Mechanism Design for Multi-Agent Alpha Discovery: Optimizing Agent Distribution in Heterogeneous LLM Markets

author names withheld

Under Review for NExT-Game 2026

Abstract

We frame multi-agent stock selection as a mechanism design problem: a principal (optimizer) allocates aggregation weight across heterogeneous LLM-powered investor types to maximize portfolio alpha. Drawing on the market disagreement hypothesis, we study mechanisms that reward consensus and penalize disagreement among agents. Through systematic ablation of four mechanism design choices - objective alignment, learnable aggregation weight, optimizer capability, and stability constraint - we demonstrate that these choices interact non-trivially: individual improvements can degrade performance when paired with an insufficient optimizer, and weak stability constraints are worse than none. Critically, at 64 agents the simple baseline outperforms all improved configurations (10d Rank IC 0.033 ± 0.010 vs 0.023 and 0.021), while a single 512-agent run suggests that the same improvements can become useful at larger scale (0.044 vs 0.033). We also find substantial run-to-run variability that challenges single-seed conclusions common in the literature. These results connect adaptive mechanism design, Stackelberg-style aggregation, and LLM-based strategic reasoning in financial markets.

1. Introduction

Large language models (LLMs) are increasingly deployed as autonomous agents in complex decision-making environments, including financial forecasting and trading [5, 7, 8]. A central question in multi-agent LLM systems is *mechanism design*: how should a principal allocate resources across heterogeneous agents to maximize collective performance?

We study this question in the context of the Multi-Agent Stock Selection (MASS) framework [2], which simulates a market of heterogeneous LLM-powered investor types. Each type receives different data modalities (price history, sector returns, macro indicators) and generates investment signals. A backward optimizer then adjusts the aggregation weights over agent types to maximize the predictive power of the aggregated signal. This system naturally maps to a *Stackelberg-style* mechanism: the principal (optimizer) commits to an aggregation rule, the agents generate investment decisions from their local views, and the principal adapts based on observed outcomes.

The key insight from MASS is the *market disagreement hypothesis*: consensus among agents predicts future returns, while disagreement predicts lower returns [1, 6]. The combined signal is $S = \alpha \cdot \text{consensus} - (1 - \alpha) \cdot \text{disagreement}$, where α controls the tradeoff. The original system uses $\alpha = 0.5$ and optimizes the distribution via simulated annealing (SA) [4] on a low-disagreement signal.

We identify four mechanism design choices in this pipeline and systematically ablate them:

1. **Objective alignment**: The optimizer should maximize the combined signal's Rank IC, not just disagreement.
2. **Learnable α**: The consensus-disagreement tradeoff should be optimized, not hardcoded.
3. **Optimizer capability**: SA's random perturbations cannot handle the expanded search space; CMA-ES with covariance adaptation is needed.
4. **Stability constraint**: A turnover penalty prevents the mechanism from overfitting to short-term noise.

Our contributions are: (1) a game-theoretic reframing of backward optimization as adaptive mechanism design; (2) systematic ablation showing these choices interact non-trivially—individual fixes can *degrade* performance when paired with an insufficient optimizer; (3) multi-seed experiments (3 seeds per config) revealing that the simple baseline outperforms all improved configurations at 64 agents, and that a single-seed improvement reversed with more seeds; (4) the finding that weak stability constraints are worse than none; and (5) preliminary evidence that mechanism design improvements are scale-dependent: they fail at 64 agents, but one 512-agent run improves over the 64-agent baseline by 31%.

2. Method

2.1. Stackelberg-Style Formulation

The MASS system defines a repeated Stackelberg-style aggregation game:

- **Leader (Principal)**: At each trading day t, chooses nonnegative aggregation weights $\mathbf{d}_t$ over n agent types, plus aggregation weight $\alpha \in [0, 1]$ and stability parameter $\lambda \geq 0$. The weights are normalized inside the scoring rule.
- **Followers (Agents)**: Each agent type i generates investment signal $s_{t,i}$ based on its data modality and investment style; the principal's weights determine how those signals are aggregated.
- **Aggregation**: The combined signal is $S_t = \alpha\bar{s}_t - (1 - \alpha)\sigma_t$, where $\bar{s}_t$ is the consensus (weighted mean) and σ_t is the disagreement (weighted standard deviation).
- **Payoff**: The principal's objective is RankIC$(S_t, R_{t+\Delta})$, the Spearman correlation between the signal and future returns at horizon Δ.

Unlike classical Stackelberg games where the leader commits once, here the leader reweights the aggregation mechanism over time—an adaptive mechanism design problem with non-stationary agent signals driven by changing market regimes.

2.2. Mechanism Design Choices

We identify four design choices in the mechanism, each corresponding to a principled fix:

Fix 1: Objective alignment. The original system optimizes Rank IC of the low-disagreement component ($-\sigma_t$) against returns, but the actual trading signal is $S_t = \alpha\bar{s}_t - (1 - \alpha)\sigma_t$. Optimizing only one component creates a misalignment: the optimizer may improve the component signal without improving the combined signal. We fix this by optimizing Rank IC of S_t directly.

Fix 2: Learnable α. The aggregation weight $\alpha = 0.5$ is hardcoded, but prior evidence shows consensus alone ($\alpha = 1.0$) outperforms the combined signal (10d Rank IC: 0.109 vs. 0.073). We add α to the optimizer's search space, allowing the mechanism to discover a better consensus-disagreement tradeoff.

Fix 3: Optimizer capability. SA performs 20 random pairwise weight perturbations per optimization step. When α is added to the search space (dimensionality increases from 8 to 9), SA's random exploration is insufficient—it cannot model correlations between agent type weights. We replace SA with CMA-ES [3], which maintains a full covariance matrix of the search distribution and adapts both step size and search direction based on fitness rankings.

Fix 4: Stability constraint. Without penalty, the optimizer can radically change $\mathbf{d}_t$ between steps, potentially overfitting to short-term noise. We add a turnover penalty: fitness = RankIC $- \lambda \cdot \|\mathbf{d}_t - \mathbf{d}_{t-1}\|_1/2$. This is a mechanism design choice: the principal explicitly controls the adaptivity rate.

3. Experiments

3.1. Setup

We run all configurations on the Chinese A-share market (SSE50 index, "ih" stock pool) with 8 investor types × 8 agents per type $= 64$ agents, from December 2022 to June 2023. We also run the best configuration (Config E) at 32 investor types × 16 agents $= 512$ agents to test scaling. Each agent type receives a distinct combination of data modalities: base OHLCV, cross-industry returns, price-value features, and macroeconomic indicators. Evaluation uses Rank IC from January 2023 onward (excluding a 1-month warmup). We report 10-day forward return Rank IC as the primary metric.

Each configuration is run with multiple random seeds to assess variability. LLM stochasticity (temperature sampling in agent decisions) and random stock selection introduce run-to-run variance.

3.2. Multi-Seed Ablation Results

We define six configurations that incrementally apply the four fixes. Table 3.2 reports 10d Rank IC for each.

Config	Fitness	α	Optimizer	λ	10d Rank IC
A (baseline)	Disagreement	0.5 fixed	SA	0	$\mathbf{0.033 \pm 0.010}$
B	Combined	0.5 fixed	SA	0	0.007
C	Combined	Learned	SA	0	-0.019
D	Combined	Learned	CMA-ES	0	0.023 ± 0.006
F	Combined	Learned	CMA-ES	0.01	0.013 ± 0.026
E	Combined	Learned	CMA-ES	0.1	0.021 ± 0.004
E-512	Combined	Learned	CMA-ES	0.1	0.044
A-512	Disagreement	0.5 fixed	SA	0	-0.006

Finding 1: Individual fixes can degrade performance. Fix 1 alone (Config B) drops Rank IC from 0.033 to 0.007. Adding learnable α with SA (Config C) collapses performance to -0.019.

Only when CMA-ES is introduced (Config D) does performance partially recover to 0.023—still below baseline. This demonstrates an *optimizer capability mismatch*: expanding the search space without upgrading the optimizer is counterproductive.

Finding 2: Weak stability constraints are worse than none. Config F (λ=0.01, 3 seeds) achieves Rank IC of 0.013 ± 0.026—the highest variance and lowest mean among CMA-ES configs. One seed gets 0.048, another -0.010, a third 0.000. This is worse than Config D (λ=0, 0.023 ± 0.006) or Config E (λ=0.1, 0.021 ± 0.004). A weak penalty is too weak to stabilize CMA-ES but strong enough to distort the optimization landscape. This non-monotonic λ sensitivity connects to the exploration-exploitation tradeoff in bandit problems: either full freedom or strong constraint works, but a half-measure is worst.

Finding 3: Multi-seed variability challenges single-seed conclusions. Our earlier single-seed results suggested Config E (0.057) beat baseline A (0.053). With 3 seeds each, baseline A (0.033 ± 0.010) outperforms both D (0.023 ± 0.006) and E (0.021 ± 0.004). The single-seed improvement was favorable variance, not a real effect. This highlights the importance of multi-seed evaluation in LLM-based systems, where stochasticity is substantial.

Finding 4: The simple baseline is hard to beat at 64 agents. Config A (SA + Signal_std + fixed α=0.5) achieves the highest 3-seed mean Rank IC (0.033) at 64 agents. Neither CMA-ES with learned α (D: 0.023) nor with turnover penalty (E: 0.021) improves upon it. The disagreement fitness function with simple SA optimization is already near-optimal for this problem scale—principled improvements increase the search space but cannot reliably find better solutions.

Finding 5: Mechanism improvements are scale-dependent—and the baseline fails at scale. Config E at 512 agents (32 investor types × 16 agents per type) achieves 10d Rank IC of 0.044, beating the 64-agent baseline (0.033) by 31%. Critically, the SA baseline at 512 agents (A-512) **collapses to** -0.006—SA's random perturbation strategy cannot handle the 32-dimensional weight space. CMA-ES with covariance adaptation succeeds where SA fails. This is not merely that improvements help at scale; the simple optimizer that works at 64 agents becomes *catastrophic* at 512 agents, making the mechanism design improvement **necessary** rather than optional.

3.3. Consensus vs. Combined Signal

Across all configurations, the consensus signal (Signal_mean) outperforms the combined signal (Signal). For the baseline (Config A), 10d consensus Rank IC is approximately double the combined Rank IC. This suggests the optimal α favors consensus heavily, consistent with the learned α values in Configs D and E (median $\alpha > 0.5$). However, despite consensus being the stronger signal component, optimizing directly for the combined signal (Configs B–E) does not improve over optimizing for disagreement alone (Config A).

4. Discussion

Mechanism design as a lens for multi-agent LLM systems. Our results show that mechanism design choices—objective alignment, parameter learning, optimizer capability, and stability—interact in non-trivial ways in multi-agent LLM systems. At 64 agents, principled improvements do not help: the simple baseline outperforms all improved configurations. At 512 agents, the same improvements become *necessary*: the SA baseline collapses to -0.006 while CMA-ES achieves 0.044. This reveals a scale-dependent optimizer capability threshold: the simple optimizer that suffices at small scale fails catastrophically at larger scale, making sophisticated optimization essential.

Optimizer-agent capability matching. A key insight is that the optimizer's capability must match the complexity of the mechanism design problem. SA suffices for the original 8-dimensional search (fixed α, Signal_std fitness), but fails catastrophically when α is added (9 dimensions). This mirrors findings in reinforcement learning where agent capability must match task complexity.

Non-monotonic stability. The λ sensitivity result—weak penalty worse than none—has implications beyond finance. In any adaptive mechanism, the principal must choose not just *whether* to constrain adaptation, but *how strongly*. A weak constraint creates a false sense of stability while still allowing harmful oscillations.

Variability in LLM-based systems. The substantial run-to-run variability we observe (e.g., Config A: 0.020 to 0.044 across seeds; Config D: 0.019 to 0.031) raises concerns about single-seed conclusions in the growing literature on LLM-based multi-agent systems. Our own single-seed result (E $>$ A) reversed with 3 seeds. We recommend multi-seed evaluation as standard practice.

Limitations. Our experiments are limited to a single market (Chinese A-shares) and a single time period (6 months). The 512-agent result is based on a single seed, so its advantage over the 64-agent baseline requires confirmation with additional seeds. We are running an analogous U.S. equity experiment; if completed and validated, the camera-ready version will include it as a cross-market generalization check. The computational cost ($\sim$8K LLM calls per 64-agent seed, $\sim$128K for 512 agents) also limits the number of seeds we can practically run.

References

[1] Karl B. Diether, Christopher J. Malloy, and Anna Scherbina. Differences of opinion and the cross section of stock returns. *The Journal of Finance*, 57(5):2113–2141, 2002. doi: 10.1111/0022-1082.00490.

[2] Taian Guo, Haiyang Shen, JinSheng Huang, Zhengyang Mao, Junyu Luo, Binqi Chen, Zhuoru Chen, Luchen Liu, Bingyu Xia, Xuhui Liu, Yun Ma, and Ming Zhang. Mass: Multi-agent simulation scaling for portfolio construction. *arXiv preprint arXiv:2505.10278*, 2025.

[3] Nikolaus Hansen. The cma evolution strategy: A comparing review. *Towards a New Evolutionary Computation*, 2006.

[4] Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[5] K. J. L. Koa et al. Learning to generate explainable stock predictions using self-reflective large language models. In *Proceedings of the ACM Web Conference*, 2024. doi: 10.1145/3589334.3645611.

[6] Edward M. Miller. Risk, uncertainty, and divergence of opinion. *The Journal of Finance*, 32 (4):1151–1168, 1977. doi: 10.1111/j.1540-6261.1977.tb03317.x.

[7] Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. Tradingagents: Multi-agents LLM financial trading framework. In *The First MARW: Multi-Agent AI in the Real World Workshop at AAAI 2025*, 2025. URL `https://openreview.net/forum?id=4QPrXwMQt1`.

[8] Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan W. Suchow, Zhenyu Cui, Qianqian Xu, et al. Fincon: A synthesized LLM multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. In *Advances in Neural Information Processing Systems*, 2024.