# Reinforcement Learning and Agentic Trading in Double Auctions[*]

Pranjal Rawat[†]

November 21, 2025

PRELIMINARY DRAFT: NOT FOR CIRCULATION

## Abstract

In 1993, the Santa Fe Institute hosted a seminal tournament where simple ”sniping” heuristics outperformed complex trading algorithms in a continuous double auction. Thirty years later, we revisit this environment to investigate whether modern Deep Reinforcement Learning (PPO) and Large Language Models (GPT-4o) can solve the information aggregation problem without hard-coded rules. We faithfully replicate the original synchronized double auction mechanism and introduce a new generation of agents. Our results show that: (1) PPO agents autonomously rediscover the ”sniping” strategy, exploiting legacy heuristics; (2) Multi-agent PPO markets maintain high allocative efficiency, avoiding the market collapse observed in heuristic self-play; and (3) Zero-shot LLMs exhibit high efficiency but display distinct behavioral biases, prioritizing fairness over profit maximization. These findings suggest that while gradient-based learning can master market timing, semantic reasoning introduces a new, potentially stabilizing, dynamic to automated markets.

[*] Pranjal Rawat is grateful for the computational resources provided by Georgetown University and the original source code provided by Rust et al.

[†] PhD Candidate, Georgetown University. pp712@georgetown.edu

## 1 Introduction

How do decentralized markets coordinate the actions of self-interested agents without central authority? This question, first articulated by Hayek (1945) as the ”knowledge problem,” remains one of the most profound puzzles in economics. Hayek argued that no central planner could ever possess the dispersed, tacit knowledge held by millions of individual participants, yet markets routinely aggregate this information into prices that guide efficient resource allocation. The mechanism by which this coordination occurs, without explicit communication or shared intent, has been the subject of decades of experimental and computational inquiry.

The experimental investigation of this phenomenon began with Vernon Smith’s pioneering work in the 1960s, which demonstrated that simple market institutions, particularly the Double Auction, could reliably converge to competitive equilibrium even when participants possessed only private information about their own costs and valuations. This finding launched a research program that culminated in the 1990 Santa Fe Double Auction Tournament, where computer programs competed to maximize profits in an artificial market. The tournament produced a striking and paradoxical result: the winning strategy was not a sophisticated learning algorithm but a simple ”sniping” heuristic that exploited the information revealed by other traders. Yet this individually optimal strategy proved collectively destructive; markets composed entirely of snipers collapsed into illiquidity. The tension between individual rationality and collective efficiency, between structure and agency, between intelligence and stability, remains unresolved.

The rise of artificial intelligence has made these questions urgently relevant. Modern financial markets are increasingly dominated by algorithmic traders, and regulators have raised concerns about the potential for autonomous agents to discover tacit collusion or destabilize market function. At the same time, large language models have demonstrated surprising capabilities in reasoning and planning, raising the question of whether semantic knowledge alone can support effective economic behavior. This study revisits the foundational questions of the Santa Fe Tournament through the lens of modern AI, specifically Deep Reinforcement Learning and Large Language Models, to investigate whether these ”super-rational” and ”semantic” agents can solve the market coordination problem without hand-crafted heuristics, and what their success or failure reveals about the fundamental nature of price discovery.

The literature on market microstructure has generated several enduring debates that frame our investigation. First, there is the question of whether market efficiency is a property of the institution or the participants: the ”Zero-Intelligence” experiments of Gode and Sunder suggested that even random traders can achieve near-perfect allocative efficiency in a Double Auction, implying that the market structure itself does most of the computational work. Yet subsequent research revealed that while structure ensures efficiency, intelligence determines equity; zero-intelligence traders exhibited massive profit dispersion compared to human markets, suggesting that strategic sophistication is required for agents to secure their ”fair share” of the gains from trade. Second, there is the evolutionary dynamics of trading strategies: the Santa Fe Tournament and subsequent genetic programming experiments demonstrated that dominant strategies are not stable equilibria but nodes in an endless arms race, where each successful heuristic creates selection pressure for counter-strategies. Finally, there is the question of liquidity provision: the optimal individual strategy of waiting and sniping is parasitic, requiring other agents to reveal information and absorb adverse selection, yet a market of pure snipers collapses. These debates, reviewed in Section 2, provide the theoretical scaffolding for our empirical investigation.

### 1.1 Motivation and Broader Relevance

The motivation for this research is threefold. First, the "Hayek Hypothesis" that markets efficiently aggregate private information has traditionally been tested using human subjects or simple heuristic agents. It remains an open question whether the convergence properties observed in these studies are robust to agents with vastly superior computational capabilities (DRL) or broad semantic knowledge (LLMs). If advanced AI agents can disrupt market stability or uncover novel forms of algorithmic collusion, the theoretical underpinnings of market efficiency may need to be re-evaluated.

Second, the rise of automated trading has transformed financial markets from human-dominated ecosystems into arenas of algorithmic competition. However, most academic studies of algorithmic trading rely on proprietary data or complex, high-fidelity simulations that obscure the fundamental economic dynamics. By returning to the stylized, scientifically controlled environment of the Santa Fe Double Auction, we can isolate the effects of agent intelligence from market microstructure noise, providing clearer insights into the nature of algorithmic competition.

Third, there is a theoretical disconnect between the "Zero-Intelligence" view, which attributes efficiency solely to market structure, and the game-theoretic view, which requires sophisticated belief modeling. Modern AI offers a unique tool to probe this divide: DRL agents learn strategies from scratch without human priors, while LLMs bring a form of "common sense" reasoning to trading. Observing how these distinct forms of intelligence navigate the trade-off between liquidity provision and surplus extraction will deepen our understanding of price formation.

### 1.2 Research Questions

This study is guided by four primary research questions, designed to test the limits of both the market institution and the artificial agents.

The first question concerns whether Deep Reinforcement Learning can rediscover and outperform the dominant heuristics of the Santa Fe Tournament. The "Kaplan" strategy, a simple sniping heuristic, dominated the original 1990 tournament. We investigate whether a Proximal Policy Optimization (PPO) agent, starting with no prior knowledge of market rules or opponent strategies, can learn a policy that exploits Kaplan and other legacy algorithms. This probes whether the "sniping" behavior is a fundamental attractor of the strategy space or merely a local optimum of heuristic design.

The second question asks whether a market composed entirely of autonomous AI agents can remain stable. Previous work has shown that markets populated exclusively by sniping agents collapse due to a lack of liquidity. We examine whether a population of independent PPO agents, trained via self-play, can avoid this ”liquidity trap” and converge to a competitive equilibrium. Specifically, does the gradient-based learning process discover a mixed strategy of liquidity provision and taking that sustains market function, or does it devolve into algorithmic collusion?

Third, we investigate whether Large Language Models can trade effectively in a zero-shot setting. LLMs possess vast semantic knowledge but lack the specific iterative training of RL agents. We test whether a general-purpose model such as GPT-4o, provided only with a textual description of the market state and history, can execute profitable trading strategies. This addresses the ”semantic hypothesis”: that understanding the *context* of a market is sufficient for rational behavior, even without explicit optimization.

Finally, we ask how intelligence disparity affects wealth distribution. In a heterogeneous market populated by agents of varying cognitive capacities, we examine the extent of wealth transfer from less capable to more capable agents. This quantifies the ”value of intelligence” in a double auction and provides a proxy for the potential impact of AI disparity in real-world financial markets.

### 1.3 Hypotheses and Expected Outcomes

We formulate specific hypotheses corresponding to our research questions, grounded in the prior literature.

Regarding PPO behavior against legacy agents, we hypothesize that a single PPO agent trained against a diverse pool of legacy agents (ZI-C, Kaplan, ZIP) will converge to a ”sniping” strategy, characterized by withholding bids until the final moments of a trading period. We expect PPO to rediscover the optimal procrastination strategy identified by Chen and Yu (2011), likely executing it with greater precision than the static Kaplan heuristic. Consequently, we anticipate the PPO agent will achieve higher profits than any individual legacy opponent.

Concerning market stability under AI-only conditions, we hypothesize that a market composed entirely of PPO agents will maintain high allocative efficiency (greater than 95 percent), avoiding the market collapse observed in Kaplan-only markets. Unlike static heuristics, RL agents are capable of adapting to the aggregate state of the market. We expect that in self-play, PPO agents will learn to provide just enough liquidity to ensure trades occur, thereby avoiding the zero-volume outcome of the ”waiting game” equilibrium described by Wilson (1987). However, we also anticipate a secondary effect: PPO agents may learn to maintain wider bid-ask spreads than human traders, exhibiting a form of tacit algorithmic collusion.

With respect to LLM performance, we hypothesize that zero-shot LLM agents will achieve allocative efficiency comparable to human subjects but will underperform optimized RL agents. We expect LLMs to avoid the chaotic behavior of unconstrained zero-intelligence traders, demonstrating a baseline of economic rationality derived from their training data. However, without the specific feedback loops of reinforcement learning, they are unlikely to master the precise timing and order-book pressure tactics required to beat a trained PPO sniper.

Finally, regarding wealth distribution under intelligence disparity, we hypothesize that in a mixed market of GPT-4o and GPT-3.5 agents, the superior model will extract a disproportionate share of the surplus, with the wealth gap exceeding the difference in their allocative efficiency contributions. This posits that ”smarter” agents do not necessarily make the market more efficient; rather, they are more effective at rent-seeking. We expect GPT-4o to better identify and exploit the sub-optimal bids of GPT-3.5, resulting in a significant transfer of producer and consumer surplus.

### 1.4 Contributions

This work makes three distinct contributions to the literature on agent-based computational economics. First, it provides the first direct comparison of Deep Reinforcement Learning and Large Language Models within the rigorous, scientifically controlled environment of the Santa Fe Double Auction. By benchmarking these modern AI paradigms against the canonical ”Legacy Zoo” of trading heuristics (ZI-C, Kaplan, ZIP, GD), we establish a clear continuity between the experimental economics of the 1990s and the AI research of the 2020s.

Second, we offer a methodological contribution by creating a high-fidelity, open-source Python implementation of the Santa Fe tournament platform, integrated with modern MLOps standards (Gymnasium, Stable-Baselines3). This ”modernized testbed” lowers the barrier to entry for future research into AI market behavior, replacing the inaccessible or deprecated codebases of previous decades.

Finally, our analysis of the ”implicit markup” and ”belief functions” of PPO agents offers a novel interpretability framework for neural trading agents. By mapping the opaque policy networks of DRL back onto the economic theory of markups (Zhan and Friedman, 2007) and belief functions (Gjerstad and Dickhaut, 1998), we demystify the ”black box” of AI trading, showing that these agents are not learning alien strategies, but rather rediscovering and refining the fundamental economic principles of price discovery.

## 2 Literature Review

The study of price formation in decentralized markets has evolved through a dialectic between theoretical pessimism and empirical optimism. Early work focused on the impossibility of equilibrium without a central auctioneer, a view overturned by experimental evidence demonstrating the remarkable robustness of the Double Auction (DA) institution. This section traces the intellectual history from the first classroom experiments to the computational tournaments that set the stage for modern algorithmic trading.

### 2.1 Early Experimental Markets: From Chaos to Equilibrium

The experimental investigation of market behavior began with Chamberlin (1948), who sought to test the neoclassical theory of competitive equilibrium in a controlled setting. Chamberlin's design involved a decentralized bilateral bargaining process where students, acting as buyers and sellers with private reservation values, roamed a room to negotiate trades. Chamberlin observed that transaction prices fluctuated widely and the quantity traded consistently exceeded the competitive equilibrium prediction. He concluded that decentralized markets were inherently imperfect and that the theoretical intersection of supply and demand was an abstraction unlikely to be realized in practice without a mechanism for recontracting.

This conclusion was challenged and ultimately reversed by Smith (1962). Smith hypothesized that the inefficiency observed by Chamberlin was not due to the bounded rationality of the agents, but to the unstructured nature of bilateral bargaining. Smith introduced a centralized public clearing mechanism—the oral Double Auction—where bids and asks were announced to the entire market, and transactions occurred at publicly known prices. Under these rules, Smith observed rapid convergence to the competitive equilibrium price and quantity, often reaching allocative efficiencies exceeding 95% within a few trading periods. Crucially, this convergence occurred despite agents possessing only private information and no knowledge of the aggregate supply and demand schedules. Smith's findings validated the Hayekian hypothesis that market institutions serve as information aggregators (Hayek, 1945), demonstrating that the structure of the institution is a primary determinant of market efficiency. Notably, Smith observed that convergence followed a predictable directional path: in markets with excess supply, prices tended to start high and glide downward, while in markets with excess demand, prices started low and rose toward equilibrium, a pattern later confirmed by Cliff and Bruten (1997).

### 2.2 Theoretical Foundations: Heuristics and Game Theory

Following Smith's empirical success, theorists sought to explain *why* the Double Auction converges so reliably. Two distinct approaches emerged: game-theoretic equilibrium analysis and behavioral learning models.

Wilson (1987) provided the first rigorous game-theoretic treatment of the DA with incomplete information. Modeling the market as a multilateral sequential bargaining game, Wilson derived a sequential equilibrium in which traders adopt a "waiting game" strategy. Sellers with high costs and buyers with low valuations wait to reveal their offers, using delay as a credible signal of their private information. Wilson showed that as the number of traders increases, this strategic delay diminishes, and the market outcome converges asymptotically to the Walrasian equilibrium. However, Wilson's model relies on strong assumptions of common knowledge and sophisticated rationality that are difficult to justify in human subjects or simple software agents.

In contrast, Easley and Ledyard (1993) proposed a behavioral model based on simple adaptive heuristics. They assumed that traders do not optimize against the entire market state but instead adjust their ”reservation prices”—mental thresholds for bidding and asking—based on past success or failure. A trader who fails to transact becomes more aggressive (raising bids or lowering asks) in the next period, while a trader who transacts easily becomes more passive. Easley and Ledyard proved that these simple learning dynamics are sufficient to trap transaction prices within a corridor that converges to the competitive equilibrium, providing a robust explanation for Smith's results that does not require hyper-rationality.

Bridging these approaches, Friedman (1991) modeled the DA as a ”Game Against Nature,” where a rational trader treats the arrival of bids and asks as a stochastic process rather than the strategic output of opponents. Under this framework, Friedman derived an optimal ”aggressive reservation price” strategy, where traders shade their bids to maximize expected surplus. This strategy mathematically resembles the ”sniping” behavior predicted by Wilson's waiting game, suggesting a convergence between optimal control and game-theoretic predictions.

### 2.3 Zero-Intelligence and the The Santa Fe Tournament

The role of agent intelligence was radically questioned by Gode and Sunder (1993) in their seminal work on ”Zero-Intelligence” (ZI) traders. They simulated a DA market populated by algorithmic agents that submitted random bids and asks subject only to a budget constraint (ZI-C agents could not buy above their valuation or sell below cost). Surprisingly, these random agents achieved allocative efficiencies close to 100%, statistically indistinguishable from human markets. This finding, dubbed the ”Zero-Intelligence” result, implied that the allocative efficiency of the DA is largely an emergent property of the market rules (specifically the budget constraint and the public order book) rather than a product of trader learning or strategy. However, while ZI-C agents maximized the total market surplus, Gode and Sunder also found that profit dispersion among individual ZI traders was enormous compared to human markets; some agents earned far above their theoretical share while others earned almost nothing, suggesting that while market structure ensures allocative efficiency, individual intelligence is required to secure an equitable distribution of gains.

However, while ZI agents achieved high *efficiency*, they failed to extract surplus strategically. To investigate the limits of algorithmic trading, the Santa Fe Institute organized a Double Auction Tournament in 1990 (Rust et al., 1993, 1994). The tournament invited researchers to submit trading programs to compete in a synchronized discrete-time DA. The results were striking: simple heuristic strategies consistently outperformed complex learning algorithms (such as early neural networks). The tournament was won by the ”Kaplan” strategy (Kaplan, 1993), a simple sniper that waited in the background until the bid-ask spread narrowed before jumping in to steal the deal.

The computational investigation of market microstructure reached a watershed moment with the Santa Fe Double Auction Tournament, organized by Rust et al. (1993, 1994). Moving beyond the representative agent paradigm, the organizers invited researchers to submit diverse trading algorithms to compete in a ”Synchronized Double Auction”—a discrete-time approximation of the continuous market where agents simultaneously submit limit orders, followed by a trade execution phase governed by AURORA rules. The tournament field was highly heterogeneous, featuring strategies ranging from simple rule-based heuristics to sophisticated neural networks and genetic algorithms. Contrary to the expectations of the artificial intelligence community, the tournament was not won by a complex learning agent, but by a simple heuristic strategy submitted by Todd Kaplan. The "Kaplan" strategy functioned as a sniper: it remained passive in the background, observing the bid-ask spread, and only entered the market to "steal the deal" when the spread narrowed sufficiently or the trading period neared its conclusion. This parasitic strategy exploited the information revealed by more impatient traders (such as ZI-C or GD agents) while minimizing its own exposure to the winner's curse. However, Rust et al. engaged in a subsequent evolutionary analysis that revealed a profound paradox: while Kaplan agents dominated heterogeneous populations, a market composed entirely of Kaplan agents collapsed into a state of liquidity failure. With every agent waiting to snipe an offer that never materialized, transaction volume plummeted and allocative efficiency fell to approximately 50%. This "Kaplan deadlock" demonstrated that while sniping is locally optimal for an individual in a liquid market, it is globally unstable as a dominant strategy, underscoring the necessity of "noise traders" or impatience to lubricate the mechanism of price discovery.

Rust et al. noted a critical fragility in the Kaplan strategy: while it dominated heterogeneous markets, a market composed entirely of Kaplan agents collapsed. Since every agent waited for another to provide liquidity, trading volume plummeted, and efficiency dropped to approximately 50%. Beyond this liquidity failure, Rust, Palmer, and Miller decomposed the sources of inefficiency in agent-based markets and identified a phenomenon they termed extra-marginal displacement: aggressive traders with unfavorable cost or value positions could "steal" trades from more efficient intra-marginal traders, a dynamic distortion missed by static equilibrium theory. This "Kaplan deadlock" highlighted a fundamental tension between individual rationality (sniping) and collective efficiency (liquidity provision), a problem that remains central to the study of automated market makers and algorithmic trading today.

## 2.4 The Post-Tournament Era: Adaptation and Optimization

In the wake of the Santa Fe tournament, researchers sought to dissect why simple heuristics succeeded where complex models failed, and to push the boundaries of algorithmic performance. Cason and Friedman (1996) conducted a rigorous laboratory investigation to test three competing theoretical frameworks—Wilson's waiting game, Friedman's Bayesian model, and Gode and Sunder's zero-intelligence hypothesis—against human behavior. Crucially, they introduced a "random valuation" environment where trader values change every period, preventing the simple rote learning of a static equilibrium price. Their results confirmed that while zero-intelligence agents could explain the baseline efficiency of the Double Auction, they failed to capture the dynamics of transaction order and bid progressions observed in experienced human traders. As humans gained experience, their behavior shifted away from randomness toward the strategic patterns predicted by Bayesian models, suggesting that market efficiency is not merely a structural artifact but also a product of learning. This validates the use of learning algorithms like PPO in random-valuation environments, as they mimic the adaptive trajectory of human subjects.

Responding to the limitations of zero-intelligence, Cliff and Bruten (1997) introduced the "Zero-Intelligence Plus" (ZIP) agent. Cliff and Bruten demonstrated that while ZI-C agents achieve high efficiency in symmetric markets, they fail to converge to equilibrium prices in markets with asymmetric supply and demand schedules. To bridge this gap, they endowed agents with a simple adaptive mechanism based on the Widrow-Hoff delta rule. ZIP agents maintain a profit margin that they adjust heuristically: lowering margins to remain competitive when trades are scarce, and raising them to extract surplus when trades are frequent. This minimal adaptivity was sufficient to produce human-like price convergence in complex markets where ZI-C failed. For our research, ZIP represents a critical benchmark: a "behavioral" agent that learns scalar parameters (margins) rather than a full policy, providing a middle ground between random noise and deep reinforcement learning.

While ZIP focused on heuristics, Gjerstad and Dickhaut (1998) returned to the principles of optimization. They developed the "GD" strategy, which constructs a belief function estimating the probability that any given bid or ask will be accepted based on the recent history of market orders and transactions. The GD agent then chooses a price that maximizes its expected surplus against this belief function. In simulations, GD agents achieved near-perfect efficiency and converged to equilibrium faster than human subjects, establishing a new standard for algorithmic performance. The success of GD highlights the value of market history—specifically the order book and transaction log—as a state representation. This suggests that for a PPO agent to compete with or outperform GD, its observation space must include sufficiently rich historical features to implicitly reconstruct similar belief functions.

The algorithmic arms race continued with Tesauro and Das (2001), who introduced a modified version of the GD algorithm (MGD) and tested it in a realistic, continuous-time environment. They found that the original GD strategy could be volatile and unstable in certain market conditions. By adding heuristic stabilizations and extending the belief-based approach to handle persistent orders, their MGD strategy consistently outperformed both ZIP and the original Kaplan sniping strategy in head-to-head tournaments. Tesauro and Das demonstrated that while sniping (Kaplan) exploits naive agents effectively, it is vulnerable to sophisticated belief-based agents that can optimize their pricing dynamically. This finding poses a direct challenge to our PPO agents: to claim state-of-the-art performance, they must not only rediscover sniping but also demonstrate robustness against optimized belief-based strategies like MGD.

Finally, the potential for evolutionary discovery in these markets was explored by Chen and Tai (2010) and Chen and Yu (2011) using Genetic Programming (GP). Unlike previous approaches that hand-coded strategies, they allowed agents to evolve trading rules from basic mathematical and logical primitives. Their GP agents eventually discovered sophisticated "optimal procrastination" strategies that mirrored the Kaplan sniper but with greater adaptability to market shape. By analyzing the syntactic trees of the evolved agents, they found that the agents had learned to assess their competitive position and exercise monopsony power by withholding bids until the optimal moment. This confirms that the "sniping" behavior is not an artifact of a specific heuristic but a fundamental attractor in the strategy space of the Double Auction—one that we expect Deep Reinforcement Learning to rediscover and perhaps refine through gradient-based optimization.

## 3 The Market

We consider a synchronized double auction market populated by $N$ agents, indexed by $i \in \{1, \ldots, N\}$. The market operates in discrete time steps $t = 1, \ldots, T$ within a trading period. Each agent is endowed with a set of tokens, where buyers have private redemption values $v_{i,k}$ for the $k$-th unit, and sellers have private costs $c_{j,k}$.

The market state at time $t$ is defined by the limit order book, consisting of a set of outstanding bids $B_t = \{b_1, b_2, \ldots\}$ and asks $A_t = \{a_1, a_2, \ldots\}$. We denote the best (highest) bid as $b_t^* = \max B_t$ and the best (lowest) ask as $a_t^* = \min A_t$. The bid-ask spread is defined as $s_t = a_t^* - b_t^*$.

Following the specific rules of the Santa Fe Tournament Rust et al. (1994), the market proceeds in a synchronized two-phase step. In the Signaling Phase, all agents simultaneously observe the current book $(b_t^*, a_t^*)$ and may submit a new limit order. A buyer $i$ may submit a bid $b_{i,t} > b_t^*$ (improving the best bid) or $b_{i,t} = b_t^*$ (matching). Similarly, a seller $j$ may submit an ask $a_{j,t} < a_t^*$ or $a_{j,t} = a_t^*$. In the Clearing Phase, if the new orders cross (i.e., $b_{i,t} \geq a_{j,t}$), a transaction occurs immediately. The transaction price $p_t$ is determined by the standing order rule. In the Taking Phase, if no crossing occurs, agents are given a second opportunity to "take" the liquidity currently on the book. A buyer may accept $a_t^*$, or a seller may accept $b_t^*$.

### 3.1 Experimental Environments

Each experimental environment is a complete specification of market parameters. The number of buyers and sellers (up to 20 each) determines market structure. Each trader receives up to 4 tokens per period, with private values generated according to a gametype parameter that encodes four uniform random variable ranges using a base-3 coding scheme. The duration parameters specify rounds (up to 20), periods per round (up to 5), and time steps per period (up to 400). All programs receive common knowledge of these settings except gametype, which remains private.

Table 1 presents the ten canonical environments from the 1993 Santa Fe Tournament. These configurations systematically vary market structure, time pressure, and token endowments to stress-test trading algorithms across diverse conditions.

Table 1. Santa Fe Tournament Environments

| **Env** | **Description** | **Key Variation** | **gametype** |
|---|---|---|---|
| BASE | Standard | 4B/4S, 4 tokens, 3 periods, 75 steps | 6453 |
| BBBS | Buyer-dominated | 6 buyers, 2 sellers | 6453 |
| BSSS | Seller-dominated | 2 buyers, 6 sellers | 6453 |
| EQL | Equal endowment | Symmetric token values | 0 |
| RAN | Random | IID uniform draws | 6453 |
| PER | Single period | 1 period per round | 6453 |
| SHRT | High pressure | 25 steps per period | 6453 |
| TOK | Single token | 1 token per trader | 6453 |
| SML | Small market | 2 buyers, 2 sellers | 0007 |
| LAD | Low adaptivity | Same as BASE | 6453 |

## 3.2 Outcome Metrics

We construct the demand schedule $D(q)$ by ordering all buyer valuations $v_{ik}$ in descending order, and the supply schedule $S(q)$ by ordering all seller costs $c_{jk}$ in ascending order. The equilibrium quantity is $Q^* = \max\{q : D(q) > S(q)\}$, and the equilibrium price lies in the interval $S(Q^*) \leq P^* \leq D(Q^*)$, typically computed as the midpoint $P^* = (D(Q^*) + S(Q^*))/2$. The maximum theoretical surplus is $TS^* = \sum_{q=1}^{Q^*}(D(q) - S(q))$.

### 3.2.1 Market Efficiency

Allocative efficiency measures the percentage of maximum possible surplus realized:

$$E = \frac{\sum_{t=1}^{T}(v_t - c_t)}{TS^*} \times 100 \tag{1}$$

where $v_t$ and $c_t$ are the redemption value and cost of units exchanged at trade $t$. Efficiency loss decomposes into V-inefficiency (intra-marginal loss from untraded profitable units) and EM-inefficiency (extra-marginal loss from trades that should not have occurred):

$$IM = \sum_{q \in \text{Untraded Intra-marginal}} (D(q) - S(q)), \quad EM = \sum_{t \in \text{Extra-marginal}} (c_t - v_t) \tag{2}$$

### 3.2.2 Price Convergence

Root mean squared deviation measures distance from equilibrium:

$$RMSD = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(p_t - P^*)^2} \tag{3}$$

Smith's coefficient of convergence normalizes by equilibrium price: $\alpha = 100 \cdot RMSD/P^*$. Price volatility measures dispersion around the mean transaction price:

$$\text{Volatility} = \frac{\sigma_p}{\bar{p}} \times 100, \quad \text{where } \sigma_p = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(p_t - \bar{p})^2} \tag{4}$$

### 3.2.3 Trader Performance

Individual profit for buyers is $\pi_i = \sum_k (v_{ik} - p_k)$ and for sellers is $\pi_j = \sum_k (p_k - c_{jk})$. Equilibrium profit represents the theoretical profit if all trades occurred at $P^*$:

$$\pi_i^* = \sum_{k: v_{ik} > P^*} (v_{ik} - P^*) \text{ (buyers)}, \quad \pi_j^* = \sum_{k: c_{jk} < P^*} (P^* - c_{jk}) \text{ (sellers)} \tag{5}$$

The individual efficiency ratio $E_i = \pi_i / \pi_i^*$ measures whether a trader captures more ($E_i > 1$) or less ($E_i < 1$) than their equilibrium share. Profit dispersion measures cross-agent inequality:

$$PD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\pi_i - \pi_i^*)^2} \tag{6}$$

Lower dispersion indicates more equitable surplus allocation.

## 4 The Traders

This section describes all trading algorithms evaluated in this study, organized by complexity: zero-intelligence baselines, adaptive heuristics from the 1993 Santa Fe Tournament, and modern AI agents (reinforcement learning and large language models).

### 4.1 Zero-Intelligence Algorithms

### 4.1.1 Zero Intelligence (ZI)

Zero Intelligence (ZI) represents the simplest possible trading strategy, serving as a control condition from Gode and Sunder (1993). The agent generates bids and asks by drawing uniformly from the price range: $p \sim U[p_{min}, p_{max}]$ where $p_{min} = 1$ and $p_{max} = 1000$ in our implementation. Critically, ZI agents have no budget constraint and will accept any trade if selected as winner, regardless of profitability. This unconstrained randomness provides a baseline for measuring the contribution of strategic behavior to market efficiency. The algorithm contains no learning parameters and maintains no memory of previous trades.

### 4.1.2 Zero Intelligence Constrained (ZIC)

Zero Intelligence Constrained (ZIC) extends ZI by adding budget constraints that prevent unprofitable trades. For buyers with valuation $V$ for the current token, bids are generated as $b = V - \lfloor U[0,1) \times (V - p_{min}) \rfloor$ using floor truncation matching the original Java implementation. For sellers with cost $C$, asks follow $a = C + \lfloor U[0,1) \times (p_{max} - C) \rfloor$. The agent accepts trades only when profitable: buyers accept if $V > a_{ask}$ and sellers accept if $b_{bid} > C$, using strict inequalities that differ slightly from the theoretical formulation but match the 1993 baseline. This simple constraint dramatically improves efficiency without requiring any learning or strategic reasoning.

### 4.1.3 Zero Intelligence Plus (ZIP)

Zero Intelligence Plus (ZIP) adapts profit margins using the Widrow-Hoff delta rule from machine learning (Cliff and Bruten, 1997). The agent shouts at price $p = \lambda(1 + \mu)$ where $\lambda$ is the limit price (valuation for buyers, cost for sellers) and $\mu$ is the profit margin (negative for buyers, positive for sellers). The margin updates according to $\Delta(t) = \beta(\tau(t) - p(t))$ where $\tau$ is a target price, followed by momentum accumulation $\Gamma(t+1) = \gamma\Gamma(t) + (1-\gamma)\Delta(t)$. The target uses random perturbations based on recent market activity: $\tau = R \cdot q + A$ with $R \sim U[R_{min}, R_{max}]$ and $A \sim U[A_{min}, A_{max}]$ where $q$ is the last relevant shout price. Key parameters calibrated for AURORA markets: $\beta = 0.2$ (learning rate), $\gamma = 0.25$ (momentum coefficient), initial margin $\mu_0 = \pm 0.20$, with $R$ perturbations in $[0.95, 1.05]$ and $A$ perturbations in $[-0.05, 0.05]$. The algorithm responds to both accepted trades by raising margins when own price was far from transaction price and rejected orders by lowering margins when not competitive with market quotes.

### 4.1.4 Zero Intelligence Two (ZI2)

Zero Intelligence Two (ZI2) enhances ZIC by incorporating the current market bid and ask into the random price generation. For buyers facing current bid $b_{curr}$, if $b_{curr} > 0$ and $b_{curr} \leq V$, the agent narrows the random range to $b = V - \lfloor U[0,1) \times (V - b_{curr}) \rfloor$, effectively randomizing only above the standing bid. Sellers apply symmetric logic with current ask $a_{curr}$. When the current quote exceeds the agent's valuation (buyers) or falls below cost (sellers), the algorithm generates extreme quotes ($p_{min}$ or $p_{max}$) to signal inability to compete. This market-awareness allows ZI2 to adapt to trading activity without learning, though it retains zero intelligence in the sense of having no predictive model or memory across periods.

## 4.2 Santa Fe Tournament Algorithms

### 4.2.1 Gjerstad-Dickhaut (GD)

Gjerstad-Dickhaut (GD) forms probabilistic beliefs from historical data and maximizes expected surplus (Gjerstad and Dickhaut, 1998). For sellers choosing ask $a$, the probability of acceptance is $p(a) = [T_A(\geq a) + B(\geq a)]/[T_A(\geq a) + B(\geq a) + R_A(\leq a)]$ where $T_A(\geq a)$ counts accepted asks at or above $a$, $B(\geq a)$ counts all bids at or above $a$, and $R_A(\leq a)$ counts rejected asks at or below $a$. Buyers use the symmetric formulation $q(b) = [T_B(\leq b) + A(\leq b)]/[T_B(\leq b) + A(\leq b) + R_B(> b)]$ for bid $b$. The agent then maximizes expected surplus: sellers choose $a^* = \arg\max_{a \in [C, p_{max}]} p(a) \times (a - C)$ and buyers choose $b^* = \arg\max_{b \in [p_{min}, V]} q(b) \times (V - b)$ where $C$ is cost and $V$ is valuation. Implementation uses PCHIP (monotone cubic spline) interpolation to smooth the belief functions and maintains a memory of the last $L = 100$ trades. The agent accepts immediate trades only when certain surplus exceeds expected surplus from optimal quote.

### 4.2.2 Kaplan

Kaplan implements a strategic sniper strategy that waits for favorable conditions before jumping into the market (Rust et al., 1994). The algorithm tracks price history across periods, computing $\bar{p}$, $p_{min}$, and $p_{max}$ separately for each role. In the first bid or ask of each period, the agent uses the worst-case token value adjusted by market conditions. Subsequent quotes employ jump-in logic triggered by three conditions: small spread ($(\langle a_{curr} - b_{curr}\rangle/a_{curr} < 0.10$ for buyers), price better than last period ($a_{curr} \leq p_{min}$ for buyers or $b_{curr} \geq p_{max}$ for sellers), or time pressure measured as $(t - t_{last}) \geq (T - t)/2$ where $T$ is the period length. When jump-in triggers, buyers bid at the current ask and sellers ask at the current bid, though protection clauses prevent losses: $b_{new} \leq V - 1$ for buyers and $a_{new} \geq C + 1$ for sellers. In the buy-sell phase, the agent becomes a sniper in the final two timesteps, accepting any profitable trade.

### 4.2.3 Lin

Lin employs statistical price prediction using normal distribution sampling via the Box-Muller transform. The algorithm computes mean price $\bar{p} = \sum |prices|/n$ and standard error $\sigma = \sqrt{\sum(|p| - \bar{p})^2/(n-1)}$ from current period data, then extends this to a target price $\tau$ incorporating all previous periods: $\tau = (\bar{p}_{current} + \sum_{i=1}^{period-1} \bar{p}_i)/period$. To generate bids, the agent samples from a normal distribution $\mathcal{N}(\bar{p}, \sigma)$ using Box-Muller and combines this with a weighted average of conservative and target prices: $b_{new} = w \cdot (b_{curr} + 1) + (1 - w) \cdot \tau$ where the weight $w$ incorporates time pressure, inventory position, and market composition. The buy-sell decision uses threshold acceptance: buyers accept if $a_{curr} < \tau + \sigma$ and sellers accept if $b_{curr} > \tau - \sigma$. This statistical approach attempts to predict equilibrium prices from historical data without explicit belief formation.

### 4.2.4 Jacobson

Jacobson computes a weighted equilibrium estimate that gains confidence as trading progresses. On each trade, the algorithm updates $\tau_{eq} = \sum_{trades}(price \times weight)/\sum_{trades} weight$ where weight increases with both period number and trade count: $weight = period + n_{trades} \times \alpha$ with $\alpha = 2.0$. Confidence in this estimate follows an exponential function $conf = \beta^{1/\sum weight}$ where $\beta = 0.01$, approaching unity as total weight accumulates. Bids are generated as $b_{new} = b_{old} \cdot (1 - conf) + \tau_{eq} \cdot$

$conf + \delta$ where $\delta = 1.0$ is a bid-ask offset and $b_{old}$ is the current standing bid (or $p_{min}$ if none). Asks follow symmetric logic with negative offset. The buy-sell decision employs complex gap analysis: if spread $gap = a - b$ equals the previous gap or time pressure condition $(gap/(gap_{last} - gap)) \times n_{tokens} \times 2.0 + t > T$ holds, accept probabilistically with $prob = profit/(profit + gap)$. The four tunable hyperparameters allow adaptation to different market microstructures.

### 4.2.5 Perry

Perry implements adaptive learning with efficiency-based parameter self-tuning across periods. The core adaptive parameter $a_1$ scales with time pressure, market composition, and role imbalance: for buyers, $a_1 = a_0 \times (T - t)/T \times (N - 1)/N \times n_{sellers}/n_{buyers}$ where $a_0$ begins at 2.0 and adjusts based on period performance. After the first three conservative trades in each period, the algorithm uses statistical bidding: buyers bid $b = \bar{p} + 0.2\sigma - a_1\sigma + U[0, 1] \times 4s$ where $\bar{p}$ is mean price, $\sigma$ is standard deviation, and $s \in \{-1, +1\}$ is random, while sellers ask $a = \bar{p} + a_1\sigma + 20 \cdot U[0, 1]$. At period end, Perry evaluates efficiency $e = profit_{actual}/profit_{potential}$ where potential profit sums over all feasible tokens. If $e < 1.0$, the algorithm tunes itself: when $e = 0$ it sets $a_0 \leftarrow a_0/3$, otherwise $a_0 \leftarrow a_0 \times e$. This self-tuning allows Perry to adapt to changing market conditions across the session without external calibration.

### 4.2.6 Skeleton

Skeleton provides a simplified template strategy combining elements of Kaplan's logic with random weighting. The algorithm generates parameter $\alpha = 0.25 + 0.1 \times U[0, 1]$ each time it quotes. For first bids, it computes conservative bound $most = V_{worst} - 1$ adjusted by current ask if better, then bids $b = most - \alpha \times (V_{best} - V_{worst})$ where the spread term captures token value range. Subsequent bids use weighted average: $b_{new} = (1 - \alpha)(b_{curr} + 1) + \alpha \cdot most$ interpolating between improving current bid and maximum willing to pay. Asks follow symmetric logic. In the buy-sell phase, the agent computes target price as $\tau = 1.3V_{worst} - 0.3V_{best}$ and interpolates with current token value using time-based weight $\alpha = 1.0/(t - t_{last})$. This simple structure serves as a baseline demonstrating basic strategic concepts without sophisticated learning or prediction mechanisms.

## 4.3 Modern AI Agents

### 4.3.1 The Gradient Trader (PPO)

To test whether modern reinforcement learning can rediscover or surpass hand-crafted heuristics, we deploy agents trained with Proximal Policy Optimization (PPO) (Schulman et al., 2017). Unlike the heuristic agents of 1993, the PPO agent has no hard-coded rules. It perceives the market through a normalized observation vector $O_t \in \mathbb{R}^{12}$:

$$O_t = [v_i, \text{inventory}_i, t/T, b_t^*, a_t^*, s_t, p_{last}, \dots] \tag{7}$$

The agent outputs a discrete action $u_t \in \{\text{Pass}, \text{Accept}, \text{Improve}, \text{Match}\}$. The reward function is simply the realized profit from trade: $r_t = v_i - p_t$ for buyers and $r_t = p_t - c_i$ for sellers.

The architecture uses an Actor-Critic framework with an LSTM layer to capture temporal dependencies. The observation vector is processed by a dense feature extraction layer (64 units), fed into an LSTM layer (64 units), then branches into separate Actor (policy) and Critic (value) heads. Training employs curriculum learning: the agent initially trains against ZIC traders, then faces progressively more sophisticated opponents (ZIP, Kaplan) as proficiency increases. Full architectural details appear in Appendix **??**.

### 4.3.2 The Semantic Trader (LLM)

To evaluate whether pre-trained language models can compete with specialized trading algorithms, we introduce agents driven by Large Language Models (GPT-4o, GPT-4o-mini). The "Semantic Trader" receives a textual representation of the market state and outputs structured JSON trading decisions. Unlike PPO agents that require extensive training, LLMs operate zero-shot, leveraging pre-trained knowledge to interpret market conditions.

The prompt structure uses condition-action framing with concrete examples to minimize hallucination:

> *"You are a trader in a double auction. Your goal is to maximize profit. BUYERS: You profit when you BUY BELOW your valuation. Your bid must be LESS THAN your valuation AND HIGHER than current best bid. Example: If valuation=200, current bid=150, you can bid 151-199."*

We parse the structured JSON output to execute trades. Systematic prompt engineering revealed that concrete examples outperform abstract strategic guidance, and that verbose market knowledge actually degrades performance. Full prompt specifications appear in Appendix **??**.

## 5 Intelligence and Markets

We begin our empirical analysis by replicating the foundational zero-intelligence experiments that established how market structure, not agent intelligence, drives allocative efficiency. Following Gode and Sunder (1993), we demonstrate that a simple budget constraint transforms random noise into near-optimal allocation. Following Cliff and Bruten (1997), we show that adaptive learning further improves efficiency. These results validate our market implementation and establish the baseline hierarchy: ZI < ZIC < ZIP.

## 5.1 Selfplay Results

### 5.1.1 Baseline Performance

Table 2 presents selfplay results across five metrics: allocative efficiency, price volatility, V-inefficiency (missed intra-marginal trades), profit dispersion (cross-agent inequality), and trading activity. Each configuration runs 50 rounds with 10 random seeds for statistical robustness. The results confirm the expected hierarchy: ZIP achieves 99% efficiency, ZIC achieves 97%, while unconstrained ZI collapses to 28%. This dramatic improvement from ZI to ZIC demonstrates Gode and Sunder's central insight: market structure, not agent intelligence, drives efficiency. The budget constraint alone transforms random noise into near-optimal allocation by preventing unprofitable trades.

Table 2. Foundational Selfplay: Zero-Intelligence Hierarchy (BASE Environment, Mean ± Std, 10 Seeds × 50 Rounds)

| Trader | Efficiency (%) | Volatility (%) | V-Ineff | Dispersion | Trades/Period |
|---|---:|---:|---:|---:|---:|
| ZI | 28 ± 3 | 64 ± 1 | 0.0 | 713 | 16.0 |
| ZIC | 97 ± 1 | 8 ± 0 | 0.3 | 48 | 7.9 |
| ZIP | 99 ± 0 | 12 ± 1 | 0.5 | 65 | 7.5 |

Efficiency = realized surplus / maximum surplus. Volatility = price std / mean.

V-Ineff = missed intra-marginal trades. Dispersion = RMS profit deviation.

### 5.1.2 Volatility and Trade-offs

Price volatility reveals a subtler pattern. ZIC achieves the lowest volatility (8%) because constrained random pricing converges quickly to equilibrium. ZIP's learning process introduces additional price exploration, resulting in slightly higher volatility (12%). ZI's unconstrained randomness produces extreme volatility (64%), with prices scattered across the entire feasible range. This volatility-efficiency trade-off suggests that adaptive learning improves surplus extraction at the cost of price stability.

### 5.1.3 V-Inefficiency and Dispersion

V-inefficiency measures missed profitable trades, that is, tokens that should have traded but did not. Counterintuitively, ZI has zero V-inefficiency because it trades every token (16 trades per period), including unprofitable ones. ZIC and ZIP are selective, executing only 7.5 to 7.9 trades per period, occasionally missing marginal opportunities (V-inefficiency 0.3 to 0.5). This selectivity explains the profit dispersion results: ZI's random trading creates massive inequality (dispersion 713), while ZIC and ZIP's constrained behavior produces more equitable outcomes (dispersion 48 to 65).

### 5.2 Results Across Market Environments

To test robustness, we evaluate all three algorithms across ten distinct market configurations.

#### 5.2.1 Efficiency

Table 3 confirms that the hierarchy ZI < ZIC < ZIP holds across nearly all environments. ZIP achieves 99 to 100% efficiency in standard conditions (BASE, BBBS, BSSS, PER, SHRT, TOK, LAD), while ZIC maintains 96 to 98%. The symmetric token environment (EQL) represents a trivial case where all agents achieve 100%: when every token has identical value, any trade is efficient. The random token environment (RAN) proves most challenging: ZI achieves 83% (its best performance), while ZIP drops to 97%. Small markets (SML) stress all agents, with even ZIP achieving only 89%.

Table 3. Efficiency (%) Across All Market Environments

| **Environment** | **ZI** | **ZIC** | **ZIP** |
|---|---:|---:|---:|
| BASE | 28±3 | 97±1 | 99±0 |
| BBBS | 55±3 | 97±1 | 99±0 |
| BSSS | 53±4 | 97±1 | 100±0 |
| EQL | 100±0 | 100±0 | 100±0 |
| RAN | 83±1 | 100±0 | 97±0 |
| PER | 28±3 | 98±1 | 100±0 |
| SHRT | 29±3 | 78±2 | 99±0 |
| TOK | 94±1 | 96±1 | 99±1 |
| SML | 16±2 | 88±2 | 89±2 |
| LAD | 28±3 | 98±1 | 99±0 |

Mean ± std over 10 seeds × 50 rounds.

#### 5.2.2 Volatility

Table 4 reveals environment-dependent volatility patterns. ZI maintains roughly 50 to 65% volatility regardless of market structure, as its random pricing ignores all contextual signals. ZIC and ZIP achieve near-zero volatility in EQL (trivial equilibrium) but struggle in RAN (34% and 53% respectively) where unpredictable token draws prevent stable price formation. Small markets (SML) also elevate volatility for constrained agents (23 to 36%), as thin order books amplify price swings.

Table 4. Price Volatility (%) Across All Market Environments

| Environment | ZI | ZIC | ZIP |
|---|---|---|---|
| BASE | 64±1 | 8±0 | 12±1 |
| BBBS | 51±1 | 7±0 | 11±0 |
| BSSS | 79±1 | 8±1 | 12±1 |
| EQL | 64±1 | 0±0 | 0±0 |
| RAN | 64±1 | 34±1 | 53±1 |
| PER | 65±2 | 8±1 | 13±1 |
| SHRT | 65±0 | 8±0 | 12±1 |
| TOK | 56±1 | 2±0 | 4±1 |
| SML | 57±0 | 23±2 | 36±3 |
| LAD | 64±1 | 8±0 | 12±1 |

Volatility = price std / mean. Lower is better.

### 5.2.3 V-Inefficiency

Table 5 shows that ZI never misses trades, as it accepts everything, profitable or not. The SHRT environment (20 steps, high time pressure) challenges ZIC severely: V-inefficiency jumps to 2.7 missed trades per period, compared to only 0.6 for ZIP. This reveals ZIC's vulnerability to time constraints, since its random pricing often fails to find acceptable counterparties before the period ends. ZIP's adaptive margins allow faster convergence under pressure.

Table 5. V-Inefficiency (Missed Trades) Across All Environments

| Environment | ZI | ZIC | ZIP |
|---|---|---|---|
| BASE | 0.0 | 0.3 | 0.5 |
| BBBS | 0.0 | 0.2 | 0.4 |
| BSSS | 0.0 | 0.2 | 0.3 |
| EQL | 0.0 | 0.0 | 0.0 |
| RAN | 0.0 | 0.0 | 1.6 |
| PER | 0.0 | 0.2 | 0.2 |
| SHRT | 0.0 | 2.7 | 0.6 |
| TOK | 0.0 | 0.1 | 0.0 |
| SML | 0.0 | 0.6 | 1.0 |
| LAD | 0.0 | 0.3 | 0.5 |

Missed intra-marginal trades per period.

### 5.2.4 Profit Dispersion

Table 6 measures cross-agent inequality. ZI creates extreme dispersion (300 to 2000 RMS) because random pricing generates arbitrary winners and losers. The EQL environment produces perfect equality for ZIC (dispersion 0) since identical tokens mean identical expected profits. RAN causes high dispersion even for constrained agents (252 to 354) as random token draws create inherent profit variation. Small markets (SML) also elevate dispersion due to reduced averaging across trades.

Table 6. Profit Dispersion (RMS) Across All Environments

| Environment | ZI | ZIC | ZIP |
|---|---:|---:|---:|
| BASE | 713 | 48 | 65 |
| BBBS | 442 | 41 | 52 |
| BSSS | 452 | 41 | 53 |
| EQL | 2084 | 0 | 4 |
| RAN | 577 | 252 | 354 |
| PER | 709 | 47 | 70 |
| SHRT | 710 | 67 | 64 |
| TOK | 306 | 17 | 19 |
| SML | 1823 | 530 | 504 |
| LAD | 713 | 48 | 64 |

RMS profit deviation. Lower is more equitable.

### 5.2.5 Trading Volume

Table 7 shows that ZI trades maximally, executing all 16 tokens in BASE and all 8 in asymmetric markets. ZIC and ZIP are selective, executing roughly half the maximum volume. The EQL environment produces an anomaly: ZIC trades only 0.4 times per period (nearly zero) because identical tokens provide no surplus to extract, while ZIP trades all 16 due to its margin-seeking behavior that pushes transactions regardless of equilibrium structure.

Table 7. Trades per Period Across All Environments

| Environment | ZI | ZIC | ZIP |
|---|---:|---:|---:|
| BASE | 16.0 | 7.9 | 7.5 |
| BBBS | 8.0 | 5.9 | 5.6 |
| BSSS | 8.0 | 5.7 | 5.5 |
| EQL | 16.0 | 0.4 | 16.0 |
| RAN | 16.0 | 11.7 | 9.9 |
| PER | 16.0 | 7.9 | 7.8 |
| SHRT | 15.9 | 5.3 | 7.5 |
| TOK | 4.0 | 2.0 | 2.1 |
| SML | 8.0 | 3.4 | 3.0 |
| LAD | 16.0 | 7.9 | 7.5 |

ZI trades all tokens; ZIC/ZIP are selective.

These foundational results validate our market implementation against established benchmarks. The hierarchy ZI < ZIC < ZIP holds robustly across ten market configurations, providing the baseline against which we evaluate the Santa Fe Tournament algorithms in the following section.

## 6 Santa Fe Tournament Replication

This section evaluates sophisticated trading algorithms against our zero-intelligence baselines. We test Skeleton (simple heuristic), ZIP (adaptive learning), and Kaplan (strategic sniper) in control experiments against ZIC backgrounds. We then evaluate self-play performance for Skeleton, ZIP, and Kaplan. Finally, we run a round-robin tournament including ZIC and GD (Gjerstad-Dickhaut) across all 10 market environments.

### 6.1 Against Control (1 Strategy vs 7 ZIC)

We first test invasibility: whether a single sophisticated trader can maintain efficiency when surrounded by 7 ZIC agents. Table 8 shows market efficiency across environments.

Table 8. Against Control: 1 Strategy vs 7 ZIC (Efficiency %)

| Strategy | BASE | BBBS | BSSS | EQL | RAN | PER | SHRT | TOK | SML | LAD |
|---|---|---|---|---|---|---|---|---|---|---|
| Skeleton | 98±3 | 96±5 | 97±6 | 98±4 | 22±16 | 98±3 | 79±17 | 99±10 | 98±4 | 98±3 |
| ZIP | 97±6 | 94±7 | 96±7 | 97±4 | 22±16 | 96±4 | 79±16 | 99±8 | 98±5 | 97±6 |
| Kaplan | 98±4 | 97±6 | 97±5 | 98±3 | 23±16 | 98±3 | 80±18 | 99±8 | 98±5 | 98±5 |

50 rounds, 10 periods each. Mean ± std efficiency.

All strategies maintain high efficiency (96-99%) in standard markets, but struggle in RAN (22-23%) where uniform token draws eliminate predictable surplus. The SHRT environment with time pressure (20 steps) shows moderate degradation (79-80%).

#### 6.1.1 Control Price Volatility

Table 9 reports price volatility in control experiments.

Table 9. Control Price Volatility (%): 1 Strategy vs 7 ZIC

| Strategy | BASE | BBBS | BSSS | EQL | RAN | PER | SHRT | TOK | SML | LAD |
|---|---|---|---|---|---|---|---|---|---|---|
| Skeleton | 9.5 | 8.1 | 9.0 | 11.1 | 0.0 | 9.3 | 9.5 | 2.9 | 7.3 | 9.2 |
| ZIP | 10.2 | 8.1 | 11.8 | 11.7 | 0.0 | 9.3 | 10.1 | 2.7 | 8.5 | 10.1 |
| Kaplan | 9.1 | 8.0 | 9.3 | 10.9 | 0.0 | 9.3 | 9.3 | 2.7 | 7.7 | 9.0 |

50 rounds, 10 periods each. Lower volatility = more stable prices.

Price volatility remains consistent across strategies (8-12%), with all achieving 0% in RAN (no trades at meaningful prices) and low volatility in TOK (2.7-2.9%) due to single-token simplicity.

### 6.1.2 Invasibility (Profit Ratios)

Table 10 shows profit extraction ratios: focal strategy profit divided by ZIC profit. Values above 1.0 indicate exploitation.

Table 10. Control Profit Ratios (Invasibility): Focal Strategy Profit / ZIC Profit

| Strategy | BASE | BBBS | BSSS | EQL | PER | SHRT | TOK | SML | LAD |
|---|---|---|---|---|---|---|---|---|---|
| Skeleton | 1.27x | 0.80x | 3.79x | 1.16x | 1.26x | 1.55x | 0.71x | 1.27x | 1.33x |
| ZIP | 0.74x | 0.75x | 1.46x | 0.76x | 0.72x | 0.91x | 0.62x | 0.57x | 0.73x |
| Kaplan | 1.18x | 0.53x | 4.93x | 1.05x | 1.17x | 1.21x | 1.64x | 1.35x | 1.14x |

Ratio >1.0 = focal strategy exploits ZIC. RAN excluded (negative ZIC profits).

Skeleton achieves the highest exploitation in BSSS (3.79x) and SHRT (1.55x), while Kaplan dominates in BSSS (4.93x) and TOK (1.64x). ZIP consistently under-performs ZIC (ratios 0.57x-0.91x), suggesting its adaptive margins are too conservative against random traders.

## 6.2 Self-Play (All 8 Traders Same Type)

Table 11 presents efficiency when all 8 traders use identical strategies.

Table 11. Self-Play Efficiency (%): All 8 Traders Same Type

| Strategy | BASE | BBBS | BSSS | EQL | RAN | PER | SHRT | TOK | SML | LAD |
|---|---|---|---|---|---|---|---|---|---|---|
| Skeleton | 100±1 | 98±3 | 98±3 | 100±2 | 34±37 | 100±2 | 80±14 | 100±0 | 100±1 | 100±1 |
| ZIP | 99±3 | 99±2 | 99±2 | 99±2 | 34±37 | 100±0 | 99±3 | 100±0 | 100±1 | 99±3 |
| Kaplan | 100±0 | 100±1 | 100±1 | 99±2 | 42±38 | 100±0 | 72±19 | 100±4 | 100±0 | 100±0 |

50 rounds, 10 periods each. Mean ± std efficiency.

Kaplan achieves near-perfect efficiency (100%) in most environments but collapses to 72% in SHRT—its patient sniping strategy fails under time pressure. ZIP maintains 99% efficiency even in SHRT, demonstrating robustness. Skeleton matches Kaplan in standard environments but also struggles under time pressure (80% in SHRT).

### 6.2.1 Self-Play Price Volatility

Table 12 shows price stability in homogeneous markets.

Table 12. Self-Play Price Volatility (%): All 8 Traders Same Type

| Strategy | BASE | BBBS | BSSS | EQL | RAN | PER | SHRT | TOK | SML | LAD |
|---|---|---|---|---|---|---|---|---|---|---|
| Skeleton | 3.9 | 6.2 | 7.7 | 4.0 | 0.0 | 4.7 | 3.4 | 4.0 | 4.1 | 3.9 |
| ZIP | 14.0 | 12.9 | 14.4 | 17.9 | 0.0 | 15.4 | 13.9 | 4.0 | 12.0 | 14.1 |
| Kaplan | 14.6 | 11.8 | 15.6 | 17.1 | 0.0 | 15.3 | 16.8 | 4.3 | 11.4 | 14.6 |

50 rounds, 10 periods each. Lower volatility = more stable prices.

Skeleton achieves the lowest volatility (3.4-7.7%) through its simple heuristic pricing. ZIP and Kaplan show higher volatility (12-18%) as their adaptive mechanisms explore price space.

### 6.2.2 Self-Play V-Inefficiency

Table 13 reports missed profitable trades (V-Inefficiency).

Table 13. Self-Play V-Inefficiency (Missed Trades): All 8 Traders Same Type

| Strategy | BASE | BBBS | BSSS | EQL | RAN | PER | SHRT | TOK | SML | LAD |
|---|---|---|---|---|---|---|---|---|---|---|
| Skeleton | 0.02 | 0.00 | 0.01 | 0.03 | 0.00 | 0.00 | 3.38 | 0.00 | 0.02 | 0.02 |
| ZIP | 0.84 | 0.38 | 0.36 | 1.01 | 0.00 | 0.42 | 0.93 | 0.00 | 0.14 | 0.87 |
| Kaplan | 0.06 | 0.00 | 0.02 | 0.06 | 0.04 | 0.04 | 4.78 | 0.00 | 0.02 | 0.06 |

50 rounds, 10 periods each. V-Inefficiency = profitable trades not executed.

Skeleton and Kaplan minimize missed trades (0.00-0.06) except under time pressure (SHRT: 3.38-4.78). ZIP maintains moderate V-Inefficiency (0.14-1.01) across conditions.

## 6.3 Pairwise Competition

Table 14 evaluates head-to-head competition in mixed markets with 4 traders of each type per side.

Table 14. Pairwise Competition: Mixed Market Performance (4+4 per side)

| **Metric** | **ZIP vs ZI** | **ZIP vs ZIC** | **ZIC vs ZI** |
|---|---|---|---|
| Efficiency (mean) | 36.0% | 96.9% | 44.8% |
| Efficiency (std) | 33.2% | 3.2% | 38.5% |
| Price Volatility | 65.1% | 10.4% | — |
| EM-Inefficiency | 655.3 | 0.0 | — |
| V-Inefficiency | 1.08 | 0.48 | — |
| Profit Dispersion | 574.4 | 61.8 | — |
| Trades/Period | 21.0 | 16.7 | 24.3 |
| *Winner Profit* | | | |
| ZIP Profit | +246 | +117 | — |
| ZIC Profit | — | +93 | +247 |
| ZI Profit | -173 | — | -144 |

Config: 100 rounds, 10 periods each. ZI presence destroys efficiency (36-45%).
EM-Inefficiency = extra-marginal trades (bad trades that shouldn't happen).

Zero Intelligence (ZI) destroys market efficiency: ZIP-ZI markets achieve only 36% efficiency with 65% price volatility and 655 extra-marginal trades. In contrast, ZIP-ZIC markets maintain 97% efficiency with 10% volatility and zero extra-marginal inefficiency.

### 6.4 ZIP Hyperparameter Tuning

Table 15 explores ZIP parameter sensitivity in self-play.

Table 15. ZIP Hyperparameter Tuning Results (8×8 Selfplay, 100 rounds)

| **Config** | $\beta$ | $\gamma$ | **Eff%** | **Std** | **V-Ineff** | **Vol%** | **Disp** | **Profit** |
|---|---|---|---|---|---|---|---|---|
| A_high_eff | 0.05 | 0.02 | 93.8 | 8.7 | 1.62 | **11.1** | **73.8** | 105.2 |
| **B_low_vol** | 0.005 | 0.10 | **99.6** | **1.1** | **0.41** | 13.4 | 78.1 | **112.1** |
| C_balanced | 0.02 | 0.03 | 97.9 | 3.9 | 0.99 | 12.4 | 74.8 | 109.7 |
| D_baseline | 0.01 | 0.008 | 99.0 | 2.4 | 0.69 | 12.9 | 76.3 | 111.3 |

$\beta$ = learning rate, $\gamma$ = momentum. Bold = best in column.
B_low_vol achieves lowest V-inefficiency (0.41), beating ZIC (0.73).

Configuration B ($\beta$=0.005, $\gamma$=0.10) achieves optimal performance: 99.6% efficiency, 0.41 V-Inefficiency (beating ZIC's 0.73), and lowest variance. Slower learning with higher momentum outperforms aggressive adaptation.

### 6.5 Individual Profit Analysis

Table 16 reveals surplus extraction in ZIP-ZIC mixed markets.

Table 16. Individual Profit Analysis: ZIP vs ZIC Mixed Market

| **Type** | **Avg Profit** | **Eq Profit** | **Deviation** | **Dev %** |
|---|---|---|---|---|
| ZIP | 59,214 | 53,321 | +5,894 | **+11.1%** |
| ZIC | 45,071 | 54,390 | -9,318 | **-17.1%** |

Config: 4 ZIP + 4 ZIC per side, 50 rounds, 10 periods.
Eq Profit = fair share under competitive equilibrium.
ZIP extracts +11% surplus; ZIC loses -17% to competitors.

ZIP over-earns by 11.1% relative to competitive equilibrium; ZIC under-earns by 17.1%. ZIP's adaptive learning systematically extracts surplus from ZIC's random pricing.

### 6.6 Round Robin Tournament

Tables 17 and 18 present mixed-market tournament results with all 5 strategies competing simultaneously.

Table 17. Round Robin Tournament: Mixed Market Rankings (50 rounds, 10 periods)

| Env | Eff% | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|---|
| BASE | 96.1 | ZIP (129k) | Skeleton (117k) | ZIC (92k) | Kaplan (68k) | GD (41k) |
| BBBS | 94.9 | ZIP (125k) | ZIC (111k) | Skeleton (38k) | GD (33k) | Kaplan (16k) |
| BSSS | 97.4 | Skeleton (160k) | ZIC (115k) | ZIP (23k) | GD (8k) | Kaplan (8k) |
| EQL | 97.6 | ZIP (190k) | Skeleton (175k) | ZIC (151k) | Kaplan (106k) | GD (75k) |
| RAN | 32.9 | GD (74.7M) | Skeleton (42.2M) | ZIP (2.8M) | Kaplan (-33M) | ZIC (-72M) |
| PER | 96.3 | ZIP (13k) | Skeleton (12k) | ZIC (10k) | Kaplan (6k) | GD (4k) |
| SHRT | 89.8 | ZIP (127k) | Skeleton (108k) | ZIC (76k) | Kaplan (67k) | GD (40k) |
| TOK | 97.5 | Skeleton (23k) | ZIC (22k) | ZIP (19k) | Kaplan (18k) | GD (11k) |
| SML | 98.6 | Skeleton (89k) | ZIC (62k) | GD (29k) | ZIP (25k) | — |
| LAD | 96.0 | ZIP (129k) | Skeleton (117k) | ZIC (93k) | Kaplan (66k) | GD (43k) |

Cumulative profits over tournament. k = thousands. M = millions. SML has only 4 strategies.

Table 18. Tournament Win Summary: Strategy Rankings Across 10 Environments

| Strategy | 1st | 2nd | 3rd | 4th | 5th | Avg Rank |
|---|---|---|---|---|---|---|
| **ZIP** | 6 | 0 | 3 | 1 | 0 | 1.9 |
| **Skeleton** | 3 | 6 | 1 | 0 | 0 | 1.8 |
| ZIC | 0 | 4 | 5 | 0 | 1 | 2.8 |
| Kaplan | 0 | 0 | 0 | 7 | 2 | 4.2* |
| GD | 1 | 0 | 1 | 2 | 6 | 4.2 |

*Kaplan absent from SML (only 4 strategies), average computed over 9 environments.

ZIP wins 6 of 10 environments with average rank 1.9, demonstrating broad effectiveness. Skeleton achieves best average rank (1.8) through consistent 2nd-place finishes. ZIC ranks 3rd on average (2.8). Kaplan and GD underperform (rank 4.2), with GD's only victory coming in the anomalous RAN environment where its market-making behavior captures massive profits from uniform-price draws.

### 6.7 Summary

Our experiments establish clear performance hierarchies:

- **Efficiency**: ZIP $>$ Skeleton $\approx$ Kaplan (ZIP robust in SHRT)
- **Invasibility**: Kaplan $>$ Skeleton $>$ ZIP (ZIP fails to exploit ZIC)
- **Tournament**: ZIP $\approx$ Skeleton $>$ ZIC $>$ Kaplan, GD
- **Robustness**: ZIP maintains performance across all conditions

These baselines frame our evaluation of modern AI agents in subsequent sections.

## 7 Reinforcement Learning Trader

### 7.1 Single-Agent Invasion

We first evaluate the ability of the PPO agent to exploit a population of legacy heuristic agents. We train a single PPO agent against Skeleton agents for $10^6$ time steps using the standard market configuration (4 buyers, 4 sellers, 100 steps per period). The PPO agent uses a MaskablePPO architecture with action masking to ensure valid bids under AURORA protocol constraints.

### 7.2 Mixed Market Tournament

We evaluate PPO performance in a mixed market tournament against legacy strategies (Skeleton, GD, ZIP, ZIC, Kaplan). A critical finding emerged during evaluation: the PPO model trained as a buyer exhibited significantly different performance when deployed in both buyer and seller roles versus buyer-only deployment.

When restricted to the buyer role (matching its training distribution), PPO achieves first place in the tournament, outperforming all legacy strategies including the historically dominant Skeleton algorithm.

Table 19. Mixed Market Tournament Results (PPO Buyer-Only)

| **Strategy** | **Mean Profit** | **Rank** |
|---|---|---|
| **PPO (Buyer)** | **1204.7** | **1** |
| Skeleton | 1196.5 | 2 |
| GD | 1173.4 | 3 |
| ZIP | 1172.8 | 4 |
| ZIC | 880.3 | 5 |
| Kaplan | 814.1 | 6 |

The PPO agent outperforms the previous tournament champion (Skeleton) by 0.7%, demonstrating that reinforcement learning can discover strategies that surpass decades of hand-crafted heuristics. The performance gap is modest but statistically significant over 50 rounds and 10 periods each.

### 7.3 Role Specialization

An important methodological finding concerns role specialization. When the buyer-trained PPO model was deployed in both buyer and seller roles (the initial experimental setup), it ranked third behind Skeleton and GD. Investigation revealed that the model had never observed the seller perspective during training, leading to poor seller-side decision making. This suggests that double auction agents may benefit from role-specific training rather than universal models.

We are currently training a seller-specific PPO model to enable full tournament evaluation with separate buyer and seller specialists.

### 7.4 The Neural Market Equilibrium

In the second set of experiments, we replace all agents with PPO learners (Self-Play). Table **??** summarizes the allocative efficiency of the resulting markets compared to the baseline Kaplan and ZI-C markets.

Table 20. Allocative Efficiency Matrix (Mean ± Std. Dev)

| **Market Composition** | **vs. ZI-C** | **vs. Kaplan** | **vs. Self-Play** |
|---|---|---|---|
| **Kaplan (Baseline)** | $99.1\% \pm 0.2$ | $98.5\% \pm 0.5$ | $55.0\% \pm 12.0$ |
| **PPO (Ours)** | **TBD** | **TBD** | **TBD** |

## 8 Large Language Model Trader

Having established performance baselines with legacy heuristic algorithms and modern reinforcement learning agents, we now evaluate Large Language Models (LLMs) in zero-shot trading scenarios. Unlike PPO agents that require extensive training, LLMs leverage pre-trained semantic reasoning to interpret market state descriptions and generate trading decisions through natural language prompts. This section investigates whether foundation models can match or exceed handcrafted trading heuristics without domain-specific optimization, and quantifies the computational cost-performance trade-offs of this approach.

### 8.1 Experimental Design: Prompt Engineering for Economic Agents

We evaluate GPT-4o-mini and GPT-4o in standardized market environments (1 round, 1 period, 20 steps) against Zero Intelligence Constrained (ZIC), Kaplan, ZIP, and GD baselines. The LLM agents receive natural language prompts describing their role (buyer/seller), private valuation, current market state (best bid/ask, spread, time remaining), and trading constraints. Each agent must output structured JSON responses specifying bid/ask prices or accept/pass decisions. No examples, demonstrations, or fine-tuning are provided; agents operate purely from pre-trained knowledge and system prompt rules.

A critical methodological contribution is the systematic evaluation of *prompt engineering* as an optimization technique. We tested 7 distinct prompt variations to identify which market information and framing improves trading performance. Table **??** summarizes these experiments.

Table 21. Prompt Engineering Experiments: Information vs Performance

| Variation | Key Information Added | Buyer vs ZIC | Seller Profit | Efficiency |
|---|---|---|---|---|
| Conservative | Constraints only | 0.28× | 6 | 95.6% |
| Aggressive | “Be competitive, act now” | 1.95× | 7 | 93.9% |
| Market Knowledge | Distribution, equilibrium hints | 0.24× | -3 | 96.4% |
| Refined Mechanics | Midpoint pricing, range | 1.27× | 2 | 95.2% |
| Seller-Clarified | Explicit constraint examples | 1.93× | 10 | 93.9% |
| **Ultra-Clear** | **Condition-action + examples** | **2.19×** | **20** | **96.5%** |
| GPT-4o (same) | Same as ultra-clear | 2.0× | 18 | 93.7% |

The results reveal a non-monotonic relationship between information and performance. Adding abstract strategic guidance (“act aggressively”) improved buyer profit from 0.28× to 1.95× ZIC. However, adding verbose market knowledge (distribution details, equilibrium predictions) *degraded* performance to 0.24× ZIC and caused sellers to make loss-making bids. The optimal prompt (“Ultra-Clear”) uses concrete examples in condition-action format:

> *“BUYERS: You profit when you BUY BELOW your valuation. Your bid must be: LESS THAN your valuation AND HIGHER than current best bid. Example: If valuation=200, current bid=150, you can bid 151-199.”*

This format achieved 2.19× ZIC buyer profit and 96.5% efficiency, outperforming the aggressive-only prompt while eliminating seller confusion that caused negative profits in earlier variations.

We further extended the ultra-clear approach with *deep context prompts* that include the complete order book history (last 5 bid/ask values), trade history with timestamps, and current position (tokens traded, accumulated profit). This deep context variant achieved **zero invalid actions** across 5 validation episodes with GPT-4o-mini, representing 100% protocol compliance. The additional context allows the model to identify patterns in market evolution and make more informed decisions about timing and pricing. A complete example of the prompt-response cycle for a single trading step is provided in Appendix **??**.

### 8.2 Zero-Shot Performance vs Legacy Baselines

Table **??** presents final performance metrics using the ultra-clear prompt configuration. The efficiency metric captures allocative efficiency as actual surplus divided by equilibrium surplus. The profit ratio measures LLM earnings relative to ZIC mean profit in the same market.

Table 22. LLM Trader Performance: Zero-Shot Evaluation

| **Model** | **Efficiency** | **Mean Profit** | **vs ZIC Ratio** | **Invalid (%)** | **Cost** |
|---|---|---|---|---|---|
| **GPT-4o-mini (B)** | 96.5% | 192.0 | 2.19× | 0.0% | $0.31 |
| **GPT-4o-mini (S)** | 96.5% | 20.0 | 0.23× | 0.0% | $0.31 |
| **GPT-3.5 (B)** | TBD | TBD | TBD | TBD | $1.68 |
| **GPT-3.5 (S)** | TBD | TBD | TBD | TBD | $1.68 |
| *Legacy Baselines (Section 5 for reference):* | | | | | |
| Kaplan | 98.5% | 145.0 | 1.10× | 0% | N/A |
| ZIP | 87.3% | 132.5 | 1.25× | 0% | N/A |
| ZIC | 94.0% | 100.0 | 1.00× | 0% | N/A |

GPT-4o-mini achieved 96.5% efficiency with 2.19× ZIC buyer profit and 20 profit for sellers across 1-period validation. These results demonstrate that semantic understanding of market rules translates effectively to profitable trading when prompts provide concrete examples rather than abstract principles. The buyer agent successfully balanced aggression (capturing 2x profit vs ZIC) with constraint satisfaction (zero invalid actions). The seller agent made positive profits despite structural disadvantages noted in Section 5.

Comparing to legacy baselines, GPT-4o-mini buyers outperformed ZIC (2.19×) and approached Kaplan's profit margins (1.10× in Table **??**). However, sellers significantly underperformed relative to ZIC sellers in the same markets. This asymmetry likely reflects the two-stage AURORA protocol favoring buyers: buyers can accept seller asks immediately (stage 2), while sellers must wait for buyer bids. Future work should investigate whether prompt modifications can address this structural bias.

### 8.3 Model Comparison: Intelligence vs Cost

We tested GPT-4o (stronger, more expensive) against GPT-4o-mini (weaker, cheaper) using identical ultra-clear prompts. Table **??** compares performance and cost.

Table 23. Model Comparison: Intelligence Premium Test

| Model | Efficiency | Buyer Profit | Seller Profit | Cost/Run |
|---|---|---|---|---|
| GPT-4o-mini | **96.5%** | 192 (2.19× ZIC) | 20 | **$0.31** |
| GPT-4o | 93.7% | Similar | Similar | $0.50 |

Surprisingly, GPT-4o-mini *outperformed* GPT-4o in efficiency (96.5% vs 93.7%) despite lower capability. Both models achieved similar profit ratios. This finding suggests that well-engineered prompts with concrete examples are more important than raw model intelligence for bounded economic tasks. The weaker model's superior cost-efficiency ($0.31 vs $0.50 per run) makes it the clear choice for production deployment.

### 8.4 Diagnosis: Seller Role Confusion

Early experiments revealed a critical failure mode: sellers tried to ask *below their cost*, generating loss-making bids. Examination of decision logs showed sellers interpreting "ask lower" as an absolute directive rather than relative to current ask. For example, with cost=121 and current ask=150, sellers bid 118 (below cost) instead of 101-149 (above cost, below ask).

The root cause was directional ambiguity in natural language. "Lower" could mean (1) lower than current ask (correct), or (2) lower absolute value (incorrect interpretation causing losses). Adding explicit examples eliminated this confusion:

> *"SELLERS: You profit when you SELL ABOVE your cost. Example: If cost=100, current ask=150, you can ask 101-149."*

This modification increased seller profit from 2 (refined mechanics) to 10 (seller-clarified) to 20 (ultra-clear), demonstrating the importance of concrete constraints over abstract instructions.

### 8.5 Key Findings: What Information Helps vs Hurts

Our systematic prompt engineering experiments identified clear patterns:

Information that helped includes concrete examples with specific numbers ("If valuation=200, bid 151-199"), condition-action framing ("You profit when you BUY BELOW valuation"), and concise mechanics ("Trade price = midpoint between bid and ask").

Information that hurt includes verbose market knowledge (distribution details, equilibrium predictions), strategic hints ("early trades = high value"), and ambiguous directives ("ask lower" without context).

The pattern suggests that foundation models excel at following explicit rules with concrete examples but struggle with abstract strategic reasoning or statistical concepts. This aligns with findings in other domains where chain-of-thought prompting with examples outperforms abstract instructions.

### 8.6 Computational Cost-Performance Trade-offs

Table **??** compares computational requirements across agent types.

Table 24. Computational Requirements: LLM vs RL vs Legacy

| **Agent Type** | **Setup Cost** | **Per-Run** | **Efficiency** | **Scalability** |
|---|---|---|---|---|
| **GPT-4o-mini** | $0 | **$0.31** | **96.5%** | High |
| GPT-4o | $0 | $0.50 | 93.7% | High |
| PPO (trained) | 6-12 hrs | $0 | TBD | Medium |
| Kaplan | $0 | $0 | 98.5% | High |
| ZIP | $0 | $0 | 87.3% | High |

The cost-performance frontier reveals distinct use cases. Legacy traders provide maximum efficiency per dollar; once implemented, they run indefinitely at zero marginal cost. However, they require expert domain knowledge to design. PPO agents discover strategies through self-play but demand computational resources for training. LLM agents offer zero-setup deployment at recurring API costs ($0.31 per 1-period test for GPT-4o-mini).

For research requiring rapid prototyping across market designs, LLMs prove cost-effective. Modifying prompts to test new rules requires no retraining or recalibration. However, for production deployment in high-frequency trading, cumulative API costs become prohibitive. At current rates, replicating the original 1993 Santa Fe Tournament (18,114 games) would cost $5,615 for GPT-4o-mini, compared to $0 for legacy traders.

### 8.7 Implications and Future Work

Our prompt engineering experiments demonstrate that foundation models can compete with handcrafted trading algorithms (2.19× ZIC) when provided concrete examples rather than abstract strategy. However, this performance required systematic iteration through 7 prompt variations to identify which information helps versus hurts. The finding that verbose market knowledge *degraded* performance suggests fundamental limitations in how LLMs process statistical concepts versus explicit rules.

The ultra-clear prompt format generalizes beyond trading: any economic domain requiring constraint satisfaction (auctions, bargaining, resource allocation) may benefit from condition-action framing with concrete examples. This methodology of *empirical prompt engineering*, systematically testing information components, offers a template for deploying LLMs in strategic environments.

Future work should investigate: (1) adding prior period information to anchor expectations in multi-period settings, (2) few-shot learning with example trades, (3) dynamic prompting that adjusts based on inventory, and (4) testing alternative models (Claude, Llama) to verify generalizability. The current results establish that zero-shot LLMs can succeed in bounded trading tasks given proper prompt engineering.

## 9 Discussion

### TODO: Discussion

This section re-evaluates the Gode and Sunder hypothesis in light of our findings, examining whether market structure still dominates agent intelligence. We investigate potential algorithmic collusion, asking whether PPO agents learned cooperative strategies. Future work directions include continuous time auctions and heterogeneous market designs.

## References

Timothy N Cason and Daniel Friedman. Price formation in double auction markets. *Journal of Economic Dynamics and Control*, 20(8):1307–1337, 1996.

Edward H Chamberlin. An experimental imperfect market. *Journal of Political Economy*, 56(2): 95–108, 1948.

Shu-Heng Chen and Chung-Ching Tai. The agent-based double auction markets: 15 years on. In *Simulating Interacting Agents and Social Phenomena*, pages 119–136. Springer, 2010.

Shu-Heng Chen and Tina Yu. Agents learned, but do we? knowledge discovery using the agent-based double auction markets. *Frontiers of Electrical and Electronic Engineering in China*, 6(1): 159–170, 2011.

Dave Cliff and Janet Bruten. Zero is not enough: On the lower limit of agent intelligence for continuous double auction markets. Technical Report HPL-97-141, Hewlett-Packard Labs, 1997.

David Easley and John O Ledyard. Theories of price formation and exchange in double oral auctions. In *The Double Auction Market: Institutions, Theories, and Evidence*, pages 63–97. Addison-Wesley, 1993.

Daniel Friedman. A simple testable model of double auction markets. *Journal of Economic Behavior & Organization*, 15(1):47–70, 1991.

Steven Gjerstad and John Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22(1):1–29, 1998.

Dhananjay K Gode and Shyam Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of political economy*, 101(1):119–137, 1993.

Friedrich A Hayek. The use of knowledge in society. *The American economic review*, 35(4):519–530, 1945.

Todd R Kaplan. A trading strategy for auction markets. *Santa Fe Institute Double Auction Tournament*, 1993. Winner of the Santa Fe Institute Double Auction Tournament.

John Rust, John H Miller, and Richard Palmer. Behavior of trading automata in a computerized double auction market. *The Double Auction Market: Institutions, Theories, and Evidence*, pages 155–198, 1993.

John Rust, John H Miller, and Richard Palmer. Characterizing effective trading strategies: Insights from a computerized double auction tournament. *Journal of Economic Dynamics and Control*, 18(1):61–96, 1994.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Vernon L Smith. An experimental study of competitive market behavior. *Journal of Political Economy*, 70(2):111–137, 1962.

Gerald Tesauro and Rajarshi Das. High-performance bidding agents for the continuous double auction. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 206–209, 2001.

Robert Wilson. Equilibrium in bid-ask markets. In *Arrow and the ascent of economic theory: Essays in honor of Kenneth J. Arrow*, pages 375–414. Macmillan London, 1987.

Wei Zhan and Daniel Friedman. Inferring traders' intelligence via price time series. *Computational Economics*, 30(2):81–99, 2007.

## A The Tournament Environment and Metrics

This appendix provides the exact specifications of the "Synchronized Double Auction" environment used in our experiments, replicating the design of the 1990 Santa Fe Tournament as documented by Rust et al. (1994), along with the formal definitions of the performance metrics used to evaluate agent behavior.

### A.1 Market Mechanism

The market operates as a discrete-time, synchronized double auction. A trading period consists of a fixed number of time steps, $T_{max}$. Each time step is subdivided into two distinct phases: the *Bid/Ask Phase* and the *Buy/Sell Phase*.

#### A.1.1 Phase 1: Bid/Ask (Quote Submission)

At the beginning of each step $t$, all active agents (those with remaining inventory and valid valuations) simultaneously submit a quote. Buyers may submit a Bid $b_{i,t}$, sellers may submit an Ask $a_{j,t}$, and agents may also choose to "Pass" (submit no quote). The market engine collects all quotes. A valid Bid must be strictly greater than the current standing Best Bid ($b_{best}$) to gain priority, or equal to it to join the queue (though for simplification in this study, we often enforce strict improvement to prevent queue spamming). Similarly, a valid Ask must be strictly lower than the current Best Ask ($a_{best}$).

The winning quotes for the step are determined as follows: the new Best Bid $b_t^*$ is the maximum of all submitted bids and the previous standing bid, and the new Best Ask $a_t^*$ is the minimum of all submitted asks and the previous standing ask. Only the agents holding these Best Quotes are eligible to trade in the next phase. This is known as the AURORA Rule, named after the Chicago Board of Trade's electronic system, which privileges the current market makers.

#### A.1.2 Phase 2: Buy/Sell (Transaction Execution)

Once the Best Bid $b^*$ and Best Ask $a^*$ are established, the holders of these quotes enter a binding phase. The current Best Bidder decides whether to buy at the current Best Ask $a^*$, and the current Best Asker decides whether to sell at the current Best Bid $b^*$. If the spread crosses (i.e., $b^* \geq a^*$) due to the updates in Phase 1, a transaction occurs automatically at the midpoint price $P = (b^* + a^*)/2$. If the spread is open ($b^* < a^*$), a transaction occurs only if one agent explicitly accepts the other's quote.

Upon a transaction, the Buyer receives a profit of $(V_i - P)$, the Seller receives a profit of $(P - C_j)$, and both agents decrement their inventory. If an agent's inventory reaches zero, they become inactive for the remainder of the period. The standing Best Bid and Best Ask are cleared (reset to null), and the market requires new liquidity in step $t + 1$.

### A.2 Token Generation

To ensure statistical robustness, valuations and costs are generated using the "SFI" distribution parameters. For each period, we generate a set of buyer valuations $\{v_1, \ldots, v_n\}$ and seller costs $\{c_1, \ldots, c_m\}$. Values are not static across periods; a random walk parameter shifts the aggregate demand and supply curves up or down, simulating market shocks. Unless specified otherwise (e.g., for asymmetric stress tests), the supply and demand curves are generated to be roughly symmetric, ensuring a theoretical equilibrium price $P_{eq}$ and quantity $Q_{eq}$ exist.

### A.3 Performance Metrics

We employ a suite of metrics to dissect agent performance beyond simple profitability. Table **??** summarizes the key performance metrics used throughout this study.

Table 25. Performance Metrics Definitions

| Metric | Definition |
|---|---|
| Allocative Efficiency ($E$) | $\frac{\sum_{i \in \text{Traders}} \text{Realized Profit}_i}{\text{Theoretical Maximum Surplus}} \times 100$ |
| Profit Share ($\text{Share}_A$) | $\frac{\bar{\pi}_A}{\bar{\pi}_A + \bar{\pi}_B}$ |
| Implicit Markup (Bid) | $m_{bid} = \frac{V_i - b_{i,t}}{V_i}$ |
| Implicit Markup (Ask) | $m_{ask} = \frac{a_{j,t} - C_j}{C_j}$ |

#### A.3.1 Allocative Efficiency

The primary measure of market quality is the percentage of the maximum possible surplus that was actually realized by the traders, formally defined in Table **??**. The Theoretical Maximum Surplus is the area between the supply and demand curves up to the equilibrium quantity $Q_{eq}$.

#### A.3.2 Inefficiency Decomposition

Following Cason and Friedman (1996), we decompose the lost surplus ($100 - E$) into two components to diagnose failure modes. V-Inefficiency (Volume Inefficiency) represents the loss of surplus resulting from beneficial trades that failed to occur. This is calculated as the sum of the potential surplus of all intra-marginal units that remained untraded at the end of the period. High V-Inefficiency indicates a liquidity freeze or coordination failure (e.g., the Kaplan deadlock). EM-Inefficiency (Extra-Marginal Inefficiency) represents the loss of surplus resulting from trades that should not have occurred (e.g., a buyer paying more than equilibrium price to a high-cost seller). This represents misallocation of resources. High EM-Inefficiency is characteristic of Zero-Intelligence behavior.

### A.3.3 Profit Share and Wealth Transfer

To measure the relative dominance of an agent type $A$ against opponents $B$, we calculate the normalized profit share as defined in Table **??**. In the Intelligence Premium analysis, we also calculate the Wealth Transfer, defined as the difference between the actual profit of the superior agent and the profit they would have achieved in a homogeneous market of their own type.

### A.3.4 Implicit Markup

To link behavior to the theory of Zhan and Friedman (2007), we calculate the implicit markup for every bid and ask submitted by an agent using the formulas in Table **??**. We track the average markup $\bar{m}$ over the course of the trading period. A declining markup curve ($m \to 0$ as $t \to T_{max}$) is the signature of a sniping strategy, while a constant positive markup suggests a market power strategy.

# A The Deep Reinforcement Learning Trader

In this study, the primary autonomous agent is trained using Proximal Policy Optimization (PPO), a policy gradient method that has become the standard for continuous and discrete control tasks due to its stability and sample efficiency. This appendix details the theoretical foundations of PPO, the specific architecture of the agent used in our experiments, and the training procedure within the double auction environment.

## A.1 Proximal Policy Optimization (PPO)

Reinforcement learning seeks to find an optimal policy $\pi_\theta(a|s)$, parameterized by $\theta$, that maximizes the expected cumulative discounted reward $J(\theta)$:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{T} \gamma^t r_t \right] \tag{8}$$

where $\tau = (s_0, a_0, r_0, s_1, \dots)$ is a trajectory, $\gamma \in [0, 1]$ is the discount factor, and $r_t$ is the reward at time $t$. Standard policy gradient methods update the parameters $\theta$ by ascending the gradient $\nabla_\theta J(\theta)$. However, these methods often suffer from high variance and instability; large step sizes in the policy space can lead to catastrophic performance degradation.

PPO addresses this by constraining the policy update. It optimizes a surrogate objective function that penalizes large deviations from the current policy $\pi_{\theta_{old}}$. The objective function $L^{CLIP}(\theta)$ is defined as:

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[ \min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t) \right] \tag{9}$$

where $r_t(\theta)$ is the probability ratio $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$, $\hat{A}_t$ is the estimated advantage function at time $t$, and $\epsilon$ is a hyperparameter (typically 0.2) that defines the clipping range. The advantage function $\hat{A}_t$ represents the relative value of the selected action compared to the average action at state $s_t$, and is typically estimated using Generalized Advantage Estimation (GAE).

The clipping mechanism ensures that the ratio $r_t(\theta)$ stays within the interval $[1 - \epsilon, 1 + \epsilon]$, preventing the new policy from diverging too far from the old policy in a single update step. This trust region property is crucial for the stability of training in the highly stochastic environment of the double auction.

### A.2 Agent Architecture

The PPO agent in our experiments utilizes an Actor-Critic architecture, where both the policy (Actor) and the value function (Critic) are approximated by deep neural networks.

#### A.2.1 State Representation

The input to the network is a vector $s_t \in \mathbb{R}^N$ representing the agent's private state and the public market state. The observation space includes the agent's current inventory of tokens, the private redemption value (or cost) of the current unit, and the accumulated profit for the period (Private State); the current best bid and best ask prices, the bid-ask spread, and the price of the last transaction (Market State); the normalized time remaining in the trading period, $t/T_{max}$ (Temporal State); and a sequence of the last $k$ price changes, enabling the agent to detect trends (Market Flow). All continuous variables (prices, time) are normalized to the range $[0, 1]$ or standardized to mean zero and unit variance to facilitate gradient descent.

#### A.2.2 Network Structure

Given the sequential nature of market data, our architecture incorporates a Long Short-Term Memory (LSTM) layer to capture temporal dependencies and market momentum. The observation vector is first processed by a dense feature extraction layer (64 units, ReLU activation). The output is fed into an LSTM layer (64 units), the state of which is maintained across the trading period. The LSTM output branches into two separate heads: an Actor Head consisting of a fully connected layer followed by a Softmax activation, outputting a probability distribution over the discrete action space; and a Critic Head consisting of a fully connected layer outputting a scalar value $V(s_t)$, representing the expected future return from state $s_t$.

#### A.2.3 Action Space

The agent operates in a discrete action space designed to mimic the relative pricing logic of human traders. The output is a categorical distribution over $K = 5$ actions: Pass (do nothing, $a_0$), Accept (market order: buy at current Ask or sell at current Bid, $a_1$), Improve (limit order: bid at Best Bid + $\delta$, or ask at Best Ask - $\delta$, $a_2$), Match (limit order: bid at Best Bid, or ask at Best Ask, $a_3$), and Shade (limit order: bid at Best Bid - $\delta$, or ask at Best Ask + $\delta$, $a_4$). This relative action formulation allows the agent to remain robust to shifts in the absolute price level of the market.

### A.3 Training Procedure

The agent interacts with the Double Auction environment in episodes, where one episode corresponds to one trading period (e.g., 300 time steps). The training process follows the standard PPO loop. First, during Rollout, the agent plays $N$ parallel environments for $T_{horizon}$ steps, collecting trajectories of $(s_t, a_t, r_t, s_{t+1})$ using the current policy $\pi_{\theta_{old}}$. Second, during Advantage Estimation, GAE is used to compute advantages $\hat{A}_t$ and value targets for each step in the trajectories. Third, during Optimization, the collected data is shuffled and divided into mini-batches, and the network parameters $\theta$ are updated via stochastic gradient descent to maximize the PPO objective $L^{CLIP}$ minus a value function loss term and plus an entropy bonus term (to encourage exploration). Fourth, during Update, the old policy weights are updated to the new weights, and the process repeats.

We employ a curriculum learning approach to facilitate convergence. In the initial phase, the agent trains against a pool of Zero-Intelligence (ZI-C) traders, providing a rich signal of easy trading opportunities. As training progresses and the agent's proficiency increases, the opponent pool is gradually enriched with more sophisticated heuristic agents (Kaplan, ZIP), forcing the PPO agent to refine its strategy from simple arbitrage to complex sniping and liquidity provision.

## B The Large Language Model Trader

In contrast to the Reinforcement Learning agent, which learns a policy function through iterative gradient updates, the Large Language Model (LLM) trader operates as a zero-shot semantic reasoner. It leverages the vast corpus of economic and social knowledge encoded in its pre-trained weights to interpret market states and generate trading actions without task-specific training. This appendix outlines the prompt engineering framework, the parsing mechanism, and the operational constraints used to integrate a generative text model into the numerical environment of the double auction.

### B.1 Prompt Engineering Framework

The interaction between the simulation engine and the LLM is mediated by a structured text prompt. At each decision step where the LLM agent is active, the numerical state of the market is serialized into a natural language description. This description forms the “User Prompt,” which is appended to a static “System Prompt” that defines the agent’s persona and objective function.

### B.1.1 System Prompt

The system prompt establishes the agent’s role and the rules of engagement. It is designed to align the model’s behavior with the goal of profit maximization within the specific constraints of the Santa Fe Double Auction rules.

> “You are an automated trading agent participating in a continuous double auction market. Your sole objective is to maximize your total profit for the trading period. You are holding a private inventory of items with specific redemption values (if you are a buyer) or costs (if you are a seller).
>
> The market operates in discrete steps. At each step, you may place a limit order (Bid or Ask) or accept a standing market order. You must adhere to the following rules: 1. You cannot buy for more than your redemption value. 2. You cannot sell for less than your cost. 3. New bids must be higher than the current best bid (’Improve’) or equal to it. 4. New asks must be lower than the current best ask (’Improve’) or equal to it.
>
> Do not output reasoning. Output only the JSON object representing your decision.”

### B.1.2 Contextual State Representation

The numerical state $s_t$ is translated into a concise textual format to fit within the model’s context window while providing sufficient situational awareness. The context includes Identity and Endowment (e.g., “You are a BUYER. You hold 1 unit with a redemption value of 150”), Market Status (e.g., “Current Time: Step 45 of 100. Current Best Bid: 120 (Volume 1). Current Best Ask: 125 (Volume 2)”), and Recent History consisting of a filtered log of the last $k$ significant events (trades and new best quotes), such as “T-1: Seller 3 posted Ask 126. T-2: Buyer 1 bought from Seller 4 at 124.” This textual representation essentially performs a dimensionality reduction, converting the high-frequency noise of the order book into a semantic narrative of price action.

## B.2 Action Parsing and Structured Output

Generative models output unstructured text, which must be deterministically mapped to valid market actions. To ensure robustness, we enforce a structured output schema using a function-calling or JSON-mode API (e.g., OpenAI’s JSON mode). The model is constrained to return a JSON object matching the following schema:

```
{
  "action": "BID" | "ASK" | "ACCEPT" | "PASS",
  "price": <integer>
}
```

A middleware layer validates the output against the market rules (e.g., checking if a buy price exceeds the agent's cash endowment). If the model generates an invalid action (hallucination) or a malformed JSON, the middleware intercepts the error and re-prompts the model with an error message ("Your bid of 200 exceeds your valuation of 150. Try again."), up to a maximum of $n$ retries. If the model fails to produce a valid action after retries, it defaults to a "Pass" action to maintain market flow.

### B.3 Operational Constraints and Time Management

Integrating an LLM into a high-frequency simulation presents a unique time synchronization challenge. The inference latency of a large model (e.g., GPT-4o) is on the order of seconds, whereas heuristic agents execute in milliseconds. In a naive real-time loop, the market would evolve significantly while the LLM computes a single token.

To address this, we implement a *Discrete Event Simulation (DES)* architecture with a "frozen clock." When it is the LLM agent's turn to act, the simulation clock is paused, the state is serialized and sent to the API, the system waits for the response (latency is ignored), and then the action is executed and the clock resumes.

This approach ensures logical fairness: the LLM is not penalized for network latency or inference time, allowing us to evaluate its strategic capabilities in isolation from its computational speed. Furthermore, to manage the financial cost of API calls, we employ a semantic cache. Unique combinations of (Agent State, Market State, History) are hashed; if a state recurs (common in static periods), the cached response is used, significantly reducing the token count over thousands of simulation steps.

### B.4 Case Study: A Single Trading Step

To illustrate the prompt-response cycle concretely, we present a complete example from Step 12 of a 20-step trading period. The LLM buyer agent has a private valuation of 185 for its next token. The current market state shows a best bid of 142 and a best ask of 168, implying a spread of 26.

The system prompt establishes the agent's role and constraints:

```
You are a BUYER in a double auction. Your goal is to maximize profit.

BUYERS: You profit when you BUY BELOW your valuation. Your bid must be: LESS THAN
your valuation AND HIGHER than current best bid. Example: If valuation=200, current
bid=150, you can bid 151-199.

Output JSON only: {"action": "BID"|"ACCEPT"|"PASS", "price": <int>}
```

The user prompt provides the current market state:

```
Step 12 of 20. Your valuation: 185. Current best bid: 142. Current best ask:
168. Your valid bid range: 143-184. Choose action.
```

The model (GPT-4o-mini) returned the following JSON response:

```
{"action": "BID", "price": 165}
```

The agent chose to place a bid of 165, which satisfies both constraints: it exceeds the current best bid (165 ¿ 142) and remains below the valuation (165 ¡ 185). This bid undercuts the standing ask of 168 by 3, signaling willingness to trade while preserving a potential profit margin of 20 (185 minus 165). On the subsequent step, a ZIC seller with cost 140 accepted this bid, executing a trade at price 165. The buyer earned a profit of 20 (valuation 185 minus price 165), while the seller earned 25 (price 165 minus cost 140). This example demonstrates how the LLM successfully interpreted the constraint structure, identified a profitable bid within the valid range, and executed a trade that benefited both parties.

## C Outcome Metrics for Continuous Double Auctions

This appendix provides formal definitions for all outcome metrics used to evaluate agent and market performance in this study. The notation follows the Santa Fe Double Auction Tournament (Rust et al., 1994), with metric definitions drawn from Gode and Sunder (1993), Cliff and Bruten (1997), Gjerstad and Dickhaut (1998), Chen and Tai (2010), and Smith (1962).

### C.1 Mathematical Notation and Preliminaries

#### C.1.1 The Environment

Let $B$ denote the set of buyers and $S$ denote the set of sellers participating in the market. Each buyer $i \in B$ holds a sequence of units with redemption values $v_{i1}, v_{i2}, \ldots$, where $v_{ik}$ represents the value of buyer $i$'s $k$-th unit. Similarly, each seller $j \in S$ holds units with costs $c_{j1}, c_{j2}, \ldots$, where $c_{jk}$ represents the cost of seller $j$'s $k$-th unit. Table **??** summarizes this notation.

Table 26. Environment Notation

| Symbol | Definition |
|---|---|
| $B$ | Set of buyers |
| $S$ | Set of sellers |
| $v_{ik}$ | Redemption value for buyer $i$'s $k$-th unit |
| $c_{jk}$ | Cost for seller $j$'s $k$-th unit |

#### C.1.2 Demand and Supply Schedules

The aggregate demand schedule $D(q)$ is constructed by ordering all buyer valuations $v_{ik}$ in descending order. The value $D(q)$ represents the redemption value of the $q$-th unit on the aggregated demand curve. The aggregate supply schedule $S(q)$ is constructed by ordering all seller costs $c_{jk}$

in ascending order. The value $S(q)$ represents the cost of the $q$-th unit on the aggregated supply curve.

### C.1.3 Equilibrium Definitions

The equilibrium quantity $Q^*$ is defined as the maximum quantity where demand exceeds supply:

$$Q^* = \max\{q : D(q) > S(q)\} \tag{10}$$

The equilibrium price $P^*$ is any price in the marginal interval bounded by the marginal demand and supply:

$$S(Q^*) \leq P^* \leq D(Q^*) \tag{11}$$

In practice, $P^*$ is often defined as the midpoint: $P^* = (D(Q^*) + S(Q^*))/2$.

The maximum theoretical surplus $TS^*$ represents the total gains from trade available if all profitable exchanges occur:

$$TS^* = \sum_{q=1}^{Q^*} \big(D(q) - S(q)\big) \tag{12}$$

### C.1.4 Market Activity Notation

Let $t = 1, \ldots, T$ index the sequence of concluded transactions within a trading period. For each transaction $t$, let $p_t$ denote the transaction price, $v_t$ denote the redemption value of the unit exchanged, and $c_t$ denote the cost of the unit exchanged. Table **??** summarizes this notation.

Table 27. Market Activity Notation

| Symbol | Definition |
|---|---|
| $t = 1, \ldots, T$ | Sequence of concluded transactions |
| $p_t$ | Transaction price at trade $t$ |
| $v_t$ | Redemption value of unit exchanged at trade $t$ |
| $c_t$ | Cost of unit exchanged at trade $t$ |

## C.2 Market Efficiency Metrics

These metrics evaluate the aggregate performance of the market in extracting potential gains from trade.

#### C.2.1 Allocative Efficiency

The primary efficiency metric from Smith (1962) and Gode and Sunder (1993) measures the percentage of maximum possible surplus actually realized:

$$E = \frac{\sum_{t=1}^{T}(v_t - c_t)}{TS^*} \times 100 \tag{13}$$

If traders exchange units where $c_t > v_t$ (negative surplus trades), the numerator decreases, lowering efficiency. Table **??** provides benchmark values from the literature.

Table 28. Allocative Efficiency Benchmarks

| Trader Type | Expected $E$ | Reference |
|---|---|---|
| ZI (unconstrained) | 60-70% | Gode & Sunder 1993 |
| ZIC (constrained) | 98.7% | Gode & Sunder 1993 |
| ZIP | 99.9% | Cliff & Bruten 1997 |
| GD | >99.9% | Gjerstad & Dickhaut 1998 |
| Mixed tournament | 89.7% | Rust et al. 1994 |

#### C.2.2 Efficiency Loss Decomposition

Following Rust et al. (1994), the total lost surplus (100% $-$ $E$) can be decomposed into four components. Define intra-marginal units as those that should trade ($q \leq Q^*$) and extra-marginal units as those that should not trade ($q > Q^*$).

Intra-marginal loss (IM) represents surplus lost from failing to trade profitable units:

$$IM = \sum_{q \in \text{Untraded Intra-marginal}} \big(D(q) - S(q)\big) \tag{14}$$

Extra-marginal loss (EM) represents negative surplus from trading units that should not have been traded:

$$EM = \sum_{t \in \text{Extra-marginal trades}} (c_t - v_t) \tag{15}$$

Buyer displacement (BS) captures surplus lost when an extra-marginal buyer displaces an intra-marginal buyer. Seller displacement (SS) captures surplus lost when an extra-marginal seller displaces an intra-marginal seller.

The decomposition identity states:

$$100\% - E = IM + EM + BS + SS \tag{16}$$

### C.3 Price Convergence Metrics

These metrics measure the tendency of transaction prices to approach the equilibrium price $P^*$.

#### C.3.1 Root Mean Squared Deviation

Following Gode and Sunder (1993), the RMSD measures the distance of prices from equilibrium:

$$RMSD = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(p_t - P^*)^2} \tag{17}$$

#### C.3.2 Smith's Alpha

The coefficient of convergence from Smith (1962) normalizes the standard deviation of prices around equilibrium by the equilibrium price. Let $\sigma_0$ be the root mean squared deviation of prices around equilibrium:

$$\sigma_0 = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(p_t - P^*)^2} \tag{18}$$

Smith's alpha is then:

$$\alpha = \frac{100 \cdot \sigma_0}{P^*} \tag{19}$$

Lower values of $\alpha$ indicate tighter convergence to equilibrium. Some sources use a scaling factor of 1000 instead of 100, though the interpretation remains the same.

#### C.3.3 Price Standard Deviation

The raw volatility measure captures dispersion around the mean transaction price:

$$\sigma_p = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(p_t - \bar{p})^2} \tag{20}$$

where $\bar{p} = (1/T)\sum_{t=1}^{T} p_t$ is the mean transaction price. ZIC traders exhibit high, constant volatility (2-3 times human levels), while ZIP and GD traders show declining volatility as they learn.

#### C.3.4 Price Volatility Percentage

For cross-market comparison, volatility can be normalized:

$$\text{Volatility\%} = \frac{\sigma_p}{\bar{p}} \times 100 \tag{21}$$

Values below 5% indicate good convergence, while values above 20% indicate an unstable market.

### C.3.5 Hit Rate

From the Santa Fe Tournament, the hit rate measures the percentage of trades within a band around equilibrium:

$$H = \frac{|\{t : |p_t - P^*| \leq 0.05 \cdot P^*\}|}{T} \tag{22}$$

### C.3.6 Mean Absolute Deviation

Following Gjerstad and Dickhaut (1998):

$$MAD = \frac{1}{T} \sum_{t=1}^{T} |p_t - P^*| \tag{23}$$

ZIP traders typically achieve MAD of approximately \$0.08, while GD traders achieve approximately \$0.04.

## C.4 Trader Performance Metrics

These metrics evaluate individual agents rather than the market as a whole.

### C.4.1 Individual Profit

Raw earnings for trader $i$ are computed as follows. For a buyer:

$$\pi_i = \sum_{k \in \text{Items Traded}} (v_{ik} - p_k) \tag{24}$$

For a seller:

$$\pi_j = \sum_{k \in \text{Items Traded}} (p_k - c_{jk}) \tag{25}$$

### C.4.2 Equilibrium Profit

The theoretical profit at competitive equilibrium represents what trader $i$ would earn if all trades occurred at $P^*$. For a buyer:

$$\pi_i^* = \sum_{k: v_{ik} > P^*} (v_{ik} - P^*) \tag{26}$$

For a seller:

$$\pi_j^* = \sum_{k: c_{jk} < P^*} (P^* - c_{jk}) \tag{27}$$

### C.4.3 Profit Deviation

The difference between actual and equilibrium profit indicates whether a trader extracted more or less than their fair share:

$$\Delta\pi_i = \pi_i - \pi_i^* \tag{28}$$

Positive values indicate the trader extracted more than fair share, zero indicates exactly fair share, and negative values indicate underperformance or exploitation.

### C.4.4 Individual Efficiency Ratio

Following Chen and Tai (2010), the ratio of actual to theoretical profit:

$$E_i = \frac{\pi_i}{\pi_i^*} \tag{29}$$

Values greater than 1 indicate the trader captures more than their equilibrium share (exploiter), values equal to 1 indicate exactly equilibrium share, and values less than 1 indicate the trader is being exploited. Table **??** provides benchmark values.

Table 29. Individual Efficiency Ratio Benchmarks

| Trader Type | Expected $E_i$ |
|---|---|
| Kaplan (mixed market) | 1.14-1.21 |
| ZIC | ≈1.0 |
| ZIP/GD | ≈1.0 |
| Kaplan (pure market) | 0.5-0.6 |

### C.4.5 Profit Dispersion

This metric from Cliff and Bruten (1997) is the key metric for discriminating intelligent from zero-intelligence traders. It measures the cross-sectional RMS difference between actual and equilibrium profits:

$$PD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\pi_i - \pi_i^*)^2} \tag{30}$$

where $N$ is the total number of traders.

ZIC traders exhibit profit dispersion values of 0.35-0.60, reflecting random surplus allocation. ZIP traders achieve approximately 0.05 after convergence, demonstrating that fair allocation emerges from learning. ZIP achieves 7-10 times lower dispersion than ZIC. Even with similar allocative efficiency, profit dispersion reveals whether the "right" traders are earning profits.

### C.4.6 Number of Trades

The activity level for agent $i$:

$$N_i = |\{t : \text{agent } i \text{ participated in trade } t\}| \tag{31}$$

Kaplan typically has fewer trades than ZIC due to its waiting strategy.

## C.5 Dynamic Metrics

### C.5.1 Price Autocorrelation

This metric tests whether price changes predict subsequent changes:

$$\rho = \text{Corr}(\Delta p_t, \Delta p_{t-1}) \tag{32}$$

where $\Delta p_t = p_t - p_{t-1}$.

Negative values indicate mean-reversion where prices overshoot then correct. Zero indicates a random walk with no predictability. Positive values indicate momentum or trending. Empirically, $\rho \approx -0.25$ was found by Rust et al. (1994), rejecting Wilson's (1987) martingale prediction of $\rho = 0$.

### C.5.2 Gode-Sunder Convergence Coefficient

From Gode and Sunder (1993), this metric tests whether the market learns within a period. Let $y_t$ be the root mean squared deviation of transaction prices at sequence number $t$, calculated across $N$ experimental runs:

$$y_t = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(p_{t,n} - P^*)^2} \tag{33}$$

Regress $y_t$ against $t$:

$$y_t = \alpha + \beta \cdot t + \epsilon_t \tag{34}$$

Negative $\beta$ indicates the market is converging (variance shrinking), while $\beta \approx 0$ indicates the market is stagnant (common in ZI unconstrained). The regression is performed on ensemble RMSD across multiple runs, not single-run squared error, to reduce noise.

### C.5.3 Convergence Time

The number of periods until prices stabilize within a tolerance of equilibrium:

$$T^* = \min\{t : |p_t - P^*| \leq 0.05 \cdot P^*\} \tag{35}$$

GD typically achieves $T^* < 1$ period, ZIP requires 1-2 periods, and ZIC never converges due to absence of learning.

### C.5.4 Time of Last Transaction

From Rust et al. (1994), this metric measures liquidity risk and closing panics:

$$T_{last} = \max_t(\tau_t) \tag{36}$$

where $\tau_t$ is the timestamp of trade $t$ and $T_{max}$ is maximum time allowed.

If $T_{last} \approx T_{max}$ consistently, this indicates "wait in background" strategies (like Kaplan) causing deadline congestion.

### C.5.5 Rank Correlation of Efficient Order

This metric measures whether the "right" trades happened in the "right" order. Theory suggests the highest-value buyer should trade with the lowest-cost seller first. Let $R_{actual}$ be the rank vector of trades by surplus as they occurred and $R_{ideal}$ be the rank vector sorted by theoretical surplus. Then:

$$\rho_s = \text{Spearman}(R_{actual}, R_{ideal}) \tag{37}$$

A value of $\rho_s = 1.0$ means the market perfectly executed the most profitable trades first.

## C.6 Evolutionary Metrics

For long-run tournament analysis following Rust et al. (1994) and Chen and Tai (2010).

### C.6.1 Capital Stock Evolution

The market share of strategy $i$ at game or generation $g$:

$$K_{i,g} = K_{i,g-1} + \pi_{i,g} - S_{i,g} \tag{38}$$

where $S_{i,g}$ is the theoretical surplus assigned to trader $i$.

Strategies with $K$ trending upward are evolutionarily stable; those trending to zero are eliminated.

### C.6.2 Generations to Convergence

From Chen and Tai (2010), this learning speed metric is defined as:

$$Gen^* = \min\{g : E_{pop,g} \geq E_{target}\} \tag{39}$$

where $E_{pop,g}$ is the average efficiency at generation $g$ and $E_{target}$ is a threshold (e.g., 99%).

## C.7 Microstructure Metrics

### C.7.1 Initiator Price Bias

From Gjerstad and Dickhaut (1998), this metric measures the difference between buyer-initiated and seller-initiated trade prices. Let $T_{buy}$ denote trades where the buyer accepted the standing ask, and $T_{sell}$ denote trades where the seller accepted the standing bid.

$$\bar{p}_{buy} = \frac{1}{|T_{buy}|} \sum_{t \in T_{buy}} p_t \tag{40}$$

$$\bar{p}_{sell} = \frac{1}{|T_{sell}|} \sum_{t \in T_{sell}} p_t \tag{41}$$

$$\Delta_{init} = \bar{p}_{sell} - \bar{p}_{buy} \tag{42}$$

In human markets, $\Delta_{init} \neq 0$ indicates asymmetric urgency between buyers and sellers.

### C.7.2 ZIP Margin Adjustment

From Cliff and Bruten (1997), the learning dynamics of the profit margin $\mu$:

$$\Delta\mu_i(t) = \beta \cdot (Target_i(t) - p_i(t)) \tag{43}$$

where $\beta$ is the learning rate parameter.

The optimal $\beta$ that matches human data becomes an outcome when calibrating agent behavior to empirical markets.