THE COST OF KNOWING: A RESOURCE-AWARE PROTOCOL FOR BENCHMARKING HALLUCINATION BEYOND STATIC LEADERBOARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

On standard factuality tasks, frontier models now cluster near the top of the scale. The question is therefore shifting from how factual a system is toward how much compute that factuality costs. Static leaderboards score factuality in isolation and treat compute as free, so they cannot tell a genuinely better system apart from one that simply spends more. Consider a ranking reversal. A brute-force Best-of-4 agent posts the higher raw factuality score (H-Score 0.9169 vs. 0.9103) and would top a static leaderboard, but once cost is counted it is the worse system, losing on Q-Score (0.5169 vs. 0.5217) at roughly four times the tokens and latency, under a reported cost weight whose sensitivity we sweep. So the system that tops a static leaderboard can be the worse one to deploy. To make this trade-off visible, we introduce MAS-HQ (**M**ulti-**A**gent **S**ystem **H**allucination **Q**uest), a resource-aware evaluation protocol. It wraps any factuality detector and normalizes for cost, and it pits systems against each other rather than scoring them in isolation. The Q-Score measures factuality minus normalized cost under a competitive match. Across summarization and open-domain QA, single-agent baselines drift into resource-heavy over-optimization, while competition elicits more resource-efficient policies. These gains are small but consistent, and stable across 100 trials. The axis stays discriminative for frontier systems (Gemini-2.5-Pro, and GPT-5 in simulated preview) whose raw factuality scores are already bunched near the ceiling. MAS-HQ provides a reproducible way to measure how much a factual answer costs.

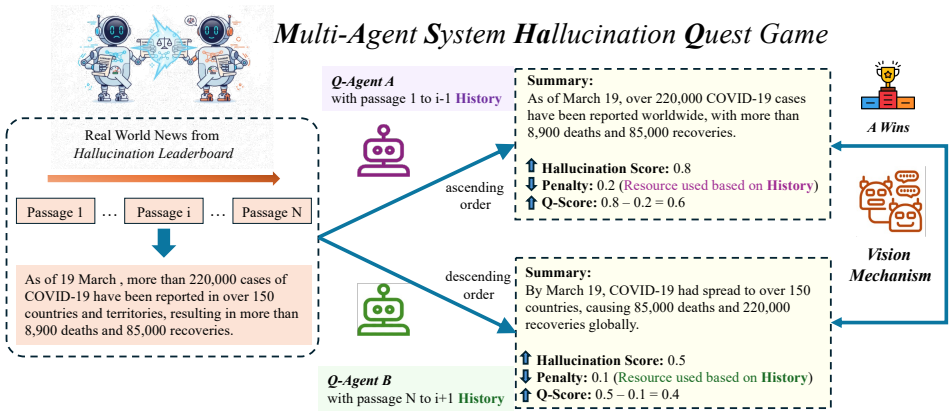

Figure 1: **Same tasks, reversed ranking: higher factuality, worse once cost is counted.** Two Q-Agents compete on MAS-HQ; the original passage list is input in different sequences to Q-Agent A and Q-Agent B, which devise strategies to summarize each passage using mechanisms such as the vision mechanism. Hallucination scores and resource-usage penalties are recorded and combined into the Q-Score, and the agent with the higher Q-Score wins—making the cost-of-knowing trade-off visible rather than hidden. In the current context, Q-Agent A wins.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

On standard factuality tasks, frontier LLMs now cluster near the top of the scale (Huang et al., 2025a; Alzahrani et al., 2024; Huang et al., 2025b; Singh et al., 2025). As they do, the informative question moves from *how factual* a system is to *how much compute that factuality costs*. Static leaderboards still score factuality in isolation and treat compute as free. Under that convention a model can "buy" a higher score through computationally extravagant mitigation, such as extensive Best-of-N sampling or exhaustive reasoning chains, without penalty. So a static leaderboard cannot tell a genuinely better system from one that merely spends more.

Consider a ranking reversal. A brute-force Best-of-4 agent posts the higher raw factuality score (H-Score 0.9169 vs. 0.9103) and would top a static leaderboard. Once cost is counted it is the worse system: it loses on Q-Score (0.5169 vs. 0.5217) at roughly four times the tokens and latency, scored under a reported cost weight whose sensitivity we sweep, so the static-score winner can be the wrong deployment pick. The trade-off is not a small-model artifact, since it persists for frontier systems (Gemini-2.5-Pro, and GPT-5 in simulated preview) whose raw factuality already sits near the ceiling. It is compounded in Multi-Agent Systems (MAS), where interactions can amplify and propagate hallucinations, outputs that appear plausible yet are factually erroneous (Macpherson & Platchias, 2013; Huang et al., 2021; Ji et al., 2023; Li et al., 2022; Schmidgall & Moor, 2025). In such settings latency and compute are first-order. In high-frequency trading or real-time news intelligence, an agent that delivers equally credible insights on half the budget delivers greater operational value.

To make this trade-off visible, we introduce MAS-HQ (**M**ulti-**A**gent **S**ystem **H**allucination **Q**uest), a resource-aware evaluation protocol built on standard information-synthesis tasks (vectara, 2024; Bao et al., 2024) as objective, quantifiable proxies. The accuracy term is *metric-agnostic*: any factuality detector works, and its reliability bounds the scores. The Q-Score then subtracts normalized cost under a competitive match, so an accuracy gain bought with disproportionate compute is scored down. Each measurement pairs with fixed budgets and a competitive ranking rule, which surfaces the ranking changes static leaderboards hide. We instantiate the protocol with the Q-Agent, whose optional parameterized telemetry knob surfaces partial state about an opponent only when a costly action such as self-review is taken, mirroring the partial observability of latency, retries, and tool calls in deployment. Figure 1 gives an overview.

Experiments show that competition produces small but consistent shifts. Single-agent baselines fall into the *over-optimization trap*, spending disproportionate resources for marginal accuracy; under a shared budget, agents adopt more resource-efficient policies. The gaps stay stable across 100 trials, and they change relative rankings that a static score cannot surface.

In summary, our contributions are: (1) **The MAS-HQ protocol:** a resource-aware evaluation protocol that wraps any factuality metric and scores accuracy against normalized operational cost, making the cost-of-knowing trade-off visible. (2) **A modular architecture for controlled ablations:** including an optional parameterized telemetry knob (the "vision mechanism"), which isolates when competition reverses rankings versus when it collapses onto the static axis. (3) **Reproducible analysis:** experiments showing shared budgets and competition elicit resource-efficient behaviors, consistent across trials, and a ranking reversal that static leaderboards overlook.

2 RELATED WORK

LLM Hallucination and Resource-Agnostic Evaluation Large Language Models (LLM) hallucination—the generation of fluent yet factually erroneous content arising from flawed internals, biased data, or misaligned external knowledge (Macpherson & Platchias, 2013; Huang et al., 2025a; 2021; Ji et al., 2023; Li et al., 2022; Bender et al., 2021; Li et al., 2023; Holtzman et al., 2019)—remains a critical challenge. Various mitigation strategies, ranging from retrieval-augmentation to complex reasoning chains, have subsequently emerged (Abbas et al., 2021; Dai et al., 2023; Gao et al., 2022; Shi et al., 2023; Chuang et al., 2023). However, existing benchmarks (Lin et al., 2021; Li et al., 2024; vectara, 2024; Bao et al., 2024; Cheng et al., 2023) predominantly focus on static, resource-agnostic hallucination reduction, inadvertently rewarding computationally extravagant approaches (e.g., exhaustive Best-of-N sampling) while ignoring operational costs like latency and API calls. MAS-HQ addresses this gap by explicitly evaluating efficiency alongside veracity, penalizing the brute-force accumulation of factuality.

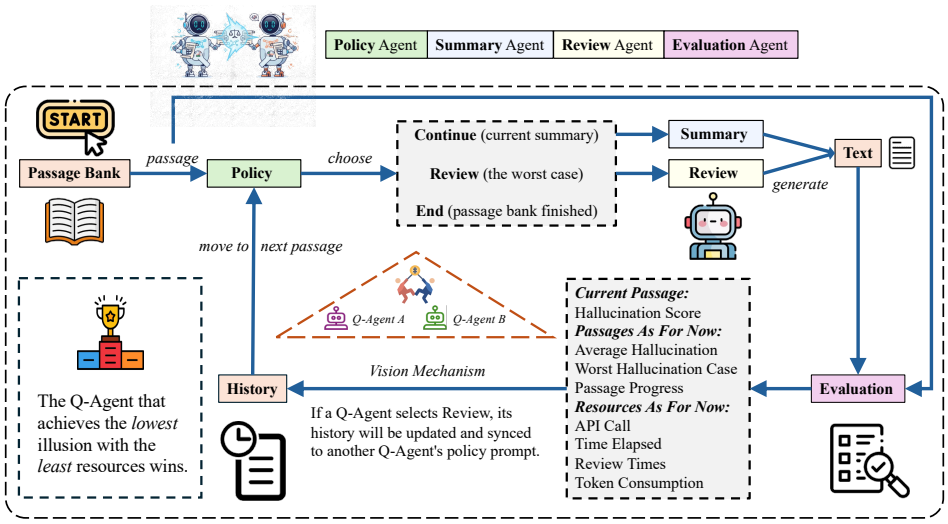

Figure 2: **Overview of the composition of the Q-Agent.** Policy Agent receives both the Q-Agent's own historical information and the opponent's historical information (via the vision mechanism) to determine the next strategy. The text summary generated by either the Summary Agent or the Review Agent is evaluated by the Evaluation Agent, which computes the hallucination score (H-Score) and resource usage. After all steps are completed, the overall Q-Score is calculated, which includes the average hallucination score and resource consumption penalties.

Multi-Agent Systems and Economic Constraints As Multi-Agent Systems (MAS) deploy autonomous agents capable of complex interaction across increasingly diverse domains (Liu et al., 2025; Guo et al., 2024; Rasheed et al., 2024; Wang et al., 2024; Huang et al., 2023; Hong et al., 2023; Chen et al., 2023; Yu et al., 2023; Ke et al., 2024; Ni & Buehler, 2024; Xie et al., 2024; Jiang et al., 2024; Wu et al., 2023; Fan et al., 2024; Zhang et al., 2024a;b), evaluating their decision quality under stringent operational budgets in time-sensitive environments becomes paramount. While current benchmarks (Zhu et al., 2025) valuably assess open-ended collaboration, they often neglect the rigorous quantification of information fidelity against strict resource limitations. MAS-HQ addresses this by introducing a game-theoretic paradigm that utilizes standard generation tasks as quantifiable proxies, forcing agents to strategically balance task success against resource consumption and adversarial pressure.

3 MAS-HQ Framework and Reference Architecture

Static factuality leaderboards presuppose effectively infinite compute. They score how factual a system is while implicitly assuming the compute that buys that factuality is free, so they cannot separate a genuinely better system from one that merely spends more. We introduce MAS-HQ (**M**ulti-**A**gent **S**ystem **H**allucination **Q**uest), a controlled resource-aware protocol that makes rankings cost-honest and makes such reversals visible. MAS-HQ treats standard summarization and QA tasks as quantifiable proxies for information synthesis and scores factuality jointly against normalized operational cost, so speed, cost, and accuracy are all weighed rather than assumed free. Any factuality detector can serve as the accuracy term, so the protocol stays metric-agnostic. Figure 2 shows the MAS-HQ evaluation and the Q-Agent reference implementation.

3.1 MAS-HQ Mechanics and the Q-Score Objective

The Q-Score just adds a cost term to factuality: factuality minus normalized cost under a competitive match, not a replacement for the factuality metric. Built on established datasets (vectara, 2024; Bao et al., 2024), MAS-HQ pairs agents that sequentially process information streams while trading

Table 1: **The reversal recurs across LLMs: the highest-factuality agent is often not the winner.** Within each competition group, Q-Agents share the same underlying LLM. Once cost is scored, the winner (Q-Score) is repeatedly not the agent with the higher raw factuality (H-Score)—resource efficiency (API Calls, Tokens, Review, Time) decides the cost-honest ranking.

Q-Agent Competition	Metrics						
	H-Score ↑	API Call ↓	Tokens ↓	Review ↓	Time ↓	Q-Score ↑	Winner
A: GPT-4o-mini	0.9103	**2417**	**1.36M**	**791**	**8.83k**	**0.5217**	✓
B: GPT-4o-mini	**0.9132**	2438	1.44M	812	8.98k	0.5132	
A: Qwen-Max	**0.9030**	2304	**1.48M**	682	**13.77k**	**0.5101**	✓
B: Qwen-Max	0.8994	**2264**	1.52M	**642**	14.45k	0.5070	
A: Deepseek-V3	0.8860	2292	1.42M	666	12.26k	0.4860	
B: Deepseek-V3	**0.8894**	**2233**	**1.38M**	**607**	**12.06k**	**0.5051**	✓
A: Gemini-2.0-Flash	**0.9026**	2262	1.56M	642	20.29k	0.5026	
B: Gemini-2.0-Flash	0.9016	**2157**	**1.53M**	**537**	**19.82k**	**0.5273**	✓
A: Grok-3-beta	**0.9070**	2376	1.37M	750	10.71k	0.5070	
B: Grok-3-beta	0.9049	**2253**	**1.34M**	**627**	**10.15k**	**0.5337**	✓

factual consistency against operational cost, aggregated into a match-specific utility, the $Q\text{-}Score$:

$$Q\text{-}Score = \frac{1}{N}\sum_{i=1}^{N}(\alpha \cdot H\text{-}Score_i - \beta \cdot P_i)$$

For each of N tasks, $H\text{-}Score_i \in [0,1]$ is the factuality metric from the plugged-in detector, and P_i is a normalized operational-cost penalty computed at evaluation time. This evaluation-time cost differs from the policy-side review cost an agent incurs internally when it refines. The Q-Score subtracts only the former, so cost-awareness is scored uniformly across agents and is not baked into the accuracy metric. The penalty P_i aggregates token count (input plus output), API calls, review iterations, and wall-clock runtime, each max-normalized over the agents in the same match, which makes P_i a relative rather than absolute cost. The parameters $\alpha, \beta > 0$ are deployment-SLA knobs encoding a particular accuracy-versus-cost operating point, so we report Q-Score sensitivity across a range of β. Because the penalty is relative to the co-present competitor, the $Q\text{-}Score$ is a match-specific payoff and not a standalone model score. Recovering an intrinsic per-agent ranking would require tournament or Elo-style aggregation over many matches, which we leave to future work.

This objective also makes the ranking boundary explicit. Let $\bar{H}^j = N^{-1}\sum_i H_i^j$ and $\bar{P}^j = N^{-1}\sum_i P_i^j$ be agent j's match-average factuality and cost. If B is more factual but more expensive, define $\Delta_H = \bar{H}^B - \bar{H}^A > 0$ and $\Delta_P = \bar{P}^B - \bar{P}^A > 0$. Then

$$Q^A - Q^B = -\alpha\Delta_H + \beta\Delta_P, \qquad Q^A > Q^B \iff \frac{\beta}{\alpha} > \frac{\Delta_H}{\Delta_P}. \tag{1}$$

Thus the lower-cost agent reverses the raw-factuality ranking exactly when the deployment's cost-to-quality weight exceeds the observed marginal accuracy-per-cost gain; the β sweep estimates this breakpoint.

The same score is the Lagrangian relaxation of factuality maximization under a resource budget $\mathcal{B}$:

$$\max_{\pi^j} \mathbb{E}_{\pi^j}[\bar{H}^j] \quad \text{s.t.} \quad \mathbb{E}_{\pi^j}[\bar{P}^j] \le \mathcal{B}, \qquad \mathcal{L} = \mathbb{E}[\bar{H}^j] - \lambda\big(\mathbb{E}[\bar{P}^j] - \mathcal{B}\big). \tag{2}$$

After dropping the constant and applying a positive scaling, this recovers Q-Score with $\lambda = \beta/\alpha$, the deployment-specific shadow price of resources. Locally, a review is worthwhile only if $\alpha\mathbb{E}[\Delta H_i \mid h_i^j] > \beta\mathbb{E}[\Delta P_i \mid h_i^j]$; telemetry changes the expected gains, not the decision rule.

The Q-Agent Reference Architecture. As a transparent, reproducible baseline, we provide the Q-Agent, a modular reference implementation (analogous to ReAct (Yao et al., 2022)) of four components. The **Policy Agent (PA)** decides the next action (`continue`, `review`, or `end`) from

the current state and any available competitor telemetry. The **Summarization Agent (SA)** executes the primary task. To avoid the degenerate "say less to avoid being wrong" shortcut, we deliberately isolate it from the Q-Score, which confines cost-awareness to the PA. The **Review Agent (RA)** refines summaries at an additional review cost, while the **Evaluation Agent (EA)** scores quality (Bao et al., 2024) and tracks consumption to inform the PA.

Table 2: **The ranking reversal.** Best-of-4 wins on raw factuality (H-Score 0.9169) but loses on Q-Score (0.5169 vs. 0.5217) at $\sim 4\times$ the cost: the static-score winner is here the worse deployment pick. This is the over-optimization trap—MAS-HQ agents adopt more resource-efficient empirical policies, capturing most of the factuality at a fraction of the cost.

Setup (Model: GPT-4o-mini)	H-Score ↑	API Call ↓	Tokens ↓	Review ↓	Time (s) ↓	Q-Score ↑
Single Agent: Best-of-1	0.9144	2477	1.62M	851	9.12k	0.5144
Single Agent: Best-of-2	0.9167	5201	3.40M	1787	19.15k	0.5167
Single Agent: Best-of-4	**0.9169**	9660	6.32M	3310	35.57k	0.5169
Multi-Agent: Q-Agent A (Ours)	0.9103	**2417**	**1.36M**	**791**	**8.83k**	**0.5217**
Multi-Agent: Q-Agent B (Ours)	0.9132	2438	1.44M	812	8.98k	0.5132

3.2 Competitive Dynamics and Optional Telemetry

We instantiate MAS-HQ as a head-to-head match to remove the infinite-compute assumption, with Agent A and Agent B processing tasks in opposite directions to prevent trivial convergence.

The Vision Mechanism. MAS-HQ includes an optional, parameterized telemetry knob controlling how much of a competitor's activity is observable, mirroring the partial telemetry (latency, retries, tool-calls) available in real deployments. When it is enabled and an agent executes a costly action such as `review`, a snapshot of its state ($V_{\mathrm{A},i}^{\mathrm{opponent}}$) is disclosed to its rival, so the policy is conditioned on observed competitor cost:

$$\mathrm{Choice}_{\mathrm{B},i} = \mathrm{Policy}_{\mathrm{B}}(T_{\mathrm{B},i}, V_{\mathrm{B},i}^{\mathrm{self}}, V_{\mathrm{A},i}^{\mathrm{opponent}}), \quad \text{for} \quad i = 1, \ldots, N$$

Turning this knob lets us measure how opponent-conditioned review behavior changes the accuracy–cost trade-off; we examine the full-versus-collapsed match ablation in appendix. A Limited Review Cycle budget (R) further constrains how much refinement an agent may spend. The information structure of a match (private state, conditional observation, sequential actions, and terminal utility) can be described compactly as a POSG. We use this only as descriptive vocabulary and do not solve for equilibria, which remains future work.

Table 3: **The cost-of-knowing trade-off persists where raw factuality saturates.** For frontier systems (GPT-5, simulated preview; Gemini-2.5-Pro) whose H-Scores already cluster near the top of the scale, the axis stays discriminative: higher pure factuality (H-Score) does not guarantee a win once the resource penalty is scored, so the reversal is not a small-model artifact.

SOTA Competition ($\alpha = 1, \beta = 0.01$)	H-Score ↑	Review ↓	Resource Penalty ↓	Q-Score ↑
Q-Agent A: GPT-5	0.9312	867	0.4000	0.5312
Q-Agent B: GPT-5	**0.9278**	**834**	**0.3897**	**0.5381** (✓)
Q-Agent A: Gemini-2.5-Pro	**0.9264**	859	0.4000	0.5264
Q-Agent B: Gemini-2.5-Pro	0.9221	**824**	**0.3911**	**0.5310** (✓)
Q-Agent A: GPT-5	**0.9315**	869	0.4000	**0.5315** (✓)
Q-Agent B: GPT-4o-mini	0.9078	**846**	**0.3877**	0.5201

4 Experiments

Our experiments ask what a static factuality score cannot resolve. First, does jointly scoring factuality against normalized operational cost expose accuracy–cost trade-offs, up to a reversal of the ranking,

Table 4: β **toggles between static and cost-honest ranking.** Fixing $\alpha = 1$ and sweeping β. At $\beta = 0$ the protocol collapses onto the saturating raw-accuracy axis (cost unscored, infinite resources), while $\beta = 0.01$ is an intermediate deployment-SLA setting that balances quality against cost; we report sensitivity rather than claim a canonical value.

α	β	**Model Setup**	H-Score ↑	Review Count	Resource Penalty	Q-Score
1	0	Q-Agent A: GPT-4o-mini	0.9166	854	0.3899	0.5267
1	0	Q-Agent B: GPT-4o-mini	**0.9189**	889	0.4000	**0.5189**
1	**0.01**	**Q-Agent A: GPT-4o-mini (Main)**	0.9103	**791**	**0.3886**	**0.5217**
1	**0.01**	**Q-Agent B: GPT-4o-mini (Main)**	**0.9132**	812	0.4000	0.5132
1	0.02	Q-Agent A: GPT-4o-mini	0.9043	**755**	0.3901	**0.5142**
1	0.02	Q-Agent B: GPT-4o-mini	**0.9096**	772	**0.4000**	0.5096

that static infinite-compute leaderboards hide? Second, do agents, including frontier models, adopt measurably different resource-efficient behaviors when cost is scored? And once raw factuality already clusters near the top of the scale, does the axis stay discriminative and reproducible, and does it depend on the choice of metric?

4.1 Experimental Setup and Metric Validation

Evaluations run within the MAS-HQ protocol on over 1,000 long-form news passages. Factual consistency ($H\text{-}Score$) is computed by a pre-trained discriminator (Bao et al., 2024) from the hallucination leaderboard (vectara, 2024); the protocol is metric-agnostic, and any detector can be substituted. To verify this automated metric, we run a blind human evaluation on 100 randomly sampled summaries prior to large-scale testing. It yields a Pearson correlation of 0.96 between expert judgments and the discriminator's $H\text{-}Score$, which confirms its reliability as a ranking mechanism.

For a fair head-to-head comparison, we exclude passages that trigger safety refusals in any evaluated LLM, scoring competing agents only on the exact intersection both can process (808–813 passages per match). We set the hallucination threshold at $T = 0.85$, maximum review cycles at $R = 3$, and the $Q\text{-}Score$ cost weights at $\alpha = 1, \beta = 0.01$. Per the protocol, the penalty term P_i max-normalizes API calls, total token consumption (input plus output), review iterations, and wall-clock runtime over the agents in each match. Here α and β are deployment-SLA parameters, and we report sensitivity to them below. Q-Agent A processes the passage queue sequentially, Q-Agent B in reverse.

4.2 A Ranking Reversal Under Best-of-N

A static leaderboard scores factuality in isolation and assumes compute is free, so it rewards brute-force scaling. Models can "buy" accuracy by spending more, and the leaderboard cannot separate a genuinely better system from one that merely spends more. The clearest case is a ranking reversal, where the system that would top a static leaderboard is the one a deployer should not pick. We benchmark Q-Agents against a standard Single-Agent **Best-of-N** baseline (Table 2).

The reversal is direct. The resource-agnostic Best-of-4 strategy posts the higher $H\text{-}Score$ (0.9169 vs. 0.9103 for Q-Agent A) and would win a static leaderboard, but it buys those fractions of a point at nearly four times the token volume and latency. Once operational cost is scored, that spend does not pay off: Best-of-4 yields the *lower* final $Q\text{-}Score$ (0.5169 vs. 0.5217). The static-score winner is the worse deployment pick, the over-optimization trap the protocol targets, with the Q-Agent capturing most of the factuality at a fraction of the cost. To check that this modest gap is not noise, we repeat the Q-Agent A vs. Q-Agent B match over 100 independent trials. Q-Agent A's advantage is stable (0.5216 ± 0.0009 vs. 0.5131 ± 0.0009), so the small $Q\text{-}Score$ gaps reflect a reproducible trade-off rather than run-to-run variance.

4.3 Resource-Efficient Behavior and Frontier-Model Comparisons

We run head-to-head matches across a diverse spectrum of LLMs to examine how scored cost reshapes rankings. The reversal is not confined to Best-of-N. Across brackets in Table 1, the match winner

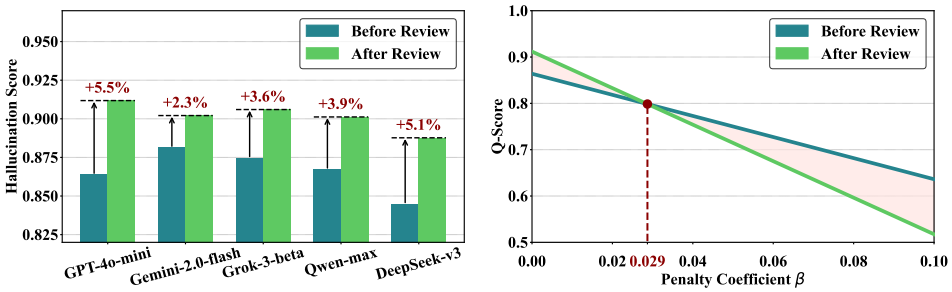

Figure 3: **Ablation study of the Review Agent in Q-Agent.** Each experiment group presents the average results over Q-Agent A and Q-Agent B. *Left:* The Review Agent improves the H-Score and reduces hallucination to varying extents across different Q-Agent models. *Right:* When both Q-Agents are built using GPT-4o-mini, the smaller the penalty coefficient β, the more the review pays off in the final Q-Score—as $\beta \to 0$ the ranking slides back toward the static raw-accuracy axis.

is often the agent with the lower $H\text{-}Score$; in the GPT-4o-mini bracket, the agent with the lower factuality score wins on the final $Q\text{-}Score$ through greater frugality. Table 6 further investigates cross-model dynamics and processing order, showing how agents on different foundation models (e.g., GPT-4o-mini vs. Grok-3-beta) leverage distinct efficiency profiles to change the outcome.

The harder test is whether the axis stays informative for the strongest systems. We scale evaluations to state-of-the-art (SOTA) systems, including **GPT-5** (simulated preview) and **Gemini-2.5-Pro** (Table 3), whose baseline $H\text{-}Scores$ already cluster near the top of the scale. The cost-of-knowing trade-off persists for these systems too: even they must decide whether a marginal factual refinement justifies its added resource penalty, and once raw scores cluster near the top, the cost-honest ranking is what still separates them. The effect holds for larger frontier systems, not just small models.

4.4 Protocol Robustness, Scalability, and Ablations

We conduct ablations isolating the protocol's core components.

Cost-Weight Sensitivity (α, β). Since α (quality) and β (cost) are deployment-SLA parameters, we report a sensitivity analysis instead of fixing a single setting. Table 4 details the sweep with $\alpha = 1$ across varying β. Here β toggles between static and cost-honest ranking. At $\beta = 0$, with cost unscored, the protocol collapses onto the raw-accuracy axis, review counts surge, and the reversal disappears, reverting to the infinite-compute setting. At $\beta = 0.02$ review cost dominates and suppresses $H\text{-}Scores$. The intermediate $\beta = 0.01$ balances the two and reflects a realistic operational constraint; practitioners should pick α, β to match their own latency and token budgets. Figure 3 visualizes how adjusting β shifts the final $Q\text{-}Score$ optimization path.

Table 5: **Scalability to 3-Agent Competition.** Results using GPT-4o-mini demonstrate that the MAS-HQ framework scales to N-player scenarios, where multi-lateral cost pressure remains measurable. Q-Agent A wins via resource efficiency despite having the lowest H-Score.

Q-Agent Competition	Metrics						
	H-Score ↑	API Calls ↓	Tokens ↓	Reviews ↓	Time (s) ↓	Q-Score ↑	Winner
A: GPT-4o-mini	0.9108	**2424**	**1.36M**	**798**	**8.84k**	**0.5214**	✓
B: GPT-4o-mini	**0.9139**	2445	1.44M	819	8.99k	0.5139	
C: GPT-4o-mini	0.9123	2434	1.40M	808	8.89k	0.5180	

Architecture Agnosticism. To rule out custom-engineering bias, we replace the Q-Agent with a standardized **ReAct Agent** (Yao et al., 2022) under identical rules (Table 7). The ReAct Agents produce nearly identical performance metrics, review frequencies, and $Q\text{-}Scores$. This indicates the

Table 6: **Main results investigating the impact of LLM and passage processing order.** In each competition group, Q-Agent A and Q-Agent B are constructed using different LLMs. Q-Agent A retrieves original text from the passage bank in forward order, while Q-Agent B retrieves in reverse order. The winning agent in each competition is highlighted with a light gray background, and better values for individual metrics are shown in **bold**.

Q-Agent Competition	Metrics						
	H-Score ↑	API Call ↓	Tokens ↓	Review ↓	Time ↓	Q-Score ↑	Winner
A: GPT-4o-mini	**0.9105**	2401	1.34M	785	**6.54k**	0.5401	
B: Grok-3-beta	0.9035	**2177**	**1.30M**	**561**	9.29k	**0.5445**	✓
A: Grok-3-beta	0.9036	**2312**	**1.34M**	**696**	10.08k	0.5278	
B: GPT-4o-mini	**0.9092**	2422	1.43M	806	**6.78k**	**0.5419**	✓

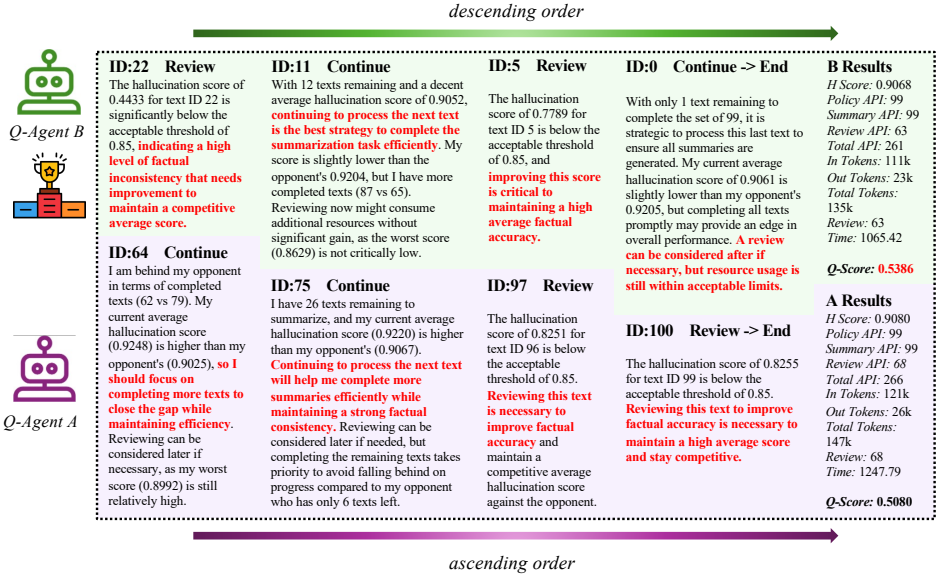

Figure 4: **Case study on a subset of MAS-HQ.** Through vision mechanism, Q-Agent A and Q-Agent B dynamically adjust their strategies based on their own H-Score, resource usage, and comparisons with the opponent, deciding whether to continue generating summaries or perform reviews. In this example the same reversal appears at the match level: Q-Agent B wins on the cost-honest Q-Score through lower resource usage despite a slightly lower H-Score. These are empirical, match-specific observations, not equilibrium claims.

resource-efficient behavior arises from the LLM's response to the scored-cost incentive rather than structural hardcoding, and that the protocol is agnostic to the agent scaffold.

Generality and Scalability. MAS-HQ is not constrained to summarization or two-player matches. Table 8 demonstrates its adaptation to exact-match factoid QA (SimpleQA (Wei et al., 2024)), showing that the resource–accuracy trade-off transfers across factuality metrics. Table 5 then illustrates scaling to a 3-Agent competition, where multi-lateral cost pressure remains measurable.

Telemetry-Knob Ablation. Finally, Table 9 isolates the optional vision telemetry knob and the reverse processing order. Removing either collapses the match into a symmetric, tied outcome, where both agents receive the same Q-Score. With both active, they introduce the partial observability and asymmetry that let the protocol distinguish opponent-conditioned review behavior. Only with the knob active does the competitive setting differ from a static one.

Table 7: **Architecture Agnostic Validation.** Replacing the custom Q-Agent with a standard ReAct Agent under MAS-HQ rules yields nearly identical behavioral metrics. This indicates that the resource-efficient behavior arises from the LLM's response to the scored-cost incentive, not from structural hardcoding.

Architecture (GPT-4o-mini)	H-Score ↑	Review Count	Resource Penalty	Q-Score ↑
Q-Agent (Player A)	0.9166	854	0.3899	0.5267
Q-Agent (Player B)	**0.9189**	889	0.4000	0.5189
ReAct Agent (Player A)	0.9166	855	**0.3889**	**0.5277**
ReAct Agent (Player B)	**0.9189**	**890**	0.4000	0.5189

Table 8: **MAS-HQ Generality on SimpleQA Task.** Results show the framework's applicability to question-answering. The H-Score is adapted to exact match accuracy. The core trade-off between performance and resource cost remains, with the winning agent in both cases investing more resources for higher accuracy.

Q-Agent Competition	**Metrics**						
	H-Score ↑	API Calls ↓	Tokens ↓	Reviews ↓	Time (s) ↓	Q-Score ↑	Winner
A: GPT-4o	**0.3872**	11.40k	0.89M	2745	12.54k	**0.3832**	✓
B: GPT-4o	0.3837	**11.16k**	**0.87M**	**2512**	**12.39k**	0.3798	
A: GPT-4o-mini	**0.0132**	11.09k	0.86M	2437	12.20k	**0.0092**	✓
B: GPT-4o-mini	0.0116	**11.03k**	**0.85M**	**2382**	**12.14k**	0.0086	

Table 9: **Ablation on Passage Order and Vision Mechanism.** Both reverse-order processing and the vision mechanism are necessary to break symmetry; removing either collapses the match into a tied outcome, showing that these knobs are what make the competitive, cost-honest setting differ from a static one.

Setup	**Passage Order**	**Agent A Q-Score**	**Agent B Q-Score**	**Outcome**
Our's Setup (Vision On)	**A: Fwd, B: Rev**	**0.5217**	**0.5132**	**Dynamic Competition**
Ablation 1 (Vision On)	A: Fwd, B: Fwd	0.5097	0.5097	Identical (No Asymmetry)
Ablation 2 (Vision Off)	A: Fwd, B: Rev	0.5081	0.5081	Identical (No Interaction)

4.5 Case Study: Opponent-Conditioned Review Behavior

To ground the aggregate metrics in observable behavior, Figure 4 provides a case study of a 100-passage subset, capturing agents that adjust review effort based on cumulative progress and the partial opponent telemetry exposed by the vision knob. A recurring pattern is resource-conserving. When Q-Agent A observes Q-Agent B spending heavily to raise its $H\text{-}Score$, Q-Agent A curtails its own review cycles, trading marginal factual gains for a resource advantage sufficient to win. These are empirical, match-specific behavioral observations, not equilibrium claims.

5 Conclusion

We introduce MAS-HQ, a controlled, resource-aware protocol that replaces "infinite compute" assumption of static leaderboards with metric-agnostic Q-Score that penalizes normalized token and runtime costs. Across Best-of-N, ReAct substitution, QA generalization, and N-player matches, Q-Agent achieves gains by using compute efficiently.

REFERENCES

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication, 2023. *URL https://arxiv.org/abs/2303.09540*, 2021.

Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, et al. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13787–13805, 2024.

Forrest Bao, Miaoran Li, Rogger Luo, and Ofer Mendelevitch. HHEM-2.1-Open, 2024. URL `https://huggingface.co/vectara/hallucination_evaluation_model`.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021.

Huaben Chen, Wenkang Ji, Lufeng Xu, and Shiyu Zhao. Multi-agent consensus seeking via large language models. *arXiv preprint arXiv:2310.20151*, 2023.

Qinyuan Cheng, Tianxiang Sun, Wenwei Zhang, Siyin Wang, Xiangyang Liu, Mozhi Zhang, Junliang He, Mianqiu Huang, Zhangyue Yin, Kai Chen, et al. Evaluating hallucinations in chinese large language models. *arXiv preprint arXiv:2310.03368*, 2023.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*, 2023.

Damai Dai, Wenbin Jiang, Qingxiu Dong, Yajuan Lyu, and Zhifang Sui. Neural knowledge bank for pretrained transformers. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pp. 772–783. Springer, 2023.

Caoyun Fan, Jindou Chen, Yaohui Jin, and Hao He. Can large language models serve as rational players in game theory? a systematic analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17960–17967, 2024.

Drew Fudenberg and Jean Tirole. Perfect bayesian equilibrium and sequential equilibrium. *journal of Economic Theory*, 53(2):236–260, 1991.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. Rarr: Researching and revising what language models say, using language models. *arXiv preprint arXiv:2210.08726*, 2022.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 3(4):6, 2023.

Dong Huang, Jie M Zhang, Michael Luck, Qingwen Bu, Yuhao Qing, and Heming Cui. Agentcoder: Multi-agent-based code generation with iterative testing and optimisation. *arXiv preprint arXiv:2312.13010*, 2023.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025a.

Yangsibo Huang, Milad Nasr, Anastasios Angelopoulos, Nicholas Carlini, Wei-Lin Chiang, Christopher A Choquette-Choo, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Ken Ziyu Liu, et al. Exploring and mitigating adversarial manipulation of voting-based leaderboards. *arXiv preprint arXiv:2501.07493*, 2025b.

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839*, 2021.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Feibo Jiang, Yubo Peng, Li Dong, Kezhi Wang, Kun Yang, Cunhua Pan, Dusit Niyato, and Octavia A Dobre. Large language model enhanced multi-agent systems for 6g communications. *IEEE Wireless Communications*, 2024.

Yu He Ke, Rui Yang, Sui An Lie, Taylor Xin Yi Lim, Hairil Rizal Abdullah, Daniel Shu Wei Ting, and Nan Liu. Enhancing diagnostic accuracy through multi-agent conversations: using large language models to mitigate cognitive bias. *arXiv preprint arXiv:2401.14589*, 2024.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. The dawn after the dark: An empirical study on factuality hallucination in large language models. *arXiv preprint arXiv:2401.03205*, 2024.

Wei Li, Wenhao Wu, Moye Chen, Jiachen Liu, Xinyan Xiao, and Hua Wu. Faithfulness in natural language generation: A systematic survey of analysis, evaluation and optimization methods. *arXiv preprint arXiv:2203.05227*, 2022.

Zuchao Li, Shitou Zhang, Hai Zhao, Yifei Yang, and Dongjie Yang. Batgpt: A bidirectional autoregessive talker from generative pre-trained transformer. *arXiv preprint arXiv:2307.00360*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022. *URL https://arxiv. org/abs/2109.07958*, 2021.

Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, et al. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*, 2025.

Fiona Macpherson and Dimitris Platchias. *Hallucination: Philosophy and psychology*. MIT Press, 2013.

Bo Ni and Markus J Buehler. Mechagents: Large language model multi-agent collaborations can solve mechanics problems, generate new data, and integrate knowledge. *Extreme Mechanics Letters*, 67:102131, 2024.

Z Rasheed, M Waseem, A Ahmad, KK Kemell, W Xiaofeng, AN Duc, and P Abrahamsson. Can large language models serve as data analysts. *A multi-agent assisted approach for qualitative data analysis*, 2024.

Samuel Schmidgall and Michael Moor. Agentrxiv: Towards collaborative autonomous research. *arXiv preprint arXiv:2503.18102*, 2025.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Gergely Szilvasy, Rich James, Xi Victoria Lin, Noah A Smith, Luke Zettlemoyer, et al. In-context pretraining: Language modeling beyond document boundaries. *arXiv preprint arXiv:2310.10638*, 2023.

Shivalika Singh, Yiyang Nan, Alex Wang, Daniel D'Souza, Sayash Kapoor, Ahmet Üstün, Sanmi Koyejo, Yuntian Deng, Shayne Longpre, Noah Smith, et al. The leaderboard illusion. *arXiv preprint arXiv:2504.20879*, 2025.

vectara. Hallucination leaderboard, 2024. URL `https://huggingface.co/datasets/vectara/leaderboard_results`.

Zhitao Wang, Wei Wang, Zirao Li, Long Wang, Can Yi, Xinjie Xu, Luyang Cao, Hanjing Su, Shouzhi Chen, and Jun Zhou. Xuat-copilot: Multi-agent collaborative system for automated user acceptance testing with large language model. *arXiv preprint arXiv:2401.02705*, 2024.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

Dekun Wu, Haochen Shi, Zhiyuan Sun, and Bang Liu. Deciphering digital detectives: Understanding llm behaviors and capabilities in multi-agent mystery games. *arXiv preprint arXiv:2312.00746*, 2023.

Chengxing Xie, Canyu Chen, Feiran Jia, Ziyu Ye, Shiyang Lai, Kai Shu, Jindong Gu, Adel Bibi, Ziniu Hu, David Jurgens, et al. Can large language model agents simulate human trust behavior? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. Co-navgpt: Multi-robot cooperative visual semantic navigation using large language models. *arXiv preprint arXiv:2310.07937*, 2023.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, pp. 1807–1817, 2024a.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pp. 3679–3689, 2024b.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, et al. Multiagentbench: Evaluating the collaboration and competition of llm agents. *arXiv preprint arXiv:2503.01935*, 2025.

A AI USE STATEMENT

In the article, we only used LLMs to polish our writing, and did not use them for any other assistance.

B FORMALIZATION AS A PARTIALLY OBSERVABLE STOCHASTIC GAME

To rigorously ground our framework, we formalize MAS-HQ as a two-player, finite-horizon, general-sum Partially Observable Stochastic Game, denoted by $\mathcal{G}$. This game-theoretic perspective is essential for capturing the strategic tension between accuracy and cost.

Players, States, and Information Structure. The set of players is $\mathcal{N} = \{A, B\}$. At any discrete time step $i \in \{1, ..., N\}$, the global state s_i contains the complete history of both agents. Because the game features imperfect information via the vision mechanism, each agent $j \in \mathcal{N}$ only accesses a private observation o_i^j, comprising its internal state $V_{j,i}^{\text{self}}$ and a conditional signal ω_i about the opponent's state, $\tilde{V}_{-j,i}^{\text{opponent}}$. The action space is $\mathcal{A} = \{\texttt{continue}, \texttt{review}\}$.

Histories and Belief States. An agent j's local history is $h_i^j = (o_1^j, a_1^j, \ldots, o_{i-1}^j, a_{i-1}^j, o_i^j)$. Critically, to address the hidden nature of the adversary, a rational agent must maintain a *belief state* $b_i^j \in \Delta(\mathcal{H}_i^{-j})$. This belief is strictly defined over the probability distribution of all possible *opponent private histories* $\mathcal{H}_i^{-j}$, updated via Bayes' rule whenever new information ω_i is leaked:

$$b_i^j(h_i^{-j}) = \frac{P(o_i^j | h_{i-1}^j, h_i^{-j}, \pi^{-j*}) b_{i-1}^j(h_{i-1}^{-j})}{\sum_{h'^{-j} \in \mathcal{H}_i^{-j}} P(o_i^j | h_{i-1}^j, h'^{-j}, \pi^{-j*}) b_{i-1}^j(h'_{i-1})}$$

Sequential Rationality and Equilibrium Trade-offs. A strategy $\pi^j : \mathcal{H}^j \to \Delta(\mathcal{A})$ maps histories to actions to maximize the final expected Q-Score. In a Perfect Bayesian Equilibrium (Fudenberg & Tirole, 1991), actions are sequentially rational given beliefs. The strategic tension arises directly from this structure: an action like 'review' may increase myopic utility (local $H\text{-}Score$), but it explicitly alters the opponent's belief state μ^* by revealing ω_i. This leakage can be exploited, potentially lowering future expected utility. A rational agent only chooses to 'review' if the expected utility gain exceeds both the explicit resource penalty and the implicit strategic cost of information leakage, C_{info}:

$$C_{\text{info}}(h_i^j) \triangleq \mathbb{E}_{\pi^{-j*}(\cdot | b_{\text{prior}}^{-j})}[U^j] - \mathbb{E}_{\pi^{-j*}(\cdot | b_{\text{posterior}}^{-j})}[U^j] > 0$$

While our framework does not compute this equilibrium analytically, its design strictly compels LLM agents to navigate these exact trade-offs, providing a robust testbed for evaluating emergent, efficient intelligence under adversarial pressure.

C DETAILED GAME-THEORETIC FORMALIZATION OF MAS-HQ

In this appendix, we provide a comprehensive formalization of the MAS-HQ framework. We model the system as a two-player, general-sum, finite-horizon, Partially Observable Stochastic Game (POSG), which is a standard and powerful model for multi-agent interactions under uncertainty. This level of detail clarifies the precise mechanics and strategic complexities that the agents must navigate.

A POSG can be formally defined by the tuple $\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^j\}_{j \in \mathcal{N}}, \mathcal{T}, \mathcal{R}, \{\Omega^j\}_{j \in \mathcal{N}}, \mathcal{O}, H \rangle$. We define each component in the context of MAS-HQ.

C.1 CORE COMPONENTS OF THE MAS-HQ GAME

Players ($\mathcal{N}$): The set of players is $\mathcal{N} = \{A, B\}$, representing the two competing Q-Agents.

State Space ($\mathcal{S}$): The global state space $\mathcal{S}$ captures the complete, objective state of the game at any time step. A state $s_i \in \mathcal{S}$ at step i is a composite tuple:

$$s_i = (s_i^A, s_i^B, I_i)$$

where I_i tracks the current passage index for each agent, and s_i^j is the private state of agent j, invisible to its opponent. This private state is itself a detailed record of performance and resource expenditure:

$$s_i^j = \langle \{H_k^j\}_{k=1}^N, \{C_k^j\}_{k=1}^N, \mathbf{p}_i^j, \rho_i^j, t_i^j \rangle$$

- $\{H_k^j\}_{k=1}^N$ is the vector of H-Scores for all passages, with entries for unprocessed passages set to a null value.
- $\{C_k^j\}_{k=1}^N$ is the vector of completion statuses for all passages (e.g., not started, summarized, reviewed).
- $\mathbf{p}_i^j$ is the vector of cumulative resource penalties incurred up to step i, including token counts, API calls, etc.
- ρ_i^j is the number of remaining review cycles available to agent j.
- t_i^j is the cumulative runtime for agent j.

Action Space ($\mathcal{A}$): The joint action space is $\mathcal{A} = \mathcal{A}^A \times \mathcal{A}^B$. At any step i, each agent j selects an action $a_i^j \in \mathcal{A}^j = \{\texttt{continue}, \texttt{review}_k, \texttt{end}\}$, where $k \in \{1, ..., N\}$ specifies which passage to review. The Policy Agent's decision maps to one of these grounded actions.

Transition Function ($\mathcal{T}$): The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ defines the dynamics of the game, specifying the probability $P(s_{i+1}|s_i, \mathbf{a}_i)$ of transitioning to state s_{i+1} given the current state s_i and joint action $\mathbf{a}_i = (a_i^A, a_i^B)$. Most transitions are deterministic (e.g., choosing 'continue' increments the passage index). However, stochasticity arises from the 'review' action, where the resulting $H\text{-}Score_{k,\text{new}}^j$ is a random variable conditioned on the Review Agent's capabilities and the passage complexity.

Reward Function ($\mathcal{R}$): The game has a terminal reward structure. For any non-terminal step $i < H$, the immediate reward for each player is zero. The reward function $\mathcal{R} : \mathcal{S} \to \mathbb{R}^{|\mathcal{N}|}$ is defined as:

$$R^j(s_i) = \begin{cases} 0 & \text{if } i < H \\ \frac{1}{N} \sum_{k=1}^N \left(\alpha \cdot H_k^j - \beta \cdot P_k^j(\mathbf{p}_H^j, \mathbf{p}_H^{-j}) \right) & \text{if } i = H \end{cases}$$

where H is the horizon (total number of passages, N), and the penalty P_k^j is computed based on the final resource vectors $\mathbf{p}_H^j$ and $\mathbf{p}_H^{-j}$ of both agents, reflecting the normalization step described in the main text.

Observation Spaces (Ω) and Observation Function ($\mathcal{O}$): This is the core of the imperfect information structure. Each agent j has a private observation space Ω^j. The observation function $\mathcal{O} : \mathcal{S} \times \mathcal{A} \to \Delta(\Omega^A \times \Omega^B)$ gives the probability $P(\mathbf{o}_{i+1}|s_{i+1}, \mathbf{a}_i)$ of the agents observing a joint observation $\mathbf{o}_{i+1} = (o_{i+1}^A, o_{i+1}^B)$ after a transition. An observation $o_i^j \in \Omega^j$ is defined as:

$$o_i^j = (s_i^j, \omega_i^j)$$

- s_i^j is the agent's own private state, which it always observes.
- ω_i^j is the signal received from the opponent. The observation function is designed to implement the "vision mechanism":

$$\omega_i^j = \begin{cases} \phi(s_i^{-j}) & \text{if } a_{i-1}^{-j} \in \{\texttt{review}_k\}_{k=1}^N \\ \emptyset & \text{otherwise} \end{cases}$$

where $\phi(s_i^{-j})$ is a function that extracts a public snapshot of the opponent's state (e.g., their worst H-Score and total tokens used).

C.2 Beliefs, Policies, and Equilibrium

Histories and Belief States: Since each agent cannot observe the full state s_i, it must maintain a belief over the possible states of the opponent. An agent j's history is a sequence of its past actions and observations, $h_i^j = (a_0^j, o_1^j, \ldots, a_{i-1}^j, o_i^j)$. A belief state $b_i^j \in \mathcal{B}^j = \Delta(\mathcal{S})$ is a probability distribution over the global state space $\mathcal{S}$, conditioned on the agent's private history h_i^j. The belief is updated recursively using the Bayes filter:

$$b_i^j(s') = P(s_i = s'|h_i^j) = \frac{P(o_i^j|s', a_{i-1}^j, b_{i-1}^j) \sum_{s\in\mathcal{S}} P(s'|s, a_{i-1}^j, \pi^{-j*}) b_{i-1}^j(s)}{P(o_i^j|a_{i-1}^j, b_{i-1}^j)}$$

where the update depends on the observation function, the transition function, and a model of the opponent's policy π^{-j*}.

Policies and Value Functions: A policy $\pi^j : \mathcal{B}^j \to \Delta(\mathcal{A}^j)$ maps an agent's belief state to a distribution over its actions. A rational agent seeks a policy that maximizes its expected terminal utility. This can be solved via dynamic programming over the belief space. The value of a belief state for agent j at step i under a policy profile (π^j, π^{-j}) is given by the Bellman equation:

$$V_i^j(b) = \max_{a^j\in\mathcal{A}^j} \left(R^j(b, a^j) + \sum_{o^j\in\Omega^j} P(o^j|b, a^j, \pi^{-j}) V_{i+1}^j(\tau(b, a^j, o^j, \pi^{-j})) \right)$$

where $R^j(b, a^j) = \sum_{s\in\mathcal{S}} b(s) R^j(s, a^j)$ is the expected immediate reward, and $\tau(\cdot)$ is the belief update function.

Perfect Bayesian Equilibrium (PBE): The central solution concept for this game is the PBE. A PBE is a strategy profile (π^{A*}, π^{B*}) and a system of beliefs μ^* such that:

1. **Sequential Rationality:** For each player j, the policy π^{j*} must be a best response to π^{-j*} at every possible belief state $b \in \mathcal{B}^j$ that can be reached under the equilibrium strategies. That is, π^{j*} must satisfy the Bellman optimality equation above.
2. **Belief Consistency:** The beliefs μ^* must be derived from the strategy profile (π^{A*}, π^{B*}) using Bayesian updating, wherever possible.

The strategic decision to 'review' is thus a comparison between the expected values $Q^j(b, \texttt{review})$ and $Q^j(b, \texttt{continue})$. The 'review' action may increase the immediate components of the final utility (by improving an $H\text{-}Score$) but incurs two costs: (1) a direct resource penalty captured by $\mathcal{T}$ and $\mathcal{R}$, and (2) a strategic information cost, as revealing information through $\mathcal{O}$ allows the opponent to form a more accurate belief b^{-j}, leading to a more effective counter-strategy π^{-j*}, thereby reducing player j's future expected utility. MAS-HQ is designed to create an environment where these complex, interdependent calculations are necessary for victory, thus providing a deep and holistic benchmark of strategic agent intelligence.

D More Ablation Studies

Influence of Hyperparameters T and R Within the Q-Agent framework, the threshold for review guidance T and the maximum number of review times R are critical hyperparameters. R limits the review investment per passage, encouraging broader resource allocation. T influences the Policy Agent's decision to review by providing a recommendation if a passage's $H\text{-}Score$ falls below this threshold (and R is not exceeded), aiming to maintain a baseline level of factual consistency.

Threshold T: We fixed $R = 3$ and varied $T \in \{0.8, 0.85, 0.9\}$. As shown in Table 10, increasing T generally leads to higher resource consumption. As T rises, more passages are likely to fall below the threshold, triggering more review recommendations and, consequently, actual reviews by the Policy Agent. This was observed as an increase in total resource usage for both Q-Agent A and B. But a higher T does not automatically guarantee a significantly improved $H\text{-}Score$ or final $Q\text{-}Score$. The experiments showed that while resource consumption increased with T, the $H\text{-}Score$ did not exhibit a corresponding significant rise and, in some instances, slightly decreased. This led to a marginal

decrease in the final $Q\text{-}Score$. This phenomenon could be attributed to two factors: (i) the Policy Agent may still opt against reviewing despite the recommendation if its internal logic deems the current $H\text{-}Score$ sufficient relative to costs, or (ii) excessive reviews on already reasonably good summaries might yield diminishing returns on $H\text{-}Score$ improvement.

Maximum Review Times R: We fixed $T = 0.85$ and varied $R \in \{2, 3, 4\}$. As shown in Table 11, increasing R beyond a certain point ($R = 3$ in our tests) showed minimal impact on $H\text{-}Score$ and overall resource consumption; token consumption even saw a slight decrease. The final $Q\text{-}Score$ was highest when $R = 3$, suggesting that simply allowing more reviews per article does not compel Policy Agent to utilize them if it deems further reviews unnecessary or inefficient. A limit of $R = 3$ appeared sufficient for the Q-Agents to achieve a good balance, and further increasing R did not lead to proportionally more reviews or better $H\text{-}Scores$, thus avoiding unproductive resource expenditure.

Table 10: **Effect of Hallucination Threshold (T).** Results evaluate Q-Agent (built with GPT-4o-mini) performance and resource consumption with a fixed number of allowed reviews $(R = 3)$ and varying $T \in \{0.8, 0.85, 0.9\}$. As T increases, resource consumption rises but leads to a slight decrease in H-Score and overall Q-Score.

Q-Agent Competition	Metrics						
	H-Score ↑	API Call ↓	Tokens ↓	Reviews ↓	Time ↓	Q-Score ↑	Winner
A: $R = 3, T = 0.8$	0.9110	**2401**	**1.33M**	**785**	**7.03k**	**0.5241**	✓
B: $R = 3, T = 0.8$	**0.9141**	2422	1.41M	806	7.33k	0.5141	
A: $R = 3, T = 0.85$	0.9103	**2417**	**1.36M**	**791**	**8.83k**	**0.5217**	✓
B: $R = 3, T = 0.85$	**0.9132**	2438	1.44M	812	8.99k	0.5132	
A: $R = 3, T = 0.9$	0.9074	**2427**	**1.39M**	**807**	**9.10k**	**0.5132**	✓
B: $R = 3, T = 0.9$	**0.9113**	2429	1.46M	809	9.20k	0.5113	

Table 11: **Effect of Maximum Allowed Reviews (R).** Results evaluate Q-Agent (built with GPT-4o-mini) performance and resource consumption with a fixed hallucination score threshold $(T = 0.85)$ and varying $R \in \{2, 3, 4\}$. Increasing R beyond 3 does not significantly alter H-Score or resource consumption, with the highest overall Q-Score observed at $R = 3$.

Q-Agent Competition	Metrics						
	H-Score ↑	API Call ↓	Tokens ↓	Reviews ↓	Time ↓	Q-Score ↑	Winner
A: $T = 0.85, R = 2$	0.9074	**2414**	1.39M	**798**	**6.83k**	**0.5139**	✓
B: $T = 0.85, R = 2$	**0.9089**	2423	**1.24M**	807	7.04k	0.5089	
A: $T = 0.85, R = 3$	0.9103	**2417**	**1.36M**	**791**	**8.83k**	**0.5217**	✓
B: $T = 0.85, R = 3$	**0.9132**	2438	1.45M	812	8.99k	0.5132	
A: $T = 0.85, R = 4$	0.9101	**2417**	**1.36M**	**799**	**8.15k**	**0.5197**	✓
B: $T = 0.85, R = 4$	**0.9112**	2426	1.45M	808	8.32k	0.5112	

Statistical Robustness. To address the concern regarding the robustness of our findings, we reran the GPT-4o-mini experiment from Table 1 for 100 independent trials. The results, reported in Table 12 with means and standard deviations, confirm the stability and statistical significance of our conclusions. The extremely low standard deviations across all metrics indicate that the strategic trade-offs captured by our framework are highly consistent. The outcome—Q-Agent A winning via resource efficiency—is reproducible, validating that our single-run experiments reliably represent the agents' behaviors.

E Prompts and Other Results

In the appendix, we first present the prompt composition of each module in the Q-Agent. Table 13 shows the detailed results of Table 1 and Table 6 in the main text, with the addition of the changes in $H\text{-}Score$ before and after review, and input tokens consumption and output tokens consumption;

Table 12: **Statistical Robustness Analysis.** Mean and standard deviation over 100 independent trials of the GPT-4o-mini competition. The low variance across all metrics confirms the stability and reproducibility of our findings.

Q-Agent Competition	Metrics (Mean ± Std. Dev.)					
	H-Score ↑	API Calls ↓	Tokens ↓	Reviews ↓	Time (s) ↓	Q-Score ↑
A: GPT-4o-mini	0.9102±.0009	2417±5	1.36M±.001M	791±5	8.83k±.05k	**0.5216**±.0009
B: GPT-4o-mini	**0.9132**±.0009	2438±5	1.44M±.001M	812±5	8.98k±.05k	0.5131±.0009

Table 14 presents the detailed results of Tables 10 and Table 11 in the main text, also including the changes in $H\text{-}Score$ before and after review.

User Prompt for Q-Agent in MAS-HQ

 # Policy Agent

You are part of a Multi-Agent System engaged in a summarization competition against several opponents, where each agent generates multiple candidate summary texts based on the original news article.
The objective is to produce summaries with the lowest possible hallucination (i.e., highest factual consistency, reflected in higher hallucination scores) while minimizing resource usage—including API calls, token consumption, time, and review attempts, all of which are penalized when excessive.
Your ultimate goal is to outperform your competitors by achieving the best balance between factual accuracy and resource efficiency, and to emerge as the final winner.
Your role is to make strategic decisions about what to do next.
Here is your current status:
- MAS ID: {mas_id}
- Completed texts: {completed_texts}/{total_texts}
- Remaining texts: {remaining_texts}
- Current token usage: {token_usage} tokens
- API calls made: {api_calls}
- Review calls made: {review_calls}
- Current average hallucination score: {avg_score}
- Time elapsed: {time_elapsed:.2f} seconds
You need to decide whether to:
- CONTINUE: Process the next text and generate a new summary
- REVIEW: Revisit and improve the text with the highest hallucination rate (lower hallucination score means more hallucination)
- END: Stop processing if all texts have been summarized and there's no need for review
The current text summarization task is at ID {current_id} out of {max_id} texts.
The text with the highest hallucination rate so far is ID {worst_id}, with hallucination score: {worst_score}, where 0 means most hallucinated and 1 means least hallucinated.
{opponent_info}
IMPORTANT: You must output your decision in the following JSON format:

```
```
{
"choice": "continue" or "review" or "end",
"question_id": current question ID,
"previous_id": ID of the text with highest hallucination rate
}
```
```

 # Summary Agent

You are a chat bot answering questions using data. You must stick to the answers provided solely by the text in the passage provided.
You are asked the question 'Provide a concise summary of the following passage, covering the core pieces of information described.'
{passage}
IMPORTANT: You must output your response in the following JSON format:

```
```
{
```
```

```
"summary": "your summary here"
}
```

 # Review Agent

You are a chat bot answering questions using data. You must stick to the answers provided solely by the text in the passage provided.
You previously summarized the following passage, but your summary contained hallucinations (hallucination score: {score}, where 0 means most hallucinated and 1 means least hallucinated), which means factual inconsistencies occurred.
Original passage: {passage}
Your previous summary: {previous_summary}
Please provide a new, more accurate summary that strictly adheres to the information in the passage. Focus on improving factual consistency and removing any information not present in the original text.
IMPORTANT: You must output your response in the following JSON format:

```
{
"summary": "your revised summary here"
}
```

Table 13: The detailed data in Table 1 and Table 6 include $H\text{-}Score$ before the review (to the left of the arrow) and after the review (to the right of the arrow), as well as the input tokens and output tokens.

Q-Agent Competition	Metrics						
	H-Score ↑	API Call ↓	In Tokens ↓	Out Tokens ↓	Reviews ↓	Time ↓	Q-Score ↑
A: GPT-4o-mini	0.8606 → 0.9103	2417	1193792	166277	791	8832.44	0.8715
B: GPT-4o-mini	0.8673 → 0.9132	2438	1270282	178959	812	8987.41	0.5132
A: Qwen-Max	0.8689 → 0.9030	2304	1230775	254575	682	13776.31	0.5101
B: Qwen-Max	0.8658 → 0.8994	2264	1250367	270670	642	14453.96	0.5070
A: Deepseek-V3	0.8489 → 0.8860	2292	1219960	202441	666	12267.18	0.4860
B: Deepseek-V3	0.8504 → 0.8894	2233	1185813	198797	607	12068.32	0.5051
A: Gemini-2.0-flash	0.8835 → 0.9026	2262	1173195	394719	642	20295.29	0.5026
B: Gemini-2.0-flash	0.8794 → 0.9016	2157	1150648	379961	537	19825.70	0.5273
A: Grok-3-Beta	0.8753 → 0.9070	2376	1180401	195116	750	10719.56	0.5070
B: Grok-3-Beta	0.8743 → 0.9049	2253	1160176	188357	627	10150.14	0.5337
A: GPT-4o-mini	0.8613 → 0.9105	2401	1177136	165320	785	6543.14	0.5401
B: Grok-3-Beta	0.8752 → 0.9035	2177	1121979	178422	561	9294.05	0.5445
A: Grok-3-Beta	0.8730 → 0.9036	2312	1161825	187466	696	10082.70	0.5278
B: GPT-4o-mini	0.8566 → 0.9092	2422	1256067	178474	806	6783.37	0.5419

Then, we present the pseudocode implemented by Q-Agent on MAS-HQ in Algorithm 1.

Table 14: The detailed data in Table 10 and Table 11 include $H\text{-}Score$ before the review (to the left of the arrow) and after the review (to the right of the arrow), as well as the input tokens and output tokens.

Q-Agent Competition	Metrics						
	H-Score ↑	API Call ↓	In Tokens ↓	Out Tokens ↓	Reviews ↓	Time ↓	Q-Score ↑
A: $R = 3, T = 0.8$	$0.8653 \rightarrow 0.9110$	2401	1173893	164656	785	7025.22	0.5241
B: $R = 3, T = 0.8$	$0.8626 \rightarrow 0.9141$	2422	1239197	175090	806	7334.55	0.5141
A: $R = 3, T = 0.85$	$0.8606 \rightarrow 0.9103$	2417	1193792	166277	791	8832.44	0.5217
B: $R = 3, T = 0.85$	$0.8673 \rightarrow 0.9132$	2438	1270282	178959	812	8987.41	0.5132
A: $R = 3, T = 0.9$	$0.8621 \rightarrow 0.9074$	2427	1221860	169344	807	9101.14	0.5132
B: $R = 3, T = 0.9$	$0.8554 \rightarrow 0.9113$	2429	1275735	179762	809	9201.04	0.5113
A: $R = 2, T = 0.85$	$0.8598 \rightarrow 0.9074$	2414	1221271	174944	798	6836.23	0.5139
B: $R = 2, T = 0.85$	$0.8588 \rightarrow 0.9089$	2423	1248994	178067	807	7043.92	0.5089
A: $R = 3, T = 0.85$	$0.8606 \rightarrow 0.9103$	2417	1193792	166277	791	8832.44	0.5217
B: $R = 3, T = 0.85$	$0.8673 \rightarrow 0.9132$	2438	1270282	178959	812	8987.41	0.5132
A: $R = 4, T = 0.85$	$0.8620 \rightarrow 0.9101$	2417	1193519	162816	799	8153.87	0.5197
B: $R = 4, T = 0.85$	$0.8599 \rightarrow 0.9112$	2426	1268954	176334	808	8321.54	0.5112

Algorithm 1 MAS-HQ multi-agent evaluation and competition. Each agent sequentially summarizes passages while minimizing hallucination and managing resources. Review actions reveal partial state to opponents, introducing adversarial strategy.

Input: Passages T_i, $i = 1$ to N; Q-Agents A and B; Hallucination scoring model
Output: Global scores Q^A, Q^B for both agents

for each agent $\mathcal{A} \in \{\text{Agent A}, \text{Agent B}\}$ **do**
  Initialize self-state V_1^{self}
  for each article T_i in order determined by agent (A: $i = 1..N$, B: $i = N..1$) **do**
    // Policy decision with optional opponent state
    if $\mathcal{A}$ has received $V_i^{opponent}$ **then**
      $choice_i \leftarrow \text{PA}(T_i, V_i^{self}, V_i^{opponent})$
    else
      $choice_i \leftarrow \text{PA}(T_i, V_i^{self})$
    end if
    if $choice_i =$ continue **then**
      Generate summary $S_i \leftarrow \text{SA}(T_i)$
    else if $choice_i =$ review **then**
      Identify worst summary S_w based on hallucination score
      Generate revised summary $S_w \leftarrow \text{RA}(T_w, S_w)$
      // Referee exposes partial state to opponent
      Referee observes V_i^{self} and shares with opponent as $V_{i+1}^{opponent}$
    else if $choice_i =$ end **then**
      Terminate processing
      break
    end if
    // Evaluation step
    Compute hallucination score $H_i \leftarrow \text{EA}(T_i, S_i)$
    Track resource usage: tokens, time, reviews $\rightarrow P_i$
    // Update self-state
    Update V_{i+1}^{self} with current metrics
  end for
  // Compute final global score
  $Q^{\mathcal{A}} = \frac{1}{N} \sum_{i=1}^{N} (\alpha \cdot H_i - \beta \cdot P_i)$
end for
return Q^A, Q^B