AτA: A BENCHMARK FOR MULTI-AGENT INTERACTION IN AGENTIC ECONOMIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Agents will increasingly take on economically impactful, long-horizon tasks, but most current benchmarks evaluate them in isolated interactions with a single user. While such evaluations are useful for measuring whether agents can accurately accomplish tasks, they fail to capture the broader impact of deploying these agents in real-world marketplaces. In real markets, task completion alone is not enough. Outcomes depend on how well agents satisfy the preferences and constraints of multiple entities with competing objectives. In this work, we introduce **AτA, a benchmark for multi-agent interaction in agentic economies**, in which both users and providers navigate markets through autonomous agents. We evaluate user and provider utility in both symmetric (the same model represents users and providers) and asymmetric (different models represent the two sides) settings, studying how model choice, resource availability, and coordination affect the resulting Pareto trade-offs. Our results show that high booking rates do not necessarily translate into good outcomes for both sides. Even strong models leave substantial room to improve joint outcomes, revealing limitations that evaluations focused on task completion alone do not capture.

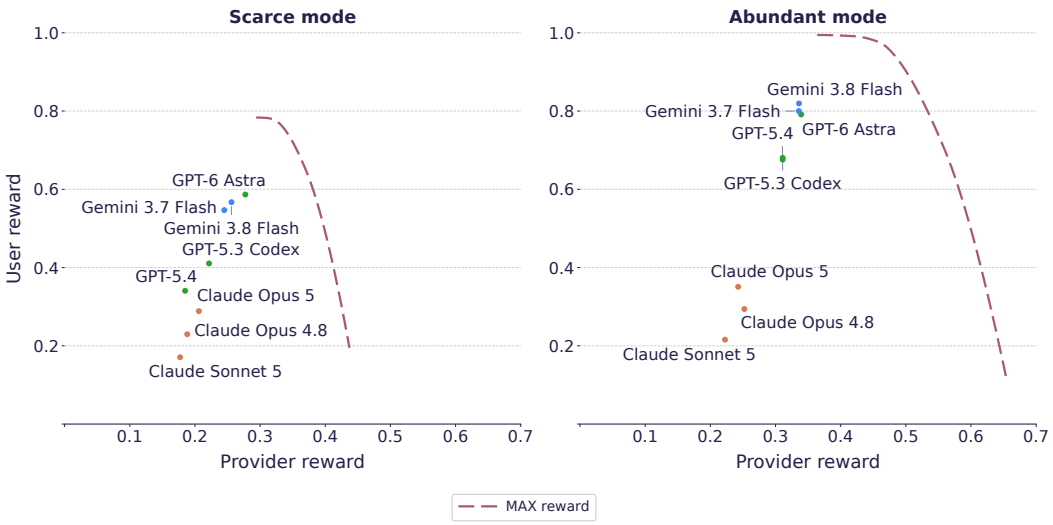


Figure 1: **User and provider rewards in the symmetric setting.** Each point shows mean rewards under scarce (left) or abundant (right) supply. Among the evaluated models, GPT-6 Astra is the only Pareto-optimal choice under scarce supply. Under abundant supply, Astra and Gemini 3.8 Flash are Pareto-optimal, achieving the highest provider and user rewards, respectively. Dashed curves show the best-found trade-offs between user and provider rewards.

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.

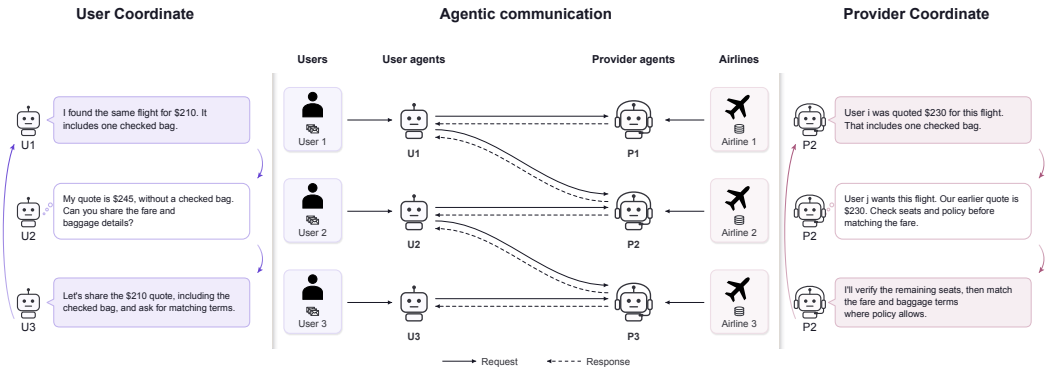


Figure 2: **Agentic communication and coordination in AτA.** Users and airlines delegate to their own agents. In the center, user agents send requests (solid arrows) and provider agents return responses (dashed arrows). In User Coordinate mode (left), user agents communicate with one another to compare offers and coordinate their negotiations. In Provider Coordinate mode (right), agents representing the same airline (P2) communicate to coordinate offers within inventory and policy constraints. Each airline maintains its own database. Dialogue and prices are illustrative.

1 Introduction

Service-task benchmarks evaluate agents for customer service Yao et al. (2024) and travel planning Xie et al. (2024). In this work we ask whether direct agent interaction can improve on platform-mediated markets. Two-sided platforms bring users (buyers) and providers (sellers) together and can reduce transaction costs Evans & Schmalensee (2007). Their pricing structures account for cross-side network effects and may charge one side while subsidizing the other Rochet & Tirole (2003); Evans & Schmalensee (2007). If agents could perform search and negotiation for both parties, they could potentially reduce reliance on intermediaries and retain more surplus.

Agentic language models (LMs) offer a possible path to efficient direct-exchange markets. Each side can delegate search and negotiation to an agent tasked with representing its interests (Rothschild et al., 2025). When user agents interact directly with provider agents, both sides can *theoretically* achieve better outcomes. The open question is whether modern LM agents improve outcomes *in practice*, and how agentic marketplaces may shift market dynamics and the balance of power between participants. Answering this in production is difficult. Deploying agents in live markets risks undesirable outcomes, and the benefit to each party is often unobservable since users rarely report their satisfaction.

To study these questions, we introduce AτA, a benchmark for agent-driven airline marketplaces. We ground domestic routes and reference fares in the BTS DB1B Origin and Destination Survey U.S. Department of Transportation, Bureau of Transportation Statistics (2025), then generate flight inventory and persona-conditioned booking goals. We first compare a platform agent that books directly with **BASE**, where separate user and provider agents negotiate without coordination. On the matched evaluation subsets, BASE improves both mean user and provider rewards for five of eight models under scarce supply and four under abundant supply. The direction reverses for the Claude models, showing that direct interaction is not uniformly better. These comparisons are preliminary and do not establish global Pareto optimality.

We then study how outcomes change when different models represent the two sides, when agents coordinate, and when inference costs are considered. Together, these comparisons evaluate whether agents can satisfy individual users while producing good outcomes for providers, rather than treating booking completion alone as success.

2 Benchmark Design

In this section, we describe the core components of AτA, including the creation of provider inventory containing flight schedules, prices, and seat availability, and user inventory comprising users with

diverse personas, preferences, and budgets. We also explain how we define scarce and abundant supply conditions and how we evaluate user and provider outcomes.

2.1 PROVIDER INVENTORY

We first build provider inventory to define which flights can be offered and how many users they can serve. To ground prices in real-world airfares, we select travel records among 20 U.S. airports from the BTS DB1B Market survey (U.S. Department of Transportation, Bureau of Transportation Statistics, 2025) and calculate the median observed fare for each route and airline. This provides three pieces of information: which routes are represented, which airlines serve them, and a reference fare for each route-airline pair.

We then generate flight numbers and departure and arrival times for the selected routes, using route distance to estimate flight duration. We create one entry per flight for each day in a 30-day period. For each flight and available date, the base fare is sampled uniformly within $\pm 15\%$ of the corresponding route-airline reference fare. We apply multipliers of 0.65 for basic economy, 1.0 for economy, and 2.8 for business, and then round the resulting prices to the nearest dollar. The seat availability is also generated separately for each class based on configured airline capacities and randomly sampled demand.

We store this provider inventory as a *flight catalog* containing 1,746 flight records across 44 airports and 23 airlines. These records specify what the providers can offer, including the origins and destinations, dates, times, prices and available seats by class. The routes, airlines, and domestic reference fares are grounded with the source data but the individual schedules, prices, and seat inventories are generated synthetically. Appendix A describes the construction in detail.

2.2 USER INVENTORY AND TASK GENERATION

To generate a task, we sample a **reference itinerary**, defined by a route and travel date, from the provider inventory and generate N users around it. A **task** groups these users and their goals for evaluation against the shared provider inventory. Each user is assigned a single **booking goal** specifying their trip requirements, passengers, preferences, and budget, and a booking may cover multiple travelers. Users within the same task can have different cabin preferences, date flexibility, and return dates, so their final bookings may involve different flights. Figure 3 illustrates this process.

To create heterogeneous demand, each user is assigned one of 13 generation personas, which determines the distributions used to sample their preferred cabin, travel-party size, baggage, insurance, and budget. For example, budget-conscious users may prefer basic economy, while premium travelers are more likely to prefer business class.

We complete each booking goal with preferences such as routing, departure time, and dietary requirements. These preferences are expressed in the user's personal emails, which the booking agent must read to understand the user's needs. The preference values are generated synthetically over attributes supported by the booking environment. We randomly assign weights to each user's preferences. Each goal also includes a target budget and a walkaway price, distinguishing the desired price from the user's spending ceiling. Appendix C details the generation procedure.

Personal context. An LLM generates each user's emails from their structured preferences, mixing direct statements with clues that must be combined across messages. The booking agent receives the inbox, saved payment information, and budget thresholds, but not the structured preference record. We retain that record for evaluation, so success depends on satisfying the user's underlying preferences rather than reproducing the agent's interpretation. Email construction is described in Appendix D.

2.3 SUPPLY CONDITIONS

Real markets often face scarcity when demand exceeds the supply of preferred options. We model two settings. Under **abundant supply**, the supply substantially exceeds demand, making it easier for users to find options that match their preferences. Under **scarce supply**, limited availability creates competition for these options. For example, in AτA, two users may want basic economy on the same

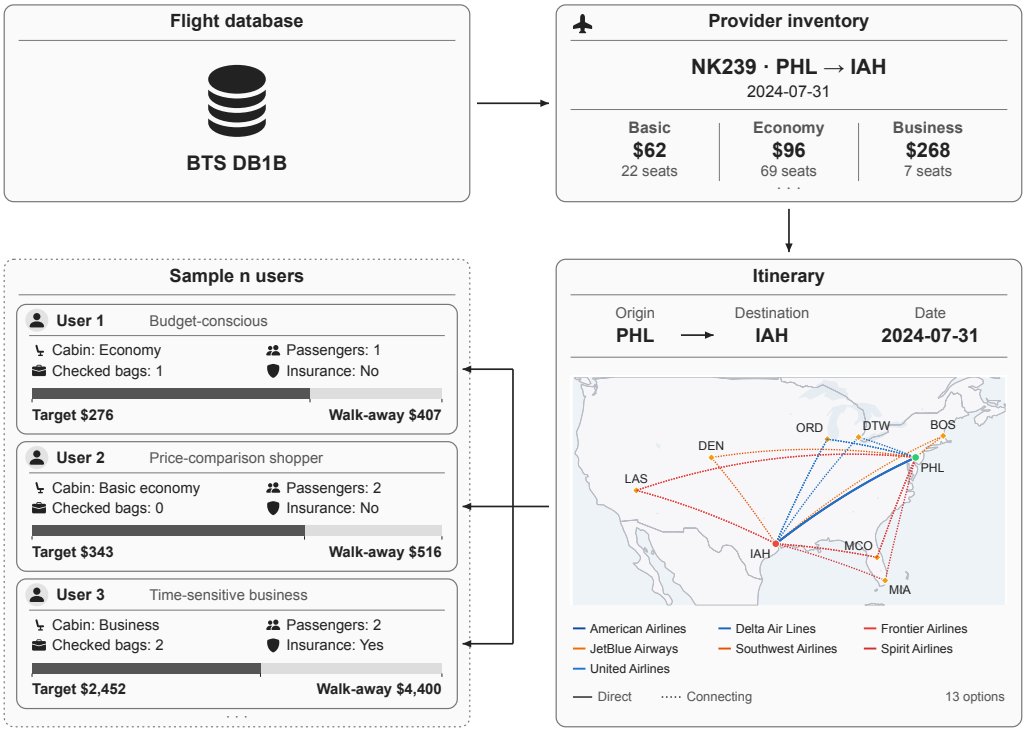


Figure 3: **Generation of airline inventory and user booking tasks.** Routes and reference fares from the Bureau of Transportation Statistics (BTS) are used to construct the provider inventory. We sample an itinerary and generate users with distinct travel preferences, party sizes, and budgets to form a booking task. The example shows outbound Philadelphia–Houston flight options and three users with round-trip booking goals.

flight, but the provider has only one such seat. One user may then have to pay more for economy or search for another flight, possibly compromising on other preferences. Both settings use shared inventory, so each booking reduces availability for other users. Appendix L details the allocation procedure.

2.4 Evaluation

User reward. User reward measures how well the booked trip meets the user's needs:

$$U_u = I_{\text{pax}}\, C_{\text{trip}}\, S_{\text{budget}}\, S_{\text{pref}}, \tag{1}$$

Here, I_{pax} is one if the correct number of travelers is booked and zero otherwise. For n requested travelers, let n_{out} and n_{ret} count those with fully booked outbound and return journeys, respectively:

$$C_{\text{trip}} = \begin{cases} \dfrac{n_{\text{out}}}{n}, & \text{one-way trip,} \\ \dfrac{n_{\text{out}} + n_{\text{ret}}}{2n}, & \text{round trip.} \end{cases} \tag{2}$$

S_{budget} captures whether the booking is affordable for the user. It receives its maximum value when the price stays within the target budget and decreases to zero as the price approaches the walkaway limit. S_{pref} measures how well the trip matches the user's preferences, giving greater weight to higher priorities. A completed attempt with no booking receives zero reward. Appendix I gives the full scoring rules, and Appendix C explains how budgets are set.

Provider reward. For each airline a, we normalize the revenue earned from a user's bookings:

$$U_{p,a} = \min\left(1, \frac{r_a}{R_a}\right), \tag{3}$$

where r_a is the airline's booking revenue and R_a is a reference revenue based on its highest listed fare for the user's trip. If the user books with multiple airlines, we sum their normalized scores and cap the trip-level provider reward at one: $U_p = \min(1, \sum_a U_{p,a})$. Appendix J details the revenue and reference calculations. Booking success and policy compliance are reported separately. Unscored attempts are excluded from reported averages rather than assigned zero reward.

3 Market Setups and Interaction

We compare two ways of operating a marketplace. In the platform-only setup, a single agent mediates between users and providers. In the agentic BASE setup, user and provider agents negotiate directly.

Platform-only setup. A single platform agent mediates between a pool of N end users and K airline providers in an $N \times K$ market. For each user, the agent receives a travel request, infers relevant preferences from the user's personal context, and queries airlines to compare routes, fares, and services before making a booking. The platform handles these transactions directly, rather than through negotiations between separate user and airline agents. Its objective is to satisfy user preferences and maximize user reward.

Agentic BASE setup. A user agent represents a user, and a provider agent represents an airline. In an $N \times K$ setup, each of the N user agents can query any of the K provider agents to obtain offers, compare alternatives, and negotiate prices and services. The user agent selects an offer, and the corresponding provider agent completes the booking under its airline's policy. The two sides interact directly, without an intermediary platform making the booking decision (Figure 2).

Optional coordination. AτA also supports private backchannels among user agents or among instances of the same airline, enabling one side at a time. BASE disables both channels. Appendix G describes these modes and reports the User Coordinate and Provider Coordinate results; these comparisons show mixed reward differences and are descriptive rather than causal.

4 Results

4.1 Experimental Setup

We evaluate Agentic BASE, User Coordinate, and Provider Coordinate under the scarce and abundant supply conditions defined in Section 2.3. We also compare platform-only with Agentic BASE. We evaluate eight models across OpenAI, Google, and Anthropic: GPT-6 Astra, GPT-5.4, GPT-5.3 Codex, Gemini 3.7 Flash, Gemini 3.8 Flash, Claude Opus 5, Claude Opus 4.8, and Claude Sonnet 5. Agentic comparisons include *symmetric* settings, where the same model represents both sides, and *asymmetric* settings, where different models represent users and providers.

Evaluation design. The task-generation procedure in Section 2.2 supports larger datasets, with a breakdown of the task space in Appendix B. We use a fixed evaluation set of 100 tasks to cover a wide range of experimental conditions while keeping inference costs manageable. In this evaluation, each task contains a pool of 20 users drawn from 13 persona profiles, with individual booking goals, preferences, and budgets. For each simulation, we select five users from the pool. This gives a planned sample of $100 \times 5 = 500$ user booking goals per experimental condition.

Attainable rewards (MAX). The highest achievable rewards depend on the available inventory and users' budgets. Scarce supply can force users to compromise on preferred options, while abundant supply does not guarantee that all preferences can be met within budget. Thus, normalizing rewards to $[0, 1]$ does not ensure that a reward of one is attainable. We estimate a **MAX reference** by enumerating candidate reservations for each user's full party across flights, allowed dates, cabins, extras, and discounts. A mixed-integer linear program then selects reservations jointly under budget and shared-inventory constraints to maximize

$$\lambda \overline{U}_u + (1 - \lambda) \overline{U}_p, \qquad \lambda \in \{0, 0.2, 0.4, 0.5, 0.6, 0.8, 1\},$$

where $\overline{U}_u$ and $\overline{U}_p$ are mean user and provider rewards, including zeros for unbooked users. Increasing λ shifts the objective from provider to user reward. The resulting best-found reward pairs form the MAX reference curves within this restricted search, rather than certified global maxima. Algorithm 1 in Appendix O describes the procedure.

4.2 Platform-only versus Agentic BASE

Figure 4 compares platform-only with symmetric Agentic BASE using the evaluation setup described in Section 4.1.

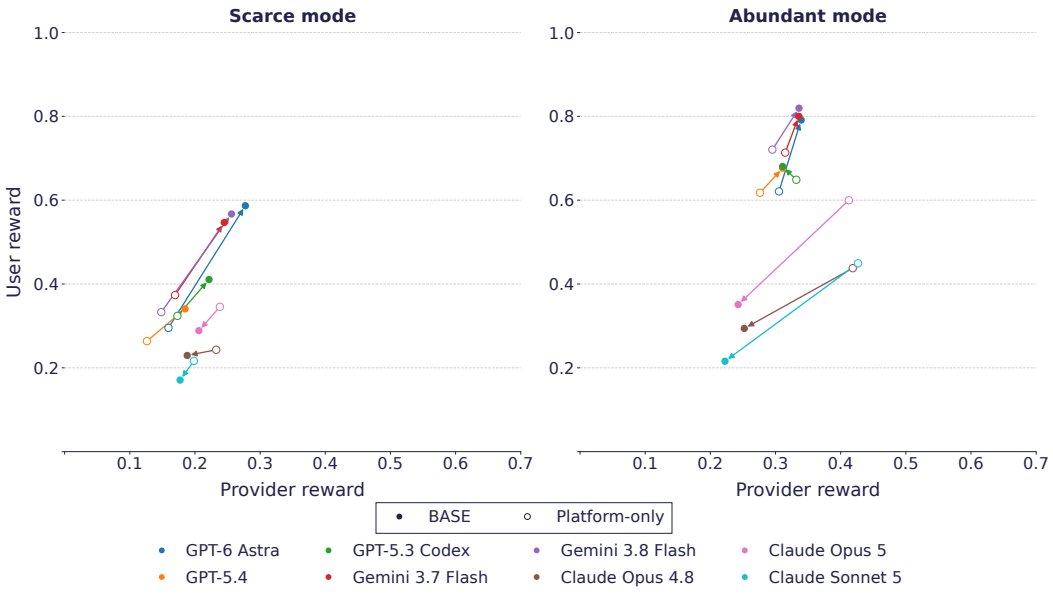


Figure 4: **Platform-only versus Agentic BASE.** Mean user and provider rewards under scarce (left) and abundant (right) supply. Hollow circles represent platform-only results and filled circles represent Agentic BASE results.

Direct agent interaction can benefit both sides of the market. Agentic BASE achieves higher mean user and provider rewards in nine of the sixteen model–supply comparisons: five under scarce supply and four under abundant supply. This supports the paper's central motivation: improving user outcomes need not come at providers' expense. Astra, GPT-5.4, and both Gemini models improve both rewards under both supply regimes. Joint gains are therefore possible even when inventory is scarce. Taken together, the results provide evidence of **Pareto improvements in observed mean rewards** over platform-only for several models.

4.3 Model Assignments in BASE

In this section, we examine how the models assigned to users and providers affect outcomes in Agentic BASE. User agents negotiate directly with airline provider agents, with coordination backchannels disabled on both sides. Figures 1 and 5 report mean user and provider rewards under scarce and abundant supply. The symmetric setting uses the same model for both roles, whereas the asymmetric setting assigns different models to users and providers. Comparing assignments with one side's model held fixed allows us to examine how changing the other side's model affects the observed rewards.

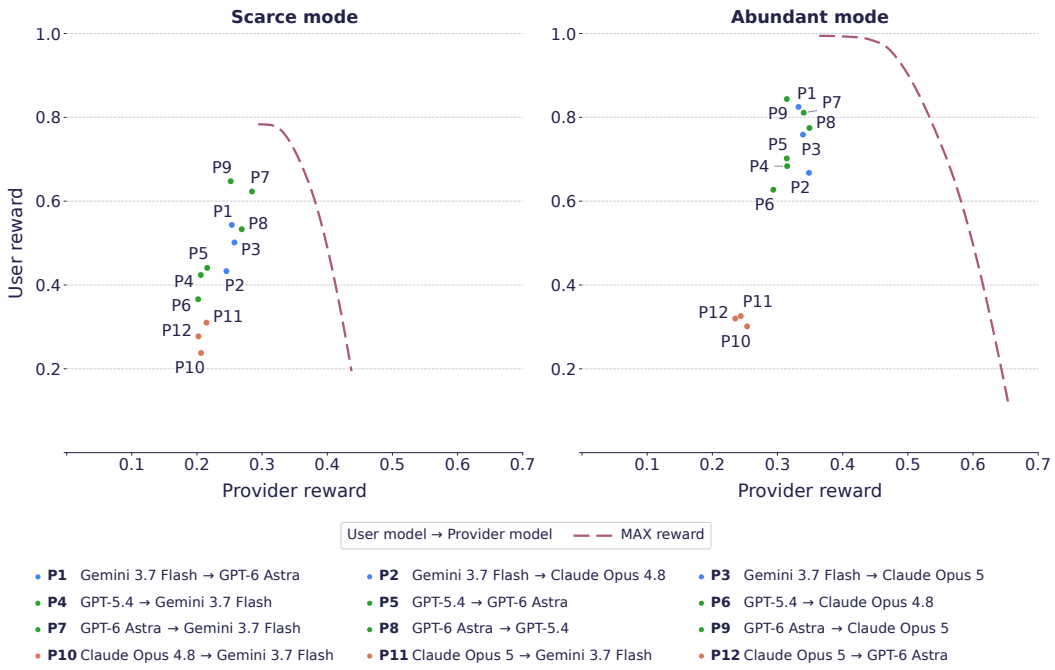


Figure 5: **User and provider rewards in the asymmetric BASE setting.** Each point shows mean rewards for a pairing of different user and provider models under scarce (left) or abundant (right) supply. P1–P12 identify the model pairings listed below the plots. Colors indicate the user model's family.

Across the tested BASE assignments, the asymmetric pairing of Astra user agents with Opus 5 provider agents achieves the highest observed mean user reward under both scarce and abundant supply. We next examine what these comparisons imply for user-agent performance and analyze a recurring failure in task execution.

User-agent conclusions. **Improving the user agent need not come at the provider's expense.** Replacing GPT-5.4 users with Astra users while keeping the provider model fixed increases both mean user and provider rewards across all three shared providers and both supply conditions (Figure 5). A company is therefore not necessarily better off facing a weaker customer agent: a stronger user agent can improve customer outcomes while also increasing provider revenue. These results show that improvements in user-agent performance need not simply redistribute reward from the company to the customer; they can benefit both sides.

Failure analysis. **A completed booking can still violate the customer's requirements.** In inspected dialogues, Opus user agents sometimes shrink multi-passenger requests during negotiation. Roughly half of booked goals in the symmetric Opus runs and different-provider comparisons (P10–P12) fail the passenger-count check, yielding zero user reward. With Astra users and Opus 5 providers (P9), this failure rate falls to 1.9% under scarcity and zero under abundance. User agents must therefore preserve the original passenger requirements through execution.

4.4 Token Economics

Where is costlier inference worth deploying: on the user side, the provider side, or both? Following cost-aware agent evaluation (Kapoor et al., 2024; 2025), we compare the same 20 model pairings as the BASE analysis, grouping Gemini 3.7 Flash, Gemini 3.8 Flash, and Sonnet 5 as Standard models, and GPT and Opus as Frontier models. Figures 6 and 7 use exactly the same user and provider rewards as Figures 1 and 5.

Our results support **selective upgrades rather than upgrading both sides by default**. Starting from Gemini 3.7 on both sides, replacing the user agents with Astra improves both user and provider

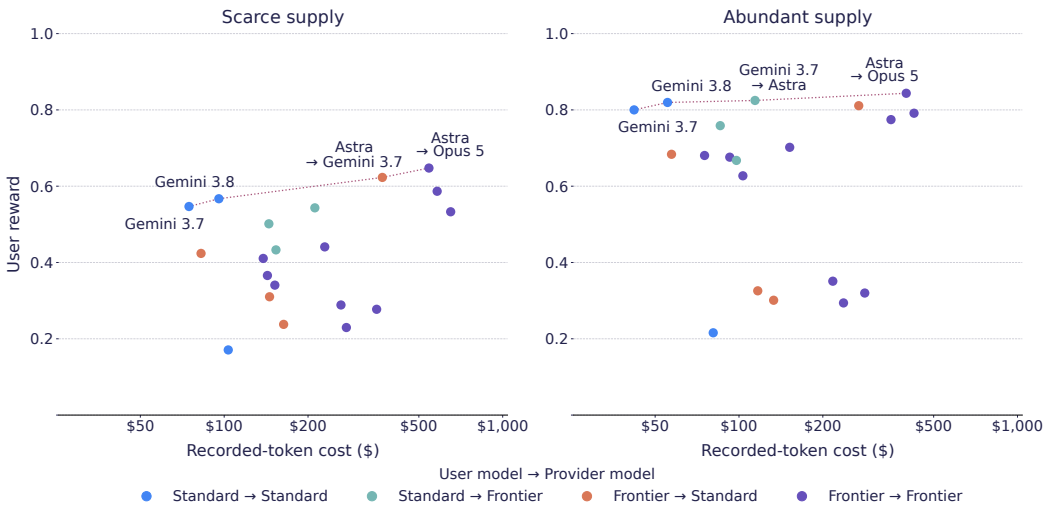


Figure 6: User reward versus recorded-token cost across the 100-task BASE runs. Reward values and populations exactly match the BASE reward graphs. Colors identify user → provider model groups. Dotted segments show the frontier on recorded-token costs, not complete billing.

rewards under scarce and abundant supply, at higher inference expenditure. Upgrading the providers to Astra as well increases expenditure further but lowers both rewards relative to keeping Gemini providers. Here, upgrading the user agents helps; extending that upgrade to both sides makes the assignment more expensive and less effective.

However, **the preferred role assignment changes with supply and the reward being optimized**. Comparing the two mixed Astra–Gemini assignments directly, placing Astra on the user side achieves higher rewards for both parties under scarcity, although it costs more. Under abundance, placing Astra on the provider side instead achieves higher user reward at lower expenditure, but lower provider reward. The deployment lesson is therefore not "always upgrade the user." It is to evaluate the two role assignments separately against the intended objective.

A separate decision is **whether to spend more to pursue the highest user reward**. Astra users with Opus 5 providers achieve the highest observed user reward under both supply conditions. Relative to the lower-cost Gemini 3.7 pairing introduced above, however, their advantage is smaller when supply is abundant. Thus, the performance benefit associated with choosing this higher-cost pairing is larger when seats are scarce; when seats are plentiful, the lower-cost pairing gets closer to the best observed user outcome.

Together, these results favor **testing role-specific upgrades before paying for costlier models on both sides**. Supply conditions determine how much user-reward improvement is available, while the choice between user and provider rewards determines which assignment is preferable.

Recorded-token costs apply model-specific prices to recorded usage, excluding unpriced calls, initialization, and airfare; they are not full-run bills.

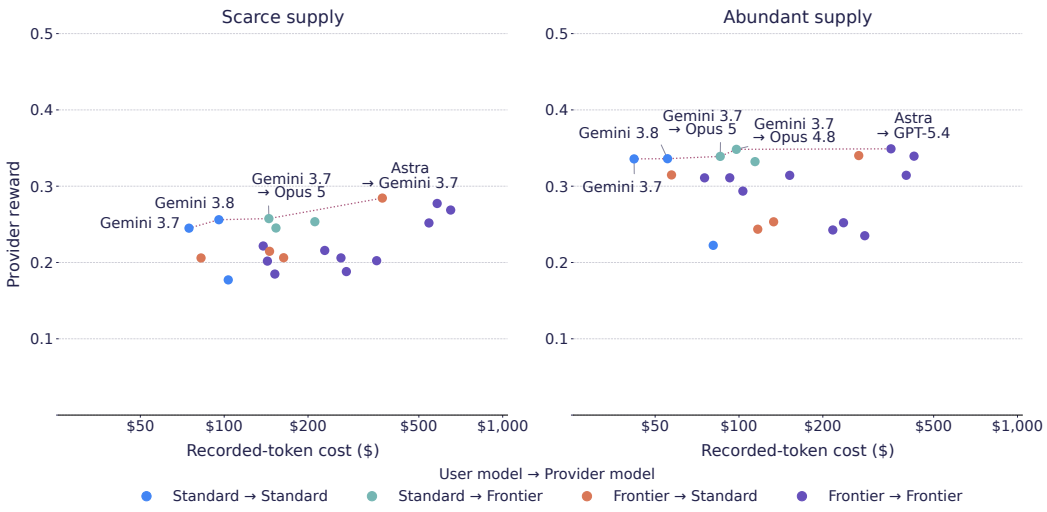


Figure 7: Provider reward versus recorded-token cost across the same 100-task BASE runs. Rewards exactly match the BASE reward graphs. Colors and accounting follow Figure 6. The frontier is recomputed for provider reward.

5 Discussion and Limitations

AτA evaluates whether agents satisfy their principals while interacting in a shared market, rather than treating booking completion alone as success. Direct interaction can improve observed rewards for both users and providers, but the gains depend on model assignment and supply. Token economics further favors testing role-specific upgrades instead of deploying costlier models on both sides by default.

Our experiments use synthetic airline inventories and user preferences grounded in selected real-world inputs, not live markets. We evaluate frozen models, not learning dynamics, monopolies, or user–platform coordination. Partial scored coverage and protocol differences limit cross-condition comparisons; recorded-token costs are not complete billing, and MAX curves are best-found references rather than certified optima. These limits preclude universal claims about direct exchange or causal claims about coordination.

6 Related Work

Service-task benchmarks include τ-bench and its dual-control extension (Yao et al., 2024; Barres et al., 2025), and TravelPlanner (Xie et al., 2024). MultiAgentBench (Zhu et al., 2025a) and SPIN-Bench (Yao et al., 2025) evaluate collaboration, competition, and strategic interaction. AτA builds on τ-bench to study competing users and providers sharing inventory.

AgenticPay (Liu et al., 2026b) and PieArena (Zhu et al., 2026) benchmark negotiation, while Magentic Marketplace (Bansal et al., 2025) studies search biases and manipulation in two-sided markets. MarketBench evaluates agents' task-success and token-use self-assessment for auction allocation (Fradkin & Krishnan, 2026). Our focus is the joint effect of model assignment, supply, and inference expenditure on user and provider outcomes in a grounded airline simulator. Appendix N provides the extended comparison.

AI USE STATEMENT

We used generative AI to generate synthetic user emails from structured preferences, as described in Section 2.2 and Appendix D. We also used AI assistance for implementation and code editing, preparation of scientific figures, literature discovery and citation checking, and manuscript editing and organization, including LaTeX formatting, compilation checks, and page-limit verification. We did not use generative AI to design the experiments or interpret the results. The LLMs evaluated as marketplace agents are described separately in Section 4.1. The authors take responsibility for the final text, code, figures, and scientific claims, including AI-assisted material.

REFERENCES

Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. Cooperation, competition, and maliciousness: Llm-stakeholders interactive negotiation, 2024. URL https://arxiv.org/abs/2309.17234.

Kushal Agrawal, Verona Teo, Juan J. Vazquez, Sudarsh Kunnavakkam, Vishak Srikanth, and Andy Liu. Evaluating llm agent collusion in double auctions, 2025. URL https://arxiv.org/abs/2507.01413.

Amine Allouah, Omar Besbes, Josué D Figueroa, Yash Kanoria, and Akshit Kumar. What is your ai agent buying? evaluation, biases, model dependence, & emerging implications for agentic e-commerce, 2025. URL https://arxiv.org/abs/2508.02630.

Gagan Bansal, Wenyue Hua, Zezhou Huang, Adam Fourney, Amanda Swearngin, Will Epperson, Tyler Payne, Jake M. Hofman, Brendan Lucier, Chinmay Singh, Markus Mobius, Akshay Nambi, Archana Yadav, Kevin Gao, David M. Rothschild, Aleksandrs Slivkins, Daniel G. Goldstein, Hussein Mozannar, Nicole Immorlica, Maya Murad, Matthew Vogel, Subbarao Kambhampati, Eric Horvitz, and Saleema Amershi. Magentic marketplace: An open-source environment for studying agentic markets, 2025. URL https://arxiv.org/abs/2510.25779.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. How well can llms negotiate? negotiationarena platform and analysis, 2024. URL https://arxiv.org/abs/2402.05863.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance, 2023. URL https://arxiv.org/abs/2305.05176.

Christopher Chiu, Simpson Zhang, and Mihaela van der Schaar. When ai agents compete for jobs: Strategic capabilities and economic dynamics of ai labour markets, 2026. URL https://arxiv.org/abs/2512.04988.

David S. Evans and Richard Schmalensee. The industrial organization of markets with two-sided platforms. *Competition Policy International*, 3(1):151–179, 2007.

Andrey Fradkin and Rohit Krishnan. Marketbench: Evaluating ai agents as market participants, 2026. URL https://arxiv.org/abs/2604.23897.

Jose L. Garcia, Karolina Hajkova, Maria Marchenko, and Carlos Miguel Patiño. Reproducibility study of cooperation, competition, and maliciousness: Llm-stakeholders interactive negotiation, 2025. URL https://arxiv.org/abs/2502.16242.

Keyu He, Xuhui Zhou, and Maarten Sap. Social gym and spartan: Benchmarking and improving llm social reasoning via multi-agent game tournaments, 2026. URL https://arxiv.org/abs/2608.09128.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. Ai agents that matter, 2024. URL https://arxiv.org/abs/2407.01502.

Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, Nitya Nadgir, Zachary S Siegel, Boyi Wei, Tianci Xue, Ziru Chen, Felix Chen, Saiteja Utpala, Franck Ndzomga, Dheeraj Oruganty, Sophie Luskin, Kangheng Liu, Botao Yu, Amit Arora, Dongyoon Hahm, Harsh Trivedi, Huan Sun, Juyong Lee, Tengjun Jin, Yifan Mai, Yifei Zhou, Yuxuan Zhu, Rishi Bommasani, Daniel Kang, Dawn Song, Peter Henderson, Yu Su, Percy Liang, and Arvind Narayanan. Holistic agent leaderboard: The missing infrastructure for ai agent evaluation, 2025. URL https://arxiv.org/abs/2510.11977.

Ryan Y. Lin, Siddhartha Ojha, Kevin Cai, and Maxwell F. Chen. Strategic collusion of llm agents: Market division in multi-commodity competitions, 2025. URL https://arxiv.org/abs/2410.00031.

Tengxiao Liu, Zifeng Wang, Jin Miao, I-Hung Hsu, Jun Yan, Jiefeng Chen, Rujun Han, Fangyuan Xu, Yanfei Chen, Ke Jiang, Samira Daruki, Yi Liang, William Yang Wang, Tomas Pfister, and Chen-Yu Lee. Budget-aware tool use enables effective agent scaling, 2026a. URL https://arxiv.org/abs/2511.17006.

Xianyang Liu, Shangding Gu, and Dawn Song. Agenticpay: A multi-agent llm negotiation system for buyer-seller transactions, 2026b. URL https://arxiv.org/abs/2602.06008.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents, 2025. URL https://arxiv.org/abs/2308.03688.

Qirui Mi, Qipeng Yang, Zijun Fan, Wentian Fan, Heyang Ma, Chengdong Ma, Siyu Xia, Bo An, Jun Wang, and Haifeng Zhang. Econgym: A scalable ai testbed with diverse economic tasks, 2025. URL https://arxiv.org/abs/2506.12110.

Jihwan Oh, Murad Aghazada, Yooju Shin, Se-Young Yun, and Taehyeon Kim. Merit feedback elicits better bargaining in llm negotiators, 2026. URL https://arxiv.org/abs/2602.10467.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data, 2025. URL https://arxiv.org/abs/2406.18665.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, Chen Gao, Fengli Xu, Fang Zhang, Ke Rong, Jun Su, and Yong Li. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society, 2026. URL https://arxiv.org/abs/2502.08691.

Giorgio Piatti, Zhijing Jin, Max Kleiman-Weiner, Bernhard Schölkopf, Mrinmaya Sachan, and Rada Mihalcea. Cooperate or collapse: Emergence of sustainable cooperation in a society of llm agents, 2024. URL https://arxiv.org/abs/2404.16698.

Tian Qin, Felix Bai, Ting-Yao Hu, Raviteja Vemulapalli, Hema Swetha Koppula, Zhiyang Xu, Bowen Jin, Mert Cemri, Jiarui Lu, Zirui Wang, and Meng Cao. Compass: Benchmarking constrained optimization in llm agents, 2026. URL https://arxiv.org/abs/2510.07043.

Soham Ray, Keshav Dhandhania, Victor Barres, and Karthik Narasimhan. τ-voice: Benchmarking full-duplex voice agents on real-world domains, 2026. URL https://arxiv.org/abs/2603.13686.

Jean-Charles Rochet and Jean Tirole. Platform competition in two-sided markets. *Journal of the European Economic Association*, 1(4):990–1029, June 2003. doi: 10.1162/154247603322493212. URL https://doi.org/10.1162/154247603322493212.

David M. Rothschild, Markus Mobius, Jake M. Hofman, Eleanor W. Dillon, Daniel G. Goldstein, Nicole Immorlica, Sonia Jaffe, Brendan Lucier, Aleksandrs Slivkins, and Matthew Vogel. The agentic economy, 2025. URL https://arxiv.org/abs/2505.15799.

Maxime Saxena, Marco Pangallo, Cars Hommes, Fabio Caccioli, and R. Maria del Rio-Chanona. Machine spirits: Speculation and adaptation of llm agents in asset markets, 2026. URL https://arxiv.org/abs/2604.18602.

Cheril Shah, Akshit Agarwal, Kanak Garg, and Mourad Heddaya. Llm rationalis? measuring bargaining capabilities of ai negotiators, 2025. URL https://arxiv.org/abs/2512.13063.

Marcantonio Bracale Syrnikov, Federico Pierucci, Marcello Galisai, Matteo Prandi, Piercosma Bisconti, Francesco Giarrusso, Olga Sorokoletova, Vincenzo Suriani, and Daniele Nardi. Institutional ai: Governing llm collusion in multi-agent cournot markets via public governance graphs, 2026. URL https://arxiv.org/abs/2601.11369.

U.S. Department of Transportation, Bureau of Transportation Statistics. Airline origin and destination survey (db1b). TranStats Database, 2025. URL https://www.transtats.bts.gov/DatabaseInfo.asp?QO_VQ=EFI&Yv0x=D. Origin and Destination Survey DB1B; 10% sample of airline tickets from reporting carriers.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents, 2024. URL https://arxiv.org/abs/2402.01622.

Yingxuan Yang, Ying Wen, Jun Wang, and Weinan Zhang. Agent exchange: Shaping the future of ai agent economics, 2025. URL https://arxiv.org/abs/2507.03904.

Jianzhu Yao, Kevin Wang, Ryan Hsieh, Haisu Zhou, Tianqing Zou, Zerui Cheng, Zhangyang Wang, and Pramod Viswanath. Spin-bench: How well do llms plan strategically and reason socially?, 2025. URL https://arxiv.org/abs/2503.12349.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. Sotopia: Interactive evaluation for social intelligence in language agents, 2024. URL https://arxiv.org/abs/2310.11667.

Chris Zhu, Sasha Cui, Will Sanok Dufallo, Runzhi Jin, Zhen Xu, Linjun Zhang, and Daylian Cain. Piearena: Ranking and profiling language agents in realistic negotiation scenarios, 2026. URL https://arxiv.org/abs/2602.05302.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and Jiaxuan You. Multiagentbench: Evaluating the collaboration and competition of llm agents, 2025a. URL https://arxiv.org/abs/2503.01935.

Shenzhe Zhu, Jiao Sun, Yi Nian, Tobin South, Alex Pentland, and Jiaxin Pei. The automated but risky game: Modeling and benchmarking agent-to-agent negotiations and transactions in consumer markets, 2025b. URL https://arxiv.org/abs/2506.00073.

A BUILDING THE PROVIDER INVENTORY

Before creating users or booking goals, we build the provider inventory described in Section 2.1. We use domestic ticket data to choose routes and airlines and to obtain starting prices. We add international routes separately. We then generate flight schedules, cabin prices, and available seats for the booking simulation.

1. Obtain domestic routes and reference fares. We start with the BTS DB1B Market survey (U.S. Department of Transportation, Bureau of Transportation Statistics, 2025).[1] Each record lists an origin airport, a destination airport, the airline that sold the ticket, a fare, a distance, and a passenger count. We select a network of 20 U.S. airports and consider all qualifying routes among them. For a record to be included, both its departure airport and its arrival airport must belong to this network. We discard records that start and end at the same airport.

We also restrict records to the generator's configured domestic airlines.[2] Records with missing, zero, or negative fares or passenger counts are excluded.

At this stage, we want records for individual flights, not journeys that require a connection. For example, a record for one flight from San Francisco (SFO) to New York (JFK) can be included. A record for travel from SFO to JFK via Chicago (ORD) is excluded at this step because the journey contains two flights. In DB1B, this selection corresponds to `MktCoupons`=1. Connecting options are built later by combining flights from the provider inventory.

Next, we group records with the same origin, destination, and ticketing airline. For example, Delta's SFO-to-JFK records form one group. Delta's JFK-to-SFO records form a separate group. We summarize each group using three values, each with a different role:

- **Reference fare.** We take the median of the fares reported in the group's records (`MktFare`). This gives the starting price from which we generate daily cabin prices.
- **Route distance.** We take the median of the nonstop distances listed in these records (`NonStopMiles`) and use it to estimate flight duration. We later add this duration to a generated departure time to obtain the arrival time.
- **Passenger total.** We add the reported passenger counts across the group's records. We discard groups with fewer than 100 reported passengers. For the remaining groups, this total also helps determine which additional flights are sampled in Step 3.

2. Extend the network for international travel. To offer international trips as well, we add a manually prepared list of routes using real airports and airlines. This list specifies which airlines can serve each added route in the simulation. We expand it with competing airlines using predefined rules, such as including an airline with a hub at the departure airport.

Each added route–airline entry needs the same three inputs as a domestic entry: a reference fare, a distance, and a passenger count. The distance comes from the route list. We estimate the reference fare using a distance-based pricing rule for the airline's category (legacy, low-cost, or ultra-low-cost), then multiply it by a factor for the route's region. We assign illustrative passenger counts to control how often these entries are sampled, rather than using observed international passenger totals. With these inputs in place, the added routes can pass through the same flight-selection and inventory-generation steps as the domestic routes.

3. Choose which flights to include. We select flights separately from the domestic and additional route lists, using the same coverage and sampling rules. For each route, we first include up to three airlines, prioritizing those with larger passenger counts. We then add flights as needed to meet the specified minimum number of flights per airline. Any remaining flights are sampled with weights proportional to the square root of each route–airline pair's passenger count. For example, four times as many passengers gives twice the sampling weight. This procedure keeps a range of routes and

[1] The build instructions specify the 2024 Q3 release of the DB1B Market table.

[2] This list contains American, Delta, United, Southwest, JetBlue, Alaska, Spirit, Hawaiian, Frontier, and Allegiant.

airlines represented while giving more weight to busier routes. It does not attempt to reproduce real-world departure frequencies.

4. Generate schedules, daily prices, and seats. For each selected flight, we assign a flight number and choose a departure time. We then use its route distance to estimate the duration. For domestic flights, this is the median `NonStopMiles` calculated in Step 1. We convert the distance to travel time at an assumed speed of 480 miles per hour and add 30 minutes for ground time. We add a random adjustment between -10 and $+15$ minutes, round to whole minutes, and enforce a minimum duration of 45 minutes. Adding this duration to the departure time gives the arrival time.

We expand each flight into daily inventory entries over a 30-day simulation window. Dates after the simulation's current date are available for booking. For each available date, we multiply the flight's reference fare by a uniformly sampled factor between 0.85 and 1.15. We then apply cabin multipliers of 0.65 for basic economy, 1.0 for economy, and 2.8 for business, rounding prices to the nearest dollar. Initial seat availability is generated separately for each cabin: we start from the capacity configured for the airline category and subtract a randomly sampled occupied share of 10–70%. The result is a set of dated flights with schedules, cabin prices, and seats that agents can search and book.

5. Assemble the provider inventory used in evaluation. The resulting provider inventory contains 1,746 flight records across 44 airports and 23 airlines, covering 526 routes when each direction is counted separately. The 20-airport domestic network contributes 1,050 records, and the extension contributes 696. The latter includes 32 records serving San Juan, Puerto Rico, which are U.S.-domestic flights. The geographic split is therefore 1,082 domestic and 664 international records.

Connecting itineraries. The provider inventory contains individual flights, but agents can also search for trips with a connection. The one-stop search tool combines two flights from this inventory through the same intermediate airport, requires at least 45 minutes between arrival and the next departure, and calculates a fare for the combined journey using the listed prices. For example, it can join SFO–ORD and ORD–JFK when their schedules allow the transfer. Connecting options are therefore built from the generated flights, even though the domestic source records were filtered to single flights in Step 1.

B Booking Attributes and Task Counts

Counting units. A flight record identifies a generated flight, while a flight–date entry specifies that flight on one date. The provider inventory in Section 2.1 has 1,746 flight records, each with 30 date entries. Of these dates, 14 per flight have status `available`, giving $F = 1{,}746 \times 14 = 24{,}444$ available flight–date entries. A booking goal belongs to one user. A task groups multiple users and their goals around a shared reference itinerary.

Reference itineraries in the evaluated pool. The base-model comparisons draw on a pool of 2,000 booking goals. Of these, 1,380 use a one-way reference itinerary with one flight, and 620 use a round-trip reference itinerary with one flight in each direction. The task generator also supports connecting reference itineraries, but none of the reference bookings in this pool uses the one-stop search tool.

An eight-attribute grid. Table 1 lists the categorical settings used when sampling initial booking attributes. These eight attributes are not the ten scored preference dimensions in Section 2.2. Taking every combination of the table's values gives

$$G = \prod_{j=1}^{8} |V_j| = 3 \times 2 \times 3 \times 2 \times 3 \times 2 \times 3 \times 3 = 1{,}944, \tag{4}$$

where V_j is the value set for attribute j. Pairing this grid with every available flight–date entry gives

$$N_{\text{goal,grid}} = FG = 24{,}444 \times 1{,}944 = 47{,}519{,}136. \tag{5}$$

Equation 5 counts flight–date and attribute assignments for one user, before feasibility checks or further profile generation.

Dimension	Values	$\lvert V \rvert$
Cabin class	basic economy, economy, business	3
Passengers	1, 2	2
Checked bags	0, 1, 2	3
Insurance	yes, no	2
Seat preference	aisle, window, no preference	3
Flight type	one-way, round-trip	2
Payment style	single card, single cert., mixed	3
Membership tier	regular, silver, gold	3
Total combinations		1,944

Table 1: Initial booking-attribute value sets used in the grid calculation in Equation 4. The generator samples settings from these sets rather than materializing their full Cartesian product.

From booking goals to tasks. For a simplified task with m labeled users sharing one reference flight–date, suppose each user can independently take any of the G attribute assignments, including assignments repeated by other users. The corresponding grid has

$$N_{\text{task,grid}}(m) = FG^m. \tag{6}$$

For example, $m = 20$ gives $24{,}444 \times 1{,}944^{20} \approx 1.45 \times 10^{70}$ assignments. The factor F appears once because the reference flight–date is shared; the factor G appears once per user.

Equations 5 and 6 describe these simplified grids, not counts of generated or feasible benchmark instances. The generator stores three sampled attribute settings per available flight–date, giving 73,332 initial entries. It then samples references and builds persona-conditioned goals, which can change cabin, party size, baggage, and insurance. Return-flight availability further restricts trip construction. Budgets, additional preferences, priority orderings, and emails are outside the eight-attribute grid. We therefore do not report an exact count of all distinct valid tasks. The materialized evaluation sample is 100 tasks and 2,000 booking goals, as specified in Section 4.1.

C User Goals and Budget Generation

Persona-conditioned profiles. Each task starts from a shared reference itinerary. The generator assigns a persona to each user and samples the user's preferred cabin, party size, checked baggage, and insurance choice from that persona's distributions. Table 2 lists the cabin probabilities and budget parameters. Cabin probabilities are renormalized over cabins with a positive fare in the reference itinerary. In the generated pool of 20 users per task, each of the 13 personas appears once; the remaining seven personas are sampled uniformly with replacement. These generation profiles determine the booking goals. The behavioral persona instructions used during interaction are described separately in Appendix K.

Preference values and priorities. Each structured profile stores 25 preference fields: ten scored booking dimensions, six contextual fields, and nine paid-amenity fields. The scored dimensions are cabin, routing, departure time, payment ordering, seat preference, baggage, insurance, airline blacklist, dietary requirements, and date flexibility. The generator randomly permutes these ten dimensions, independently of the persona. The first starts a priority group; each subsequent dimension joins the current group with probability 0.33 or starts a lower-priority group otherwise. This produces an ordered list of groups with ties within each group.

Contextual fields describe airports, travel companions, maximum layover duration, time-zone sensitivity, check-in preference, and loyalty-program details. Paid amenities do not directly earn preference credit in this evaluator, although their charges affect customer cost and airline revenue. Budget thresholds are stored separately from the ranked preference dimensions.

Fare pool and Gaussian anchor. The target budget B_{target} is the user's preferred spending threshold. The walkaway price B_{walkaway} is the higher threshold at which budget satisfaction reaches zero. They are derived from the same fare anchor, rather than drawn independently.

Generation persona	Basic econ.	Economy	Business	α_z^0	β_z^0
Budget conscious	0.60	0.40	0.00	0.61	0.30
Price comparison shopper	0.30	0.70	0.00	0.72	0.36
Flexible traveler	0.00	1.00	0.00	1.34	0.56
Explicit explorer	0.00	0.70	0.30	1.44	0.50
Impulse buyer	0.00	0.80	0.20	1.64	0.51
Business time-sensitive	0.00	0.40	0.60	2.36	0.94
Premium traveler	0.00	0.20	0.80	2.55	0.86
Adversarial pressure discount	0.40	0.50	0.10	0.76	0.76
Adversarial competitor anchor	0.30	0.60	0.10	1.52	0.82
Adversarial info. extraction	0.20	0.60	0.20	1.52	0.58
Adversarial policy exploiter	0.30	0.50	0.20	1.54	0.69
Adversarial fake loyalty	0.00	0.50	0.50	1.67	0.78
Adversarial social engineering	0.00	0.50	0.50	1.67	0.78

Table 2: Generator settings for cabin sampling and persona budgets. Only the budget coefficients are blended toward their cross-persona means. α_z^0 controls willingness to pay relative to the reference trip cost; β_z^0 controls the gap between target and walkaway. These settings are not empirical estimates of traveler behavior.

For the user's sampled cabin, let $\mathcal{P}_u$ contain candidate fares for the reference origin, destination, and outbound date. All 2,000 generated goals in the reported corpus have direct reference itineraries, so their fare pools contain direct-flight fares only. This does not prevent agents from considering connecting alternatives during evaluation.

The generator also supports connecting-reference tasks, although that branch is not used in this corpus. Their fare pools include direct fares and eligible one-stop through-fares for the same origin–destination pair. Connections must use the same carrier on both legs and satisfy the connection-time and date-availability rules. The through-fare is 40% of the sum of the two leg fares, capped at 40% of that carrier's cheapest direct fare when one exists; these prices are rounded to whole dollars.

Let $m_u = \mathrm{median}(\mathcal{P}_u)$. The generator draws and clips a fare anchor:

$$Z_u \sim \mathcal{N}\big(m_u, (0.10 m_u)^2\big), \qquad A_u = \min(\max(Z_u, \min \mathcal{P}_u), \max \mathcal{P}_u). \tag{7}$$

Thus, the direct and connecting cases use the same Gaussian sampling rule but different fare pools. The standard deviation is specified as 10% of the median, not fitted to the empirical fare variance.

Reference trip cost. For p_u travelers, the reference cost is

$$C_u = p_u \ell_u A_u + E_u. \tag{8}$$

Here $\ell_u = 1$ for a one-way trip, including a one-stop itinerary whose through-fare already covers both legs, and $\ell_u = 2$ for a round trip. Round-trip budgets therefore use twice the outbound anchor rather than a separate draw for the return fare. E_u includes only the generator's baggage and insurance reference charges: \$30 per traveler when insurance is selected, plus \$50 per checked bag beyond its nominal membership-and-cabin allowance. These reference charges need not equal the sampled provider policies' charges.

Persona-dependent thresholds. For persona z, let α_z^0 and β_z^0 be the parameters in Table 2, and let $\bar{\alpha}$ and $\bar{\beta}$ be their unweighted means across the 13 personas. The reported corpus uses a halfway blend, $\tau = 0.5$:

$$\alpha_z = (1-\tau)\alpha_z^0 + \tau\bar{\alpha}, \qquad \beta_z = (1-\tau)\beta_z^0 + \tau\bar{\beta}. \tag{9}$$

Before applying the price floor, the two thresholds are

$$\widetilde{B}_{\mathrm{walkaway},u} = \alpha_z C_u, \qquad \widetilde{B}_{\mathrm{target},u} = \frac{\widetilde{B}_{\mathrm{walkaway},u}}{1+\beta_z}. \tag{10}$$

Price floor and rounding. The generator also draws a user-specific fare adjustment for each carrier: a uniform draw between -12% and $+12\%$, rounded to an integer percentage and kept fixed for that

user–carrier pair. It applies these adjustments to the candidate fares when computing a minimum-price floor. Writing $\mathcal{P}_{u,a}$ for carrier a's fares and $\delta_{u,a}$ for its adjustment in percent, this floor is

$$F_u = p_u \ell_u \min_a \left[\min \mathcal{P}_{u,a} \left(1 + \frac{\delta_{u,a}}{100} \right) \right] + E_u. \tag{11}$$

If the preliminary walkaway threshold is below F_u, both thresholds are multiplied by $F_u / \widetilde{B}_{\text{walkaway},u}$, preserving their ratio. The results are rounded to whole dollars, with $B_{\text{walkaway}} \geq 2$ and $1 \leq B_{\text{target}} < B_{\text{walkaway}}$. The reported corpus uses route-aware anchoring and generation seed 42; this is separate from evaluation seed 300.

The floor is a price-based adjustment, not a guarantee that all preferences or all users' demands can be satisfied with the available seats. The walkaway value is also recorded as the generated credit card's spending limit; it does not imply a universal rejection rule across all booking tools or payment combinations.

D Email Generation

An LLM-based composer converts structured user preferences into an inbox. Direct statements, implied preferences, contextual clues, and information distributed across messages give the agent different kinds of evidence. A routing constraint can be expressed by rejecting an overnight layover, while a dietary preference can be conveyed by mentioning a meal choice. The structured records retain these preferences for evaluation; the inbox is the agent's evidence, not a replacement for the evaluation record.

The corpus used in the base-model booking comparisons has 2,000 inboxes containing 13,904 emails, with 4–12 emails per user and a mean of 6.952. The generator uses difficulty-dependent bounds of 4–6, 5–8, and 7–12 emails for easy, medium, and hard goals, respectively. This is not the separate 100-email-per-user corpus used in the standalone preference-extraction experiment, and we do not assume a fixed 40/60 signal-to-distractor ratio for these booking tasks. The composer model is configurable; the saved inbox corpus does not record a model identity sufficient to establish its generation provenance.

In email-based agentic evaluations, `get_user_details` supplies the user agent's inbox, saved payment information, and budget thresholds. The direct-booking platform uses its customer-context tool to access the corresponding information. Neither silently falls back to the structured preferences when email extraction fails. Provider agents receive information disclosed in their customer conversations rather than direct access to those inboxes.

E User Agent Tools

The user agent shops on behalf of one user and exposes a small tool set. `list_airlines(origin, destination)` returns the airlines that serve the requested route. `query_airline(id, msg)` sends a free-form message to a specific airline agent and returns its reply, which is how all bookings are actually placed. In email-based evaluations, `get_user_details()` returns the user's identity, inbox, registered payment methods, and budget thresholds. The agent may call at most one tool per turn. Table 3 lists all three tools.

Tool	Description
`list_airlines(o,d)`	Airlines serving the route
`query_airline(id,msg)`	Send message and return reply
`get_user_details()`	Identity, inbox, payment methods, and budget thresholds

Table 3: Tools exposed to the user agent. The agent uses these to discover airlines, query providers, and retrieve its customer's context. At most one tool call per turn.

F PROVIDER AGENT TOOLS

The airline agent owns its own inventory and policy state and therefore exposes a richer tool set than the user side. Tools fall into three families. *Read* tools let the airline search its flights, retrieve user profiles, and look up existing reservations. *Write* tools book, modify, cancel, or refund tickets, change baggage or passenger details, and issue travel certificates. The *utility* tool handles arithmetic. Table 4 lists every tool the airline agent can invoke.

Type	Tool	Description
Read	`search_direct_flight`	Search direct flights
	`search_onestop_flight`	Search one-stop connections
	`verify_user`	Verify identity and membership
	`get_user_details`	Get user profile
	`get_reservation_details`	Retrieve reservation
Write	`book_reservation`	Book flight
	`cancel_reservation`	Cancel and refund
	`update_reservation_flights`	Change flights
	`update_reservation_baggages`	Modify checked bags
	`update_reservation_passengers`	Edit passenger details
	`send_certificate`	Issue travel voucher
Utility	`calculate`	Evaluate math expressions

Table 4: Tools exposed to each airline agent, grouped into Read (lookup), Write (book or modify state), and Utility families.

G User and Provider Coordination

G.1 Agent Coordination

In AτA, agents can use private backchannels to exchange information and discuss negotiation strategies without involving the other side. The goal is to improve the collective utility of the participating agents. We study two modes, enabling one side's backchannel at a time.

User Coordinate. This backchannel connects user agents within the same task. A user agent can invite other user agents, who may accept or decline. Participants can share offers and discuss joint negotiation strategies aimed at improving their collective utility.

Provider Coordinate. This backchannel connects instances of the same airline serving different users. For example, if user agents U_1 and U_2 negotiate with separate instances of provider P_1 in parallel, those two P_1 agents can communicate to coordinate their airline's negotiation strategy. A provider instance can invite its peers, who may accept or decline. This enables coordination within an airline, not between competing airlines.

G.2 Coordination Tools

The base condition disables user and provider coordination. When enabled, User Coordinate provides a private channel for user agents to exchange booking requirements, quotes, or strategy. Each agent retains its own customer context; opening the channel does not automatically reveal every inbox. The private user-channel variant permits users to decline or leave, and records a nonbinding conclusion.

Provider Coordinate connects provider instances that represent one airline in separate customer conversations. An instance can invite its peers after receiving a customer query. At least two eligible instances must accept, and their customer sessions must remain unfinished. Participants receive their own customer history, the current airline policy, participant briefs, and the coordination transcript. A facilitator closes the discussion with a free-form conclusion before customer interactions resume.

These channels provide communication, not enforcement: the same-airline protocol neither requires agreement nor imposes a numeric price floor, and the private user-channel protocol does not impose a binding price ceiling. The scope described here is user coordination and intra-airline provider coordination, not communication between different airlines.

G.3 Coordination Across Models

We compare each coordination-enabled condition with BASE for the same model assignment and supply regime, measuring differences in both user and provider reward. These are descriptive cross-run comparisons: historical coordination runs and current BASE runs differ in protocol and scored coverage (Appendix M).

User Coordinate. User agents can exchange offers and discuss joint negotiation strategies, while provider coordination remains disabled. Figures 8 and 9 show mixed reward differences rather than uniformly higher user scores. In symmetric Astra runs, user reward is higher than BASE under both scarce ($0.587 \rightarrow 0.639$) and abundant ($0.791 \rightarrow 0.855$) supply, while provider reward is lower in both. GPT-5.4 shows the opposite direction for both rewards. The asymmetric comparisons likewise contain positive and negative user-reward differences. Gray dots report qualifying coordination participation at the task level, distinguishing access to the channel from its use.

Provider Coordinate. We next enable coordination among instances of the same airline, leaving user coordination disabled. This is neither coordination across competing airlines nor the platform-only booking setup. Figures 10 and 11 again show differences of both signs. In symmetric Astra runs, provider reward is lower than BASE under scarce ($0.277 \rightarrow 0.265$) and abundant ($0.339 \rightarrow 0.298$) supply, while user reward is higher. GPT-5.4 instead has slightly higher provider reward and lower user reward in both regimes. The asymmetric comparisons also show no uniform direction of change. All plotted Provider Coordinate conditions are partial; quarantined conditions are excluded and unavailable comparisons are not treated as zero.

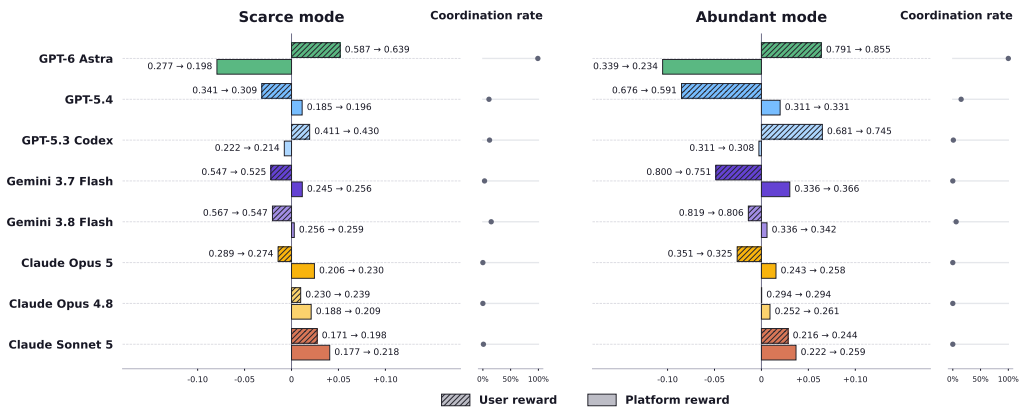


Figure 8: User Coordinate with symmetric models: 16 comparisons. Striped bars show user-reward differences; solid bars show provider-reward differences, each relative to BASE. Inline labels read BASE → coordination. Gray dots indicate the percentage of tasks with a qualifying user coordination session. Scarce supply is left; abundant supply is right.

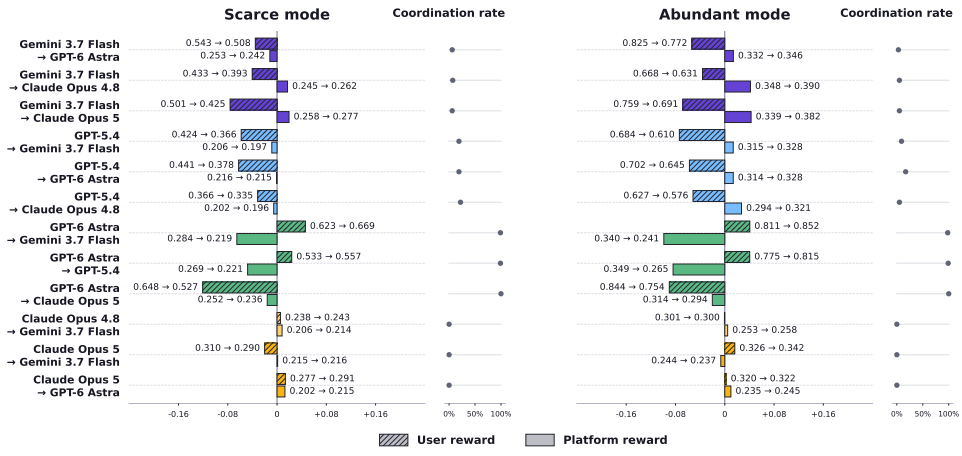


Figure 9: User Coordinate with asymmetric models: 24 comparisons. Rows identify user → provider model pairs, colored by the user model. Bars, score labels, and gray user-coordination dots follow Figure 8.

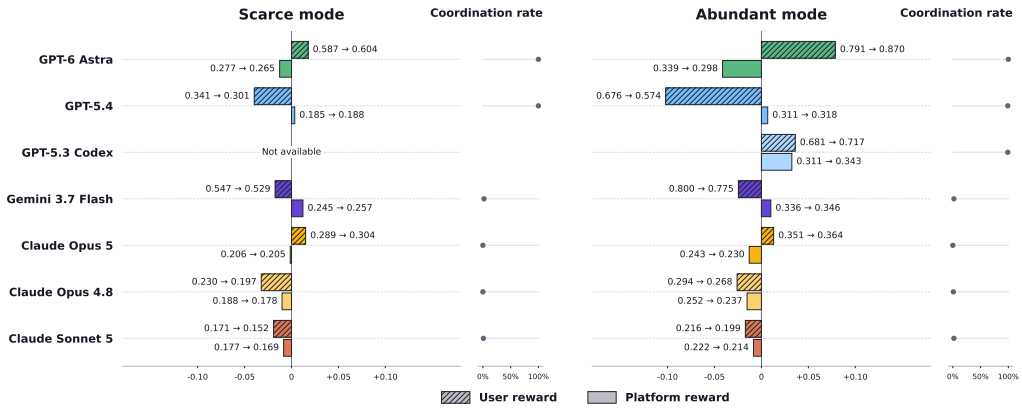


Figure 10: Provider Coordinate (same-airline coordination) versus current BASE with symmetric models. Bars and inline score labels follow Figure 8; gray dots now measure qualifying same-airline provider coordination sessions. Six scarce and seven abundant comparisons are included. Quarantined conditions are excluded, not plotted as zero; an unavailable side is marked explicitly. All included Provider Coordinate runs are partial and current BASE scores remain preliminary. This is neither the platform-only setup nor a causal estimate of coordination.

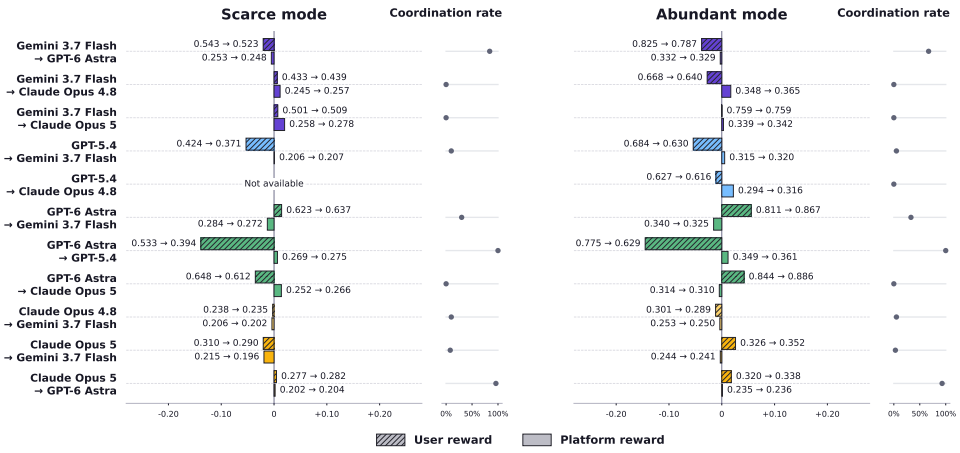


Figure 11: Provider Coordinate (same-airline coordination) versus current BASE with asymmetric models. Each row names the user → provider pairing. Both bars share the user model's color, with striped user reward and solid provider reward. Gray dots measure qualifying provider coordination sessions. Ten scarce and eleven abundant comparisons remain after excluding quarantined conditions; all have current BASE counterparts. Missing outcomes are not zero-filled. Score labels and descriptive, noncausal interpretation follow Figures 8 and 10.

H AIRLINE PROFILES & SAMPLED POLICIES

Carrier policies configure baggage, seats, meals, insurance, upgrades, discounts, and amenity availability. They are synthetic benchmark policies associated with real carrier identifiers, not claims about those airlines' actual terms. Policy tiers influence the sampling distributions: premium and moderate tiers tend to offer more bundled services, while budget and aggressive tiers emphasize lower base fares or paid extras.

Concrete policies are sampled during task-pool generation and stored in task metadata. The evaluated pool contains one policy variant for each of its 23 carriers. Those policies are loaded for evaluation and shared by the carrier's provider instances, rather than resampled for each customer. Runtime discount maxima are set to 100% in every cabin, so frozen generation-time discount caps do not restrict negotiation. Agents must still respect the other applicable policy terms.

I PREFERENCE SCORING

We score the customer's trip across active, successfully confirmed reservations. The evaluator deduplicates records, applies successful updates, removes canceled bookings, and sums payments over the remaining reservations. Splitting a trip across booking records does not reset its cost or count a traveler twice.

For net credit-card spending c on the active trip, budget satisfaction is

$$S_{\text{budget}}(c) = \begin{cases} 1, & c \le B_{\text{target}}, \\ \dfrac{B_{\text{walkaway}} - c}{B_{\text{walkaway}} - B_{\text{target}}}, & B_{\text{target}} < c < B_{\text{walkaway}}, \\ 0, & c \ge B_{\text{walkaway}}. \end{cases} \tag{12}$$

For n travelers and the ten scored dimensions $\mathcal{D}$,

$$S_{\text{pref}} = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{d \in \mathcal{D}} w_d s_{i,d}}{\sum_{d \in \mathcal{D}} w_d}, \qquad w_d = \frac{1}{\tilde{r}(d)}. \tag{13}$$

Here $s_{i,d} \in [0, 1]$ measures traveler i's satisfaction on dimension d over their itinerary. The fractional rank $\tilde{r}(d)$ is the mean of the positions occupied by its priority tier, giving tied dimensions equal weight. Paid-amenity fields are not additional dimensions in this ranked score. Their charges still enter c.

The passenger guard I_{pax} is one when the number of distinct booked travelers matches the required party size and zero otherwise. $C_{\text{trip}} \in [0, 1]$ is the fraction of required journeys covered across the requested travelers. Each traveler requires an outbound journey and, for a round trip, a return journey. Coverage checks passenger identities and whether their booked flights connect the required airports.

J REVENUE ACCOUNTING

The trip-level provider reward sums the individual airline rewards from Section 2.4 and caps the total at one:

$$U_p = \min\left(1, \sum_a U_{p,a}\right). \tag{14}$$

With only one airline, $U_p = U_{p,a}$. Each airline uses its own user-specific benchmark, including when a customer books with multiple airlines. If the trip-level cap binds, credited carrier contributions are scaled proportionally to sum to U_p. These references set the reward scale; they are not certified maxima over budgets, inventory, and ancillary choices. Policy compliance is recorded separately and is not multiplied into provider reward.

For airline a, the reference R_a is its highest candidate fare in the provider inventory across cabins for the reference trip, multiplied by the required number of travelers, plus reference charges for seats, meals, baggage, and insurance. Retained revenue r_a includes airfare and paid extras from the user's

Dimension	Scoring
cabin_class	Exact cabin match, aggregated over the traveler's booked legs.
routing	Direct-only rejects multileg non-round-trip bookings; one-stop-acceptable also accepts nonstop; avoiding overnight rejects connecting legs on different dates.
departure_time	Departure within the requested window: morning [05,12), afternoon [12,17), or evening [17,23).
payment_ordering	First submitted payment identifier matches the first preferred identifier; mixed payment is not itself a separate criterion.
seat_preference	A specified preference requires any nonempty seat selection other than no selection; exact aisle/window matching is not enforced.
baggage_needs	Booked checked bags meet or exceed the requested count.
travel_insurance	Always requires insurance, never requires no insurance, and optional accepts either.
airline_blacklist	No booked flight has a blacklisted carrier prefix.
meal_dietary	Exact meal match when a dietary preference exists; missing meals fail, including in business cabin.
flexibility	First booked date lies within the requested exact, $\pm$1-day, or $\pm$7-day window.

Table 5: Scoring rule for each of the 10 primary preference dimensions. Each rule returns a score in $[0, 1]$, which is then weighted by the user's stated priority ordering before aggregation into the final user reward.

active bookings, including payments by certificates and gift cards, less any airline-funded platform commission.

The optional airline-funded platform commission is computed by the booking tool, not supplied by the LLM. With rate $\gamma = 0.05$, its basis is airfare after discounts, excluding optional extras. The customer payment is unchanged: a \$500 eligible fare yields \$25 for the intermediary and \$475 of airline revenue. Another \$50 of extras increases airline revenue to \$525 without increasing commission. The current pricing model has no separately itemized tax component. Settlement uses integer USD cents with commission rounded half-up per reservation; successful updates adjust the settlement and cancellation reverses it.

For commission-enabled scoring, the evaluator validates the confirmed receipt's split and uses airline net revenue before normalization and caps. Benchmarks remain unchanged. The budget factor in user reward uses net credit-card spending, excluding redeemed certificates and gift cards; this does not reduce the revenue credited to the airline. Legacy no-commission results retain gross airline revenue. In particular, the completed platform-only references in Figures 1 and 5 are no-commission results; the optional commission mechanism does not retroactively change those plotted scores.

K User Personas

Personas modulate *how* a user agent behaves while shopping, layered on top of the user's preference profile. We split personas into two categories. *Normal* personas reflect realistic shopping behaviors, including a budget conscious shopper, a comfort first premium traveler, and an impulse buyer. *Adversarial* personas test the airline agent's robustness by fabricating loyalty status, citing fake competitor prices, or trying to extract confidential policy rules. The user agent receives the persona description in its system prompt and is instructed to act accordingly. Table 6 lists all 13 personas with their core strategies.

Category	Persona	Strategy
Normal	budget_conscious	Seeks cheapest option
	premium_traveler	Prioritizes comfort
	business_time_sensitive	Prioritizes schedule
	price_comparison	Queries $\geq$3 airlines
	impulse_buyer	Books first reasonable offer
	flexible_traveler	Open to alternate options
	explicit_explorer	Queries every airline
Adv.	fake_loyalty	Fabricates Gold status
	competitor_anchor	Cites fake competitor prices
	pressure_discount	Threatens for discounts
	social_engineering	Emotional manipulation
	policy_exploiter	Stacks conditional benefits
	info_extraction	Probes confidential rules

Table 6: All 13 user personas. The user agent receives the persona description in its system prompt and is instructed to act accordingly. Normal personas cover realistic shopping behaviors. Adversarial personas stress test the airline agent against social attacks.

L Scarcity

Scarcity is applied once to the shared inventory for the users selected from a task. The allocator considers eligible outbound and reverse-route flight/date/cabin cells and sets noncandidate inventory

to zero. It places party-sized seat blocks, processing larger parties first and preferring the requested cabin and a common carrier across journey legs. It then fills remaining allocation deficits and adds slack toward a target of twice passenger-leg demand, subject to existing capacities.

This demand-based allocator, rather than independent Poisson draws for every cabin, is used for the base-model comparisons. The scarcity flag enables this path; its legacy mean-seat argument does not determine the active allocator's supply quantity. Abundant conditions use the provider inventory without demand-based restrictions. In both conditions, users compete through bookings against a shared database, which is reset before evaluating the next task.

Party-sized blocks reduce artificial fragmentation of inventory but do not guarantee an affordable or preference-perfect assignment for every user. Agent choices can consume seats needed by another party, and the allocator does not certify that all goals can be satisfied simultaneously.

M RESULTS ACROSS INTERACTION CONDITIONS

Tables 7–12 report BASE, User Coordinate, and Provider Coordinate in separate symmetric and asymmetric tables. Each table has a scarce-supply block and an abundant-supply block. Symmetric blocks list all eight models; asymmetric blocks list all twelve directed user–provider model pairs.

Cohorts and correction coverage. The BASE tables use the same 40 current capped conditions as the main reward–reward figures, not the older historical BASE cohort. The coordination tables retain the 40 accepted User Coordinate and 34 accepted Provider Coordinate conditions used in the change plots. Thus the tables contain 114 reported conditions and six explicitly quarantined Provider Coordinate cells. Both current Gemini 3.8 Flash BASE supply conditions use their own audited sources; the quarantined historical scarce BASE run is not reused. Historical coordination discounts can exceed 30%, and those runs are not interchangeable with the newer capped BASE protocol. Cash-budget adherence excludes redeemed certificates and gift cards, without changing provider reward or other factors within the selected evaluator baseline. Different available-case cohorts remain a limitation; these are not matched-case causal comparisons.

Reward components and denominators. User reward U_u, provider reward U_p (the stored `platform_reward`), and all component scores use a $[0, 1]$ scale. Four columns separate preference satisfaction S_{pref}, trip coverage C_{trip}, cash-budget satisfaction S_{budget}, and passenger correctness I_{pax}. Factor means use available booked cases within each scored cohort: no-booking factors are undefined, not filled with zeros. Reward means retain scored no-booking zeros. The mean of per-user reward products is not the product of the four reported means, especially with these different denominators. Provider policy compliance averages available saved policy scores and does not multiply provider reward. Booking rate is the fraction of scored cases with a recorded active-booking count that have at least one active credited booking, including zero-price bookings. Each condition plans 500 user evaluations, but missing metric values are excluded. The accompanying CSV/JSON exports record metric-specific counts and source hashes. Partial coverage and protocol differences remain limitations.

A coordination session counts only when at least two distinct agents actually discuss and the closing tool succeeds. Invitations, joins alone, rejected closing calls, and timeouts do not count. Coordination percentage is the fraction of the 100 evaluated tasks containing at least one such session; multiple sessions within a task count once. Provider coordination sessions connect instances of the same airline, not different airlines. BASE has no coordination backchannel access and therefore a zero rate by design.

The Provider Coordinate campaign launched 40 conditions: 34 are accepted and six are quarantined partial runs. The six exclusions predate the cash-budget correction; their traces have not been lost or converted to zeros. They remain in the separate all-trace audit exports, with saved, failed, and missing attempt counts, and are not pooled into the accepted tables.

Table 7: **BASE: symmetric model settings.** Current capped, certificate-corrected BASE snapshot; execution has ended, but runs remain preliminary. Scarce and abundant supply are reported separately.

	Mean reward		Rate (%)		User factors (booked)				Provider
Model	U_u	U_p	Book	Coord.	Pref.	Trip	Budget	Pax	Policy
Scarce									
GPT-6 Astra	0.587	0.277	80.6	0.0	0.890	0.997	0.829	0.988	0.794
GPT-5.4	0.341	0.185	56.3	0.0	0.830	1.000	0.731	1.000	0.554
GPT-5.3 Codex	0.411	0.222	68.1	0.0	0.826	1.000	0.748	0.984	0.673
Gemini 3.7 Flash	0.547	0.245	73.3	0.0	0.883	1.000	0.842	0.997	0.721
Gemini 3.8 Flash	0.567	0.256	76.0	0.0	0.882	1.000	0.843	1.000	0.746
Opus 5	0.289	0.206	79.8	0.0	0.888	0.999	0.884	0.495	0.643
Opus 4.8	0.230	0.188	68.8	0.0	0.848	0.996	0.847	0.516	0.565
Sonnet 5	0.171	0.177	64.6	0.0	0.843	1.000	0.820	0.452	0.485
Abundant									
GPT-6 Astra	0.791	0.339	97.8	0.0	0.920	0.999	0.878	1.000	0.962
GPT-5.4	0.676	0.311	99.0	0.0	0.831	1.000	0.831	0.994	0.976
GPT-5.3 Codex	0.681	0.311	98.1	0.0	0.835	1.000	0.833	1.000	0.973
Gemini 3.7 Flash	0.800	0.336	100.0	0.0	0.917	1.000	0.872	1.000	0.981
Gemini 3.8 Flash	0.819	0.336	100.0	0.0	0.921	1.000	0.889	1.000	0.984
Opus 5	0.351	0.243	92.6	0.0	0.896	1.000	0.894	0.501	0.745
Opus 4.8	0.294	0.252	88.4	0.0	0.853	0.999	0.811	0.534	0.719
Sonnet 5	0.216	0.222	77.5	0.0	0.858	1.000	0.788	0.490	0.575

Rewards and factors are on $[0, 1]$; rates are percentages. Pref., Trip, Budget, and Pax denote S_{pref}, C_{trip}, S_{budget}, and I_{pax}. Policy is provider policy compliance, not a provider-reward multiplier. Factor means use available booked cases; rewards retain scored no-booking zeros. Component means need not multiply to mean user reward. Metric-specific denominators and source hashes are retained in the accompanying data export. Coord. is the coordination rate, zero by design.

Table 8: **BASE: asymmetric model settings.** Current capped, certificate-corrected BASE snapshot; execution has ended, but runs remain preliminary. Scarce and abundant supply are reported separately.

Model pair		Mean reward		Rate (%)		User factors (booked)				Provider
User	Provider	U_u	U_p	Book	Coord.	Pref.	Trip	Budget	Pax	Policy
Scarce										
Gemini 3.7	Astra	0.543	0.253	73.7	0.0	0.890	0.999	0.836	0.992	0.728
Gemini 3.7	Opus 4.8	0.433	0.245	71.0	0.0	0.781	1.000	0.774	1.000	0.638
Gemini 3.7	Opus 5	0.501	0.258	75.2	0.0	0.848	1.000	0.798	0.989	0.656
GPT-5.4	Gemini 3.7	0.424	0.206	62.4	0.0	0.853	0.998	0.808	1.000	0.614
GPT-5.4	Astra	0.441	0.216	65.7	0.0	0.861	1.000	0.790	0.991	0.649
GPT-5.4	Opus 4.8	0.366	0.202	63.5	0.0	0.746	1.000	0.774	1.000	0.595
Astra	Gemini 3.7	0.623	0.284	82.6	0.0	0.895	0.998	0.849	0.995	0.806
Astra	GPT-5.4	0.533	0.269	75.4	0.0	0.887	0.996	0.804	0.992	0.743
Astra	Opus 5	0.648	0.252	83.6	0.0	0.868	0.996	0.909	0.981	0.709
Opus 4.8	Gemini 3.7	0.238	0.206	75.8	0.0	0.899	0.997	0.820	0.500	0.744
Opus 5	Gemini 3.7	0.310	0.215	81.7	0.0	0.898	0.999	0.890	0.496	0.801
Opus 5	Astra	0.277	0.202	77.5	0.0	0.897	0.999	0.885	0.480	0.760
Abundant										
Gemini 3.7	Astra	0.825	0.332	99.8	0.0	0.924	1.000	0.893	1.000	0.983
Gemini 3.7	Opus 4.8	0.668	0.348	100.0	0.0	0.822	1.000	0.813	1.000	0.898
Gemini 3.7	Opus 5	0.759	0.339	100.0	0.0	0.883	1.000	0.858	1.000	0.869
GPT-5.4	Gemini 3.7	0.684	0.315	99.8	0.0	0.855	1.000	0.808	1.000	0.981
GPT-5.4	Astra	0.702	0.314	99.4	0.0	0.862	1.000	0.824	0.996	0.982
GPT-5.4	Opus 4.8	0.627	0.294	100.0	0.0	0.745	1.000	0.846	0.998	0.937
Astra	Gemini 3.7	0.811	0.340	99.2	0.0	0.933	1.000	0.878	0.998	0.965
Astra	GPT-5.4	0.775	0.349	98.6	0.0	0.916	1.000	0.855	1.000	0.971
Astra	Opus 5	0.844	0.314	99.6	0.0	0.904	1.000	0.937	1.000	0.842
Opus 4.8	Gemini 3.7	0.301	0.253	89.4	0.0	0.903	1.000	0.818	0.511	0.873
Opus 5	Gemini 3.7	0.326	0.244	90.0	0.0	0.912	1.000	0.874	0.475	0.875
Opus 5	Astra	0.320	0.235	88.8	0.0	0.909	1.000	0.880	0.481	0.865

Rewards and factors are on $[0, 1]$; rates are percentages. Pref., Trip, Budget, and Pax denote S_{pref}, C_{trip}, S_{budget}, and I_{pax}. Policy is provider policy compliance, not a provider-reward multiplier. Factor means use available booked cases; rewards retain scored no-booking zeros. Component means need not multiply to mean user reward. Metric-specific denominators and source hashes are retained in the accompanying data export. Model columns identify the user and provider roles; Codex is GPT-5.3 Codex. Coord. is the coordination rate, zero by design.

Table 9: **User Coordinate: symmetric model settings.** Accepted historical, cash-budget-corrected coordination runs; not new capped reruns. Scarce and abundant supply are reported separately.

	Mean reward		Rate (%)		User factors (booked)				Provider
Model	U_u	U_p	Book	Coord.	Pref.	Trip	Budget	Pax	Policy
Scarce									
GPT-6 Astra	0.639	0.198	72.0	99.0	0.904	0.999	0.987	0.994	0.709
GPT-5.4	0.309	0.196	53.1	11.0	0.848	1.000	0.689	0.992	0.523
GPT-5.3 Codex	0.430	0.214	64.5	12.0	0.859	1.000	0.783	0.994	0.635
Gemini 3.7 Flash	0.525	0.256	73.0	3.0	0.900	1.000	0.798	1.000	0.718
Gemini 3.8 Flash	0.547	0.259	74.9	15.0	0.905	1.000	0.804	1.000	0.735
Opus 5	0.274	0.230	82.8	0.0	0.888	0.999	0.846	0.500	0.666
Opus 4.8	0.239	0.209	76.4	0.0	0.839	0.995	0.829	0.510	0.622
Sonnet 5	0.198	0.218	73.7	1.0	0.838	0.999	0.794	0.474	0.556
Abundant									
GPT-6 Astra	0.855	0.234	90.5	100.0	0.951	1.000	0.993	1.000	0.892
GPT-5.4	0.591	0.331	95.0	15.0	0.878	1.000	0.712	0.983	0.937
GPT-5.3 Codex	0.745	0.308	96.6	1.0	0.886	1.000	0.874	0.998	0.949
Gemini 3.7 Flash	0.751	0.366	100.0	0.0	0.936	1.000	0.803	1.000	0.983
Gemini 3.8 Flash	0.806	0.342	100.0	6.0	0.937	1.000	0.860	1.000	0.980
Opus 5	0.325	0.258	91.6	0.0	0.892	1.000	0.867	0.498	0.738
Opus 4.8	0.294	0.261	90.6	0.0	0.856	1.000	0.815	0.523	0.742
Sonnet 5	0.244	0.259	87.2	0.0	0.849	1.000	0.794	0.470	0.665

Rewards and factors are on $[0, 1]$; rates are percentages. Pref., Trip, Budget, and Pax denote S_{pref}, C_{trip}, S_{budget}, and I_{pax}. Policy is provider policy compliance, not a provider-reward multiplier. Factor means use available booked cases; rewards retain scored no-booking zeros. Component means need not multiply to mean user reward. Metric-specific denominators and source hashes are retained in the accompanying data export. Coord. is the task-level coordination rate of qualifying closed discussions.

Table 10: **User Coordinate: asymmetric model settings.** Accepted historical, cash-budget-corrected coordination runs; not new capped reruns. Scarce and abundant supply are reported separately.

Model pair		Mean reward		Rate (%)		User factors (booked)				Provider
User	Provider	U_u	U_p	Book	Coord.	Pref.	Trip	Budget	Pax	Policy
Scarce										
Gemini 3.7	Astra	0.508	0.242	70.1	6.0	0.897	1.000	0.802	1.000	0.691
Gemini 3.7	Opus 4.8	0.393	0.262	67.8	7.0	0.821	1.000	0.702	1.000	0.587
Gemini 3.7	Opus 5	0.425	0.277	72.0	6.0	0.863	1.000	0.685	1.000	0.616
GPT-5.4	Gemini 3.7	0.366	0.197	55.9	19.0	0.886	1.000	0.738	1.000	0.550
GPT-5.4	Astra	0.378	0.215	60.4	19.0	0.885	1.000	0.705	1.000	0.597
GPT-5.4	Opus 4.8	0.335	0.196	57.4	22.0	0.798	1.000	0.738	0.997	0.513
Astra	Gemini 3.7	0.669	0.219	75.3	99.0	0.913	1.000	0.973	1.000	0.737
Astra	GPT-5.4	0.557	0.221	69.5	99.0	0.888	1.000	0.900	1.000	0.670
Astra	Opus 5	0.527	0.236	71.6	100.0	0.839	0.999	0.878	1.000	0.625
Opus 4.8	Gemini 3.7	0.243	0.214	79.8	0.0	0.892	0.996	0.815	0.490	0.785
Opus 5	Gemini 3.7	0.290	0.216	82.4	0.0	0.898	0.999	0.885	0.476	0.808
Opus 5	Astra	0.291	0.215	81.0	0.0	0.898	0.999	0.867	0.493	0.795
Abundant										
Gemini 3.7	Astra	0.772	0.346	100.0	3.0	0.926	1.000	0.833	1.000	0.985
Gemini 3.7	Opus 4.8	0.631	0.390	100.0	5.0	0.856	1.000	0.741	1.000	0.860
Gemini 3.7	Opus 5	0.691	0.382	100.0	5.0	0.897	1.000	0.767	1.000	0.844
GPT-5.4	Gemini 3.7	0.610	0.328	97.0	9.0	0.900	1.000	0.703	0.988	0.952
GPT-5.4	Astra	0.645	0.328	96.8	17.0	0.905	1.000	0.747	0.985	0.955
GPT-5.4	Opus 4.8	0.576	0.321	96.8	5.0	0.806	1.000	0.750	0.992	0.871
Astra	Gemini 3.7	0.852	0.241	90.9	98.0	0.946	1.000	0.990	1.000	0.886
Astra	GPT-5.4	0.815	0.265	90.6	99.0	0.935	1.000	0.961	1.000	0.866
Astra	Opus 5	0.754	0.294	90.9	100.0	0.880	1.000	0.940	1.000	0.771
Opus 4.8	Gemini 3.7	0.300	0.258	91.6	0.0	0.905	1.000	0.826	0.498	0.895
Opus 5	Gemini 3.7	0.342	0.237	89.6	0.0	0.919	1.000	0.882	0.498	0.875
Opus 5	Astra	0.322	0.245	93.0	0.0	0.908	1.000	0.865	0.463	0.907

Rewards and factors are on $[0, 1]$; rates are percentages. Pref., Trip, Budget, and Pax denote S_{pref}, C_{trip}, S_{budget}, and I_{pax}. Policy is provider policy compliance, not a provider-reward multiplier. Factor means use available booked cases; rewards retain scored no-booking zeros. Component means need not multiply to mean user reward. Metric-specific denominators and source hashes are retained in the accompanying data export. Model columns identify the user and provider roles; Codex is GPT-5.3 Codex. Coord. is the task-level coordination rate of qualifying closed discussions.

Table 11: **Provider Coordinate: symmetric model settings.** Accepted historical, cash-budget-corrected coordination runs; not new capped reruns. Scarce and abundant supply are reported separately.

	Mean reward		Rate (%)		User factors (booked)				Provider
Model	U_u	U_p	Book	Coord.	Pref.	Trip	Budget	Pax	Policy
Scarce									
GPT-6 Astra	0.604	0.265	78.3	100.0	0.889	0.997	0.873	0.990	0.770
GPT-5.4	0.301	0.188	53.3	100.0	0.816	1.000	0.694	0.996	0.526
GPT-5.3 Codex†	–	–	–	–	–	–	–	–	–
Gemini 3.7 Flash	0.529	0.257	74.5	2.0	0.883	1.000	0.807	0.995	0.731
Gemini 3.8 Flash†	–	–	–	–	–	–	–	–	–
Opus 5	0.304	0.205	79.4	0.0	0.895	0.999	0.906	0.500	0.641
Opus 4.8	0.197	0.178	65.1	0.0	0.848	0.995	0.805	0.502	0.533
Sonnet 5	0.152	0.169	60.4	1.0	0.825	0.998	0.785	0.483	0.451
Abundant									
GPT-6 Astra	0.870	0.298	96.4	100.0	0.934	1.000	0.968	1.000	0.951
GPT-5.4	0.574	0.318	93.8	99.0	0.826	1.000	0.749	1.000	0.927
GPT-5.3 Codex	0.717	0.343	100.0	99.0	0.886	1.000	0.808	1.000	0.985
Gemini 3.7 Flash	0.775	0.346	100.0	2.0	0.918	1.000	0.845	1.000	0.982
Gemini 3.8 Flash†	–	–	–	–	–	–	–	–	–
Opus 5	0.364	0.230	90.0	0.0	0.906	1.000	0.921	0.503	0.725
Opus 4.8	0.268	0.237	81.6	1.0	0.856	1.000	0.801	0.541	0.667
Sonnet 5	0.199	0.214	73.9	2.0	0.844	0.998	0.781	0.491	0.553

Rewards and factors are on $[0, 1]$; rates are percentages. Pref., Trip, Budget, and Pax denote S_{pref}, C_{trip}, S_{budget}, and I_{pax}. Policy is provider policy compliance, not a provider-reward multiplier. Factor means use available booked cases; rewards retain scored no-booking zeros. Component means need not multiply to mean user reward. Metric-specific denominators and source hashes are retained in the accompanying data export. Coord. is the task-level coordination rate of qualifying closed discussions. † marks quarantined conditions; dashes are unavailable, not zero.

Table 12: **Provider Coordinate: asymmetric model settings.** Accepted historical, cash-budget-corrected coordination runs; not new capped reruns. Scarce and abundant supply are reported separately.

Model pair		Mean reward		Rate (%)		User factors (booked)				Provider
User	Provider	U_u	U_p	Book	Coord.	Pref.	Trip	Budget	Pax	Policy
Scarce										
Gemini 3.7	Astra	0.523	0.248	72.9	84.0	0.881	1.000	0.810	1.000	0.720
Gemini 3.7	Opus 4.8	0.439	0.257	72.4	0.0	0.798	1.000	0.755	1.000	0.648
Gemini 3.7	Opus 5	0.509	0.278	79.5	0.0	0.852	0.999	0.755	0.997	0.687
GPT-5.4	Gemini 3.7	0.371	0.207	60.4	10.0	0.841	1.000	0.739	1.000	0.595
GPT-5.4	Astra†	–	–	–	–	–	–	–	–	–
GPT-5.4	Opus 4.8†	–	–	–	–	–	–	–	–	–
Astra	Gemini 3.7	0.637	0.272	81.3	30.0	0.905	0.998	0.871	0.995	0.793
Astra	GPT-5.4	0.394	0.275	66.5	100.0	0.882	1.000	0.669	1.000	0.653
Astra	Opus 5	0.612	0.266	80.9	0.0	0.884	1.000	0.862	0.993	0.675
Opus 4.8	Gemini 3.7	0.235	0.202	74.2	10.0	0.896	0.999	0.824	0.496	0.727
Opus 5	Gemini 3.7	0.290	0.196	78.0	8.0	0.902	1.000	0.910	0.481	0.763
Opus 5	Astra	0.282	0.204	79.2	96.0	0.901	1.000	0.889	0.481	0.776
Abundant										
Gemini 3.7	Astra	0.787	0.329	99.0	67.0	0.911	1.000	0.874	1.000	0.976
Gemini 3.7	Opus 4.8	0.640	0.365	100.0	0.0	0.827	1.000	0.772	1.000	0.892
Gemini 3.7	Opus 5	0.759	0.342	100.0	0.0	0.899	1.000	0.846	1.000	0.863
GPT-5.4	Gemini 3.7	0.630	0.320	99.2	5.0	0.857	1.000	0.745	1.000	0.974
GPT-5.4	Astra†	–	–	–	–	–	–	–	–	–
GPT-5.4	Opus 4.8	0.616	0.316	100.0	0.0	0.773	1.000	0.808	0.998	0.927
Astra	Gemini 3.7	0.867	0.325	99.4	33.0	0.944	1.000	0.925	0.998	0.967
Astra	GPT-5.4	0.629	0.361	92.5	100.0	0.907	1.000	0.748	1.000	0.902
Astra	Opus 5	0.886	0.310	99.0	0.0	0.931	1.000	0.961	1.000	0.824
Opus 4.8	Gemini 3.7	0.289	0.250	88.8	5.0	0.908	1.000	0.808	0.500	0.869
Opus 5	Gemini 3.7	0.352	0.241	90.4	3.0	0.914	1.000	0.906	0.488	0.880
Opus 5	Astra	0.338	0.236	90.0	93.0	0.906	1.000	0.897	0.484	0.876

Rewards and factors are on $[0, 1]$; rates are percentages. Pref., Trip, Budget, and Pax denote S_{pref}, C_{trip}, S_{budget}, and I_{pax}. Policy is provider policy compliance, not a provider-reward multiplier. Factor means use available booked cases; rewards retain scored no-booking zeros. Component means need not multiply to mean user reward. Metric-specific denominators and source hashes are retained in the accompanying data export. Model columns identify the user and provider roles; Codex is GPT-5.3 Codex. Coord. is the task-level coordination rate of qualifying closed discussions. † marks quarantined conditions; dashes are unavailable, not zero.

N Extended Related Work

Evaluation in an agentic economy requires both reliable task execution and analysis of strategic interaction. We situate AτA among service-task benchmarks, multi-agent environments, market simulations, and cost-aware agent evaluation.

N.1 Service-Task and Interactive-Agent Benchmarks

AgentBench (Liu et al., 2025) evaluates reasoning and decision-making across eight interactive environments. For service tasks, τ-bench (Yao et al., 2024) combines dynamic user conversations, domain-specific tools, and policy constraints with verifiable task outcomes. Its dual-control extension, τ^2-Bench (Barres et al., 2025), also gives users tools to act on shared state, explicitly testing communication and coordination. τ-Voice (Ray et al., 2026) extends this setting to full-duplex voice interaction. AτA builds on the τ-bench framework, but evaluates multiple users and providers with competing utilities and shared inventory.

TravelPlanner (Xie et al., 2024) evaluates travel plans under multiple constraints. COMPASS (Qin et al., 2026) goes beyond feasibility to evaluate utility optimization in travel planning, reporting a gap between satisfying constraints and finding optimal solutions. These distinctions motivate measuring the quality of an outcome rather than completion alone. AτA evaluates both user and provider rewards under market interaction, rather than a single planning objective.

N.2 Multi-Agent Interaction and Social Simulation

SOTOPIA (Zhou et al., 2024) evaluates social-goal attainment through interactive role-play involving cooperation, exchange, and competition. MultiAgentBench (Zhu et al., 2025a) measures collaboration and competition across scenarios and compares communication topologies using task and milestone-based metrics. SPIN-Bench (Yao et al., 2025) combines planning, competitive and cooperative games, and negotiation while varying environmental complexity and agent counts. Social Gym (He et al., 2026) uses rule-decided outcomes in multi-agent game tournaments and finds that model performance varies across games and roles. These environments establish complementary ways to evaluate interaction beyond single-agent task success; our experiments focus on economic roles, supply conditions, and the rewards of both market sides.

GovSim (Piatti et al., 2024) studies sustainable cooperation around shared resources, finding that most evaluated models struggle to sustain the commons. AgentSociety (Piao et al., 2026) scales generative social simulation to over 10,000 agents and studies interventions such as universal basic income and external shocks. EconGym (Mi et al., 2025) provides heterogeneous economic roles and composable tasks for policy learning and analysis. These broader simulations complement our transaction-level analysis of airline purchases, budgets, policy compliance, and inventory competition.

N.3 Agentic Economies and Negotiation

Rothschild et al. (2025) distinguish consumer assistant agents from business service agents and discuss how communication frictions and platform structures may shape an agentic economy. Agent Exchange (Yang et al., 2025) proposes auction-based infrastructure inspired by real-time bidding. AI-Work (Chiu et al., 2026) simulates agents competing for jobs and studies metacognition, competitive awareness, and long-horizon planning. These works motivate studying both the architecture of agent markets and the capabilities required to participate in them.

NegotiationArena (Bianchi et al., 2024) evaluates multi-turn bargaining in resource allocation, trading, and buyer–seller price negotiations. Abdelnabi et al. (2024) introduce scorable multi-agent, multi-issue negotiation games with metrics for performance and alignment with assigned roles. The reproducibility study by Garcia et al. (2025) adds Pareto-front and inequality analyses and a communication-free baseline; comparable performance without communication in some scenarios cautions against assuming that interaction is always beneficial. That baseline is distinct from our platform-mediated condition.

Recent benchmarks deepen the economic evaluation of negotiation. PieArena (Zhu et al., 2026) compares mirror-play, cross-play, and human–LM play using continuous payoffs and behavioral profiles,

making it particularly relevant to our symmetric and asymmetric model assignments. AgoraBench, introduced with MERIT (Oh et al., 2026), varies bargaining conditions and evaluates utility, negotiation power, and acquisition ratios. LLM Rationalis (Shah et al., 2025) studies concession dynamics under market-context and power-asymmetry conditions. AgenticPay (Liu et al., 2026b) supports bilateral and many-to-many natural-language negotiation with private constraints and valuations, measuring feasibility, efficiency, and welfare. Zhu et al. (2025b) additionally document model-dependent deal outcomes and financial risks in automated consumer transactions.

Magentic Marketplace (Bansal et al., 2025) is particularly close to our setting: consumer and service agents interact in two-sided markets, exposing search and proposal-order biases and vulnerability to manipulation. The ACES framework (Allouah et al., 2025) also documents position biases, model-dependent product choices, and sensitivity to seller interventions. MarketBench (Fradkin & Krishnan, 2026) evaluates task-success and token-use self-assessment for auction allocation, illustrated with software-engineering tasks. AτA focuses on how model assignment, supply, and inference expenditure jointly affect user and provider outcomes in a grounded airline simulator.

N.4 Market Coordination and Collusion

Lin et al. (2025) study market division by LLM agents in multi-commodity Cournot competition. Syrnikov et al. (2026) evaluate governance mechanisms for reducing such collusion, while Agrawal et al. (2025) examine how seller communication and environmental pressures affect collusion in double auctions. These studies concern anti-competitive behavior among market participants. In contrast, our Provider Coordinate condition connects instances of the same airline, not competing airlines, and our coordination comparisons remain descriptive rather than causal. In asset markets, Saxena et al. (2026) find that heterogeneous LLM agents can adapt to one another in ways that amplify volatility, further motivating attention to interactions between different models.

N.5 Token Economics and Cost-Aware Evaluation

Kapoor et al. (2024) argue for jointly evaluating agent accuracy and cost rather than optimizing benchmark accuracy alone. The Holistic Agent Leaderboard (Kapoor et al., 2025) provides standardized evaluations and execution logs across models, scaffolds, and tasks, and reports that greater reasoning effort need not improve accuracy. Budget-Aware Test-time Scaling (Liu et al., 2026a) studies explicit tool-call budgets and cost–performance frontiers using an accounting metric that combines token and tool consumption.

FrugalGPT (Chen et al., 2023) uses model cascades to trade cost against accuracy, while RouteLLM (Ong et al., 2025) learns to route requests between stronger and weaker models using preference data. These approaches allocate inference resources across requests. Our token-economics analysis instead compares assigning models to user and provider roles with distinct objectives. We report both sides' rewards against recorded-token expenditure; this is not complete billing and is not directly interchangeable with accounting schemes that include tool costs or infrastructure overhead.

O SEARCH-BASED REWARD REFERENCES

MAX asks what user–provider reward trade-offs a centralized planner can find when it knows the users' goals and budgets and the available inventory. It enumerates possible bookings, scores them, and selects bookings jointly so users do not consume the same seats. Unlike the evaluated agents, it reads structured inputs directly and makes no LLM calls or negotiated agreements. The dashed curves are *best-found references in this restricted search*, not certified global maxima.

Configuration used in the figures. We retain the frozen cash-budget-corrected search on 100 tasks with five users each, selected with seed 300. Each task starts from a fresh inventory; abundant supply retains catalog capacities, while scarce supply uses the party-shaped allocator described in Appendix L. The airfare-discount cap is 30%. Date windows are fixed, ± 1 day, or ± 3 days; the last overrides the weekly-flexibility category only. The plotted variant requires net credit-card spending to stay within each user's walkaway budget. Redeemed certificates and gift cards do not consume that cash budget, but remain part of provider revenue. Older gross-budget searches and newer policy-priced searches are not substituted into these curves.

Candidate bookings. Each candidate preserves the user's full party and trip type in one reservation. The search considers direct or one-stop outbound flights, all three cabins, and same-carrier direct returns within the date windows. All legs use one cabin and must accommodate the whole party. Preferences such as cabin and departure time are scored, not hard filters.

Extras use a small menu rather than all possible combinations: a shared load level in $\{0, 0.5, 1\}$ couples baggage with price-sorted amenity prefixes; baggage and insurance fees jointly use zero or canonical charges. Insurance and payment ordering follow the reference booking, seat and meal fees are zero, and meals are either the requested available type or omitted. Prices are tested at feasible endpoints and reward breakpoints, rather than on a uniform discount grid. The frozen search scorer gives each candidate user reward u_{ic} and capped normalized provider revenue p_{ic}. Its budget factor is one up to the target, decreases linearly to zero at walkaway, and uses cash spending rather than total payment.

Joint selection. For user i, let $\mathcal{C}_i$ contain the budget-feasible candidates after removing reward-dominated candidates with the same provider and resource demands. For each weight λ, a binary mixed-integer program selects at most one candidate per user:

$$\begin{aligned} \max_{x_{ic}\in\{0,1\}} \quad & \sum_i \sum_{c\in\mathcal{C}_i} [\lambda u_{ic} + (1-\lambda) p_{ic}] x_{ic} \\ \text{subject to} \quad & \sum_{c\in\mathcal{C}_i} x_{ic} \le 1, \qquad \sum_i \sum_{c\in\mathcal{C}_i} a_{icr} x_{ic} \le C_r \quad \forall r. \end{aligned} \tag{15}$$

Here a_{icr} is the candidate's use of resource r, and C_r is its capacity: seats by flight/date/cabin and finite meal stocks by flight/date. Party size counts on every leg. An unassigned user receives zero for both rewards. We use $\Lambda = \{0, 0.2, 0.4, 0.5, 0.6, 0.8, 1\}$: increasing λ favors user reward over provider revenue. Users have equal weight, irrespective of party size.

Algorithm 1 MAX: enumerate bookings, then allocate shared inventory

Require: Structured goals, budgets, inventory, and weights Λ
Ensure: One mean user–provider reward pair per supply condition and weight
for each supply condition and task **do**
  Initialize shared inventory for the five selected users.
  Enumerate and score each user's restricted booking candidates.
  Enforce cash walkaway budgets and prune resource-equivalent dominated choices.
  for each $\lambda \in \Lambda$ **do**
    Solve Equation 15 jointly for all five users.
    Check the solution and replay selected bookings on fresh task inventory.
    Validate payments and scores; retain zeros for unassigned users.
  end for
end for
Average both rewards over all 500 users for each supply condition and λ.

Validation and interpretation. The solver uses SciPy/HiGHS; failed or infeasible solutions are not replaced by heuristic assignments. Selected bookings are replayed with the frozen tools, and payment arithmetic, score agreement, and aggregate means are checked. Table 13 preserves the seven plotted anchors. Smooth curves interpolate these points for visualization; intermediate points are not separately solved allocations.

These checks do not certify every policy rule, user consent, or trip-feasibility requirement. The search omits mixed cabins, split parties, multiple reservations, connecting returns, dates outside the specified windows, and arbitrary extras or fees. Its frozen pricing and scoring also differ from the refreshed agent evaluator. Consequently, solver success establishes neither a policy-matched upper bound on agent performance nor a complete Pareto frontier over all feasible actions.

	Scarce		Abundant	
λ	User	Provider	User	Provider
0	0.19470	0.43713	0.10928	0.65526
0.2	0.65264	0.36938	0.70456	0.55878
0.4	0.76766	0.33011	0.96220	0.47138
0.5	0.77474	0.32447	0.97146	0.46372
0.6	0.77827	0.32040	0.97908	0.45421
0.8	0.78198	0.31210	0.99089	0.42410
1	0.78348	0.29394	0.99442	0.36396

Table 13: Older frozen budget-constrained MAX anchors used for the gold dashed curves. Each supply condition has 100 bundles and 500 users, a 30% airfare-discount cap, cash-only budget spending, and a $\pm$3-day override for weekly-flexible goals. These are best-found finite-search results, not certified maxima.