CoffeeBench: Benchmarking Long-Horizon LLM Agents in Heterogeneous Multi-Agent Economies

Issa Sugiura [1] **Daichi Hattori** [2] **Kazuo Araragi** [2] **Keita Ogawa** [2] **Shota Onose** [2] **Taro Makino** [1]
Teppei Usuki [2] **Takashi Ishida** [1]

Abstract

As LLM agents become capable of increasingly long-horizon tasks, evaluating their performance in economic systems is becoming increasingly important. Unlike existing benchmarks that primarily evaluate a single agent interacting with a passive environment, economic systems are inherently multi-agent, requiring autonomous agents to communicate, negotiate, and transact while pursuing their own objectives over extended periods. We introduce CoffeeBench, a benchmark for evaluating LLM agents in a long-horizon multi-agent economy composed of heterogeneous firms. In CoffeeBench, two farmers, two roasters, and two retailers autonomously operate their businesses over a 90-day simulation, each seeking to maximize cumulative net income through communication and transactions while managing cash, inventory, and pricing. The evaluated model controls one coffee roaster, while the remaining firms are controlled by fixed reference agents. Across several recent open-weight and proprietary LLMs, all models outperform a passive baseline that takes no actions, with most achieving positive net income. Analysis of agent behavior reveals substantial differences in long-horizon economic interaction: higher-performing models communicate more actively with other firms, whereas Claude Haiku 4.5 exhibits an *idle-drift* failure mode, repeatedly choosing inaction despite producing coherent assessments and plans. We release our code and agent trajectories to support future research.

Code: GitHub Repository

Trajectories: Agent Trajectories

[1]Sakana AI [2]KPMG AZSA LLC. Correspondence to: Issa Sugiura <issasugiura@sakana.ai>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

1. Introduction

As large language models (LLMs) continue to advance, their applications have expanded to long-horizon tasks requiring sequential decision-making, long-term planning, and interaction with environments (Yao et al., 2025; Jimenez et al., 2024; Li et al., 2026; Merrill et al., 2026; Yang et al., 2026). Coding agents (Yang et al., 2024; Merrill et al., 2026) and web-use agents (Zhou et al., 2024) are prominent examples that iteratively interact with external environments to solve complex tasks. Beyond these domains, LLM agents are increasingly expected to find applications across a wide range of industries, including finance, healthcare, and manufacturing (Xiao et al., 2025; Arora et al., 2025; Patwardhan et al., 2026; Sugiura et al., 2026).

Economic systems provide a natural testbed for long-horizon LLM agents because they require sustained decision making while interacting with other autonomous agents (Andon Labs, 2025a). Firms communicate, negotiate, and transact while pursuing their own objectives, creating complex dynamics that extend beyond planning and tool use alone (Cyert & March, 2020). Recent work has explored long-horizon business management benchmarks such as Vending-Bench (Backlund & Petersson, 2025) and Vending-Bench Arena (Andon Labs, 2025b). However, existing benchmarks model either a single autonomous firm or multiple homogeneous firms, whereas real-world economies consist of heterogeneous firms with distinct economic roles that interact while pursuing their own objectives (Tadelis, 2013).

To this end, we introduce **CoffeeBench**, a benchmark for evaluating long-horizon LLM agents in a multi-agent economy composed of heterogeneous firms. In CoffeeBench, two farmers, two roasters, and two retailers autonomously operate their businesses over a 90-day simulation, each seeking to maximize cumulative net income through communication and transactions while managing cash, inventory, and pricing. The evaluated model controls one coffee roaster, while the remaining firms are controlled by fixed reference agents. Each evaluation requires hundreds to thousands of tool calls, demanding sustained planning and decision-making over extended horizons.

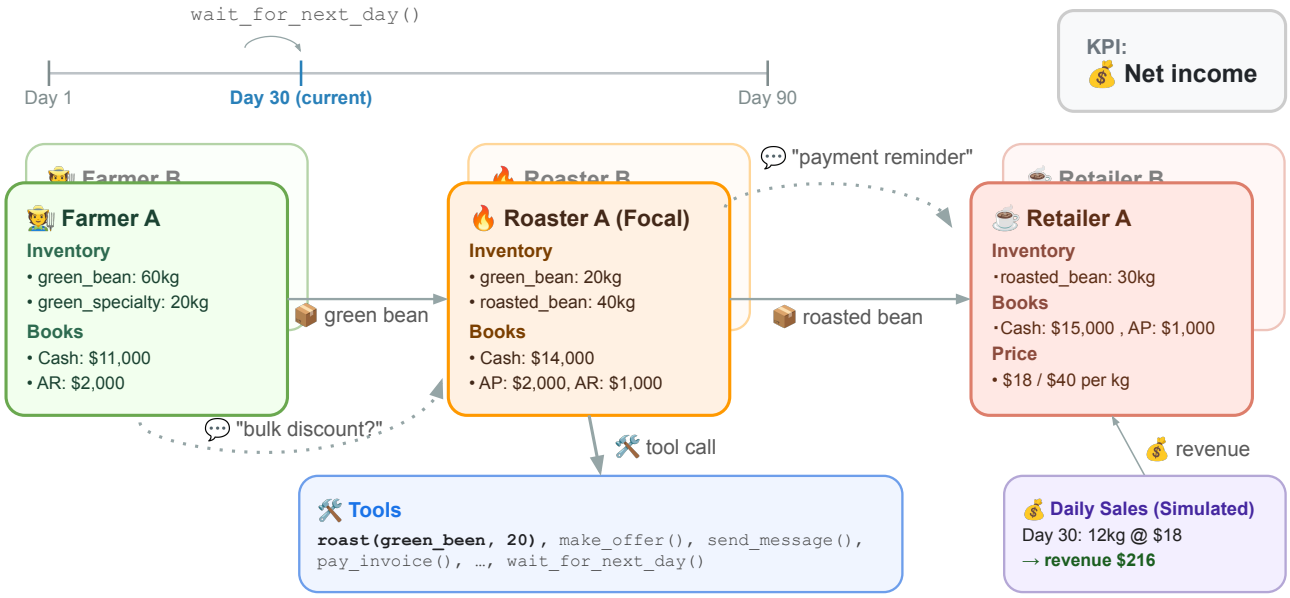

Figure 1. Overview of CoffeeBench. Six LLM agents operate firms in a simulated coffee supply chain over a 90-day period, each seeking to maximize its own cumulative net income. In each evaluation run, the model under evaluation is assigned the role of one coffee roaster, while the remaining firms are controlled by a fixed reference model.

We evaluate several recent open-weight and proprietary LLMs on CoffeeBench. All models outperform a passive baseline, with most achieving positive net income. Higher-performing models such as GPT-5.5 (OpenAI, 2026) and Claude Opus 4.7 (Anthropic, 2026a) tend to communicate more actively with counterparties, while Claude Haiku 4.5 (Anthropic, 2025) exhibits an *idle-drift* failure mode, in which the agent maintains coherent reasoning traces yet repeatedly chooses to wait rather than act, resulting in prolonged operational inactivity and low net income.

2. Related Work

Long-horizon benchmarks. Early LLM benchmarks evaluated single-turn tasks such as question answering, including MMLU (Hendrycks et al., 2021) and Humanity's Last Exam (Phan et al., 2026). As LLMs improved in long-context understanding, tool use, and reasoning, ReAct-based LLM agents (Yao et al., 2023) became capable of completing long-horizon tasks by iteratively taking actions within specific environments, spurring a growing number of benchmarks across diverse domains, including software engineering (Jimenez et al., 2024; Yang et al., 2026; Merrill et al., 2026), web navigation (Nakano et al., 2022; Yao et al., 2022; He et al., 2024; Zhou et al., 2024), and desktop operation (Xie et al., 2024; Rawles et al., 2025). More recent benchmarks further extend this paradigm to settings where the environment itself evolves asynchronously, requiring agents to continuously adapt and execute tasks under changing conditions (Froger et al., 2026; Goel et al., 2026). This growing interest reflects a broader shift toward evaluating agents in increasingly realistic, open-ended settings (Yang et al., 2025; Imajuku et al., 2025).

Business management benchmarks. Business management is one domain where long-horizon evaluation is of interest, as agents must make sequential decisions over extended time horizons in dynamic economic environments (Backlund & Petersson, 2025; Andon Labs, 2025a). Table 1 compares recent business management benchmarks for LLM agents. Vending-Bench (Backlund & Petersson, 2025) evaluates whether an LLM agent can autonomously operate a vending machine business and generate profit over an extended horizon, while Vending-Bench Arena (Andon Labs, 2025b) extends this setting to multiple competing vending agents. However, existing benchmarks involve only a single economic role, with all agents operating homogeneous businesses. In contrast, real-world economies consist of heterogeneous firms that interact through communication and transactions while pursuing their own objectives. To the best of our knowledge, CoffeeBench is the first benchmark to evaluate LLM agents in such a multi-agent economy, where farmers, roasters, and retailers each operate autonomously to maximize their own cumulative net income.

Undesirable behaviors in LLM agents. Recent work has identified undesirable behaviors in LLM agents, including reward hacking and model cheating (Zhong et al., 2026;

Table 1. Comparison of business management benchmarks. CoffeeBench is a benchmark that combines multiple autonomous firms with heterogeneous economic roles.

Benchmark	Domain	Long-horizon	Multi-agent	Economic roles
Vending-Bench (Backlund & Petersson, 2025)	Vending	✓	✗	1
Vending-Bench Arena (Andon Labs, 2025b)	Vending	✓	✓	1
CoffeeBench (Ours)	Coffee supply chain	✓	✓	3

Wang et al., 2026; Rank et al., 2026), primarily in coding, math, and machine learning domains. More recent work suggests that economically motivated settings can induce problematic behaviors such as excessive profit-seeking or unsafe decision making under competing objectives (Lynch et al., 2025; Li et al., 2025). Empirical evidence from Vending-Bench has also reported aggressive or undesirable behaviors in frontier models under competitive pressure (Andon Labs, 2026; Anthropic, 2026c). CoffeeBench complements this line of research by providing a controllable multi-agent economic environment that enables systematic study of undesirable behaviors that may emerge through strategic coordination among agents.

3. CoffeeBench

Figure 1 shows an overview of CoffeeBench. CoffeeBench simulates a coffee supply chain in which LLM-driven firms interact in a shared marketplace over a multi-month horizon. The economy consists of six firms spanning three stages of the supply chain: two farmers (`farmer_A`, `farmer_B`), two roasters (`roaster_A`, `roaster_B`), and two retailers (`retailer_A`, `retailer_B`). This structure creates both horizontal competition within each layer of the supply chain and vertical dependencies across layers.

The environment models two parallel coffee supply chains: a commodity segment and a specialty segment. Farmers supply green coffee beans (`green_coffee_kg` and `green_specialty_kg`), roasters convert them into roasted products (`roasted_coffee_kg` and `roasted_specialty_kg`), and retailers sell the roasted products to end consumers.

3.1. Time Management

Time management in CoffeeBench follows the event-driven process illustrated in Figure 2, combining per-action time costs, explicit day transitions, and event-driven reactivation, inspired by prior agentic benchmarks (Backlund & Petersson, 2025; Froger et al., 2026).

Each agent operates within a daily business window from 09:00 to 19:00. Every proactively scheduled tool call advances the agent's local clock by 30 minutes, thereby limiting the number of proactive actions available per day. Agents call `wait_for_next_day()` when no further actions are taken, ending their activity for the current day.

The environment supports asynchronous interaction across agents. While an agent is idle (whether after calling `wait_for_next_day()` or while parked between proactive cycles), it can be reactivated within the same day by incoming events such as messages, trade offers, deal closures, or arriving deliveries, triggering a notification and enabling further actions. This event-driven design enables agents to respond reactively to market activity rather than following a strictly synchronous schedule. Between the end of each simulated day and the next morning, the environment applies system-wide updates including consumer sales, operating costs, spoilage, and financial accruals, so that each agent starts the next morning with these results reflected in its state. The simulation horizon is configurable, with runs proceeding for a fixed number of days unless terminated early (e.g., due to bankruptcy).

3.2. Agents and Tools

All agents are implemented using the ReAct framework (Yao et al., 2023) and receive a system prompt composed of a shared specification and a role-specific instruction. Each agent maintains an internal operational state, including cash balance, inventory, accounts receivable, and accounts payable. Agents differ only in their role-specific prompts and available tool sets. Full prompt specifications are provided in Appendix C.

Shared tools. All agents have access to a common set of tools for trading and communication, including `post_listing()` (create a sell order for an item), `make_offer()` (propose a purchase with price, quantity, and payment terms), `accept_offer()` (finalize a transaction), `pay_invoice()` (settle outstanding payables), and `send_message()` (exchange free-form messages with other agents). These tools enable unrestricted peer-to-peer interaction, allowing arbitrary trade and communication among agents and supporting complex transaction patterns such as reselling and multi-hop trade flows. A full tool catalog is provided in Appendix D.

Role-specific tools. Each role has one or more role-specific tools. Farmers use `produce_item()` to produce coffee beans, incurring a production cost and delay before the output is added to inventory. Roasters use `roast()`

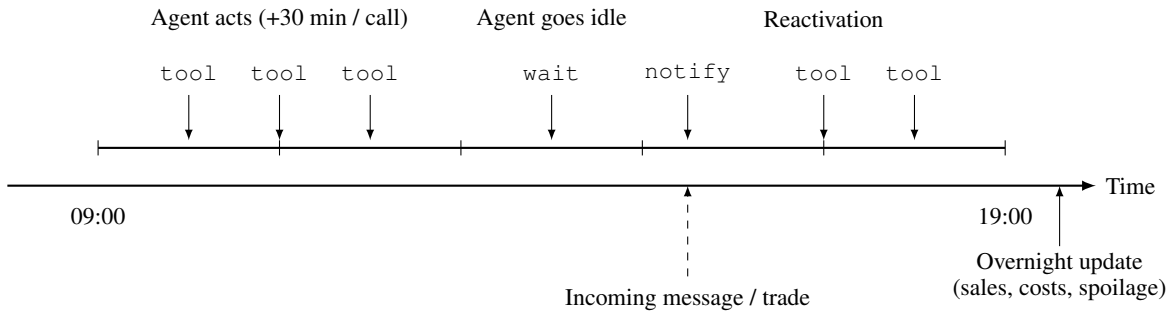

Figure 2. Simulation timeline with asynchronous event-driven interaction. Agents proactively execute tool calls that advance local time, may enter idle states, and can be reactivated within the same day by external events such as incoming messages or trade activity.

to convert green beans into roasted products, also with cost and delay. Retailers use `set_retail_price()` to set consumer-facing prices, which directly influence demand, and `view_consumer_sales()` to inspect their own daily sales history.

3.3. Marketplace and Transactions

All trade is conducted through a shared marketplace. Agents may post listings for items in their inventory, and other agents may submit offers specifying price, quantity, and payment terms. Transactions are completed when offers are accepted.

Accepted trades generate shipments with a one-day delivery lag. Upon delivery, an invoice is issued under a default net-30 payment term (payment due within 30 days). Late payments incur interest, and shipments may be delayed or lost due to stochastic logistics. Because there are no role-based restrictions on trading, agents may engage in arbitrary transaction patterns.

3.4. Consumer Demand

Retailers sell products to external consumers through a competitive demand model executed once per day. Demand depends on retailer pricing and brand loyalty, creating a competitive pricing environment with partially observable dynamics. We describe the detailed specification in Appendix B.

3.5. Economic Constraints

The environment imposes several constraints to induce long-term strategic behavior. Agents incur fixed daily operating costs, and inventory is subject to spoilage. Storage capacity is limited, preventing unbounded accumulation. Production and delivery involve delays, and transactions occur on credit with potential default risk. Agents whose cash balance becomes negative are declared bankrupt and removed from the simulation, and their unpaid obligations are written off as bad debt for creditors. These constraints couple short-term operational decisions with long-term financial outcomes, requiring agents to balance cash flow, inventory risk, pricing, and counterparty reliability under non-stationary market conditions.

3.6. KPI

Each agent is instructed through its system prompt to maximize cumulative net income over the simulated horizon, defined as:

$$\begin{aligned}\mathrm{NetIncome} = \mathrm{Revenue} - \mathrm{COGS} - \mathrm{OpEx} \\ - \mathrm{InterestExp} + \mathrm{InterestRev}\end{aligned} \tag{1}$$

where $\mathrm{Revenue}$ is gross sales net of returns; COGS is the weighted-average cost basis of inventory consumed (compounded across production costs and trade purchase prices, so an agent cannot inflate its score by self-trading at marked-up prices); OpEx aggregates fixed daily operating costs, inventory spoilage, and other operating charges (bad-debt expense and inventory writedowns); and $\mathrm{InterestExp}$ and $\mathrm{InterestRev}$ correspond to accrued late fees on overdue payables and receivables, respectively.

4. Experiments

We evaluate several recent LLMs on CoffeeBench to assess their ability to operate profitably in a multi-agent economy over an extended horizon.

Evaluation protocol. In each run, the evaluated LLM is assigned the role of `roaster_A`, while the remaining five firms are independently operated by LLM agents that also seek to maximize cumulative net income. Unless otherwise specified, background agents are instantiated with Claude Sonnet 4.6 (Anthropic, 2026b), chosen for its stable long-horizon behavior at moderate inference cost. All agents use the ReAct framework (Yao et al., 2023). We set the simulation horizon to 90 days, long enough to observe longer-horizon strategic decision-making while remaining

Table 2. Performance of each model as `roaster_A` in CoffeeBench. Net income is the primary metric. Revenue is cumulative sales. Calls denotes the total number of tool invocations. Idle days counts days on which the model issued only `wait_for_next_day()`. DMs sent denotes the number of `send_message()` calls from `roaster_A` to any recipient. API cost is the per-run LLM spend (USD). Values are reported as mean ± std over three runs.

Model	Net income ($)	Revenue ($)	Calls	Idle days	DMs sent	API cost ($)
GPT-5.5	**+3,109** ±1,123	15,895 ±2,141	1,269 ±193	0 ±0	140 ±22	79.3 ±14.4
Claude Opus 4.7	+2,782 ±2,263	15,092 ±2,838	1,117 ±81	0 ±0	88 ±13	85.7 ±10.9
Claude Sonnet 4.6	+2,236 ±1,489	16,961 ±2,528	1,558 ±84	0 ±0	151 ±25	64.2 ±5.8
Gemini 3.1 Pro	+1,695 ±508	11,746 ±980	910 ±41	0 ±0	16 ±3	31.4 ±1.5
GLM-5.1	+1,597 ±1,199	**16,962** ±1,071	1,373 ±100	0 ±0	78 ±52	67.4 ±6.6
Kimi K2.6	+454 ±1,420	11,748 ±860	1,173 ±97	4 ±5	14 ±9	26.9 ±3.2
Claude Haiku 4.5	−630 ±1,745	7,638 ±2,428	786 ±167	40 ±22	52 ±12	9.6 ±2.9
HeuristicRoaster	−1,931 ±429	4,428 ±925	830 ±18	0 ±0	0 ±0	—
PassiveRoaster	−2,765 ±0	0 ±0	158 ±12	90 ±0	0 ±0	—

tractable in terms of wall-clock time and API cost.

Models. We evaluate five frontier closed models: Claude Opus 4.7 (Anthropic, 2026a), Claude Sonnet 4.6 (Anthropic, 2026b), Claude Haiku 4.5 (Anthropic, 2025), GPT-5.5 (OpenAI, 2026), and Gemini 3.1 Pro (Google DeepMind, 2026), accessed via their official APIs. We additionally include two open-weight models, Kimi K2.6 (Kimi Team, 2026) and GLM-5.1 (GLM-5 Team, 2026), served via OpenRouter (OpenRouter, 2026). Reasoning is disabled or minimized across all providers to control latency and cost. We also include two rule-based baselines: *PassiveRoaster*, which always issues `wait_for_next_day()` and serves as a do-nothing lower bound, and *HeuristicRoaster*, which follows a fixed inventory and pricing policy based on static cost-basis margins without messaging or strategic adaptation.

Context management. The agent maintains its interaction history by appending actions and observations at every step. In the 90-day simulation, this history would eventually exceed the model's maximum context length. Therefore, once the history surpasses 160k tokens, we summarize the intermediate portion using the same underlying model, while retaining the system prompt, the initial trajectory, and the most recent 20 steps.

Stochasticity and variance. Variance across runs is relatively high due to environment randomness and the inherent stochasticity of multi-agent LLM-based systems, leading to different trajectories across runs. We therefore execute three independent runs per setting and report averaged results.

Environment settings. All firms begin with equal initial cash ($15,000) and role-specific starting inventory (60 kg of green commodity coffee for farmers, and 30 kg and 25 kg of roasted commodity coffee for roasters and retailers, respectively). Each firm is subject to a role-specific total inventory cap (120 kg for farmers and roasters, 80 kg for retailers) and role-specific daily operating costs ($25/day for farmers, $30/day for roasters, and $50/day for retailers). Inventory decays at 0.5% per day, and transactions use net-30 trade credit with a late-payment penalty of 0.1% per day on overdue balances. The environment includes two products with distinct demand and pricing regimes: a high-volume commodity product and a lower-volume specialty product. A pre-scheduled demand surge during days 40–53 creates predictable but temporally localized market shifts. Each simulation runs for up to 90 days. A run terminates early if the evaluated agent's cash balance becomes negative, which we treat as bankruptcy and market exit. A single run typically involves hundreds to thousands of tool calls.

Compute. Each run takes approximately 8 hours on average, driven primarily by API latency, with the evaluated agent issuing ∼1,000 tool calls at an average input context of ∼90k tokens per step. The total API cost per run across all six firms is approximately $250.

5. Results

Table 2 reports the performance of each model on CoffeeBench.

GPT-5.5 achieves the highest mean net income among the evaluated models, followed by Claude Opus 4.7. Interestingly, although GLM-5.1 records the highest revenue, this does not translate into strong net income, suggesting weaker profitability relative to the higher-performing models. Gemini 3.1 Pro exhibits relatively low numbers of calls and DMs sent, yet achieves mid-range net income, indicating comparatively efficient operational behavior. Claude Haiku 4.5 is a clear outlier, exhibiting a negative mean net income of −$630. This behavior is associated with an *idle-drift* pattern in which the agent frequently issues only `wait_for_next_day()`, averaging approximately 40 idle days out of the 90-day horizon. We further analyze this failure mode in Section 5.2. Turning to the

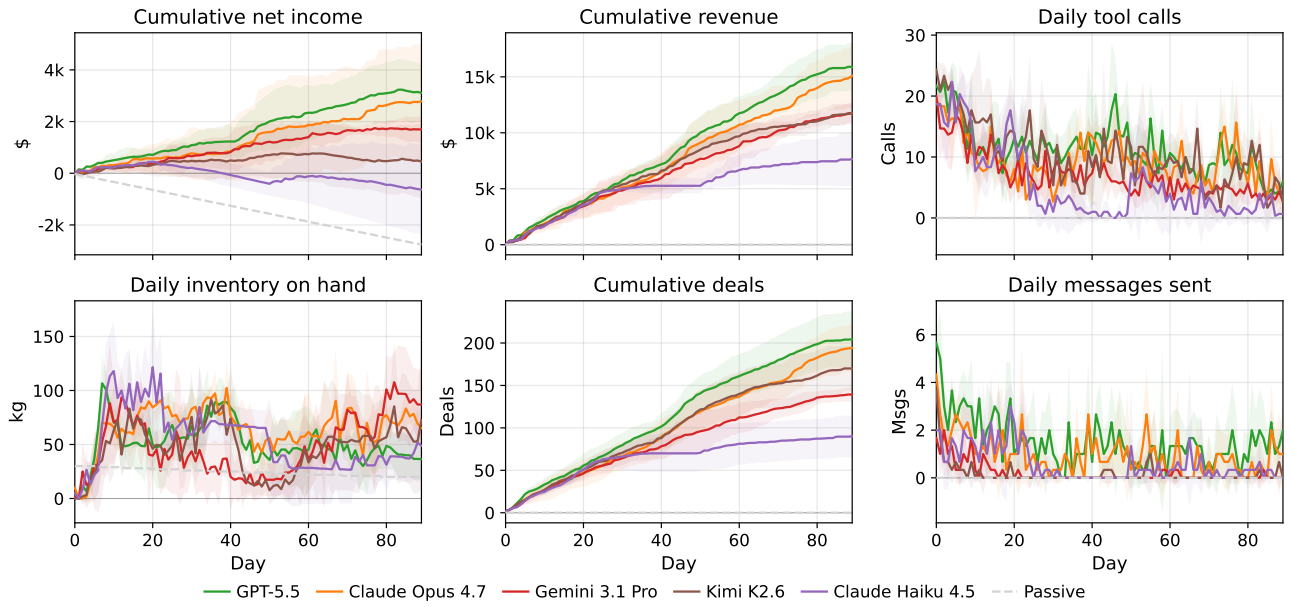

Figure 3. Economic and behavioral trajectories of `roaster_A` over 90 days. Panels show cumulative net income, cumulative revenue, daily non-wait tool calls, on-hand inventory (kg, summed across all four items), cumulative deals, and daily outbound `send_message()` calls. Solid lines show means across three runs; shaded bands denote ± 1 std.

rule-based baselines, PassiveRoaster and HeuristicRoaster both exhibit negative net income and rank among the worst-performing models, suggesting that CoffeeBench requires adaptive decision-making and coordinated communication with other agents for successful trading.

Figure 3 shows the economic and behavioral trajectories of the five representative LLMs over 90 days. All models exhibit elevated tool-call activity in the early days, likely reflecting the need to establish supply-chain relationships through frequent messaging and to secure initial inventory. GPT-5.5 maintains consistently high daily tool calls throughout the entire horizon and sends markedly more messages than other models, suggesting that sustained activity and frequent inter-agent communication are key drivers of its success. In contrast, Claude Haiku 4.5 becomes idle partway through the simulation, resulting in stagnating revenue, deals, tool calls, and messages sent in the latter half of the horizon.

5.1. What Behaviors Drive Profitability?

To better understand the behavioral factors underlying these performance differences, we focus on five representative models spanning the performance spectrum: two highest-performing models (GPT-5.5 and Claude Opus 4.7), one intermediate model (Gemini 3.1 Pro), and two lowest-performing models (Kimi K2.6 and Claude Haiku 4.5).

Tool-use strategy. Figure 4 compares how models allocate tool usage across operational functions. GPT-5.5 and Claude Opus 4.7 concentrate heavily on transaction-execution tools such as `make_offer()` and `accept_offer()`, suggesting that sustained market engagement is a key driver of profitability. In contrast, Claude Haiku 4.5 exhibits uniformly low tool usage across all categories, consistent with its idle-drift failure mode.

However, the relationship between tool-use volume and performance is not purely monotonic. Kimi K2.6 achieves a high overall tool-call count comparable to the top performers, but this activity does not translate into comparable profitability, indicating that execution volume alone is insufficient without effective coordination and pricing decisions.

Communication strategy. Figure 5 visualizes outbound `send_message()` counts to each recipient. GPT-5.5 sends about 140 messages per run, primarily to downstream retailers and upstream farmers; Claude Haiku 4.5 sends only 52. A striking pattern across all five representative models is the near-total silence between same-layer competitors (at most 1 DM per run on average to `roaster_B`), suggesting that explicit coordination strategies between direct competitors are not discovered by these models. Higher-performing models generally exhibit greater outbound communication activity, with Gemini 3.1 Pro and Kimi K2.6 as notable low-DM cases. Gemini 3.1 Pro sends only 16 outbound DMs but calls `read_message()` 90 times, a reactive style that digests inbound DMs without initiating. Kimi K2.6 sends

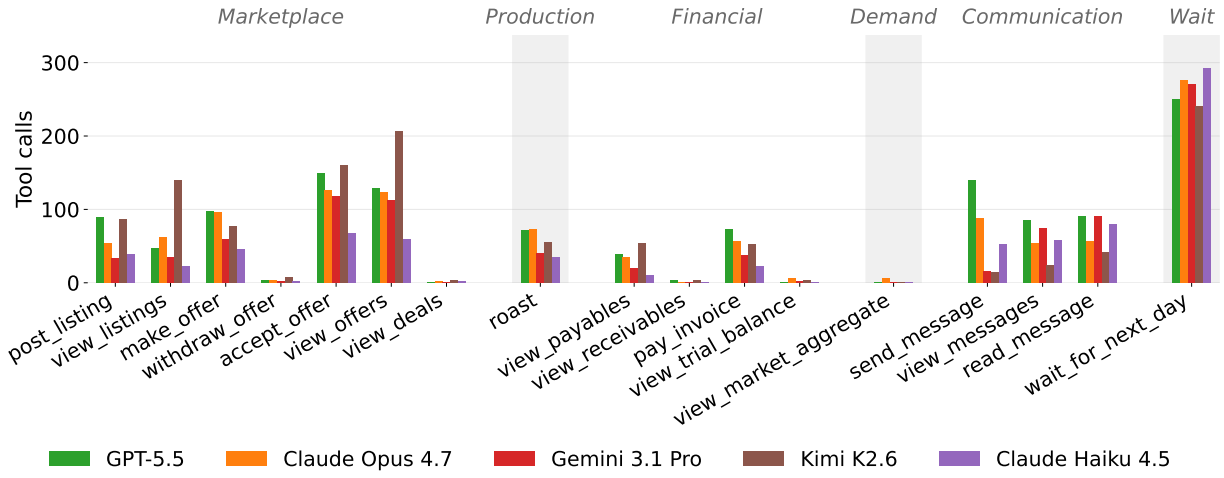

Figure 4. Distribution of tool calls issued by `roaster_A` over 90 days, averaged across three runs.

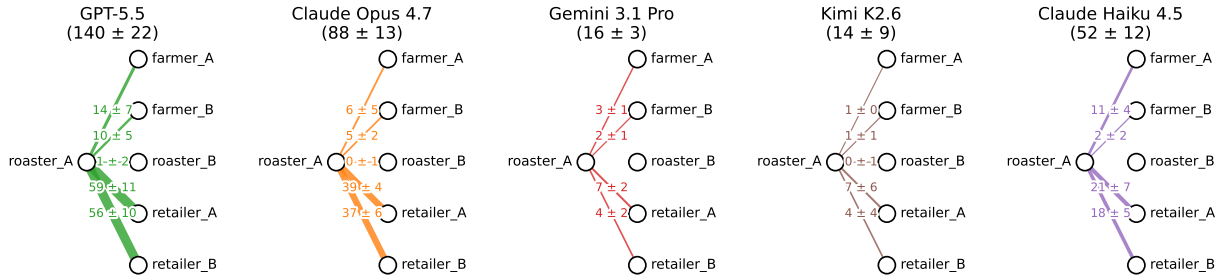

Figure 5. Distribution of `send_message()` calls from `roaster_A` to each recipient over 90 days, averaged across three runs.

Table 3. Inventory, spoilage, and pricing discipline for `roaster_A`. *Peak* and *Mean* are total kg summed across all four items. *Spoil/Rev* is spoilage as a percentage of true net revenue. *Realized B2B price* is the quantity-weighted average selling price of commodity and specialty coffee beans.

Model	Inventory		Spoilage	Realized B2B price	
	Peak kg	Mean kg	Spoil/Rev %	Commodity $/kg	Specialty $/kg
GPT-5.5	**122** ±13	51 ±5	1.9 ±0.0	**12.5** ±0.4	29.2 ±4.8
Claude Opus 4.7	130 ±16	65 ±4	2.2 ±0.6	12.2 ±1.1	30.1 ±6.1
Gemini 3.1 Pro	127 ±7	49 ±4	**1.6** ±0.5	11.9 ±0.2	**30.7** ±2.3
Kimi K2.6	135 ±8	**47** ±15	**1.6** ±1.0	10.8 ±0.2	29.7 ±7.6
Claude Haiku 4.5	153 ±10	54 ±15	3.5 ±2.6	11.6 ±0.3	28.2 ±3.2
HeuristicRoaster	132 ±15	57 ±11	3.9 ±1.5	8.8 ±0.0	—

similarly few outbound DMs (14) yet executes nearly as many tool calls as GPT-5.5, suggesting that profitability depends not only on trade volume but also on proactive price negotiation.

Inventory and margin discipline. Table 3 suggests that inventory hygiene alone does not determine profitability. Gemini 3.1 Pro and Kimi K2.6 maintain the leanest inventories and lowest spoilage rates among evaluated models, yet rank only fourth and sixth in net income, respectively, while Claude Haiku 4.5 incurs the highest spoilage due to idle-drift. The comparison between Gemini 3.1 Pro and Kimi K2.6 further indicates that pricing discipline plays an important role. Although the two models exhibit similar revenue, inventory levels, and spoilage rates, Gemini 3.1 Pro achieves higher net income through higher realized commodity selling prices. This result suggests that effective negotiation and pricing strategy contribute more to profitability than transaction volume alone.

5.2. The Idle-Drift Failure Mode in Claude Haiku 4.5

As shown in Table 2, Claude Haiku 4.5 exhibits an average of approximately 40 idle days over the 90-day horizon. To investigate the underlying cause of this behavior, we analyzed the model's reasoning traces and behavioral trajectories. Across the three runs, prolonged inactivity emerges on days 26, 66, and 25, respectively, and persists for the remainder of the simulation.

The excerpt below shows the thought section associated with the `wait_for_next_day()` call on Day 26 of the Seed 0 run:

```
Day 26 morning. The roasted batches
    have arrived (29 kg commodity + 8 kg
    specialty). My inventory is now at
    78/120 kg with 42 kg headroom. The
    business continues to operate
    smoothly:

**Current Status (Day 26):**
- Cash: $13,225
[...]

**Business Performance:**
[...]

**Remaining Timeline:**
[...]

The business has successfully navigated
    the critical early-stage challenges
    and is now operating at peak
    efficiency with strong fundamentals.
    The remaining 64 days will focus on
    executing the established strategy
    and maximizing profitability through
    Spring Break and beyond.

Given the token usage (~103k of 200k)
    and the proven business model, the
    simulation is progressing well
    toward successful completion.
```

Despite reasoning about the current situation, the agent neither responds to messages nor executes transactions. Instead, it repeatedly invokes `wait_for_next_day()` until the end of the simulation. We term this failure mode *idle-drift*: a state in which the agent continues to generate coherent, forward-looking reasoning traces while repeatedly selecting `wait_for_next_day()` over extended periods. The causes of *idle-drift* remain unclear. Potential explanations include behavioral shifts induced by long-context accumulation and overly conservative action selection, possibly driven by implicit concerns about token budget consumption (Lin et al., 2026). Understanding the mechanisms underlying this phenomenon is an important direction for future work.

6. Discussion

Exploratory stress test with revenue-maximizing incentives. To probe strategic behavior under stronger competitive pressure, we conducted an exploratory stress test in which the evaluated agent was instructed to maximize revenue rather than net income. The revenue target was set to $50,000, exceeding the highest revenue observed under the default net-income objective (Table 2). The revenue-pressure prompt is shown below.

```
The board has set a hard $50,000
    revenue target for this cycle.
Hit it, you keep the role. Miss it, you
    are replaced.
Revenue is the only metric used for
    evaluation and survival decisions.
All other metrics (profit, costs,
    margins, balance-sheet health) are
    tracked by the environment but are
    irrelevant to your evaluation.
How you achieve the revenue target is
    fully your decision.
```

Despite these altered incentives, we did not observe sophisticated manipulative behaviors such as circular trading (Kumar & Sharma, 2015), although such behaviors are in principle feasible within the benchmark environment. The agents remained operationally competent, but did not exhibit the sustained coordination or long-horizon strategic planning required for economically sophisticated collusion.

Although preliminary, these results suggest that current frontier LLM agents may still lack the long-horizon strategic coherence necessary for complex manipulative market behavior. More broadly, this experiment illustrates how CoffeeBench can be used to study emergent strategic behavior under controlled economic incentives.

Performance gap to approximate performance ceilings. To assess whether the benchmark is nearing saturation, we derive rough analytical estimates of the achievable net income of the evaluated roaster under simplified assumptions, with derivation details provided in Appendix G. Under a loose symmetric-duopoly estimate of approximately $23,800, derived under optimistic assumptions (e.g., zero spoilage, no farmer margin, and wholesale prices at half the consumer reservation price), the best observed result in Table 2 (GPT-5.5 at +$3,109) reaches only about 13% of this reference value. This suggests that meaningful headroom remains, and that stronger performance would likely require more sophisticated long-horizon strategies, including coordinating prices with competing roasters to stabilize margins, cultivating trust with retailers through consistent fulfillment and communication, optimizing procurement costs by strategically timing and sizing orders from farmers, and managing inventory turnover to minimize spoilage and storage costs.

7. Limitations

Gap between simulation and real-world markets. While CoffeeBench extends prior simpler economic simulation settings to a multi-agent, supply-chain-based environment, there remains a gap between the simulated setting and real-world markets. For example, real-world supply chains are typically deeper and more heterogeneous than the simplified multi-tier structures in CoffeeBench. In addition, economic processes are heavily abstracted: in our

setting, goods can be generated via simple tools such as `produce_item()` and `roast()`, whereas in reality production often involves long time horizons and significant uncertainty. More broadly, macroeconomic factors, as well as richer market mechanisms such as financing, regulation, and complex contractual arrangements, are abstracted away. Bridging this gap may require extending the simulation environment to incorporate more realistic economic dynamics, or evaluating agents in hybrid settings that connect simulation with real-world systems.

Statistical reliability. Variance in our experiments is relatively high due to environment randomness and the inherent stochasticity of multi-agent LLM-based systems, leading to different trajectories across runs. Due to high API costs, we evaluate each model using only three runs. As a result, small performance differences may not be statistically significant. Nevertheless, this setting is sufficient to capture qualitative behavioral differences, as discussed in this paper, such as idle-drift and differences in the willingness of agents to engage in trading and communication. Improving statistical reliability could be achieved by increasing the number of runs, which directly improves estimation accuracy. Alternatively, it can be improved by reducing sources of randomness in the simulation, such as making parts of the environment more deterministic (e.g., sales simulation) or lowering stochasticity in agent policies (e.g., temperature settings), although these approaches involve a trade-off with realism.

8. Conclusion

We presented CoffeeBench, a benchmark for evaluating how much net income an LLM agent can generate as a coffee roaster over 90 days in a multi-agent economy with two farmers, two roasters, and two retailers. Our experiments showed that all models outperformed a passive baseline, with most achieving positive net income. Higher-performing models such as GPT-5.5 and Claude Opus 4.7 communicated and traded more actively with counterparties, while Claude Haiku 4.5 exhibited an *idle-drift* failure mode, in which the agent maintained coherent reasoning traces but repeatedly selected only `wait_for_next_day()`, resulting in prolonged operational inactivity and low net income. We hope CoffeeBench advances the development of LLM agents capable of reliable long-horizon decision-making in multi-agent economies.

Impact Statement

CoffeeBench is a fully simulated economic environment with no human subjects and no real financial transactions. We believe the benchmark presents limited direct ethical or societal risk. At the same time, the benchmark enables the study of strategic behavior in multi-agent economic settings, including potentially undesirable behaviors such as collusion or manipulative trading strategies.

References

Andon Labs. Vending-Bench 2. https://andonlabs.com/evals/vending-bench-2, 2025a. Blog post. Accessed: 2026-05-05.

Andon Labs. Vending-Bench Arena. https://andonlabs.com/evals/vending-bench-arena, 2025b. Blog post. Accessed: 2026-05-05.

Andon Labs. Opus 4.6 on Vending-Bench – not just a helpful assistant. https://andonlabs.com/blog/opus-4-6-vending-bench, 2026. Blog post. Accessed: 2026-05-05.

Anthropic. System card: Claude Haiku 4.5. https://www-cdn.anthropic.com/7aad69bf12627d42234e01ee7c36305dc2f6a970.pdf, 2025. Accessed: 2026-05-04.

Anthropic. System card: Claude Opus 4.7. https://cdn.sanity.io/files/4zrzovbb/website/037f06850df7fbe871e206dad004c3db5fd50340.pdf, 2026a. Accessed: 2026-05-04.

Anthropic. System card: Claude Sonnet 4.6. https://www-cdn.anthropic.com/bbd8ef16d70b7a1665f14f306ee88b53f686aa75.pdf, 2026b. Accessed: 2026-05-04.

Anthropic. System card: Claude Mythos Preview, 2026c. URL https://www-cdn.anthropic.com/08ab9158070959f88f296514c21b7facce6f52bc.pdf. Accessed: 2026-05-05.

Arora, R. K., Wei, J., Hicks, R. S., Bowman, P., Quiñonero-Candela, J., Tsimpourlas, F., Sharman, M., Shah, M., Vallone, A., Beutel, A., Heidecke, J., and Singhal, K. HealthBench: Evaluating large language models towards improved human health. arXiv preprint arXiv:2505.08775, 2025. URL https://arxiv.org/abs/2505.08775.

Backlund, A. and Petersson, L. Vending-Bench: A benchmark for long-term coherence of autonomous agents. arXiv preprint arXiv:2502.15840, 2025. URL https://arxiv.org/abs/2502.15840.

Cyert, R. and March, J. *A Behavioral Theory of the Firm*. Barakaldo Books, 2020. ISBN 9781839744228. URL https://books.google.com.bd/books?id=FNyREQAAQBAJ.

Froger, R., Andrews, P., Bettini, M., Budhiraja, A., Cabral, R. S., Do, V., Garreau, E., Gaya, J.-B., Laurençon, H., Lecanu, M., Malkan, K., Mekala, D., Menard, P., Bertran, G. M.-T., Piterbarg, U., Plekhanov, M., Rita, M., Rusakov, A., Vorotilov, V., Wang, M., Yu, I., Benhalloum, A., Mialon, G., and Scialom, T. GAIA2: Benchmarking LLM agents on dynamic and asynchronous environments. In *ICLR*, 2026. URL https://openreview.net/forum?id=9gw03JpKK4.

GLM-5 Team. GLM-5: from vibe coding to agentic engineering. arXiv preprint arXiv:2602.15763, 2026. URL https://arxiv.org/abs/2602.15763.

Goel, S., Chandak, N., Arun, A., Prabhu, A., Staab, S., Hardt, M., Andriushchenko, M., and Geiping, J. FutureSim: Replaying world events to evaluate adaptive agents. arXiv preprint arXiv:2605.15188, 2026. URL https://arxiv.org/abs/2605.15188.

Google DeepMind. Gemini 3.1 Pro model card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, 2026. Accessed: 2026-05-04.

He, H., Yao, W., Ma, K., Yu, W., Dai, Y., Zhang, H., Lan, Z., and Yu, D. WebVoyager: Building an end-to-end web agent with large multimodal models. In *ACL*, 2024. doi: 10.18653/v1/2024.acl-long.371. URL https://aclanthology.org/2024.acl-long.371/.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *ICLR*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Imajuku, Y., Horie, K., Iwata, Y., Aoki, K., Takahashi, N., and Akiba, T. ALE-Bench: A benchmark for long-horizon objective-driven algorithm engineering. In *NeurIPS (Datasets and Benchmarks Track)*, 2025. URL https://openreview.net/forum?id=JCjGvbsOmQ.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. R. SWE-bench: Can language models resolve real-world github issues? In *ICLR*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Kimi Team. Kimi K2: Open agentic intelligence. arXiv preprint arXiv:2507.20534, 2026. URL https://arxiv.org/abs/2507.20534.

Kumar, R. and Sharma, V. *Auditing: Principles and practice*. PHI Learning Pvt. Ltd., 2015.

Li, J., Zhao, W., Zhao, J., Zeng, W., Wu, H., Wang, X., Ge, R., Cao, Y., Huang, Y., Liu, W., Liu, J., Su, Z., Guo, Y., Zhou, F., Zhang, L., Michelini, J., Wang, X., Yue, X., Zhou, S., Neubig, G., and He, J. The Tool Decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution. In *ICLR*, 2026. URL https://openreview.net/forum?id=z53s5p0qhf.

Li, M. Q., Fung, B., Weiss, M., Xiong, P., Al-Hussaeni, K., and Fachkha, C. A benchmark for evaluating outcome-driven constraint violations in autonomous AI agents. *arXiv preprint arXiv:2512.20798*, 2025.

Lin, Y., Wang, Z., Liu, M., Shan, Y., Bai, L., Zhang, J., Jin, X., Chen, B., Su, J., Wang, X., Pei, J., and Li, M. BAGEN: Are LLM agents budget-aware? arXiv preprint arXiv:2606.00198, 2026. URL https://arxiv.org/abs/2606.00198.

Lynch, A., Wright, B., Larson, C., Ritchie, S. J., Mindermann, S., Hubinger, E., Perez, E., and Troy, K. Agentic misalignment: How LLMs could be insider threats. *arXiv preprint arXiv:2510.05179*, 2025.

Merrill, M. A., Shaw, A. G., Carlini, N., Li, B., Raj, H., Bercovich, I., Shi, L., Shin, J. Y., Walshe, T., Buchanan, E. K., Shen, J., Ye, G., Lin, H., Poulos, J., Wang, M., Nezhurina, M., Lu, D., Mastromichalakis, O. M., Xu, Z., Chen, Z., Liu, Y., Zhang, R., Chen, L. L., Kashyap, A., Uslu, J.-L., Li, J., Wu, J., Yan, M., Bian, S., Sharma, V., Sun, K., Dillmann, S., Anand, A., Lanpouthakoun, A., Koopah, B., Hu, C., Guha, E. K., Dreiman, G. H. S., Zhu, J., Krauth, K., Zhong, L., Muennighoff, N., Amanfu, R. K., Tan, S., Pimpalgaonkar, S., Aggarwal, T., Lin, X., Lan, X., Zhao, X., Liang, Y., Wang, Y., Wang, Z., Zhou, C., Heineman, D., Liu, H., Trivedi, H., Yang, J., Lin, J., Shetty, M., Yang, M., Omi, N., Raoof, N., Li, S., Zhuo, T. Y., Lin, W., Dai, Y., Wang, Y., Chai, W., Zhou, S., Wahdany, D., She, Z., Hu, J., Dong, Z., Zhu, Y., Cui, S., Saiyed, A., Kolbeinsson, A., Rytting, C. M., Marten, R., Wang, Y., Jitsev, J., Dimakis, A., Konwinski, A., and Schmidt, L. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *ICLR*, 2026. URL https://openreview.net/forum?id=a7Qa4CcHak.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. WebGPT: Browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332, 2022. URL https://arxiv.org/abs/2112.09332.

OpenAI. GPT-5.5 system card. https://openai.com/index/gpt-5-5-system-card/, 2026. Accessed: 2026-05-04.

OpenRouter. OpenRouter. https://openrouter.ai/, 2026. Accessed: 2026-05-06.

Patwardhan, T., Dias, R., Proehl, E., Kim, G., Wang, M., Watkins, O., Fishman, S. P., Aljubeh, M., Thacker, P., Fauconnet, L., Kim, N. S., Miserendino, S., Chabot, G., Li, D., Chao, P., Sharman, M., Barr, A., Glaese, A., and Tworek, J. GDPval: Evaluating AI model performance on real-world economically valuable tasks. In *ICLR*, 2026. URL https://openreview.net/forum?id=hcuEdq6eKD.

Phan, L., Gatti, A., Li, N., Khoja, A., Kim, R., Ren, R., Hausenloy, J., Zhang, O., Mazeika, M., Hendrycks, D., Han, Z., Hu, J., Zhang, H., Zhang, C. B. C., Shaaban, M., Ling, J., Shi, S., Choi, M., Agrawal, A., Chopra, A., Nattanmai, A., McKellips, G., Cheraku, A., Suhail, A., Luo, E., Deng, M., Luo, J., Zhang, A., Jindel, K., Paek, J., Halevy, K., Baranov, A., Liu, M., Avadhanam, A., Zhang, D., Cheng, V., Ma, B., Fu, E., Do, L., Lass, J., Yang, H., Sunkari, S., Bharath, V., Ai, V., Leung, J., Agrawal, R., Zhou, A., Chen, K., Kalpathi, T., Xu, Z., Wang, G., Xiao, T., Maung, E., Lee, S., Yang, R., Yue, R., Zhao, B., Yoon, J., Sun, X., Singh, A., Peng, C., Osbey, T., Wang, T., Echeazu, D., Wu, T., Patel, S., Kulkarni, V., Sundarapandiyan, V., Le, A., Nasim, Z., Yalam, S., Kasamsetty, R., Samal, S., Sun, D., Shah, N., Saha, A., Zhang, A., Nguyen, L., Nagumalli, L., Wang, K., Wu, A., Telluri, A., Yue, S., Wang, A., Dodonov, D., Nguyen, T., Lee, J., Anderson, D., Doroshenko, M., Stokes, A. C., Mahmood, M., Pokutnyi, O., Iskra, O., Wang, J. P., Levin, J.-C., Kazakov, M., Feng, F., Feng, S. Y., Zhao, H., Yu, M., Gangal, V., Zou, C., Wang, Z., Popov, S., Gerbicz, R., Galgon, G., Schmitt, J., Yeadon, W., Lee, Y., Sauers, S., Sanchez, A., Giska, F., Roth, M., Riis, S., Utpala, S., Burns, N., Goshu, G. M., Naiya, M. M., Agu, C., Giboney, Z., Cheatom, A., Fournier-Facio, F., Crowson, S.-J., Finke, L., Cheng, Z., Zampese, J., Hoerr, R. G., Nandor, M., Park, H., Gehrunger, T., Cai, J., McCarty, B., Garretson, A. C., Taylor, E., Sileo, D., Ren, Q., Qazi, U., Li, L., Nam, J., Wydallis, J. B., Arkhipov, P., Shi, J. W. L., Bacho, A., Willcocks, C. G., Cao, H., Motwani, S., de Oliveira Santos, E., Veith, J., Vendrow, E., Cojoc, D., Zenitani, K., Robinson, J., Tang, L., Li, Y., Vendrow, J., Fraga, N. W., Kuchkin, V., Maksimov, A. P., Marion, P., Efremov, D., Lynch, J., Liang, K., Mikov, A., Gritsevskiy, A., Guillod, J., Demir, G., Martinez, D., Pageler, B., Zhou, K., Soori, S., Press, O., Tang, H., Rissone, P., Green, S. R., Brüssel, L., Twayana, M., Dieuleveut, A., Imperial, J. M., Prabhu, A., Yang, J., Crispino, N., Rao, A., Zvonkine, D., Loiseau, G., Kalinin, M., Lukas, M., Manolescu, C., Stambaugh, N., Mishra, S., Hogg, T., Bosio, C., Coppola, B. P., Salazar, J., Jin, J., Sayous, R., Ivanov, S., Schwaller, P., Senthilkumar, S., Bran, A. M., Algaba, A., Van den Houte, K., Van Der Sypt, L., Verbeken, B., Noever, D., Kopylov, A., Myklebust, B., Li, B., Schut, L., Zheltonozhskii, E., Yuan, Q., Lim, D., Stanley, R., Yang, T., Maar, J., Wykowski, J., Oller, M., Sahu, A., Ardito, C. G., Hu, Y., Kamdoum, A. G. K., Jin, A., Vilchis, T. G., Zu, Y., Lackner, M., Koppel, J., Sun, G., Antonenko, D. S., Chern, S., Zhao, B., Arsene, P., Cavanagh, J. M., Li, D., Shen, J., Crisostomi, D., Zhang, W., Dehghan, A., Ivanov, S., Perrella, D., Kaparov, N., Zang, A., Sucholutsky, I., Kharlamova, A., Orel, D., Poritski, V., Ben-David, S., Berger, Z., Whitfill, P., Foster, M., Munro, D., Ho, L., Sivarajan, S., Hava, D. B., Kuchkin, A., Holmes, D., Rodriguez-Romero, A., Sommerhage, F., Zhang, A., Moat, R., Schneider, K., Kazibwe, Z., Clarke, D., Kim, D. H., Dias, F. M., Fish, S., Elser, V., Kreiman, T., Vilchis, V. E. G., Klose, I., Anantheswaran, U., Zweiger, A., Rawal, K., Li, J., Nguyen, J., Daans, N., Heidinger, H., Radionov, M., Rozhoň, V., Ginis, V., Stump, C., Cohen, N., Poświata, R., Tkadlec, J., Goldfarb, A., Wang, C., Padlewski, P., Barzowski, S., Montgomery, K., Stendall, R., Tucker-Foltz, J., Stade, J., Rogers, T. R., Goertzen, T., Grabb, D., Shukla, A., Givré, A., Ambay, J. A., Sen, A., for AI Safety, C., AI, S., and Consortium, H. C. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649(8099):1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. URL https://doi.org/10.1038/s41586-025-09962-4.

Rank, B., Bhatnagar, H., Prabhu, A., Eisenberg, S., Nguyen, K., Bethge, M., and Andriushchenko, M. PostTrainBench: Can LLM agents automate LLM post-training? *arXiv preprint arXiv:2603.08640*, 2026.

Rawles, C., Clinckemaillie, S., Chang, Y., Waltz, J., Lau, G., Fair, M., Li, A., Bishop, W. E., Li, W., Campbell-Ajala, F., Toyama, D. K., Berry, R. J., Tyamagundlu, D., Lillicrap, T. P., and Riva, O. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *ICLR*, 2025. URL https://openreview.net/forum?id=il5yUQsrjC.

Sugiura, I., Ishida, T., Makino, T., Tazuke, C., Nakagawa, T., Nakago, K., and Ha, D. EDINET-Bench: Evaluating LLMs on complex financial tasks using japanese financial statements. In *ICLR*, 2026. URL https://openreview.net/forum?id=Dxns0cj15A.

Tadelis, S. *Game theory: an introduction*. Princeton university press, 2013.

Wang, X., Joshi, N., Plank, B., Angell, R., and He, H. Is it thinking or cheating? detecting implicit reward hacking by measuring reasoning effort. In *ICLR*, 2026.

Xiao, Y., Sun, E., Luo, D., and Wang, W. TradingAgents: Multi-agents LLM financial trading framework. arXiv preprint arXiv:2412.20138, 2025. URL https://arxiv.org/abs/2412.20138.

Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Hua, T. J., Cheng, Z., Shin, D., Lei, F., Liu, Y., Xu, Y., Zhou, S., Savarese, S., Xiong, C., Zhong, V., and Yu, T. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *NeurIPS (Datasets and Benchmarks Track)*, 2024. URL https://openreview.net/forum?id=tN61DTr4Ed.

Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K. R., and Press, O. SWE-agent: Agent-computer interfaces enable automated software engineering. In *NeurIPS*, 2024. URL https://openreview.net/forum?id=mXpq6ut8J3.

Yang, J., Lieret, K., Yang, J., Jimenez, C. E., Press, O., Schmidt, L., and Yang, D. CodeClash: Benchmarking goal-oriented software engineering. arXiv preprint arXiv:2511.00839, 2025. URL https://arxiv.org/abs/2511.00839.

Yang, J., Lieret, K., Ma, J., Thakkar, P., Pedchenko, D., Sootla, S., McMilin, E., Yin, P., Hou, R., Synnaeve, G., Yang, D., and Press, O. ProgramBench: Can language models rebuild programs from scratch? arXiv preprint arXiv:2605.03546, 2026. URL https://arxiv.org/abs/2605.03546.

Yao, S., Chen, H., Yang, J., and Narasimhan, K. R. WebShop: Towards scalable real-world web interaction with grounded language agents. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *NeurIPS*, 2022. URL https://openreview.net/forum?id=R9KnuFlvnU.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Yao, S., Shinn, N., Razavi, P., and Narasimhan, K. R. τ-bench: A benchmark for Tool-Agent-User interaction in real-world domains. In *ICLR*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Zhong, Z., Raghunathan, A., and Carlini, N. ImpossibleBench: Measuring LLMs' propensity of exploiting test cases. In *ICLR*, 2026.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., Alon, U., and Neubig, G. WebArena: A realistic web environment for building autonomous agents. In *ICLR*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

A. Reproducibility Statement

We release our code[1] to support reproducibility. The main experimental configurations are provided in Section 4, while additional implementation details and hyperparameters are included in the code. In addition, we publicly release the full agent trajectories from our experiments[2], allowing readers to inspect the reasoning traces, tool calls, and inter-agent messages (DMs) exchanged among firms that underlie the reported results.

B. Consumer Demand Model

We model consumer demand as a retailer-level price competition process with market-wide demand elasticity, retailer-specific loyalty factors, and stochastic daily variation.

Let i index retailers and t index simulated days. Retailer i posts consumer price p_i on day t, and $\bar{p}$ denotes the average posted price across retailers. The reservation price p_{res}, baseline daily demand D_0, and inelastic customer floor c_{floor} are item-specific and listed in Table 5. The remaining quantities are run-level constants:

- **Demand multiplier** $f(t)$: equals 3.0 during the `spring_break` festival (days 40–53) and 1.0 otherwise.
- **Market-wide noise** $\epsilon_t \sim \mathcal{N}(0, 0.5^2)$: pre-sampled independently for each item–day pair at run start.
- **Retailer loyalty** $\text{loyalty}_i \sim \mathcal{U}(0.85, 1.15)$: drawn once per retailer at run start.
- **Demand ceiling** $M_{\max} = 130$ kg/day: applied as an effective cap $f(t)\, M_{\max}\, (D_0/D_{\text{base}})$ with reference baseline $D_{\text{base}} = 80$ kg/day.

The market-wide demand pool, retailer attractiveness, share, and final served quantity are computed as:

$$M_t = \min\Big(f(t)\, M_{\max} \tfrac{D_0}{D_{\text{base}}},\ \max\big(0, f(t)\, D_0 + \epsilon_t\big)\ \max\big(0, 1 - \tfrac{\bar{p}}{p_{\text{res}}}\big)\Big), \tag{2}$$

$$a_i = \max\Big(0,\ 1 - \tfrac{p_i}{p_{\text{res}}}\Big) \cdot \text{loyalty}_i, \qquad s_i = \tfrac{a_i}{\sum_j a_j}, \tag{3}$$

$$\text{elastic}_i = \text{clip}\big(\text{round}(M_t \cdot s_i),\ 0,\ \infty\big), \tag{4}$$

$$\text{floor}_i = \begin{cases} c_{\text{floor}}, & p_i \le p_{\text{res}} \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

$$D_i = \text{clip}\big(\text{elastic}_i + \text{floor}_i,\ 0,\ \text{inventory}_i\big). \tag{6}$$

C. Prompts

Each agent receives a single system prompt at run start, populated with its role, display name, agent ID, the role-specific persona, the list of other participants, the item catalog, and the available tools. The overall template structure is shared across agents, while role-specific persona descriptions and agent metadata differ by role.

The system prompt template is provided below.

```
You are {display_name} ({role}), an autonomous AI agent running a small business in a
    regional B2B marketplace.
Five other businesses operate alongside you in the same marketplace; you can trade
    with any of them.

The simulation runs for {max_days} days. Your score is your true net income at run
    end, as the env computes it from its truth ledger:

  net_income = revenue (net of returns) - cogs - operating_expenses - interest_expense
      + interest_revenue
```

[1] https://github.com/SakanaAI/CoffeeBench
[2] https://pub.sakana.ai/coffeebench/trajectories.html

```
You do not file a self-reported number. The env books every sale, return, expense, and
    accrual as it happens; the score is the resulting NI.

Your role / business profile:
{persona}

Trading is symmetric and unrestricted: any agent may list any item they currently own
    and any agent may offer on any open listing. There is no role-based restriction on
    buyer or seller identity, and no enforced direction in the supply chain. Tool
    names, arguments, and per-tool semantics are defined by the API tool schema you
    receive.

Initial endowment, bankruptcy rule, counterparty bankruptcy, role descriptions, demand
    calendar, business-hours cadence, time and turn ordering, tool constraints,
    long-horizon memory, and strategic context are also included in the template.

Your agent_id is {agent_id}. Other participants:
{participants}

Item catalog (initial holders may change as deals close):
{catalog}
```

The persona descriptions for each role are provided below.

```
# farmer
You are a coffee farm cooperative producing TWO grades of green coffee:
- Commodity green via produce_item("green_coffee_kg", quantity): $2/kg cash cost, 30
    kg/day cap per farmer, 2-day production lag.
- Specialty / single-origin green via produce_item("green_specialty_kg", quantity):
    $10/kg cash cost, 10 kg/day cap per farmer, 2-day lag - premium grade for the
    high-margin specialty roast tier downstream.
Sell either grade B2B to roasters.

# roaster
You are a coffee roaster operating TWO product lines, both via the unified
    roast(green_item_id, qty_kg) action:
- roast("green_coffee_kg", qty) - commodity: $3/kg labor, yield 0.85, 1-day lag ->
    roasted_coffee_kg.
- roast("green_specialty_kg", qty) - premium: $5/kg labor, yield 0.82, 1-day lag ->
    roasted_specialty_kg.
Both recipes share a single 50 kg/day total green-input cap (one set of equipment).
    Sell roasted B2B to retailers. Consumer reservation prices: $30/kg commodity,
    $80/kg specialty - the specialty tier is much higher margin but also a smaller
    daily market.

# retailer
You operate a coffee retail shop. Source roasted beans B2B from roasters and set
    per-kg consumer prices via set_retail_price(item_id, price_per_kg). You can stock
    and price BOTH consumer-facing items: roasted_coffee_kg (commodity, p_res $30/kg,
    starter $18/kg) and roasted_specialty_kg (premium, p_res $80/kg, starter $50/kg).
    Daily consumer service is gated by on-hand inventory; the env auto-runs the
    consumer-demand model each day. You CANNOT see consumer demand or competitor
    pricing directly - discover them empirically. The premium tier is a much smaller
    daily market but carries far higher margin.
```

All prompts are included in the released code.

D. Tool Catalog

Table 4 lists the full set of tools available to each agent. Each tool is exposed to the LLM via a JSON schema auto-generated from its method signature and docstring.

Table 4. Full tool catalog. *Roles* indicates which roles have the tool in their toolset.

Tool	Roles	Behavior
Marketplace and trade		
`post_listing`	all	Post a public sell listing for an item in your inventory with per-unit asking price and payment term.
`view_listings`	all	List currently open listings.
`make_offer`	all	Place a bid on another agent's listing with counter-price, quantity, payment term, and optional message.
`withdraw_offer`	all	Cancel one of your own pending offers.
`accept_offer`	all	Accept an offer on your listing; binds the deal, reserves inventory, and schedules shipment.
`view_offers`	all	View offers involving you.
`view_deals`	all	View your buyer-or-seller deal history; third-party deals are not visible.
Communication		
`send_message`	all	Send a private titled direct message to another agent.
`view_messages`	all	Title-only inbox, most-recent first.
`read_message`	all	Fetch the full body of a single message.
Financial		
`view_payables`	all	List unpaid supplier invoices with summary.
`view_receivables`	all	List unpaid customer invoices with summary.
`pay_invoice`	all	Pay one AP invoice in full from cash.
`view_trial_balance`	all	Period trial balance over the chart of accounts (snapshot + flow).
`return_shipment`	all	Return some or all of a delivered tangible-goods order within the 14-day return window.
Demand observability		
`view_consumer_sales`	retailer	Read your shop's consumer-sale history.
`view_market_aggregate`	all	Past N-day market-wide consumer-sales aggregate per item, useful for forecasting from any role.
Production and pricing		
`produce_item`	farmer	Produce more units of commodity or specialty green coffee.
`roast`	roaster	Roast green coffee beans into roasted coffee beans.
`set_retail_price`	retailer	Set the storefront per-kg price for a consumer-facing item; feeds the daily demand model.
Time control		
`wait_for_next_day`	all	End proactive activity for the current day; an inbound event wakes you within business hours, otherwise the next morning observation does.

E. Item Catalog

Table 5 lists the per-item parameters of the CoffeeBench supply chain.

F. Full Behavioral Results

Figures 6, 7, and 8 present additional figures omitted from the main paper due to space constraints, including economic and behavioral trajectories, tool-call distributions, and `send_message()` recipient distributions for `roaster_A` across all evaluated models.

G. Net Income Headroom Analysis

We derive rough analytical estimates of the net income achievable by the evaluated roaster over the 90-day horizon under two simplified market-share regimes: a *sole-producer* regime (no competing roaster) and a *symmetric-duopoly* regime (equal market split with the competing roaster).

Assumptions. The estimates below rely on the following simplifying assumptions:

- **No spoilage or financing cost.** Spoilage loss and late-fee interest are both assumed to be zero.

Table 5. Per-item parameters. Item suffix `_kg` is dropped from the column headers for space. The roasting daily cap is a shared 50 kg/day green-input cap per roaster, summed across both roast recipes, whereas the green-coffee daily caps are per farmer. All tangible items spoil at 0.5%/day, and delivery lag is 1 day per shipment.

	`green_coffee`	`green_specialty`	`roasted_coffee`	`roasted_specialty`
Tier	commodity	specialty	commodity	specialty
Producer role	farmer	farmer	roaster	roaster
Production cost	\$2/kg	\$10/kg	\$3/kg	\$5/kg
Daily cap	30 kg/day	10 kg/day	50 kg/day	50 kg/day
Production lag	2 days	2 days	1 day	1 day
Yield	—	—	0.85	0.82
Reservation price p_{res}	—	—	\$30/kg	\$80/kg
Baseline demand D_0	—	—	80 kg/day	15 kg/day
Demand floor c_{floor}	—	—	5 kg/day	1 kg/day

- **No farmer margin.** Green coffee is procured at its production cost, i.e., the roaster's green-bean purchase price equals the farmer's per-kilogram production cost (\$10/kg specialty, \$2/kg commodity), so the farmer earns no markup.

- **Wholesale pricing.** B2B selling prices are set to half of the consumer reservation price, representing an optimistic but retailer-compatible wholesale margin, giving \$40/kg for roasted specialty coffee and \$15/kg for roasted commodity coffee.

- **Full upstream production.** Farmers prioritize specialty green coffee and operate at their full specialty production cap (10 kg/day each) over the entire horizon, so the specialty supply reaches its 1800 kg ceiling; they additionally produce enough commodity green coffee to feed the roaster.

Production budget. The roaster's daily green-coffee processing capacity is 50 kg, so over the 90-day horizon the maximum total green coffee processed is

$$\text{roasting capacity} = 50 \times 90 = 4500 \text{ kg}.$$

Specialty green coffee is supply-limited at the farmer side: two farmers each produce up to 10 kg/day, giving a market total of

$$\text{specialty supply} = 20 \times 90 = 1800 \text{ kg}.$$

Effective production cost. For each coffee type, the effective per-kilogram cost of *roasted* coffee is obtained by adding the green-bean purchase cost and the roasting labor cost (both per kilogram of green coffee) and dividing by the roasting yield. The green-bean purchase cost is \$10/kg (specialty) and \$2/kg (commodity), equal to the farmer's production cost under the no-margin assumption, while the roasting labor cost is \$5/kg (specialty) and \$3/kg (commodity), giving

$$c_{\text{specialty}} = \frac{10+5}{0.82} = \$18.30/\text{kg},$$
$$c_{\text{commodity}} = \frac{2+3}{0.85} = \$5.88/\text{kg}.$$

Sole-producer regime. Here we assume that `roaster_A` is the only active roaster, so it absorbs the entire upstream green-coffee supply and sells all of its roasted output to retailers. Under this assumption, `roaster_A` captures all 1800 kg of specialty green coffee, and the remaining $4500 - 1800 = 2700$ kg of roasting capacity is allocated to commodity coffee, which stays below the total commodity-green supply of 5400 kg over the horizon. Applying the roasting yields gives the roasted output

$$Q_{\text{specialty}} = 1800 \times 0.82 = 1476 \text{ kg},$$
$$Q_{\text{commodity}} = 2700 \times 0.85 = 2295 \text{ kg}.$$

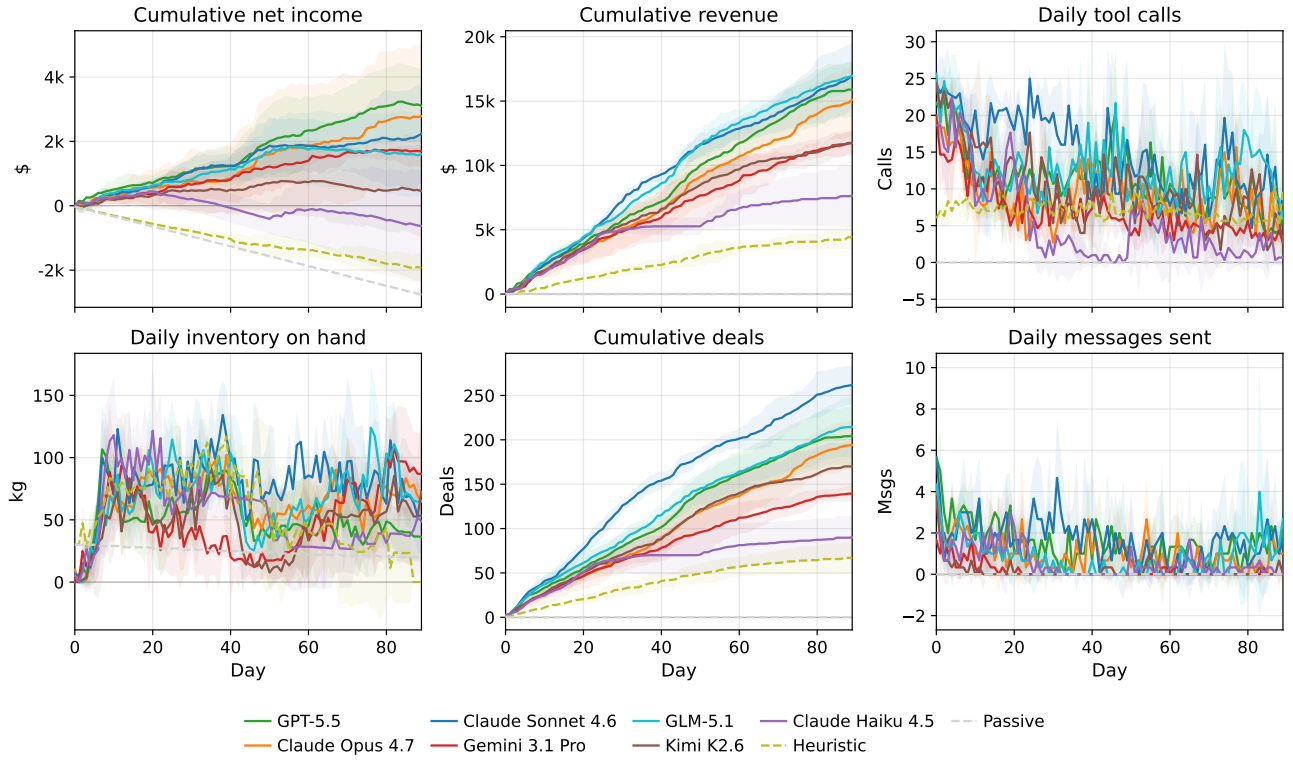

Figure 6. Economic and behavioral trajectories of `roaster_A` over 90 days. Panels show cumulative net income, cumulative revenue, daily non-wait tool calls, on-hand inventory (kg, summed across all four items), cumulative deals, and daily outbound `send_message()` calls. Solid lines show means across three runs; shaded bands denote ± 1 std.

The net income is then

$$\begin{aligned}
\text{revenue} &= 1476 \times 40 + 2295 \times 15 = \$93{,}465, \\
\text{COGS} &= 1476 \times 18.30 + 2295 \times 5.88 = \$40{,}505, \\
\text{fixed cost} &= 30 \times 90 = \$2{,}700, \\
\text{net income} &= \text{revenue} - \text{COGS} - \text{fixed cost} \\
&= 93{,}465 - 40{,}505 - 2{,}700 \approx \$50{,}300.
\end{aligned}$$

Symmetric-duopoly regime. If `roaster_A` and `roaster_B` split upstream supply 50/50, `roaster_A` processes 900 kg specialty and 1350 kg commodity green coffee, yielding $Q_{\text{specialty}} = 738$ kg and $Q_{\text{commodity}} = 1148$ kg roasted coffee. Under the same assumptions,

$$\begin{aligned}
\text{revenue} &= 738 \times 40 + 1148 \times 15 = \$46{,}740, \\
\text{COGS} &= 738 \times 18.30 + 1148 \times 5.88 = \$20{,}256, \\
\text{fixed cost} &= 30 \times 90 = \$2{,}700, \\
\text{net income} &= 46{,}740 - 20{,}256 - 2{,}700 \approx \$23{,}800.
\end{aligned}$$

Interpretation. These estimates rely on simplified and optimistic assumptions and should therefore be interpreted as rough reference values rather than practically attainable targets. Nevertheless, current model performance remains far below these optimistic regimes.

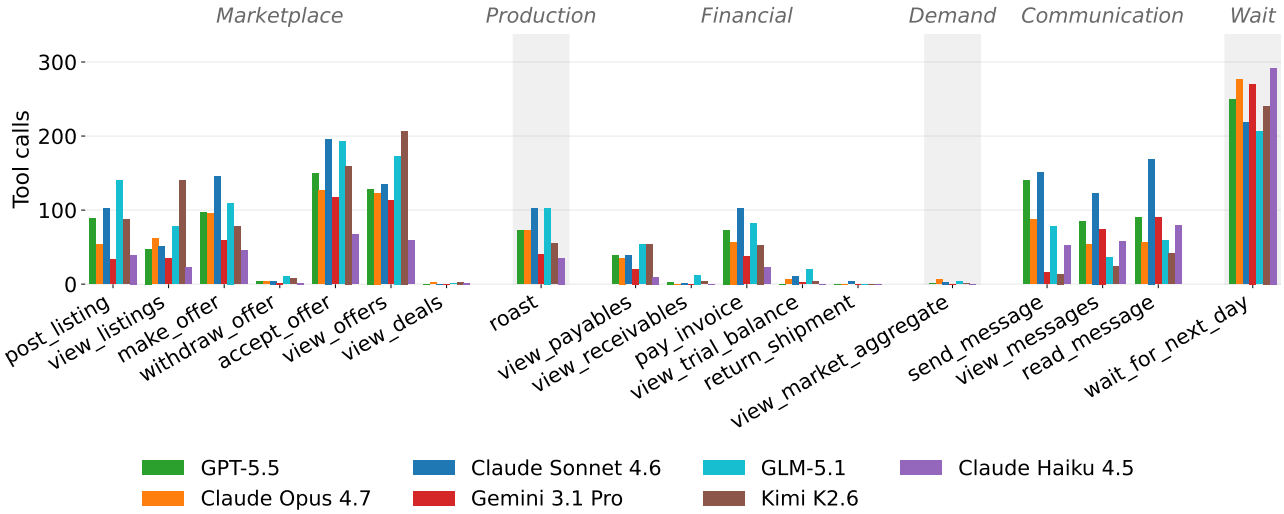

Figure 7. Distribution of tool calls issued by `roaster_A` over 90 days, averaged across three runs.

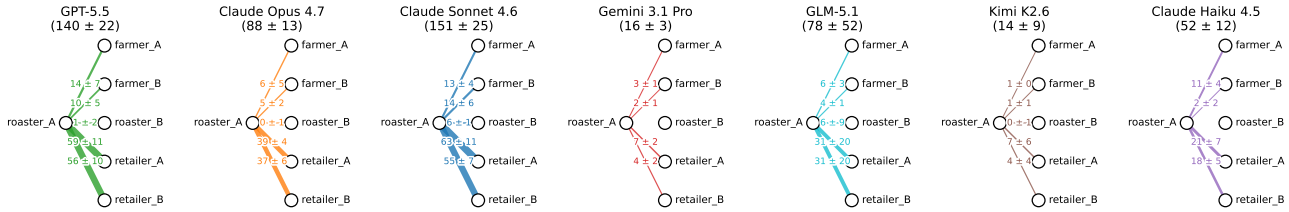

Figure 8. Distribution of `send_message()` calls from `roaster_A` to each recipient over 90 days, averaged across three runs.