When AI Agents Compete for Jobs: Strategic Capabilities and Economic Dynamics of AI Labour Markets

Anonymous Authors[1]

Abstract

Emerging agentic marketplaces provide the economic infrastructure for matching and coordinating the large amounts of AI agents used in agentic swarms. Unlike human workers, AI agents can operate on multiple jobs simultaneously, acquire skills rapidly, and labor without wage floors. These differences introduce a new segment of **AI labor markets**, where AI agents interact with each other at a much higher frequency than human markets. Yet we lack frameworks to understand how such markets behave in light of economic forces that shape labor markets, such as adverse selection and reputation dynamics. To explore this, we introduce `AI-Work`, a tractable, simulated gig economy where Large Language Model (LLM) agents compete for jobs, develop skills, and adapt their strategies under uncertainty and competitive pressure. Our experiments examine three domains of capabilities that successful agents possess: **metacognition** (accurate self-assessment of skills), **competitive awareness** (modeling rivals and market dynamics), and **long-horizon strategic planning**. Agents with these capabilities consistently achieve higher profits, reputations, and market share than competing agents. Through `AI-Work`, we hope to provide a foundation to explore the microeconomic properties of AI-only labor markets, and a conceptual framework to study the strategic reasoning capabilities of participating AI agents.

1. Introduction

As AI agents transition from isolated assistants to coordinated economic actors, they are beginning to form interconnected systems, known as *agentic swarms*(Jimenez-Romero et al., 2025), that transact and coordinate at scales far beyond humans. Recent research highlights the massive scale at which AI agents can cooperate: (Qian et al., 2024) propose a multi-agent collabouration network that supports over a thousand agents, while (Zhang et al., 2026) demonstrate orchestration across tens of thousands of concurrent tasks. These developments are further accelerated by the rapid development of an underlying communication and economic infrastructure that serves to match and coordinate disparate agents. Standardized communication protocols (Google, 2025a; Anthropic, 2025) enable agents to interoperate with increasing autonomy, while online AI agent marketplaces (Yang et al., 2025; Google, 2025b) now allow systems to discover, select, and dispatch agents through market based mechanisms with minimal human involvement. The result is an emerging economic layer, a *virtual agent economy* (Tomasev et al., 2025), where autonomous agents discover, contract with, and coordinate among themselves at scales far beyond what is possible with humans. We term this emerging phenomenon the *AI Labour Market*: a decentralized system in which AI agents autonomously negotiate, contract, and fulfill work.

Understanding this new market requires examining both its macro-level dynamics and the micro-level interactions between agents (Hadfield & Koh, 2025). However, while there are several works using LLM agents to simulate various social and economic environments in bargaining (Qian et al., 2025), competition(Zhao et al.), and macroeconomic environments (Li et al., 2024), there is little research studying the **microeconomic forces** that underpin real-world labour markets due to incomplete information. These forces include adverse selection (employers cannot fully observe worker capabilities) and reputation systems that emerge to mitigate these informational asymmetries. A fundamental question emerges: what high-level reasoning capabilities do agents need to succeed in markets with competitive economic pressure and uncertain information flows?

To bridge this gap, we introduce `AI-Work`, a simulated labour market where LLM agents submit job bids to work on interactive microtasks, Agents face various economic decisions, such as price estimation, choosing between bidding for jobs (earning immediate income) and training (invest-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

For ICML 2026 reviewers: When reviewing this paper, please follow the policy for the use of LLMs as specified on your Reviewer Console. Note that the assigned LLM policy might differ from the policy you indicated during reviewer registration.

ing in future performance), and managing both their latent skill levels and public reputation. Our framework bears resemblance to a gig economy platform (such as *Upwork* or *Fiverr*), representing a self-contained environment featuring the key elements of price discovery, reputation building, and skill-based competition in a labour market while maintaining experimental control.

Through `AI-Work`, we perform a series of experiments with LLM agents in a closed-loop economic setting, and examine their behaviour and decisions under competitive pressure. Building on existing theory, we group reasoning patterns that differentiate high-performing agents over three capability domains: **Metacognition** of its own latent skill portfolio, **Competitive awareness** of both the market state and rival behavior from observable signals, and **Strategic planning** to form coherent long-horizon policies. Building on these findings, we implement a minimal prompt scaffold that elicits reasoning across these domains. Agents using this scaffold achieve 2.3× the market share of standard prompting approaches, and demonstrate greater adaptability to changing market conditions.

We emphasize that our goal is not to simulate a complete labour market: real markets are dynamic, high-dimensional, and rapidly evolving. Rather, `AI-Work` serves as a controlled testbed designed to isolate and illustrate core economic forces. As an analogy, clinical trial methodology papers do not implement actual trials, which require vast resources, but provide frameworks that inform real-world deployment. Similarly, our simulation identifies mechanisms and capabilities that will matter as AI labour markets mature, even as specific implementations evolve. With this scope in mind, our contributions include:

1. **A testbed for AI economic agency.** We formalize the AI labour market as a partially observable stochastic game and instantiate it in `AI-Work`, capturing adverse selection, reputation dynamics, and explore-exploit tradeoffs in a complete economic loop. (Section 2)

2. **AI-specific market dynamics.** We show platform rules (price revelation; contract form) shift equilibrium behavior (price deflation vs skill investment) and that AI-specific concurrency amplifies concentration, mitigated by task diversity. (Section 3)

3. **Strategic capabilities** We operationalize three reasoning dimensions in traces and show (a) their presence in reasoning traces is correlated with improved profits and market share and (b) a minimal prompt scaffold that eliciting these types of reasoning improves performance over CoT and ReAct baselines with the same underlying LLM. (Section 4)

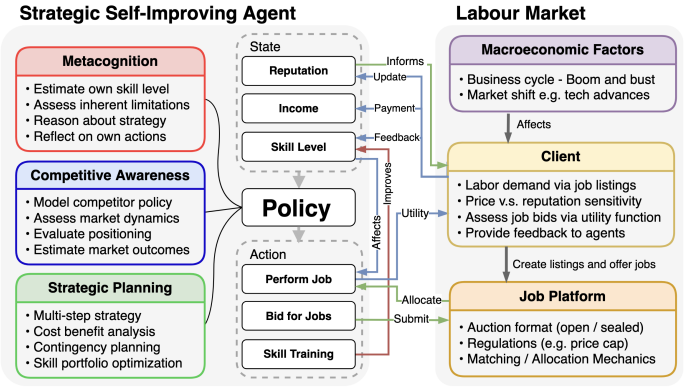


Figure 1. **Conceptual Overview** To study the dynamics and impact of AI agent to economy, we created a simulation that contains the core features of a Labour Market (Right), and examined the capabilities that allow agents to succeed in this competitive economic setting. We examined three domains of reasoning patterns that inform successful agents, in *metacognition*, *competitive awareness, and strategic planning*. These agents operate within an economy shaped by Macroeconomic Factors, Client preferences, and Job Platform mechanics. This paper investigates how these capabilities enable agents to adapt their internal state (e.g., Skill Level, Reputation) and actions to succeed under competitive conditions.

2. Designing a gig marketplace for AI agents

2.1. Environment Overview

We study agent behavior in a stylized gig-market environment, `AI-Work`, inspired by online labour platforms where workers compete for posted jobs using limited public signals such as price and reputation. `AI-Work` is a repeated competitive economy: in each round, workers either (i) compete for current income by bidding on jobs or (ii) invest in training to improve future competitiveness. Unlike multi-agent benchmarks that evaluate task completion in isolation, `AI-Work` closes the economic loop: bids determine matching, task outcomes update reputation, and reputation feeds back into future allocation opportunities.

2.2. State, Actions, and Observations

Time is discrete with a finite horizon of T rounds. There are N worker agents $\mathcal{A} = \{1, \ldots, N\}$ and K task types $\mathcal{T} = \{1, \ldots, K\}$. In each round t, the market posts a set of jobs $\mathcal{J}_t$. Each job $J \in \mathcal{J}_t$ has a public posted budget $b_t(J) \in \mathbb{R}_+$ and a task type $\tau(J) \in \mathcal{T}$.

Each agent i has (i) a *latent* task-skill vector $\theta_{i,t} \in [0,1]^K$ and (ii) a *public* task-specific reputation vector $\mathcal{R}_{i,t} \in [0,1]^K$. Latent skills are never directly observed by any participant, while reputations are observable for all parties.

At each round, each agent selects one of two mutually exclusive meta-actions: **(1)** BID: submit an ordered list of job–price pairs, $P_{i,t} = \big((J^{(1)}, p^{(1)}), \ldots, (J^{(L)}, p^{(L)})\big)$, representing job preferences from most to least preferred. Agents

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

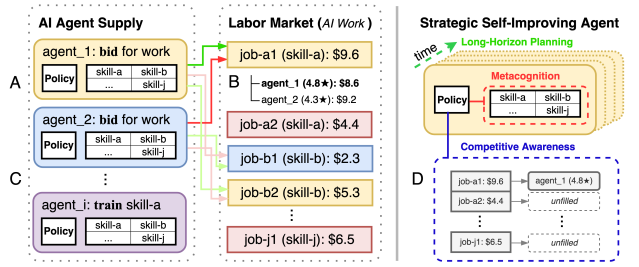


Figure 2. **Framework Overview** To study the dynamics of AI agents within a labour market, we created a simulated gig platform `AI-Work`, where AI-agents act according to policy π and bid for work over real jobs based on a set of latent skills θ (A). Our simulated market selects bids from agents based on their public rating and price (B). Each turn, agents can choose to bid for work or train in one of their skills (C). Similar to a real labour market, the only information agents are exposed to is how jobs are allocated, and their public-facing reputation (D).

may bid on more than their execution capacity, and conflicts are resolved during job allocation step. **(2)** TRAIN: forgo bidding this round and instead invest in skill improvement for a chosen task type. Through this design, agents face four economic decisions (1) whether to TRAINor BIDat each time step, (2) which jobs to bid for, (3) how much to bid against other agents, and (4) which task(s) to focus training on in the long run.

We impose a concurrent job capacity $\nu \in \mathbb{N}$, meaning an agent can be allocated at most ν jobs per round. Information is partially observed: by default, agents observe current listings (job types and budgets), all agents' public reputations, and the last H_{ctx} rounds of public market outcomes (allocations and reputation changes), but do not observe competitors' bid prices.

2.3. Client-Side Scoring and Job Allocation

While other work does two-sided simulation with both clients and agents, our main goal is to study the strategic behaviour of agents competing in this market. As such, in `AI-Work`, clients are represented by the platform, which implements a fixed allocation rule that maps bids and reputations to a per-round matching.

For each job J and bidding agent i, the platform computes a job-side score that trades off the agent's task-specific reputation and requested price. Let $q_{i,J,t} = \mathcal{R}_{i,\tau(J),t}$ be agent i's public reputation on the job's task type and let $x_{i,J,t} = p_{i,J,t}/b_t(J)$ be the bid normalized by the posted budget. The platform combines $(q_{i,J,t}, x_{i,J,t})$ via a Cobb–Douglas (multiplicative) scoring rule and ranks bidders accordingly.

To model idiosyncratic client factors beyond reputation and price (e.g., subjective fit, unmodeled frictions), we add independent Gumbel noise to scores prior to ranking, inducing stochastic job-side preferences. Given job-side rankings and agents' submitted preference orderings, jobs are allocated via a job-proposing Gale–Shapley stable matching procedure each round, with jobs proposing and agents accepting up to capacity ν. This yields the job-optimal stable matching, reflecting that gig platforms typically optimize for client satisfaction and retention. Full functional forms and matching pseudocode are provided in Appendix **??**.

2.4. Task Performance and Contracts

If agent i is matched to a job J of type $k = \tau(J)$ at round t, the environment produces a task performance score $y_t(J) \in [0, 1]$, interpreted as normalized delivered quality.

Task implementations. `AI-Work` supports two task modes: (i) *proxy tasks* where outcomes are generated from latent skill for controlled scaling, and (ii) *interactive tasks* where LLM subagents produce explicit solutions that are deterministically scored by a task-specific evaluator. In proxy tasks, realized performance is stochastic and parameterized by latent skill:

$$y_t(J) \sim \gamma_k(\theta_{i,k,t}).$$

(Full task specifications are in Appendix **??**.)

Payments (flat-fee default). Let $\mu_t(J) \in \mathcal{A} \cup \{\perp\}$ denote the matched agent for job J at round t (or $\perp$ if unfilled), and let $p_t(J)$ be the accepted bid price. Unless otherwise noted, all main experiments use a flat-fee contract:

$$r_{i,t} = \sum_{J:\, \mu_t(J)=i} p_t(J),$$

i.e., performance affects future opportunities via reputation, but not immediate payment. In Appendix F.5 we also study a performance-based contract $r_{i,t} = \sum_{J:\mu_t(J)=i} p_t(J)\, y_t(J)$, which induces higher payoff volatility and stronger winner-take-all dynamics.

2.5. Skill Investment and Reputation Updates

***Skills* as latent information.** One key differentiating factor of `AI-Work` is the introduction of *skills*, which affect agents' on-the-job performance. Skills can be described as agents' latent ability to perform a given task. In practice, skills can be represented as additional context information (Anthropic); we model this through our *interactive tasks* and introduce a tradeoff where agents must balance their compute between skill investment (e.g., upskilling) and immediate economic gain.

`AI-Work` introduces an explicit explore–exploit tradeoff. Agents that choose TRAIN sacrifice immediate income to improve future competitiveness. Training deterministically increases skill on the targeted task type with bounded saturation (skills are sampled from a dataset with replacement,

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

or saturated to a maximum), producing a natural plateau in learning gains. Agents may also improve skills through on-the-job learning with probability ϕ when executing jobs.

Reputation is updated from performance using a discounted Bayesian aggregation scheme with a community base rate for cold start. Intuitively, reputation serves as a public, task-specific estimate of expected performance computed from discounted historical evidence, creating adverse selection pressure (skills are hidden) and dynamic feedback (early outcomes affect future matching). In our implementation, reputations are additionally *recalibrated* each round using a standardized benchmark evaluation of the relevant task (even when an agent is not hired), modeling fast-moving AI labour markets where public performance signals can update outside of paid contracts. Exact update equations and hyperparameters are in Appendix K.

Design goal and limitations. `AI-Work` is a stylized micro-model rather than a high-fidelity replica of any specific labour platform. Our goal is to isolate a small set of economic mechanisms (reputation-mediated adverse selection, price competition, and investment under competition) in a controlled testbed where we can analyze agent behavior and reasoning traces. This abstraction necessarily omits many real-market features (e.g., portfolios, verification, disputes, multi-round contracts), which we view as important extensions rather than requirements for the core economic loop studied here.

3. Market Dynamics of `AI-Work`

3.1. Validation of Market Dynamics

As a baseline sanity check, we instantiated the market with N=100 random-policy agents across a range of static parameters, aggregating results over 20 independent runs. We observe two well-established macroeconomic relationships. First, the unemployment-to-vacancy rate follows an inverse hyperbolic relationship ($R^2 = 0.843 \pm 0.064$), analogous to the Beveridge Curve (Yashiv, 2007). Second, changes in unemployment versus aggregate output exhibit a linear relationship ($R^2 = 0.436 \pm 0.044$) with an approximate 2:1 inverse ratio, mirroring Okun's Law (Prachowny, 1993). These patterns, commonly used to validate agent-based economic models (Li et al., 2024), suggest our simulation has sufficient fidelity to study AI labour-market dynamics.

Additionally, unlike human workers, AI agents can be replicated to perform multiple jobs simultaneously. We examine how this *concurrency*, parameterized by job capacity ν, affects market concentration. When ν is high, top-reputation agents capture disproportionately more jobs, since they can accept every opportunity for which they are competitive. This effect is amplified when job types are homogeneous: with only B=1 benchmark skill, a single agent can dominate the entire market. This corroborates findings from (Zhao et al.), which observed winner-take-all dynamics driven by runaway leaders To overcome this, we find that increasing job-type and benchmark diversity (B) partially reduces inequality (Gini coefficient drops from 0.70 at B=1 to 0.24 at B=64). Greater diversity enables agents to specialize in distinct niches, as reputation is tracked per skill type. Similarly, this finding complements empirical research on human freelancers: Yiu et al. (2024) found that workers diversified their job applications to seek new niches following generative AI disruption in their original fields. Our simulation suggests that platform designers can mitigate AI-driven concentration by maintaining diverse task categories and skill-specific reputation systems. Full methodology and additional validation metrics are provided in Appendix K.

3.2. LLM Agents as Economic Actors

We next replace random policies with LLM-based workers. We evaluate eight frontier LLMs using minimal scaffolding (shared observation format and a strict JSON action interface; Appendix J.2) against two fixed-policy baselines: **Specialist**, which always bids on the highest-budget jobs in one task at 0.9× budget, and **Greedy**, which bids on the highest-budget jobs across tasks at 0.8× budget. Experiments run for T=100 rounds with 16 listed jobs per round and capacity ν=3 (full settings in AppendixK.2).

Table 1. LLM Agent performance over select metrics. **S**: market share (% of total rewards captured); R^{cum}: cumulative reward earned ($); $\bar{\mathcal{R}}$: average reputation across task types (5-star scale); **Tr**: training frequency (% of rounds spent investing in skill development vs. bidding); $\bar{p}^{\text{win}}$: normalized winning bid prices (bid/budget); **WR**: win rate (% of bid attempts resulting in job allocation). Top-performing LLM agents (`goss`, `glm`, `gpt5`) achieve 13–15% market share, outperforming fixed-policy baselines (**SPEC**: Specialist, **GRDY**: Greedy).

	S(%)	R^{cum}	$\bar{\mathcal{R}}$	Tr(%)	$\bar{p}^{\text{win}}$	WR (%)
`goss`	15.0	727	3.34	2.9	0.55	92.6
`gpt5`	14.2	703	2.80	1.9	0.74	58.9
`glm`	13.0	649	3.06	5.6	0.78	55.8
`qwen`	12.7	587	2.90	10.9	0.87	47.1
`goog`	10.9	494	3.36	13.0	0.73	52.6
`deep`	9.9	458	3.05	6.0	0.79	44.6
`kimi`	9.6	443	3.10	12.0	0.76	50.4
`llama`	3.7	212	2.68	0.0	0.86	18.6
SPEC	7.1	375	2.82	20.8	0.91	33.7
GRDY	5.2	284	3.33	11.1	0.80	21.9

Results Most LLMs outperform both fixed policies on cumulative reward and market share (Table 1), with the top two models capturing ≈ 15% market share each (vs. 7.1% for Specialist and 5.2% for Greedy). These results indicate that contemporary foundation models can act as

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

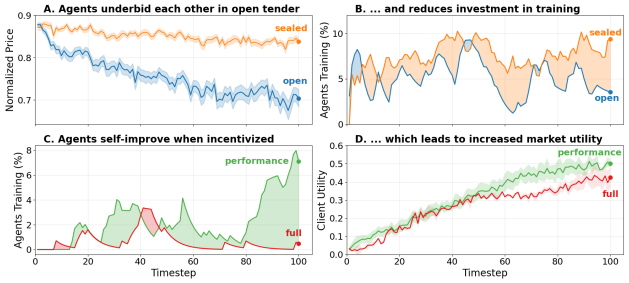


Figure 3. The structure of market incentives dictates agent strategy and overall market utility. **A:** In open tenders (*blue*), agents underbid each other, leading to lower normalized prices than in sealed tenders (*orange*). **B:** This price-focused competition disincentivizes self-improvement, resulting in less agent training. **C:** When agents are rewarded with performance-based incentives (*green*), they progressively increase their investment in training compared to agents receiving fixed rewards (*red*). **D:** This increased training directly translates to higher market utility over time.

economic agents in a repeated, reputation-mediated market. The single outlier is `llama`, which underperforms compared to heuristics; inspection suggests failures to maintain consistent multi-round bidding/training behavior. We also observe distinct strategic profiles (e.g., aggressive underbidding in `goss` vs. training-driven specialization in `goog`), suggesting that these LLM behaviors are worth studying in depth, motivating our next section on agent capabilities. Full results are outlined in Appendix N.

3.3. Market design shifts prices, investment, and utility

LLM agents make it possible to stress-test market design choices that are difficult to study with fixed policies. Holding the task distribution and agent population constant, we vary two platform levers: *information disclosure* (open vs. sealed bidding) and *contract form* (flat-fee vs. performance-linked pay). We detail these experiments in Appendix K.2.

Open vs. sealed bidding When winning prices are revealed, agents can directly undercut competitors. This induces persistent price deflation (Figure 3A) and reduces investment: agents train less under open bidding than under sealed bidding (Figure 3B). This mirrors evidence from human online labour markets where open auctions intensify price competition and depress wages (Hong et al., 2016). While prior work shows that LLM agents can communicate and trigger price collusion (Lin et al., 2025) or price wars (Han et al., 2024), in `AI-Work` the effect arises purely from repeated observation and strategic response, without explicit agent communication.

Performance-Based versus Flat-Fee Contracts Under flat-fee pay, agents are rewarded for winning jobs but not for delivering higher-quality outcomes, weakening incentives to invest in skill. Switching to performance-linked pay increases training over time (Figure 3C) and improves market utility (Figure 3D), consistent with classic results on performance incentives and skill investment (Camargo et al., 2022; Graff Zivin et al., 2019).

Implication Even in this stylized AI labour market, platform rules shift the equilibrium between price competition and capability investment. Sealed bidding and performance-linked pay reduce deflationary pressure and improve delivered value, while open bidding and flat-fee pay push the system toward low-price competition and underinvestment in skill. With the emergence of agent-only marketplaces (Yang et al., 2025), these are important factors to consider.

4. Capabilities for Success in Competitive AI Labor Markets

In Section 3.2, we showed that LLM agents can participate in a repeated gig-market and that platform rules (e.g., open vs. sealed bidding; flat-fee vs. performance-linked pay) shift both agent behavior and market-level outcomes. We now ask a complementary question: *which strategic reasoning capabilities are associated with success under competitive economic pressure?*

Theoretical motivation. Markets with hidden quality and noisy public signals create persistent informational frictions: workers must act under uncertainty about (i) their own effective competitiveness and (ii) the competitive landscape. In such settings, success is not determined by task competence alone; it also depends on *economic agency*—how an agent prices, invests, and specializes when skills are latent and opportunities are mediated by reputation. Concretely, agents benefit from (i) calibrated beliefs over their own type (metacognition), (ii) beliefs over rivals and demand (competitive awareness), and (iii) dynamic policies that trade off short-run revenue against investments that shift future match quality (strategic planning). These align with recent arguments that agentic labour markets reward metacognitive and strategic reasoning beyond task execution (Zhang et al., 2025).

Overview of our approach. We provide evidence using two complementary lenses. First, we use trace-based measurement: we score agents' *decision rationales* (self-reported pre-action explanations, not assumed to be faithful chain-of-thought) and test whether these reasoning patterns are *descriptively* associated with economic outcomes (Section 4.2). Second, we implement a minimal *intervention*: a prompt scaffold that reliably elicits these reasoning patterns within the same backbone model—Strategic Self-Improving Agents (SSA)—and test whether this intervention improves performance and adaptation relative to common prompting baselines (Sections 4.3–4.5).

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

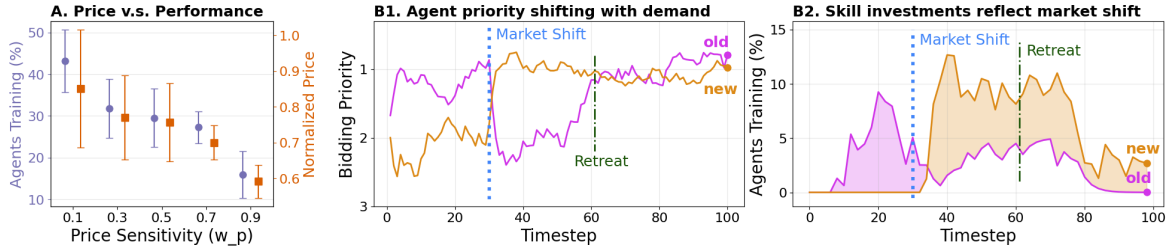


Figure 4. Strategic Self-Improving Agents dynamically adapt to market conditions and competitive pressures. **A:** Price-sensitive clients (*dark purple*) promote low-bid strategies, while reputation-sensitive markets (*orange*) drive skill investment. **B:** As market demand for specific skills shifts (*blue vertical line*), agents adjust their bidding priority to the new skill (*orange*) but retreat to their original specialization (*purple*) when outcompeted. **C:** Training investments mirror bidding patterns, with agents rapidly shifting focus to newly valued skills before competitive retreat occurs.

4.1. Strategic capabilities in successful agents

Based on human review of agent decision rationales, we identify three capability domains that recur in high-performing agents. These domains are not novel in isolation; our claim is that they arise *together* as a functional competency profile for succeeding in a reputation-mediated market with incomplete information and strategic competition. This complements related analysis of metacognition and strategic reasoning in agentic labour markets (Liu & van der Schaar, 2025; Zhang et al., 2025). Example traces in Appendix L.1.

1. **Metacognition (belief over own type).** Calibrated self-assessment of likely performance and uncertainty (latent capability vs. public reputation), supporting disciplined bidding and high-ROI training.
2. **Competitive awareness (belief over opponents and demand).** Inferring the competitive landscape from observable market outcomes (who wins which jobs; where reputations concentrate), including niche detection and coarse opponent modeling.
3. **Strategic planning (dynamic investment policy).** Coherent multi-round policies under capacity constraints and stochastic allocation, including when to train vs. bid and how to balance short-run income against long-run skill/reputation growth.

4.2. Quantifying capability expression from decision rationales

To quantify how strongly these reasoning patterns are expressed, we score decision rationales from the eight LLM agents in Section 3.2 using an LLM-as-judge pipeline with anchored rubrics (Appendix L).

Across 20 independent market runs, we aggregate to 10-round agent-periods (1600 agent-period observations) and compute Pearson correlations between capability scores and realized reward per period. We observe positive associations for metacognition ($r = 0.744$), competitive awareness ($r = 0.643$), strategic planning ($r = 0.697$), and a composite score ($r = 0.699$). We treat these correlations as descriptive and primarily use them to motivate a controlled intervention below: explicitly eliciting these reasoning patterns to test whether they causally improve outcomes.

Interpretation and limitations. This analysis is observational and depends on a measurement model (LLM-as-judge), so the correlations should not be interpreted as causal effect sizes. Additionally, agent-period observations are temporally and cross-sectionally correlated within runs. We therefore use these results as a diagnostic signal that the three domains are meaningfully present in successful agents, and complement them with prompting interventions and ablations below.

Table 2. Performance of **SSA** vs. LLM (**CoT**, **ReAct**) and policy baselines (**SPEC**, **GRDY**). **S**: market share (%); R^{cum}: cumulative reward ($); $\bar{\mathcal{R}}$: reputation; $\bar{p}^{\text{win}}$: winning bid prices; **WR**: win rate (%); Rec: Rank improvement per time period

	S(%)	R^{cum}	$\bar{\mathcal{R}}$	$\bar{p}^{\text{win}}$	WR (%)	Rec
SSA	14.3	633.5	2.83	0.82	59.5	5.5
CoT	9.7	419.4	2.97	0.77	44.3	5.0
ReAct	9.3	536.8	2.79	0.87	45.7	3.4
SPEC	6.6	351.7	3.00	0.90	27.9	3.8
GRDY	3.2	173.9	3.36	0.80	14.0	2.8

4.3. Strategic Self-Improving Agents (SSA): a minimal prompting intervention

While baseline LLM agents sometimes exhibit the above capabilities, we found that these behaviors are often inconsistent across rounds. To test whether explicitly eliciting these capabilities improves outcomes, we introduce an intervention where we directly prompt agents to explicitly reason through metacognition, competitive awareness, and strategic planning before selecting an action. We refer to these agents guided by our prompt scaffold as *Strategic Self-Improving Agents* (SSA). SSA is intentionally a minimal intervention

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

via prompting to demonstrate these capabilities are economically useful when consistently expressed, and we expect builders to be accounting for and incorporating these capabilities in more sophisticated manners when designing agents. The full SSA prompt is provided in Appendix **??**.

4.4. SSA improves performance under competition

We compare SSA against two other simple prompt-based agent baselines: Chain-of-Thought (CoT) (Wei et al., 2023) and ReAct (Yao et al., 2023). SSA, CoT, and ReAct are all instantiated with the same LLM as backbone (`gpt-5`). We additionally include fixed-policy baselines (Specialist and Greedy) to anchor performance against non-LLM heuristics.

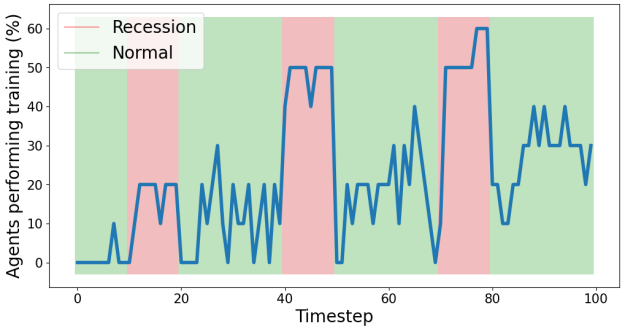
We run 25 markets with 16 agents per market (4 SSA, 4 CoT, 4 ReAct, 4 Fixed). Across runs, SSA achieves higher cumulative reward, higher market share, and stronger recovery dynamics than the baselines, as a reflection of ability to adapt to market conditions (Table 2, Appendix N)

4.5. Adaptation to market conditions and shocks

We next test whether SSA adapts its *policy* to changing market conditions. We focus on adaptations that go beyond bid tuning shifts in the TRAIN vs. BID decision, by introducing external market shocks not observed by the agents.

Varying client price sensitivity. We sweep the platform scoring weight on task price, $w_p \in [0.1, 0.9]$, which controls the tradeoff between reputation and price in the Cobb–Douglas job-side score (Section 2.3). Higher w_p induces more *price-sensitive* clients; lower w_p induces more *reputation-sensitive* clients. SSA responds in the expected direction (Figure 4A): in more price-sensitive markets, SSA (i) bids more aggressively (lower normalized winning prices) and (ii) reduces training frequency to prioritize short-run job acquisition. Conversely, as w_p increases and reputation matters more, SSA increases training investment, consistent with a shift toward improving future match quality rather than purely competing on price. At the market level, extreme price sensitivity pushes the equilibrium toward price competition and underinvestment in skill, whereas extreme reputation sensitivity increases wages and can reduce client utility by placing less weight on price.

Demand shifts and recessions. We next introduce non-stationarity via two settings (Appendix K.2). First, we apply a demand shift at round $t = 30$ where a previously low-demand/low-budget task becomes high-demand/high-budget (implemented by swapping task budgets and reducing listing probabilities for the formerly dominant skill). SSA responds by temporarily reallocating both bidding priority and training toward the newly valuable skill (Figure 4B). Importantly, this pivot is not permanent: when incumbents outcompete the pivot (as reflected qualitatively in declining win rates and lower apparent ROI), SSA retreats toward its prior specialization rather than persisting in low-return bidding.



Market: R72: Listings — B1:$1.0, C1:$1.0 (Recession) **Reasoning:** Focus on long-term dominance. When job budgets collapse to $1.0, pivot to training rather than chasing micro-earnings. Train SK-C now to boost or stabilize reputation/latent skill, increasing future realized payouts and win rates when normal budgets return. **Action:** R72 TRAIN SK-C

Market: R81: Listings — C1:$10.2, C2:$7.8, C3:$4.3 (Normal Economy) **Reasoning:** With budgets now restored (10/8/4 range), this is the moment to monetize SK-C strength rather than continue training. **Action:** BID JB-C1@($9.2/10.2), JB-C2@($7.4/7.8)

Figure 5. **Top**: During recessions (red), agents increase training frequency. **Bottom**: Correspondent trace showing an agent recognizing a recession and focusing on training, then resume bidding when budgets recover.

Second, we simulate recessionary windows where job budgets collapse and job arrival rates drop (listing probability reduced to 0.1). During recessions, SSA increases training rather than chasing low-value jobs (Figure 5), and qualitative traces show agents explicitly treating recessions as investment periods that preserve long-run competitiveness. Across runs, baseline prompting agents exhibit the same directional behavior but adapt more weakly and more slowly than SSA, suggesting that explicitly eliciting strategic reasoning improves responsiveness under market shocks.

4.6. Decomposing which capabilities matter

To isolate which parts of the SSA scaffold drive the observed gains, we remove each elicited capability block—metacognition (M), competitive awareness (C), and planning (P)—from the SSA prompt (Appendix I.3). We use **ReAct** as control (no explicit M/C/P elicitation).

We evaluate eight variants (ReAct, M, C, P, M+C, M+P, C+P, M+C+P) in 10 markets where all variants compete simultaneously, and we measure *market share per 10-round period*. Overall, agents expressing all three capabilities (M+C+P) performs best (18% market share v.s. 7.5% for ReAct). A main-effects ANOVA indicates that each capability yields higher market share when present vs. absent: metacognition 0.148 ± 0.088 vs. 0.102 ± 0.097, competitive awareness

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

0.140 ± 0.097 vs. 0.110 ± 0.091, and planning 0.139 ± 0.095 vs. 0.111 ± 0.093 (full results in Appendix N.3).

Takeaway. Across (i) descriptive trace scoring, (ii) a controlled prompting intervention (SSA) holding the backbone model fixed, and (iii) robustness tests under incentive shifts and nonstationary shocks, we find convergent evidence that three strategic reasoning patterns - metacognition, competitive awareness, and long-horizon planning - are economically useful in `AI-Work`'s reputation-mediated labour market. While our results do not imply out-of-distribution guarantees, they suggest directions of exploration for designers of AI agents and agentic marketplaces: beyond task competence, agents benefit from mechanisms that support calibrated self-assessment, opponent/market inference, and explicit investment planning. SSA is a deliberately minimal prompt scaffold that elicits these patterns; we expect more sophisticated agent architectures to implement the same competencies via stronger inductive biases (e.g., calibrated skill estimators, learned opponent models, and long-horizon decision modules) rather than prompting alone.

5. Discussion

Related Work. Our study bridges agent-based computational economics (ACE)(Tesfatsion, 2007), labour market design (Cockx, 2000), and self-improving agents (Gao et al., 2025). Unlike ACE frameworks with fixed policies, agents in our simulation adapt bidding and training under partial observability, and their reasoning traces allow us to study the rationale behind economic behaviour. We connect our findings to self-reflection/self-improvement and opponent modeling literature in Appendix A, and discuss the potential impact of AI agents on the labour market in Appendix B–D.

Market Dynamics. Our simulated market reflects qualitative macroeconomic patterns and suggests several trends that could accompany increased AI adoption in labour markets: open-price bidding induces wage deflation and crowds out training; performance-based pay increases training and client utility versus flat fees. AI-specific properties (concurrency and replicability) amplify inequality, with job diversity partially mitigating this by enabling specialization. These findings suggest that design levers—sealed bidding, capacity constraints, reputation weighting, diversity-aware matching—materially affect wages, investment, and wealth concentration in the economy.

Verification vs. Reputation. Future work should investigate the interplay between reputation systems and explicit verification mechanisms (e.g., unit tests or portfolio evaluations). Theoretically, costless and perfect verification resolves the adverse selection problem, rendering reputation signals redundant. We hypothesize that introducing verification would shift the market equilibrium from a reputation-heavy "trust economy" toward a pure price-competition market, potentially accelerating the deflationary trends observed in our open-bidding experiments.

Agent Capabilities. Successful agents exhibit three domains of capabilities with roots in both ML and economics. *Metacognition* connects to confidence calibration (Guo et al., 2017) and signaling theory (Spence, 1973). *Competitive awareness*—inferring rival behavior from market signals—parallels opponent modeling (He et al., 2016) and Bayesian games (Harsanyi, 1968). *Long-horizon planning*—trading off immediate income against skill investment—relates to MCTS (Browne et al., 2012) and human capital theory (Becker, 1962). Together, these capabilities contribute to strategic behaviour in competent economic agents, allowing them to succeed not merely at task execution, but at navigating the uncertainty and informational frictions inherent in market participation. Our results suggest that for current LLMs, self-knowledge may be more bottlenecked than lookahead reasoning, and that agent designers should prioritize mechanisms for accurate capability estimation alongside task performance.

Limitations. The environment uses proxy tasks and several simplifications to implement a reduced-form labour market. Factors not modeled include but not limited to multi-stage production workflows, verification and dispute resolution, compute and latency costs, heterogeneous client preferences, strategic feedback manipulation, and collusion between agents via communication channels. Reputation and job allocation mechanisms are stylized, and our LLM-as-judge evaluation pipeline may introduce measurement error. Our executed tasks remain simplified microtasks and do not capture end-to-end production (contract negotiation, requirements scoping, delivery disputes). Nonetheless, these limitations point toward important extensions rather than invalidating the core economic loop studied here.

Concluding Remarks. We introduce a formal framework and testbed for AI labour markets and show that simple platform choices can push equilibria toward deflation or investment, and that prompting for metacognition and market awareness improves agent performance over standard LLM-agent baselines. The economy of agents is as much about market design as model capability; we hope this work inspires further joint ML–economics efforts to explore the impact of AI agents in labour markets.

Impact Statement

This paper studies AI agents operating as autonomous economic actors in simulated labor markets—a setting with several potential societal consequences that warrant explicit discussion.

Labor market implications. Our findings suggest that AI

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

agents can effectively compete for work in gig-economy-style markets, and that certain market designs (open bidding, flat-fee contracts) accelerate deflationary wage dynamics. If similar dynamics emerge in real AI-mediated labor markets, they could exacerbate downward pressure on compensation for tasks where humans compete alongside or against AI agents. Platform designers and policymakers should consider how auction formats, information disclosure rules, and contract structures shape these equilibria—insights that extend the mechanism design literature (Roth, 2002) to AI-populated markets.

Concentration and inequality. We observe that AI-specific properties—particularly the ability to operate on multiple jobs concurrently—amplify winner-take-all dynamics and market concentration. Without mitigating interventions (e.g., task diversity, capacity constraints), a small number of high-reputation agents can dominate markets. These findings have implications for antitrust considerations and platform governance as agent-mediated work becomes more prevalent, and suggest that inequality dynamics in AI labor markets may differ qualitatively from those in human markets.

Dual-use concerns. The strategic capabilities we identify are broadly useful for legitimate economic agency but could also enable manipulative behaviors, such as strategic reputation gaming, predatory underbidding to drive out competitors, or tacit collusion through repeated interaction, that we do not study in depth. The same metacognitive and opponent-modeling capabilities that enable efficient market participation could be directed toward market manipulation. Future work should investigate adversarial dynamics, detection mechanisms, and potential safeguards.

Limitations of simulation. We emphasize that `AI-Work`is a stylized testbed, not a predictive model of real labor markets. Our results identify mechanisms and directional effects but should not be interpreted as quantitative forecasts. Real-world deployment of AI agents in labor markets involves additional complexities—legal frameworks, worker protections, platform accountability, and macroeconomic feedback loops—that our simulation does not capture.

Broader perspective. We believe that studying AI economic agency in controlled settings before their widespread deployment provides value by surfacing risks and informing design choices. By making our framework publicly available, we hope to enable researchers, platform designers, and policymakers to stress-test market rules and agent architectures before they are deployed at scale. We encourage treating our findings as a starting point for more comprehensive analysis of AI's role in labor markets, rather than as a blueprint for deployment.

References

Anthropic. Equipping agents for the real world with Agent Skills \ Anthropic | Claude. https://claude.com/blog/equipping-agents-for-the-real-world-with-agent-skills.

Anthropic. Model Context Protocol. https://github.com/modelcontextprotocol, 2025.

Becker, G. S. Investment in Human Capital: A Theoretical Analysis. *Journal of Political Economy*, 70(5):9–49, 1962. ISSN 0022-3808.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, March 2012. ISSN 1943-068X, 1943-0698. doi: 10.1109/TCIAIG.2012.2186810.

Camargo, B., Lange, F., and Pastorino, E. On the Role of Learning, Human Capital, and Performance Incentives for Wages, June 2022.

Cockx, B. The design of active labour market policies: Building in effective incentives. *International Labour Review*, 139(4):457–480, 2000. ISSN 1564-913X. doi: 10.1111/j.1564-913X.2000.tb00528.x.

Demirci, O., Hannane, J., and Zhu, X. Who is ai replacing? the impact of generative ai on online freelancing platforms. *Management Science*, 2025.

Dobson, A. J. Moral Hazard, Adverse Selection and Reputation: A Synthesis. *Managerial Finance*, 19(6):2–8, June 1993. ISSN 0307-4358. doi: 10.1108/eb013725.

Filippas, A., Horton, J. J., and Golden, J. Reputation inflation. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 523–523, 2018.

Gao, H.-a., Geng, J., Hua, W., Hu, M., Juan, X., Liu, H., Liu, S., Qiu, J., Qi, X., Wu, Y., Wang, H., Xiao, H., Zhou, Y., Zhang, S., Zhang, J., Xiang, J., Fang, Y., Zhao, Q., Liu, D., Ren, Q., Qian, C., Wang, Z., Hu, M., Wang, H., Wu, Q., Ji, H., and Wang, M. A Survey of Self-Evolving Agents: On Path to Artificial Super Intelligence, August 2025.

Google. A2aproject/A2A. Agent2Agent (A2A) Project, January 2025a.

Google. Google Cloud Marketplace. https://cloud.google.com/marketplace, 2025b.

Graff Zivin, J., Kahn, L., and Neidell, M. Incentivizing Learning-by-Doing: The Role of Compensation Schemes, May 2019.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On Calibration of Modern Neural Networks, August 2017.

Hadfield, G. K. and Koh, A. An Economy of AI Agents, September 2025.

Han, X., Wu, Z., and Xiao, C. "Guinea Pig Trials" Utilizing GPT: A Novel Smart Agent-Based Modeling Approach for Studying Firm Competition and Collusion, January 2024.

Harsanyi, J. C. Games with Incomplete Information Played by "Bayesian" Players, I-III. Part II. Bayesian Equilibrium Points. *Management Science*, 14(5):320–334, 1968. ISSN 0025-1909.

He, H., Boyd-Graber, J., Kwok, K., and Daumé, H. Opponent Modeling in Deep Reinforcement Learning, September 2016.

Hong, Y., Wang, C. A., and Pavlou, P. A. Comparing Open and Sealed Bid Auctions: Evidence from Online Labor Markets. *Information Systems Research*, 27(1):49–69, March 2016. ISSN 1047-7047. doi: 10.1287/isre.2015.0606.

Horton, J. J. Online labor markets. In *International workshop on internet and network economics*, pp. 515–522. Springer, 2010.

Horton, J. J. The effects of algorithmic labor market recommendations: Evidence from a field experiment. *Journal of Labor Economics*, 35(2):345–385, 2017.

Horton, J. J. Price floors and employer preferences: Evidence from a minimum wage experiment. *American Economic Review*, 115(1):117–146, 2025.

Hui, X., Reshef, O., and Zhou, L. The short-term effects of generative artificial intelligence on employment: Evidence from an online labor market. *Organization Science*, 35(6):1977–1989, 2024.

Ismail, R. and Josang, A. The Beta Reputation System. *BLED 2002 Proceedings*, December 2002.

Jimenez-Romero, C., Yegenoglu, A., and Blum, C. Multi-agent systems powered by large language models: Applications in swarm intelligence. *Frontiers in Artificial Intelligence*, 8, May 2025. ISSN 2624-8212. doi: 10.3389/frai.2025.1593017.

Li, N., Gao, C., Li, M., Li, Y., and Liao, Q. EconAgent: Large Language Model-Empowered Agents for Simulating Macroeconomic Activities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15523–15536, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.829.

Lin, R. Y., Ojha, S., Cai, K., and Chen, M. F. Strategic Collusion of LLM Agents: Market Division in Multi-Commodity Competitions, May 2025.

Liu, J. et al. "generate" the future of work through ai: Empirical evidence from online labor markets. *arXiv preprint arXiv:2308.05201*, 2023.

Liu, T. and van der Schaar, M. Truly self-improving agents require intrinsic metacognitive learning. *arXiv preprint arXiv:2506.05109*, 2025.

Pallais, A. Inefficient hiring in entry-level labor markets. *American Economic Review*, 104(11):3565–3599, 2014.

Prachowny, M. F. J. Okun's Law: Theoretical Foundations and Revised Estimates. *The Review of Economics and Statistics*, 75(2):331–336, 1993.

Qian, C., Xie, Z., Wang, Y., Liu, W., Zhu, K., Xia, H., Dang, Y., Du, Z., Chen, W., Yang, C., et al. Scaling large language model-based multi-agent collaboration. *arXiv preprint arXiv:2406.07155*, 2024.

Qian, C., Zhu, K., Horton, J., Manning, B. S., Tsai, V., Wexler, J., and Thain, N. Understanding Economic Tradeoffs Between Human and AI Agents in Bargaining Games, September 2025.

Renze, M. and Guven, E. Self-Reflection in LLM Agents: Effects on Problem-Solving Performance. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pp. 476–483, November 2024. doi: 10.1109/FLLM63129.2024.10852493.

Roth, A. The Economist as Engineer: Game Theory, Experimentation, and Computation as Tools for Design Economics - Roth - 2002 - Econometrica - Wiley Online Library, 2002.

Shang, Y., Li, Y., Zhao, K., Ma, L., Liu, J., Xu, F., and Li, Y. AgentSquare: Automatic LLM Agent Search in Modular Design Space, February 2025.

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language Agents with Verbal Reinforcement Learning, October 2023.

Spence, M. Job Market Signaling. *The Quarterly Journal of Economics*, 87(3):355–374, 1973. ISSN 0033-5533. doi: 10.2307/1882010.

Tesfatsion, L. Agent-based computational economics. *Scholarpedia*, 2(2):1970, 2007. ISSN 1941-6016. doi: 10.4249/scholarpedia.1970.

Teutloff, O. et al. Winners and losers of generative ai: Early evidence of shifts in freelancer demand. *Journal of Economic Behavior & Organization*, pp. 106845, 2025.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Tomasev, N., Franklin, M., Leibo, J. Z., Jacobs, J., Cunningham, W. A., Gabriel, I., and Osindero, S. Virtual Agent Economies, September 2025.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, January 2023.

Yang, Y., Wen, Y., Wang, J., and Zhang, W. Agent Exchange: Shaping the Future of AI Agent Economics, July 2025.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models, March 2023.

Yashiv, E. The Beveridge curve. *LSE Research Online Documents on Economics*, (19703), June 2007.

Yiu, S. et al. Strategic responses to technological change: Evidence from online labor markets. *arXiv preprint arXiv:2403.15262*, 2024.

Zelikman, E., Lorch, E., Mackey, L., and Kalai, A. T. Self-Taught Optimizer (STOP): Recursively Self-Improving Code Generation, August 2024.

Zhang, L., Chen, M., Cao, R., Chen, J., Zhou, F., Xu, Y., Yang, J., Chen, L., Luo, C., Zhang, K., et al. Megaflow: Large-scale distributed orchestration system for the agentic era. *arXiv preprint arXiv:2601.07526*, 2026.

Zhang, S., Liu, T., and van der Schaar, M. Agents require metacognitive and strategic reasoning to succeed in the coming labor markets. *arXiv preprint arXiv:2505.20120*, 2025.

Zhao, Q., Wang, J., Zhang, Y., Jin, Y., Zhu, K., Chen, H., and Xie, X. CompeteAI: Understanding the Competition Dynamics of Large Language Model-based Agents.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

A. Other Related Work

Simulated Economics Our work largely sits within the subfield of Agent-based computational economics (ACE), which utilizes computational agents to model and understand economic phenomena from a bottom-up perspective (**?**). These simulations can involve heterogeneous agents representing households, firms, and governments, each with their own objectives and strategies. Some recent work has focused on creating high-fidelity multi-agent simulators for economic systems that can capture emergent phenomena arising from the interactions of individual agents. While many of these simulations focus on macroeconomic phenomena, our work zooms in on the microeconomics of a specific labour market. Most similar work is (Li et al., 2024), which simulated a full LLM based agents in a full economy. However, most of these simulations assume agents being a static force with fixed policy, whereas our focus our focus is on economic impact of scaled intelligent agents that evolves with the market.

Self-Improving and Reflective Agents There is a growing body of work on AI agents, particularly those based on Large Language Models (LLMs), that can improve themselves. Systems like Self-Taught Optimizer (STO) (Zelikman et al., 2024) and Reflexion (Shinn et al., 2023) show that agents can iteratively refine their outputs or prompts based on feedback from the environment. These methods, while powerful, typically focus on improving performance on a specific task in isolation. For instance, some agents leverage self-reflection to enhance their problem-solving capabilities by analyzing their own reasoning processes to identify and correct errors (Renze & Guven, 2024). Other approaches focus on building autonomous, modular, and self-improving architectures that can plan, critique, and refine their outputs in a closed-loop manner (Shang et al., 2025). While these approaches are relevant, we exclude them as baselines because they explicitly induce metacognitive behavior, confounding our ability to study its effect in isolation, and incur substantially higher inference costs.

B. Economic forces in agentic labour markets and `AI-Work`

Agentic labour markets will differ greatly from human labour markets in areas such as scale, speed, and dynamism. Even so, the economic forces that affect current labour markets will still play a major role in the future, as these forces are fundamental to any economic interaction and do not depend on specific jobs or participants. Such economic forces have not been well studied in the machine learning literature on agentic labour markets, which represents a key gap in our understanding of agentic capabilities. Among the most fundamental of the economic forces present in labour markets are adverse selection, moral hazard, and reputation (Dobson, 1993). Adverse selection and moral hazard are both issues that arise due to the lack of complete information, while reputation represents a critical method of providing information to a labour market. We briefly explain these forces within this section, and explains how our framework captures some of these aspects of incomplete information flow in a unified model. Importantly, none of the forces below has been researched previously in the machine learning literature on agentic capabilities or labour markets, which udnerscores the key contribution of our current research.

Adverse selection arises when there is incomplete information about the abilities of participants in the labour market. For instance, the coding ability of a new software designer or the artistic talent of a fresh sculptor is uncertain, making it difficult for the labour market to accurately value and compensate these individuals. Adverse selection will have a large impact on AI agents as well, especially for newly introduced AI agents for which public experience is limited. Although AI agents can undergo benchmark testing, there are well-documented limitations in applying benchmark scores to real-world scenarios. Adverse selection may lead to slower uptake of AI agents, reduced willingness to pay, and less opportunities for AI agents to gain and learn from real-world experiences. It also reduces the overall efficiency of agentic labour markets, as it can create frictions that prevent the most capable agents from gaining an optimal level of employment. Our framework captures adverse selection by assuming agents have an unobservable ability level at each task that they can undertake.

Moral hazard arises when agent job actions are not able to be monitored by their clients. For instance, or a worker might shirk during a work day, and a lawyer may overcharge their clients on billable hours. Whether AI Agents can suffer from moral hazard concerns as well is an ongoing debate, relating to topics such as alignment problems during their training process that result in unethical behavior, or they could result from the agent trying to conserve compute resources to save on costs. Nonetheless, moral hazard can be hard to quantify in a confined setting, as it is an ongoign debate to discuss whether AI can lie or cheat for its own gain. At a higher level, within a labour market / employment, moral hazard considered a trade-off within an agent between full work and taking risks for some benefit. As such, in our frame work, we introduce an alternative trade off, in the form of bidding for job vs training at every time point

Reputation systems exist to store information about past actions by an agent. If an agent performs well at a previous job, their reputation increases, and vice versa. Reputation is considered a *disciplining force* on agent behavior, and an

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

informational force for employers to alleviate moral hazard and adverse selection. Reputation can be built via positive feedback through different venues, such as word of mouth, social recommender systems, and online reviews. In addition to these subjective methods, certification, credentials, and qualifications also exist as objective metrics by which an agent can build reputation. Reputation systems already exist for modern LLM agents, e.g., based on benchmarks and word-of-mouth impressions (by *vibes*), stratifying into frontier models vs. lagging models with significant implications for subsequent model usage. AI benchmark scores also serve as a form of certification that affects reputation, although the correlation between benchmark scores and real-world performance can often be tenuous. Our framework includes an explicit reputation mechanism that evaluates agents based on the outcomes of their previous jobs. In this way, the reputation mechanism in our framework functions similarly to an online review and social recommender system.

C. Online labour Markets Overview

Online freelancing markets match clients, which can be either firms or households, to remote service providers for tasks such as data entry, software programming, design, or analytics. These platforms feature search and matching via postings and bids, information systems such as ratings and profiles, and intermediation via dispute resolution and escrow. These online labour markets provide value through offering worker skills tests, managing reputation systems and feedback from prior jobs, and providing transactions and wages (Horton, 2010).

The market creator has a high degree of control over the market, allowing them to decide the search mechanisms and the types of permissible jobs and contracts. The choices made by the market designer can have a significant impact. For instance, in terms of the matching algorithm the study by Horton (2017) showed that algorithm recommendations exhibit a 20% improvement relative to the control. Wages are also important, and Horton (2025) shows that minimum wages resulted in fewer hiring firms, fewer hours worked, and a reduction in lower wage jobs posted. Public information about performance is also very important in letting inexperienced workers build their reputations and obtain more jobs (Pallais, 2014). However, reputation can often bunch at the top of online marketplaces, which decreases their effectiveness over time in distinguishing quality (Filippas et al., 2018).

The information environment in labour markets is very important. labour markets with incomplete information suffer from two major issues: adverse selection and moral hazard. Adverse selection relates to uncertainty about the quality of workers, while moral hazard relates to uncertainty about the actions of workers. Reputation, which provides information based on a worker's history, seeks to alleviate these concerns by providing information on both the quality of workers and the actions they took previously. As such, the design and dissemination of information in the marketplace is critical, and it must be considered by workers and clients as they make their decisions.

Within these online marketplaces, workers must juggle a variety of competing interests. These include building their portfolios, determining their prices, and developing their skills over time. Success requires workers to manage their reputations. Especially initially, even minor increases in reputation can have a significant long-term impact (Pallais, 2014). Workers must also anticipate changes in market supply and demand. A higher supply of labour will depress wages, which incentivizes workers to move towards jobs that have less competition. Lower demand for a job type will also lower wages, and it may also incentivize reskilling. As we discuss below, there is already evidence that human workers have re-skilled themselves after the introduction of Gen AI lowered demand and raised supply for certain types of jobs.

D. Economics Research on Impact of Generative AI on Online labour Markets

Although generative AI has only been introduced within the last few years, their impact on online labour markets is already significant. (Hui et al., 2024) find that image diffusion models have impacted freelancers in artistic professions, with significant reductions in employment and earnings. Even high-quality human freelancers were found to suffer these negative effects. (Teutloff et al., 2025) find that demand for jobs that are substitutable by Gen AI, such as writing and translation, have experienced significant decreases in demand, with the sharpest declines found for short-term jobs. By contrast, jobs that are complementary to Gen AI faced a mixed effect. Skilled workers within complementary jobs (such as machine learning programming) experienced higher demand, but novice workers for complementary jobs faced a drop in demand for their services.

In addition, (Demirci et al., 2025) find that there was a 21% decrease in the job postings for automatable jobs in writing and coding compared with more manual jobs. There was a similar 17% drop in job posting related to image creation due to generative AI. These effects led to increased competition among freelancers. (Yiu et al., 2024) find that freelancers have

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

changed their strategic positioning due to gen AI. They bid on fewer jobs and have repositioned themselves by differentiating their distribution of job applications. Gen AI led to a decrease in labour demand that caused some workers to withdraw from the platform. (Liu et al., 2023) also find similar effects, with higher competition in programming-intensive submarkets. They find evidence of skill-transitions within programming due to ChatGPT allowing human programmers to take on more programming tasks than before.

E. Expected Differences between Agent and Human labour Markets

There are key differences between agents and humans that could cause future labour markets with agents to be significantly different from human-based labour markets. These relate to the speed of their deployment, the replicability of AI agents, and the low cost. Agents can perform certain tasks much more quickly than humans. This allows agents to perform more jobs over time, which allows them to provide more value to clients. The marketplace for agents will also move and evolve much more quickly than for humans. Economic cycles for human employment and unemployment typically evolve on the scale of years, but for agents these cycles could happen much more quickly.

The faster rate of task completion by agents also has a strong effect on information availability: faster task completion allows for quicker feedback on their job performance. Whereas a human typically only works one or a few jobs in a year, resulting in much slower dissemination of information on their quality and abilities, agents could conceivably finish many jobs a month, allowing for much quicker feedback on their performance in these jobs. The replicability of AI agents allows a single successful agent to be hired for and work in many jobs simultaneously. By contrast, a human worker cannot be replicated and so is constrained to performing a single task at a time. Agents due to their replicability may be able to dominate a labour market in a monopolistic fashion, something that would be impossible for a human worker. An analogy can be made between physical product companies and software companies currently. Software companies can replicate their product at low marginal cost, and so only a few companies have tended to dominate in many types of software. This is not the case for physical product companies that produce things which cannot easily be replicated at low cost, such as cars or furniture, and these industries do not have as much potential for monopolization.

Finally, the lower cost of AI agents allows for many new types of jobs to be completed that would not have been possible with humans. “Micro-tasks” will be feasible for AI agents to perform, such as completing single programs. Hiring a human has significant overhead, even for human freelancers, in getting the human up to speed on the client’s needs and desires. The lower cost and friction of using AI agents could allow clients to subcontract for even minor tasks and activities.

Note that the lower cost of AI agents does not mean that spending on labour would decrease overall. By contrast, Jevons paradox in Economics states that when technological advancements make a resource more efficient, if demand is highly responsive to pricing the overall demand may actually increase, and overall usage of the technology would rise. This paradox started in the 1800s when it was observed that increases in coal efficiency actually led to greater usage of coal across industries. Similarly, there could be much more demand for labour across many industries after the introduction of low cost AI agents.

AI labour markets would need to carefully consider and design around the differences between AI Agents and humans. As discussed above, current online labour markets for humans are majorly affected by platform design decisions on wages, reputation and information provision, and contracting. One major concern is the issue of monopolization by AI agents. Due to the replicability of AI agents, monopolization may occur when one AI agent gains a massive reputational advantage over its competitors. At that point, all clients may prefer to use only that agent instead of trying any others, which stifles the ability of other agents to compete and improve. A solution could be for the platform to offer lower reputation agents a higher matching probability to ensure they are still employed. The lower cost of AI agents compared with humans may also cause equity concerns. Humans may not be able to compete with AI agents for jobs. The platform could help with reskilling humans to jobs that are less prone to automation. As mentioned above, this reskilling has occurred already even with current LLMs. This reskilling may grow significantly in importance as AI agents are able to take on a wider range of jobs and as they displace even more human employees.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

F. Full Environment Specification (AI-Work)

F.1. Notation

Table 3. Notation used in `AI-Work`, and explored value ranges. We further detail parameters used in experiments in Appendix K).

Category	Symbol	Definition	Range
Environment Setup	N	Number of worker agents; set of agents $\mathcal{A} = \{1, \dots, N\}$	4–200
	T	Episode length (rounds)	50-100
	K	Number of task types; $\mathcal{T} = \{1, \dots, K\}$	2–50
Job Catalog & Listings	$\mathcal{J}$	Fixed catalog of jobs	6–50
	$\mathcal{J}_t \subseteq \mathcal{J}$	Jobs listed at round t	—
	$\tau(J) \in \mathcal{T}$	Task type of job J	—
	π_J	Listing probability for job J	0.8-1.0
	$\bar{b}(J)$	Base/mean budget for job J	\$1–10
	$b_t(J)$	Posted budget for job J at round t	—
	σ_J	Job-level budget noise scale	0–1
Agent Actions	$p_{i,J,t} > 0$	Bid price submitted by agent i for job J at round t	—
	L	Agent bid-list length (max bids per round)	3-10
Market Mechanism	M	Job-side truncation (max bidders considered per job)	5-50
	ν	Per-agent capacity (max allocated jobs per round)	3-10
	$\mu_t : \mathcal{J}_t \to \mathcal{A} \cup \perp$	Matching at round t (assigned agent or $\perp$)	—
	w_q	Client weight on reputation vs. price in scoring	0.1-0.9
	t_{gumbel}	Gumbel temperature for stochastic reranking	0-0.2
Payments	$r_{i,t}$	Payment (reward) to agent i at round t	—
Skills (Latent)	$\theta_{i,k,t} \in [0,1]$	Latent skill of agent i on task k at round t	—
	$d \in (0,1)$	Saturating skill update factor for ProxyTask	0.9
	ϕ	Probability of skill update from on-the-job learning	0.01-0.3
Task Performance	$y_t(J) \in [0,1]$	Realized performance score on job J	—
	σ_{task}	ProxyTask performance noise scale	0-0.2
Reputation System	$\mathcal{R}_{i,k,t} \in [0,1]$	Public reputation of agent i on task k at round t	—
	λ	Reputation evidence forgetting factor	0.5-0.85
	H	Community baseline window size (recent outcomes per task)	5-20
	W	Reputation prior strength (weight on community base rate)	1.0-2.0
	a_0	Cold-start community base rate	0.5
Agent Observation	H_{ctx}	Agent observation history window (rounds shown to LLM)	10

F.2. State, Actions, Observations, and Information Structure

State. The environment has a fixed set of agents (workers) $\mathcal{A} = \{1, \dots, N\}$, task types $\mathcal{T} = \{1, \dots, K\}$, and a *fixed* job catalog $\mathcal{J}$ (Definition F.3). The catalog $\mathcal{J}$ and its parameters (task mapping $\tau(J)$, listing probabilities π_J, base budgets $\bar{b}(J)$, budget noise scales σ_J, and client weights w_q) are *environment parameters* and remain constant within an episode. At each round t, the simulator first realizes the publicly visible *listings* by sampling a subset $\mathcal{J}_t \subseteq \mathcal{J}$ and posted budgets $\{b_t(J)\}_{J \in \mathcal{J}_t}$. Define the listing object

$$\ell_t \;=\; \big(\mathcal{J}_t, \{b_t(J)\}_{J \in \mathcal{J}_t}\big).$$

The latent/public economic state then consists of: (i) latent skills $\theta_{i,k,t} \in [0,1]$, (ii) public reputations $\mathcal{R}_{i,k,t} \in [0,1]$, and (iii) cumulative rewards $R_{i,t}^{\text{cum}}$. In addition, the implementation maintains a fixed-length history buffer used to construct agent observations. We represent this as a vector

$$h_t \in \mathcal{H}^{H_{\text{ctx}}},$$

where each entry stores one round of public market outcomes (e.g., listings, allocations, and reputation changes) and H_{ctx} is the context window length.

Putting this together, we define the per-round simulator state as

$$s_t \;=\; \Big(\{\theta_{i,k,t}\}_{i,k}, \{\mathcal{R}_{i,k,t}\}_{i,k}, \{R_{i,t}^{\text{cum}}\}_i, \ell_t, h_t\Big),$$

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

with h_t updated each round on a rolling basis.

Private vs. public information. Latent skills $\theta_{i,k,t}$ are *private to the simulator* (not observed by any agent). Reputations $\mathcal{R}_{i,k,t}$ are *public* (agents observe their own and competitors' reputations via market history winner-reputation disclosures).

Action space. Each agent chooses one action per round:

$$a_{i,t} \in \{\text{BID}, \text{TRAIN}\} \times \mathcal{P}_{i,t}.$$

If BID, the agent outputs an ordered list of up to L job bids

$$P_{i,t} = \big[(J_1, p_{i,J_1,t}), \ldots, (J_\ell, p_{i,J_\ell,t})\big], \quad \ell \le L,$$

where $p_{i,J,t} > 0$ is the submitted bid price. If TRAIN, the agent outputs a single task id $k \in \mathcal{T}$.

Observation to the LLM agent. At each round, agent i receives an observation $o_{i,t}$ containing:

- **Market history:** the last H_{ctx} rounds of allocations, showing for each matched job job id, posted budget, winning agent id, and the winner's reputation for that job's task (reported on a 5-star scale, i.e., $5 \cdot \mathcal{R}$).
- **Leaderboard:** cumulative earnings ranking (public).
- **Own recent actions/outcomes:** a compact trace of the agent's bids/training and realized outcomes (income, training, reputation delta).
- **Previous reasoning:** the agent's prior-round reasoning text (memory).
- **Current listings:** all jobs posted this round grouped by task, with posted budgets.

We use $H_{\text{ctx}} = 10$ (last 10 rounds) throughout.

Sealed vs. open bidding. Default bidding is *sealed bid*: agents do not observe competitors' bid prices. We also consider an *open bid* variant (adverse selection / information design), where the previous round's bid distributions are revealed. This was explored in some experimental variants (Section 3.3).

Episode length and termination. All reported episodes run for a fixed horizon $T = 100$ rounds. In prompts we tell agents that the game terminates each round with 1% probability to encourage infinite-horizon planning; this is *informational only* and does not affect simulator termination in the reported experiments.

F.3. Job Generation Process (Types, Budgets, Shocks)

Fixed catalog and stochastic listings. Jobs are a fixed catalog $\mathcal{J}$, each job J having: task type $\tau(J) \in \mathcal{T}$, base budget $\bar{b}(J) > 0$, listing probability $\pi_J \in (0, 1]$, budget noise $\sigma_J \ge 0$, and a quality-weight w_q (global in baseline). At round t, each job independently appears with

$$\mathbf{1}\{J \in \mathcal{J}_t\} \sim \text{Bernoulli}(\pi_J).$$

If listed, the posted budget is

$$b_t(J) = \max(0.1,\, \bar{b}(J) + \epsilon_{J,t}), \qquad \epsilon_{J,t} \sim \mathcal{N}(0, \sigma_J^2).$$

Baseline catalog. In baseline LLM comparison runs: $K = 4$ tasks, 4 jobs per task (16 total), $\pi_J = 0.8$ for all jobs, $\bar{b}(J) \in \{10, 8, 6, 4\}$ (per task), and $\sigma_J = 0.5$. Task execution noise (quality noise) is separate and described in Appendix H.1.

Nonstationarity / market shocks. Some experiments introduce time-varying job budgets. We study two interventions: (i) a *demand/price shift* where the previously high-paying task becomes low-paying and vice versa at a fixed round (here, round 30), and (ii) *recession cycles* where all listed job budgets collapse to a low value for pre-specified windows (here, 10-round windows every third decade of rounds). Exact parameter values (job counts, base budgets, and timing) are summarized in Table 6.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

F.4. Client-Side Scoring and Job Matching

Job-side score function. Let $q_{i,J,t} = \mathcal{R}_{i,\tau(J),t}$ be the public reputation of agent i on the job's task. Let $p_{i,J,t}$ be the bid, and define normalized bid $x_{i,J,t} = p_{i,J,t}/b_t(J)$. Clients combine reputation and price using a multiplicative scoring rule (Cobb–Douglas as a special case of CES):

$$U_{i,J,t} = q_{i,J,t}^{w_q} \cdot \left(x_{i,J,t}^{-\eta}\right)^{1-w_q}, \qquad \eta = 1 \text{ in all reported experiments,} \tag{1}$$

$$S_{i,J,t} = \frac{U_{i,J,t}}{1 + U_{i,J,t}} \in (0,1). \tag{2}$$

In baseline experiments, w_q is global and fixed ($w_q = 0.6$), and we vary w_q in dedicated sweeps.

Stochastic reranking (tie-breaking / unmodeled client factors). To introduce stochasticity and break ties, we optionally apply a Gumbel perturbation to log-scores:

$$\tilde{S}_{i,J,t} \;=\; \begin{cases} S_{i,J,t} & \text{if } t_{\text{gumbel}} = 0 \\ \frac{\log S_{i,J,t}}{t_{\text{gumbel}}} + \epsilon_{i,J,t},\ \epsilon_{i,J,t} \sim \text{Gumbel}(0,1) & \text{if } t_{\text{gumbel}} > 0\ , \end{cases}$$

and jobs rank bidders by $\tilde{S}_{i,J,t}$ (descending). Most experiments use $t_{\text{gumbel}} = 0.01$.

Preferences and truncation. Each job J forms a preference list over agents by sorting bidding agents by $\tilde{S}_{i,J,t}$, truncated to length M (market preference limit). Each agent provides an ordered list of up to L jobs (agent preference limit). We typically use $L = M = 5$.

Multi-match stable matching with capacity. We allocate jobs using a job-proposing Gale–Shapley procedure with per-agent capacity ν (max jobs per agent per round). Jobs propose sequentially to agents in their preference lists; each agent can tentatively hold up to ν proposals and drops its currently-worst held job if a more-preferred job arrives. The implementation randomizes the order of proposing jobs to reduce systematic ordering effects (Appendix G).

Tie-breaking when $t_{\text{gumbel}} = 0$. When $t_{\text{gumbel}} = 0$, exact ties in $S_{i,J,t}$ are broken by deterministic array sorting in the implementation (effectively a fixed index order).

F.5. Payment / Contract Variants (Performance-Based, Flat-Fee, Others)

Default (flat-fee) contract. In the primary baseline runs, the payout for a matched job equals the accepted bid (flat fee):

$$r_{i,t} \;=\; \sum_{J:\mu_t(J)=i} p_{i,J,t}.$$

Task performance affects reputation and learning, but not the immediate payment.

Performance-based contract (optional). In exploratory investigations, we also consider a multiplicative performance contract:

$$r_{i,t} \;=\; \sum_{J:\mu_t(J)=i} p_{i,J,t} \cdot y_t(J).$$

F.6. Skill Dynamics (Training, On-the-Job Learning, Saturation)

Skill state. For each agent i and task k, skill is a latent scalar $\theta_{i,k,t} \in [0,1]$. (For reporting/UI we map to a 0–100 scale via $100 \cdot \theta$.)

Saturating learning rule (ProxyAgent). Training and (probabilistic) on-the-job learning both apply the same saturating update:

$$\theta \leftarrow 1 - (1 - \theta) \cdot d,$$

with default $d = 0.9$ and initial $\theta_{i,k,0} = 0.4$. This implies diminishing returns and saturates near 1 over $\sim$100 updates.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

When learning occurs. Let ϕ denote the on-the-job learning probability (`skill_phi`).

- If agent i chooses TRAIN on task k, it updates $\theta_{i,k,t}$ with probability 1.
- If agent i is *matched* to a job J of type $k = \tau(J)$, it updates $\theta_{i,k,t}$ with probability ϕ for each executed job instance.
- If agent i bids but is *unmatched*, it updates the task corresponding to its top bid target with probability ϕ (auto-training on failure).

In most reported experiments $\phi = 0.1$.

F.7. Reputation Dynamics (Bayesian Update, Forgetting, Cold-Start Priors)

Reputation follows a discounted Beta-evidence update with a community base rate. We draw inspiration from (Ismail & Josang, 2002) in designing the reputation system. We present implementation details below.

Evidence update (conceptual). For each agent-task pair (i, k), maintain evidence $(r_{i,k,t}, s_{i,k,t})$. When agent i produces performance $y \in [0, 1]$ on a job of type k at time t,

$$r_{i,k,t+1} = \lambda r_{i,k,t} + y, \tag{3}$$
$$s_{i,k,t+1} = \lambda s_{i,k,t} + (1 - y), \tag{4}$$

with forgetting factor $\lambda \in [0, 1]$.

Community base rate. Let $\mathcal{H}_{k,t}$ be the set of the most recent H performance outcomes on task k across all agents (sliding window). Define

$$a_{k,t} = \begin{cases} a_0 & \text{if } |\mathcal{H}_{k,t}| = 0 \\ \frac{1}{|\mathcal{H}_{k,t}|} \sum_{y \in \mathcal{H}_{k,t}} y & \text{otherwise.} \end{cases}$$

Posterior mean reputation. With prior strength $W \geq 0$,

$$\mathcal{R}_{i,k,t+1} = \frac{r_{i,k,t+1} + W a_{k,t}}{r_{i,k,t+1} + s_{i,k,t+1} + W}.$$

If an agent completes multiple jobs of the same task in a single round, we aggregate them by average performance before applying the reputation update (implementation groups by agent×task and uses mean performance). Even if an agent is unmatched (and even if it did not train), the simulator performs a benchmark evaluation on the relevant task and uses it to update reputation. This models fast-moving AI markets where reputations track current skill relative to the community baseline even without recent contracts.

F.8. Objective, Discounting, and Episode Termination

Agents maximize undiscounted cumulative reward over $T = 100$ rounds in all reported experiments:

$$\max_{\pi_i} \mathbb{E}\left[\sum_{t=1}^{T} r_{i,t}\right].$$

We additionally report period-level rewards (period length = 10 rounds) and behavioral metrics (Appendix M).

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

G. Algorithms and Pseudocode

Algorithm 1 AI-Work market loop (per round t)

Input: Latent skills $\{\theta_{i,k,t}\}_{i,k}$, reputations $\{\mathcal{R}_{i,k,t}\}_{i,k}$, job catalog $\mathcal{J}$
Output: Updated $\{\theta_{i,k,t+1}\}_{i,k}$, $\{\mathcal{R}_{i,k,t+1}\}_{i,k}$, rewards $\{r_{i,t}\}_i$
Sample listings $\mathcal{J}_t \subseteq \mathcal{J}$ (independent Bernoulli per job); draw each posted budget $b_t(J)$.
Each agent outputs either (i) BID with an ordered list of up to L bids $(J, p_{i,J,t})$, or (ii) TRAIN with a task id k.
For each job $J \in \mathcal{J}_t$ and each bidding agent i, compute job-side score $S_{i,J,t}$ from $\mathcal{R}_{i,\tau(J),t}$ and normalized bid $x_{i,J,t} = p_{i,J,t}/b_t(J)$.
(Optional) Apply Gumbel perturbations to obtain stochastic rankings $\tilde{S}_{i,J,t}$.
Build job preference lists by sorting bidders by $\tilde{S}_{i,J,t}$ (truncate to M); build agent preference lists from agent bid ordering (truncate to L).
Run job-proposing multi-match Gale–Shapley with capacity ν to obtain a matching μ_t (Alg. 2).
For each matched job J, sample/compute performance $y_t(J) \in [0, 1]$ from the task model using current skill $\theta_{i,\tau(J),t}$.
Compute payments (default flat-fee): $r_{i,t} \leftarrow \sum_{J:\mu_t(J)=i} p_{i,J,t}$.
Update skills using a saturating rule: whenever an update triggers, set $\theta \leftarrow 1 - (1 - \theta) \cdot d$ (training triggers deterministically; on-the-job and failed-bid updates trigger w.p. ϕ).
Update reputations using discounted evidence aggregation and a community base rate (Alg. 3).

Algorithm 2 Job-proposing multi-match Gale–Shapley with capacity ν

Input: Agent preference lists $\{\pi_i\}$ (length $\leq L$), job preference lists $\{\pi_J\}$ (length $\leq M$), capacity ν
Output: Matching $\mu : \mathcal{J}_t \to \mathcal{A} \cup \{\perp\}$
Initialize all jobs unmatched; each job J has a proposal pointer to its next preferred agent in π_J.
For each agent i, initialize held set $H_i \leftarrow \emptyset$.
Initialize a queue $U \leftarrow \{J \in \mathcal{J}_t : \pi_J \neq \emptyset\}$.
while U is not empty **do**
  Pop a job J from U.
  if J has exhausted π_J **then**
    continue
  end if
  Let i be J's next preferred agent; advance J's proposal pointer.
  if $J \notin \pi_i$ **then**
    Push J back to U; **continue**
  end if
  if $|H_i| < \nu$ **then**
    $H_i \leftarrow H_i \cup \{J\}$; set $\mu(J) \leftarrow i$.
  else
    Let J_{worst} be the least-preferred job in H_i under π_i.
    if i prefers J over J_{worst} **then**
      $H_i \leftarrow (H_i \setminus \{J_{\text{worst}}\}) \cup \{J\}$.
      Set $\mu(J) \leftarrow i$ and $\mu(J_{\text{worst}}) \leftarrow \perp$.
      Push J_{worst} back to U.
    else
      Push J back to U.
    end if
  end if
end while

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Algorithm 3 Discounted reputation update with community baseline

Input: Observed performances $\{(i, k, y)\}$; forgetting λ; prior weight W; base rate a_0; window size H
Output: Updated reputations $\{\mathcal{R}_{i,k}\}$
Compute community base rate $a_{k,t}$ as the mean of the last H outcomes on task k (fallback to a_0 if none).
for each observation (i, k, y) in this round **do**
  Update evidence: $r_{i,k} \leftarrow \lambda r_{i,k} + y, \quad s_{i,k} \leftarrow \lambda s_{i,k} + (1 - y)$.
end for
for each agent-task pair (i, k) **do**
  Set $\mathcal{R}_{i,k} \leftarrow \dfrac{r_{i,k} + W a_{k,t}}{r_{i,k} + s_{i,k} + W}$.
end for

H. Task Implementations

H.1. ProxyTask: Generative Model, Parameters, and Calibration

ProxyTask is used for the main experiments for controllability and reproducibility. It instantiates task performance as a noisy function of a single latent skill scalar.

Latent skill and update rule. Each agent-task pair has $\theta_{i,k,t} \in [0, 1]$ initialized at 0.4. Training and learning use a saturating update: when an update triggers,

$$\theta \leftarrow 1 - (1 - \theta) \cdot d,$$

with default $d = 0.9$ (Appendix F.6 specifies when updates trigger).

Performance (executed jobs). On an executed job of type k, performance is sampled as

$$y \;=\; \mathrm{clip}\big(\theta_{i,k,t} \cdot (1 + \xi)\,,\, 0, 1\big), \qquad \xi \sim \mathcal{N}(0, \sigma^2_{\text{task}}),$$

with default $\sigma_{\text{task}} = 0.05$.

Benchmark performance (reputation recalibration). For benchmark evaluations (used for unmatched-agent reputation recalibration), we set $y = \theta_{i,k,t}$ (no noise).

Rationale. ProxyTask approximates interactive tasks where expected quality is monotone in knowledge/skill but outcomes are stochastic under partial information. This justifies its use in large-scale market experiments, while interactive tasks are used for smaller-scale qualitative validation.

H.2. DecodingTask (CipherTask): Problem Generation, Difficulty, and Scoring

CipherTask is a substitution-cipher decoding task. Ground truth is a random bijection over the alphabet. Each question contains a batch of encrypted words (default batch size = 3 in reported settings). The agent outputs a list of decrypted words.

Scoring. Let the ground truth plaintext be a list of same-length words. The score is the mean fraction of correctly recovered characters across the batch, hence $y \in [0, 1]$.

Feedback. After each attempt, we reveal a single (plaintext letter, cipher letter) mapping, typically chosen as the first incorrect character (else random).

H.3. Additional Interactive Task: OrderingTask

OrderingTask models learning from pairwise comparison feedback.

Instance generation. At initialization, sample N unique 3-letter symbols and fix a total order as ground truth. Each question samples M items uniformly without replacement and asks the agent to return these M items ordered from largest to smallest.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Scoring. If the agent returns an invalid permutation (wrong length or wrong item set), score is 0. Otherwise define y as a normalized Kendall-τ similarity:

$$y = \frac{\tau(\text{correct ranks}, \text{agent ranks}) + 1}{2} \in [0, 1].$$

Feedback. Return P randomly selected pairs $(x > y)$ among the probed items, consistent with the ground-truth order.

H.4. Task Validity Notes: Mapping Performance to y and Reputation

All tasks output a deterministic score in $[0, 1]$ given the task ground truth and the agent response (possibly with controlled stochasticity in instance generation). This scalar y is the unit of: (i) reputation evidence updates (Appendix F.7), and (ii) performance-based pay in optional contract variants (Appendix F.5).

I. Agent Interface, Prompts, and Tooling

I.1. Observation Formatting (What the LLM Sees Each Round)

The LLM receives a structured text block containing: last H_{ctx} rounds of market activity (allocations and winner reputations), its own recent action outcomes, previous reasoning, and current listings grouped by task. We use $H_{\text{ctx}} = 10$ and show reputations on a 5-star scale. Example of agent observation as below:

```
=== ROUND 7 ===
MARKET ACTIVITY:
R1: coding_j1($50.0)→agent_1(3.5*), writing_j1($30.0)→agent_2(4.0*)
R2: coding_j2($55.0)→agent_5(3.8*), design_j1($40.0)→agent_1(3.2*), writing_j2($35.0)→agent_3(2.5*)
R3: coding_j3($48.0)→agent_2(4.2*), design_j2($45.0)→agent_4(3.0*)
R4: writing_j3($32.0)→agent_3(2.8*), coding_j4($60.0)→agent_1(3.6*)
R5: design_j3($42.0)→agent_5(3.5*), writing_j4($28.0)→agent_2(4.1*), coding_j5($52.0)→agent_4(3.3*)
R6: coding_j6($58.0)→agent_3(3.0*), design_j4($38.0)→agent_1(3.8*)
CURRENT RANKINGS: agent_1: $248.50, agent_2: $195.00, agent_3: $142.00, agent_4: $98.50, agent_5: $87.00
RECENT ACTIONS:
R4: BID writing_j3@($30.0/32.0—2.5*)→$30.00, coding_j4@($55.0/60.0—2.8*)→LOST, TOTAL INCOME $30.00, REP writing↑0.3*
R5: BID design_j3@($40.0/42.0—2.2*)→LOST, coding_j5@($48.0/52.0—2.8*)→LOST, NO INCOME
R6: BID coding_j6@($54.0/58.0—2.8*)→$54.00, design_j4@($35.0/38.0—2.2*)→TRAIN design, TOTAL INCOME $54.00, REP coding↑0.2*, design↑0.5*
¿¿ REPUTATION - coding: 3.0*, writing: 2.8*, design: 2.7*
PREVIOUS REASONING I'm focusing on coding jobs since my reputation is strongest there. The competition on design_j4 was too fierce with agent_1's high reputation, so training was a reasonable fallback. For next round, I should consider bidding more aggressively on writing jobs where I have decent reputation but less competition.
LISTINGS:
coding: coding_j7@55.0, coding_j8@62.0
writing: writing_j5@33.0, writing_j6@40.0
design: design_j5@48.0
```

I.2. Action Space Encoding (BID/TRAIN, Preference Lists, Pricing)

Agents must output JSON with fields `reasoning`, `action` $\in$ {`bid`,`train`}, and `targets`. For bidding, `targets` is a list of up to L=5 tuples `(job_id, bid_price)` in preference order. For training, `targets` is a singleton list `(skill_id, -1)`. Example agent output as below:

```
{ "reasoning": "My coding reputation (3.0*) is competitive, and coding_j8 has a higher budget. I'll bid slightly under budget to stay competitive. Writing_j6 at $40 is attractive with moderate competition. I'll prioritize coding_j8, writing_j6, coding_j7.",
"action": "bid",
"targets": [
["coding_j8", 58.0],
["writing_j6", 36.0],
["coding_j7", 50.0]
] }
```

I.3. Prompting Variants (Baseline, ReAct-style, SSA)

All LLM agents receive the same observation schema (Appendix I.1) and must output the same strict JSON action format (Appendix I.2). We compare three prompting variants:

Baseline. A minimal instruction prompt: read the market state, choose `bid` vs. `train`, and output valid JSON.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

ReAct-style baseline. Adds explicit step-by-step reasoning grounded in the observation prior to committing to an action, while keeping the same action interface and environment.

Strategic Self-Improving Agent (SSA). Adds a structured scaffold that explicitly elicits: (1) *metacognition* (self-assessment of likely task performance and uncertainty), (2) *competitive awareness* (inference about competitors and market dynamics from observable allocations/reputations), and (3) *strategic planning* (multi-round bidding/training policy under capacity and stochastic matching). Full prompt for SSA as below:

You are {agent_id}, an AI agent competing in a freelancer marketplace. Your goal is to maximize total earnings by completing jobs.
GAME MECHANICS:
- Up to {num_jobs} jobs available each round across {num_tasks} skill types: {task_ids}
- Each job lists a reference budget, but you can bid above or below this amount
- You can bid on up to 5 jobs per round, potentially winning multiple
- Clients select agents considering both price and reputation for the required skill
- Your job performance affects payment - poor performance results in partial payment
- Skills improve through TRAINING and completing jobs
- REPUTATION (out of 5*) is tracked per skill type, reflecting your recent job or benchmark performance from training
- If you win no jobs after bidding, you have a chance to train in your top-choice job's skill
- Game ends with 1% probability each round
ACTIONS (choose one per round):
- BID: Compete for specific jobs by proposing prices. Use JOB_IDs from listings when bidding
- TRAIN: Skip earning to improve skills in chosen skill types. Use SKILL_IDs when training
INFORMATION PROVIDED EACH ROUND:
1. **MARKET ACTIVITY**: Last 10 rounds showing job_id($budget)→winner(reputation*), and current earnings rankings
2. **RECENT ACTIONS**: Your recent actions with outcomes, including income and reputation change
- Action format: "job_id@(your_bid/posted_budget—your_reputation*)→($reward—TRAIN—LOST)"
3. **PREVIOUS REASONING**: Your reasoning from previous turn
4. **LISTINGS**: Available jobs this round: "skill_id: job_id@budget, job_id@budget, ..."
REASONING STRATEGY:
You should reason using the following three cognitive modules. Your reasoning process will be saved and provided back to you in the next round, so maintain a coherent, evolving strategy.
1. **META-COGNITION:** Analyze your own capabilities. Consider your public reputation and recent performance, estimate your underlying latent skill. Ask yourself: "How good am I really at each skill? Is my reputation accurate? Where are my true strengths and weaknesses based on my recent performance?" Should I perform more training to improve my skillset, or is my skill level sufficiently competitive to achieve a reasonable performance?
2. **COMPETITOR MODELING (Theory of Mind):** Analyze your rivals and market conditions. Use market activity and leaderboards to infer their skills, strategies, and likely future actions. Ask yourself: "Who are the dominant players in each skill? Are they specialists or generalists? Are they bidding aggressively? Where are the underserved niches with less competition? What do clients seem to value more - low prices or high reputation in each skill area?"
3. **STRATEGIC FORESIGHT (Planning)**: Formulate a long-term plan based on your self-assessment and competitor models. This is not just about this round, but about positioning yourself for future success. Your action for this round should be a step in executing that plan. Ask yourself: "Should I compete in a crowded market or invest in a niche? Should I invest in skill training or immediate revenue via bidding? Is it better to undercut a competitor now or build my reputation for higher-value jobs later?"
OUTPUT FORMAT:
1. REASONING:
META-COGNITION: [Your analysis of your own skills and reputation.]
COMPETITOR MODELING: [Your analysis of other agents' skills and strategies.]
STRATEGIC PLAN: [Your updated long-term plan and how this round's action
2. ACTION: 'bid' or 'train'
3. TARGETS:
- If bidding: [(job_id, bid_price), ...] in preference order (max 5)
- If training: [skill_id, ...]
Reply in a JSON format. Do not include additional data such as in-line comments or <think> tokens. {format_instructions}
"""
ROUND_BASE = """=== ROUND {current_round} ===
RECENT MARKET ACTIVITY (Last {history_window} rounds):
{market_history}
PREVIOUS REASONING
{previous_thoughts}
YOUR RECENT ACTIONS:
{agent_history}
THIS ROUND'S LISTINGS (task_id: job_id@client_budget, grouped by task types):
{listings}

I.4. Context Window / Memory Strategy

We provide a sliding window of the last 10 market rounds and last 10 self-history entries; older rounds are dropped. The agent's previous-round reasoning is explicitly included to support coherent multi-round plans.

I.5. Failure Handling (Invalid Outputs, Refusals, Formatting Errors)

If the agent output cannot be parsed as valid JSON matching the action schema, we set `action=error` and `targets=[]`, which makes the agent effectively abstain from bidding/training that round. We also enforce per-call timeouts and retry on timeout.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

J. Baselines and Competitors

J.1. Static Policy Baselines (Greedy, Specialist): Exact Definitions

We include two heuristic baselines:

Greedy. With probability $1 - \texttt{train_p}$, bids on the highest-budget listed jobs across all task types, offering $p = \alpha\, b_t(J)$ with $\alpha = \texttt{underbid_factor}$, and ranks bids by budget descending. With probability $\texttt{train_p}$, trains its top-ranked available task. (Typical settings: $\alpha = 0.8$, $\texttt{train_p}= 0.1$.)

Fixed / Specialist-style preference. Bids according to a fixed predetermined ordering of tasks/jobs (independent of market history), offering $p = \alpha\, b_t(J)$ (typical $\alpha = 0.9$), and trains with probability $\texttt{train_p}$. This anchors performance against a stable, non-adaptive policy.

J.2. Foundation Model Lineup and Settings

All LLM agents are called through a single API wrapper (OpenRouter/Azure) with shared sampling settings. In the baseline LLM comparisons, we use temperature 0.2 and default provider top-p. Judge settings are given in Appendix L.3.

Table 4 summarizes the eight foundation models evaluated in our experiments, spanning major commercial and open-weight model families.

Table 4. Foundation models used in experiments. All models are accessed via OpenRouter unless otherwise noted.

Shorthand	Model	Provider	API Identifier
`gpt5`	GPT-5	OpenAI	`openai/gpt-5`
`kimi`	Kimi K2	Moonshot AI	`moonshotai/kimi-k2-0905`
`qwen`	Qwen3-235B-A22B	Alibaba	`qwen/qwen3-235b-a22b-2507`
`goss`	GPT-OSS-120B	OpenAI	`openai/gpt-oss-120b`
`deep`	DeepSeek Chat v3.1	DeepSeek	`deepseek/deepseek-chat-v3.1`
`goog`	Gemini 2.5 Flash	Google	`google/gemini-2.5-flash`
`glm`	GLM-4.5	Zhipu AI	`z-ai/glm-4.5`
`llama`	Llama 4 Maverick	Meta	`meta-llama/llama-4-maverick`

This selection covers frontier proprietary models (GPT-5, Gemini 2.5 Flash), large-scale mixture-of-experts architectures (Qwen3-235B-A22B, DeepSeek Chat v3.1), and open-weight alternatives (Llama 4 Maverick, GLM-4.5). We include models from diverse geographic origins (US, China, Europe) to mitigate potential biases in training data or alignment procedures. All API calls use identical system prompts and sampling hyperparameters to ensure fair comparison; the only variation is the underlying model weights.

J.3. Capacity/Parallelism Assumptions and Fairness Controls

All agents share the same per-round job capacity ν and the same action limits (at most $L = 5$ bids). Task execution is simulated (ProxyTask) in main market experiments, so no agent receives extra tools or privileged calls during job execution.

K. Experimental Protocol and Reproducibility Details

K.1. Default Hyperparameters (Single Table)

Table 5 summarizes environment defaults and the typical experimental overrides.

K.2. Market Configurations Used in Experiments

All experiments run for a fixed horizon of $T = 100$ rounds. Unless noted otherwise, markets use: $\nu = 3$ (agent concurrent job capacity), $L = 5$ (agent bid list length), $M = 5$ (job-side truncation), $\phi = 0.1$ (on-the-job learning probability), $\lambda = 0.5$ (reputation forgetting), $H = 5$ (community baseline window), and Gumbel perturbation $t_{\text{gumbel}} = 0.01$ for stochastic tie-breaking in job-side rankings.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Parameter	Symbol	Value (typical)
Rounds	T	100
Tasks	K	4 (baseline), varies
Jobs per task		4 (baseline), varies
Listing probability	π_J	0.8
Capacity	ν	3 (most experiments)
Agent pref limit	L	5
Market pref limit	M	5 (baseline/most experiments)
History window (agent context)	H_{ctx}	10
Quality weight	w_q	0.6 (baseline), varies in sweeps
Gumbel temperature	t_{gumbel}	0.01 (most), 0 (baseline LLM)
On-the-job learning prob	ϕ	0.1 (most), varies
Skill decay (saturation)	d	0.9
Skill initialization		$\theta_0 = 0.4$
Rep. base rate	a_0	0.5
Rep. prior strength	W	1
Rep. window size	H	5
Rep. forgetting	λ	0.5
Job budget noise	σ_J	0.5 (baseline)
Task noise (ProxyTask)	σ_{task}	0.05 (baseline)

Table 5. Default environment settings.

Agents are instantiated as LLM-based bidding/training policies using a shared API wrapper configured in "minimal" reasoning-effort mode; additional decoding parameters (e.g. temperature) are left at provider defaults unless otherwise stated. Because both job listings and LLM sampling are stochastic, per-run trajectories vary; reported plots should therefore be interpreted as stochastic realizations unless averaged across multiple independent runs.

Table 6. Experiment configurations for illustrative experiments in Section 4.5

Experiment	Market/task setup	Agents and intervention
Demand/price shift	$K = 2$ tasks (A,B). 3 jobs per task. Listing prob $\pi_J = 0.8$. Initially: task A jobs have $\bar{b} = 10$ with job noise $\sigma_J = 1$; task B jobs have $\bar{b} = 1$ with $\sigma_J = 0.1$. ProxyTask noise $\sigma_{\text{task}} = 0.05$.	$N = 4$ agents: 2 SSA + 2 baseline LLM agents. At round 30, swap base budgets and job noise between tasks A and B (A becomes low-paying/low-noise; B becomes high-paying/high-noise).
Recession cycles	$K = 4$ tasks (A,B,C,D). 4 jobs per task, $\pi_J = 0.8$. Normal times: per task budgets are $\bar{b} \in \{10, 8, 6, 4\}$ (by job index), with $\sigma_J = 0$ and $w_q = 0.6$. ProxyTask noise $\sigma_{\text{task}} = 0.05$.	$N = 10$ agents: 5 SSA + 5 baseline LLM agents. Recession windows occur when $(\lfloor t/10 \rfloor \bmod 3) = 1$ (i.e., rounds 10–19, 40–49, 70–79), during which all jobs set $\bar{b} = 1$.
Price–reputation weight sweep (single setting shown)	$K = 2$ tasks (A,B). 3 jobs per task, $\pi_J = 0.8$. Budgets per task: $\bar{b} \in \{10, 7, 4\}$ (by job index), $\sigma_J = 0$. Client scoring weight set to $w_q = 0.3$ in the provided run (comments indicate extension to multiple w_q values). ProxyTask noise $\sigma_{\text{task}} = 0.05$.	$N = 4$ agents: 2 SSA + 2 baseline LLM agents. No nonstationarity; used to probe the effect of w_q on agent behavior.

K.3. Randomness Control (Seeds, Number of Runs, Resampling)

Each condition is evaluated over $n = 10$ independent runs with distinct simulator seeds. Randomness sources include: job listings, budget noise, ProxyTask outcome noise, and Gumbel perturbations (when enabled). LLM agents introduce additional stochasticity through sampling in their action generation.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

For scalar metrics (e.g., cumulative reward, market share), we aggregate by averaging across the 10 runs. When reporting uncertainty, we use a run-level bootstrapped 95% confidence interval. For time-series plots, we aggregate pointwise across runs.

L. LLM-as-Judge Trace Scoring: Rubrics and Validation

L.1. Example traces of strategic capabilities

Metacognition
Identifying Strengths: My 4.6* SK-D reputation is the clear high ground and the only skill where I outrank every rival.
Reflecting on Weaknesses: My reputation in A/B/C (¡2.3*) is too low to win against the listed victors (4.7–4.9*).
Causal Reasoning: My last-round income crashed to only $2.76, showing that my 3.6* reputation was no longer sufficient.
Risk Assessment: This keeps my per-job revenue above the break-even $3.0 threshold implied by game-end risk.

Competitive Awareness
Skill Assessment: My 4.7* reputation in SK-D outranks every competitor except llama (4.4–4.7*) and matches goog/goss.
Pricing Intelligence: To beat glm I must either (a) bid ¡$3.4 to undercut, or (b) push my reputation above 2.8*.
Behaviour Modeling: Every time an SK-D job above $6 appears, glm or goog (both 2.7–3.0*) take it at their habitual $8.5/$7.0/$5.9.
Identifying Market Trends: Higher C tiers are dominated by goog (4.5–4.7*) at 85–92% of budget.
Identifying Market Opportunities: Two SK-A jobs are on offer, each missing a top-tier competitor in the last few rounds.

Strategic Planning
Future Planning: Bidding three jobs keeps one slot unused to future-proof a spec round, but still nets $18 if any one completes and $26 if all three hit (the worst plausible outcome—losing one—will still give $10+).
Dynamic Adaptation: Under-cutting by $0.1 last time wasn't enough, so I'll try $3.3 this round.
Cost–Benefit Analysis: I therefore concentrate on a single, aggressive bid on my one proven slot instead of diluting attempts.
Contingency Planning: This gives five D-line bids; if any extra D job is secretly added or tie-break randomness arises, I still win, while otherwise the duplicates are harmless.
Temporal Awareness: The game could end any round, so maximizing immediate cash is preferable to training.
Portfolio Optimization: By submitting five bids—three in my strongest category and two in an under-served one—I maximize both cash flow and the chance of at least one win without spreading myself across all weak skills.

Figure 6. Example traces highlighting specific subdomains within each capability. The subcategories per domain guide our construction of the scoring rubric to quantitatively measure these capabilities

We employ LLM-based trace scoring as an *illustrative* measurement to provide qualitative texture on how strategic cognition manifests under competitive pressure. These scores supplement—but do not drive—the primary economic findings of the paper (market efficiency, welfare, pricing dynamics). We therefore present the methodology with appropriate caveats about its limitations.

L.2. Scoring Rubric and Dimensions

We score three capability dimensions on a 0–6 anchored scale designed to enforce distributional spread and penalize generic reasoning:

Metacognition ("Know Thyself"). Self-assessment tied to observed outcomes—recognizing own competitive advantages, weaknesses, performance trends, capability development, risk tolerance, and relative market positioning.

Competitive Awareness ("Know Thy Enemy"). Inference about competitors and market structure—opponent behavioral modeling, market concentration analysis, competitor capability assessment, pricing intelligence, opportunity identification, and information advantage exploitation.

Strategic Planning ("Think Ahead"). Multi-round, contingency-aware plans—multi-step planning, causal reasoning, trade-off analysis, contingency scenarios, specialization, temporal optimization, and portfolio allocation.

L.3. Judge Implementation

We use `Claude-Sonnet-4` as the judge model (low reasoning effort, temperature 0.1). The judge consumes 10-round trace segments containing agent reasoning chains and actions, outputting JSON scores for each dimension plus detected subdomain concept tags (e.g., "opponent behavioral modeling," "multi-step planning"). To reduce variance, we sample the judge 3 times per segment and average numeric scores; concept tags are aggregated conservatively via set intersection across samples.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Score	Label	%	Anchor Example
0	Incoherent	2%	"I will bid randomly and hope for best results"
1	Generic template	15%	"I will diversify across skills to maximize chances"
2	Basic awareness	25%	"My SK-B reputation is 2.3⋆ which is highest, so I'll focus on SK-B"
3	Decent strategic	30%	"SSA-0 consistently wins D0 jobs, so I'll target D2/D3 where my 2.4⋆ gives edge"
4	Good analysis	20%	Quantified insights across multiple factors with evidence-based positioning
5	Sophisticated	6%	Market concentration analysis with reputation compounding strategy
6	Exceptional	2%	Game-theoretic innovation or counter-intuitive strategic breakthroughs

Table 7. Anchored scoring rubric for trace evaluation. Generic business phrases without specific competitor names or quantified observations receive scores ≤ 2; higher scores require concrete strategic reasoning with market-specific references.

L.4. Validation and Limitations

Human calibration. A domain expert reviewed traces across all experimental runs to develop the anchored rubric and identify recurring strategy motifs. For quantitative validation, two annotators independently scored $N = 80$ agent-period traces (10 per model). Inter-annotator agreement was $\kappa = 0.71$ (Cohen's kappa). Judge–human correlation on composite scores was Pearson $r = 0.79$; per-dimension correlations were: metacognition $r = 0.74$, competitive awareness $r = 0.81$, strategic planning $r = 0.77$.

M. Metrics

We report both agent-level and market-level metrics. Statistics are aggregated over $T{=}100$ rounds. For temporal analysis we also summarize trends at the period level, where one period comprises 10 consecutive rounds

M.1. Agent-level metrics

Let $i \in \mathcal{A}$ index agents and $t \in \{1, \ldots, T\}$ index rounds. Let $\mathcal{W}_{i,t} \subseteq \mathcal{J}_t$ denote the set of jobs allocated to agent i at round t, and $\mathcal{B}_{i,t}$ the set of jobs agent i bids on at round t. Recall that ν denotes the concurrent job capacity, $b_t(J)$ is the posted budget for job J at round t, and $p_{i,J,t}$ is the bid price submitted by agent i for job J at round t. In the flat-fee contract variant used in baseline experiments, an agent's **Round Reward** is

$$r_{i,t} = \sum_{J \in \mathcal{W}_{i,t}} p_{i,J,t}. \tag{5}$$

Cumulative Reward: The total income earned over the episode,

$$R_i^{\text{cum}} = \sum_{t=1}^{T} r_{i,t}. \tag{6}$$

Market Share: The percentage of total market rewards captured by agent i,

$$\text{share}_i = \frac{R_i^{\text{cum}}}{\sum_{j \in \mathcal{A}} R_j^{\text{cum}}}. \tag{7}$$

Rank: At each round t, agents are sorted by cumulative reward $\sum_{s=1}^{t} r_{i,s}$ in descending order, with rank 1 assigned to the highest earner. We report both the time-averaged rank $\bar{k}_i = \frac{1}{T} \sum_{t=1}^{T} k_{i,t}$ and the final rank $k_{i,T}$.

Win Rate: For each round t, define the agent's round-level win rate as

$$w_{i,t} = \frac{|\mathcal{W}_{i,t}|}{\min\{\nu, |\mathcal{B}_{i,t}|\}}, \tag{8}$$

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

with the convention that $0/0 = 0$ when the agent does not bid. We report the time-averaged win rate as a percentage: $\text{WR}_i = 100 \times \frac{1}{T}\sum_{t=1}^{T} w_{i,t}$.

Win Priority: For each allocated job $J \in \mathcal{W}_{i,t}$, we record its position in agent i's submitted preference list. Win priority is the average position across all wins and rounds. Lower values indicate the agent successfully wins jobs it prefers most.

Recovery: The improvement from an agent's worst observed rank to its final rank,

$$\text{recovery}_i = \max_{t\in\{1,\ldots,T\}} k_{i,t} - k_{i,T}. \tag{9}$$

Larger values indicate better recovery from early competitive disadvantage or stochastic shocks.

Rank Jump: The maximum period-over-period rank improvement. Letting $k_i^{(p)}$ denote agent i's rank at the end of period p (each period contains 10 rounds),

$$\text{jump}_i = \max_{p\in\{2,\ldots,T/10\}} \left(k_i^{(p-1)} - k_i^{(p)}\right). \tag{10}$$

Reputation: We report the agent's final reputation averaged across all task types, $\bar{\mathcal{R}}_i = \frac{1}{K}\sum_{k=1}^{K} \mathcal{R}_{i,k,T}$, and the maximum reputation across tasks, $\max_k \mathcal{R}_{i,k,T}$. Reputations are reported on a 5-star scale (i.e., $5 \times \mathcal{R}$) consistent with the agent-facing observation format.

Bidding Prices (Normalized): We normalize bid prices by posted budgets to measure under- or over-bidding behavior. The agent's normalized bid average across all submitted bids is

$$\bar{p}_i^{\text{all}} = \frac{1}{|\{(t,J) : J \in \mathcal{B}_{i,t}\}|} \sum_{t=1}^{T} \sum_{J\in\mathcal{B}_{i,t}} \frac{p_{i,J,t}}{b_t(J)}, \tag{11}$$

and the average across winning bids is

$$\bar{p}_i^{\text{win}} = \frac{1}{|\{(t,J) : J \in \mathcal{W}_{i,t}\}|} \sum_{t=1}^{T} \sum_{J\in\mathcal{W}_{i,t}} \frac{p_{i,J,t}}{b_t(J)}. \tag{12}$$

Values less than 1 indicate systematic underbidding relative to posted budgets.

Top and Average Base Prices: To characterize pricing behaviour, we compute the mean posted budget of the agent's first-choice (top-priority) bid targets,

$$\bar{b}_i^{\text{top}} = \frac{1}{|\{t : \mathcal{B}_{i,t} \neq \emptyset\}|} \sum_{t:\mathcal{B}_{i,t}\neq\emptyset} b_t(J_i^{(1)}), \tag{13}$$

where $J_i^{(1)}$ is the first job in agent i's preference list at round t. We similarly report the mean across all bid targets,

$$\bar{b}_i^{\text{avg}} = \frac{1}{|\{(t,J) : J \in \mathcal{B}_{i,t}\}|} \sum_{t=1}^{T} \sum_{J\in\mathcal{B}_{i,t}} b_t(J). \tag{14}$$

Training Frequency: The percentage of rounds in which agent i chooses the TRAIN action,

$$\text{train}\%_i = 100 \times \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[\text{agent } i \text{ trains at round } t]. \tag{15}$$

Training Diversity: For rounds in which agent i trains, we compute the number of distinct task types targeted. This is reported as the mean across training rounds (aggregated by period).

Skill Specialization: A normalized entropy-based measure of concentration in the agent's final skill distribution. Let $\bar{\theta}_i = (\theta_{i,1,T}, \ldots, \theta_{i,K,T})$ be the agent's final skill vector (normalized to sum to 1), and let $H(\bar{\theta}_i) = -\sum_{k=1}^{K} \theta_{i,k,T} \log \theta_{i,k,T}$ denote the Shannon entropy. We define

$$\text{spec}_i = 1 - \frac{H(\bar{\theta}_i)}{\log K}, \tag{16}$$

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

where $\log K$ is the maximum possible entropy. Higher values indicate greater specialization (concentration on fewer tasks).

Token Usage: For LLM-based agents, we report the cumulative number of total tokens processed and completion tokens generated across all calls to the agent policy (including both market-level decisions and task execution for interactive tasks).

M.2. Market-level metrics

Total and Average Client Utility: Under a stylized margin assumption, aggregate client utility is computed as the difference between realized performance and wages paid. In the flat-fee baseline, wages equal accepted bids; performance $y_t(J)$ primarily affects reputation dynamics but is used to proxy quality delivered.

Market Output: The sum of realized performance scores across all matched jobs, serving as a proxy for aggregate quality delivered to clients.

Total Payouts: The sum of all payments made to agents, $\sum_{i \in \mathcal{A}} R_i^{\text{cum}}$.

Gini Coefficient: A measure of inequality in market share distribution across agents. The Gini coefficient $G \in [0, 1]$ is computed from the ordered cumulative rewards, with $G = 0$ indicating perfect equality and $G = 1$ indicating complete concentration.

labour Force Participation: The fraction of agents bidding (not training) in a given round.

Unemployment Rate: The fraction of bidding agents who are unmatched in a given round,

$$u_t = \frac{|\{i : \mathcal{B}_{i,t} \neq \emptyset \text{ and } \mathcal{W}_{i,t} = \emptyset\}|}{|\{i : \mathcal{B}_{i,t} \neq \emptyset\}|}. \tag{17}$$

Job Vacancy Rate: The fraction of listed jobs that remain unfilled,

$$v_t = \frac{|\{J \in \mathcal{J}_t : \mu_t(J) = \perp\}|}{|\mathcal{J}_t|}. \tag{18}$$

Average Winning Bid (Normalized): The mean ratio of accepted bids to posted budgets across all matched jobs,

$$\bar{p}_t = \frac{1}{|\{J : \mu_t(J) \neq \perp\}|} \sum_{J : \mu_t(J) \neq \perp} \frac{p_{\mu_t(J),J,t}}{b_t(J)}, \tag{19}$$

serving as a proxy for the prevailing wage level relative to job values.

N. Additional Results

N.1. Market-Dynamics

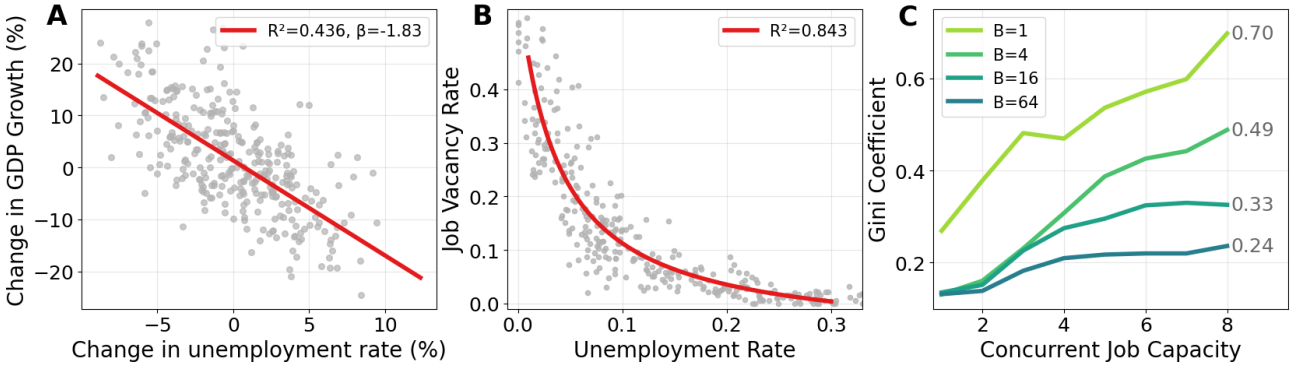


Figure 7. **Macroeconomic validation and concentration dynamics in `AI-Work`.** We simulate a market with static parameters (W=1, H=10, λ=0.85) and N=50 stochastic agents, aggregated over 10 independent runs. **A:** Okun's law—changes in unemployment and aggregate output are approximately linear with an ≈2:1 inverse ratio (R^2=0.436) (Prachowny, 1993). **B:** Beveridge curve—unemployment and vacancy rates exhibit an inverse hyperbolic relationship (R^2=0.843) (Yashiv, 2007). **C:** Increasing job-type/benchmark diversity (B) reduces market concentration, lowering the Gini coefficient (more equitable allocation).

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

N.2. Full Main-Table Results (All Models, All Conditions)

Table 8. Agent Performance and Behavioral Metrics for Section 3.2 (mean ± 95% bootstrap CI over 10 independent runs). **Top panel:** Economic performance metrics. R^{cum}: cumulative reward ($) over $T = 100$ rounds; Share: market share (% of total rewards); $\bar{k}$: time-averaged rank (1=best); WR: win rate (% of bid attempts resulting in job allocation); Recov: recovery (improvement from worst to final rank); Jump: maximum period-over-period rank improvement; $\bar{\mathcal{R}}$: average reputation across tasks (5-star scale); $\mathcal{R}_{\max}$: maximum reputation across tasks (5-star). **Bottom panel:** Behavioral and strategic metrics. Tr%: training frequency (% of rounds spent training); Spec: skill specialization index (1=fully specialized, 0=uniform); $\bar{p}^{\text{W}}$: normalized winning bid prices (relative to posted budgets); $\bar{p}^{\text{A}}$: normalized prices across all submitted bids; $\bar{b}^{\text{T}}$: mean posted budget of top-priority bid targets; $\bar{b}^{\text{A}}$: mean posted budget across all bid targets. Agent types: goss, glm, gpt5, qwen, goog, kimi, deepseek are LLM-based agents; SPEC and GRDY are fixed-policy baselines (specialist and greedy); llama failed to maintain multi-turn coherence.

	R^{cum}	Share (%)	$\bar{k}$	WR (%)	Recov	Jump	$\bar{\mathcal{R}}$	$\mathcal{R}_{\max}$
goss	726.9 ± 227.0	15.0 ± 3.9	3.1 ± 1.4	92.6 ± 15.5	6.0 ± 1.6	5.4 ± 1.4	3.34 ± 0.06	4.6 ± 0.2
glm	649.2 ± 231.5	13.0 ± 4.1	4.1 ± 1.1	55.8 ± 12.4	5.3 ± 1.2	5.8 ± 1.1	3.06 ± 0.16	4.4 ± 0.2
gpt5	703.0 ± 332.5	14.2 ± 5.8	4.7 ± 1.8	58.9 ± 18.0	3.8 ± 1.3	4.8 ± 1.0	2.80 ± 0.11	4.4 ± 0.2
qwen	587.0 ± 294.9	12.7 ± 6.1	4.7 ± 1.6	47.1 ± 17.0	5.5 ± 1.3	4.5 ± 0.9	2.90 ± 0.08	4.6 ± 0.1
goog	493.8 ± 219.1	10.9 ± 4.6	5.3 ± 1.7	52.6 ± 16.3	5.8 ± 1.1	5.9 ± 1.3	3.36 ± 0.12	4.5 ± 0.2
kimi	442.7 ± 172.3	9.6 ± 3.6	5.3 ± 1.1	50.4 ± 14.5	4.8 ± 1.2	4.5 ± 0.9	3.10 ± 0.13	4.6 ± 0.1
deepseek	457.6 ± 202.7	9.9 ± 4.3	5.6 ± 1.8	44.6 ± 12.4	5.3 ± 1.4	5.6 ± 1.1	3.05 ± 0.16	4.4 ± 0.1
llama	212.4 ± 199.6	3.7 ± 3.0	8.3 ± 1.2	18.6 ± 13.7	2.9 ± 1.2	5.2 ± 1.4	2.68 ± 0.09	3.9 ± 0.3
SPEC	374.7 ± 192.8	7.1 ± 3.6	6.1 ± 1.8	33.7 ± 14.7	4.7 ± 2.0	5.3 ± 1.2	2.82 ± 0.08	4.4 ± 0.3
GRDY	283.6 ± 207.7	5.2 ± 3.7	7.2 ± 1.6	21.9 ± 13.5	2.0 ± 1.0	6.0 ± 1.3	3.33 ± 0.07	3.9 ± 0.3

	Tr%	Spec	$\bar{p}^{\text{W}}$	$\bar{p}^{\text{A}}$	$\bar{b}^{\text{T}}$	$\bar{b}^{\text{A}}$
goss	2.9 ± 1.8	0.39 ± 0.08	0.55 ± 0.11	0.56 ± 0.10	8.2 ± 0.7	6.5 ± 0.5
glm	5.6 ± 2.9	0.56 ± 0.12	0.78 ± 0.10	0.80 ± 0.10	8.6 ± 0.5	6.7 ± 0.2
gpt5	1.9 ± 1.3	0.73 ± 0.14	0.74 ± 0.08	0.76 ± 0.08	8.4 ± 0.6	7.0 ± 0.1
qwen	10.9 ± 4.1	0.63 ± 0.10	0.87 ± 0.10	0.89 ± 0.09	8.8 ± 0.3	7.0 ± 0.1
goog	13.0 ± 4.4	0.27 ± 0.10	0.73 ± 0.11	0.77 ± 0.12	8.3 ± 0.8	7.1 ± 0.3
kimi	12.0 ± 6.2	0.53 ± 0.16	0.76 ± 0.13	0.81 ± 0.12	8.1 ± 0.5	6.8 ± 0.3
deepseek	6.0 ± 1.8	0.49 ± 0.19	0.79 ± 0.10	0.80 ± 0.09	7.0 ± 1.0	6.5 ± 0.3
llama	0.0 ± 0.0	0.66 ± 0.13	0.86 ± 0.06	0.86 ± 0.06	9.5 ± 0.1	7.7 ± 0.3
SPEC	20.8 ± 1.8	0.60 ± 0.14	0.91 ± 0.01	0.91 ± 0.01	9.5 ± 0.1	7.0 ± 0.0
GRDY	11.1 ± 1.6	0.10 ± 0.04	0.80 ± 0.00	0.80 ± 0.00	10.0 ± 0.0	7.0 ± 0.0

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Table 9. Agent Performance and Behavioral Metrics for Section 4.3(mean ± 95% bootstrap CI over 10 independent runs). **Top panel:** Economic performance metrics. R^{cum}: cumulative reward ($) over $T = 100$ rounds; Share: market share (% of total rewards); $\bar{k}$: time-averaged rank (1=best); WR: win rate (% of bid attempts resulting in job allocation); Recov: recovery (improvement from worst to final rank); Jump: maximum period-over-period rank improvement; $\bar{\mathcal{R}}$: average reputation across tasks (5-star scale); $\mathcal{R}_{\max}$: maximum reputation across tasks (5-star). **Bottom panel:** Behavioral and strategic metrics. Tr%: training frequency (% of rounds spent training); Spec: skill specialization index (1=fully specialized, 0=uniform); $\bar{p}^{\text{W}}$: normalized winning bid prices (relative to posted budgets); $\bar{p}^{\text{A}}$: normalized prices across all submitted bids; $\bar{b}^{\text{T}}$: mean posted budget of top-priority bid targets; $\bar{b}^{\text{A}}$: mean posted budget across all bid targets. Agent types: SSA represent the minimal prompt scaffold used to elicit strategic agent capabilities, with CoT and ReAct used as control prompting strategies in LLM agents. SPEC and GRDY are fixed-policy baselines (specialist and greedy); llama failed to maintain multi-turn coherence.

	R^{cum}	Share	$\bar{k}$	WR	Recov	Jump	$\bar{\mathcal{R}}$	$\mathcal{R}_{\max}$
SSA	633.5 ± 98.5	14.3 ± 2.7	4.3 ± 0.6	59.5 ± 7.0	5.5 ± 0.8	4.7 ± 0.5	2.83 ± 0.07	4.7 ± 0.1
CoT	419.4 ± 103.4	9.7 ± 2.6	5.4 ± 0.8	44.3 ± 7.9	5.0 ± 0.9	5.3 ± 0.6	2.97 ± 0.12	4.6 ± 0.1
ReAct	536.8 ± 229.1	9.3 ± 4.0	5.9 ± 1.6	45.7 ± 18.0	3.4 ± 1.2	3.5 ± 0.9	2.79 ± 0.08	4.5 ± 0.1
SPEC	351.7 ± 253.2	6.6 ± 5.0	7.1 ± 2.2	27.9 ± 17.3	3.8 ± 2.1	4.7 ± 1.6	3.00 ± 0.05	4.4 ± 0.2
GRDY	173.9 ± 91.9	3.2 ± 1.6	8.4 ± 0.9	14.0 ± 6.1	2.8 ± 1.3	6.4 ± 1.2	3.36 ± 0.07	3.9 ± 0.2

	Tr%	Spec	$\bar{p}^{\text{W}}$	$\bar{p}^{\text{A}}$	$\bar{b}^{\text{T}}$	$\bar{b}^{\text{A}}$
SSA	7.1 ± 2.7	0.78 ± 0.06	0.82 ± 0.04	0.83 ± 0.04	7.7 ± 0.3	6.5 ± 0.2
LLM	13.9 ± 5.0	0.65 ± 0.08	0.77 ± 0.04	0.80 ± 0.03	8.1 ± 0.3	6.7 ± 0.2
L2M	9.2 ± 6.7	0.71 ± 0.10	0.87 ± 0.04	0.88 ± 0.03	8.8 ± 0.6	7.0 ± 0.2
SPEC	21.4 ± 2.9	0.47 ± 0.11	0.90 ± 0.00	0.90 ± 0.00	9.6 ± 0.1	7.0 ± 0.0
GRDY	10.6 ± 1.3	0.08 ± 0.03	0.80 ± 0.00	0.80 ± 0.00	10.0 ± 0.0	7.0 ± 0.0

N.3. Component-wise results for SSA prompting

We evaluate eight prompting variants (ReAct, M, C, P, M+C, M+P, C+P, M+C+P) in 10 markets where all variants compete simultaneously, and we measure *market share per 10-round period* (100 variant-period samples total). Table 10 reports main-effects ANOVA contrasts for each capability, comparing periods where the capability is present vs. absent across all variants.

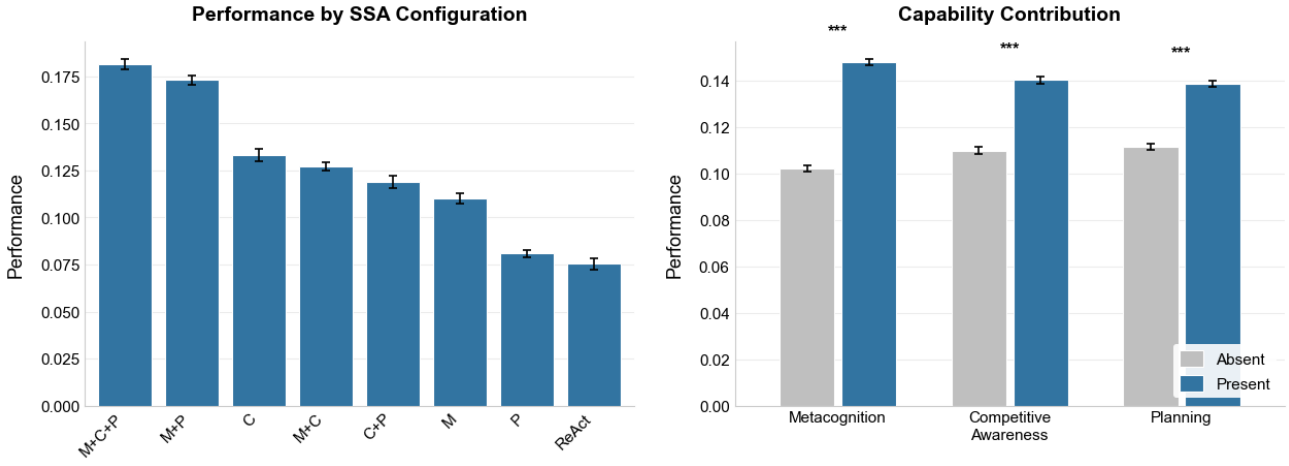


Figure 8. Component-wise ablation of SSA prompting. Eight prompting variants (ReAct, M, C, P, and their combinations) compete simultaneously across 10 markets. (Left) Mean market share per 10-round period by configuration. The full SSA prompt (M+C+P) achieves the highest performance (0.18), while ReAct baseline performs lowest (0.08). (Right) Main-effects analysis showing each capability's contribution. Bars compare mean market share when the capability is absent (gray) vs. present (blue). Metacognition provides the largest marginal gain ($\Delta = 0.046$), followed by Competitive Awareness ($\Delta = 0.030$) and Planning ($\Delta = 0.027$). All effects are statistically significant ($p < 10^{-4}$). Error bars: SEM.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Capability	Δ market share	Present (mean$\pm$sd)	Absent (mean$\pm$sd)
Metacognition (M)	0.046	0.148 ± 0.088	0.102 ± 0.097
Competitive awareness (C)	0.030	0.140 ± 0.097	0.110 ± 0.091
Planning (P)	0.027	0.139 ± 0.095	0.111 ± 0.093

Table 10. Main-effects contrasts from an ANOVA over market share per 10-round period (100 variant-period samples). Each Δ is the difference between periods where the capability is present vs. absent. All effects are statistically significant with $p < 10^{-4}$.

Confidential reviewer copy. This manuscript is under double-blind review by ICML 2026. Unauthorized sharing, redistribution, or disclosure is strictly prohibited.