WHEN LEARNERS SHAPE REWARDS: LEARNING DYNAMICS IN TRADING POPULATIONS

Anonymous Authors
Paper under double-blind review

ABSTRACT

Market simulation is valuable for understanding collective market behavior and evaluating trading strategies under interaction. Existing work emphasizes how traders interact or how well individual agents perform. Less clear is how collective behavior and the benefits of learning change as more participants continually revise their decision rules. We introduce FactorMarket Lab (FML), coupling factor-based stock predictors with a multi-stock order-book market. Market interaction generates new training examples that, once their labels become available, enter subsequent model updates. Traders share an initial model within each family but independently sample available data and update separate copies. Historical experiments demonstrate learning benefits, and interactive measurements show behavioral differentiation from a shared start. We vary peer learning density and update frequency, comparing the same focal cohort with and without learning. Across the tested populations, behavioral differentiation varies with learning density and update timing, with distinct responses across models. The gain from a trader's own updates also depends on how its peers learn: more widespread or frequent learning is not uniformly better. In mixed-model populations, conditions favoring collective outcomes need not maximize focal learning gains. These findings motivate evaluating continual learning as a population process, not only as an individual capability.

1 INTRODUCTION

Market simulation is valuable for understanding how individual decisions produce collective market behavior and for evaluating strategies under interaction (Lux & Marchesi, 1999; LeBaron, 2006; Byrd et al., 2020). Traders do not only respond to prices: they also learn from experience and revise their decision rules. Such updates can change both their own performance and the market faced by others. Capturing this feedback matters when using simulation to study the consequences of deploying learning-based traders.

Existing research provides two complementary foundations. Market simulators emphasize interaction and emergent outcomes (Vidler & Walsh, 2024; Yang et al., 2025), while financial learning agents develop decision-making capabilities through memory, reflection, and feedback (Zhang et al., 2024; Yu et al., 2025; Chen et al., 2025). Learning is also incorporated in market simulation (Amrouni et al., 2021; Mascioli et al., 2024; Zou et al., 2026). Yet evaluating individual learners does not by itself explain what happens when many traders learn together, and incorporating adaptive agents into a simulator leaves open how the number of learning traders and their update frequency affect collective outcomes.

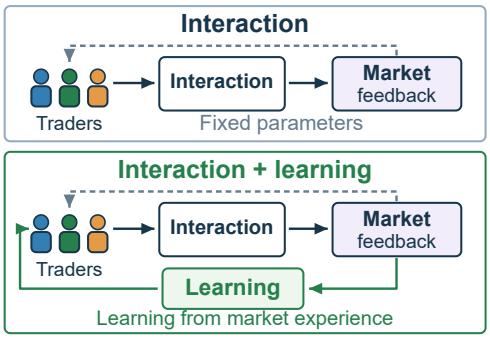


Figure 1: Interaction changes the market; continued learning changes the decision rules of its participants.

We take a population-learning perspective on market simulation: *from interacting traders to traders who learn through interaction*, as illustrated in

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Figure 1. Market feedback not only changes the information available for the next decision; it also supplies experience for updating the decision rule itself. Updated traders then return to the market, shaping the interactions from which subsequent learning data arise. This feedback connects learning to the environment it helps generate (Perdomo et al., 2020; Mendler-Dünner et al., 2020). Our focus is therefore not simply to place a learning agent in a market, but to study learning as a process distributed across its participants.

To realize this perspective, we introduce *FactorMarket Lab* (FML), which connects factor-based prediction, trading, and continued learning. As shown in Figure 2, predictions guide orders, and market outcomes supply labeled examples for subsequent updates. To enable diversity from a shared start, we design traders to sample independently from available historical and market-generated data and update separate model copies. A variance-based proposition clarifies when this produces parameter dispersion. Historical learning gains and behavioral differentiation in FML establish the foundations for studying learning populations.

We then ask how learning participation and update frequency change population behavior and the value of continued learning. We vary peer learning density ρ and frequency ν, comparing the same focal cohort with and without updates to separate its own learning from the influence of learning peers. Across MLP, Transformer, and Qwen populations, **behavioral differentiation depends on learning density and update timing, with distinct responses across learners. The value of continued learning depends on how other participants learn**, without a common best schedule across models. Mixed-model populations further show that **conditions favoring collective outcomes need not maximize focal learning gains.**

Our contributions are threefold. **(i) An interactive learning framework:** FML connects factor-based trading, market-generated training examples, and continued parameter updates within a shared market, with independent sampling and separate model updates designed to enable trader diversity from a common initialization. **(ii) A population-learning experimental design:** matched focal comparisons distinguish the effects of one's own updates from the influence of learning peers as their learning density and update frequency vary. **(iii) Insights into learning in trading populations:** learning participation and update timing shape behavioral differentiation and focal learning gains, while collective and focal benefits need not align. These findings motivate evaluating learning within the population in which it occurs.

2 Related Work

Financial trading agents. Financial language models such as FinBERT and FinGPT support financial text analysis and domain adaptation (Huang et al., 2023; Liu et al., 2023), with broader applications reviewed by Lee et al. (2024); Nie et al. (2024). Trading agents incorporate memory, reasoning, and feedback into decision-making. FinAgent combines multimodal information with tools and reflection (Zhang et al., 2024), and FinMem uses layered memory to organize financial experience (Yu et al., 2025). FinCon and TradingAgents coordinate specialized agents for analysis and trading decisions (Yu et al., 2024; Xiao et al., 2024). FinHEAR incorporates expert-guided retrieval, position sizing, and feedback-driven reasoning (Chen et al., 2025). These studies inform the design of individual traders and collaborating teams. Our focus is complementary: how continued parameter updates affect traders when other market participants also learn.

Interactive market simulation. Agent-based finance studies how heterogeneous and adapting participants produce market dynamics (Lux & Marchesi, 1999; Hommes, 2006; LeBaron, 2006). ABIDES provides discrete-event market simulation (Byrd et al., 2020), while ABIDES-Gym and PyMarketSim support the training and evaluation of trading agents (Amrouni et al., 2021; Mascioli et al., 2024). Message-driven simulation and GPU-based order books improve computational efficiency (Belcak et al., 2021; Frey et al., 2023). More recent work explores bilateral trading in TraderTalk (Vidler & Walsh, 2024), behavioral and social emergence in TwinMarket (Yang et al., 2025), and adaptation and strategy evolution in FinEvo (Zou et al., 2026). These approaches provide foundations for studying interaction and adaptation. FML builds on this perspective by varying the fraction of learning traders and their update frequency, and observing both population behavior and the benefits of focal learning.

Learning dynamics in multi-agent populations. Evolutionary game theory describes changes in population strategy frequencies (Hofbauer & Sigmund, 1998; Sandholm, 2010), and analyses of multi-agent learning connect update rules to repeated-game dynamics (Tuyls et al., 2006; Mannor & Shamma, 2007). League training in AlphaStar (Vinyals et al., 2019) and autocurricula (Leibo et al., 2019) illustrate learning through interaction with other agents. Population-based training adapts models and hyperparameter schedules within a training population (Jaderberg et al., 2017). Performative prediction studies a related feedback: deployed predictions can change the data distribution used for subsequent learning (Perdomo et al., 2020; Mendler-Dünner et al., 2020). We study persistent traders with independently updated parameters, rather than changes in the population shares of fixed strategies. Drawing on these perspectives, we examine how learning distributed across market participants shapes collective behavior and the value of an individual trader's updates.

3 From Updating Models to Learning Populations

FML couples market interaction with continued model updating. Traders submit orders based on predictions; the resulting market trajectory supplies new observations and, once outcomes become available, labeled training examples. These examples join historical data in the learning pool. Independently sampled updates then inform subsequent trading decisions. The formulation connects individual decision rules, joint market dynamics, and learning from the trajectory they help generate.

3.1 Agent–environment formulation

Individual decisions. Let $\mathcal{I} = \{1, \dots, N\}$ index traders and Z_t denote the market state. Trader i receives $o_{i,t} = \mathcal{O}_i(Z_t)$, comprising public features $X_t \in \mathbb{R}^{J \times K}$ and prices P_t, together with its account information. For stock j, its parameterized scorer and decision maps are

$$\begin{aligned} \alpha_{i,j,t} &= f_{\theta_{i,t}}(x_{j,t}), \qquad p_{i,t} = \mathrm{softmax}(\alpha_{i,t}/T_{\mathrm{soft}}), \\ w^*_{i,t} &= \mathcal{P}(p_{i,t}; o_{i,t}), \quad q_{i,t} = \mathcal{Q}(w^*_{i,t-1}; o_{i,t}). \end{aligned} \tag{1}$$

Here α predicts relative return, $\mathcal{P}$ maps stock scores to feasible target weights, and $\mathcal{Q}$ maps the stored target to buy/sell orders. The lag separates signal formation from execution. A frozen trader keeps θ_i fixed, but its actions still depend on changing observations.

Joint interaction. The population acts through $\mathbf{q}_t = (q_{1,t}, \dots, q_{N,t})$, inducing

$$(Z_{t+1}, \mathcal{F}_{t+1}) = \mathcal{E}(Z_t, \mathbf{q}_t, q_t^{\mathrm{bg}}; \xi_{t+1}), \tag{2}$$

where q_t^{bg} denotes background orders, ξ exogenous inputs, and $\mathcal{F}$ fills. The resulting state determines both subsequent observations and marked wealth $W_{i,t} = c_{i,t} + h_{i,t}^\top P_t$. Thus traders share an endogenous observation process even when their learning updates are independent.

Learning from the induced trajectory. Let $y_{j,s}$ denote the relative return following signal date s, observable at $s + 2$. The available pool and prediction loss are

$$\mathcal{D}_t = \mathcal{D}_{\mathrm{hist}} \uplus \{(x_{j,s}, y_{j,s}) : s + 2 \le t,\ j \in \mathcal{V}_s\}, \qquad \ell(\theta; x, y) = (f_\theta(x) - y)^2, \tag{3}$$

where $\uplus$ retains example multiplicities, $\mathcal{D}_{\mathrm{hist}}$ is the initial historical pool, and $\mathcal{V}_s$ the valid stock set. Supervision concerns price movements, not realized trading profit. Equations (1)–(3) connect individual prediction, joint action, and future training data; execution and label construction are specified in Appendix E.

3.2 From individual updates to population learning

Shared initialization, independent updates. Within each family, $\theta_{i,0} = \theta_0$. Index scheduled update events by k, occurring at t_k, and let $\zeta_{i,k}$ contain the trainable parameters and optimizer state immediately before an event. With learning indicator $u_i \in \{0, 1\}$, the population update is

$$\zeta_{i,k+1} = \begin{cases} \mathcal{U}(\zeta_{i,k}; \mathcal{B}_{i,k}), & u_i = 1, \\ \zeta_{i,k}, & u_i = 0, \end{cases} \qquad \mathcal{B}_{i,k} \overset{\mathrm{ind}}{\sim} \mathrm{Unif}(\mathcal{D}_{t_k})^{Lb}. \tag{4}$$

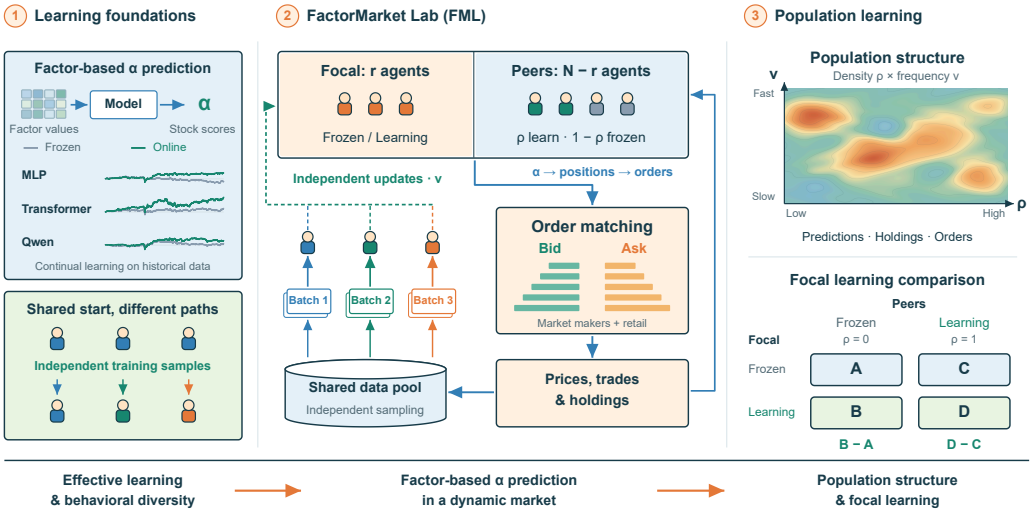


Figure 2: From factor-based learning to population interaction. **Left:** historical learning and shared-start differentiation establish the two prerequisites. **Center:** FactorMarket Lab links r focal traders and $N - r$ peers through order matching and independently sampled updates. **Right:** peer learning density ρ and active-learner frequency ν define comparisons of population structure and focal learning. A–D are the density endpoints. The field is conceptual, not measured evidence of an interior optimum. Historical insets show long-only marked returns before final liquidation; the main historical results include that liquidation.

Here $\mathcal{U}$ performs L loss-minimizing minibatch updates of size b; draws are independent with replacement conditional on the shared pre-update history. This separates a common information pool from independent learning trajectories. For Qwen, θ_i denotes the trainable head; the backbone and adapter remain frozen.

The following proposition clarifies why a shared data pool and initialization permit parameter differences when independent sampling induces nonzero update variance. For $n \geq 2$ same-family traders, define parameter dispersion by $D_\theta = n^{-1} \sum_i \|\theta_i - \bar{\theta}\|^2$, where $\bar{\theta} = n^{-1} \sum_i \theta_i$.

Proposition 1 (Parameter dispersion under independent sampling)**.** *Condition on the joint pre-update history $\mathcal{H}_k$. Suppose traders draw independent minibatches with replacement and update independently as in* (4)*. Let their resulting parameters have conditional means $\mu_{i,k}$ and finite covariances $V_{i,k}$. Then*

$$\mathbb{E}[D_{\theta,k+1} \mid \mathcal{H}_k] = \frac{1}{n} \sum_i \|\mu_{i,k} - \bar{\mu}_k\|^2 + \frac{n-1}{n^2} \sum_i \operatorname{tr} V_{i,k}, \tag{5}$$

where $\bar{\mu}_k = n^{-1} \sum_i \mu_{i,k}$. In particular, if all traders learn from identical parameter and optimizer states with the same update law, then $D_{\theta,k} = 0$ and $\mathbb{E}[D_{\theta,k+1} \mid \mathcal{H}_k] = (1 - 1/n) \operatorname{tr} V_k > 0$ whenever the common update covariance V_k has positive trace.

The proposition follows from a variance decomposition: independent sampling adds dispersion around the mean update without requiring different objectives or score perturbations. It establishes an increase from a shared start, not monotonic growth along the entire trajectory. Whether parameter variation survives prediction and execution is measured below. The proof and its minibatch interpretation are in Appendix A.

Learning participation and update frequency. Fix $F \subset \mathcal{I}$, $|F| = r$, before observing outcomes, with $u_i = a$ for $i \in F$ and $a \in \{0, 1\}$. Peer participation and update frequency are

$$\rho = \frac{\sum_{i \notin F} u_i}{N - r}, \qquad \nu = \frac{1}{\tau}, \qquad \rho_{\text{all}} = \frac{ar + \rho(N - r)}{N}. \tag{6}$$

All enabled learners share interval τ in the main comparison. With L, b fixed, ν changes both timing and training effort, not the optimizer learning rate. The relative-speed extension separates focal and peer intervals τ_f, τ_o at $\rho = 1$.

Focal learning comparisons. At fixed ν, the endpoints $(a, \rho) = (0, 0), (1, 0), (0, 1), (1, 1)$ define A, B, C, and D, respectively. Paired runs share initial states, focal identities, and external random streams. These are fixed-cohort interventions, not an exchangeability assumption; the grid is given in Appendix E.

3.3 Population behavior and learning returns

Behavioral differentiation. We measure $\mathcal{S}_t = (S_{\alpha,t}, S_{w,t}, S_{q,t})$. S_α is mean pairwise Pearson correlation between stock-score vectors; S_w is mean pairwise cosine similarity of target weights; S_q is cosine similarity of signed submitted-order notional vectors. Undefined constant-score or zero-vector pairs are excluded, not assigned perfect agreement. These quantities distinguish predictions, intended allocations, and order activity; they do not substitute for realized holdings or market liquidity.

For normalized behavior vectors v_i, similarity and dispersion satisfy $1 - S = nD_v/(n-1)$. Thus lower similarity corresponds to greater behavioral dispersion, distinct from parameter dispersion in Proposition 1. Appendix A.2 gives this identity and its within- and between-group decomposition.

Value to the focal cohort. Let $Y_\omega(a; \rho, \nu)$ be the mean cumulative net return of the focal cohort in market seed ω. Its learning value is the matched contrast

$$\Delta_{\text{learn}}(\rho, \nu) = \mathbb{E}_\omega[Y_\omega(1; \rho, \nu) - Y_\omega(0; \rho, \nu)]. \tag{7}$$

The endpoints $B - A$ and $D - C$ measure focal learning with frozen and learning peers, respectively; $C - A$ measures peer learning with the focal cohort frozen. Higher return can mean less loss, not profit. We therefore retain absolute returns, within-market ranks, and population wealth alongside these contrasts. Behavioral similarities describe how the population changes, but are neither market-liquidity measures nor identified causal mediators of learning value.

4 Establishing Learning and Differentiation

Before studying population learning, we examine two prerequisites: a scorer whose updates can improve historical performance, and an interacting population whose shared initialization permits subsequent behavioral differences.

FML setting. FML contains 32 research traders, 20 synthetic stocks, and 32 features, over 120 decision days. Each trader starts with 100,000 currency units and an independent copy of its model family's pretrained parameters. The interactive comparison is long-only with a 40% target cap. Research and retail fills incur 5 bps; market makers quote five levels with a 3 bps half-spread parameter. Realized spreads emerge from the book, with no additional historical execution offset. We use five market seeds for each of MLP, Transformer, and Qwen. Appendix E specifies training examples, model architectures, and accounting; Algorithm 1 gives the interactive learning loop.

4.1 Does continued learning help on historical data?

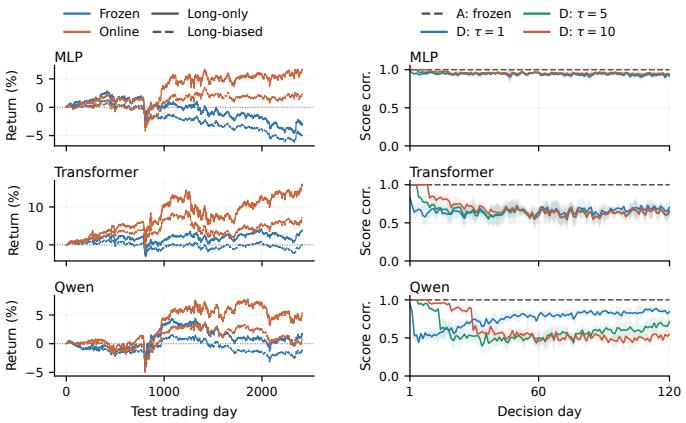
Setting. Each scorer is initially fitted on 2016 data and evaluated on the available 2017–2026 historical path, using 100 stocks and 32 features. Frozen and online models both receive current features and produce new predictions; only the online branch updates its parameters. We compare long-only exposure of 40%, long-biased exposure of 30% long and 10% short, and balanced exposure of 20% on each side. Positions persist, with a daily executed-notional limit of 10% of equity. The reference execution charges 5 bps per traded notional and a 3 bps adverse execution offset on each side. This historical execution proxy is distinct from the endogenous bid–ask spread in FML.

Observation. Figure 3(a) overlays the long-only and long-biased paths; Appendix H separates all three directions. Appendix C reports the final liquidated returns in Table 3. Online return exceeds frozen return for all three models and directions at this reference setting. Improvement does not always mean profit: the balanced portfolios remain negative. Across the 15 execution configurations evaluated per direction, online improves long-biased returns in all 15 for each model; the corresponding long-only counts are 15, 15, and 13 for MLP, Transformer, and Qwen, and the balanced

counts are 14, 15, and 10. The complete cost and turnover grid is in Appendix C. These reuse historical predictions and are sensitivity checks, not 15 independent markets. The result establishes a useful starting comparison: continued fitting can improve the deployed rule on this path. It does not establish universal predictability or imply that the improvement survives population interaction.

4.2 Can a common starting model produce different behavior?

Figure 3(b) compares prediction similarity in all-frozen and all-learning populations at three update intervals; Appendix H adds targets and orders. Frozen traders retain identical scores and target weights, whereas updating traders develop different predictions and intended allocations despite their common starting model.



(a) Historical returns (b) Prediction similarity

Figure 3: Learning and differentiation across three models. (a) Historical net returns: frozen vs. online, long-only vs. long-biased; 5 bps fees, 3 bps execution offset, 10% turnover cap, including liquidation. (b) FML: all frozen (A) vs. all learning (D), updating every τ days; five-seed means ± 1 SD.

MLP remains relatively similar after an initial separation, while Transformer exhibits stronger differentiation. Qwen reveals a different temporal pattern: with daily updates, mean score correlation rises from 0.598 over days 1–10 to 0.856 over days 111–120, an increase in every seed. At a ten-day interval it instead falls from 0.996 to 0.521, with a decrease in every seed. These windows describe trajectories, not an optimal interval. Thus a common starting point can lead to persistent differences, renewed agreement, or both at different times.

Even the all-frozen population develops different submitted orders as executions and holdings differ. Independent training draws are a source of parameter variation, but a shared-batch intervention would be needed to attribute observed behavioral differences specifically to sampling. The prerequisite for our next experiment is therefore not ever-increasing diversity, but a population whose behavior can evolve beyond identical model copies.

5 Population Behavior and Learning Returns

We examine how learning changes population behavior and the benefits received by individual traders. MLP, Transformer, and Qwen distinguish shared patterns from learner-dependent responses, rather than serve as a model leaderboard. Their training and update configurations differ, so these comparisons do not isolate architecture effects. Unless stated otherwise, results use long-only exposure and report day-120 outcomes across five market seeds. We vary peer learning density ρ and update frequency $\nu = 1/\tau$, where τ is the interval in days.

5.1 Population differentiation across learning densities and intervals

Figure 4 shows how prediction, target-weight, and order similarity vary with ρ and τ when focal learning is enabled. The mean prediction and target curves decline more clearly with peer density for Transformer than for MLP, whose predictions remain relatively similar. Qwen shows a stronger separation between update intervals.

The matched focal-frozen comparisons reinforce this commonality: enabling focal learning reduces density-averaged score and target similarity in all nine model–interval combinations, with the same sign in every seed (Appendix B). This comparator differs from the all-frozen A reference drawn in

Model	Focal outcome	A	B		C		D	
		Value	Value	ΔA	Value	ΔA	Value	ΔA
MLP	Return (%) ↑	−0.242	−0.157	+0.086	−0.221	+0.021	−0.155	+0.087
	Rank pct. ↑	46.05	48.63	+2.58	44.03	−2.02	46.61	+0.56
Transformer	Return (%) ↑	+0.066	+0.004	−0.062	−0.025	−0.090	−0.089	−0.154
	Rank pct. ↑	42.98	47.02	+4.03	48.79	+5.81	44.68	+1.69
Qwen	Return (%) ↑	−0.166	−0.311	−0.145	−0.189	−0.023	−0.232	−0.066
	Rank pct. ↑	29.03	29.52	+0.48	49.35	+20.32	45.89	+16.85

Table 1: Focal outcomes at $\tau = 5$, day 120; five-seed means. A: neither learns; B: focal only; C: peers only; D: both. ΔA subtracts A within each seed (return: pp; rank: percentile points). Rank pct. is the within-population percentile; ↑ means higher is better.

Figure 4. Updating more often does not invariably produce more differentiated orders: Qwen's mean order-similarity reduction is 0.056 with daily updates and 0.132 with ten-day updates. MLP instead shows a slight mean increase in order similarity, whereas Transformer shows reductions across all three decision layers.

Changing ρ changes which peers revise their decisions, while changing τ changes both when they update and how many updates they receive. Their effects need not be interchangeable. Orders also depend on existing holdings and execution constraints, so differences in predictions need not pass directly into trading. These are plausible channels, not an identified decomposition of the observed effects. **Population differentiation depends on both learning participation and update timing, with responses that vary across learners rather than following a common more-learning–more-diversity rule.**

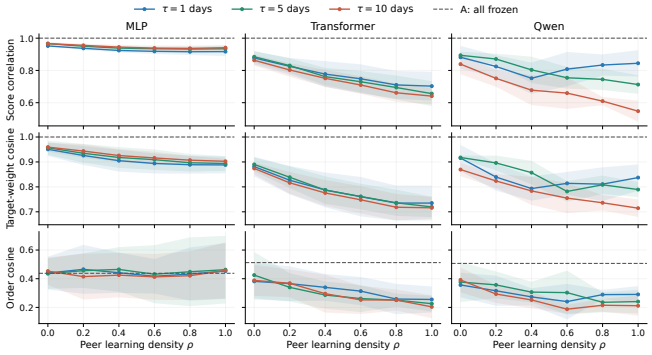


Figure 4: Population similarity at day 120. Rows: scores, target weights, and orders; columns: models. Colored curves enable focal learning at each peer density; dashed A freezes everyone. Bands show ±1 seed SD around five-seed means. Lower values indicate less similar behavior.

5.2 The value of learning depends on learning peers

We compare the same focal cohort with and without updates at each peer setting. At the density endpoints, $B - A$ and $D - C$ measure its learning gain with frozen and learning peers; $C - A$ measures the effect of peer learning on a frozen cohort. Absolute outcomes in Table 1 anchor these contrasts.

At $\tau = 5$, MLP benefits from focal learning with either frozen or learning peers, whereas Transformer and Qwen have negative mean gains in both cases. For Qwen, the loss from updating is smaller when peers learn: $D - C = -0.044$ pp versus $B - A = -0.145$ pp. This is a reduced disadvantage, not a positive learning gain. For Transformer and Qwen, B also has lower return than A but a higher percentile: relative position can improve without a financial improvement. Learning can change both the focal cohort and the market faced by everyone else. Rank captures relative standing within that market, whereas return measures the change in the cohort's own wealth; neither substitutes for the other.

Figure 5 shows model-dependent learning gains. Qwen's density-averaged mean is negative at all intervals despite lower order similarity (Appendix Table 2). MLP's five-day mean gain is 0.121 pp, but positive in only three seeds; Transformer's ten-day mean is 0.044 pp, with a negative median seed effect. Differentiation does not ensure a return advantage, and the mean alone can conceal substantial seed variation.

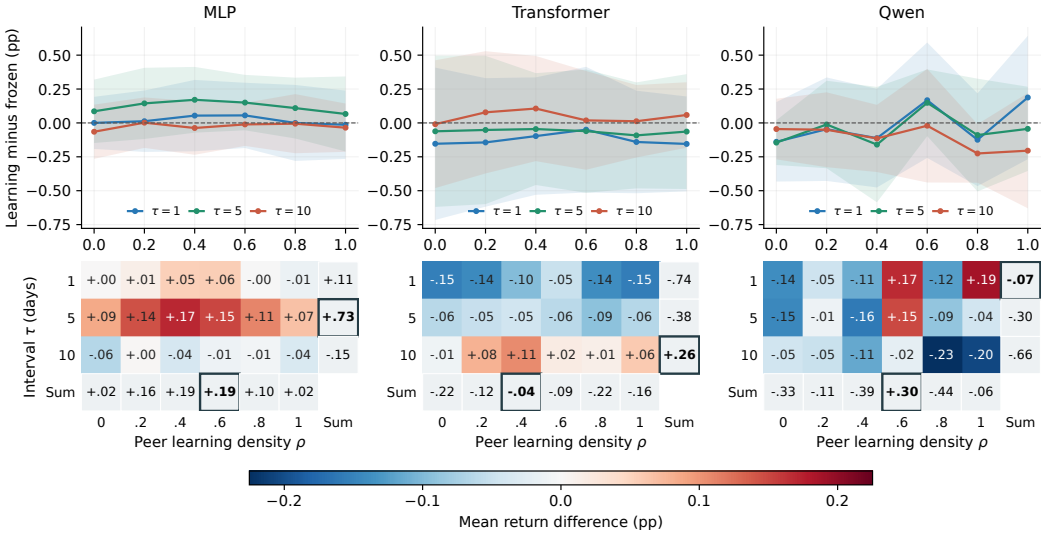


Figure 5: How many peers learn, and how often do learners update together? We vary peer density ρ and the shared interval $\tau_f = \tau_o = \tau$. Day-120 focal learning minus frozen return at matched peer settings (pp). Curves show five-seed means ± 1 SD; heatmaps show the means. Margins sum rows and columns; outlines mark the largest sums. Positive gains need not imply positive returns.

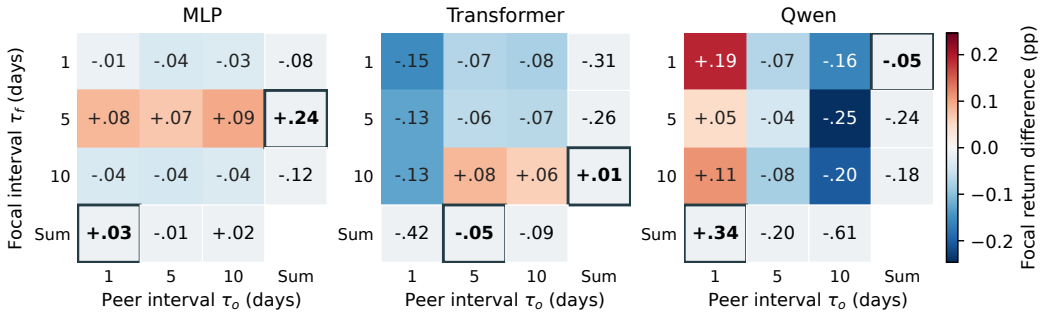


Figure 6: Does updating faster than peers help? All peers learn ($\rho = 1$), while focal and peer intervals vary independently. Rows: τ_f; columns: τ_o (days). The diagonal matches the $\rho = 1$ conditions in Figure 5. Cells show day-120 five-seed mean focal learning minus frozen return at the same peer interval (pp), with a shared color scale. Margins sum rows and columns; outlines mark the largest sums.

Partial participation can favor learning gains. The average gain at $\rho \in \{.4, .6\}$ exceeds that at $\rho \in \{0, 1\}$ in all nine model–interval means. At $\tau = 5$, the difference is 0.084 pp for MLP (five positive seeds) and 0.089 pp for Qwen (four). An intermediate interval is less general: the five-day gain exceeds the average one- and ten-day gain for MLP and Qwen, but not Transformer. Thus partial participation is more consistently favorable in these mean contrasts than any single update interval is across models. A larger gain can still mean a smaller learning disadvantage, not a positive return. These post-hoc grid comparisons support a non-monotone view of population learning, not a universal intermediate optimum.

Does learning faster than one's peers help? We separate focal and peer intervals, $\tau_f, \tau_o \in \{1, 5, 10\}$, at $\rho = 1$. Each cell retains a focal-frozen comparator with the same peer interval and seed; optimizer learning rates are unchanged (Appendix I).

Figure 6 shows no common faster-is-better pattern. For Qwen with daily focal updates, changing peers from daily to ten-day updates changes the mean learning gain from $+0.188$ to -0.163 pp (seed SDs 0.455 and 0.653; four and two positive seeds). Its frozen baseline changes from -0.400% to -0.023%, while both learning arms remain negative. Thus peers' update speed changes the value

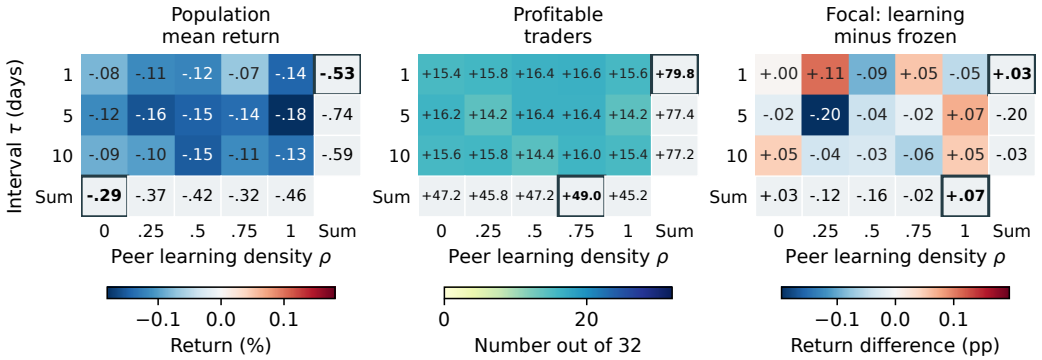


Figure 7: Approximately uniform model mixture, day 120; five-seed means. Left: research-trader mean return. Center: profitable research-trader count. Both enable focal learning. Right: focal learning minus matched frozen return (pp). Margins sum rows and columns; outlines mark the largest sums.

of focal learning, not merely the focal cohort's absolute return. Timing and optimization effort still vary together; this is not an equal-budget comparison.

Peer updates can change orders, realized market paths, and the data available for subsequent fitting. The comparisons capture their combined influence; they do not identify which channel explains the return changes. Historical improvement in Section 4.1 is therefore a starting point, not a guarantee of an advantage in the interacting market. **The value of continued learning depends not only on the learner, but also on how other participants learn.**

5.3 Collective and focal benefits need not coincide

We extend the single-family experiments to ask whether collective and focal benefits favor the same conditions when different learners trade together. We distribute traders approximately uniformly across MLP, Transformer, and Qwen, retaining the market and update rules. Appendix G specifies the composition and assignment protocol. Figure 7 separates the number of winners from the size of collective gains. After averaging over intervals, profitable-trader count peaks at $\rho = .75$, whereas mean return peaks at $\rho = 0$. At the cell level, the largest focal learning gain and the strongest collective outcomes also occur at different densities. These are distinct summaries of the grid, not a shared optimal setting across seeds.

The measures capture different aspects of the return distribution: profitable count records how many traders cross zero, whereas mean return also reflects the sizes of gains and losses. Focal learning gain asks a further question, comparing the focal cohort against its own frozen alternative. The observed separation does not itself identify a transfer of wealth from peers to the focal cohort. These patterns do not establish a benefit from mixing: seeds differ substantially, and identity assignment differs from the single-family design. Appendix G gives seed-level results and exact contrasts. **Conditions that favor collective outcomes need not maximize the gain from the focal cohort's continued learning.**

6 Conclusion

We introduced FML to study traders who learn through market interaction. Across three learner families, continued updates produce behavioral differentiation, but more widespread or frequent learning does not uniformly improve returns. The benefits depend on the learner and its learning peers; conditions favoring collective outcomes need not maximize focal learning gains. These findings support evaluating continued learning as a population process, rather than an isolated capability. Our evidence concerns a finite set of simulated markets, not a universal optimal schedule or a claim of real-market profitability. The experiments do not isolate execution, price changes, and learning-data feedback as explanations of the gains.

AI USE STATEMENT

Generative AI assisted with writing and polishing, and with retrieval and discovery of related literature. The authors checked cited sources and reviewed the resulting text. AI tools also assisted with analysis and figure code.

REFERENCES

Selim Amrouni, Aymeric Moulin, Jared Vann, Svitlana Vyetrenko, Tucker Balch, and Manuela Veloso. ABIDES-Gym: gym environments for multi-agent discrete event simulation and application to financial markets. In *Proceedings of the Second ACM International Conference on AI in Finance*, pp. 1–9, 2021.

Peter Belcak, Jan-Peter Calliess, and Stefan Zohren. Fast agent-based simulation framework with applications to reinforcement learning and the study of trading latency effects. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pp. 42–56. Springer, 2021.

David Byrd, Maria Hybinette, and Tucker Hybinette Balch. ABIDES: Towards high-fidelity multi-agent market simulation. In *Proceedings of the 2020 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, pp. 11–22, 2020.

Jiaxiang Chen, Mingxi Zou, Zhuo Wang, Qifan Wang, Dongning Sun, Chi Zhang, and Zenglin Xu. Finhear: Human expertise and adaptive risk-aware temporal reasoning for financial decision-making. *arXiv preprint arXiv:2506.09080*, 2025.

Sascha Yves Frey, Kang Li, Peer Nagy, Silvia Sapora, Christopher Lu, Stefan Zohren, Jakob Foerster, and Anisoara Calinescu. JAX-LOB: A gpu-accelerated limit order book simulator to unlock large scale reinforcement learning for trading. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 583–591, 2023.

Josef Hofbauer and Karl Sigmund. *Evolutionary games and population dynamics*. Cambridge university press, 1998.

Cars H Hommes. Heterogeneous agent models in economics and finance. *Handbook of computational economics*, 2:1109–1186, 2006.

Allen H Huang, Hui Wang, and Yi Yang. FinBERT: A large language model for extracting information from financial text. *Contemporary Accounting Research*, 40(2):806–841, 2023.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Blake LeBaron. Agent-based computational finance. *Handbook of computational economics*, 2: 1187–1233, 2006.

Jean Lee, Nicholas Stevens, Soyeon Caren Han, and Minseok Song. A survey of large language models in finance (finllms). *arXiv preprint arXiv:2402.02315*, 2024.

Joel Z Leibo, Edward Hughes, Marc Lanctot, and Thore Graepel. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *arXiv preprint arXiv:1903.00742*, 2019.

Xiao-Yang Liu, Guoxuan Wang, Hongyang Yang, and Daochen Zha. FinGPT: Democratizing internet-scale data for financial large language models. *arXiv preprint arXiv:2307.10485*, 2023.

Thomas Lux and Michele Marchesi. Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature*, 397(6719):498–500, 1999.

Shie Mannor and Jeff S Shamma. Multi-agent learning for engineers. *Artificial Intelligence*, 171(7): 417–422, 2007.

Chris Mascioli, Anri Gu, Yongzhao Wang, Mithun Chakraborty, and Michael Wellman. A financial market simulation environment for trading agents using deep reinforcement learning. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pp. 117–125, 2024.

Celestine Mendler-Dünner, Juan Perdomo, Tijana Zrnic, and Moritz Hardt. Stochastic optimization for performative prediction. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4929–4939, 2020. URL https://arxiv.org/abs/2006.06887.

Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M Mulvey, H Vincent Poor, Qingsong Wen, and Stefan Zohren. A survey of large language models for financial applications: Progress, prospects and challenges. *arXiv preprint arXiv:2406.11903*, 2024.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7599–7609, 2020. URL https://proceedings.mlr.press/v119/perdomo20a.html.

William H Sandholm. *Population games and evolutionary dynamics*. MIT press, 2010.

Karl Tuyls, Pieter Jan 't Hoen, and Bram Vanschoenwinkel. An evolutionary dynamical analysis of multi-agent learning in iterated games. *Autonomous Agents and Multi-Agent Systems*, 12(1): 115–153, 2006.

Alicia Vidler and Toby Walsh. TraderTalk: An LLM behavioural abm applied to simulating human bilateral trading interactions. In *2024 IEEE International Conference on Agents (ICA)*, pp. 164–167. IEEE, 2024.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. TradingAgents: Multi-agents LLM financial trading framework. *arXiv preprint arXiv:2412.20138*, 2024.

Yuzhe Yang, Yifei Zhang, Minghao Wu, Kaidi Zhang, Yunmiao Zhang, Honghai Yu, Yan Hu, and Benyou Wang. TwinMarket: A scalable behavioral and social simulation for financial markets. *arXiv preprint arXiv:2502.01506*, 2025. URL https://arxiv.org/abs/2502.01506.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, et al. FinCon: A synthesized LLM multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems*, 37:137010–137045, 2024.

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Jordan W Suchow, Denghui Zhang, and Khaldoun Khashanah. FinMem: A performance-enhanced LLM trading agent with layered memory and character design. *IEEE Transactions on Big Data*, 2025.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang, and Bo An. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. *arXiv preprint arXiv:2402.18485*, 2024. URL https://arxiv.org/abs/2402.18485.

Mingxi Zou, Jiaxiang Chen, Aotian Luo, Jingyi Dai, Chi Zhang, Dongning Sun, and Zenglin Xu. Finevo: From isolated backtests to ecological market games for multi-agent financial strategy evolution. *arXiv preprint arXiv:2602.00948*, 2026.

A INDEPENDENT SAMPLING AND POPULATION DIFFERENTIATION

Proposition 1 characterizes parameter dispersion through conditional variance decomposition, not learning returns.

A.1 PROOF OF PROPOSITION 1

Let $\mathcal{H}_k$ include the joint market history, available training pool, pre-update parameters, and optimizer states. Conditional on this history, the pool is fixed throughout the update event and the update map is deterministic given each trader's independent random draws. Thus its outputs are conditionally independent even though the traders share a market trajectory. Any additional optimizer randomness must also be independent across traders. All second moments below are assumed finite.

Proof. Suppress the event index and write θ_i^+ for the post-update parameters. The sum-of-squares identity gives

$$D_\theta^+ = \frac{1}{n}\sum_i \|\theta_i^+\|^2 - \|\bar{\theta}^+\|^2.$$

By conditional independence, $\mathrm{Cov}(\bar{\theta}^+ \mid \mathcal{H}_k) = n^{-2}\sum_i V_i$. Using $\mathbb{E}\|X\|^2 = \|\mathbb{E}X\|^2 + \operatorname{tr}\mathrm{Cov}(X)$ on both terms yields

$$\begin{aligned}\mathbb{E}[D_\theta^+ \mid \mathcal{H}_k] &= \frac{1}{n}\sum_i (\|\mu_i\|^2 + \operatorname{tr} V_i) - \|\bar{\mu}\|^2 - \frac{1}{n^2}\sum_i \operatorname{tr} V_i \\ &= \frac{1}{n}\sum_i \|\mu_i - \bar{\mu}\|^2 + \frac{n-1}{n^2}\sum_i \operatorname{tr} V_i.\end{aligned}$$

At a shared learning state with identical update laws, $\mu_i = \mu$ and $V_i = V_k$ for every trader. The first term vanishes, and the second becomes $(1 - 1/n)\operatorname{tr} V_k$. Since pre-update dispersion is zero, positive trace gives a strict increase in expected dispersion. □

Connection to minibatch gradients: sampling with replacement. At a common parameter θ, let $g(z) = \nabla_\theta \ell(\theta; z)$ for $z \sim \mathrm{Unif}(\mathcal{D})$, with covariance Σ. One SGD step uses $\theta_i^+ = \theta - \eta\bar{g}_i$, where $\bar{g}_i = b^{-1}\sum_{s=1}^b g(z_{i,s})$. Independent sampling with replacement gives $\mathrm{Cov}(\bar{g}_i) = \Sigma/b$, and hence

$$\mathbb{E}D_\theta^+ = \left(1 - \frac{1}{n}\right)\frac{\eta^2}{b}\operatorname{tr}\Sigma.$$

Thus gradient variability, not merely different sample indices, is the relevant source of separation. Our experiments use multi-step AdamW updates; Proposition 1 applies to their complete output covariance, not the single-step SGD expression.

What follows over successive events. By the tower property of conditional expectation,

$$\mathbb{E}D_{\theta,k+1} \geq \frac{n-1}{n^2}\sum_i \mathbb{E}[\operatorname{tr} V_{i,k}].$$

Nonzero update variance supplies variation at later events, but mean updates may contract existing differences; dispersion need not increase monotonically.

Why behavior must also be measured. Different batches can yield identical gradients, and softmax removes common score offsets. Portfolio caps, rounding, and unfilled orders can further suppress differences, so parameter dispersion does not guarantee behavioral diversity.

A.2 CONNECTING PARAMETER AND BEHAVIORAL DISPERSION

Prediction on a common input. For a deterministic readout $z_i^+ = \phi(\theta_i^+)$ on fixed common inputs, conditional independence is preserved. The same proof therefore gives (5) with the means and covariances of z_i^+ in place of those of the parameters. From a shared start, prediction dispersion is positive in expectation if $\operatorname{tr}\mathrm{Cov}(z_i^+ \mid \mathcal{H}_k) > 0$. Nonzero parameter covariance alone does not establish this condition. This readout argument also does not presume independent realized fills after market interaction.

Similarity as dispersion. For a given similarity, retain its $n \geq 2$ valid behavior vectors and normalize them to unit vectors v_i. For score correlation, subtract each score vector's stock-wise mean before normalization. For target and order cosines, normalize without centering. Thus

$$S = \frac{1}{n(n-1)} \sum_{i \neq j} v_i^\top v_j, \qquad D_v = \frac{1}{n} \sum_i \|v_i - \bar{v}\|^2 = 1 - \|\bar{v}\|^2.$$

Expanding the pairwise sum yields

$$S = \frac{\|\sum_i v_i\|^2 - n}{n(n-1)} = \frac{n\|\bar{v}\|^2 - 1}{n-1}, \qquad 1 - S = \frac{n}{n-1} D_v.$$

This is an identity for realized vectors and requires no trader independence.

Within- and between-group contributions. Partition valid vectors into nonempty learning and frozen groups, with learning fraction $\lambda = n_L/n$, group means μ_L, μ_F, and within-group mean squared dispersions D_L, D_F. Applying the law of total variance to a uniformly selected trader gives

$$D_v = \lambda D_L + (1-\lambda) D_F + \lambda(1-\lambda)\|\mu_L - \mu_F\|^2.$$

Increasing the learner fraction can therefore raise or lower total dispersion; both within-group variation and between-group separation matter. The group distributions themselves change with participation and frequency, so this identity does not predict an optimal density.

The identity holds separately at every observation time. If one group is empty, the between-group term is absent and $1 - S = nD_g/(n-1)$ for the remaining group. Fewer than two valid vectors leaves S undefined. Since λ includes focal traders and excludes invalid vectors, it need not equal peer density ρ. These are decompositions of measured behavior, not causal explanations of return or proofs that diversity increases with time.

B Matched Population Similarity Comparisons

τ (days)	Return gain (pp)	Seed SD (pp)	Order-similarity change
1	−0.011	0.297	−0.056
5	−0.050	0.165	−0.061
10	−0.110	0.189	−0.132

Table 2: Qwen under long-only exposure at day 120. Focal learning minus matched focal frozen, averaged over densities within each seed and then over five seeds. SD describes between-seed variation, not variation across traders.

Focal learning can change observations, fills, and parameters; this comparison does not isolate sampling or identify a causal mediator of returns.

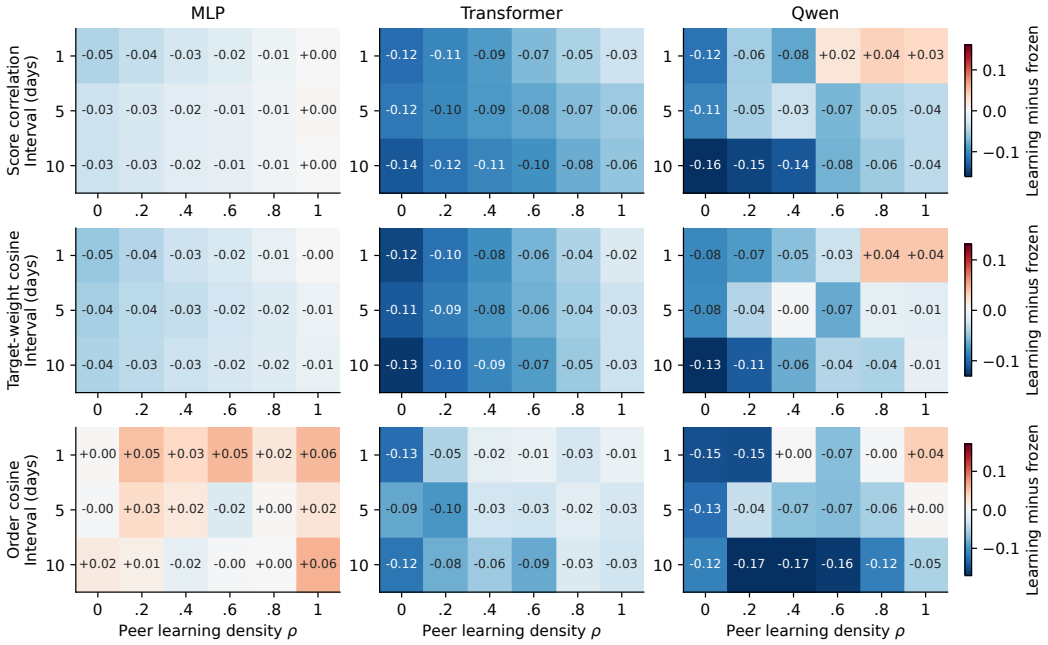


Figure 8: Focal cohort learning minus focal cohort frozen, at the same peer density and interval. Negative cells indicate less similar behavior after enabling focal updates; positive cells indicate more similarity. Each row has a common scale across model families. Unlike the dashed all-frozen reference in Figure 4, these cells hold the peer-learning assignment fixed when changing focal updates.

C Historical Execution Sensitivity

Model	Long-only (40/0) Frozen	Long-only (40/0) Online	Long-biased (30/10) Frozen	Long-biased (30/10) Online	Balanced (20/20) Frozen	Balanced (20/20) Online
MLP	−3.12	6.61	−5.10	2.19	−13.88	−10.20
Transformer	3.77	15.91	−0.07	7.15	−14.88	−12.03
Qwen	1.79	5.38	−1.00	0.81	−9.87	−8.42

Table 3: Continued learning on a fixed historical path. Cumulative net returns (%; higher is better) include final liquidation. The reference uses 5 bps fees, a 3 bps per-side execution offset, and a 10% daily traded-notional cap. Each frozen/online pair shares its initial fitted model. These are exploratory comparisons on one historical path, not nine independent replications.

We retain three investment mandates: long-only (40/0), long-biased (30/10), and balanced long–short (20/20). For each model and mandate, we replay the same stored causal frozen and online predictions under five fee/offset combinations, $(5, 3), (5, 0), (2, 3), (2, 1), (0, 0)$ bps, and daily executed-notional caps of 2%, 5%, and 10% of equity. Each setting recomputes cash, holdings, fees, and execution costs rather than subtracting an estimated cost from an old return. Changing costs can therefore change subsequent holdings. The zero-cost setting is diagnostic, not a proposed implementable venue.

The initial capital is 100,000. Positions persist, and the last test observation is followed by cost-bearing liquidation; terminal liquidation is exempt from the daily turnover cap. Historical returns in the first experiment therefore include this closing cost, unlike the marked day-120 FML outcomes. Borrowing availability is idealized and interest and borrow fees are zero. Existing survivorship concerns and an anomalous opening-price observation in HUBB also limit the historical evidence; execution sensitivity is not a substitute for an independently curated dataset. These limitations preclude a claim of real-market profitability or universally effective online learning.

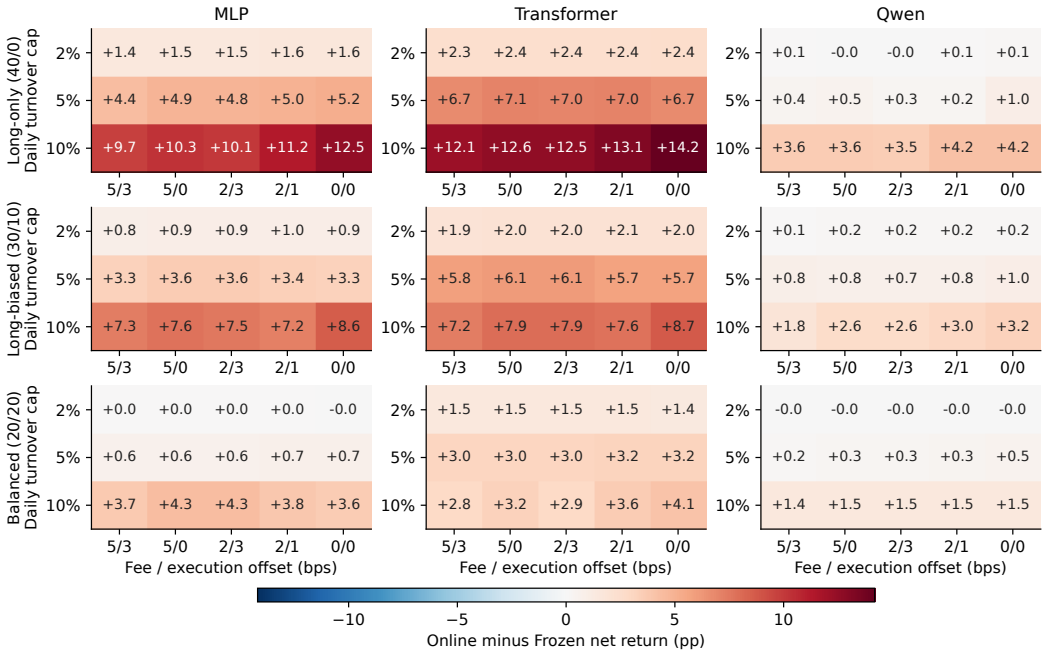


Figure 9: Historical execution sensitivity: online minus frozen cumulative net return in percentage points. Each panel spans three turnover caps and five fee/offset pairs. Online improves 127 of the 135 displayed pairs, but these are correlated variations on the same historical path, not 135 independent replications. Color scales are shared across panels.

D LONG-BIASED POPULATION LEARNING

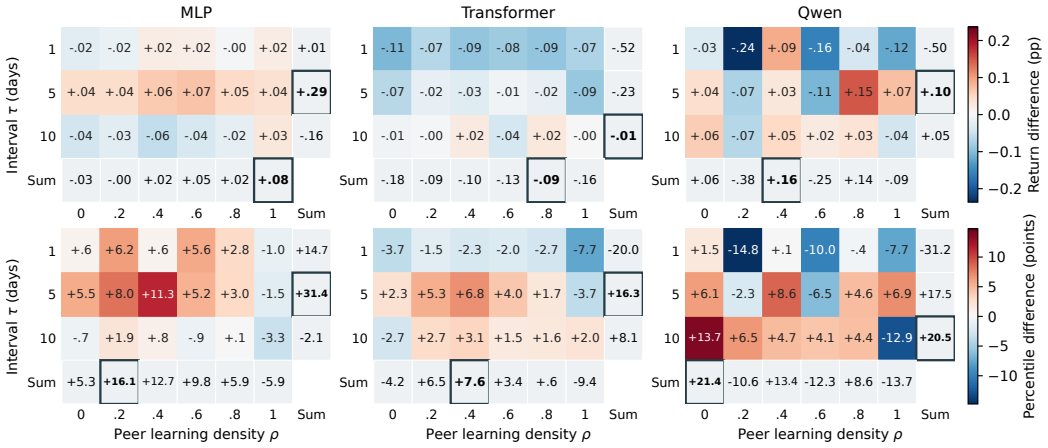


Figure 10: Long-biased (30% long, 10% short) focal outcomes at day 120. Focal learning minus frozen at matched density, interval, and seed; five-seed means. Top: return change (pp). Bottom: rank-percentile change (points). Each row shares a color scale across models. Margins sum rows and columns; outlines mark maximal sums, not established optimal settings.

This comparison repeats the six densities, three intervals, two focal learning states, three model families, and five market seeds under 30% long and 10% short target exposure. All 540 paths reach day 120. It retains the initial models, market-seed assignment, update rule, fees, and background participants of the long-only comparison; realized prices, fills, and data pools may differ after orders change. Matching across mandates is therefore not a comparison on an identical realized price path.

Absolute returns, ranks, and learning gains answer different questions. At $\tau = 5$, MLP has a positive return gain with both frozen peers ($B - A = 0.041$ points) and learning peers ($D - C = 0.036$), but

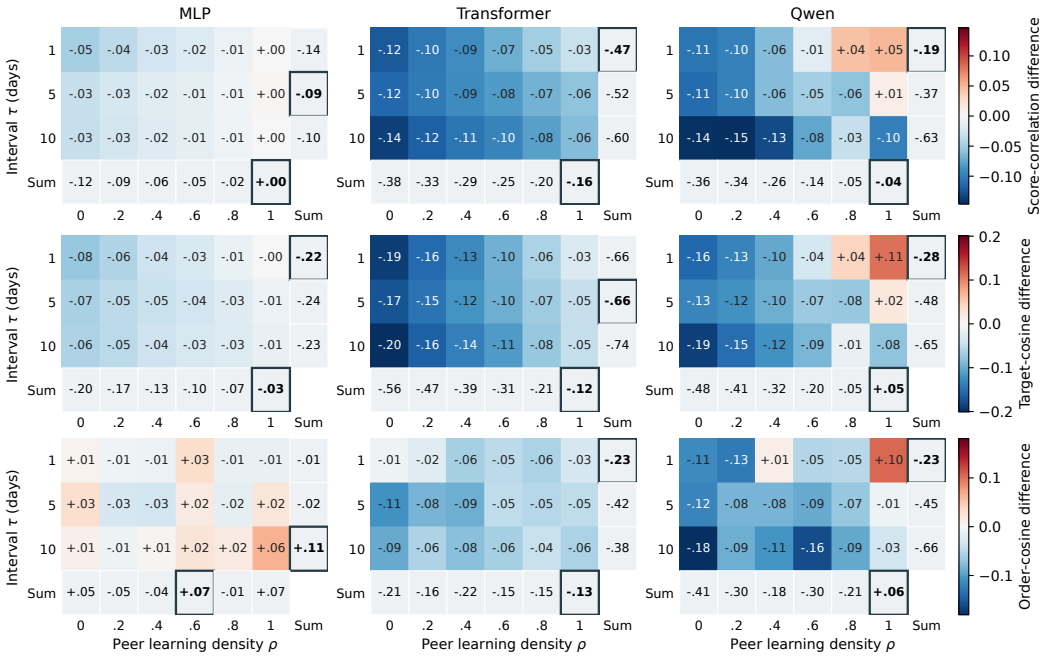


Figure 11: Long-biased population structure: focal learning minus focal frozen at fixed peer density and update interval, averaged over five matched seeds. Rows distinguish score correlation, target-weight cosine, and order cosine, each with a common scale across models. Negative cells mean greater directional disagreement. Margins sum the displayed grid means; outlined bold entries mark the largest signed row and column sums, not the greatest differentiation.

Model	Focal outcome	A	B		C		D	
		Value	Value	ΔA	Value	ΔA	Value	ΔA
MLP	Return (%) ↑	−0.340	−0.299	+0.041	−0.315	+0.025	−0.280	+0.061
	Rank pct. ↑	28.55	34.03	+5.48	38.39	+9.84	36.85	+8.31
Transformer	Return (%) ↑	−0.066	−0.132	−0.065	−0.104	−0.038	−0.193	−0.127
	Rank pct. ↑	39.68	41.94	+2.26	50.16	+10.48	46.45	+6.77
Qwen	Return (%) ↑	−0.367	−0.328	+0.039	−0.439	−0.072	−0.367	−0.000
	Rank pct. ↑	33.95	40.08	+6.13	36.37	+2.42	43.23	+9.27

Table 4: Long-biased ABCD outcomes at $\tau = 5$, day 120. Entries are five-seed means; each ΔA column subtracts A within each seed before averaging. Rank pct. denotes percentile. Higher return and higher percentile are preferable. Return changes are percentage points; percentile changes are points on the 0–100 scale. Changes use unrounded data, so a displayed near-zero change need not be exactly zero.

its rank gain switches from $+5.48$ to -1.53 percentile points. Transformer has negative return gains in both cases (-0.065 and -0.089), whereas Qwen has positive gains ($+0.039$ and $+0.072$). All four mean absolute focal returns remain negative for each model at this interval. Neither a positive learning contrast nor a higher relative rank is synonymous with making money.

Model	Return difference (pp) Mean ± SD	Positive seeds	Percentile difference (points) Mean ± SD	Positive seeds
MLP	$+0.008 \pm 0.116$	2/5	$+2.44 \pm 14.09$	3/5
Transformer	-0.042 ± 0.295	1/5	$+0.25 \pm 25.39$	1/5
Qwen	-0.020 ± 0.124	2/5	$+0.38 \pm 13.24$	2/5

Table 5: Long-biased grid summaries. For each seed, focal-learning minus focal-frozen differences are averaged over the 18 density–interval cells. The table then summarizes these five seed-level averages. SD is the sample standard deviation across seeds, not across the 18 cells or 32 traders.

E Experimental Protocol and FML Implementation

We specify the grid, training pairs, models, and execution chronology; Algorithm 1 summarizes a paired experimental path.

E.1 Population protocol

Population learning grid. The main density–frequency comparison uses $N = 32$ research traders and a fixed focal cohort of $r = 8$. Among the remaining 24 peers, $k \in \{0, 5, 10, 14, 19, 24\}$ update; figures label these settings nominally as $\rho \in \{0, .2, .4, .6, .8, 1\}$, whereas the realized fractions are $k/24$. Update intervals are $\tau \in \{1, 5, 10\}$ decision days, corresponding to $\nu \in \{1, .2, .1\}$ updates per day. Enabled focal and peer learners use the same interval in this grid. Each event comprises five independent batches of 256 examples per learner; more frequent updates therefore use more training batches over a fixed horizon. This is not an equal-compute comparison.

E.2 From stock histories to training pairs

Each observation is a stock–date pair: one shared scorer maps a $J \times 32$ feature matrix to J stock scores.

Features and scaling. Let $P_{j,t}$ be the available historical close or the observed FML midquote, and $u_{j,t} = P_{j,t}/P_{j,t-1} - 1$. The 32 features use at most 41 prices:

Group	Raw feature at window h	Windows h
Momentum	$P_{j,t}/P_{j,t-h} - 1$	1,2,3,5,10,15,20,40
Reversal	$-(P_{j,t}/\overline{P}_{j,t,h} - 1)$	2,3,5,8,10,15,20,40
Low volatility	$-\operatorname{sd}(u_{j,t-h+1:t})$	2,3,5,8,10,15,20,40
Low downside	$-\sqrt{h^{-1}\sum_{k=0}^{h-1}\min(u_{j,t-k}, 0)^2}$	2,3,5,8,10,15,20,40

Here $\overline{P}_{j,t,h}$ is the mean of the last h prices; standard deviations use population normalization. At each date, each feature is centered and scaled across eligible stocks (scale floor 10^{-12}), then clipped to $[-5, 5]$. A second feature-wise scaler, with standard-deviation floor 10^{-8}, is fitted to the initial historical examples and held fixed thereafter. Thus current cross-sectional normalization changes with observations, but its training-set scaler is not refitted on future data. FML excludes stocks lacking valid two-sided quotes anywhere in the feature window; fewer than two eligible stocks produces no new valid examples.

Labels and availability. Historical signal features use the close at t, and the supervised response is the next open-to-open return relative to the stock universe:

$$y_{j,t} = 100\left(\frac{O_{j,t+2}}{O_{j,t+1}} - 1 - \frac{1}{|\mathcal{V}_t|}\sum_{k\in\mathcal{V}_t}\left(\frac{O_{k,t+2}}{O_{k,t+1}} - 1\right)\right). \tag{8}$$

The factor 100 expresses the target in percentage points. FML replaces the historical opens by public midquotes at the corresponding two future decision observations. Its set $\mathcal{V}_t$ requires valid features at t and valid quotes at both $t + 1$ and $t + 2$; the relative-return mean is computed over that same set. At t we save $x_{j,t}$, at $t + 1$ we save the entry reference, and only at $t + 2$ may the pair $(x_{j,t}, y_{j,t})$

enter training. No price imputation creates a label for a missing quote. Labels are public relative price movements, not an agent's realized profit, and contain neither fees nor execution slippage.

The initial pool contains 20,100 examples with labels matured before 2017 (Table 6). Historical evaluation follows initial fitting without a validation split. FML imports only this pool and fitted model, not test-period labels or final online weights. Traders append mature market observations and retain independent sampling, parameters, and optimizer states, including when frozen Qwen representations share storage.

Stage	Available examples and use
Initial fitting	20,100 historical stock–date pairs: 201 signal dates and 100 stocks, with labels matured before 2017. The input scaler is fitted here.
Historical evaluation	Current features produce predictions on the 2017–2026 path. Only the online branch updates, using available matured labels; the frozen branch still recomputes predictions from new observations.
Interactive deployment	The original historical pool plus M_t valid matured FML examples. Future labels and historical test-period labels are excluded.
One update event	Five batches of 256 draws per enabled trader, independently sampled with replacement from the available pool. There are 1,280 draws, not necessarily 1,280 distinct examples.

Table 6: Training examples and their availability. M_t counts eligible interactive stock–date pairs, not traders or optimization steps.

Sampling is uniform over examples in the combined pool. The expected fraction of historical draws is $20{,}100/(20{,}100 + M_t)$; there is no fixed historical mixture weight. Repeated indices within a batch and overlap between traders' batches are permitted. Independence means separate random draws, not disjoint datasets or guaranteed distinct gradients. Features and labels of an example remain fixed once that example enters the pool.

E.3 Scorers and optimization

Family	Scorer and online trainable components
MLP	$32 \to 64 \to 32 \to 1$, GELU hidden activations; all scorer parameters update.
Transformer	Each scalar feature is projected $1 \to 32$ and added to a learned feature-identity embedding. The 32 tokens pass through two encoder layers, four attention heads, feed-forward width 64 with ReLU, zero dropout, and post-layer normalization. Token mean pooling, LayerNorm, and a $32 \to 1$ linear map produce the score. All scorer parameters update.
Qwen	Qwen3.5-2B with an adapter fitted before deployment. Standardized features are serialized as one Markdown table per stock. A 2048-dimensional representation feeds LayerNorm and a $2048 \to 64 \to 32 \to 1$ GELU head. Online updates change each trader's independent head only; backbone and adapter remain frozen.

Table 7: All families output a numeric stock score. They are not matched in capacity or pretraining, and Qwen does not generate textual trading instructions.

MLP uses no dropout; Transformer attention operates across one stock's factors. For Qwen, factor names and standardized values (five decimal places) form a Markdown table, following the instruction to predict next-session relative stock return. A final "Stock representation:" marker supplies the last non-padding hidden state to the numeric head. The prompt contains no ticker, date, future label, or textual answer to generate. Softmax is applied only when allocating stock weights; the supervised score remains an unconstrained scalar and is not an up-move probability.

Initial MLP and Transformer fitting uses 50 epochs, batch size 256, and AdamW at 3×10^{-4}. Qwen initial fitting uses three epochs with rank-8 LoRA on the query and value projections, scaling 16 and zero adapter dropout. Its adapter learning rate is 2×10^{-5} and head rate 3×10^{-4}; microbatches of four accumulate to an effective batch of 32, with the final partial batch weighted by its actual size. All initial losses are mean squared error on the numeric label. These architecture-specific budgets are not an equal-compute comparison. After Qwen initial fitting, both backbone and adapter are frozen; subsequent sampling and updates concern the head alone.

Each active trader uses AdamW with learning rate 3×10^{-4}, weight decay 10^{-5}, and gradient-norm clipping at 5. Initial optimizer states are identical. Each scheduled event uses the sampling scheme in Table 6. The target is relative return in percentage points, converted to basis points for allocation. Softmax temperature is 100 bps, and normalization statistics remain fixed from initial fitting. FML initializes a fresh optimizer per trader from the common fitted weights. In the historical comparison, MLP and Transformer continue their initial optimizer state, whereas Qwen starts a fresh head-only optimizer after adapter fitting. Within each setting the frozen and online branches share the same initial weights. There is no additional noise on the trading scores.

E.4 Portfolios and market timing

Historical and interactive portfolios. The historical comparison includes 40/0 long-only, 30/10 long-biased, and 20/20 balanced target exposures. FML's main comparison uses long-only exposure capped at 40%, a 5% single-stock cap, and up to ten selected stocks out of twenty. The daily submission budget is 10% of current equity. Small adjustments below 0.5% of equity are suppressed except for required risk reduction. Cash limits, share rounding, and missing fills can leave actual exposure below its target. Historical short execution is idealized and is not a claim about borrowing availability in an actual market. The long-biased FML comparison assigns up to ten top-scoring eligible stocks to the long side and up to ten disjoint bottom-scoring stocks to the short side. Softmax of scores and of negated scores distributes the 30% and 10% budgets, respectively, with a 5% absolute per-stock cap. Risk reduction precedes opening new exposure; short liabilities require 150% cash reservation. These are target budgets, not guaranteed realized exposures.

Decision-day chronology. At day t, traders observe valid public quotes and update their price-derived features. A saved day-$t-2$ example can now receive its relative return label, using observations from $t-1$ and t. Only matured examples enter a scheduled update. The scorer then produces a target for $t+1$, while today's orders adjust positions toward the previously stored target. Risk reduction precedes new exposure, and anticipated sale proceeds are not treated as available cash. Outstanding day orders are canceled at the session end; holdings persist. Missing two-sided quotes are not replaced by invented executable prices: affected positions are held and invalid training labels excluded.

Background trading and price formation. Two market makers quote five levels around latent-value-informed prices. Each stock also has eight retail participants, each initialized with 25,000 currency units, random order direction, five-share orders, and a 100-share position limit; half use passive orders. Research and retail traders pay 5 bps on actual fills; makers pay zero without rebates. Latent values evolve exogenously, but transaction prices and the observed book also depend on participant orders. This is not replay of the historical test-price path. The event scheduler's short session is a simulation device, not calibrated minute-frequency data. Matching priority still matters: simultaneous participation does not imply identity-invariant fills. We do not establish this with a permutation test.

E.5 Core interactive learning loop

Observation, submission, and fill events follow the venue scheduler. The algorithm separates model updates from execution toward stored target shares; new predictions do not trigger liquidation.

The first decision has no stored signal and opens no new positions. Frozen traders skip only parameter updates. Paired runs share initial models, accounts, focal identities, and external random streams; changing learning masks or intervals can subsequently produce different prices and training pools.

Algorithm 1 FML: interaction with independently sampled updates

Require: Historical pool $\mathcal{D}_{\text{hist}}$, fitted scorers and scaler, market seed ω, masks u_i, intervals τ_i, horizon T

Initialize the venue, background participants, and trader accounts
Copy each family's fitted parameters to its traders; initialize independent optimizers and RNGs
Initialize pending examples and mature buffers; set stored targets to initial holdings
for $t = 0, \ldots, T-1$ **do**
    Start the session with its exogenous inputs and carried cash and holdings
    while scheduled session events remain **do**
        Process the next event in venue order
        if trader i receives its decision observation **then**
            Observe quotes; retain validity masks and price history
            Save the entry reference for pending day-$t-1$ examples
            Mature valid day-$t-2$ labels using Eq. (8); append to the data pool
            if $u_i = 1$ and $t \bmod \tau_i = 0$ **then**
                for $l = 1, \ldots, 5$ **do**
                    Draw 256 examples independently with replacement using trader i's RNG
                    Update θ_i by AdamW on prediction MSE, with gradient clipping
                end for
            end if
            Construct current features and predict stock scores with θ_i
            Save features and predictions as pending examples; compute the next target
            Request orders toward the previously stored target, then store the new target
        else if an execution or order-status event arrives **then**
            Settle actual fills and fees; release reservations as applicable
            Advance risk reduction before new exposure, subject to cash and submission limits
        else
            Process background orders, quotes, and other venue events
        end if
    end while
    Cancel outstanding day orders; retain holdings, cash, model states, and data
    Record returns, ranks, similarities, quotes, submissions, fills, and learning diagnostics
end for

E.6 SELECTED IMPLEMENTATION EXCERPTS

These verbatim excerpts from the density–frequency runner and inherited market agent retain caching and diagnostics; they are not standalone programs.

Labels become available before sampling. Listing 1 stores the next observation as the entry reference and waits one further observation before adding a labeled example. The validity filter excludes missing features or quotes; no future label is available at the signal date.

```
            if day == item["day"] + 1:
                item.update(entry=v11.v9.intern(mid.copy()), entry_valid=
    valid.copy())
            if day == item["day"] + 2:
                keep, y = v11.matured_target(item, mid, valid)
                self.discarded_labels += len(mid) - len(y)
                if len(y):
                    x = v11.v9.intern(item["x"][keep].copy())
                    y = v11.v9.intern(y)
                    self.buffer.append((x, y))
                    self.matured_count += len(y)
                    pred = item["prediction_percent"][keep]
```

Listing 1: Delayed label construction in the population runner.

Independent draws, independent optimization. The enclosing condition is `learn and day % self.update_interval == 0`. In Listing 2, `xx` and `yy` contain historical and matured interactive examples. `self.rng.integers` samples with replacement using the trader's own random stream. Each `self.model` and optimizer is independent; reusing the data tensor does not share updates. The recorded sample hashes audit the draws without changing them.

```
            self.model.train()
            for _ in range(5):
                indices = self.rng.integers(0, len(yy), 256)
                self.sample_hashes.append(hashlib.sha256(indices.tobytes
    ()).hexdigest())
                self.historical_draws += int((indices < n0).sum())
                self.market_draws += int((indices >= n0).sum())
                ix = torch.as_tensor(indices, device="cuda")
                self.optimizer.zero_grad()
                loss = ((self.model(xx[ix]) - yy[ix]) ** 2).mean()
                assert torch.isfinite(loss)
                loss.backward()
                torch.nn.utils.clip_grad_norm_(self.model.parameters(),
    5.0)
                self.optimizer.step()
                self.parameter_steps += 1
```

Listing 2: Five independently sampled minibatch updates per event.

A new prediction is not an immediate position. Listing 3 computes a new plan but chooses the previously stored target for execution. With no previous target it keeps current holdings; with an invalid quote it holds that stock. The subsequent code stores the new plan for the next decision and submits risk-reducing orders before opening exposure.

```
        scores,eligible=self.controller.act_masked(self.day_index,mid,
    self.history,valid,self.learning_enabled)
        pos=np.array([self.get_holdings(s) for s in self.settings.symbols
    ])
        equity=float(self.holdings['CASH']+pos@mid)
        plan=small_target(self.holdings['CASH'],pos,mid,scores,eligible)
        target=pos.copy() if self.controller.next_target is None else
    self.controller.next_target.copy()
        target[~valid]=pos[~valid]
        self.mid=mid;self.quote_valid=valid
        self.bid=np.array([q[0] if valid[i] else 0 for i,q in enumerate(
    quotes)],dtype=np.int64)
        self.ask=np.array([q[2] if valid[i] else 0 for i,q in enumerate(
    quotes)],dtype=np.int64)
        self.execution_target=target;self.remaining_budget=.1*equity;self.
    initial_budget=self.remaining_budget
```

Listing 3: Separating current scoring from execution of the saved target.

F Measurements and Aggregation

We mark wealth as $W_{i,t} = c_{i,t} + h_{i,t}^{\top} P_t$ and report cumulative net return $R_{i,t} = 100(W_{i,t}/W_{i,0} - 1)$. The focal outcome averages its eight traders. Within-arm percentile for trader i is

$$Q_{i,t} = \frac{100}{N-1} \sum_{j \neq i} \left[\mathbf{1}\{R_{j,t} < R_{i,t}\} + \tfrac{1}{2} \mathbf{1}\{R_{j,t} = R_{i,t}\} \right].$$

Equal stored returns receive midranks. A higher percentile can accompany a negative absolute return and compares peers in the same arm, not a fixed cross-arm reference population. Prediction Rank IC instead compares stock scores with subsequently matured labels; it is not trader wealth rank.

For valid nonconstant stock-score vectors, S_α averages pairwise Pearson correlation. S_w averages cosine similarity of target stock weights, and S_q averages cosine similarity of signed submitted limit-notional vectors. Constant-score and zero-vector pairs are excluded where the corresponding similarity is undefined. Figure 3(b) shows daily observations; the density fields use the terminal day-120 decision snapshot. Neither describes realized holdings. Days without valid order pairs remain missing rather than being assigned zero or perfect similarity. Order cancellation and resubmission count as submission activity, not executed turnover.

Seeds are the replication unit. All three models use seeds 7, 11, 23, 37, and 41. At each grid point, we first subtract matched learning and frozen outcomes within each seed, then compute the mean and sample standard deviation across seeds. Marginal summaries first average the six density points within a seed, retaining seeds as the unit for dispersion. Shaded bands are mean plus or minus one sample standard deviation, not confidence intervals. Different traders, days, or execution configurations on the same historical path do not create independent market replications.

The density labels 0, .2, .4, .6, .8, and 1 correspond to 0, 5, 10, 14, 19, and 24 updating peers. Their realized fractions are those integer counts divided by 24. The total fraction of updating research traders also includes the focal cohort when enabled; it is not equal to the peer density in general.

G Mixed-Model Population Details

The mixed market contains 11 MLP, 11 Transformer, and 10 Qwen traders, the nearest integer allocation to equal shares in a population of 32. Model identities are randomly assigned to trader IDs once per seed. Eight focal IDs are sampled without replacement, without enforcing a balanced model composition. The remaining 24 peers follow a fixed random inclusion order, yielding nested learning sets of size 0, 6, 12, 18, and 24. The assignment, focal cohort, and inclusion order are

identical across all conditions within a seed. With three update intervals and two focal states, this gives 30 paths per seed and 150 paths over five seeds. Market and execution settings match the main experiment in Appendix E.

Historical initialization and update rules are unchanged: MLP and Transformer update independent scorers, while Qwen updates independent heads with a shared frozen encoder and adapter.

Seed	Focal MLP/TF/Qwen	Mean return (%)	Profitable / 32	Learning gain (pp)
7	2/2/4	+0.133	19.07	-0.020
11	3/1/4	-1.088	0.07	+0.075
23	2/4/2	-1.123	0.53	-0.012
37	0/3/5	+0.392	26.67	+0.113
41	4/2/2	+1.069	31.80	-0.226

Table 8: Mixed-population seed variation. Mean return and profitable count describe all 32 research traders with focal learning enabled, averaged over the 15 density–interval settings within each seed. Learning gain subtracts the matched focal-frozen return before averaging. Fractional profitable counts arise from averaging configurations, not fractional traders. TF denotes Transformer. These five markets, not the traders or configurations within them, are the repeated units.

Aggregate patterns and seed variation. Averaging over update intervals and seeds, the profitable-count density marginal peaks at $\rho = .75$ (16.33 of 32), whereas mean return peaks at $\rho = 0$ (-0.095%, compared with -0.107% at $\rho = .75$). At the cell level, both collective metrics peak at $(\rho, \tau) = (.75, 1)$, while the focal learning gain peaks at $(.25, 1)$ ($+0.109$ pp). Grid-averaged profitable counts range from 0.07 to 31.80 across seeds, and grid-averaged focal learning gains are positive in only two seeds. These different aggregations should not be interpreted as the behavior of a single representative market.

At $\rho = .75$, profitable-count differences from the endpoint average are 4.33, 0, -0.50, 0.50, and 0.33 traders. The daily-versus-ten-day difference in density-averaged profitable count is -1.0, 0, -0.6, 4.2, and 0 across seeds. Aggregate maxima are exploratory grid summaries, not common seed-level or out-of-sample optima.

Focal composition varies by seed (Table 8); seed 37 has no MLP focal trader. Without matched identities across pure and mixed markets, their differences cannot be attributed solely to model heterogeneity.

H Full Learning and Differentiation Trajectories

The compact figures in the main text overlay the two investment directions and show prediction differentiation, respectively. Here we retain all original panels, including balanced exposure, target weights, and submitted orders.

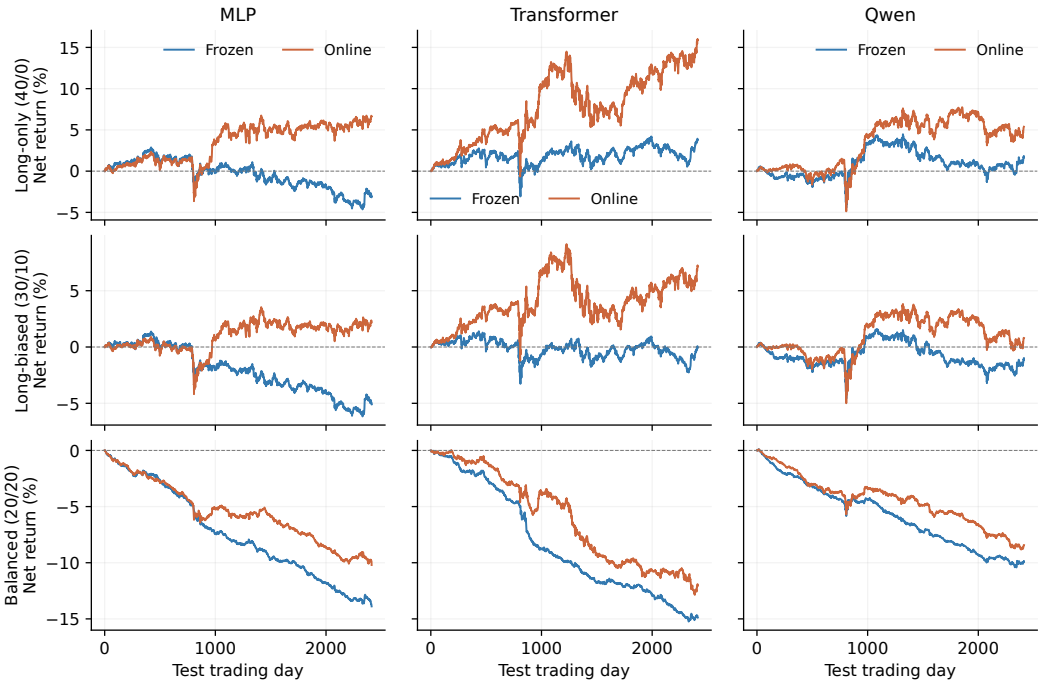


Figure 12: Historical learning across all investment directions. Columns are model families; rows are long-only, long-biased, and balanced portfolios. Blue: frozen parameters; orange: online updates. Cumulative net marked returns include the final cost-bearing liquidation. The reference charges 5 bps fees and a 3 bps per-side execution offset, with a 10% daily turnover cap.

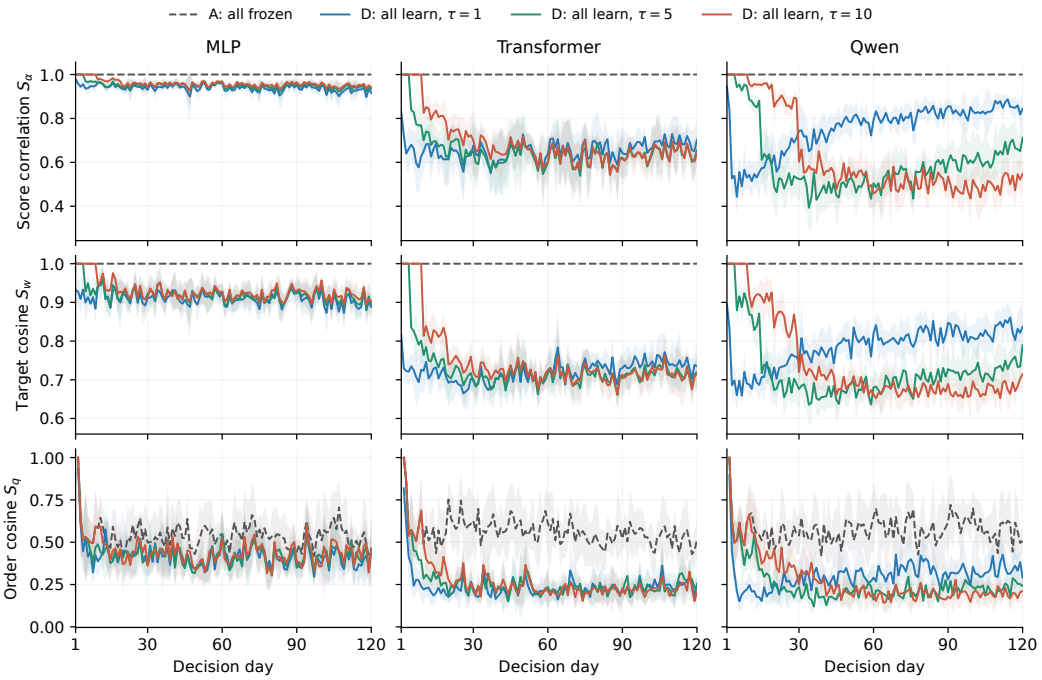


Figure 13: Full differentiation trajectories in FML. Columns are model families; rows show score correlation, target-weight cosine, and submitted-order cosine. A freezes all traders; D updates all traders every τ days. Lines and bands show five-seed means and one sample SD. Daily observations are unsmoothed, and undefined order similarities remain missing. Day 1 is the first recorded decision, not an artificial common pre-update point. Lower similarity denotes greater disagreement, not necessarily better financial outcomes.

I RELATIVE ADAPTATION SPEED

Design and reuse. This extension retains the single-family long-only market: 32 research traders, 20 stocks, 32 features, 120 decision days, and seeds 7, 11, 23, 37, and 41. The eight focal traders and 24 peers update every τ_f and τ_o days, respectively, with both intervals in $\{1, 5, 10\}$. All peers learn ($\rho = 1$). Updates occur on days divisible by the trader's interval, with five independently sampled batches of 256 examples per event. Optimizer learning rates and all other learning and execution settings are unchanged. Consequently, changing an interval changes both the update timing and the number of gradient steps; this is not a learning-rate or equal-compute experiment.

The 90 newly completed paths cover six unequal-interval combinations per model and seed. The 45 equal-interval paths and 45 focal-frozen baselines are reused from the original density–frequency experiment. Before combining results, we check completed endpoints, deployment-parameter hashes, initial market states, learning assignments, and update intervals. This yields 135 learning paths paired with 45 frozen paths, not 180 independent market seeds.

Contrasts. Write $Y_s(\tau_f, \tau_o)$ for the focal cohort's mean cumulative net return in seed s, and $Y_s(\text{frozen}, \tau_o)$ for its matched frozen baseline. The return contrast is

$$G_s(\tau_f, \tau_o) = Y_s(\tau_f, \tau_o) - Y_s(\text{frozen}, \tau_o).$$

Each column shares a frozen comparator; the diagonal is the original $D - C$ contrast. Frozen traders still observe and trade. The population contrast ΔP_s applies the same subtraction to the mean return of all 32 research traders, excluding background participants and the fee account. Percentiles follow Appendix F, including midranks and the $N - 1$ peer denominator, using endpoint accounts.

Individual and collective outcomes. Figure 14 separates relative position from collective financial outcomes. For example, daily focal and peer learning in Qwen improves focal return by 0.188 pp on average, while the population contrast is -0.013 pp. These small means relative to seed dispersion do not establish a general private–collective trade-off. Figure 15 shows associated behavioral changes, not identified return mediators.

Model	τ_f	τ_o	Frozen (%)	Learn (%)	Gain (pp)	Seed SD	+ seeds	ΔP (pp)
MLP	1	1	−0.261	−0.275	−0.014	0.252	3/5	+0.012
	1	5	−0.221	−0.259	−0.038	0.290	3/5	+0.007
	1	10	−0.226	−0.259	−0.033	0.290	2/5	+0.067
	5	1	−0.261	−0.180	+0.081	0.247	3/5	+0.036
	5	5	−0.221	−0.155	+0.066	0.277	2/5	+0.055
	5	10	−0.226	−0.133	+0.093	0.271	3/5	+0.056
	10	1	−0.261	−0.299	−0.038	0.196	1/5	+0.020
	10	5	−0.221	−0.264	−0.043	0.322	3/5	+0.040
	10	10	−0.226	−0.261	−0.035	0.178	3/5	+0.036
Transformer	1	1	+0.099	−0.056	−0.155	0.350	1/5	−0.055
	1	5	−0.025	−0.097	−0.072	0.480	2/5	−0.025
	1	10	−0.017	−0.097	−0.080	0.412	1/5	−0.013
	5	1	+0.099	−0.031	−0.129	0.507	2/5	−0.053
	5	5	−0.025	−0.089	−0.064	0.424	1/5	−0.015
	5	10	−0.017	−0.085	−0.068	0.457	2/5	−0.022
	10	1	+0.099	−0.032	−0.131	0.329	2/5	−0.055
	10	5	−0.025	+0.058	+0.083	0.321	3/5	−0.005
	10	10	−0.017	+0.042	+0.059	0.242	3/5	+0.002
Qwen	1	1	−0.400	−0.212	+0.188	0.455	4/5	−0.013
	1	5	−0.189	−0.260	−0.071	0.316	3/5	−0.030
	1	10	−0.023	−0.186	−0.163	0.653	2/5	−0.103
	5	1	−0.400	−0.355	+0.045	0.474	3/5	−0.081
	5	5	−0.189	−0.232	−0.044	0.311	1/5	+0.006
	5	10	−0.023	−0.269	−0.246	0.521	1/5	−0.094
	10	1	−0.400	−0.289	+0.111	0.347	3/5	−0.042
	10	5	−0.189	−0.270	−0.082	0.333	2/5	−0.036
	10	10	−0.023	−0.228	−0.205	0.425	1/5	−0.064

Table 9: Absolute focal returns and learning contrasts at day 120. Frozen, Learn, Gain, and ΔP are five-seed means; Seed SD is the sample standard deviation of the five paired focal gains. The positive-seed count refers to the focal gain, not positive absolute return. ΔP is the change in all-research-trader mean return. Cells within a seed share a market replication and, within a column, the same frozen comparator.

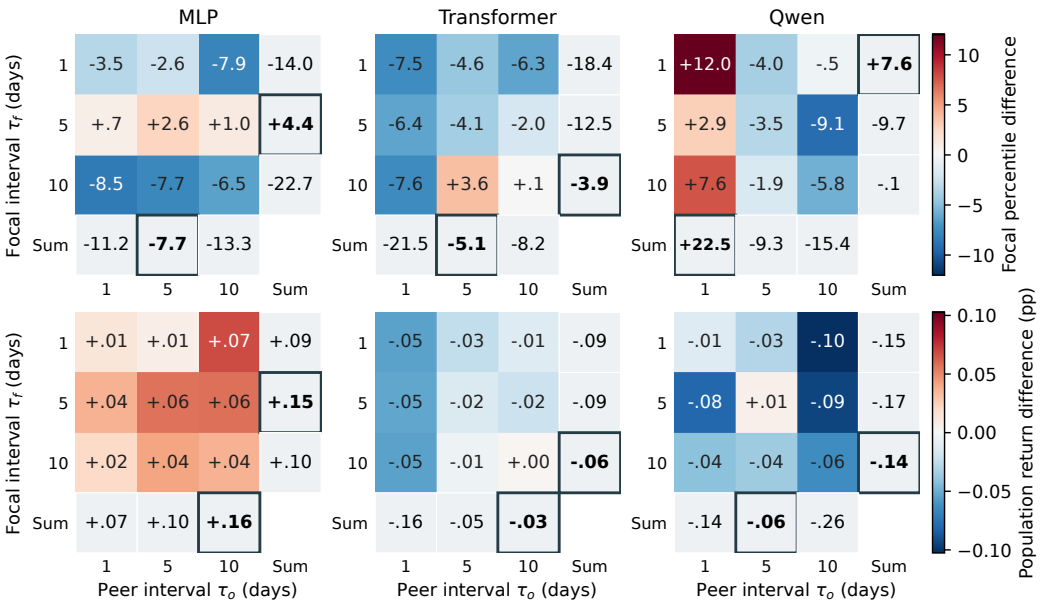


Figure 14: Focal rank and population return under unequal update intervals. Top: change in focal within-population percentile (points). Bottom: change in all-research-trader mean return (pp). Each cell subtracts focal frozen from focal learning at the same peer interval and seed, then averages five seeds. Rows and columns index focal and peer intervals; each panel row shares a color scale across models. Neutral margins and outlined maxima are descriptive grid sums.

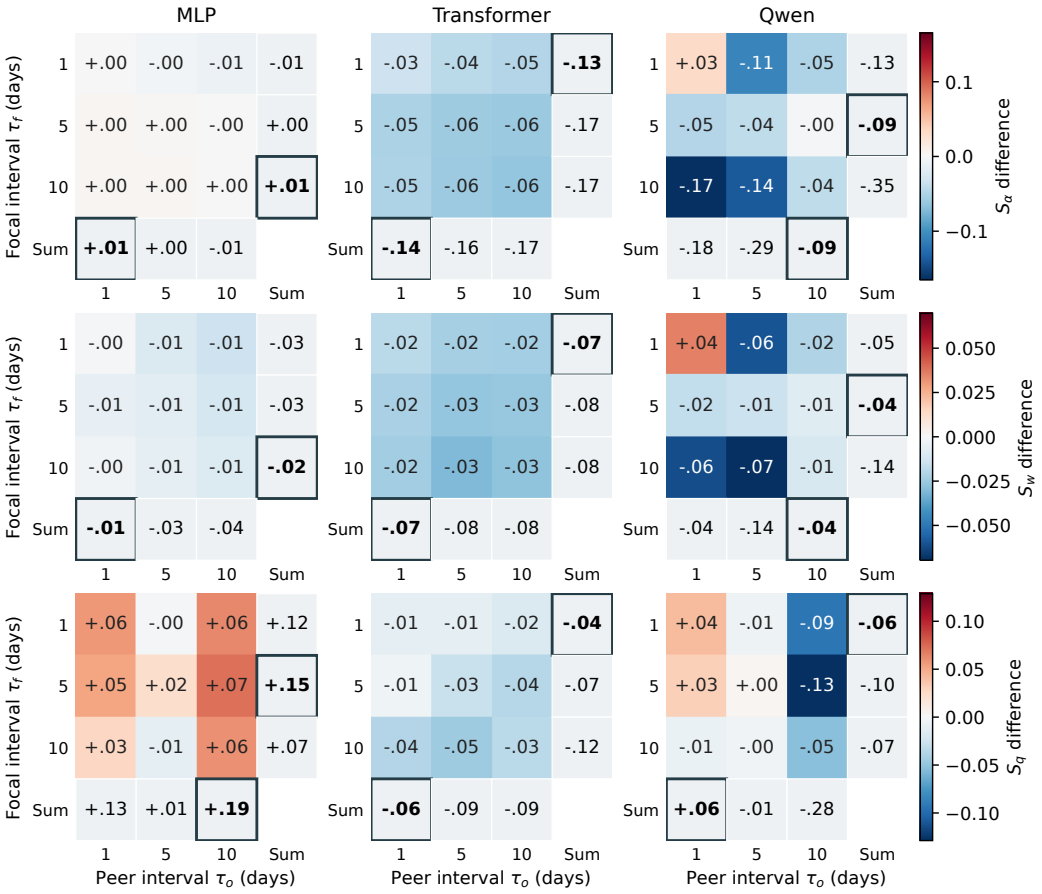


Figure 15: Behavioral changes under unequal update intervals. Rows of panels show prediction correlation, target-weight cosine similarity, and submitted-order cosine similarity at day 120. Each value is focal learning minus focal frozen at the same peer interval, averaged over five seeds. Negative values indicate less similar behavior. Row and column sums are descriptive; they do not supply additional independent observations or measure realized holdings.