta^\dagger_{\pi(2)})$ and $\lambda^\beta = \mathcal{O}\left(e^{-2\beta\Delta V^\dagger}\right)$ is the correction term defined as in Lemma 2.

Using such bounds on $(\diamond)$ and on $(\bullet)$, the price of anarchy satisfies:

$$\begin{aligned}\mathrm{PoA}(\mathcal{G}) &\geq \frac{\mathrm{W}_{\pi^*(1)}(\theta^*_{\pi^*(1)}) \cdot \left(1 - e^{-\beta\Delta V^*}\right) - e^{-\beta\Delta V^*} \cdot \sum_{i\geq 2}^N \left(W_{\pi^*(i)}(\theta^*_{\pi^*(i)}) + W_{\pi^*(1)}(\theta^*_{\pi^*(1)})\right)}{\mathrm{W}_{\pi(1)}(\theta^\dagger_{\pi(1)}) \cdot \left(1 - e^{-\beta\Delta V^\dagger}\right) + e^{-\beta\Delta V^\dagger} \cdot \sum_{i\geq 2}^N \left(W_{\pi(i)}(\theta^\dagger_{\pi(i)}) - W_{\pi(1)}(\theta^\dagger_{\pi(1)})\right) + \lambda^\beta} \\ &= \frac{\left(1 - e^{-\beta\Delta V^*}\right) - e^{-\beta\Delta V^*} \cdot \sum_{i\geq 2}^N \left(1 + \frac{\mathrm{W}_{\pi^*(i)}(\theta^*_{\pi^*(i)})}{\mathrm{W}^*})\right)}{\frac{\mathrm{W}_{\pi(1)}(\theta^\dagger_{\pi(1)})}{\mathrm{W}^*} \cdot \left(1 - e^{-\beta\Delta V^\dagger}\right) - e^{-\beta\Delta V^\dagger} \cdot \sum_{i\geq 2}^N \left(\frac{W_{\pi(1)}(\theta^\dagger_{\pi(1)}) - W_{\pi(i)}(\theta^\dagger_{\pi(i)})}{\mathrm{W}^*}\right) + \frac{\lambda^\beta}{\mathrm{W}^*}},\end{aligned} \tag{62}$$

where we have defined $\mathrm{W}^* \coloneqq \mathrm{W}_{\pi^*(1)}(\theta^*_{\pi^*(1)})$. The above can be further simplified by noting that:

$$\begin{aligned}\frac{\mathrm{W}_{\pi(1)}(\theta^\dagger_{\pi(1)}) - \mathrm{W}_{\pi(i)}(\theta^\dagger_{\pi(i)})}{\mathrm{W}^*} &= \frac{q_{\pi(1)}(\theta^\dagger_{\pi(1)}) - c_{\pi(1)}(\theta^\dagger_{\pi(1)}) - (q_{\pi(i)}(\theta^\dagger_{\pi(i)}) - c_{\pi(i)}(\theta^\dagger_{\pi(i)}))}{\mathrm{W}^*} \\ &= \frac{\overbrace{V_{\pi(1)}(\theta^\dagger_{\pi(1)}) - V_{\pi(i)}(\theta^\dagger_{\pi(i)})}^{\geq 0}}{\mathrm{W}^*} + \frac{p_{\pi(1)}(\theta^\dagger_{\pi(1)}) - c_{\pi(1)}(\theta^\dagger_{\pi(1)})}{\mathrm{W}^*} \\ &\quad - \frac{p_{\pi(i)}(\theta^\dagger_{\pi(i)}) - c_{\pi(i)}(\theta^\dagger_{\pi(i)})}{\mathrm{W}^*}) \\ &\geq \frac{p_{\pi(1)}(\theta^\dagger_{\pi(1)}) - c_{\pi(1)}(\theta^\dagger_{\pi(1)}) - \left(p_{\pi(i)}(\theta^\dagger_{\pi(i)}) - c_{\pi(i)}(\theta^\dagger_{\pi(i)})\right)}{\mathrm{W}^*} \coloneqq \frac{\Delta\rho^\dagger_i}{\mathrm{W}^*},\end{aligned}$$

where $\Delta\rho^\dagger_i$ represents the (normalized) difference in profits at equilibrium. Similarly, we can define:

$$\Delta\mathrm{W} \coloneqq \mathrm{W}_{\pi^*(1)}(\theta^*_{\pi^*(1)}) - \mathrm{W}_{\pi(1)}(\theta^\dagger_{\pi(1)})$$

where $\Delta\mathrm{W}_1$ represents the (normalized) difference in the contribution to the social welfare by the provider offering the highest value at equilibrium, and at compute $\boldsymbol{\theta}^*$. Using the two above simplifications in the bound in Eq. 62 and Taylor-expanding up to second order the function $t \mapsto$

$1/(1-t)$ around $t=0$, we obtain:

$$\text{PoA}(\mathcal{G}) \geq \frac{\left(1-e^{-\beta\Delta V^*}\right) - e^{-\beta\Delta V^*}\cdot\sum_{i\geq 2}^{N}\left(1+\frac{\mathrm{W}_{\pi^*(i)}(\theta^*_{\pi^*(i)})}{\mathrm{W}^*}\right)\right)}{1-\Delta\mathrm{W}/\mathrm{W}^* - e^{-\beta\Delta V^\dagger} - e^{-\beta\Delta V^\dagger}\cdot\sum_{i\geq 2}^{N}\Delta\rho_i^\dagger/\mathrm{W}^* + \mathcal{O}\left(\|e^{-\beta\Delta V^\dagger}, \Delta\mathrm{W}\|^2\right)} \tag{63}$$

$$\geq \left(\left(1-e^{-\beta\Delta V^*}\right) - e^{-\beta\Delta V^*}\cdot\sum_{i\geq 2}^{N}\left(1+\frac{\mathrm{W}_{\pi^*(i)}(\theta^*_{\pi^*(i)})}{\mathrm{W}^*}\right)\right) \tag{64}$$

$$\cdot\left(1+\Delta\mathrm{W}/\mathrm{W}^* + e^{-\beta\Delta V^\dagger} + e^{-\beta\Delta V^\dagger}\sum_{i\geq 2}^{N+1}\Delta\rho_i^\dagger/\mathrm{W}^*\right) + \mathcal{O}\left(\|e^{-\beta\Delta V^\dagger}, e^{-\beta\Delta V^*}, \Delta\mathrm{W}\|^2\right) \tag{65}$$

$$\geq \left(1-e^{-\beta\Delta V^*}\left(1+\cdot\sum_{i\geq 2}^{N}\left(1+\frac{\mathrm{W}_{\pi^*(i)}(\theta^*_{\pi^*(i)})}{\mathrm{W}^*}\right)\right)\right) \tag{66}$$

$$\cdot\left(1+\Delta\mathrm{W}/\mathrm{W}^* + e^{-\beta\Delta V^\dagger}\left(1+\sum_{i\geq 2}^{N+1}\Delta\rho_i^\dagger/\mathrm{W}^*\right)\right) + \mathcal{O}\left(\|e^{-\beta\Delta V^\dagger}, e^{-\beta\Delta V^*}, \Delta\mathrm{W}\|^2\right) \tag{67}$$

$$\geq 1+\Delta\mathrm{W}/\mathrm{W}^* + e^{-\beta\Delta V^\dagger}\cdot\left(1+\sum_{i\geq 2}^{N}\Delta\rho_i^\dagger/\mathrm{W}^*\right) - e^{-\beta\Delta V^*}\cdot\left(1+\sum_{i\geq 2}^{N}\left(1+\frac{\mathrm{W}_{\pi^*(i)}(\theta^*_{\pi^*(i)})}{\mathrm{W}^*}\right)\right) \tag{68}$$

$$+\mathcal{O}\left(\|e^{-\beta\Delta V^\dagger}, e^{-\beta\Delta V^*}, \Delta\mathrm{W}\|^2\right) \tag{69}$$

$$\geq 1+\frac{1}{\mathrm{W}^*}\cdot\left[\Delta\mathrm{W} + e^{-\beta\Delta V^\dagger}\cdot\left(\mathrm{W}^* + \sum_{i\geq 2}^{N}\Delta\rho_i^\dagger\right)\right. \tag{70}$$

$$\left. - e^{-\beta\Delta V^*}\cdot\left(1+\sum_{i\geq 2}^{N}\left(\mathrm{W}^* + \mathrm{W}_{\pi^*(i)}(\theta^*_{\pi^*(i)})\right)\right)\right] + \mathcal{O}\left(\|e^{-\beta\Delta V^\dagger}, e^{-\beta\Delta V^*}, \Delta\mathrm{W}\|^2\right). \tag{71}$$

To obtain the same form of in the statement of Theorem 4, we can let $\Delta V = \min(\Delta V^*, \Delta V^\dagger)$ and define:

$$f(\beta) = e^{-\beta\Delta V^\dagger}\cdot\left(\mathrm{W}^* + \sum_{i\geq 2}^{N}\Delta\rho_i^\dagger\right) - e^{-\beta\Delta V^*}\cdot\left(1+\sum_{i\geq 2}^{N}\left(\mathrm{W}^* + \mathrm{W}_{\pi^*(i)}(\theta^*_{\pi^*(i)})\right)\right) = \mathcal{O}\left(e^{-\beta\Delta V}\right)$$

C.5 PROOF OF THEOREM 5

Fix a game $\widetilde{\mathcal{G}}$, a provider i, and any compute levels $\boldsymbol{\theta}_{-i}$ and bid prices $\mathbf{p}_{-i}$. We will prove that, for provider i, selecting $\theta_i = \theta_i^*$ and $p_i = c_i(\theta_i^*)$ is a dominant strategy. To this end, we first show that given a fixed θ_i, the provider maximizes their utility by bidding $p_i = c_i(\theta_i)$, and then show that $\theta_i = \theta_i^*$ is their best choice of compute. Recall that the utility of provider i is (Eq. 9):

$$\begin{aligned} U_i(\theta_i, p_i; \boldsymbol{\theta}_{-i}, \mathbf{p}_{-i}) &= (P(\boldsymbol{\theta}, \mathbf{p}) - c_i(\theta_i)) \cdot \mathbb{1}\left\{V_i(\theta_i, p_i) > \max_{j \neq i} V_j(\theta_j, p_j)\right\} \\ &= \left(q_{\pi(1)}(\theta_{\pi(1)}) - c_i(\theta_i) - (q_{\pi(2)}(\theta_{\pi(2)}) - p_{\pi(2)})\right) \cdot \mathbb{1}\left\{i = \pi(1)\right\}. \end{aligned}$$

Bidding generation cost is optimal. Let us first prove that, for a fixed θ_i, bidding $p_i = c_i(\theta_i)$ is always optimal. To this end, we assume that the provider bids $p_i = c_i(\theta_i)$ and show that no deviation to a different price bid can increase their utility. Consider first that provider i wins the bid when bidding $(q_i(\theta_i), c_i(\theta_i))$. Thus, it must be that $i = \pi(1)$ and $q_i(\theta_i) - c_i(\theta_i) > q_{\pi(2)}(\theta_{\pi(2)}) - p_{\pi(2)}$. Then, consider a deviation where provider i bids a different price $p_i \neq c_i(\theta_i)$. With their new bid $(q_i(\theta_i), p_i)$, they either lose the auction, in which case they strictly decrease their utility to 0, or they are still winning the auction. However, in the latter case, since conditional on winning, their utility is independent of p_i, we conclude that this deviation maintains their utility, and hence the provider cannot profit from it.

Now, suppose that provider i initially loses the auction by bidding $(q_i(\theta_i), c_i(\theta_i))$, which implies that:

$$q_i(\theta_i) - c_i(\theta_i) < q_{\pi(2)}(\theta_{\pi(2)}) - p_{\pi(2)}. \tag{72}$$

Then, any deviation where they bid a higher $p_i > c_i(\theta_i)$ will further decrease their offered value, and hence they will still lose the auction and maintain their null utility. In case the provider deviates to a lower price $p_i < c_i(\theta_i)$ and wins the auction by doing so, their new utility will be:

$$U_i(\theta_i, p_i; \boldsymbol{\theta}_{-i}, \mathbf{p}_{-i}) = q_i(\theta_i) - c_i(\theta_i) - \left(q_{\pi(2)}(\theta_{\pi(2)}) - p_{\pi(2)}\right) < 0, \tag{73}$$

meaning that the deviation is not profitable. Hence, we conclude that bidding $p_i = c_i(\theta_i)$ is always optimal.

Bidding the socially optimal compute is dominant. We now prove that bidding with $\theta_i = \theta_i^*$ is dominant for provider i. To this end, we assume that for any compute θ_i, provider i selects their optimal price $p_i = c_i(\theta_i)$. Then, the utility of the provider can be written as:

$$\begin{aligned} U_i(\theta_i, c_i(\theta_i); \boldsymbol{\theta}_{-i}, \mathbf{p}_{-i}) = &\left(q_{\pi(1)}(\theta_{\pi(1)}) - c_i(\theta_i)\right) \cdot \mathbb{1}\left\{i = \pi(1)\right\} \\ &- (q_{\pi(2)}(\theta_{\pi(2)}) - p_{\pi(2)}) \cdot \mathbb{1}\left\{i = \pi(1)\right\} \end{aligned}$$

Since, conditional on provider i winning the bid, the second term in the above expression does not depend on the action of provider i, their optimal compute level is to select:

$$\begin{aligned} \arg\max_{\theta_i \in \Theta} \left\{q_i(\theta_i) - c_i(\theta_i) | i = \pi(1)\right\} &= \arg\max_{\theta_i \in \Theta} \left\{q_i(\theta_i) - c_i(\theta_i) \middle| q_i(\theta_i) - c_i(\theta_i) > q_{\pi(2)}(\theta_{\pi(2)}) - p_{\pi(2)}\right\} && (74) \\ &= \arg\max_{\theta_i \in \Theta} \left\{q_i(\theta_i) - c_i(\theta_i)\right\} && (75) \\ &= \theta_i^*, && (76) \end{aligned}$$

which proves the claim.

D ADDITIONAL EXPERIMENTAL DETAILS

Here, we provide additional experimental details for our empirical evaluation in Section 5. We have released the complete code and implementation at `https://github.com/Human-Centric-Machine-Learning/strategic-ttc`.

Hardware setup. Our experiments are executed on a compute server equipped with 2 $\times$ Intel Xeon Gold 5317 CPU, 1,024 GB main memory, and 2 $\times$ A100 Nvidia Tesla GPU (80 GB, Ampere Architecture). In each experiment, a single Nvidia A100 GPU is used.

Models. We use the following LLMs in our experiments. From the `Llama` family, we use `Llama-3-8B-Instruct`, `Llama-3.1-8B-Instruct`, `Llama-3.2-1B-Instruct`, and `Llama-3.2-3B-Instruct`. From the `Qwen` family, we use `Qwen-2-0.5B-Instruct`, `Qwen-2-1.5B-Instruct`, `Qwen-2-7B-Instruct`, `Qwen-2.5-3B-Instruct`, and `Qwen-2.5-7B-Instruct`. Finally, we include three reasoning models distilled from `DeepSeek-R1`: `DeepSeek-R1-Distill-Llama-8B`, `DeepSeek-R1-Distill-Qwen-1.5B`, and `DeepSeek-R1-Distill-Qwen-7B`. Additionally, when using best-of-n sampling, we use the `ArmoRM-Llama3-8B-v0.1` reward model to score the generated outputs by the above models. We obtain all the models from the publicly available Hugging Face repository.[6]

Datasets. We use three datasets to evaluate the performance of all LLMs and simulate test-time compute games: `GPQA` Rein et al. (2024), a multiple-choice STEM question-answering benchmark; `GS8MK` Cobbe et al. (2021), a mathematical benchmark containing grade-school level problems; and `AIME` Zhang & Math-AI (2025), a mathematical reasoning benchmark with problems from the American Invitational Mathematics Examination. All datasets have, for each query, the corresponding verifiable ground-truth that we use to evaluate the accuracy of the models. All datasets are obtained from the publicly available Hugging Face repository.[7][8][9]

Generation details. When generating model outputs and evaluating them on the above datasets, we adhere to the temperature settings recommended in the official Hugging Face model cards. That is, temperature 0.6 for the `Llama` family and the corresponding distilled reasoning model, and temperature 0.7 for the `Qwen` family and the corresponding distilled reasoning models. We disable top-p and top-k sampling when generating model outputs. To ensure robust evaluation and enable bootstrap resampling to estimate uncertainties, we generated 128 candidate responses per query for non-reasoning models and 32 for reasoning models. For the simulations, we used subsets of these pools (up to $\theta = 64$ for non-reasoning and the full distribution for reasoning quantiles). We report 95% bootstrapped confidence intervals for the accuracies of all models. Our prompts and answer verification pipelines are adapted from established evaluation frameworks, specifically OpenCompass[10], EvalScope[11], and the LM Evaluation Harness[12].

Test-time compute methods. In our experiments, we consider the following three test-time compute methods, each used to simulate a different test-time compute game $\mathcal{G}$ in Section 5:

- **Majority voting.** For each query, each model generates $\theta = \{2^0, \ldots, 2^6\}$ independent outputs. The final response of the model is taken as the most frequent response across the θ generations. The prices $p(\theta)$ and costs $c(\theta)$ are computed based on the total number of generated tokens across all θ outputs, averaged across queries.
- **Best-of-n (Chow et al., 2025).** For each query, the model generates $\theta = \{2^0, \ldots, 2^6\}$ independent outputs. The final response of the model is taken as the highest-scored response according to the scoring model `ArmoRM-Llama3-8B-v0.1`. The prices $p(\theta)$ and costs

[6] `https://huggingface.co`
[7] `https://huggingface.co/datasets/Idavidrein/gpqa`
[8] `https://huggingface.co/datasets/openai/gsm8k`
[9] `https://huggingface.co/datasets/Maxwell-Jia/AIME_2024`
[10] `https://github.com/open-compass/opencompass`
[11] `https://github.com/modelscope/evalscope`
[12] `https://github.com/EleutherAI/lm-evaluation-harness`

$c(\theta)$ are computed based on the total number of generated tokens across all θ outputs, averaged across queries.

- **Chain-of-though (Wei et al., 2022).** Since the Hugging Face API does not allow explicit control over the level of test-time compute or reasoning used for chain-of-thought for the models we consider, we adopt the following approach. For each query, we generate 32 independent model outputs and then group these outputs into five quantile-based bins according to the number of reasoning tokens generated. In this process, we discard outputs where the models do not generate reasoning tokens. More concretely, the first bin contains outputs whose reasoning-token counts fall within the lowest 20-th percentile, and analogously for the remaining bins. Then, we compute the accuracy for each bin and average it over queries.

Test-time compute prices and costs. In our experiments in Section 5, we instantiate the test-time compute games $\mathcal{G}$ with each provider serving a different LLM. To determine the per-output token price for each provider/model, we refer to the Hugging Face list of inference providers[13] and, for each LLM, compute the average token price across the listed providers that offer access to that model. The resulting prices per million output tokens are as follows: \$0.1455, \$0.1245, \$0.10, and \$0.08 for `Llama-3-8B-Instruct`, `Llama-3.1-8B-Instruct`, `Llama-3.2-1B-Instruct`, and `Llama-3.2-3B-Instruct`, respectively; \$0.10, \$0.10, \$0.20, \$0.065, and \$0.1465 for `Qwen-2-0.5B`, `Qwen-2-1.5B`, `Qwen-2-7B`, `Qwen-2.5-3B`, and `Qwen-2.5-7B`, respectively; and \$0.125, \$0.10, and \$0.175 for the models `R1-D-Llama-8B`, `R1-D-Qwen-1.5B`, and `R1-D-Qwen-7B`, respectively. Then, we determine the per-output token generation cost by assuming that the per-token prices are 25% higher than the per-token costs, which simply corresponds to the per-token margin of providers. Lastly, for each player (provider) i in the game $\mathcal{G}$, each test-time method with compute level θ, and each dataset, we separately compute the quantities $p_i(\theta)$ (and correspondingly $c_i(\theta_i)$ using the margin 25%) by considering the average number of output tokens across all model outputs in the dataset and multiplying it by the per-token price.

Converting accuracies to user value. To compute the average accuracy of each model (measure as a percentage) into a quantity comparable to a price (measured in \$, we consider that each (average) percentage point of accuracy is worth $\{\$0.008, \$0.02, \$0.05\}$, respectively for `GSM8K`, `GPQA` and `AIME`. This choice is motivated by assigning a higher monetary value to accuracy the harder the dataset is; as seen in Appendix E.1, models typically perform best on `GSM8K` and worst on `AIME`, with `GPQA` falling in the middle. We have conducted experiments with different prices per accuracy point and have obtained similar results.

Dynamics of test-time compute games. To simulate the dynamics of a test-time compute game, we proceed as follows. First, we consider that all providers start with their lowest test-time compute level, that is, $\theta_i^1 = \min_{\theta \in \Theta} \theta, \forall i \in [N]$, which corresponds to using a single sample for best-of-n and majority voting, and generating responses in the first 20-th percentile in terms of reasoning tokens for chain-of-thought, see Appendix D. Then, at each iteration t, we update the providers' compute levels from $\boldsymbol{\theta}^t$ to $\boldsymbol{\theta}^{t+1}$ by randomly selecting a provider $i_t \in [N]$ who does not have maximum utility when the others select $\boldsymbol{\theta}^t_{-i_t}$. We keep $\boldsymbol{\theta}^{t+1}_{-i_t} = \boldsymbol{\theta}^t_{-i_t}$ fixed, and take $\theta^{t+1}_{i_t}$ to be the smallest compute that is larger than $\theta^t_{i_t}$ if $\theta^t_{i_t} \prec \arg\max_\theta U_{i_t}(\theta; \boldsymbol{\theta}^t_{-i_t})$, or to be the largest compute that is smaller than $\theta^t_{i_t}$ if $\theta^t_{i_t} \succ \arg\max_\theta U_{i_t}(\theta; \boldsymbol{\theta}^t_{-i_t})$. If all providers are already maximizing their utilities, the better-response dynamics have reached a Nash equilibrium. At each time step, we use the values offered by providers to determine their market share and compute the price of anarchy according to Eq. 7, see Appendix E.2.

[13] `https://huggingface.co/docs/inference-providers/index`, consulted on December 30, 2025

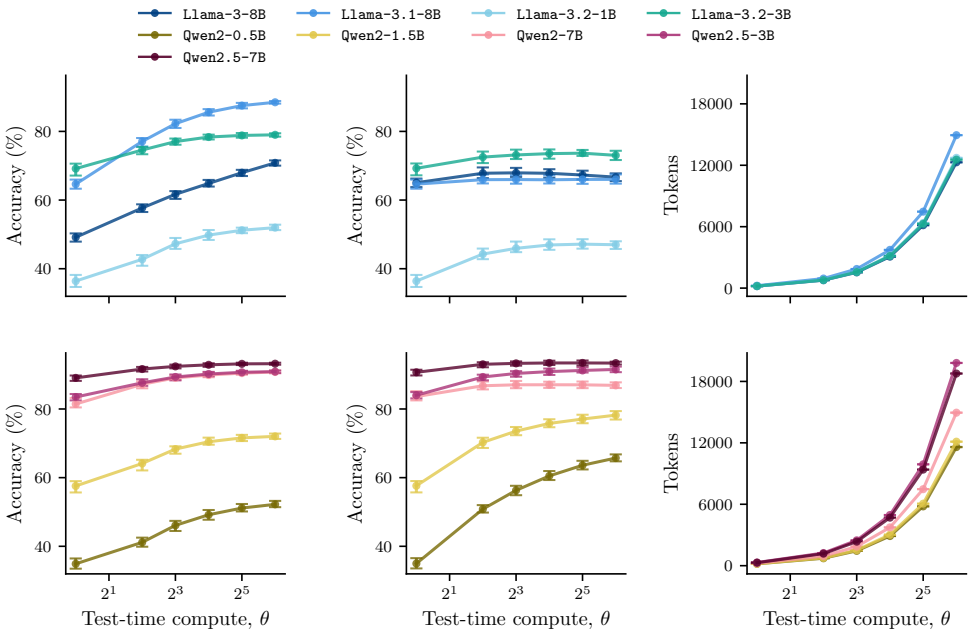
E ADDITIONAL EXPERIMENTAL RESULTS

This section contains additional experimental results that complement those discussed in Section 5. In Appendix E.1, we summarize the results (accuracy and number of generated tokens) of evaluating the LLMs served by each provider in the test-time compute games on `GSM8K`, `AIME`, and `GPQA`. Appendix E.2 shows additional outcomes of test-time compute games across all datasets and test-time compute methods.

E.1 MODEL EVALUATION

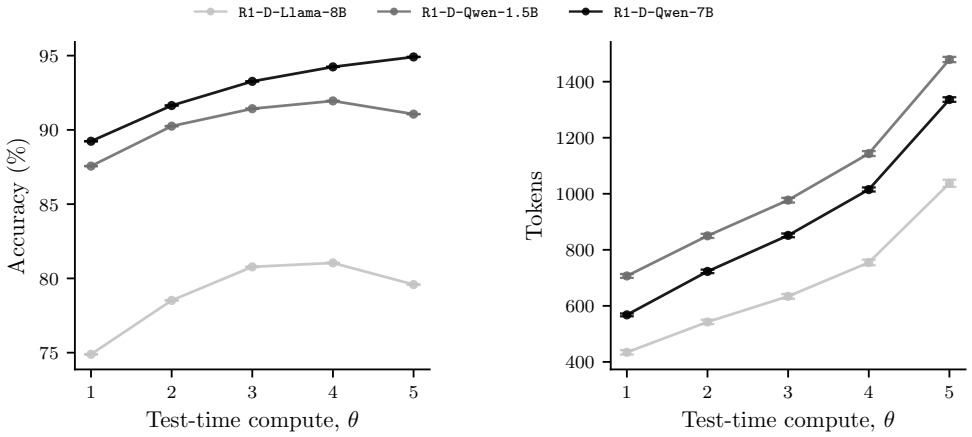
Here, we report, for each dataset (`GSM8K`, `AIME`, and `GPQA`) and each test-time compute method (majority voting and best-of-n for non-reasoning models, and chain-of-thought for reasoning models), the average accuracy (proportion of correct response) of the models as a function of the test-time compute θ. We also report the average number of generated tokens to obtain the response to each query. See Appendix D for more details regarding the generation of model outputs.

E.1.1 MODEL EVALUATION ON GSM8K



(a) Majority Vote (b) Best-of-n (c) Number of Tokens

Figure 2: **Accuracy of `Llama` and `Qwen` models on GSM8K using majority voting and best-of-n.** Panels (a) and (b) show the average accuracy of various LLMs from the `Llama` and `Qwen` families over questions from the GSM8K dataset, where the responses of the models are obtained using majority voting or best-of-n, respectively. Panel (c) shows, as a function of the number of samples used to generate the response, the total number of tokens that the models generate to obtain the response to each question, averaged across questions. We show 95% confidence intervals obtained by bootstrapping 50 times.

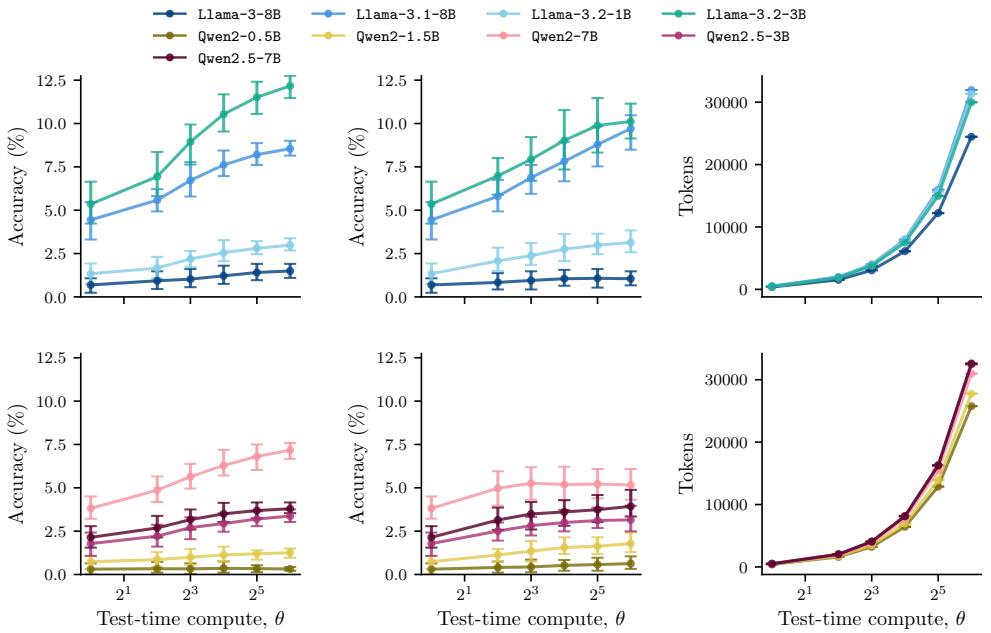


(a) Accuracy vs Reasoning Effort

(b) Tokens vs Reasoning Effort

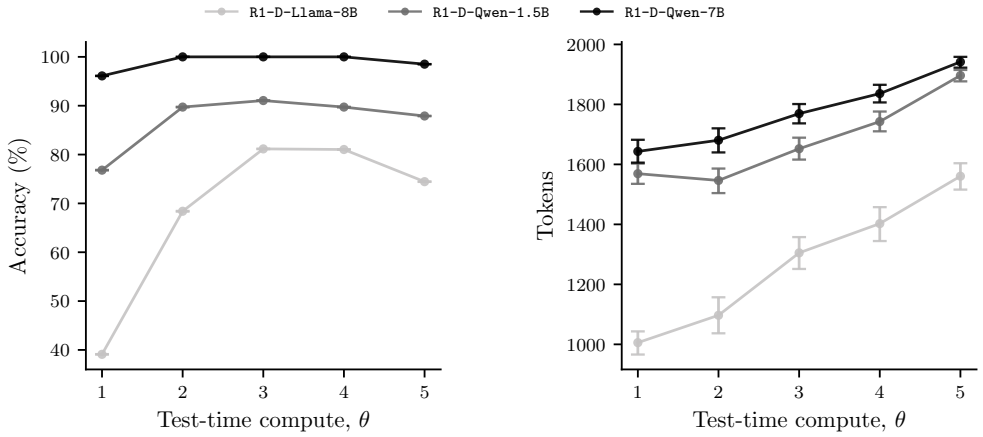
Figure 3: **Accuracy of reasoning models distilled from `DeepSeek-R1` on GSM8K using chain-of-thought.** Panel (a) shows the average accuracy of various reasoning models from the `Llama` and `Qwen` families distilled from `DeepSeek-R1` over questions from the GSM8K dataset. Here, the reasoning effort is defined by binning the model outputs into quantiles based on the number of reasoning tokens (see Appendix D). Panel (c) shows, as a function of the reasoning effort, the total number of tokens (including reasoning and non-reasoning tokens) that the models generate as a response to each question, averaged across questions. We show 95% confidence intervals obtained by bootstrapping 50 times. Refer to Appendix D for further details regarding the evaluation of the models.

E.1.2 Model Evaluation on AIME



(a) Majority Vote (b) Best-of-n (c) Number of Tokens

Figure 4: **Accuracy of `Llama` and `Qwen` models on AIME using majority voting and best-of-n.** Panels (a) and (b) show the average accuracy of various LLMs from the `Llama` and `Qwen` families over questions from the AIME dataset, where the responses of the models are obtained using majority voting or best-of-n, respectively. Panel (c) shows, as a function of the number of samples used to generate the response, the total number of tokens that the models generate to obtain the response to each question, averaged across questions. Here, we compute the accuracies for majority voting, and best-of-n are computed across the same outputs, and hence both majority voting and best-of-n generate the exact same number of average tokens. We show 95% confidence intervals obtained by bootstrapping 50 times. Refer to Appendix D for further details regarding the evaluation of the models.

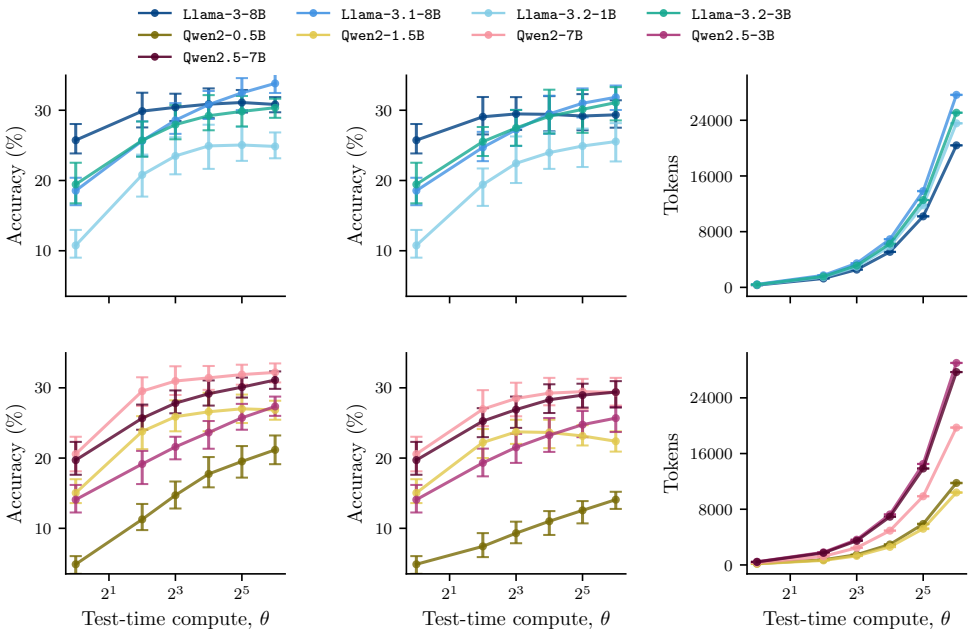


(a) Accuracy vs Reasoning Effort

(b) Tokens vs Reasoning Effort

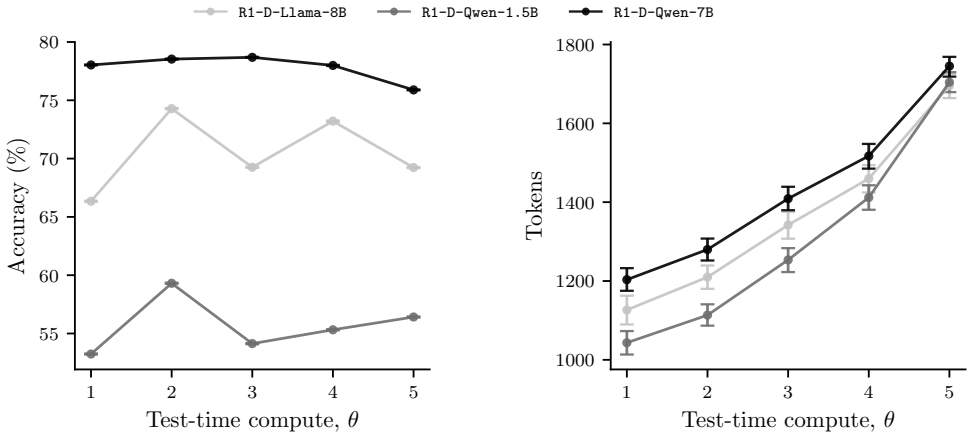
Figure 5: **Accuracy of reasoning models distilled from `DeepSeek-R1` on AIME using chain-of-thought.** Panel (a) shows the average accuracy of various reasoning models from the `Llama` and `Qwen` families distilled from `DeepSeek-R1` over questions from the AIME dataset. Here, the reasoning effort is defined by binning the model outputs into quantiles based on the number of reasoning tokens (see Appendix D). Panel (c) shows, as a function of the reasoning effort, the total number of tokens (including reasoning and non-reasoning tokens) that the models generate as a response to each question, averaged across questions. We show 95% confidence intervals obtained by bootstrapping 50 times. Refer to Appendix D for further details regarding the evaluation of the models.

E.1.3 MODEL EVALUATION ON GPQA



(a) Majority Vote (b) Best-of-n (c) Number of Tokens

Figure 6: **Accuracy of `Llama` and `Qwen` models on GPQA using majority voting and best-of-n.** Panels (a) and (b) show the average accuracy of various LLMs from the `Llama` and `Qwen` families over questions from the GPQA dataset, where the responses of the models are obtained using majority voting or best-of-n, respectively. Panel (c) shows, as a function of the number of samples used to generate the response, the total number of tokens that the models generate to obtain the response to each question, averaged across questions. Here, we compute the accuracies for majority voting, and best-of-n are computed across the same outputs, and hence both majority voting and best-of-n generate the exact same number of average tokens. We show 95% confidence intervals obtained by bootstrapping 50 times. Refer to Appendix D for further details regarding the evaluation of the models.



(a) Accuracy vs Reasoning Effort

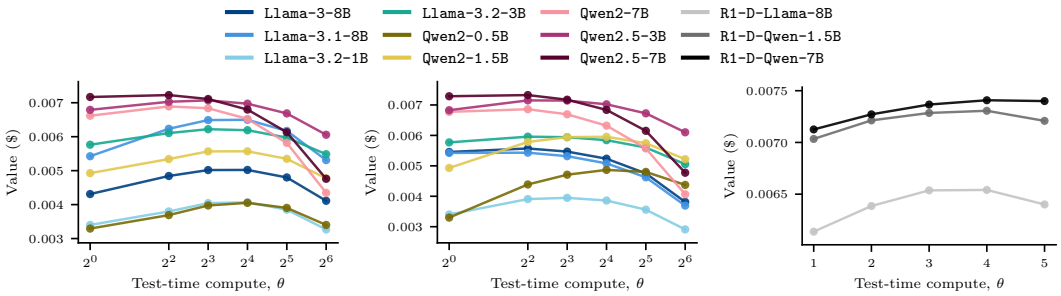
(b) Tokens vs Reasoning Effort

Figure 7: **Accuracy of reasoning models distilled from `DeepSeek-R1` on GPQA using chain-of-thought.** Panel (a) shows the average accuracy of various reasoning models from the `Llama` and `Qwen` families distilled from `DeepSeek-R1` over questions from the GPQA dataset. Here, the reasoning effort is defined by binning the model outputs into quantiles based on the number of reasoning tokens (see Appendix D). Panel (c) shows, as a function of the reasoning effort, the total number of tokens (including reasoning and non-reasoning tokens) that the models generate as a response to each question, averaged across questions. We show 95% confidence intervals obtained by bootstrapping 50 times. Refer to Appendix D for further details regarding the evaluation of the models.

E.2 Dynamics and Equilibria of Test-Time Compute Games

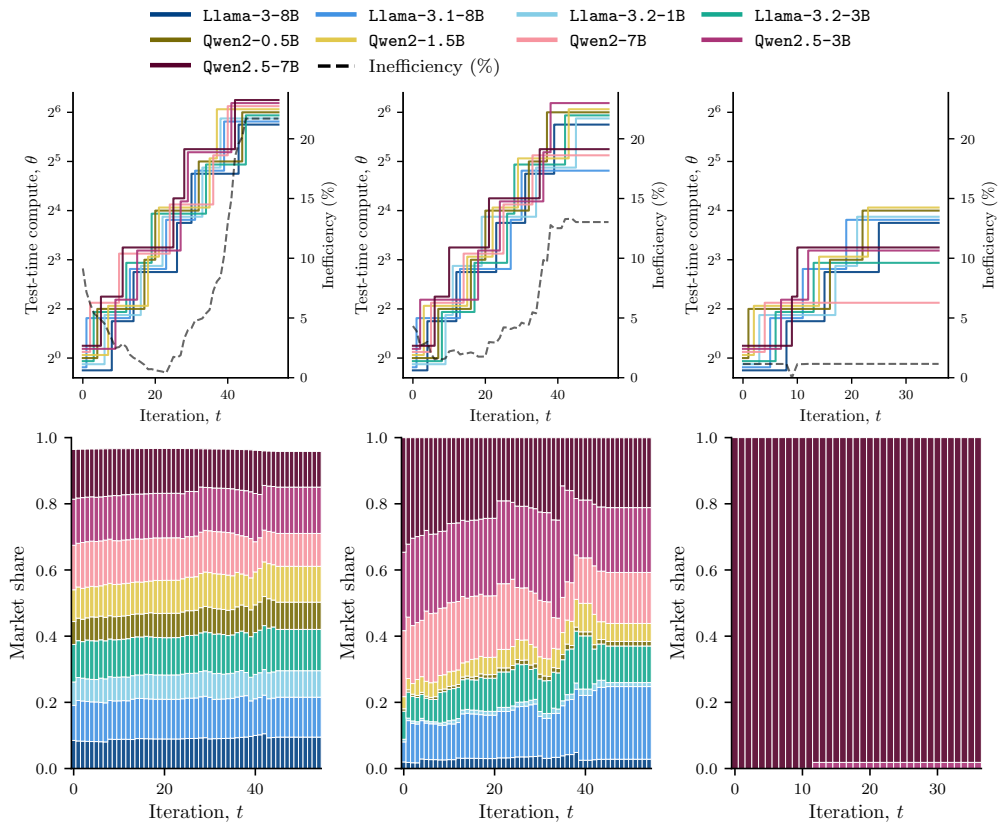
Here, for each dataset (GSM8K, AIME, and GPQA) and each test-time compute method (majority voting and best-of-n for non-reasoning models, and chain-of-thought for reasoning models), we report: (i) the value offered by providers as a function of their test time compute, (ii) the compute levels and market shares when providers better-respond to each other, as a function of the iteration t, (iii) the evolution of the potential Φ (Eq. 6) when providers better-respond, and (iv), the inefficiency (PoA($\mathcal{G}$)-1) at equilibrium for each game as a function of user's rationality β. We also report the inefficiency at each time step, which corresponds to $\max_{\boldsymbol{\theta}} \mathrm{W}(\boldsymbol{\theta})/\mathrm{W}(\boldsymbol{\theta}^t)$ whenever $\boldsymbol{\theta}^t$ is not an equilibrium.

E.2.1 Test-Time Compute Equilibria on GSM8K



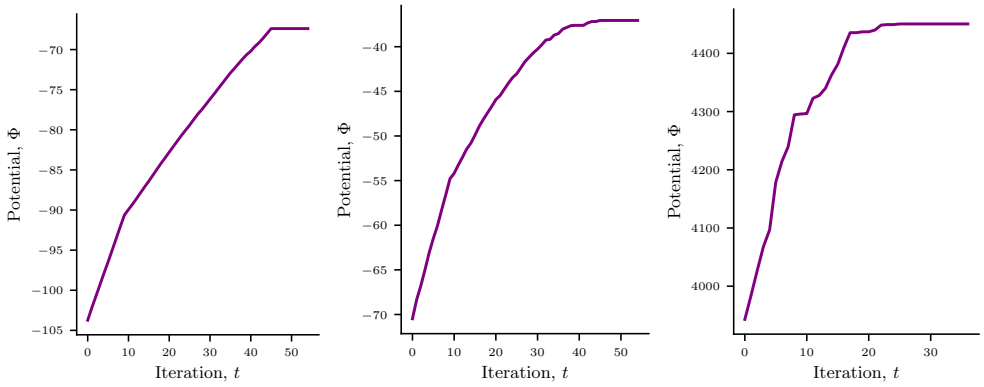
(a) Majority voting (b) Best-of-n (c) Chain-of-thought

Figure 8: **User values offered by providers in a test-time compute game on GSM8K.** The figure shows the user values $V_i(\theta)$ offered by providers in a test-time compute game $\mathcal{G}$, as a function of their test-time compute θ. Panel (a) and (b) correspond to games with $N = 9$ providers serving non-reasoning models from the Llama and Qwen families, where providers use, respectively, majority voting and best-of-n across θ samples. Panel (c) corresponds to a game with $N = 3$ providers serving reasoning models distilled from DeepSeek-R1, where θ represents reasoning effort, defined by binning the model outputs into quantiles based on the number of reasoning tokens (see Appendix D). In both games, providers serve queries Q from the GSM8K dataset, we set $\beta = 1000$ and consider that each (average) percentage point of accuracy offers a value of \$0.008 to the users.



(a) $\beta = 2 \cdot 10^2$ (b) $\beta = 10^3$ (c) $\beta = 10^5$

Figure 9: **Dynamics of a test-time compute game using majority-voting.** The figure shows, for different levels of user rationality β, the better-response dynamics of a test-time compute game $\mathcal{G}$ where $N = 9$ providers sequentially select a test-time compute level that increases their utility. The upper panels show the compute levels θ selected by each provider and the resulting market inefficiency ($\text{PoA}(\mathcal{G}) - 1$), and the lower panels show the market share of each provider. Here, all providers use majority-voting across θ samples as their test-time compute method to serve queries Q from the `GSM8K` dataset. We consider that providers operate with a margin of 25% between per-token price and per-token cost, and that each (average) percentage point of accuracy offers a value of $\$0.008$ to the users.



(a) Potential ($\beta = 2 \cdot 10^2$) (b) Potential ($\beta = 10^3$) (c) Potential ($\beta = 10^5$)

Figure 10: **Potential of a test-time compute game using majority-voting.** The figure shows, for different levels of user rationality β, the evolution of the potential Φ (see Eq. 6) in the test-time compute games in Figure 9 where $N = 9$ providers sequentially select a test-time compute level that increases their utility.

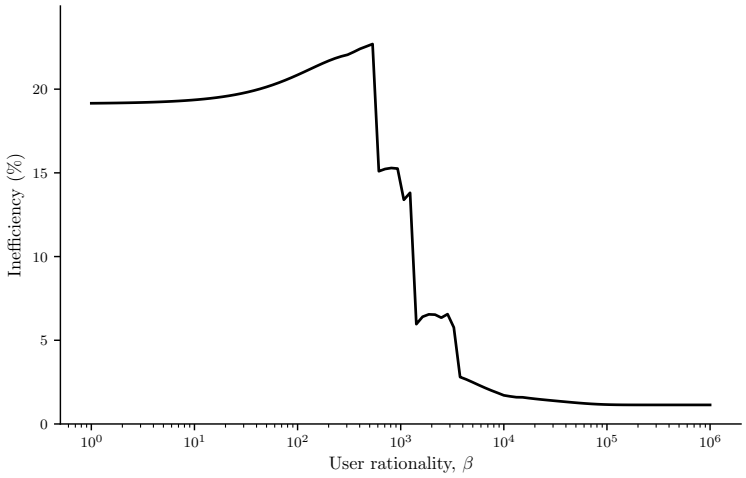
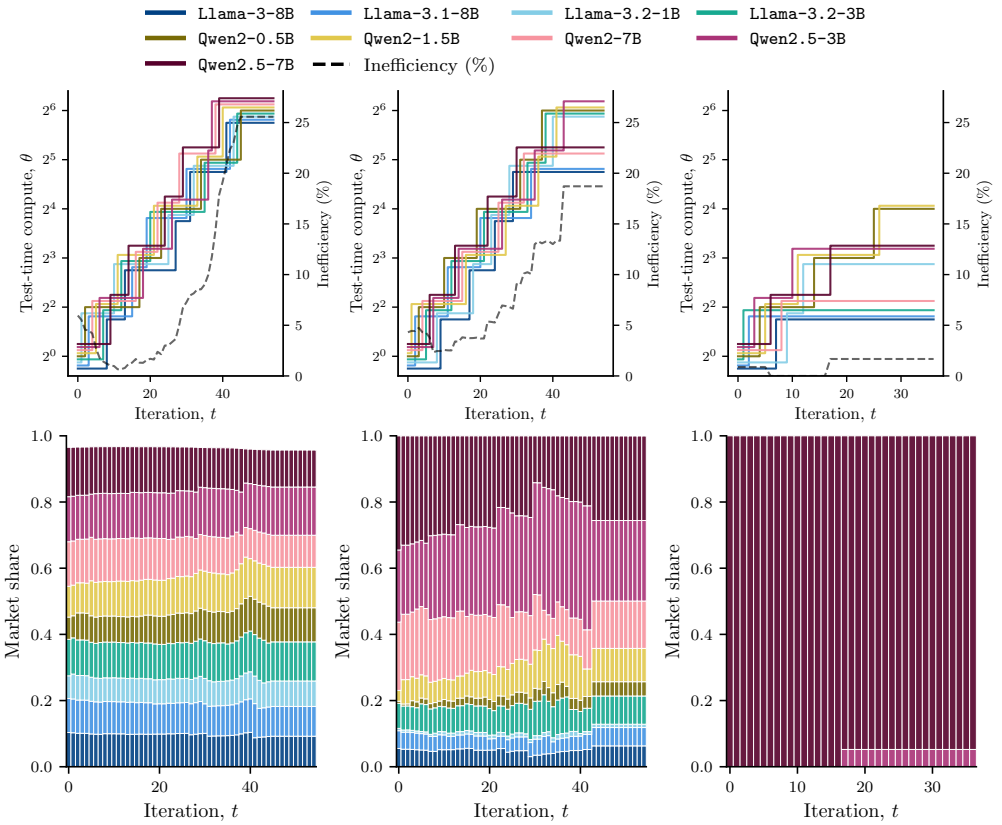
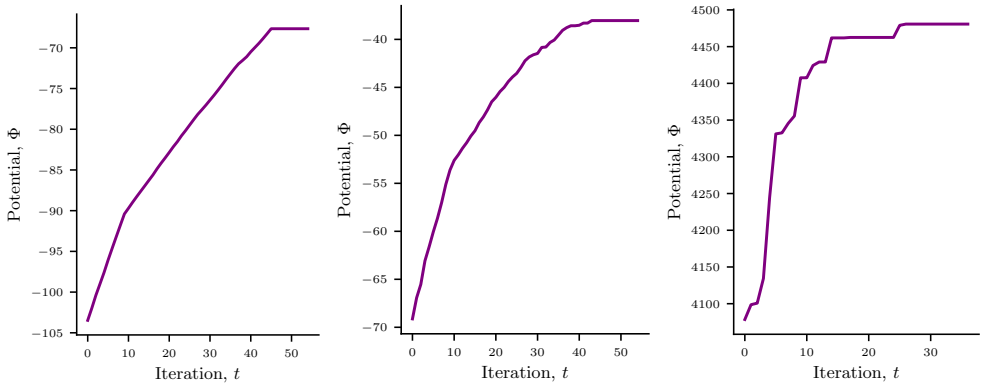


Figure 11: **Inefficiency of a test-time compute game using majority-voting.** The figure shows, as a function of users' rationality β, the inefficiency ($\text{PoA}(\mathcal{G}) - 1$) of the test-time compute game in Figure 9, where $N = 9$ providers sequentially select a test-time compute level that increases their utility.



(a) $\beta = 2 \cdot 10^2$ (b) $\beta = 10^3$ (c) $\beta = 10^5$

Figure 12: **Better-response dynamics of a test-time compute game using best-of-n.** The upper panels show, for varying levels of user rationality β, better-response dynamics when providers serving models from the `Llama` and `Qwen` families use best-of-n to serve queries from the GSM8K dataset. The solid colored lines (left y-axis) represent the test-time compute θ selected by each provider at each iteration, corresponding to the number of samples used for best-of-n. The dashed black line (right y-axis) tracks the Market Inefficiency, defined as $(\text{PoA} - 1) \times 100$ (see Eq. 7). The lower panels show the evolution of the market share at each time step of the better-response dynamics. The initial compute level $\boldsymbol{\theta}^1$ is taken as the lowest possible compute. We apply a small vertical jitter to the strategy lines to distinguish overlapping providers and take a fixed profit margin of 25%.



(a) Potential ($\beta = 2 \cdot 10^2$) (b) Potential ($\beta = 10^3$) (c) Potential ($\beta = 10^5$)

Figure 13: **Potential of a test-time compute game using best-of-n.** The figure shows, for different levels of user rationality β, the evolution of the potential Φ (see Eq. 6) in the test-time compute games in Figure 12 where $N = 9$ providers sequentially select a test-time compute level that increases their utility.

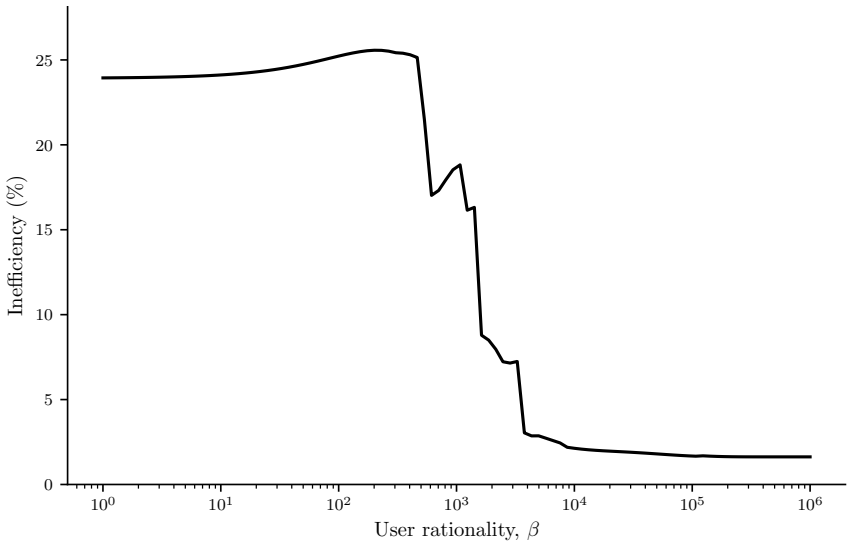
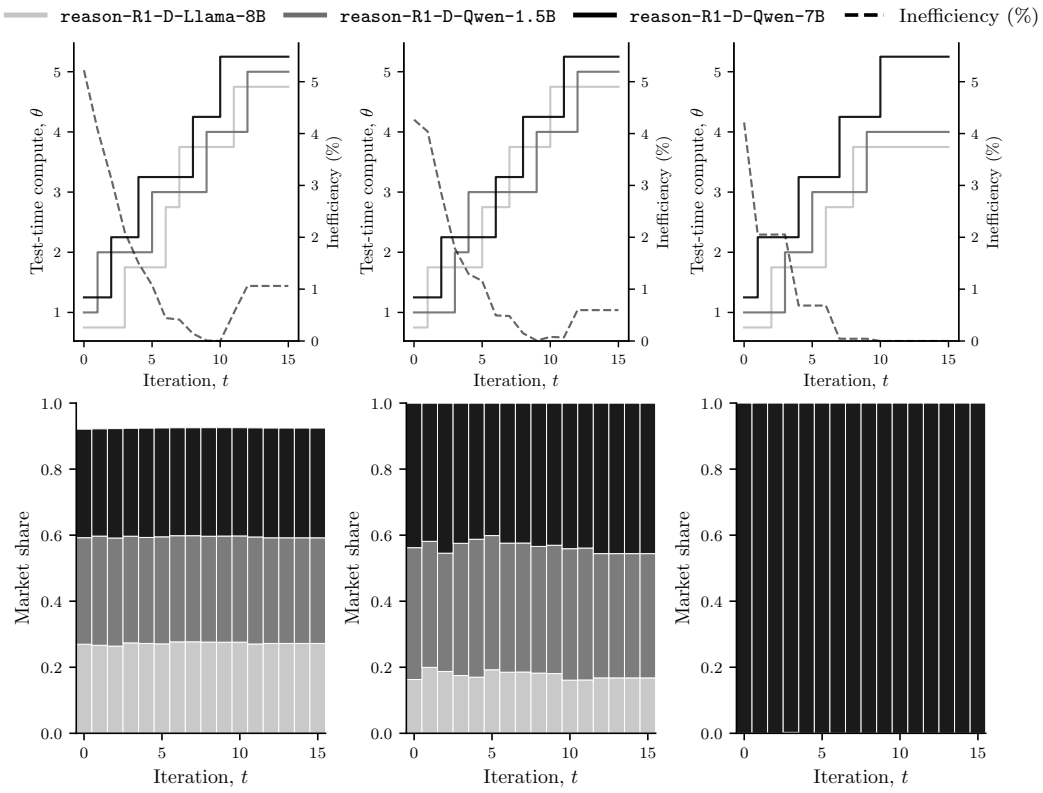
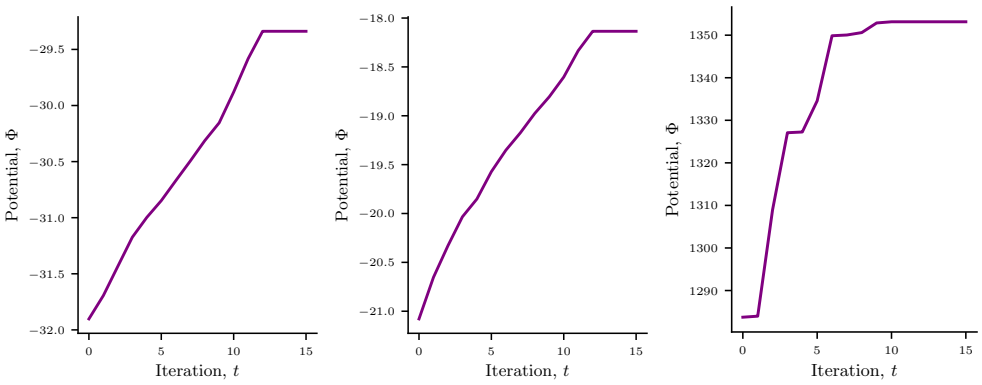


Figure 14: **Inefficiency of a test-time compute game using best-of-n.** The figure shows, as a function of users' rationality β, the inefficiency ($\text{PoA}(\mathcal{G}) - 1$) of the test-time compute game in Figure 12, where $N = 9$ providers sequentially select a test-time compute level that increases their utility.



(a) $\beta = 2 \cdot 10^2$ (b) $\beta = 10^3$ (c) $\beta = 10^5$

Figure 15: **Better-response dynamics of a test-time compute game using CoT.** The upper panels show, for varying levels of user rationality β, better-response dynamics when providers serving models from the `Llama` and `Qwen` families distilled from `DeepSeek-R1` use chain-of-thought to serve queries from the GSM8K dataset. The solid colored lines (left y-axis) represent the test-time compute θ selected by each provider at each iteration, corresponding to the reasoning effort used. The dashed black line (right y-axis) tracks the Market Inefficiency, defined as $(\text{PoA} - 1) \times 100$ (see Eq. 7). The lower panels show the evolution of the market share at each time step of the better-response dynamics. The initial compute level $\boldsymbol{\theta}^1$ is taken as the lowest possible compute. We apply a small vertical jitter to the strategy lines to distinguish overlapping providers and take a fixed profit margin of 25%.



(a) Potential ($\beta = 2 \cdot 10^2$) (b) Potential ($\beta = 10^3$) (c) Potential ($\beta = 10^5$)

Figure 16: **Potential of a test-time compute game using CoT.** The figure shows, for different levels of user rationality β, the evolution of the potential Φ (see Eq. 6) in the test-time compute games in Figure 15 where $N = 3$ providers sequentially select a test-time compute level that increases their utility.

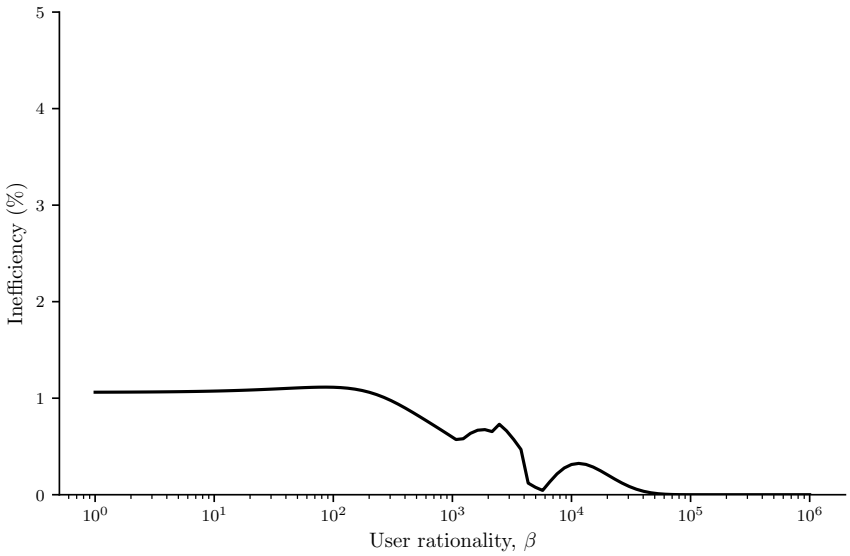
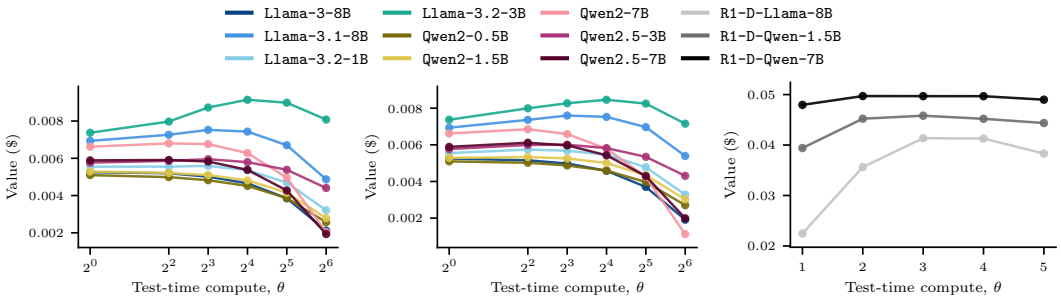


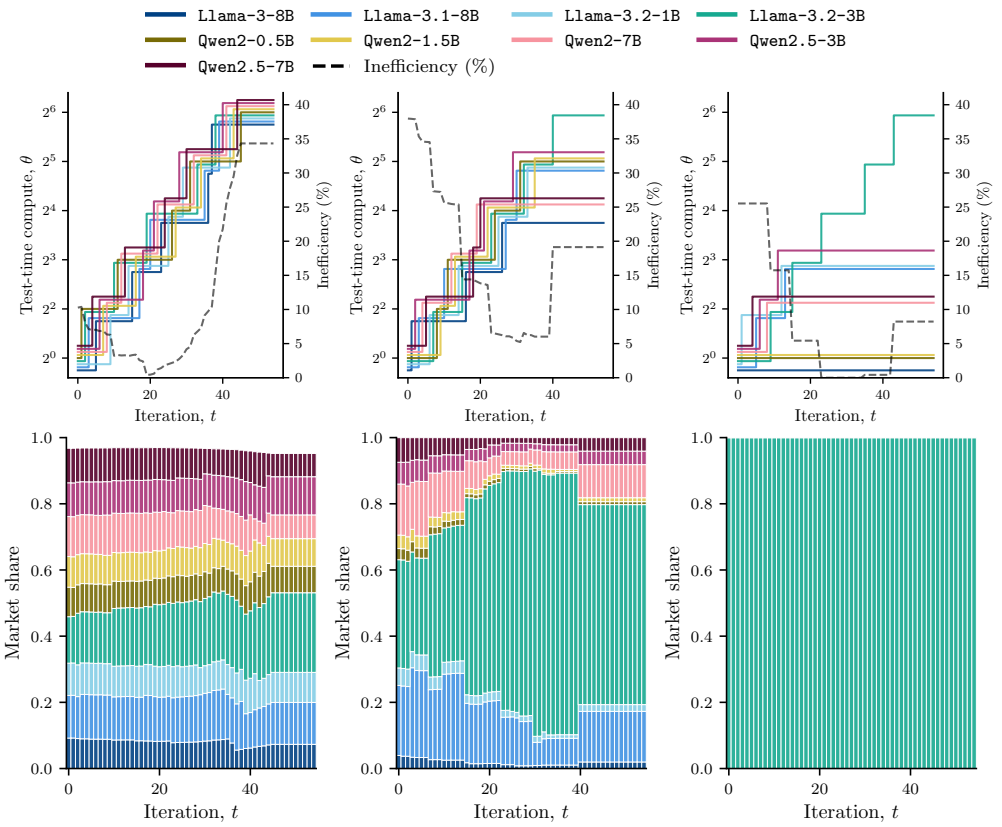
Figure 17: **Inefficiency of a test-time compute game using CoT.** The figure shows, as a function of users' rationality β, the inefficiency ($\text{PoA}(\mathcal{G}) - 1$) of the test-time compute game in Figure 15, where $N = 3$ providers sequentially select a test-time compute level that increases their utility.

E.2.2 TEST-TIME COMPUTE EQUILIBRIA ON AIME



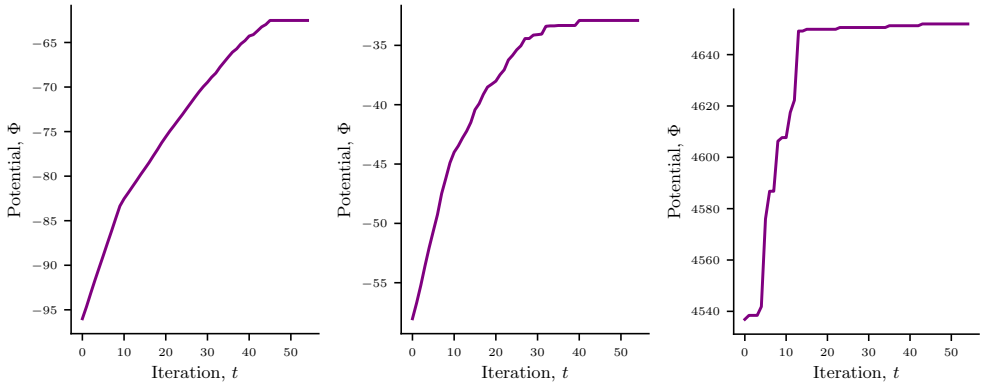
(a) Majority voting (b) Best-of-n (c) Chain-of-thought

Figure 18: **User values offered by providers in a test-time compute game on AIME.** The figure shows the user values $V_i(\theta)$ offered by providers in a test-time compute game $\mathcal{G}$, as a function of their test-time compute θ. Panel (a) and (b) correspond to games with $N = 9$ providers serving non-reasoning models from the `Llama` and `Qwen` families, where providers use, respectively, majority voting and best-of-n across θ samples. Panel (c) corresponds to a game with $N = 3$ providers serving reasoning models distilled from `DeepSeek-R1`, where θ represents reasoning effort, defined by binning the model outputs into quantiles based on the number of reasoning tokens (see Appendix D). In both games, providers serve queries Q from the `AIME` dataset, we set $\beta = 1000$ and consider that each (average) percentage point of accuracy offers a value of $\$0.05$ to the users.



(a) $\beta = 2 \cdot 10^2$ (b) $\beta = 10^3$ (c) $\beta = 10^5$

Figure 19: **Dynamics of a test-time compute game using majority-voting.** The figure shows, for different levels of user rationality β, the better-response dynamics of a test-time compute game $\mathcal{G}$ where $N = 9$ providers sequentially select a test-time compute level that increases their utility. The upper panels show the compute levels θ selected by each provider and the resulting market inefficiency ($\text{PoA}(\mathcal{G}) - 1$), and the lower panels show the market share of each provider. Here, all providers use majority-voting across θ samples as their test-time compute method to serve queries Q from the `AIME` dataset. We consider that providers operate with a margin of 25% between per-token price and per-token cost, and that each (average) percentage point of accuracy offers a value of $\$0.008$ to the users.



(a) Potential ($\beta = 2 \cdot 10^2$) (b) Potential ($\beta = 10^3$) (c) Potential ($\beta = 10^5$)

Figure 20: **Potential of a test-time compute game using majority-voting.** The figure shows, for different levels of user rationality β, the evolution of the potential Φ (see Eq. 6) in the test-time compute games in Figure 19 where $N = 9$ providers sequentially select a test-time compute level that increases their utility.

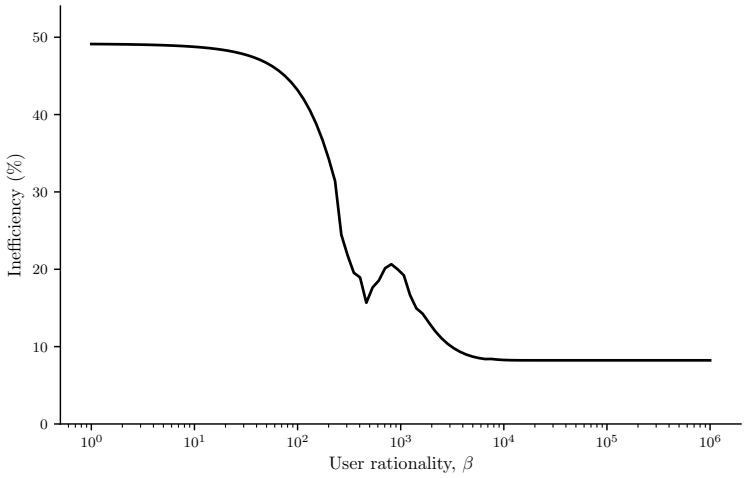
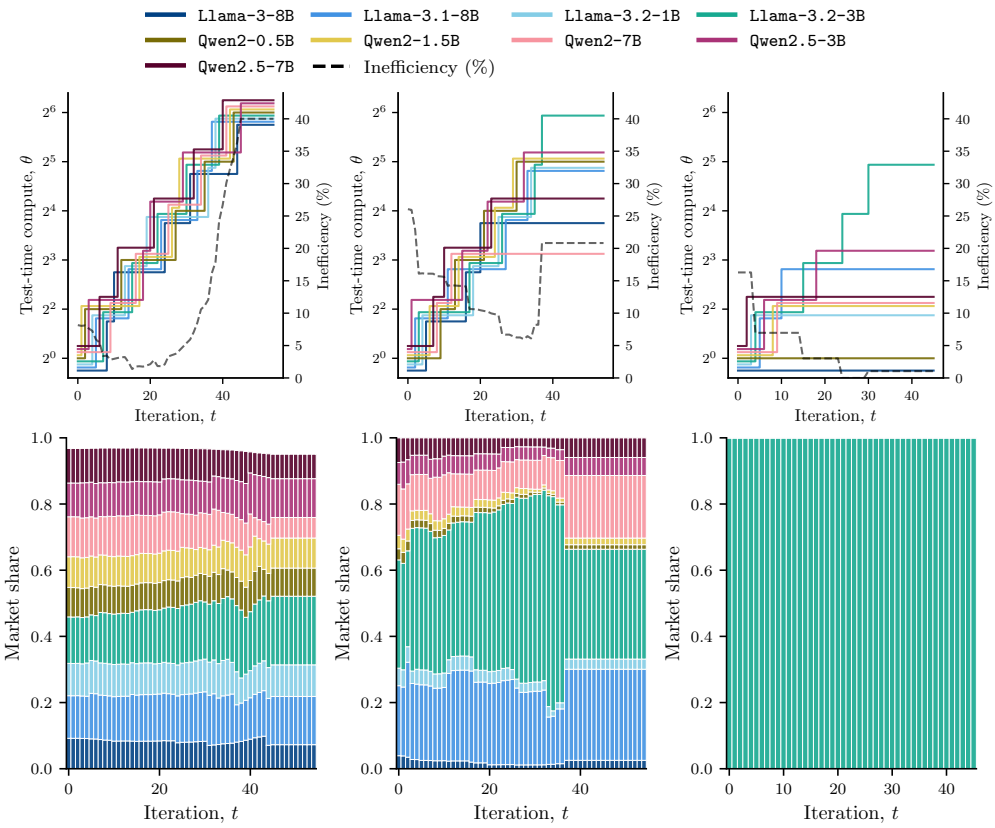
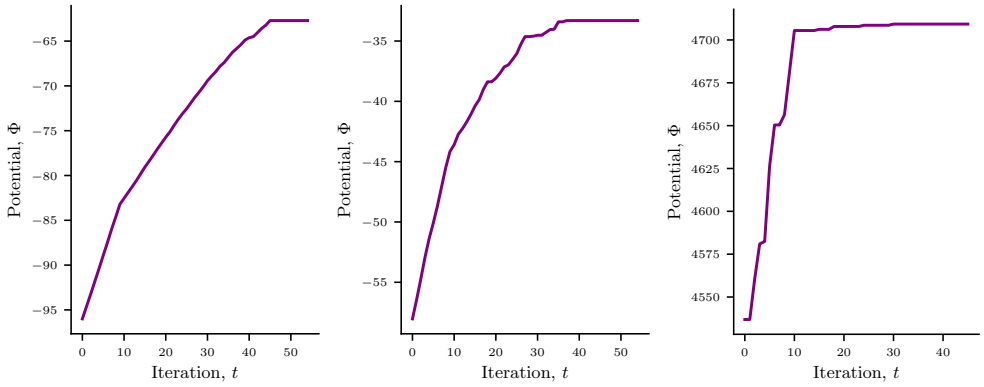


Figure 21: **Inefficiency of a test-time compute game using majority-voting.** The figure shows, as a function of users' rationality β, the inefficiency ($\text{PoA}(\mathcal{G}) - 1$) of the test-time compute game in Figure 19, where $N = 9$ providers sequentially select a test-time compute level that increases their utility.



(a) $\beta = 2 \cdot 10^2$ (b) $\beta = 10^3$ (c) $\beta = 10^5$

Figure 22: **Better-response dynamics of a test-time compute game using best-of-n.** The upper panels show, for varying levels of user rationality β, better-response dynamics when providers serving models from the `Llama` and `Qwen` families use best-of-n to serve queries from the AIME dataset. The solid colored lines (left y-axis) represent the test-time compute θ selected by each provider at each iteration, corresponding to the number of samples used for best-of-n. The dashed black line (right y-axis) tracks the Market Inefficiency, defined as $(\text{PoA} - 1) \times 100$ (see Eq. 7). The lower panels show the evolution of the market share at each time step of the better-response dynamics. The initial compute level $\boldsymbol{\theta}^1$ is taken as the lowest possible compute. We apply a small vertical jitter to the strategy lines to distinguish overlapping providers and take a fixed profit margin of 25%.



(a) Potential ($\beta = 2 \cdot 10^2$) (b) Potential ($\beta = 10^3$) (c) Potential ($\beta = 10^5$)

Figure 23: **Potential of a test-time compute game using best-of-n.** The figure shows, for different levels of user rationality β, the evolution of the potential Φ (see Eq. 6) in the test-time compute games in Figure 22 where $N = 9$ providers sequentially select a test-time compute level that increases their utility.

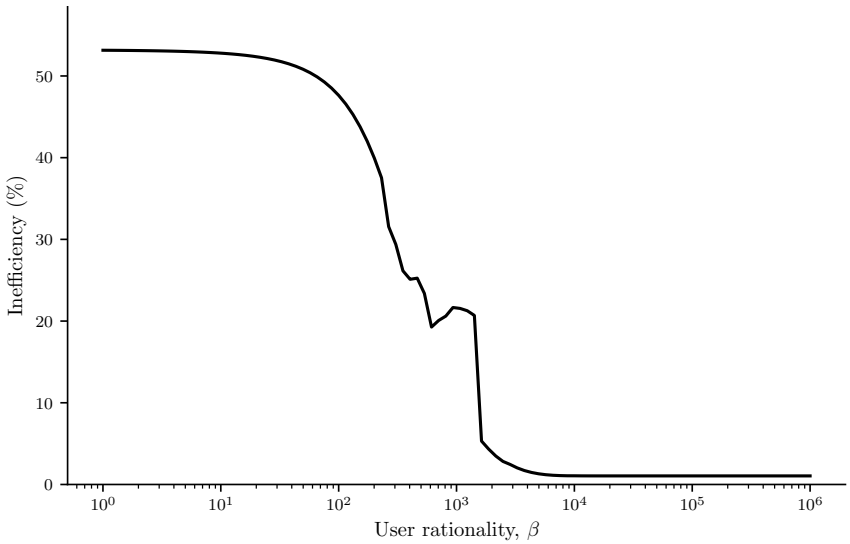
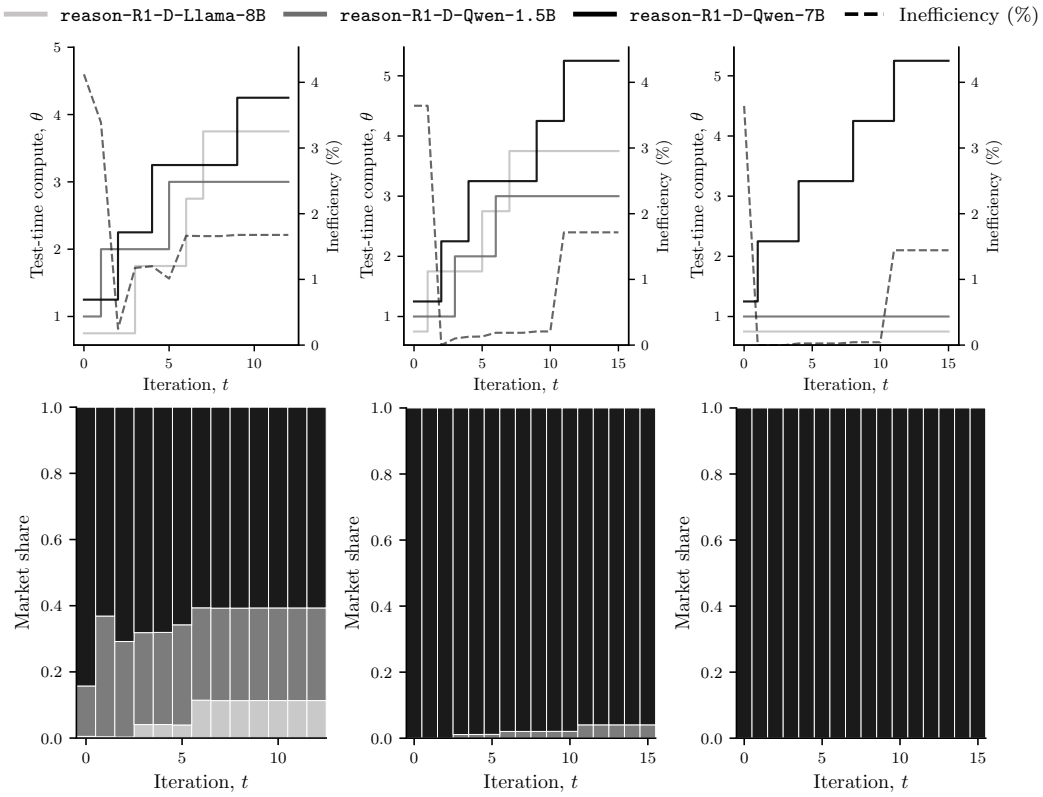
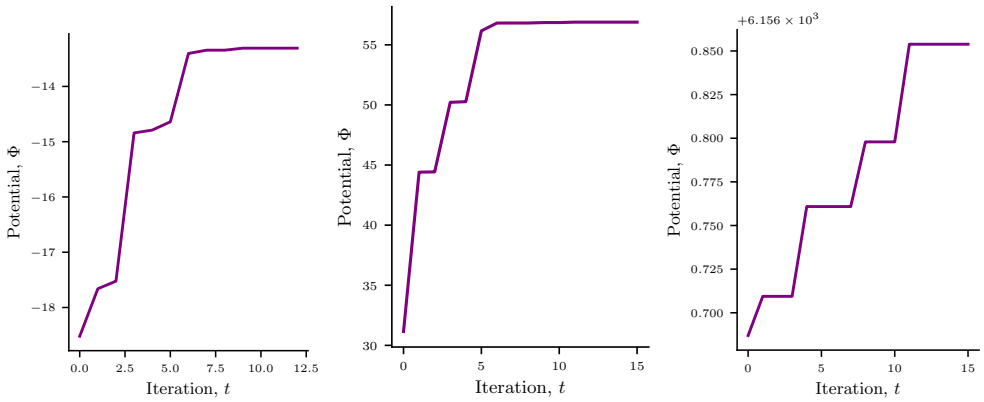


Figure 24: **Inefficiency of a test-time compute game using best-of-n.** The figure shows, as a function of users' rationality β, the inefficiency ($\text{PoA}(\mathcal{G}) - 1$) of the test-time compute game in Figure 22, where $N = 9$ providers sequentially select a test-time compute level that increases their utility.



(a) $\beta = 2 \cdot 10^2$ (b) $\beta = 10^3$ (c) $\beta = 10^5$

Figure 25: **Better-response dynamics of a test-time compute game using CoT.** The upper panels show, for varying levels of user rationality β, better-response dynamics when providers serving models from the `Llama` and `Qwen` families distilled from `DeepSeek-R1` use chain-of-thought to serve queries from the AIME dataset. The solid colored lines (left y-axis) represent the test-time compute θ selected by each provider at each iteration, corresponding to the reasoning effort used. The dashed black line (right y-axis) tracks the Market Inefficiency, defined as $(\text{PoA} - 1) \times 100$ (see Eq. 7). The lower panels show the evolution of the market share at each time step of the better-response dynamics. The initial compute level $\boldsymbol{\theta}^1$ is taken as the lowest possible compute. We apply a small vertical jitter to the strategy lines to distinguish overlapping providers and take a fixed profit margin of 25%.



(a) Potential ($\beta = 2 \cdot 10^2$) (b) Potential ($\beta = 10^3$) (c) Potential ($\beta = 10^5$)

Figure 26: **Potential of a test-time compute game using CoT.** The figure shows, for different levels of user rationality β, the evolution of the potential Φ (see Eq. 6) in the test-time compute games in Figure 25 where $N = 3$ providers sequentially select a test-time compute level that increases their utility.

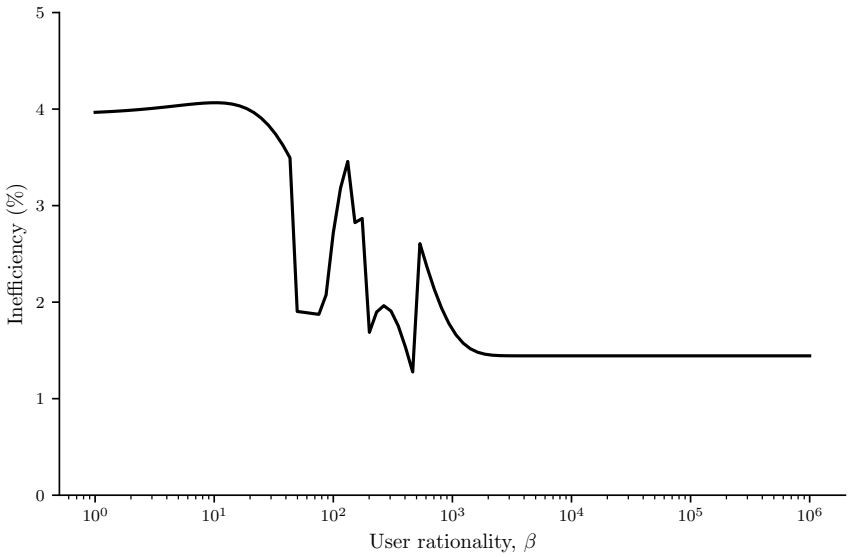
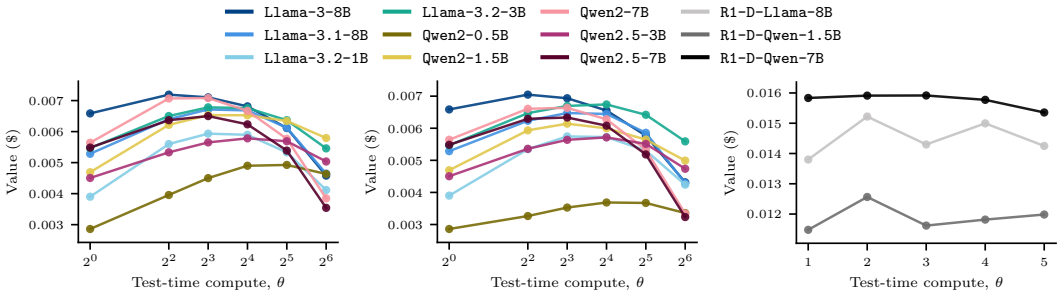


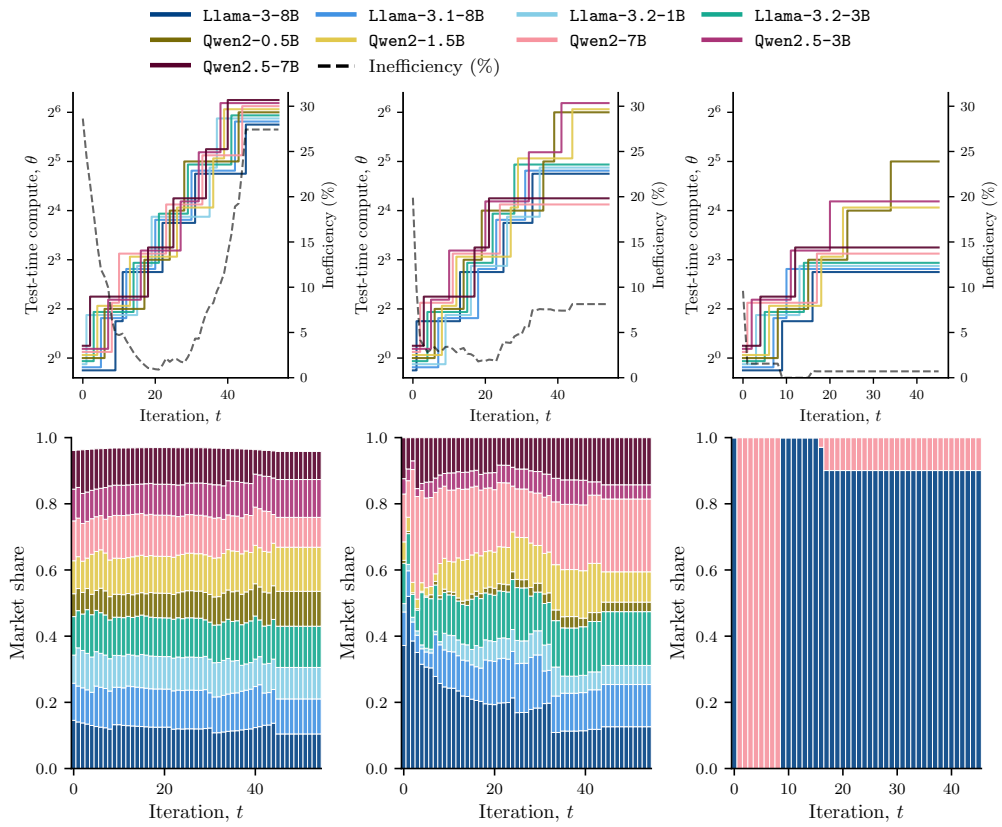
Figure 27: **Inefficiency of a test-time compute game using CoT.** The figure shows, as a function of users' rationality β, the inefficiency ($\text{PoA}(\mathcal{G}) - 1$) of the test-time compute game in Figure 25, where $N = 3$ providers sequentially select a test-time compute level that increases their utility.

E.2.3 TEST-TIME COMPUTE EQUILIBRIA ON GPQA



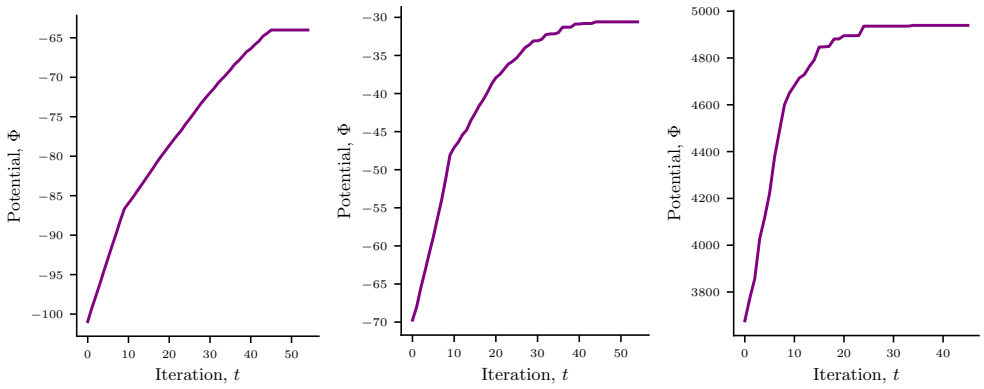
(a) Majority voting (b) Best-of-n (c) Chain-of-thought

Figure 28: **User values offered by providers in a test-time compute game on GPQA.** The figure shows the user values $V_i(\theta)$ offered by providers in a test-time compute game $\mathcal{G}$, as a function of their test-time compute θ. Panel (a) and (b) correspond to games with $N = 9$ providers serving non-reasoning models from the `Llama` and `Qwen` families, where providers use, respectively, majority voting and best-of-n across θ samples. Panel (c) corresponds to a game with $N = 3$ providers serving reasoning models distilled from `DeepSeek-R1`, where θ represents reasoning effort, defined by binning the model outputs into quantiles based on the number of reasoning tokens (see Appendix D). In both games, providers serve queries Q from the `GPQA` dataset, we set $\beta = 1000$ and consider that each (average) percentage point of accuracy offers a value of \$0.02 to the users.



(a) $\beta = 2 \cdot 10^2$ (b) $\beta = 10^3$ (c) $\beta = 10^5$

Figure 29: **Dynamics of a test-time compute game using majority-voting.** The figure shows, for different levels of user rationality β, the better-response dynamics of a test-time compute game $\mathcal{G}$ where $N = 9$ providers sequentially select a test-time compute level that increases their utility. The upper panels show the compute levels θ selected by each provider and the resulting market inefficiency ($\text{PoA}(\mathcal{G}) - 1$), and the lower panels show the market share of each provider. Here, all providers use majority-voting across θ samples as their test-time compute method to serve queries Q from the GPQA dataset. We consider that providers operate with a margin of 25% between per-token price and per-token cost, and that each (average) percentage point of accuracy offers a value of \$0.008 to the users.



(a) Potential ($\beta = 2 \cdot 10^2$) (b) Potential ($\beta = 10^3$) (c) Potential ($\beta = 10^5$)

Figure 30: **Potential of a test-time compute game using majority-voting.** The figure shows, for different levels of user rationality β, the evolution of the potential Φ (see Eq. 6) in the test-time compute games in Figure 29 where $N = 9$ providers sequentially select a test-time compute level that increases their utility.

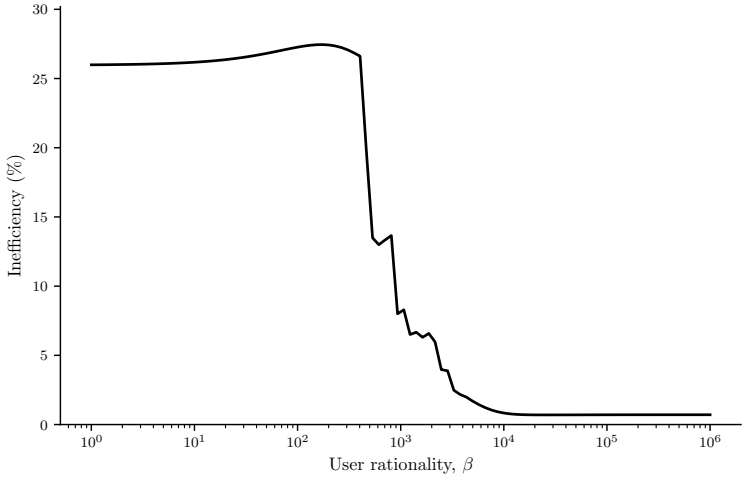
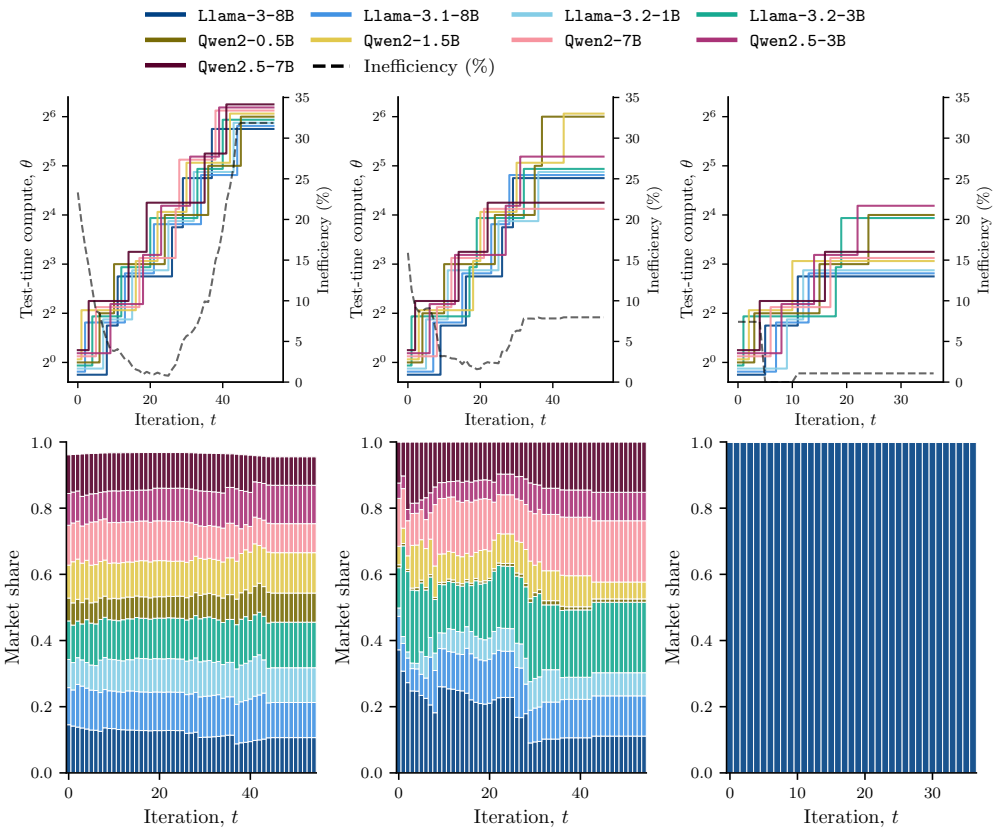
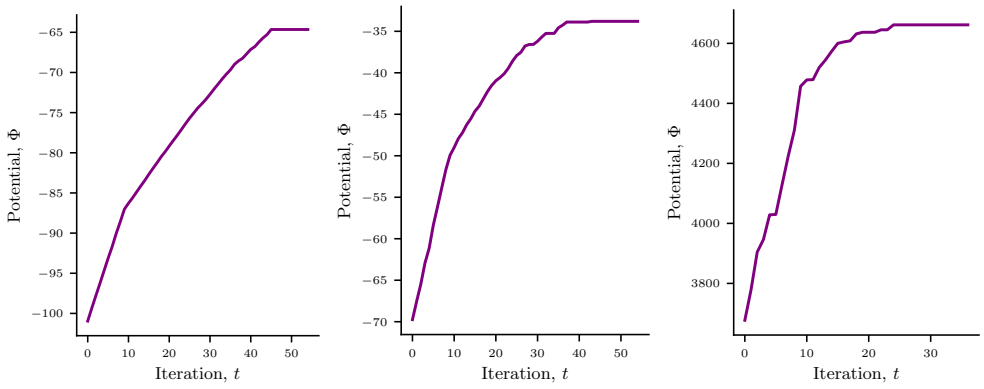


Figure 31: **Inefficiency of a test-time compute game using majority-voting.** The figure shows, as a function of users' rationality β, the inefficiency ($\text{PoA}(\mathcal{G}) - 1$) of the test-time compute game in Figure 29, where $N = 9$ providers sequentially select a test-time compute level that increases their utility.



(a) $\beta = 2 \cdot 10^2$ (b) $\beta = 10^3$ (c) $\beta = 10^5$

Figure 32: **Better-response dynamics of a test-time compute game using best-of-n.** The upper panels show, for varying levels of user rationality β, better-response dynamics when providers serving models from the `Llama` and `Qwen` families use best-of-n to serve queries from the GPQA dataset. The solid colored lines (left y-axis) represent the test-time compute θ selected by each provider at each iteration, corresponding to the number of samples used for best-of-n. The dashed black line (right y-axis) tracks the Market Inefficiency, defined as $(\text{PoA} - 1) \times 100$ (see Eq. 7). The lower panels show the evolution of the market share at each time step of the better-response dynamics. The initial compute level $\boldsymbol{\theta}^1$ is taken as the lowest possible compute. We apply a small vertical jitter to the strategy lines to distinguish overlapping providers and take a fixed profit margin of 25%.



(a) Potential ($\beta = 2 \cdot 10^2$) (b) Potential ($\beta = 10^3$) (c) Potential ($\beta = 10^5$)

Figure 33: **Potential of a test-time compute game using best-of-n.** The figure shows, for different levels of user rationality β, the evolution of the potential Φ (see Eq. 6) in the test-time compute games in Figure 32 where $N = 9$ providers sequentially select a test-time compute level that increases their utility.

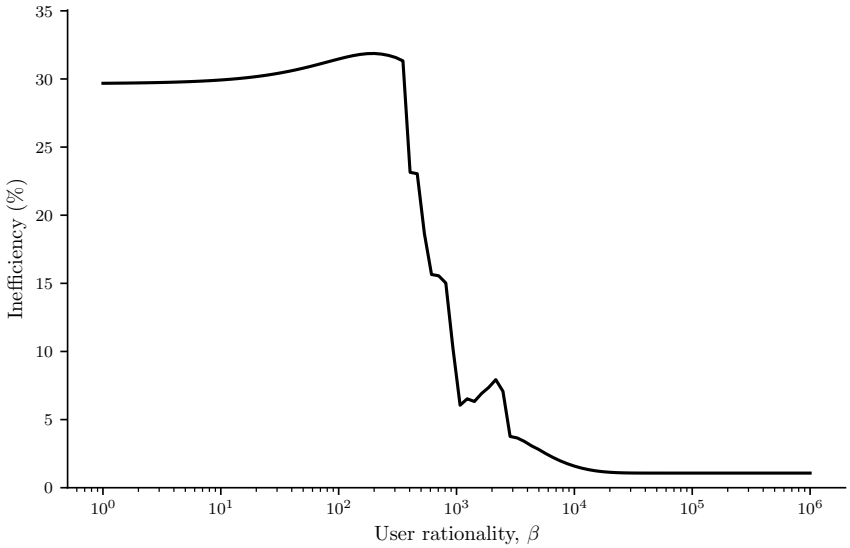
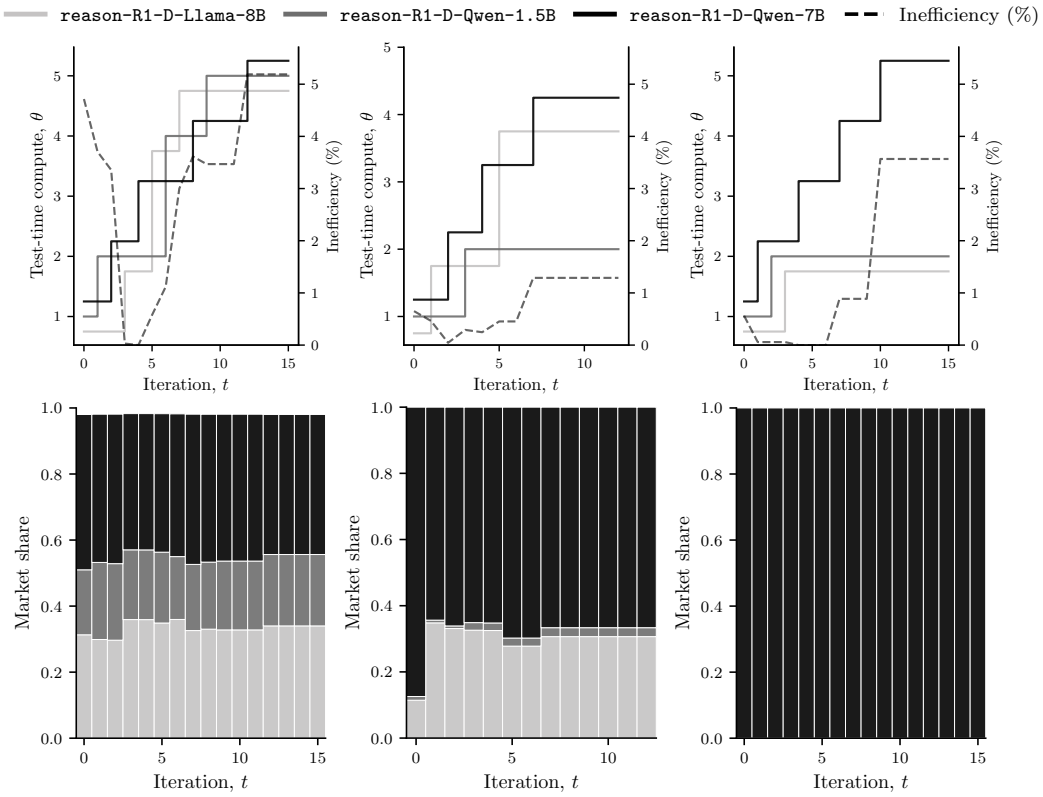
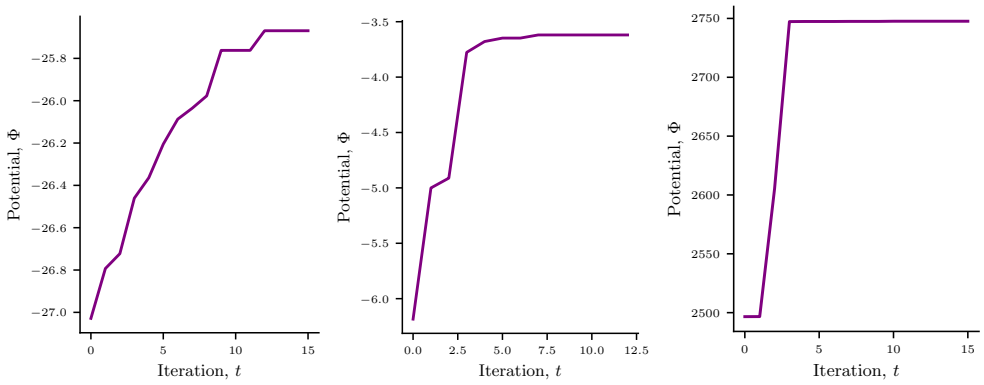


Figure 34: **Inefficiency of a test-time compute game using best-of-n.** The figure shows, as a function of users' rationality β, the inefficiency ($\text{PoA}(\mathcal{G}) - 1$) of the test-time compute game in Figure 32, where $N = 9$ providers sequentially select a test-time compute level that increases their utility.



(a) $\beta = 2 \cdot 10^2$ (b) $\beta = 10^3$ (c) $\beta = 10^5$

Figure 35: **Better-response dynamics of a test-time compute game using CoT.** The upper panels show, for varying levels of user rationality β, better-response dynamics when providers serving models from the `Llama` and `Qwen` families distilled from `DeepSeek-R1` use chain-of-thought to serve queries from the GPQA dataset. The solid colored lines (left y-axis) represent the test-time compute θ selected by each provider at each iteration, corresponding to the reasoning effort used. The dashed black line (right y-axis) tracks the Market Inefficiency, defined as $(\text{PoA} - 1) \times 100$ (see Eq. 7). The lower panels show the evolution of the market share at each time step of the better-response dynamics. The initial compute level $\boldsymbol{\theta}^1$ is taken as the lowest possible compute. We apply a small vertical jitter to the strategy lines to distinguish overlapping providers and take a fixed profit margin of 25%.



(a) Potential ($\beta = 2 \cdot 10^2$) (b) Potential ($\beta = 10^3$) (c) Potential ($\beta = 10^5$)

Figure 36: **Potential of a test-time compute game using CoT.** The figure shows, for different levels of user rationality β, the evolution of the potential Φ (see Eq. 6) in the test-time compute games in Figure 35 where $N = 3$ providers sequentially select a test-time compute level that increases their utility.

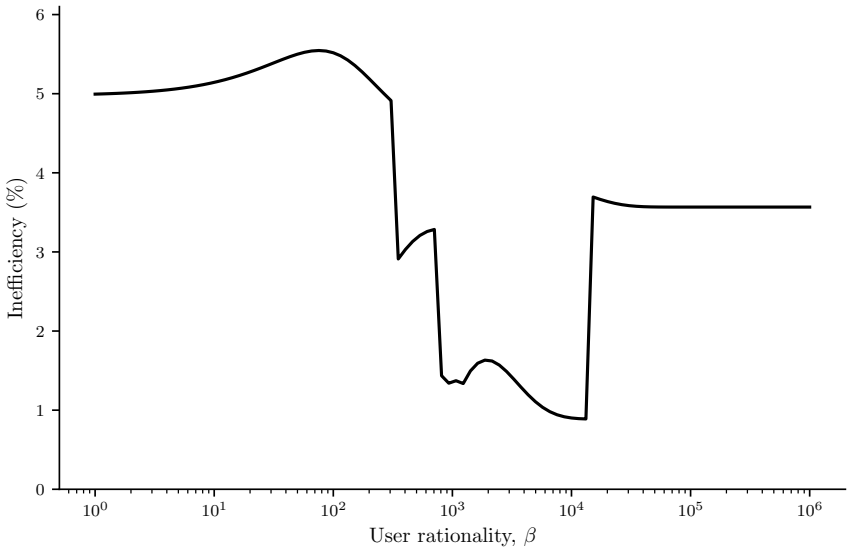


Figure 37: **Inefficiency of a test-time compute game using CoT.** The figure shows, as a function of users' rationality β, the inefficiency ($\text{PoA}(\mathcal{G}) - 1$) of the test-time compute game in Figure 35, where $N = 3$ providers sequentially select a test-time compute level that increases their utility.

E.3 RESULTS FOR THE AUCTION MECHANISM

Here, we compare the equilibrium outcomes of the test-time compute game $\mathcal{G}$ with the outcomes of the auction mechanism $\widetilde{\mathcal{G}}$ introduced in Section 4.

E.3.1 RESULTS FOR THE AUCTION MECHANISM ON GSM8K

Table 2: **Comparison of the equilibrium between $\mathcal{G}$ and the auction $\widetilde{\mathcal{G}}$ on GSM8K using majority voting.** The table shows, for $\widetilde{\mathcal{G}}$, the value received by the user (once the auction is conducted), the price they pay (according to Eq. 8), the provider's utility, the social welfare, and the provider (LLM) that wins the auction. For $\mathcal{G}$, the table shows the provider's average utility, the user's average value and price, all weighted by their equilibrium market shares, together with the social welfare. All quantities have units ($\$ \times 10^{-3}$).

	Game $\mathcal{G}$			Auction $\widetilde{\mathcal{G}}$
	$\beta = 2 \cdot 10^2$	$\beta = 10^3$	$\beta = 10^5$	`Qwen2.5-7B`
User value	4.5	5.8	7.1	7.1
Price	1.6	1.2	0.34	0.30
Provider(s') utility	0.32	0.25	0.07	0.186
Social welfare	4.8	6.0	7.2	7.3

Table 3: **Comparison of the equilibrium between $\mathcal{G}$ and the auction $\widetilde{\mathcal{G}}$ on GSM8K using best-of-n.** The table shows, for $\widetilde{\mathcal{G}}$, the value received by the user (once the auction is conducted), the price they pay (according to Eq. 8), the provider's utility, the social welfare, and the provider (LLM) that wins the auction. For $\mathcal{G}$, the table shows the provider's average utility, the user's average value and price, all weighted by their equilibrium market shares, together with the social welfare. All quantities have units ($\$ \times 10^{-3}$).

	Game $\mathcal{G}$			Auction $\widetilde{\mathcal{G}}$
	$\beta = 2 \cdot 10^2$	$\beta = 10^3$	$\beta = 10^5$	`Llama-3-8B`
User value	4.4	5.5	7.1	7.2
Price	1.6	1.2	0.33	0.32
Provider(s') utility	0.32	0.25	0.07	0.18
Social welfare	4.7	5.8	7.2	7.3

Table 4: **Comparison of the equilibrium between $\mathcal{G}$ and the auction $\widetilde{\mathcal{G}}$ on GSM8K using chain-of-thought.** The table shows, for $\widetilde{\mathcal{G}}$, the value received by the user (once the auction is conducted), the price they pay (according to Eq. 8), the provider's utility, the social welfare, and the provider (LLM) that wins the auction. For $\mathcal{G}$, the table shows the provider's average utility, the user's average value and price, all weighted by their equilibrium market shares, together with the social welfare. All quantities have units ($\$ \times 10^{-3}$).

	Game $\mathcal{G}$			Auction $\widetilde{\mathcal{G}}$
	$\beta = 2 \cdot 10^2$	$\beta = 10^3$	$\beta = 10^5$	`R1-D-Qwen-7B`
User value	6.5	7.1	7.4	7.3
Price	0.16	0.17	0.23	0.30
Provider(s') utility	0.03	0.04	0.05	0.12
Social welfare	6.5	7.1	7.4	7.4

E.3.2 RESULTS FOR THE AUCTION MECHANISM ON AIME

Table 5: **Comparison of the equilibrium between $\mathcal{G}$ and the auction $\widetilde{\mathcal{G}}$ on GSM8K using majority voting.** The table shows, for $\widetilde{\mathcal{G}}$, the value received by the user (once the auction is conducted), the price they pay (according to Eq. 8), the provider's utility, the social welfare, and the provider (LLM) that wins the auction. For $\mathcal{G}$, the table shows the provider's average utility, the user's average value and price, all weighted by their equilibrium market shares, together with the social welfare. All quantities have units ($\$ \times 10^{-3}$).

	Game $\mathcal{G}$			Auction $\widetilde{\mathcal{G}}$
	$\beta = 2 \cdot 10^2$	$\beta = 10^3$	$\beta = 10^5$	`Llama-3.2-3B`
User value	4.2	7.2	8.0	7.6
Price	3.0	2.0	2.4	2.1
Provider(s') utility	0.62	0.42	0.48	1.6
Social welfare	4.8	7.6	8.5	9.2

Table 6: **Comparison of the equilibrium between $\mathcal{G}$ and the auction $\widetilde{\mathcal{G}}$ on GSM8K using best-of-n.** The table shows, for $\widetilde{\mathcal{G}}$, the value received by the user (once the auction is conducted), the price they pay (according to Eq. 8), the provider's utility, the social welfare, and the provider (LLM) that wins the auction. For $\mathcal{G}$, the table shows the provider's average utility, the user's average value and price, all weighted by their equilibrium market shares, together with the social welfare. All quantities have units ($\$ \times 10^{-3}$).

	Game $\mathcal{G}$			Auction $\widetilde{\mathcal{G}}$
	$\beta = 2 \cdot 10^2$	$\beta = 10^3$	$\beta = 10^5$	`Llama-3.2-8B`
User value	3.9	6.5	8.2	7.7
Price	3.0	1.7	1.2	1.3
Provider(s') utility	0.62	0.35	0.24	0.86
Social welfare	4.5	6.9	8.4	8.5

Table 7: **Comparison of the equilibrium between $\mathcal{G}$ and the auction $\widetilde{\mathcal{G}}$ on GSM8K using chain-of-thought.** The table shows, for $\widetilde{\mathcal{G}}$, the value received by the user (once the auction is conducted), the price they pay (according to Eq. 8), the provider's utility, the social welfare, and the provider (LLM) that wins the auction. For $\mathcal{G}$, the table shows the provider's average utility, the user's average value and price, all weighted by their equilibrium market shares, together with the social welfare. All quantities have units ($\$ \times 10^{-3}$).

	Game $\mathcal{G}$			Auction $\widetilde{\mathcal{G}}$
	$\beta = 2 \cdot 10^2$	$\beta = 10^3$	$\beta = 10^5$	`R1-D-Qwen-7B`
User value	47	48	48	45
Price	0.26	0.33	0.34	4.1
Provider(s') utility	0.05	0.07	0.07	3.9
Social welfare	47	48	49	49

E.3.3 RESULTS FOR THE AUCTION MECHANISM ON GPQA

Table 8: **Comparison of the equilibrium between $\mathcal{G}$ and the auction $\widetilde{\mathcal{G}}$ on GPQA using majority voting.** The table shows, for $\widetilde{\mathcal{G}}$, the value received by the user (once the auction is conducted), the price they pay (according to Eq. 8), the provider's utility, the social welfare, and the provider (LLM) that wins the auction. For $\mathcal{G}$, the table shows the provider's average utility, the user's average value and price, all weighted by their equilibrium market shares, together with the social welfare. All quantities have units ($\$ \times 10^{-3}$).

	Game $\mathcal{G}$			Auction $\widetilde{\mathcal{G}}$
	$\beta = 2 \cdot 10^2$	$\beta = 10^3$	$\beta = 10^5$	`Llama-3-8B`
User value	4.5	6.1	7.1	7.1
Price	2.3	1.2	0.38	0.2
Provider(s') utility	0.47	0.24	0.08	0.05
Social welfare	4.9	6.3	7.1	7.2

Table 9: **Comparison of the equilibrium between $\mathcal{G}$ and the auction $\widetilde{\mathcal{G}}$ on GPQA using best-of-n.** The table shows, for $\widetilde{\mathcal{G}}$, the value received by the user (once the auction is conducted), the price they pay (according to Eq. 8), the provider's utility, the social welfare, and the provider (LLM) that wins the auction. For $\mathcal{G}$, the table shows the provider's average utility, the user's average value and price, all weighted by their equilibrium market shares, together with the social welfare. All quantities have units ($\$ \times 10^{-3}$).

	Game $\mathcal{G}$			Auction $\widetilde{\mathcal{G}}$
	$\beta = 2 \cdot 10^2$	$\beta = 10^3$	$\beta = 10^5$	`Llama-3-8B`
User value	4.1	5.9	6.9	6.8
Price	2.3	1.1	0.37	0.39
Provider(s') utility	0.47	0.23	0.07	0.2
Social welfare	4.6	6.1	7.0	7.0

Table 10: **Comparison of the equilibrium between $\mathcal{G}$ and the auction $\widetilde{\mathcal{G}}$ on GPQA using chain-of-thought.** The table shows, for $\widetilde{\mathcal{G}}$, the value received by the user (once the auction is conducted), the price they pay (according to Eq. 8), the provider's utility, the social welfare, and the provider (LLM) that wins the auction. For $\mathcal{G}$, the table shows the provider's average utility, the user's average value and price, all weighted by their equilibrium market shares, together with the social welfare. All quantities have units ($\$ \times 10^{-3}$).

	Game $\mathcal{G}$			Auction $\widetilde{\mathcal{G}}$
	$\beta = 2 \cdot 10^2$	$\beta = 10^3$	$\beta = 10^5$	`R1-D-Qwen-7B`
User value	13	15	15	15
Price	0.24	0.24	0.31	0.91
Provider(s') utility	0.05	0.05	0.06	0.72
Social welfare	14	15	15	16