FROM MARKET FACTS TO EXECUTABLE MARKETS: GROUNDING HETEROGENEOUS TRADER POPULATIONS IN EMPIRICAL MARKET EVIDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Data-driven financial market models can reproduce or predict observable outcomes, but they provide limited access to the participant-level mechanisms that generate them. Agent-based simulators make such mechanisms explicit, yet typically begin from prescribed trader populations rather than structures constrained by real market evidence. We address this gap with MarketPop, a population-centric framework that treats a heterogeneous, executable trader population as the object of empirical grounding. The population jointly captures who controls resources, where those resources are deployed, which decision rules shape orders, and how those rules behave, allowing changes in participant structure to propagate endogenously through order generation, limit-order-book execution, price formation, and resource updates. MarketPop uses observed stylized facts to constrain the space of plausible populations and searches for candidate markets whose simulated behavior is consistent with these observations. To make this high-dimensional problem tractable, it separates the representation of the market from the representation used to search it: constrained allocations are reparameterized, high-dimensional variation is factorized into shared directions, and every proposal is decoded back into a complete legal population before simulation. Experiments show that the resulting markets reproduce key financial stylized facts and achieve better held-out matching than alternative search strategies under comparable simulation budgets. These solutions should be viewed as evidence-consistent executable markets rather than unique reconstructions of the real trader population, providing empirically constrained starting points for studying how changes in resources, strategies, and behavior propagate through endogenous market dynamics. In this sense, MarketPop connects empirical market evidence with controlled analysis of the participant-level mechanisms that generate it.

1 INTRODUCTION

Financial market simulation is useful not only for reproducing realistic market outcomes, but also for studying how those outcomes change when the market itself is perturbed. Many questions of scientific and practical interest concern changes in participant structure—the concentration of trading resources, the composition of strategies, or the influence of particular trader classes. Addressing such questions requires more than generating plausible prices or order flows: it requires an explicit market mechanism through which participant-level changes propagate into trading decisions, execution, price formation, and subsequent resource updates. A useful simulator for mechanism-oriented analysis should therefore connect manipulable participant structure to the aggregate market behavior it generates.

Existing approaches provide complementary but incomplete foundations for this goal. Data-driven predictive and generative models learn directly from observable market outputs and can reproduce important aspects of empirical market behavior (Zhang et al., 2019; Li et al., 2020; Coletta et al., 2021; Kawawa-Beaudan et al., 2026). Yet the participant structure underlying these outputs is typically implicit, so changes to trader resources, strategies, or behavioral composition are difficult to define and interpret. Agent-based market simulators take the opposite route, making traders and exchange rules explicit so that market dynamics emerge from interacting participants (Byrd et al.,

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.

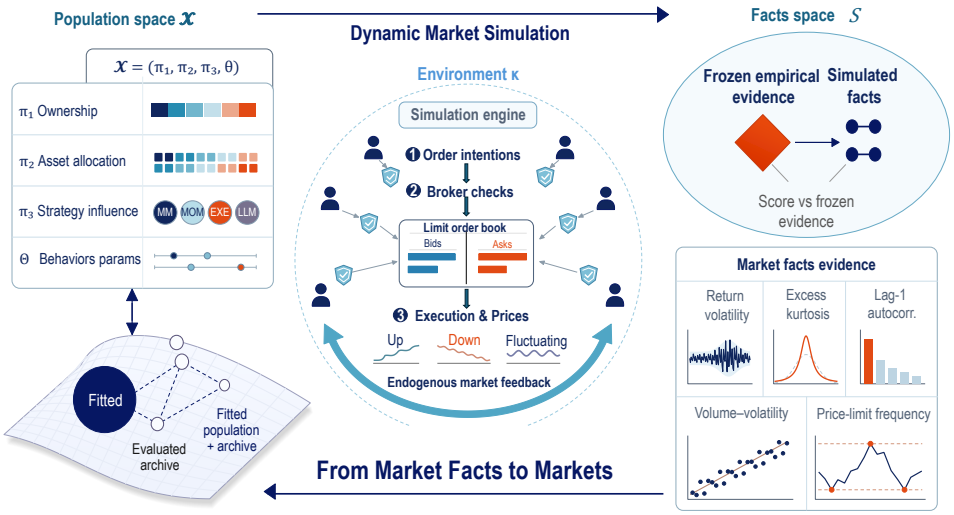


Figure 1: The framework of MarketPop. A complete heterogeneous trader population is executed under a fixed market environment κ and scored against frozen empirical market facts; evidence-guided calibration iteratively refines the population and returns the fitted population together with its evaluated archive.

2020; Yang et al., 2025); but the trader population is usually prescribed in advance, and existing calibration studies estimate type-level, shared, or relatively small sets of parameters rather than a complete heterogeneous population (Bai et al., 2022; Storchan et al., 2021; Stillman et al., 2023). The missing link is therefore a way to construct an executable trader population whose internal structure is explicit enough to manipulate and sufficiently constrained by empirical evidence to provide a credible basis for market analysis.

This leads to the central inverse problem considered in this work: can aggregate market evidence constrain a complete heterogeneous trader population while preserving the execution mechanism that gives the population economic meaning? The problem is inherently underdetermined: different allocations of resources, strategies, and behavioral parameters may generate similar aggregate observations, so market-level evidence generally cannot identify a unique underlying trader society. Our objective is therefore not to recover the real population, but to identify evidence-consistent executable markets—complete trader configurations that, under a fixed market institution, generate simulated behavior compatible with prespecified empirical evidence. We represent such a population through four coupled levels, $X = (\pi_1, \pi_2, \pi_3, \theta)$, capturing who controls resources, where those resources are deployed, which decision rules shape orders, and how those rules behave. These configurations should be interpreted as model-conditional executable markets rather than reconstructions of the hidden composition of the real market.

Treating the full population as the object of empirical grounding, however, creates a difficult computational problem. Its dimensionality grows rapidly with the numbers of traders, assets, strategies, and behavioral parameters; different components obey different constraints; and their effects are strongly coupled through market execution and feedback. Moreover, every candidate configuration must ultimately be assessed through stochastic market simulation rather than a convenient differentiable objective, making direct search over the complete population largely impractical. The central computational challenge is therefore to make the complete population searchable without reducing the simulated population itself.

We address this challenge with MarketPop, a population-centric framework for empirically grounding executable markets (Figure 1). MarketPop separates the representation of the market from the representation used to search it. The simulated market retains the complete four-level population, while the search representation combines exact reparameterizations of constrained population blocks with factorized representations of high-dimensional variation. A deterministic decoder maps every

proposal back to a complete, legal trader population before simulation. Together, these choices lower the effective dimension of search without reducing the simulated population itself. Within MarketPop, we instantiate the calibration procedure as Progressive Hierarchical Inference (PHI), which coordinates level-specific and cross-level changes while adapting representation capacity as search progresses.

The resulting evidence-consistent markets provide empirically constrained starting points for studying how changes in participant structure propagate through endogenous market dynamics. Resources, strategy composition, or behavioral parameters can be varied while holding the market institution and the execution-and-feedback process fixed. Such analyses remain model-conditional rather than identified counterfactuals of the real market. In this sense, MarketPop connects empirical market regularities with controlled analysis of the participant-level mechanisms that generate them.

Our contributions are threefold:

- **MarketPop: a population-centric framework for empirically grounding executable markets.** We introduce MarketPop, which treats the complete heterogeneous trader population as the object constrained by empirical market evidence.
- **Structure-preserving search over complete heterogeneous populations.** We separate the market representation from its search representation: the search representation combines exact reparameterizations of constrained blocks with factorized high-dimensional variation, and a deterministic decoder maps every proposal back to a complete legal population before simulation.
- **Evidence-consistent executable markets for mechanism-oriented analysis.** MarketPop constructs evidence-consistent executable markets for mechanism-oriented analysis. We evaluate this formulation under comparable simulation budgets, including alternative search organizations, progressively enriched population representations, and market evidence outside the calibration window.

2 Related Work

MarketPop lies at the intersection of three lines of research: data-driven and agent-based market modeling, simulation-based calibration of agent populations, and structured search in high-dimensional black-box systems. We review these areas with emphasis on what is modeled, what is empirically constrained, and how population structure is handled during search.

2.1 From Market Outputs to Executable Agent Markets

Data-driven financial models learn observable market processes directly. DeepLOB models price movements from limit-order-book states (Zhang et al., 2019), while generative approaches synthesize order streams or market events using adversarial, point-process, diffusion, and related sequence models (Li et al., 2020; Coletta et al., 2021; Shi & Cartlidge, 2024). More recent systems such as MarS and TradeFM combine learned market representations with interactive or cross-asset generation (Li et al., 2025; Kawawa-Beaudan et al., 2026). These approaches can reproduce important market-level regularities, but the participant structure generating those outcomes is generally not their primary modeling object.

Agent-based models instead make participants and market interaction explicit, spanning classical heterogeneous-agent models of how diverse beliefs and trading rules generate aggregate dynamics (Brock & Hommes, 1998; Lux & Marchesi, 1999; Hommes, 2006), execution platforms such as ABIDES and JAX-LOB (Byrd et al., 2020; Frey et al., 2023), and recent systems with reinforcement learning, LLM-based traders, or empirically informed investor profiles (Karpe et al., 2020; Zhang et al., 2024; Xiao et al., 2024; Yang et al., 2025). MarketPop connects these perspectives by retaining an explicit executable population while using empirical market evidence to constrain that population itself.

2.2 Calibration and Inference for Agent-Based Markets

Simulation-based inference connects observations to mechanistic simulators with intractable likelihoods (Cranmer et al., 2020), spanning Bayesian estimation, approximate Bayesian computation, neural density estimation, and history matching for agent-based models (Beaumont et al., 2002; Grazzini et al., 2017; Dyer et al., 2024; Drovandi et al., 2021).

Financial-market calibration has typically focused on relatively compact parameterizations: trader-type counts, arrival rates, order sizes, and fundamental-process parameters in ABIDES (Bai et al., 2022), small participant-composition variable sets (Storchan et al., 2021), or distributions over shared zero-intelligence and extended Chiarella parameters (Stillman et al., 2023). These studies show that agent-based markets can be empirically calibrated, but their calibration targets are mainly type-level, shared, or low-dimensional parameter sets.

MarketPop instead treats the complete heterogeneous trader population as the calibrated object. Because its components interact through execution, their effects cannot generally be interpreted independently. We therefore view calibration as constructing evidence-consistent executable markets, rather than recovering a unique real-world participant configuration.

2.3 Structured Representations and Search for Population Calibration

Scaling from a small parameter vector to a complete population requires structure in both representation and search. Log-ratio coordinates provide convenient representations for compositional variables (Aitchison, 1986), matrix and tensor factorizations capture shared low-dimensional variation across large parameter arrays (Eckart & Young, 1936; Kolda & Bader, 2009; De Lathauwer et al., 2000), and high-dimensional black-box optimization exploits local, blockwise, or nested subspace structure, as in zeroth-order block-coordinate methods, TuRBO, and BAxUS, as well as adaptive proposal mechanisms such as TPE (Cai et al., 2021; Eriksson et al., 2019; Papenmeier et al., 2022; Bergstra et al., 2011).

MarketPop adopts a different principle: the search representation need not coincide with the market representation. Calibration restricts the directions explored in structured coordinates while decoding every proposal to a complete legal population, rather than replacing the simulated market with a lower-dimensional economic model. Within this framework, PHI organizes level-specific and cross-level updates with adaptive representation capacity rather than serving as a generic black-box optimizer.

3 Methodology

MarketPop empirically grounds a heterogeneous executable trader population by coupling dynamic market simulation with structured calibration. This section develops the framework from the executable population and its search representation to empirical scoring and calibration; Figure 2 provides an overview of the full pipeline.

3.1 Four-Level Trader Population and Dynamic Market Simulation

A simulated market contains an environment and a population of traders acting within it. We denote the environment by κ, which specifies the market institution and the broader simulation setting in which the population operates. The population comprises N traders indexed by $\mathcal{I}$, A assets indexed by $\mathcal{A}$, and S strategy types indexed by $\mathcal{S}$. We represent it as $X = (\pi_1, \pi_2, \pi_3, \theta)$, with $\pi_1 \in \Delta(\mathcal{I})$, $\pi_2 \in \Delta(\mathcal{A})^N$, $\pi_{3,ia} \in \Delta(\mathcal{S})$, and $\theta_{ias} \in \Theta_s$, where $\Delta(\mathcal{U})$ denotes the unit simplex over a finite nonempty set $\mathcal{U}$ and Θ_s denotes the admissible parameter domain of strategy type s. The indices satisfy $i \in \mathcal{I}$, $a \in \mathcal{A}$, and $s \in \mathcal{S}$. π_1 allocates the initial endowment across traders, $\pi_{2,i}$ distributes trader i's endowment across assets, $\pi_{3,ia}$ governs strategy influence within each trader–asset pair, and θ_{ias} specifies the behavioral parameters of each strategy instance.

The four-level population generates market behavior through strategy computation and market execution. In addition to strategy-internal computation, the framework provides a strategy-external information channel, the *Strategy-External Information System* (SEIS; Appendix D), through which information, for example from dynamic trader interaction and belief propagation, enters trader deci-

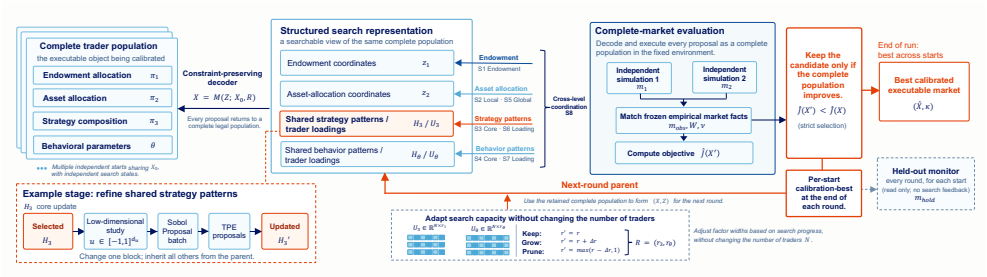


Figure 2: Overview of MarketPop. Structured search coordinates Z are decoded by a constraint-preserving decoder into a complete legal population $X = M(Z; X^0, \mathcal{R})$, executed in the fixed environment κ, and scored by the cross product $\widehat{J}(X)$ of two independent simulation paths against frozen empirical evidence. PHI coordinates the eight stages S1–S8 and returns the best calibrated executable market $(\widehat{X}, \kappa)$.

sion making. The resulting strategy-external state, together with current strategy outputs, induces at each decision time t a bounded transient effective mixture $\widetilde{\pi}_{3,ia}(t)$ anchored at the population-level $\pi_{3,ia}$, thereby shaping how those outputs are combined before they form the trader actions passed to the market-execution process. These trader actions generate executable orders that are matched in the limit order book (Appendix B). Execution updates prices, resources, and market states that inform subsequent decisions, allowing the effects of heterogeneous population structure to propagate endogenously through the evolving market.

The market facts generated by these trajectories make candidate populations empirically comparable. During calibration, these simulated facts are compared with observed market facts while the environment κ is kept common across candidates, so that systematic differences in simulated outcomes can be traced to population configuration rather than changes in the setup. We therefore take the full structured population X as the calibration target, since resource allocation, strategy influence, and behavioral parameters remain coupled through market execution and jointly shape the resulting market outcomes.

3.2 From Population to Search Coordinates

The number of population parameters grows rapidly across the four increasingly fine-grained levels, making direct optimization over the full population X largely impractical. We therefore define structured search coordinates Z around a frozen legal baseline population X^0: X is the market representation, Z the search representation, and every proposal is decoded back to a complete legal population before simulation. The construction separates two roles: exact reparameterizations place constrained population blocks in convenient coordinates, while structured factorization restricts the high-dimensional residual variation that must be searched.

The four population levels induce corresponding search blocks. z_1 parameterizes the traders' endowment shares, and z_2 parameterizes each trader's asset allocation around the baseline; the reported experiments use the exact full-contrast form for the second level, with a scalable low-rank variant available for larger asset sets. The high-dimensional strategy-influence and behavioral-parameter residuals are represented by factorized blocks: trader loadings U_3 and a core H_3 underlie π_3, and trader loadings U_θ and a core H_θ underlie θ. In both cases, trader heterogeneity is expressed through shared latent directions. The full decoding maps, dimensions, variants, and legality conditions are given in Appendix F.

These blocks form the complete search coordinates

$$Z = \big(z_1,\ z_2,\ U_3, H_3,\ U_\theta, H_\theta\big). \tag{1}$$

A frozen deterministic decoder M maps each candidate back to an executable population,

$$X(Z) = M(Z; X^0, \mathcal{R}), \tag{2}$$

where $\mathcal{R}$ specifies the representation architecture. Every decoded population is validated for legality before simulation, and the baseline coordinates reproduce X^0 exactly. The latent directions are

optimization constructs rather than identified economic trader types. The ranks r_3 and r_θ control the capacity of the corresponding factorized residual representations and may be adapted by PHI during search.

3.3 Calibration Objective and Progressive Hierarchical Inference

Each candidate Z is evaluated through its decoded population $X(Z)$, simulated in the shared environment κ and summarized by stylized facts. Let $m_a^{(1)}(X)$ and $m_a^{(2)}(X)$ denote the stylized-fact vectors of asset a obtained from two simulation paths of population X with independent draws from the same simulation-input distribution, and m_a^{obs} the corresponding vector computed from historical data (Cont, 2001). The resulting score is

$$\widehat{J}(X) = \sum_{a=1}^{A} \nu_a \left[m_a^{(1)}(X) - m_a^{\mathrm{obs}}\right]^\top W \left[m_a^{(2)}(X) - m_a^{\mathrm{obs}}\right], \tag{3}$$

where $W = W^\top \succeq 0$ is a fact metric frozen before the run and shared across assets, and $\nu_a \geq 0$ are fixed asset weights. The same pair of random input streams is held fixed across candidates so that scores remain comparable. The calibration search minimizes $\widehat{J}(X(Z))$ over the structured coordinates Z. If either simulation path fails or produces an undefined required fact, the evaluation has no valid score and the candidate cannot replace the current parent.

By independence of the two paths, the expected cross score is obtained by replacing both simulated fact vectors in (3) with their common expectation: the objective thus targets the discrepancy in expected stylized facts, in the moment-matching tradition of agent-based model calibration (Grazzini et al., 2017; Lamperti et al., 2018), while avoiding the simulation-variance bias that squaring a single path's deviation would introduce. Each candidate evaluation therefore requires two complete market simulations.

The search coordinates Z remain organized across four coupled levels, so the search must combine level-specific updates with repeated cross-level coordination. We instantiate this search as *Progressive Hierarchical Inference* (PHI), which additionally adapts representation capacity. Rather than optimizing all coordinates of Z jointly, PHI propagates a complete scored parent through rounds of eight sequential stages (S1–S8): S1 updates z_1; S2/S5 (local/global) revisit z_2; S3/S6 and S4/S7 update H_3, H_θ and U_3, U_θ, respectively; and S8 (Cross) moves all four levels jointly. Every proposal is evaluated through the complete decoded population, and a stage commits only a strict improvement, so accepted changes propagate through the remainder of the round. Alternating stage orders across odd and even rounds repeatedly coordinate the coupled levels.

Between rounds, PHI may adapt the capacity of the factorized third and fourth levels through Keep, Grow, and Prune alternatives from the same scored parent: Keep preserves the current ranks, whereas Grow and Prune increase or decrease either r_3 or r_θ. Grow initially preserves the decoded parent before the added capacity is explored.

Within each stage, a *study* explores a low-dimensional temporary search space around the current parent, so the proposal dimension can remain small even when Z is high-dimensional. Directions combine stage-specific moves, recent successful moves, and fresh exploration, including cross-level directions where appropriate. Each study is warm-started with the parent's validated evaluation at the center of the temporary search space, then begins with scrambled Sobol proposals and switches to TPE after enough distinct valid candidates have been scored. Direction construction, local geometry, and radius adaptation, as well as the structural branch rules, are detailed in Appendix H.

The full run executes this procedure in several independent *starts* sharing the same legal baseline X^0, each maintaining its own search state without exchange. At the end of every round, each start's calibration-best candidate is monitored on held-out data immediately following the calibration window; this evaluation is observe-only and affects neither proposals nor final selection. The run returns the best candidate retained by the calibration search, with coordinates $\widehat{Z}$; its decoded population $\widehat{X} = M(\widehat{Z}; X^0, \mathcal{R})$, together with the environment κ, constitutes the calibrated executable market.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate all configurations in the same multi-asset limit-order-book simulator under the institutional rules of the Chinese A-share market, including T+1 settlement and daily price limits. Market fidelity is measured by five stylized facts: return volatility (Q_1), excess kurtosis (Q_2), first-order autocorrelation of absolute returns (Q_3), volume–volatility correlation (Q_4), and price-limit frequency (Q_5). The empirical reference is constructed from 1,500 A-share stocks over 2022–2025 and summarized by the pooled mean and standard deviation of each fact.

We compare MarketPop with four descriptive population baselines and two methodological baselines. Original is the uncalibrated reference population X^0. Chiarella-3 contains 86/85/85 noise, fundamentalist, and momentum traders; RMSC04-mix contains 229/23/3/1 noise, value, momentum, and market-making traders; and Bai/Storchan uses a 250/5/1 noise–value–market-maker composition. These literature-inspired configurations reproduce representative trader-role compositions within our simulator rather than the original platforms or their calibration procedures (Chiarella et al., 2009; Byrd et al., 2020; Bai et al., 2022; Storchan et al., 2021). Direct-X searches a dense parameterization of the complete population X directly, whereas Joint-Z uses the structured representation Z but optimizes all coordinates jointly. MarketPop uses the same structured representation with the proposed structured calibration procedure.

All comparisons use the same simulator, empirical reference, random-input protocol, and evaluation procedure; the methodological comparison additionally shares the baseline population X^0, calibration objective, and evaluation schedule. The five facts in Table 1 are evaluated over the five-day calibration horizon, with d_{cal} summarizing their aggregate discrepancy. We additionally report d_{HO} over three subsequent continuation days as a temporal check beyond the calibration period. The fact definitions, the empirical reference construction, and the frozen metric are detailed in Appendix G.

4.2 MAIN RESULTS

Table 1 summarizes the market-fidelity results for the descriptive population baselines and the calibration methods.

Table 1: Market-fidelity comparison. Empirical values report pooled mean $\pm$ std; bold facts fall within one empirical standard deviation, bold discrepancies mark the best result (lower is better).

Configuration	Q_1	Q_2	Q_3	Q_4	Q_5	$d_{\mathrm{cal}} \downarrow$	$d_{\mathrm{HO}} \downarrow$
Empirical	17.28 ± 8.63	4.562 ± 10.835	0.188 ± 0.125	0.381 ± 0.166	0.0129 ± 0.099	—	—
Original	**21.31**	**0.381**	0.044	**0.272**	**0**	13.17	15.10
Chiarella-3	4.14	**7.212**	**0.133**	0.131	**0.016**	7.82	12.04
RMSC04-mix	**9.68**	-0.081	**0.077**	0.085	**0.004**	9.09	10.03
Bai/Storchan	**9.82**	-0.326	**0.076**	0.076	**0**	9.46	10.88
Direct-X	**17.90**	**0.082**	0.061	0.185	**0**	9.88	10.15
Joint-Z	**17.27**	-0.144	**0.124**	0.206	**0**	5.60	6.03
MarketPop (Ours)	**13.35**	**0.551**	**0.158**	**0.260**	**0**	**1.47**	**4.89**

The descriptive population baselines reproduce different subsets of the empirical market facts, but none of the three literature-inspired compositions reproduces the empirical range of Q_4. These results indicate that transferring trader-role compositions alone does not provide a uniformly faithful market configuration, motivating calibration of the underlying heterogeneous population rather than selection among fixed population templates. We caution that a zero Q_5 is not by itself evidence of realism: the empirical price-limit frequency is low and its reference band is wide, so both zero and small nonzero frequencies remain compatible with the empirical reference.

The methodological comparison shows successive gains from the two components of MarketPop. Replacing the dense population parameterization of Direct-X with the structured representation Z reduces d_{cal} from 9.88 to 5.60 and d_{HO} from 10.15 to 6.03. Applying the proposed structured cal-

ibration procedure further reduces these discrepancies to 1.47 and 4.89, respectively. MarketPop is also the only calibrated configuration whose five reported facts all fall within one empirical standard deviation, including Q_4, which remains below the empirical range for both Direct-X and Joint-Z. The continuation-period result provides an additional temporal check, with the same improvement pattern persisting beyond the calibration horizon.

4.3 Ablation Studies

Table 1 establishes the benefit of the proposed method as a whole. To isolate the contribution of the four-level population hierarchy, we keep the simulator, calibration objective, and structured calibration procedure fixed and vary only which population levels are open to calibration. The levels are opened successively in their natural order, and levels outside the active slice remain fixed at the baseline population X^0. This differs from Direct-X in Section 4.2, which jointly searches a dense parameterization of the complete population rather than ablating population levels.

Table 2: Population-hierarchy ablation. Each row opens the indicated prefix of population levels for calibration; the rest stay fixed at X^0. Boldface as in Table 1. The final row coincides with MarketPop (Ours).

Open levels	Q_1	Q_2	Q_3	Q_4	Q_5	$d_{\mathrm{cal}} \downarrow$	$d_{\mathrm{HO}} \downarrow$
π_1	**16.54**	-0.268	**0.160**	**0.260**	**0**	2.98	**4.85**
$\pi_1+\pi_2$	**17.72**	**0.956**	**0.170**	**0.248**	**0**	3.60	4.89
$\pi_1+\pi_2+\pi_3$	**11.34**	**0.546**	**0.177**	0.197	**0**	2.56	5.44
$\pi_1+\pi_2+\pi_3+\theta$ (**MarketPop**)	**13.35**	**0.551**	**0.158**	**0.260**	**0**	**1.47**	4.89

As shown in Table 2, the first population level already accounts for much of the achievable market fidelity: calibrating π_1 alone places all five targeted facts within their empirical ranges, with $d_{\mathrm{cal}} = 2.98$ and the lowest held-out discrepancy in the ladder ($d_{\mathrm{HO}} = 4.85$). Opening the higher levels does not produce monotonic gains, but provides additional degrees of freedom for further refinement: the full four-level configuration ultimately reduces d_{cal} to 1.47 and, relative to the three-level configuration, brings Q_4 back into the empirical range. In other words, the four-level hierarchy does not require every level to undergo large adjustments; it exposes the complete adjustable space and lets calibration itself determine which levels carry the main adjustment.

We additionally ablate SEIS while keeping the calibrated population $\widehat{X}$ fixed and performing no recalibration: disabling SEIS changes d_{cal} from 1.47 to 1.95 and d_{HO} from 4.89 to 5.23, while all five targeted facts remain within their empirical bands. The complete fact-level comparison is reported in Appendix K.

We close the section by asking where the calibrated population $\widehat{X}$ departs from the initial configuration X^0; the level-by-level comparison is reported in Appendix L. The departure concentrates in the endowment level π_1—the uniform initial endowment becomes a skewed allocation (Gini $0 \to 0.073$; the wealthiest decile holds 12.6%)—while the remaining levels move only marginally. This anatomy complements the ladder from the opposite direction: opening π_1 alone is already sufficient for band-level fidelity, and when all four levels are open, calibration still concentrates its adjustment in π_1. Together, the two views identify the initial-resource level as the *dominant calibration margin* for the targeted market facts; the higher levels provide additional flexibility for refinement rather than uniformly large contributions.

4.4 Robustness Analysis

We report three robustness checks; the formal criteria and all diagnostics are in Appendix J.8.

(R1, measurement precision) A preregistered window audit on the frozen baseline population finds the discrepancy score measured well within the noise our comparisons require: the worst-case normalized standard error of any window mean is 0.096 on the conservative per-asset dispersion scale, and 0.254—at the $\gamma = 0.25$ reference boundary—under the main held-out protocol on the pooled

scale. Precision is not stationarity or fit: 6 of 20 nondegenerate series retain detectable drift, and lengthening the window raises the aggregate discrepancy from 67.27 to 69.07 even as cross-path disagreement falls from 0.45% to 0.22%.

(R2, budget and search path) The improvement in held-out agreement is front-loaded: the best-so-far d_{HO} falls rapidly early in the search and enters a diminishing-improvement region within roughly the first 14% of the evaluation budget, so the budget behind Table 1 is spent well past the knee. The two search starts from the same X^0 reach registered held-out scores within a few percent of each other, so the final result in Table 1 does not hinge on a single favorable search path.

(R3, random realizations) The paired cross-path construction makes the score unbiased for fixed windows, so random-stream sensitivity enters only through variance; the measured cross-path disagreement (0.45% down to 0.22% of the score) is an order of magnitude below the differences separating configurations in Table 1. Finally, similar aggregate agreement need not imply similar population structure: the empirical moments identify an equivalence set of executable populations rather than a unique microstate, and we reserve structural claims for features stable across this set.

4.5 Financial Properties and Controlled Market Analysis

Stylized-Fact Profile

Table 1 and the stylized-fact profile of Appendix J.10 give a dimension-by-dimension economic reading of the calibrated outcome. MarketPop is the only calibrated configuration within one empirical standard deviation of the pooled reference on all five facts. In particular, it restores the volume–volatility coupling Q_4, which remains more than one standard deviation below the pooled mean for Direct-X and Joint-Z. The gain is therefore not merely a lower weighted score: one executable population jointly aligns return scale, tail thickness, volatility persistence, trading-activity coupling, and price-limit incidence. We report this profile per dimension in the appendix rather than reducing it to average coverage.

Population Size and Dynamic Stability

We test whether market activity mechanically diversifies as the population grows. Fixing total wealth and the strategy mix, we vary $N \in \{64, 128, 256, 512\}$ over three preregistered paths per size and rescale each trader's resources by $256/N$ (days 2–10). Equal wealth slices close the granularity channel (Gabaix, 2011), making $\sigma \propto N^{-1/2}$ (Giardina & Bouchaud, 2003) a transparent independent-trading null for interaction-generated scaling (Lux & Marchesi, 1999; Alfi et al., 2009). The full scaling grid is reported in Appendix J.9 (Figure J.3).

Across the lot-consistent grid $N = 64, 128, 256$, volatility is 19.94, 19.42, and 19.54 bp. The -0.20 bp-per-octave slope is non-monotone within seed variation and, in relative terms, roughly 30 times flatter than the 29% decline per octave implied by $N^{-1/2}$. The round-lot-constrained $N = 512$ result (16.59 bp) is shown only as a sensitivity point. All four estimates remain within one empirical standard deviation of the pooled reference. At fixed composition and aggregate wealth, correlated strategy-driven demand therefore offsets the diversification predicted by independent trading; increasing headcount alone does not mechanically attenuate aggregate volatility.

5 Conclusion

We introduce MarketPop, a population-centric framework that treats the complete heterogeneous trader population as the object of empirical grounding. By separating the representation of the market from the representation used to search it, MarketPop lowers the effective dimension of search without reducing the simulated population itself. The present study is limited to a restricted equity universe and time windows under a single institutional setting, and the resulting evidence-consistent executable markets remain conditional on the chosen population schema and baseline; controlled interventions within these markets characterize endogenous model responses rather than identified real-world causal effects. More broadly, this work moves financial market simulation from reproducing market outputs toward empirically grounded experimentation on the mechanisms that generate them.

AI USE STATEMENT

During the preparation of this work, the authors used Large Language Models (LLMs) for the following purposes, with all uses disclosed in accordance with the ICLR 2027 AI Policy for Authors.

Writing assistance and polishing. LLMs were used to aid and polish the writing of this paper, including improving grammar, clarity, and readability of author-written text. All technical content, arguments, and conclusions were formulated, verified, and finalized by the human authors.

Retrieval and discovery. LLMs were used to assist in finding and discovering related work. The authors independently reviewed, verified, and selected all cited references, and take full responsibility for the accuracy and appropriateness of the literature discussed.

Drafting sections. LLMs were used to draft portions of the paper's text. All AI-drafted content was subsequently reviewed, edited, and validated by the human authors. The authors take full responsibility for the final content of the paper, including any text initially generated by AI tools.

No AI tools were used for research ideation or execution, generating synthetic datasets, proving mathematical claims, or any purpose other than those explicitly stated above.

The authors confirm that they have reviewed and verified all content in this paper and take full responsibility for its accuracy, originality, and integrity.

REPRODUCIBILITY STATEMENT

Reproducibility is organized around three complementary layers: the formal model, the executable protocol, and the identity of a reported run. Section 3 specifies the four-level population, the structured coordinates and legality-preserving decoder, the calibration objective, and the PHI lifecycle; Section 4.1 fixes the comparison and evaluation protocol. The operational components are documented separately in the appendices: population initialization, strategy evaluation, and order aggregation in Appendix B; the strategy catalogue in Appendix C; the market institution in Appendix E; the strategy-external information mechanism in Appendix D; and the decoding maps and legality conditions in Appendix F. Appendix G defines the empirical facts, frozen metric, two-path score, common-random-input treatment, and the separation between calibration, temporal held-out monitoring, and fresh-seed evaluation. The search and rank-adaptation algorithms are given in Appendices H and H.9–H.11, with assumptions and proofs in Appendix I. Appendix J then binds these definitions to the reported experiments by recording the data-filtering and window conventions, assets and dates, numerical profile, evaluation budget, path accounting, checkpoint state, evaluation identity, and replay-manifest requirements. This separation is intended to make an exact replay auditable without conflating it with independent fresh-randomness assessment.

REFERENCES

John Aitchison. *The Statistical Analysis of Compositional Data*. Chapman and Hall, London, 1986.

Valentina Alfi, Matthieu Cristelli, Luciano Pietronero, and Andrea Zaccaria. Mechanisms of self-organization and finite size effects in a minimal agent based model. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03016, 2009. doi: 10.1088/1742-5468/2009/03/P03016.

Robert Almgren and Neil Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3(2): 5–40, 2001. doi: 10.21314/JOR.2001.041.

Marco Avellaneda and Sasha Stoikov. High-frequency trading in a limit order book. *Quantitative Finance*, 8(3):217–224, 2008. doi: 10.1080/14697680701381228.

Yuanlu Bai, Henry Lam, Tucker Balch, and Svitlana Vyetrenko. Efficient calibration of multi-agent simulation models from output series with bayesian optimization. In *Proceedings of the Third ACM International Conference on AI in Finance (ICAIF 2022)*, pp. 437–445, 2022. doi: 10.1145/3533271.3561755.

Mark A. Beaumont, Wenyang Zhang, and David J. Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002. doi: 10.1093/genetics/162.4.2025.

James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, volume 24, pp. 2546–2554. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html.

William A. Brock and Cars H. Hommes. Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, 22(8–9):1235–1274, 1998.

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 (SIGSIM-PADS '20)*, pp. 11–22, 2020. doi: 10.1145/3384441.3395986.

Hanqin Cai, Yuchen Lou, Daniel Mckenzie, and Wotao Yin. A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1193–1203. PMLR, 2021. URL https://proceedings.mlr.press/v139/cai21d.html.

Carl Chiarella, Giulia Iori, and Josep Perelló. The impact of heterogeneous trading rules on the limit order book and order flows. *Journal of Economic Dynamics and Control*, 33(3):525–537, 2009. doi: 10.1016/j.jedc.2008.08.001.

Andrea Coletta, Matteo Prata, Michele Conti, Emanuele Mercanti, Novella Bartolini, Aymeric Moulin, Svitlana Vyetrenko, and Tucker Balch. Towards realistic market simulations: A generative adversarial networks approach. In *Proceedings of the Second ACM International Conference on AI in Finance (ICAIF 2021)*, pp. 1–9, 2021. doi: 10.1145/3490354.3494411.

Rama Cont. Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236, 2001.

Rama Cont and Jean-Philippe Bouchaud. Herd behavior and aggregate fluctuations in financial markets. *Macroeconomic Dynamics*, 4(2):170–196, 2000. doi: 10.1017/S1365100500015029.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000. doi: 10.1137/S0895479896305696.

Christopher Drovandi, David J. Nott, and Daniel E. Pagendam. A semiautomatic method for history matching using sequential monte carlo. *SIAM/ASA Journal on Uncertainty Quantification*, 9(3):1034–1063, 2021. doi: 10.1137/19M1286694.

Joel Dyer, Patrick Cannon, J. Doyne Farmer, and Sebastian M. Schmon. Black-box bayesian inference for agent-based models. *Journal of Economic Dynamics and Control*, 161:104827, 2024. doi: 10.1016/j.jedc.2024.104827.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D. Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/6c990b7aca7bc7058f5e98ea909e924b-Abstract.html.

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 4th ACM International Conference on AI in Finance (ICAIF 2023)*, pp. 583–591, 2023. doi: 10.1145/3604237.3626880.

Xavier Gabaix. The granular origins of aggregate fluctuations. *Econometrica*, 79(3):733–772, 2011. doi: 10.3982/ECTA8769.

Irene Giardina and Jean-Philippe Bouchaud. Bubbles, crashes and intermittency in agent based market models. *European Physical Journal B*, 31(3):421–437, 2003.

Paul Glasserman and David D. Yao. Some guidelines and guarantees for common random numbers. *Management Science*, 38(6):884–908, 1992. doi: 10.1287/mnsc.38.6.884.

Dhananjay K. Gode and Shyam Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, 1993. doi: 10.1086/261868.

Jakob Grazzini, Matteo G. Richiardi, and Mike G. Tsionas. Bayesian estimation of agent-based models. *Journal of Economic Dynamics and Control*, 77:26–47, 2017. doi: 10.1016/j.jedc.2017.01.014.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012. URL https://www.jmlr.org/papers/volume13/gretton12a/gretton12a.pdf.

Cars H. Hommes. Heterogeneous agent models in economics and finance. In *Handbook of Computational Economics*, volume 2, pp. 1109–1186. Elsevier, 2006. doi: 10.1016/S1574-0021(05)02023-X.

Michaël Karpe, Jin Fang, Zhongyao Ma, and Chen Wang. Multi-agent reinforcement learning in a realistic limit order book market simulation. In *Proceedings of the First ACM International Conference on AI in Finance*, pp. 1–7, 2020. doi: 10.1145/3383455.3422570.

Maxime Kawawa-Beaudan, Srijan Sood, Kassiani Papasotiriou, Daniel Borrajo, and Manuela Veloso. TradeFM: A generative foundation model for trade-flow and market microstructure. arXiv preprint arXiv:2602.23784, 2026.

Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. doi: 10.1137/07070111X.

Francesco Lamperti, Andrea Roventini, and Amir Sani. Agent-based model calibration using machine learning surrogates. *Journal of Economic Dynamics and Control*, 90:366–389, 2018. doi: 10.1016/j.jedc.2018.03.011.

Junjie Li, Yang Liu, Weiqing Liu, Shikai Fang, Lewen Wang, Chang Xu, and Jiang Bian. MarS: A financial market simulation engine powered by generative foundation model. In *International Conference on Learning Representations (ICLR 2025)*, 2025.

Junyi Li, Xintong Wang, Yaoyang Lin, Arunesh Sinha, and Michael Wellman. Generating realistic stock market order streams. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):727–734, 2020. doi: 10.1609/aaai.v34i01.5415.

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

Daniel McFadden. A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica*, 57(5):995–1026, 1989. doi: 10.2307/1913621.

Art B. Owen. Scrambled net variance for integrals of smooth functions. *The Annals of Statistics*, 25(4):1541–1562, 1997. doi: 10.1214/aos/1031594731.

Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive Bayesian optimization in nested subspaces. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/4b7439a4ab0b8e4bcb4e2412c6a10a58-Abstract-Conference.html.

Zijian Shi and John Cartlidge. Neural stochastic agent-based limit order book simulation with neural point process and diffusion probabilistic model. *Intelligent Systems in Accounting, Finance and Management*, 31(2):e1553, 2024. doi: 10.1002/isaf.1553.

Namid R. Stillman, Rory Baggott, Justin Lyon, Jianfei Zhang, Dingqiu Zhu, Tao Chen, and Perukrishnen Vytelingum. Deep calibration of market simulations using neural density estimators and embedding networks. In *Proceedings of the Fourth ACM International Conference on AI in Finance (ICAIF 2023)*, pp. 46–54, 2023. doi: 10.1145/3604237.3626881.

Victor Storchan, Svitlana Vyetrenko, and Tucker Balch. Learning who is in the market from time series: Market participant discovery through adversarial calibration of multi-agent simulators. arXiv preprint arXiv:2108.00664, 2021.

Y. Xiao, E. Sun, D. Luo, and W. 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. In Danielle Belgrave, Cheng Zhang, Hongzhou Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, and Nan Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 63469–63519. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5bf234ecf83cd77bc5b77a24ba9338b0-Paper-Conference.pdf.

C. Zhang, X. Liu, Z. Zhang, M. Jin, L. Li, Z. Wang, W. Hua, D. Shu, S. Zhu, X. Jin, S. Li, M. Du, and Y. Zhang. When AI meets finance (StockAgent): Large language model-based stock trading in simulated real-world environments. arXiv preprint arXiv:2407.18957, 2024.

Zihao Zhang, Stefan Zohren, and Stephen Roberts. DeepLOB: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, 2019. doi: 10.1109/TSP.2019.2907260.

A Notation and Appendix Roadmap

These appendices specify the forward simulator and its use in population calibration. Appendices B–E specify the executable market and its simulation environment: Appendix B describes how an executable population becomes a simulated market, Appendix C specifies the seven strategy templates in the reported population, Appendix D specifies the strategy-external information system, and Appendix E describes the pluggable market-institution layer, instantiated with the A-share trading rules used in the reported experiments. Appendices F–I develop the calibration methodology: Appendix F defines the search-space representation and the deterministic decoder, Appendix G the calibration objective and the frozen metric, Appendix H states the Progressive Hierarchical Inference procedure, including its adaptive capacity mechanism, and Appendix I collects the analytical properties and their scope. Appendix J records the experimental binding and reproducibility information.

All population and search symbols follow Section 3. Table A.1 collects the notation shared across the appendices; symbols local to a single appendix are defined where they are used, and the profile's key numerical settings are reported in Appendix J.

B Executable Population and Dynamic Market Simulation

The four-level population $X = (\pi_1, \pi_2, \pi_3, \theta)$ is instantiated as an executable trader population in three steps: initialization materializes the share blocks π_1, π_2 into account state, strategy instances governed by θ generate order intentions that are combined through an execution mixture anchored at π_3, and the combined intentions are aggregated into executable orders and matched in the limit order book, whose fills update the market and account state observed by subsequent decisions.

Table A.1: Shared population, simulation, and search notation.

Symbol	Meaning and shape
Indices and dimensions	
i, a, s	Trader, asset, and strategy-type indices, $i \in \mathcal{I}$, $a \in \mathcal{A}$, $s \in \mathcal{S}$.
N, A, S, P	Numbers of traders, assets, strategy types, and active numerical strategy-parameter fields.
ℓ	Strategy-contrast index, $\ell = 1, \ldots, S-1$.
p	Flattened numerical-field index, $p = 1, \ldots, P$.
t, d, h	Distinct clocks: simulation tick, trading day, and within-day price-grid index.
$\times_n$	Mode-n product of a tensor with a matrix.
$\|\cdot\|_\infty$	Largest absolute entry of an array.
Population and search representation	
X	Complete heterogeneous trader population.
π_1	Trader-level endowment shares; shape N.
π_2	Trader–asset allocation shares; shape $N \times A$.
π_3	Strategy-influence shares within trader–asset pairs; shape $N \times A \times S$.
θ	Strategy-instance behavioral parameters; θ_{ias} is a type-specific parameter object.
X^0	Frozen legal baseline population.
M	Deterministic decoder from search coordinates to a complete population, $X(Z) = M(Z; X^0, \mathcal{R})$.
$\mathcal{R}$	Representation architecture used by the decoder.
Z	Full search tuple $(z_1, z_2, U_3, H_3, U_\theta, H_\theta)$.
z_1	First-level endowment reference logits, $z_1 \in \mathbb{R}^{N-1}$.
z_2	Second-level search block; its internal representation depends on the declared second-level architecture and initialization branch (Appendix F). The reported profile uses the full-contrast block B_2.
d_Z	Persistent coordinate count of Z.
r_3, r_θ	Trader-mode ranks of the strategy and parameter factorizations.
U_3, H_3	Strategy residual factors, $N \times r_3$ and $r_3 \times A \times (S-1)$.
U_θ, H_θ	Parameter residual factors, $N \times r_\theta$ and $r_\theta \times A \times P$.
E_2, E_3, E_θ	Residual arrays relative to the baseline in the second-, third-, and fourth-level decoder spaces.
W_0	Total endowment, divided across trader–asset accounts by π_1 and π_2; materialization variants are given in Appendix B.1.
c_2^0	Frozen cash-fraction table used by the fixed cash-fraction initialization profile, $N \times A$.
Simulation and calibration	
κ	Simulation environment: the market institution and the broader simulation setting, held fixed across candidate evaluations.
$\widetilde{\pi}_3(t)$	Transient effective strategy mixture, same shape as π_3.
$C_{ia}^{\mathrm{avail}}, Q_{ia}^{\mathrm{sell}}$	Available cash and sellable holdings of account (i, a), net of reservations against open orders.
ξ	Simulation random input; candidate scores use one frozen pair $\xi^{(1)}, \xi^{(2)}$ shared across candidates.
$m_a(X;\xi), \mu_a(X), m_a^{\mathrm{obs}}$	Simulated, expected, and observed fact vectors of asset a, K_f-dimensional, with $\mu_a(X) = \mathbb{E}_\xi\, m_a(X;\xi)$.
K_f	Fact-vector length.
W	Common $K_f \times K_f$ fact metric, $W = W^\top \succeq 0$, frozen before candidate evaluation.
$J, \widehat{J}$	Expected-fact calibration loss and realized cross-path score.
ν_a	Fixed asset aggregation weight, $\nu_a \geq 0$.

B.1 POPULATION INITIALIZATION

Initialization converts the share blocks into concrete account state before any strategy is called. The endowment W_0 is divided hierarchically—first across traders by π_1, then across assets within each trader by π_2—defining ideal budgets $B^*_{ia} = W_0\pi_{1,i}\pi_{2,ia}$; for a declared endowment representable in the account's monetary unit, a deterministic largest-remainder allocation in the smallest currency unit, applied first across traders and then across assets within each trader, with ties broken by frozen identity order, maps these ideals to representable monetary budgets B_{ia}, preserving W_0 exactly. Initialization prices $p^0_a > 0$, lot sizes lot_a, and the settlement status of initial holdings are frozen across all candidates of a run, and initial holdings are established directly, without a buy in the order book and without transaction fees. Three initialization variants are supported; they are configuration choices and are never combined within one run.

- **Branch A (exogenous holdings).** Each budget is materialized entirely as cash, $C_{ia,0} = B_{ia}$. Initial holdings $H_{ia,0}$ come from a separate frozen holdings policy and are recorded in the holdings ledger; they do not reduce the cash accounts and do not enter the simplex blocks.
- **Branch B (wealth split, fixed cash-fraction table).** Each account budget is partitioned between initial cash and initial holdings value according to a frozen cash-fraction table $c^0_2 = (c_{2,ia}) \in [0,1]^{N\times A}$ supplied by configuration: holdings are rounded down to whole lots at the initialization price, and the cash account absorbs the rounding remainder,

$$H_{ia,0} = \mathrm{lot}_a \Big\lfloor \frac{B_{ia}(1-c_{2,ia})}{p^0_a \,\mathrm{lot}_a} \Big\rfloor, \qquad C_{ia,0} = B_{ia} - p^0_a H_{ia,0}. \tag{B.1}$$

  By construction $H_{ia,0} \in \mathrm{lot}_a\,\mathbb{Z}_{\geq 0}$, $C_{ia,0} \geq B_{ia}c_{2,ia} \geq 0$, and $C_{ia,0} + p^0_a H_{ia,0} = B_{ia}$, so realized totals conserve W_0 exactly; the cash fraction is therefore a target proportion, and lot quantization can only raise the realized cash share.
- **Branch C (wealth split, calibrated cash fraction).** As in Branch B, but the cash fraction becomes a searched sub-block of the second level, decoded per trader–asset account by the sigmoid map of Appendix F.5. This variant is supported by the framework but is not instantiated in the reported profile.

The reported experiments use Branch B: the fraction c^0_2 is fixed by configuration and occupies no search coordinates, so the searched second-level block is the asset allocation π_2 alone.

B.2 STRATEGY OUTPUTS AND THE EXECUTION MIXTURE

For each trader–asset account, the simulator distinguishes total, available, and reserved cash, together with total, sellable, and reserved holdings; these quantities form the authoritative resource state of the account. All strategy instances of the same trader–asset pair consume this shared state: each reads the same available cash C^{avail}_{ia} and sellable holdings Q^{sell}_{ia}, both net of reservations against open orders. Strategy weights create no subaccounts—resources are owned by the trader–asset account, and the execution mixture acts only on validated intention sizes after strategy computation. The parameter object θ_{ias} governs how the instance of strategy type s responds to its observations; its output is a batch of raw order intentions, each carrying side, order type, size, and pricing information where applicable. The strategy-visible state comprises the current market view, the authoritative account and resource state, the strategy scope, and the visible active orders, with strategy-specific auxiliary views supplied where declared. The execution interface chains three steps: the strategy-visible state and θ produce a batch of raw order intentions, which validation turns into validated intentions; the population-level π_3, the strategy-external state, and the validated intentions determine the effective mixture $\widetilde{\pi}_3(t)$; and applying that mixture to the validated intentions yields the weighted intentions passed to aggregation.

Before any weighting, each strategy's complete batch must respect the account's current resources, checked per side:

$$0 \leq \sum_j b^{\mathrm{raw}}_{iasj} \leq C^{\mathrm{avail}}_{ia}, \qquad 0 \leq \sum_j v^{\mathrm{raw}}_{iasj} \leq Q^{\mathrm{sell}}_{ia}, \tag{B.2}$$

where b^{raw}_{iasj} is buy notional and v^{raw}_{iasj} is sell quantity of intention j. If either cumulative bound is violated, the offending side of the batch fails closed—it is rejected rather than silently rescaled,

normalized, or pre-quantized—while cancellation commands consume no new buy or sell resource and are validated separately through their own control path.

The execution mixture is applied once, outside the strategy logic, to the size of each validated new-order intention. The mixture is the population-level π_3 when no strategy-external intervention is active and the transient $\widetilde{\pi}_3(t)$ of Appendix D otherwise; in both cases its entries over s sum to one within each active trader–asset pair. Writing $\gamma_{ias}(t) \in [0,1]$ for the proposal-level confidence output of the strategy when it exposes one (and $\gamma_{ias}(t) = 1$ otherwise)—per-intention confidence values, where present, are audit metadata and do not participate in scaling—the effective weight and the weighted sizes are

$$\begin{aligned} w_{ias}(t) &= \widetilde{\pi}_{3,ias}(t)\,\gamma_{ias}(t), \\ b^{\mathrm{wtd}}_{iasj}(t) = w_{ias}(t)\,b^{\mathrm{val}}_{iasj}(t), &\qquad v^{\mathrm{wtd}}_{iasj}(t) = w_{ias}(t)\,v^{\mathrm{val}}_{iasj}(t). \end{aligned} \tag{B.3}$$

Since each strategy's validated batch satisfies (B.2), the mixture rows sum to one, and proposal confidences lie in $[0,1]$, the weighted aggregate across strategies still respects the account-level resource bounds before fees and final admission. Cancellation and similar control commands act on existing orders, carry no intention-size variable, and are never weighted.

The endowment shares π_1, π_2 shape the initial account state only and are never multiplied into orders again. After initialization, cash, holdings, reservations, fees, releases, settlement, and trade proceeds evolve through market execution; the simulator does not reapply π_2 to rebalance accounts or rescale orders during the path.

B.3 Order aggregation and market execution

Weighted intentions become executable orders through one frozen aggregation procedure. Within each trader–asset–side scope, intentions with the same execution attributes are merged by limit-price tier, with amounts summing exactly within a tier; buy and sell intentions are aggregated separately and are never netted. Each tier is then legalized by a rolling lot quantization, and the direction of the roll differs by side because the constrained resource differs: buys consume cash, sells consume inventory.

For buys, the n_B tiers are processed in descending price order; tier k at price p_k carries merged notional m_k together with the incoming cash residual c_k (with $c_1 = 0$ and lot_a written L):

$$\begin{aligned} A_k = m_k + c_k, \qquad v_k = L\Big\lfloor \frac{A_k}{p_k L} \Big\rfloor, \qquad c_{k+1} = A_k - p_k v_k, \\ \sum_k p_k v_k + c_{n_B+1} = \sum_k m_k, \qquad 0 \le c_{n_B+1} < p_{n_B} L. \end{aligned} \tag{B.4}$$

For sells, the n_S tiers are processed in ascending price order; tier k carries merged quantity q_k together with the incoming share residual r_k (with $r_1 = 0$):

$$\begin{aligned} V_k = q_k + r_k, \qquad v_k = L\Big\lfloor \frac{V_k}{L} \Big\rfloor, \qquad r_{k+1} = V_k - v_k, \\ \sum_k v_k + r_{n_S+1} = \sum_k q_k, \qquad 0 \le r_{n_S+1} < L. \end{aligned} \tag{B.5}$$

Issued orders use the resolved execution prices associated with their intentions—an intention-provided price where present, otherwise the fixed resolution rule applied to the current market state—and the rolling procedure never creates a new price level; residuals always migrate toward more conservative prices, and within each limit-order rolling chain the roll itself is lossless, the only discarded quantity being the terminal sub-lot remainder (c_{n_B+1} cash or r_{n_S+1} shares). Since every step rounds down, issued buy notional never exceeds the total weighted buy notional and issued sell quantity never exceeds the total weighted sell quantity, and the procedure reduces to per-tier rounding when all intentions share one price. Market orders carry no price tier and are rounded down once. Aggregated physical orders retain the identities of their contributing strategy slots for provenance and diagnostics; this metadata does not alter matching. The broker then applies the authoritative checks—cash, fees, inventory, reservations, and the market rules of Appendix E—and

admitted orders enter the limit order book, where matching updates both the market state and the account state observed by later decisions.

All initialization completes before the first strategy call; within each simulation path the decoded population is fixed while accounts and market state evolve endogenously.

C Strategy Templates in the Reported Population

The reported population instantiates seven strategy templates— Fundamentalist, Market Maker, Momentum, Noise, Execution, Graph Signal, and LLM. A template is a reusable decision mechanism rather than a trader type: several templates may be active within one trader, their intentions combined through the strategy-influence block, and the population hierarchy gains no further level from them. For each active trader–asset–strategy slot (i, a, s), the template maps its admissible market and account view into raw order or cancellation intentions. The strategy-influence block then combines these intentions, and the common account, aggregation, and market interfaces of Appendix B determine which executable orders reach the limit-order book.

C.1 Shared decision conventions

All seven templates obey a common decision contract, and three distinctions apply to every per-template rule below.

Intentions, not fills. A template returns zero or more raw order or cancellation intentions: a buy or sell intention carries the symbol, side, target notional, a price indication, and an optional rationale; a cancellation carries the identity of one of the trader's own active orders. Downstream, the strategy-influence block and the planner decide which intentions become submittable orders under

Table C.1: Seven strategy templates in the reported population. Controls are strategy-local only where declared in the typed parameter schema; runtime directives and external services belong to the fixed environment.

Template	Economic mechanism	Main strategy-local controls
Fundamentalist	Mean-reverting demand activated by the gap between the quoted price and a fixed or drifting perceived fair value.	Entry and aggressiveness thresholds, fair-value evolution, size bounds, and stale-order cancellation.
Market Maker	Multi-level two-sided liquidity provision around an inventory-adjusted reservation price, with spreads widened by recent volatility.	Minimum spread, inventory skew, quote levels and spacing, size decay, and refresh probability.
Momentum	State-conditioned trend following: a recent up, down, or flat price state changes the probabilities of buying, selling, or abstaining.	Lookback, state-specific action weights and price offsets, size distribution, and cancellation probability.
Noise	Uninformed stochastic order flow, optionally tilted by a short moving-average signal, with one or more same-side proposals per decision.	Buy/sell/hold weights, size distribution, price jitter, burst count, and cancellation probability.
Execution	Schedule-based execution of a runtime-supplied parent order through passive posting and aggressive catch-up.	Passive and aggressive phase lengths, slice sizes, aggressive offset, and post-deadline catch-up bound.
Graph Signal	Network-informed trading driven by neighbors' current returns in a graph constructed from lagged public simulated returns.	Correlation window and edge threshold, action threshold, capacity fraction, and signal-age bound.
LLM	A validated structured decision generated from a strategy-safe current context through an injected inference service; invalid or failed responses produce abstention.	Declared response validation, call-budget, and order-cap controls; model and service configuration are frozen externally.

the declared influence weights and visible capacity, and the market decides acceptance and fills. "Buys" below therefore means "emits a buy intention," and neither an empty frame nor an explicit abstention produces an order.

Asset-level capacity, no strategy sub-accounts. Templates read the trader's asset-level buy budget and the asset's total or sellable holdings. There are no per-strategy cash or holdings accounts, and no template multiplies its sizes by π_2 or π_3: influence-weight combination happens once, downstream, under the aggregation rules of Appendix B.

Prices, ticks, and reference resolution. Prices are integer cents with tick size τ, and a non-negative width β in basis points at reference price p converts to ticks as

$$T(p,\beta) = \max\left\{1,\ \mathrm{round}\left(\frac{p\,\beta}{10^4\tau}\right)\right\}, \tag{C.1}$$

so the same basis-point width need not map to the same integer tick count at different price levels, and low-priced assets feel the one-tick floor. The reference price is resolved, in order, from the current mark, recent price history, and the previous close of the public price state, then the book's last price, and finally an injected base price; best quotes come from the book's level analysis, with a synthetic bid–ask pair constructed around the reference when no valid two-sided quote exists.

Capacity scaling. The raw buy capacity at price p under visible budget B is $C = \lfloor B/p \rfloor$; the raw sell capacity is the non-negative sellable share count. Noise, Momentum, and Fundamentalist share one scaling rule that maps an intensity draw onto this capacity. Given capacity C, an optional hard cap L, an intensity numerator n with denominator $d \geq 1$, and a thin-order threshold c:

$$\bar{C} = \begin{cases} \min(C, L) & \text{if } L \text{ is declared,} \\ C & \text{otherwise,} \end{cases} \qquad q_0 = \left\lfloor \bar{C}\,\frac{\min(n,d)}{d} \right\rfloor,$$
$$q = \begin{cases} 0, & \bar{C} \leq 0 \text{ or } n \leq 0, \\ \bar{C}, & c > 0 \text{ and } q_0 < \min(c, \bar{C}), \\ q_0, & \text{otherwise.} \end{cases} \tag{C.2}$$

The middle branch consolidates thin orders: when the scaled size would be smaller than both the threshold and the capacity, the full capacity is used instead of shredding the intention. A drawn intensity is therefore a fraction of visible capacity, not an absolute share count, and only a call that declares a level cap imposes an absolute ceiling.

C.2 Noise

The Noise template supplies capacity-constrained stochastic liquidity demand (Gode & Sunder, 1993): it is not an unconstrained coin flip, and not a fixed-size zero-intelligence order generator. Each call reseeds a local deterministic 64-bit generator from the declared seed plus a persisted per-slot step counter, so the global random state is never touched and the sequence is reproducible across pause and resume.

A call proceeds in a fixed order. First, if a cancellation probability is declared and active orders exist, one active order is canceled with that probability and the call ends. Otherwise the template draws a side from the normalized non-negative weights (w_b, w_s, w_h), $P(b) = w_b/(w_b + w_s + w_h)$ and analogously; a zero total falls back to buy and sell at one half each with no abstention. An optional moving-average tilt, active only when its switch is enabled and the feature is present, shifts the weights by $\delta = 0.2 \times \texttt{analysis_bias_strength}$: when $p > \mathrm{MA5}$, $w_b \leftarrow \min(0.9,\ w_b + \delta)$ and $w_s \leftarrow \max(0,\ w_s - \delta)$, and mirror-symmetrically when $p < \mathrm{MA5}$; the hold weight is unchanged and the weights are re-normalized before the draw. A hold draw returns an empty frame.

On a buy or sell draw, the template emits a burst of same-side proposals. Each proposal draws its intensity n from the declared discrete volume distribution, or uniformly from the integer interval $[\texttt{min_volume}, \texttt{max_volume}]$ when none is declared, with denominator d equal to the largest choice key or to `max_volume` respectively; draws a price perturbation

$$J \sim U_{\mathbb{Z}}\big[-T(p,\beta_{\text{jitter}}),\ T(p,\beta_{\text{jitter}})\big], \qquad p' = \max(1,\ p + J\tau), \tag{C.3}$$

and sizes the proposal through (C.2) with the consolidation threshold equal to the intensity denominator and no absolute level cap. Capacity is debited within the call, so successive proposals in one burst see the remaining cash or holdings.

C.3 MOMENTUM

The Momentum template is a randomized trend follower (Lux & Marchesi, 1999; Chiarella et al., 2009), not a deterministic "up implies buy" rule. Given a lookback of k price-history samples, it classifies the market state as

$$s_t = \begin{cases} \text{up} & p_t > p_{t-k}, \\ \text{down} & p_t < p_{t-k}, \\ \text{flat} & p_t = p_{t-k}, \end{cases} \tag{C.4}$$

falling back to flat when fewer than $k+1$ usable prices exist. The lookback counts samples, not seconds, and no return magnitude, slope, or oscillator is computed: only the three-way state shapes the decision distributions.

A random cancellation of one active order is attempted first, as in Noise. Otherwise the state-conditioned buy and sell weights are normalized together with the hold weight—for example, the default up-state weights $(0.65, 0.20, 0.10)$ give a realized buy probability of $0.65/0.95$—and the side is drawn categorically; a zero total again falls back to buying and selling at one half each. A hold draw returns an empty frame.

A non-hold draw produces at most one intention. Its price offset is drawn from the state- and side-specific basis-point interval: both endpoints are converted to signed ticks, one integer offset o is drawn uniformly from the inclusive tick interval, and the quote is $p + o\tau$; sell-side intervals are already signed and are not negated again. When no matching interval is declared, a symmetric jitter of half-width $T(p, \texttt{price_jitter_bp})$ is used, and the offset is zero when that is undeclared as well. Size is drawn and scaled through (C.2) under the same rules as Noise. The per-slot state is the seed and step counter; price history itself is part of the shared market view.

C.4 MARKET MAKER

The Market Maker template posts a multi-level bid–ask ladder around an inventory-adjusted reservation price (Avellaneda & Stoikov, 2008). It is a rule-based quoting mechanism, not the closed-form solution of an optimal-control model.

Reservation price. When the best quotes satisfy $a > b$ the mid is $m = \lfloor (a+b)/2 \rfloor$; otherwise the reference price is used. On its first computation the template records the total holdings I_0 as the inventory anchor, and later deviation is measured against total holdings I, not the sellable subset. With inventory tolerance κ_I and lot size ℓ, the inventory scale is

$$S_I = \begin{cases} \max\big(\ell,\ \lfloor (I_0 \kappa_I + 10000)/20000 \rfloor\big), & I_0 > 0, \\ \max(\ell, \kappa_I), & I_0 \le 0, \end{cases} \tag{C.5}$$

and with K_{sk} the maximum skew in ticks (converted from `max_inventory_skew_bp` at the mid) and risk aversion γ, the inventory skew in ticks is

$$k_I = \text{clip}\left(\text{round}_{\text{ha}}\left(\frac{I - I_0}{S_I}\, \gamma K_{\text{sk}} \right),\ -K_{\text{sk}},\ K_{\text{sk}} \right), \tag{C.6}$$

where round_{ha} rounds halves away from zero. Long inventory shifts the reservation price down and short inventory shifts it up.

Half-spread and ladder. Short-window variation is the mean absolute price move, not a return standard deviation: $v = \lceil \text{mean}_j\, |p_j - p_{j-1}| / \tau \rceil$, and zero with fewer than two usable prices. The half-spread in ticks is

$$H_{\text{spr}} = \max\left(T(m, \beta_{\min}),\ \left\lceil \frac{\max(1, a-b)}{2\tau} \right\rceil \right) + \lceil \lambda v \rceil, \tag{C.7}$$

combining the declared floor, the prevailing book spread, and the volatility widening with factor λ. The reservation price is $r = m - k_I\tau + k_{\text{bias}}\tau$, and the innermost quotes are

$$b_0 = \min(b,\ r - H_{\text{spr}}\tau), \qquad a_0 = \max(a,\ r + H_{\text{spr}}\tau), \tag{C.8}$$

with level j shifted a further $j \times$ `level_spacing` ticks outward. Quotes are clipped to the daily price range and a level is skipped when its ask no longer exceeds its bid; the ladder does not deliberately cross the opposite best quote.

Sizes and refresh. Buy-side capacity is priced conservatively at the highest bid, and sell-side capacity is the sellable share count. Levels receive weights $w_j =$ `size_decay`^{j}, are floored to whole shares, capped per level at `max_volume`, and leftover shares are distributed from the innermost level outward; if the innermost level would be thinner than `base_volume`, all size consolidates onto it. Active orders whose prices are absent from the new target set, or that are drawn for refresh with `refresh_probability`, are canceled first and the call ends; missing levels are posted on subsequent calls, and an existing same-side same-price order suppresses a duplicate. The per-slot state is the inventory anchor and the private generator used for refresh draws.

C.5 Fundamentalist

The Fundamentalist template supplies stabilizing value demand (Brock & Hommes, 1998; Lux & Marchesi, 1999): it buys when its perceived fair value is sufficiently above the quoted reference and sells when sufficiently below, with stronger mispricing permitting more aggressive prices and larger orders.

Fair value. The internal fair value F is initialized from the declared fixed value when one exists, else from the injected base price under the default anchor, else from the first reference price. Whenever at least `fair_value_update_seconds` of simulation time have elapsed since the last update, F takes one integer-tick drift step,

$$D \sim U_{\mathbb{Z}}\big[-\bar{T}(F,\beta_d),\ \bar{T}(F,\beta_d)\big], \qquad F \leftarrow \max(1,\ F + D\tau), \tag{C.9}$$

at most once per decision call, with no catch-up for skipped intervals. Here $\bar{T}$ is the conversion of (C.1) without the one-tick floor, so that a zero amplitude disables drift. A declared fixed value pins only the initial anchor: under a nonzero drift amplitude β_d, F continues to evolve. F is an internal valuation produced by initialization and bounded stochastic drift; it does not read earnings, dividends, or any exogenous fundamental.

Entry and pricing. With reference price p, mispricing $d = F - p$, and entry threshold $E_{\text{ent}} = \tau\, T(p,\beta_e)$, a call with $|d| < E_{\text{ent}}$ opens nothing and instead cancels the asset's active orders with `stale_cancel_probability`; otherwise $d > 0$ buys and $d < 0$ sells, with exact equality admitting entry. Aggressiveness follows the mispricing magnitude: with $A = \max\big(T(p,\beta_e),\ T(p,\beta_a)\big)$ the coarser of the entry and aggressiveness scales and $K_{\text{cap}} = T(p,\beta_{\max})$ the capped offset,

$$g = \max\big(\tau,\ \lfloor A\tau/2 \rfloor\big), \qquad k = \min\big(K_{\text{cap}},\ \lfloor |d|/g \rfloor\big). \tag{C.10}$$

When $k > 0$ the template buys at $\text{ask}+k\tau$ and sells at $\max(1,\ \text{bid}-k\tau)$; otherwise it posts passively at the best bid or ask. Quotes are clipped to the daily range and checked against the market's price rules.

Size and order management. The mispricing strength multiple is $r = \max\big(1, \lfloor |d| / \max(\tau, E_{\text{ent}}) \rfloor\big)$, giving intensity numerator $n = \min(V_{\max}, V_{\text{base}}\, r)$, scaled through (C.2) with denominator and level cap $V_{\max}$ and consolidation threshold V_{base}. An active order on the opposite side, or off the new target price by more than one tick, is canceled first and the call returns; a compatible active order suppresses a duplicate. The per-slot state is the current F, the last update time, and the private generator.

C.6 Execution

The Execution template works a parent instruction—side, target quantity, and time window—supplied at runtime by the environment, in the spirit of schedule-based execution (Almgren & Chriss,

2001). It generates no alpha signal of its own, and the instruction and the account's initial holdings contribute no coordinates to the behavioral block; only the scheduling controls below are strategy-local.

Progress is estimated from the shared account's net inventory change,

$$Q_{\text{done}} = \begin{cases} \max(0,\, Y_t - Y_0), & \text{buy}, \\ \max(0,\, Y_0 - Y_t), & \text{sell}, \end{cases} \tag{C.11}$$

not from a private fill ledger: other strategies trading the same asset advance or regress the estimate. Before the start time the template is inactive; once Q_{done} reaches the target it cancels its same-side active orders and stops.

Between start and end, calls alternate between passive and aggressive phases with cycle length $T_c = T_p + T_a$. During a passive phase ($\text{elapsed} \bmod T_c < T_p$) the template rests at the best bid (buy) or best ask (sell), maintaining in-flight volume of $\min(\texttt{passive_buffer_volume},\ \text{remaining target})$: off-price orders are canceled first, same-price orders are topped up to the target within visible capacity. During an aggressive phase, with horizon $T_{\text{win}} = \max(1,\, \text{end} - \text{start})$, the linear schedule and the backlog are

$$Q_{\text{exp}} = \text{RoundToLot}\left(\left\lfloor \frac{Q_{\text{target}}\ \text{elapsed}}{T_{\text{win}}} \right\rfloor\right), \qquad \text{backlog} = \max\bigl(0,\ Q_{\text{exp}} - Q_{\text{done}}\bigr). \tag{C.12}$$

A zero backlog cancels same-side active orders; otherwise aggressive in-flight volume is capped by the minimum of $\texttt{aggressive_slice_volume}$, the remaining target, and the backlog, priced at $\text{ask} + T(\text{ask}, \beta_a)\tau$ for buys and $\max\bigl(1,\ \text{bid} - T(\text{bid}, \beta_a)\tau\bigr)$ for sells. After the deadline the template keeps a bounded catch-up in-flight volume of the minimum of the remaining target and $\texttt{aggressive_slice_volume} \times \texttt{max_catchup_slices_after_end}$; the multiplier caps each computation's catch-up size and is not a cumulative counter of post-deadline calls. Internally the template rounds to whole lots with a 100-share default, earlier than the other templates; final lot and capacity enforcement still happens in the common order chain.

C.7 Graph Signal

The Graph Signal template is a network-informed chartist (Cont & Bouchaud, 2000; Yang et al., 2025): it follows the correlation-weighted current returns of positively correlated neighboring assets. The graph is built from the simulation's own public price marks—no industry relations, external knowledge, knowledge-graph service, or trained model—by a shared runtime that belongs to the fixed environment; the template itself only consumes the frozen signal frame.

Graph construction. Nodes are the traded assets in a fixed canonical order. At each synchronized sampling boundary, public marks give single-interval log returns $r_{i,t} = \log\bigl(p_{i,t}/p_{i,t-1}\bigr)$, formed only when both adjacent samples are valid; a missing observation occupies its slot, and no return is ever stitched across a gap. The graph at boundary t is estimated from the previous w aligned returns $W_{t-1} = \{t-w, \ldots, t-1\}$—excluding the current interval—as the Pearson correlation of the demeaned columns,

$$\rho_{ij,t} = \frac{\sum_{k \in W_{t-1}} (r_{i,k} - \bar{r}_i)(r_{j,k} - \bar{r}_j)}{\sqrt{\sum_{k \in W_{t-1}} (r_{i,k} - \bar{r}_i)^2 \ \sum_{k \in W_{t-1}} (r_{j,k} - \bar{r}_j)^2}}. \tag{C.13}$$

The edge weight $a_{ij,t}$ equals $\rho_{ij,t}$ when the pair has a complete aligned window, both variances are positive, and $\rho_{ij,t} \geq \rho_{\min} > 0$, and equals zero otherwise: no self-loops, no negative-correlation edges, and an empty graph is a legitimate state. A constant-price asset therefore creates no edges, rather than a spurious perfect correlation.

Signal and threshold. With the eligible neighborhood $J_i(t) = \{j :\ a_{ij,t} > 0 \text{ and } r_{j,t} \text{ valid}\}$, the signal is the correlation-weighted current neighbor return

$$g_{i,t} = \frac{\sum_{j \in J_i(t)} a_{ij,t}\, r_{j,t}}{\sum_{j \in J_i(t)} a_{ij,t}}, \tag{C.14}$$

and the threshold $\eta = \texttt{threshold_bp}/10^4$ gives a buy intention when $g_{i,t} > \eta$, a sell intention when $g_{i,t} < -\eta$, and abstention otherwise—exact equality does not trade. An asset whose own current return or reference price is invalid, a warm-up window, and an empty neighborhood all produce typed abstentions. The rule is neighbor-momentum propagation, not residual mean reversion against the neighborhood.

Causal order and lifecycle. Each boundary is processed in a fixed order: the market completes previously submitted actions; marks are sampled and current returns computed; the graph is estimated from the old window only; signals are computed from the current returns; the immutable frame is published; and only then are the current returns appended to the window. The same interval is therefore never used both to select neighbors and to score them. A day change resets the window and the price anchors; a phase transition establishes new anchors without appending a return and without emitting a signal, so no cross-phase return is ever fabricated. Signals are consumed only while their simulation-time age stays within the fixed freshness bound `signal_max_age_sim_seconds`, a graph-execution-policy setting of the environment (Appendix J.1).

Materialization. On an active signal, the template reads the current mark. An existing same-side order of its own suppresses a duplicate; when its own active orders are all on the opposite side they are canceled first, and no reversed order is posted in the same call; orders shared with other origins block the action. A buy intention targets a notional of `order_fraction` times the asset's available cash, and a sell intention the same fraction of sellable shares times the reference price; no π_2 or π_3 factor is applied at this layer, and lot rounding is left to the common chain. The correlation window w and the edge threshold $\rho_{\min}$ parameterize the single shared graph policy of a simulation and are identical for all graph slots of a candidate; the action threshold and the capacity fraction are slot-local.

C.8 LLM

The LLM template maps a strategy-safe view of the current market and the trader's own account into validated structured intentions through an injected inference port (Zhang et al., 2024; Xiao et al., 2024). Unlike the six explicit-rule templates, its decision function has no closed form; the call pipeline is

$$y_t = \mathrm{Infer}_{\mathcal{M},\lambda}\big(\mathrm{Render}(P, \mathcal{C}_t)\big), \qquad D_t = \mathrm{ValidateAndMaterialize}(y_t), \tag{C.15}$$

where $\mathcal{C}_t$ is the frozen context, P the declared prompt template, and $(\mathcal{M}, \lambda)$ the externally resolved model and inference settings. The kernel performs no online training, gradient update, or reinforcement learning. The frozen context contains four summary blocks—market, account, strategy scope, and active orders—and nothing else: the template does not read unbounded simulator internals, and a missing inference port is a hard error, not a silent fallback to random trading.

Budget and validation. Each request is bound to its trader, slot, tick, and snapshot identity, and a per-slot preparation counter enforces `max_inferences_per_run` when declared; an exhausted budget is an explicit abstention. Validation is fail-closed: a non-success inference status or an empty structured output abstains; the top-level object must be exactly one `intents` array, and an empty array is an explicit abstention; every item must be a non-empty mapping with side in buy, sell, or an abstention alias—cancellation is not an accepted action—with confidence in $[0, 1]$ and all numeric fields finite and in range; an intention notional is preferred, a quantity–price pair is the fallback, and a missing positive amount rejects the item; and one malformed item invalidates the entire result rather than silently keeping the rest.

Pricing, caps, and asynchrony. A model-supplied price must parse to a positive integer number of cents. For a notional-only intention the kernel falls back to the frozen quotes—best bid for sells, best ask for buys, then the current mark, then the previous close. When `max_buy_cash_fraction` $= \alpha$ is declared, each buy intention is capped individually at α times the account cash visible at preparation; the cap is per intention, not a cumulative budget across the intentions of one response. Under the default deferred mode the owner freezes the request, the quotes, and the optional cap, a provider worker performs the inference, and the completed result is materialized at consumption

under the live lifecycle and capacity checks; nothing is injected back into the frozen request, and a successful inference never skips the downstream capacity and validity chain.

Boundary between strategy templates and the environment. Graph Signal and LLM are templates of the fixed catalogue: they occupy slots in $\mathcal{S}_{ia}$, receive influence through $\pi_{3,ia}$, and expose only the local controls declared in θ_{ias}. The systems that produce their context—the graph-frame construction and its update rule, and the LLM model, prompt assets, provider, inference service, credentials, and service-side state—belong to the fixed environment κ. These systems are held identical across candidate evaluations and contribute no search coordinates, so graph- or language-conditioned orders affect the endogenous market trajectory while the population under calibration remains specified entirely by its own coordinates.

C.9 STRATEGY-LOCAL PARAMETER SCHEMA

Table C.2 lists the strategy-local parameter names behind the behavioral block θ_{ias}: items 1–52 are the numeric fields declared search-eligible in the current catalogue metadata. Items 50–51, marked ‡, are two candidate-level graph-policy scalars, denoted θ_p^{graph} to distinguish them from the per-slot behavioral fields: one value of each is broadcast identically to all graph slots of one candidate rather than replicated per trader–asset slot. The count of fifty-two ranges over strategy–field pairs: the same short name in two strategies denotes two different parameters, and each strategy's fields are replicated across its active slots in the θ tensor, so the count is not the dimension of the behavioral block. Defaults are catalogue template values, not calibrated winners, and the ranges reproduce the declared search metadata. Probability-named fields are relative weights, normalized at decision time, and carry no sum-to-one constraint.

Table C.2: Strategy-local parameters: names, meanings, catalogue defaults, and search ranges. ‡: candidate-level shared graph-policy parameter.

#	Parameter	Meaning	Default	Range
Noise (8 fields)				
1	`buy_probability`	Relative buy weight before normalization; not a final standalone buy probability	0.5	$[0, 1]$
2	`sell_probability`	Relative sell weight	0.5	$[0, 1]$
3	`hold_probability`	Relative weight of abstaining this call	0.0	$[0, 0.5]$
4	`min_volume`	Lower bound of the uniform intensity draw when no discrete volume distribution is declared; mapped onto capacity	100	$[100, 500]$
5	`max_volume`	Upper bound of that draw; $\geq$ `min_volume`	200	$[200, 2000]$
6	`price_jitter_bp`	Two-sided random quote perturbation, converted to ticks	4.0	$[0, 20]$ bp
7	`cancel_probability`	Probability of canceling one active order before proposing	0.03	$[0, 0.3]$
8	`analysis_bias_strength`	Strength of the optional MA5 tilt on the buy/sell weights; inert unless its switch and feature are present	0.2	$[0, 1]$
Momentum (12 fields)				
9	`lookback_steps`	Samples between the current and compared price in the state rule (C.4)	3	$[1, 20]$ samples
10	`up_buy_prob`	Relative buy weight in the up state	0.65	$[0.4, 0.9]$
11	`down_sell_prob`	Relative sell weight in the down state	0.65	$[0.4, 0.9]$
12	`up_sell_prob`	Relative sell weight in the up state	0.2	$[0.05, 0.4]$
13	`down_buy_prob`	Relative buy weight in the down state	0.2	$[0.05, 0.4]$
14	`flat_buy_prob`	Relative buy weight in the flat state, including insufficient history	0.45	$[0.2, 0.6]$
15	`flat_sell_prob`	Relative sell weight in the flat state	0.45	$[0.2, 0.6]$
16	`hold_probability`	Relative abstain weight, normalized jointly with the state's buy/sell weights	0.1	$[0, 0.4]$
17	`min_volume`	Lower bound of the uniform intensity draw when no discrete volume distribution is declared	100	$[100, 500]$
18	`max_volume`	Upper bound of that draw; $\geq$ `min_volume`	500	$[500, 2000]$
19	`price_jitter_bp`	Symmetric fallback offset when no state-specific interval applies	4.0	$[0, 12]$ bp
20	`cancel_probability`	Probability of canceling one active order before proposing	0.08	$[0, 0.3]$
Market Maker (13 fields)				
21	`min_half_spread_bp`	Half-spread floor at the mid	4.0	$[1, 20]$ bp
22	`num_levels`	Quote levels maintained per side	2	$[1, 5]$
23	`level_spacing_bp`	Spacing between adjacent levels	8.0	$[2, 24]$ bp
24	`base_volume`	Consolidation threshold when the innermost level would be thin; not a fixed per-level size	200	$[100, 1000]$ sh

continued on next page

Table C.2 (continued)

#	Parameter	Meaning	Default	Range
25	max_volume	Per-level raw share cap; $\geq$ base_volume	400	[200, 50000] sh
26	size_decay	Level-to-level size decay factor	0.75	$[0.4, 0.95]$
27	volatility_lookback	Window of the mean absolute price move; not a standard-deviation window	20	[5, 60] samples
28	volatility_spread_factor	Multiplier mapping recent moves to spread widening	1.5	$[0.5, 3]$
29	inventory_tolerance	Reference scale of the inventory skew	20000	[5000, 50000]
30	max_inventory_skew_bp	Cap on the inventory-driven reservation shift	4.0	[0, 20] bp
31	inventory_risk_aversion	Amplification of relative inventory pressure in (C.6)	1.0	$[0.2, 3]$
32	reservation_bias_bp	Static reservation-price bias; positive shifts up	0.0	$[-8, 8]$ bp
33	refresh_probability	Probability of refreshing an otherwise valid quote	0.02	$[0, 0.2]$
Fundamentalist (8 fields)				
34	entry_threshold_bp	Mispricing required before a trade target exists	4.0	[1, 20] bp
35	aggressive_threshold_bp	Aggressiveness scale in (C.10), not a binary switch; $\geq$ entry_threshold_bp + 2	10.0	[4, 40] bp
36	max_aggressive_bp	Cap on price improvement beyond the best quote	4.0	[1, 20] bp
37	base_volume	Base role size and consolidation threshold; not a fixed final size	200	[100, 500] sh
38	max_volume	Scaled raw share cap; $\geq$ base_volume	50000	[500, 50000] sh
39	fair_value_update_seconds	Minimum spacing of fair-value drift updates	60	[10, 300] s
40	fair_value_drift_bp	Amplitude of each fair-value drift; zero disables drift	2.0	[0, 10] bp
41	stale_cancel_probability	Cancel probability while no entry condition holds	0.2	$[0, 0.4]$
Execution (6 fields)				
42	passive_buffer_volume	Resting in-flight cap during passive phases	500	[100, 1000] sh
43	aggressive_slice_volume	Per-slice cap during aggressive phases	700	[100, 1500] sh
44	passive_seconds	Passive phase length	40	[10, 120] s
45	aggressive_seconds	Aggressive phase length	20	[5, 60] s
46	aggressive_offset_bp	Aggressive offset from the opposite best quote	4.0	[1, 20] bp
47	max_catchup_slices_after_end	Post-deadline in-flight multiple of the slice volume; not a cumulative call counter	3	[1, 10]
Graph Signal (4 fields)				
48	threshold_bp	Buy/sell trigger on the weighted neighbor return in (C.14)	1.0	[0.1, 20] bp
49	order_fraction	Fraction of visible capacity targeted per action; downstream π_3 scaling still applies	1.0	$[0.1, 1]$
50	correlation_window_samples‡	Aligned return intervals behind each graph estimate, excluding the signal interval	20	[5, 60] samples
51	minimum_correlation‡	Minimum correlation retaining an edge	0.30	$[0.05, 0.95]$
LLM (1 field)				
52	max_buy_cash_fraction	Per-intention buy-notional cap as a fraction of available cash; not a cumulative budget and no constraint on sells	1	$[0.1, 1]$

Four qualifications apply to the schema. First, the joint constraints max_volume $\geq$ min_volume (Noise, Momentum), max_volume $\geq$ base_volume (Market Maker, Fundamentalist), and aggressive_threshold_bp $\geq$ entry_threshold_bp + 2 define the feasible region; candidates are generated jointly against these constraints, never field by field. Second, activation conditions can mask a field: a declared discrete volume distribution overrides min_volume/max_volume in Noise and Momentum, state-specific offsets override price_jitter_bp in Momentum, and analysis_bias_strength is inert while its switch is off; search outcomes over masked fields must not be read as identification. Third, the two graph-policy scalars θ_p^{graph} are shared by all graph slots of a candidate—a simulation rejects divergent graph policies across its slots. Fourth, ranges do not imply identifiability: order_fraction overlaps the influence-weight strength of π_3, and tick, lot, and timing discretization can map distinct parameter values to identical realized behavior.

D Strategy-External Information System

The Strategy-External Information System (SEIS) is part of the forward simulator specified by the environment κ. It maintains a compact lagged information state, constructs a dynamic trader-interaction graph, and uses the resulting belief to form a bounded transient strategy mixture $\widetilde{\pi}_{3,ia}(t)$ before the order pipeline of Appendix B. The population-level mixture $\pi_{3,ia}$ remains the structural anchor and is not modified during simulation.

D.1 SCOPE: GENERAL ARCHITECTURE AND CURRENT REALIZATION

In general, the information state maintained by SEIS is $\mathbb{I}_t = \mathbb{I}_t^{\text{endo}} \cup \mathbb{I}_t^{\text{exo}}$: endogenous information generated by the simulation itself—trader behavior, beliefs, prices, and the realized interaction structure—and exogenous information that may be supplied from outside the simulation, such as news, macroeconomic events, social media, external agent worlds, or manually scripted scenarios. The architecture admits two information channels: a *local channel*, through which each trader reads a sparse time-varying neighborhood of peers, and a *public channel*, through which all traders receive one shared market interpretation.

The implemented system realizes both channels. The local channel maintains compressed lagged behavior and a dynamic trader graph carrying belief diffusion, as specified in the remainder of this appendix. The public channel maintains a single shared large-language-model market summary: a bounded market summary and the published beliefs are condensed into one request issued in the background under a bounded budget, validated responses enter as a bounded, decaying public correction to published beliefs, and the corrected beliefs can only attenuate intentions that oppose them before the standard order pipeline. The public channel holds no funds, submits no orders, and reads no account balances.

The experiments reported in this paper instantiate the endogenous local channel only: $\mathbb{I}_t^{\text{exo}} = \varnothing$, and no external or language-model information source is active. The remainder of this appendix specifies this realization. The mechanism selected for a run and its associated parameters are components of κ, declared ex ante, rather than coordinates of the search space Z, and are never adapted using calibration or held-out loss. The realization specified below instantiates the local channel with a Top-K relation operator, lagged-behavior diffusion, and a bounded sub-simplex projection, with associated parameters K, ρ_{seis}, λ_{seis}, $\delta_{\max}$, and η_π; other relation operators and integration rules remain compatible with the architecture. The interaction rule is fixed ex ante, while the realized graph at each tick is deterministically induced from lagged endogenous simulation states—fixed rule, endogenous realization.

SEIS is distinct from the Graph Signal and LLM strategy templates of Appendix C: those are strategy-internal decision rules whose outputs enter the intention pipeline, whereas SEIS operates outside individual templates by modifying the runtime strategy mixture.

D.2 RUNTIME STATE AND TICK CAUSALITY

The realization maintains two arrays,

$$U_t^{\text{beh}}, B_t^{\text{seis}} \in [-1, 1]^{N \times A}, \qquad U_0^{\text{beh}} = B_0^{\text{seis}} = 0, \tag{D.1}$$

where the entries $U_{ia}^{\text{beh}}(t)$ form compressed baseline intent behavior and $B_{ia}^{\text{seis}}(t)$ form the strategy-external belief. At the beginning of tick t, strategy-internal computation and the information update share frozen tick inputs but have no current-tick data dependency on one another: the strategy branch computes and validates current contributions (Appendix B.2), while the information update reads only U_{t-1}^{beh} and B_{t-1}^{seis}. The completed branches join before the runtime projection. Current validated contributions subsequently produce U_t^{beh}, which can affect B_{t+1}^{seis} but not B_t^{seis}:

$$\mathcal{O}_t^{\text{val}} \longrightarrow U_t^{\text{beh}} \longrightarrow B_{t+1}^{\text{seis}}. \tag{D.2}$$

The dependency in (D.2) is the causality requirement; the two branches may execute concurrently or serially provided it is preserved. The initialization in (D.1) makes the first tick a natural warm-up, with $B_1^{\text{seis}} = 0$.

For trader i and asset a, let $M_{ias}^{\text{act,buy}}(t)$ and $Q_{ias}^{\text{act,sell}}(t)$ denote the actionable buy notional and sell quantity of strategy slot s in the validated contribution collection. Their baseline aggregates use the structural mixture,

$$M_{ia}^{\text{base,buy}}(t) = \sum_s \pi_{3,ias} M_{ias}^{\text{act,buy}}(t), \qquad Q_{ia}^{\text{base,sell}}(t) = \sum_s \pi_{3,ias} Q_{ias}^{\text{act,sell}}(t). \tag{D.3}$$

Using the same pre-matching capacity snapshot that validated the contributions, define

$$U_{ia}^{\mathrm{beh}}(t) = \mathrm{clip}\Big[\rho_c\Big(M_{ia}^{\mathrm{base,buy}}(t), C_{ia}^{\mathrm{buy}}(t)\Big) - \rho_c\Big(Q_{ia}^{\mathrm{base,sell}}(t), Q_{ia}^{\mathrm{sellable}}(t)\Big), -1, 1\Big], \tag{D.4}$$

where $\rho_c(x, c) = 0$ for $c \le 0$ and x/c otherwise. Cancellations, holds, and invalid contributions do not enter (D.4). The capacities $C_{ia}^{\mathrm{buy}}(t)$ and $Q_{ia}^{\mathrm{sellable}}(t)$ are the available cash and sellable holdings of Appendix B.2 (C_{ia}^{avail}, Q_{ia}^{sell}), read from the pre-matching snapshot.

D.3 Trader graph and belief diffusion

For the Top-K relation operator used in this realization, one trader graph is constructed per tick from the complete cross-asset rows of U_{t-1}^{beh}. For $i \neq j$, the normalized behavioral distance is

$$d_{ij}(t) = \frac{\|U_{i,:}^{\mathrm{beh}}(t-1) - U_{j,:}^{\mathrm{beh}}(t-1)\|_2}{2\sqrt{A}}, \qquad 0 \le d_{ij}(t) \le 1. \tag{D.5}$$

Let $\tau_i(t)$ be the K-th smallest distance from trader i to the other traders. The tie-inclusive neighborhood is

$$\mathcal{N}_i(t) = \{j \neq i : d_{ij}(t) \le \tau_i(t)\}. \tag{D.6}$$

All peers tied at the boundary are included, so $|\mathcal{N}_i(t)| \ge K$; the numerical profile is recorded in Appendix J.1. Within the neighborhood, define the affinity and row-stochastic peer matrix by

$$w_{ij}(t) = e^{-d_{ij}(t)}, \qquad P_{ij}^{\mathrm{peer}}(t) = \frac{w_{ij}(t)}{\sum_{k \in \mathcal{N}_i(t)} w_{ik}(t)}, \tag{D.7}$$

with $w_{ij} = P_{ij}^{\mathrm{peer}} = 0$ outside the neighborhood and on the diagonal. $P_{ij}^{\mathrm{peer}} > 0$ means that trader i reads trader j, so the canonical information-flow edge is $j \to i$; row-specific boundaries and normalization need not produce a symmetric graph. The same $P_t^{\mathrm{peer}} \in \mathbb{R}^{N \times N}$ acts on all asset columns.

For a trader permutation matrix Π, the graph operator is permutation equivariant,

$$\mathcal{G}_{\mathrm{peer}}(\Pi U^{\mathrm{beh}}) = \Pi \mathcal{G}_{\mathrm{peer}}(U^{\mathrm{beh}})\Pi^{\top}, \tag{D.8}$$

since pairwise distances and each row's order statistic are unchanged by a consistent relabeling and every boundary tie is included. If belief and strategy inputs are permuted consistently, diffusion and cell-wise projection permute in the same way.

The dynamic graph carries both newly injected lagged behavior and previously propagated belief,

$$S_t^{\mathrm{src}} = P_t^{\mathrm{peer}} U_{t-1}^{\mathrm{beh}}, \qquad B_t^{\mathrm{prop}} = P_t^{\mathrm{peer}} B_{t-1}^{\mathrm{seis}}. \tag{D.9}$$

For retention $\rho_{\mathrm{seis}} \in [0, 1]$, the current belief is

$$B_t^{\mathrm{seis}} = (1 - \rho_{\mathrm{seis}}) P_t^{\mathrm{peer}} U_{t-1}^{\mathrm{beh}} + \rho_{\mathrm{seis}} P_t^{\mathrm{peer}} B_{t-1}^{\mathrm{seis}}. \tag{D.10}$$

Since P_t^{peer} is row-stochastic and both inputs are bounded, $B_t^{\mathrm{seis}} \in [-1, 1]^{N \times A}$. Expanding (D.10) recursively shows that earlier information propagates along time-varying multi-tick graph paths with geometrically decaying weight: for $t \ge 2$,

$$\begin{aligned} B_t^{\mathrm{seis}} = {} & (1 - \rho_{\mathrm{seis}}) P_t^{\mathrm{peer}} U_{t-1}^{\mathrm{beh}} + \rho_{\mathrm{seis}}(1 - \rho_{\mathrm{seis}}) P_t^{\mathrm{peer}} P_{t-1}^{\mathrm{peer}} U_{t-2}^{\mathrm{beh}} \\ & + \rho_{\mathrm{seis}}^2 P_t^{\mathrm{peer}} P_{t-1}^{\mathrm{peer}} B_{t-2}^{\mathrm{seis}}. \end{aligned} \tag{D.11}$$

The first two terms are one- and two-hop information paths; the last retains the earlier history. At $\rho_{\mathrm{seis}} = 0$, belief is the current one-hop signal from lagged behavior; at $\rho_{\mathrm{seis}} = 1$, no new behavior is injected and zero initial belief stays zero.

D.4 RUNTIME ELIGIBILITY AND BOUNDED PROJECTION

For the projection, each strategy slot is summarized by its actionable buy notional, actionable sell quantity, and static projection eligibility, while the validated contribution collection remains the order content passed downstream. A slot is runtime eligible only if it is statically eligible and has actionable contributions on exactly one side; a slot with no action or with mixed buy/sell contributions bypasses the projection and retains its base weight. Static eligibility is a frozen configuration-level mask declared before the run. Suppress the fixed cell and tick indices (i, a, t) in the slot quantities below. For each eligible slot s, define

$$d_s^{\mathrm{act}} = \mathrm{clip}\big[\rho_c(M_s^{\mathrm{act,buy}}, C^{\mathrm{buy}}) - \rho_c(Q_s^{\mathrm{act,sell}}, Q^{\mathrm{sellable}}), -1, 1\big]\,. \tag{D.12}$$

Eligible buy slots have $d_s^{\mathrm{act}} > 0$ and eligible sell slots have $d_s^{\mathrm{act}} < 0$. For trader–asset belief $B_{ia}^{\mathrm{seis}}(t)$, compatibility is $c_s = B_{ia}^{\mathrm{seis}}(t) d_s^{\mathrm{act}}$, and the bounded logit shift is

$$\Delta_s = \mathrm{clip}\big(\lambda_{\mathrm{seis}} c_s, -\delta_{\max}, +\delta_{\max}\big), \qquad \lambda_{\mathrm{seis}} \geq 0, \quad \delta_{\max} \geq 0. \tag{D.13}$$

The sensitivity, clipping bound, and trust strength are fixed simulation parameters, recorded in Appendix J.1.

For one trader–asset–tick cell, suppress (i, a, t) and write $\pi_s = \pi_{3,ias}$ and $\widetilde{\pi}_s = \widetilde{\pi}_{3,ias}(t)$. The symbols $p_s, q_s, \widetilde{p}_s$ below are local sub-simplex weights; the field index p of Appendix F is unrelated. Let $\mathcal{E}$ be the runtime eligible slots and $\mathcal{E}^{\mathrm{bypass}}$ the bypass slots, with $\widetilde{\pi}_s = \pi_s$ for $s \in \mathcal{E}^{\mathrm{bypass}}$. Let

$$m_{\mathcal{E}} = \sum_{s \in \mathcal{E}} \pi_s, \qquad \mathcal{E}_+ = \{s \in \mathcal{E} : \pi_s > 0\}, \qquad p_s = \frac{\pi_s}{m_{\mathcal{E}}} \quad (m_{\mathcal{E}} > 0). \tag{D.14}$$

If the eligible mass is zero, the positive support has size at most one, the belief is zero, the trust strength is zero, or all clipped shifts on $\mathcal{E}_+$ are equal, the projection returns the base vector exactly. Otherwise the shifted eligible sub-simplex is

$$q_s = \frac{p_s e^{\Delta_s}}{\sum_{k \in \mathcal{E}_+} p_k e^{\Delta_k}}, \qquad s \in \mathcal{E}_+, \tag{D.15}$$

with $q_s = 0$ for eligible zero-support slots; the exponential is evaluated by a stable softmax over the positive support. For $0 \leq \eta_\pi \leq 1$,

$$\widetilde{p}_s = (1 - \eta_\pi) p_s + \eta_\pi q_s, \qquad \widetilde{\pi}_s = m_{\mathcal{E}} \widetilde{p}_s, \quad s \in \mathcal{E}. \tag{D.16}$$

The projection preserves nonnegativity, total probability, every bypass weight, the total eligible mass, and all base zero support, redistributing only the currently eligible positive support.

The same construction yields an explicit intervention bound. For probability vectors x, y, define $\mathrm{TV}(x, y) = \frac{1}{2} \sum_s |x_s - y_s|$. The convex blend in (D.16) gives

$$\mathrm{TV}(\widetilde{p}, p) = \eta_\pi \, \mathrm{TV}(q, p) \leq \eta_\pi, \qquad \mathrm{TV}(\widetilde{\pi}, \pi) = m_{\mathcal{E}} \, \mathrm{TV}(\widetilde{p}, p) \leq m_{\mathcal{E}} \eta_\pi, \tag{D.17}$$

where the second identity uses the unchanged bypass weights. The bound applies to one cell and one update, anchored at the structural mixture.

D.5 ONE-TICK UPDATE

One tick of the mechanism proceeds as follows. The information branch computes P_t^{peer} from U_{t-1}^{beh} and updates B_t^{seis} by (D.10); independently, the strategy branch computes raw outputs and validates their contributions against one pre-matching capacity snapshot. The branches join at the projection: each trader–asset cell either satisfies an identity condition of Appendix D.4 and keeps its base mixture, or computes the clipped shifts, the positive-support softmax, and the eligible-mass-preserving blend. The resulting $\widetilde{\pi}_3(t)$ is applied by strategy slot to the validated contributions, which

then enter the aggregation and execution pipeline of Appendix B.3. Finally, U_t^{beh} is computed from the structural π_3, the validated contributions, and the same capacity snapshot, and both arrays are retained for the next tick; the replay information is collected in Appendix J.7.

E MARKET INSTITUTION: SETTLEMENT, PRICE LIMITS, AND THE PLUGGABLE RULE LAYER

The environment κ includes a market institution: the rules under which orders are admitted and settled. The reported experiments pin one fixed configuration, the Chinese A-share main-board rules with T+1 settlement and daily price limits, loaded through a pluggable rule layer. The population, the decoder, and the search interact with the institution only through day-initialization, order-admission, and settlement-lifecycle hooks, so another market's conventions can be substituted by replacing the rule module.

E.1 VERSIONED RULE REGISTRY

A central registry assigns every security a trading-rule profile—venue, board, and security type, together with a settlement profile and a price-limit profile—so that rule formulas are shared through central profiles rather than duplicated per security. Rule sets are versioned by effective date. Two selection modes are supported: a fixed mode that pins one ruleset version, and an as-of-simulation-date mode that replays the rules in force on each simulated trading day. The reported experiments use the fixed mode, so results do not depend on the date on which the simulation is run.

E.2 T+1 SETTLEMENT ACCOUNTING

Settlement status is tracked per trader account. Total holdings decompose as *owned* $=$ *settled* $+$ *unsettled*, where every unsettled buy lot records its quantity, acquisition trading day, and unlock trading day; sellable holdings are *available* $=$ *settled* $-$ *reserved*. A buy fill appends a locked lot whose unlock day is the next trading day according to the trading calendar: it increases total holdings but not sellable holdings. Upon accepting a sell order the broker checks the sellable balance and immediately reserves the admitted amount, releasing the reservation on rejection, cancellation, or expiry. At each trading-day switch, all lots whose unlock day has been reached migrate into settled holdings. Proceeds from sell fills remain tradable cash within the day. Initial holdings established at initialization (Appendix B.1) enter as settled.

E.3 DAILY PRICE-BAND CONSTRUCTION AND ORDER ADMISSION

Before each trading day opens, the exchange constructs an immutable daily price band for every symbol from the reference price (previous close, or the adjusted reference on corporate-action days), the tick size, and the price-limit profile in force. Under the pinned A-share main-board profile, the lower and upper limits are the reference price scaled by 0.90 and 1.10, rounded half-up to the tick; the profile mechanism also admits sessions without daily limits, such as the first trading days after an initial listing, as a profile attribute. Limit orders arriving at exchange ingress are checked against the band and rejected if outside it; orders at boundary prices queue and match normally, so trading continues at a locked price while the limit state persists. The price-band frequency stylized fact of Appendix G measures proximity to the corresponding 10% price-limit band under the proxy convention stated there.

E.4 TRADING-DAY LIFECYCLE

Each trading-day switch executes a fixed sequence: determine the next trading day from the trading calendar; load the ruleset version and security states for that day; construct the daily price bands; unlock due T+1 lots; refresh the broker account projections; open the market and begin accepting orders. Every step is deterministic given the frozen calendar and registry, so the institutional layer introduces no additional randomness into the simulation-input distribution of Appendix G.

F DECODER AND SEARCH-SPACE REPRESENTATION

The calibration search operates on explicit coordinates Z decoded by $X(Z) = M(Z; X^0, \mathcal{R})$. This appendix gives the decoding maps in full, together with the conventions and legality conditions on which they depend. Throughout, a *residual* is a change of scores relative to the frozen baseline, not a deviation of simulated facts from observed ones. All formulas are stated for general population size N, asset count A, strategy-type count S, and field count P; the configuration used in the experiments is reported in Appendix J.

F.1 BASELINE, ARCHITECTURE, AND SCHEMA

The decoder is specified by a frozen baseline and schema together with explicit architecture metadata.

Baseline. X^0 is a legal population selected and frozen before the run. It supplies the baseline shares π_1^0, π_2^0, π_3^0 and baseline parameter values θ^0, from which the baseline scores L_2^0, L_3^0, and T^0 used below are derived; the reference logits z_1^0 together with zero residuals at the remaining levels reproduce X^0 exactly. Because the coordinate maps below take logarithms of baseline shares, the baseline share vectors are strictly positive on the declared trader, asset, and strategy sets, so all baseline scores are finite. The baseline may carry rich heterogeneity; it is held fixed across candidates, and changing it changes the coordinate family.

Architecture. $\mathcal{R}$ collects the discrete representation choices: the second-level decoding mode (Appendix F.3 or the optional variant of Appendix F.4), the ranks r_3 and r_θ, the fixed contrast basis C_A, the reference-trader and reference-strategy conventions, and the axis conventions of the third- and fourth-level cores. These entries are metadata, not continuous search coordinates. The ranks are adapted by the search procedure of Appendices H and H.9, and every such adaptation is recorded together with the coordinate identity it produces (Appendix F.8).

Schema. A frozen schema lists, for each active numerical strategy-parameter field, its flattened slot index p, the owning strategy type and internal field name, an admissible range $[a_p, b_p]$ with $a_p < b_p$, a declared type, a quantization or rounding rule, and any joint constraints across fields. Fields with $a_p = b_p$ are constants and carry no coordinates; unordered categorical fields are frozen at their baseline values in the present profile. The schema also fixes the flattening order $(s, j) \leftrightarrow p$ used by the fourth level.

Legality. Every candidate is fully decoded and validated before it enters the simulator: softmax maps are evaluated in stabilized form, decoded shares must be nonnegative and sum to one within floating-point tolerance, and restored parameter values must satisfy their declared ranges, types, and joint constraints. Candidates failing validation are rejected.

F.2 FIRST-LEVEL DECODER

One trader, fixed by the frozen identity ordering, serves as the reference. Writing $z_1 \in \mathbb{R}^{N-1}$ for the reference logits,

$$\pi_1 = \mathrm{softmax}(z_1, 0), \qquad z_{1,i}^0 = \log \frac{\pi_{1,i}^0}{\pi_{1,N}^0}. \tag{F.1}$$

Adding a common constant to all scores leaves the shares unchanged, so fixing the reference score at zero removes the redundancy and leaves exactly $N - 1$ coordinates; the reference trader's own share still moves with the remaining coordinates. The baseline coordinates z_1^0 reproduce π_1^0 exactly, and the coordinate count is $d_1 = N - 1$.

F.3 SECOND-LEVEL DECODER: FULL-CONTRAST MAIN PROFILE

Each row $\pi_{2,i}$ of the second level is a share vector over assets. The decoder works with row-centered log-ratio scores (Aitchison, 1986): for a positive share row p, the centered scores $l_a = \log p_a - \frac{1}{A}\sum_b \log p_b$ satisfy $\sum_a l_a = 0$ and reproduce p through a row-wise softmax, so the log-

ratio step is an exact change of coordinates. The baseline scores are the row-wise centered log-ratios of the baseline shares, $L^0_{2,ia} = \log \pi^0_{2,ia} - \frac{1}{A}\sum_{b=1}^{A} \log \pi^0_{2,ib}$, and the searched scores are

$$L_2 = L^0_2 + E_2, \qquad \pi_{2,i} = \mathrm{softmax}(L_{2,i}). \tag{F.2}$$

The residual is expressed in a fixed contrast basis $C_A \in \mathbb{R}^{A\times(A-1)}$ satisfying

$$C_A^\top C_A = I_{A-1}, \qquad C_A^\top \mathbf{1} = 0, \qquad C_A C_A^\top = \Pi_A := I_A - \tfrac{1}{A}\mathbf{1}\mathbf{1}^\top, \tag{F.3}$$

whose columns span the zero-sum contrast subspace. The searched object is $B_2 \in \mathbb{R}^{N\times(A-1)}$ with

$$E_2 = B_2 C_A^\top. \tag{F.4}$$

This is an exact reparameterization, not a structural restriction: every centered residual satisfies $E_2 = (E_2 C_A)C_A^\top$ and therefore has the unique coordinates

$$B_2 = E_2 C_A, \tag{F.5}$$

so the map between centered residuals and contrast coordinates is a bijection. Searching B_2 directly removes the translation and invertible-basis redundancies that two-factor coordinates would carry, while keeping each trader's $A-1$ free directions intact. The explicit coordinate count is $d_2 = N(A-1)$; the fixed basis C_A and the baseline scores L^0_2 are frozen metadata, not coordinates.

Nonnegativity needs no separate constraint: for any finite B_2, the row softmax yields $\pi_{2,ia} > 0$ with unit row sums by construction. Under the calibrated cash-fraction variant of Appendix B.1, a second, sigmoid-decoded sub-block joins this level; its coordinates are defined in Appendix F.5. The reported five-asset profile searches B_2 only.

F.4 Optional scalable second-level variant

When the number of assets is large enough that the $N(A-1)$ direct contrast coordinates of Appendix F.3 are undesirable, the residual can instead be represented by low-rank factors,

$$E_2 = U_2 V_2^\top \Pi_A, \qquad U_2 \in \mathbb{R}^{N\times r_2}, \quad V_2 \in \mathbb{R}^{A\times r_2}, \tag{F.6}$$

decoded through the same baseline addition and row-wise softmax as in (F.2). The variant stores $(N+A)r_2$ explicit coordinates. At rank $r_2 \geq \min(N, A-1)$ it represents every centered residual and is therefore equal in expressive range to the main profile; at smaller rank it imposes the structural restriction $\mathrm{rank}(E_2) \leq r_2$, so that deviations from the baseline are generated by a few shared directions linking trader and asset identities.

Its coordinates are redundant: for any invertible matrix Λ, the pairs (U_2, V_2) and $(U_2\Lambda, V_2\Lambda^{-\top})$ decode to the same residual, so factor entries are representation coordinates rather than interpretable quantities. The two modes are alternative frozen architecture choices recorded in $\mathcal{R}$; the full-contrast profile of Appendix F.3 is the default, and the variant is retained for large-asset configurations.

F.5 Optional calibrated cash-fraction sub-block

Under Branch C of Appendix B.1, the per-account cash fraction leaves the frozen configuration and joins the searched coordinates as a second sub-block of the second level. Unlike the share rows, the cash fractions carry no sum-to-one constraint, so no contrast basis is involved. For an interior baseline table $c^0_2 \in (0,1)^{N\times A}$, define the baseline logits and the searched scores by

$$L^{c,0}_2 = \mathrm{logit}\big(c^0_2\big), \qquad c_2 = \sigma\big(L^{c,0}_2 + E^c_2\big), \tag{F.7}$$

where $\sigma(x) = (1+e^{-x})^{-1}$ acts entrywise and $E^c_2 \in \mathbb{R}^{N\times A}$ is the cash-fraction residual. The logit step is an exact change of coordinates for interior baseline fractions; entries of c^0_2 on the boundary

$\{0, 1\}$ have no finite logit and are frozen at their baseline values, carrying no coordinates. The decoded table enters initialization exactly as the frozen table does in (B.1), with c_2 replacing c_2^0, so the per-account budget identity and the conservation of W_0 are unchanged.

Two coordinate forms parallel the asset-allocation block. The direct form searches the entries of E_2^c themselves, NA coordinates; the scalable form factors the residual,

$$E_2^c = U_2^c \big(V_2^c\big)^\top, \qquad U_2^c \in \mathbb{R}^{N\times r_2^c}, \quad V_2^c \in \mathbb{R}^{A\times r_2^c}, \tag{F.8}$$

storing $(N + A)r_2^c$ coordinates with the same gauge redundancy as the low-rank share variant of Appendix F.4. Either form is a frozen architecture choice recorded in $\mathcal{R}$, and the searched coordinates of the sub-block belong to the second-level block for stage scheduling: the local second-level stage of Appendix H.5 includes them in its sparse group support whenever the sub-block is active. The reported profile fixes the cash fraction by configuration and activates neither form.

F.6 Third-level decoder

The third level represents, for each trader–asset pair, the mixing weights $\pi_{3,ia}$ over the S strategy types. One strategy type, fixed in $\mathcal{R}$, serves as the reference; with contrast indices $\ell = 1, \dots, S-1$,

$$L_{3,ia\ell} = \log \frac{\pi_{3,ia\ell}}{\pi_{3,iaS}}, \qquad \pi_{3,ia} = \mathrm{softmax}(L_{3,ia}, 0), \tag{F.9}$$

so the log-ratio step is again exact, and the baseline scores $L^0_{3,ia\ell} = \log\big(\pi^0_{3,ia\ell}/\pi^0_{3,iaS}\big)$ are finite by the baseline condition of Appendix F.1.

The residual $E_3 = L_3 - L_3^0 \in \mathbb{R}^{N\times A\times(S-1)}$ follows a partial Tucker construction (Kolda & Bader, 2009) in which trader heterogeneity is compressed through shared latent directions while the asset and strategy-contrast axes are kept explicit. In general form,

$$E_{3,ia\ell} = \sum_{k=1}^{r_{3N}} \sum_{m=1}^{r_{3A}} U_{3,ik}\, V_{3,am}\, \mathcal{G}_{3,km\ell}, \tag{F.10}$$

with trader loadings $U_3 \in \mathbb{R}^{N\times r_{3N}}$, asset loadings $V_3 \in \mathbb{R}^{A\times r_{3A}}$, and core $\mathcal{G}_3 \in \mathbb{R}^{r_{3N}\times r_{3A}\times(S-1)}$. Keeping the strategy-contrast axis explicit allows each strategy contrast to combine the shared directions through its own core slice, rather than forcing one pairwise direction numbering as in a CP decomposition. The profile used in this paper fixes the asset axis to the identity, $V_3 = I_A$, and writes $H_3 \in \mathbb{R}^{r_3\times A\times(S-1)}$ for the resulting core, giving the identity-axis form used throughout,

$$E_{3,ia\ell} = \sum_{k=1}^{r_3} U_{3,ik} H_{3,ka\ell}. \tag{F.11}$$

Any learned-axis representation can be rewritten in this form by absorbing the asset factor into the core, $H_3 = \mathcal{G}_3 \times_2 V_3$, so the learned-axis family is nested in the identity-axis family under the same baseline, schema, and trader-rank limit. Under the identity axis the explicit coordinate count is $d_3 = r_3\big(N + A(S-1)\big)$.

Equation (F.11) decodes the structural mixture π_3 of Section 3.1. Its transient effective counterpart $\widetilde{\pi}_3(t)$ is formed during forward simulation as specified in Appendix D.4.

The loading row $U_{3,i}$ measures how strongly trader i participates in each shared direction, and the core slice $H_{3,k}$ records how direction k shifts the strategy contrasts of each asset. These factors are not uniquely identified by the residual: for any invertible Λ, the pairs (U_3, H_3) and $(U_3\Lambda,\ H_3 \times_1 \Lambda^{-1})$ decode identically. The latent directions are therefore optimization constructs rather than identified economic trader types; the interpretable objects are the decoded populations and their simulation outcomes.

F.7 FOURTH-LEVEL DECODER

The fourth level represents the strategy parameters θ. The schema of Appendix F.1 flattens the active numerical fields of all strategy types into slots $p = 1, \dots, P$ and records each slot's admissible range and declared type. Each baseline value is first mapped to $[0, 1]$ by the exact affine change of coordinates

$$T^0_{iap} = \frac{\theta^0_{iap} - a_p}{b_p - a_p}, \qquad \theta_{iap} = a_p + (b_p - a_p)\, T_{iap}, \tag{F.12}$$

and the residual in normalized units follows the same partial Tucker form with identity asset and field axes:

$$\widetilde{T}_{iap} = T^0_{iap} + E_{\theta,iap}, \qquad E_{\theta,iap} = \sum_{k=1}^{r_\theta} U_{\theta,ik}\, H_{\theta,kap}, \tag{F.13}$$

with $U_\theta \in \mathbb{R}^{N \times r_\theta}$ and $H_\theta \in \mathbb{R}^{r_\theta \times A \times P}$; the explicit coordinate count is $d_\theta = r_\theta(N + AP)$. Keeping the field axis explicit gives every slot its own position in the core rather than treating heterogeneous parameters as one variable; the rank restriction limits how trader heterogeneity is organized, not which fields exist.

Decoding reverses the normalization in three steps:

$$T = \mathrm{clip}_{[0,1]}(\widetilde{T}), \qquad \theta_{iap} = \mathcal{Q}_p\big\{a_p + (b_p - a_p)\, T_{iap}\big\}, \tag{F.14}$$

where $\mathcal{Q}_p$ is the frozen per-field type rule—the identity for continuous fields and the declared rounding rule for integer fields—after which the schema's joint constraints are validated as in Appendix F.1. Clipping creates plateaus: distinct normalized residuals can map to the same boundary value, and integer quantization adds further jumps, so the slot count P counts storage positions rather than independent continuous degrees of freedom. No softmax is involved at this level, because ranges and types, rather than a sum-to-one constraint, define legality. The coordinates deterministically generate each concrete parameter value handed to the strategies; no fresh parameter draw occurs inside the simulation.

F.8 DECODER IDENTITY AND REPRESENTATION BOOKKEEPING

Let z_2 denote the active second-level coordinates, $z_2 = B_2$ in full-contrast mode and $z_2 = (U_2, V_2)$ in low-rank mode. The complete search coordinates assemble the per-level blocks as

$$Z = \big(z_1,\, z_2,\, U_3,\, H_3,\, U_\theta,\, H_\theta\big), \tag{F.15}$$

with explicit coordinate counts

$$d_1 = N - 1, \qquad d_2 = N(A-1), \qquad d_3 = r_3\big(N + A(S-1)\big), \qquad d_\theta = r_\theta(N + AP), \tag{F.16}$$

under the main profile; the optional second-level variant replaces d_2 by $(N + A)r_2$, and the calibrated cash-fraction sub-block of Appendix F.5 appends its own NA direct or $(N + A)r^c_2$ factored coordinates to the second-level block. Given the baseline, architecture, and schema, the decoder $X(Z) = M(Z; X^0, \mathcal{R})$ is deterministic, and the baseline logits z^0_1 and zero residuals at levels two to four reproduce X^0 exactly. Coordinates are interpreted together with their architecture, reference choices, and schema; comparisons use the complete decoded population, not factor coordinates. Appendix H.7 specifies candidate reuse, and the rank changes performed by the search and their parent mappings are specified in Appendices H.10 and H.11.

G CALIBRATION OBJECTIVE AND FROZEN METRIC

The operational calibration score $\widehat{J}$ has a fixed-candidate expectation property, stated and proved in Appendix I.4. This appendix gives the full evaluation procedure behind it: the stylized-fact vector and its empirical reference, the construction and freezing of the common metric W, the repeated-pair estimator of which the operational score is the single-pair case, and the treatment of negative scores, invalid candidates, and common random inputs.

G.1 STYLIZED FACTS AND EMPIRICAL REFERENCE

A frozen fact calculator $\mathcal{F}$ maps a price and volume series—historical or simulated—to a fact vector. For asset a,

$$m_a^{\mathrm{obs}} = \mathcal{F}_a(\mathcal{D}), \qquad m_a(X;\xi) = \mathcal{F}_a\big(\mathcal{Y}_a(X;\xi)\big), \tag{G.1}$$

where $\mathcal{D}$ is the observed market data and $\mathcal{Y}_a(X;\xi)$ is the simulated trajectory of asset a under population X with random input ξ in the common environment κ. Dependence on κ is suppressed in m_a and $\mathcal{Y}_a$, as in Section 3.3. Every asset uses the same ordered fact list of common length K_f; the current list summarizes the return distribution and its interaction with trading activity—return volatility, return kurtosis, the first-order autocorrelation of absolute returns, the correlation between absolute returns and volume, and a minute-grid price-band frequency proxy—in the empirical tradition of Cont (2001). Sampling frequency, window lengths, the exact statistic conventions, and the treatment of missing values are fixed before the search and applied identically to observed and simulated series.

If a required fact is undefined or non-finite for an asset, the evaluation is invalid and the candidate receives no valid score for it, following the invalid-candidate rules of Appendix G.5.

G.2 DAILY FACT DEFINITIONS AND PERIOD TARGETS

The current fact specification has $K_f = 5$. For one asset and trading day, let $p_h^{\mathrm{px}} > 0$ and $v_h^{\mathrm{vol}} \geq 0$ be the minute-grid price and volume for $h = 1, \ldots, n_{\mathrm{grid}}$, with $n_{\mathrm{ret}} = n_{\mathrm{grid}} - 1$. Define

$$r_h = 10^4 \log \frac{p_{h+1}^{\mathrm{px}}}{p_h^{\mathrm{px}}}, \quad a_h^{\mathrm{abs}} = |r_h|, \quad \bar{r} = \frac{1}{n_{\mathrm{ret}}} \sum_{h=1}^{n_{\mathrm{ret}}} r_h, \quad \mu_j^{\mathrm{cen}} = \frac{1}{n_{\mathrm{ret}}} \sum_{h=1}^{n_{\mathrm{ret}}} (r_h - \bar{r})^j. \tag{G.2}$$

The factor 10^4 expresses log returns in basis points and is shared by historical and simulated calculators. The five daily facts, in order, are

$$\begin{aligned}
f_1 &= \left[\frac{1}{n_{\mathrm{ret}} - 1} \sum_h (r_h - \bar{r})^2 \right]^{1/2}, \\
f_2 &= \frac{\mu_4^{\mathrm{cen}}}{(\mu_2^{\mathrm{cen}})^2} - 3, \\
f_3 &= \mathrm{Corr}\big((a_1^{\mathrm{abs}}, \ldots, a_{n_{\mathrm{ret}}-1}^{\mathrm{abs}}), (a_2^{\mathrm{abs}}, \ldots, a_{n_{\mathrm{ret}}}^{\mathrm{abs}})\big), \\
f_4 &= \mathrm{Corr}\big((a_1^{\mathrm{abs}}, \ldots, a_{n_{\mathrm{ret}}}^{\mathrm{abs}}), (v_2^{\mathrm{vol}}, \ldots, v_{n_{\mathrm{grid}}}^{\mathrm{vol}})\big), \\
f_5 &= \frac{1}{n_{\mathrm{grid}}} \sum_{h=1}^{n_{\mathrm{grid}}} \mathbf{1}\{p_h^{\mathrm{px}} \geq 1.10(0.999) c_{a,d}^{\mathrm{ref}} \text{ or } p_h^{\mathrm{px}} \leq 0.90(1.001) c_{a,d}^{\mathrm{ref}}\}.
\end{aligned} \tag{G.3}$$

Here $c_{a,d}^{\mathrm{ref}}$ is the reference price known before day d. f_1 is sample volatility with denominator $n_{\mathrm{ret}} - 1$, without annualization. f_2 is uncorrected Fisher excess kurtosis. The calculator returns zero for f_2 when $\mu_2^{\mathrm{cen}} = 0$ and for a Pearson correlation with zero denominator.

The fifth component is a frequency near a 10% band relative to an inferred prior close, with the displayed 0.1% relative boundary tolerance. It is a price-limit-frequency proxy, not an official historical limit-hit classification.

The empirical price-grid, reference-price and screening choices are reported in Appendix J.2.

Daily vectors are aggregated over a declared period. Write $g_{a,d} = (f_1, \ldots, f_5)^\top$ for the daily vector. For a declared period $\mathcal{D}_{\text{target}}$ of n_{day} valid trading days, the period fact vector is

$$m_a = \frac{1}{n_{\text{day}}} \sum_{d \in \mathcal{D}_{\text{target}}} g_{a,d}. \tag{G.4}$$

Daily nonlinear facts are computed first and then averaged; historical targets and simulated path facts use the same order and units. The calibration and held-out horizons and the asset-weight convention are given in Appendix J.3.

G.3 Frozen common metric W

All assets share one fact metric $W = W^\top \succeq 0$: two assets with the same fact-deviation vector incur the same within-asset loss. The asset weights ν_a aggregate per-asset terms only; all fact scaling, empirical variability, and co-movement are handled by W, together with an optional common diagonal matrix Q_f of fact importances applied uniformly across assets.

The current construction uses the empirical distribution of fixed-length historical windows. Let n_{hist} be the reference calendar length, h_{ref} the window length, and $\mathcal{T}_a^{\text{win}}$ the set of starts whose h_{ref} consecutive calendar days all have valid daily facts. With $n_a^{\text{win}} = |\mathcal{T}_a^{\text{win}}|$,

$$\begin{aligned}
\bar{g}_{a,d}^{\text{win}} &= \frac{1}{h_{\text{ref}}} \sum_{j=0}^{h_{\text{ref}}-1} g_{a,d+j}, \qquad d \in \mathcal{T}_a^{\text{win}}, \\
\bar{g}_a^{\text{win}} &= \frac{1}{n_a^{\text{win}}} \sum_{d \in \mathcal{T}_a^{\text{win}}} \bar{g}_{a,d}^{\text{win}}, \\
\widehat{\Sigma}_a &= \frac{1}{n_a^{\text{win}}} \sum_{d \in \mathcal{T}_a^{\text{win}}} (\bar{g}_{a,d}^{\text{win}} - \bar{g}_a^{\text{win}})(\bar{g}_{a,d}^{\text{win}} - \bar{g}_a^{\text{win}})^\top.
\end{aligned} \tag{G.5}$$

Every complete window is enumerated at stride one, with denominator n_a^{win}, since the target is the finite uniform distribution of those windows; missing days exclude the entire window. This construction measures the historical variation of fixed-window average facts, including variation between historical market states.

Center within each reference asset, then pool the covariances equally:

$$\widehat{\Sigma}_{\text{pool}} = \frac{1}{A_{\text{ref}}} \sum_{a \in \mathcal{A}_{\text{ref}}} \widehat{\Sigma}_a, \qquad W^{\text{stat}} = \widehat{\Sigma}_{\text{reg}}^{-1}, \qquad W = Q_f^{1/2} W^{\text{stat}} Q_f^{1/2}. \tag{G.6}$$

Here $A_{\text{ref}} = |\mathcal{A}_{\text{ref}}|$ is the reference-pool size, distinct from the A scored assets. Equal pooling precedes inversion, so that no asset contributes its own inverse metric.

For a symmetrized pooled covariance with positive diagonal, let $S_f = \operatorname{diag}(s_1, \ldots, s_{K_f})$, $s_j^2 = (\widehat{\Sigma}_{\text{pool}})_{jj}$. Regularize in correlation units:

$$\begin{aligned}
R_f &= S_f^{-1} \widehat{\Sigma}_{\text{pool}} S_f^{-1}, & R_{f,\alpha} &= (1-\alpha_f) R_f + \alpha_f I_{K_f}, \\
R_{f,\alpha} &= V_f \operatorname{diag}(\zeta_j) V_f^\top, & \widetilde{R}_f &= V_f \operatorname{diag}(\max\{\zeta_j, \epsilon_f\}) V_f^\top, \\
\widehat{\Sigma}_{\text{reg}} &= S_f \widetilde{R}_f S_f.
\end{aligned} \tag{G.7}$$

V_f is an orthogonal eigenvector matrix; ζ_j are eigenvalues. The eigenvalue floor is dimensionless; the numerical choices are reported in Appendix J.4.

The first four marginal variances must be positive. A nonpositive fifth variance triggers a prespecified price-band tolerance fallback, which clears its cross-covariances and substitutes the declared tolerance variance before regularization; its use in the documented artifact is reported in Appendix J.4.

The reference-window scale is a declared empirical choice in the simulated-moments tradition (McFadden, 1989), and remains fixed if a subsequent simulation uses a different horizon. W is constructed before candidate evaluation and is common to all assets. The reference data and actual numerical artifact are described in Appendix J.4.

G.4 Expected objective and the two-path estimator

Let $\mu_a(X) = \mathbb{E}_\xi\, m_a(X;\xi)$ denote the expected fact vector of asset a under a fixed population X. The theoretical calibration target is the expected-fact discrepancy

$$J(X) = \sum_{a=1}^{A} \nu_a \left[\mu_a(X) - m_a^{\mathrm{obs}}\right]^\top W \left[\mu_a(X) - m_a^{\mathrm{obs}}\right], \qquad \nu_a \geq 0. \tag{G.8}$$

Equivalently, stacking the weighted deviations $g(X) = \mathrm{col}_a\big(\sqrt{\nu_a}\,[\mu_a(X) - m_a^{\mathrm{obs}}]\big)$ gives $J(X) = g(X)^\top (I_A \otimes W) g(X)$, which makes explicit that every asset block carries the same W and that no cross-asset error terms enter the objective. J is a nonnegative theoretical calibration loss and cannot be computed exactly, because the expectation over simulation inputs is inaccessible.

The operational score is the two-path cross product: drawing two independent random inputs $\xi^{(1)}, \xi^{(2)}$ and writing $m_a^{(r)}(X) = m_a(X;\xi^{(r)})$,

$$\widehat{J}(X) = \sum_{a=1}^{A} \nu_a \left[m_a^{(1)}(X) - m_a^{\mathrm{obs}}\right]^\top W \left[m_a^{(2)}(X) - m_a^{\mathrm{obs}}\right]. \tag{G.9}$$

A path spans the full declared simulation horizon; two segments of one evolving path are not two independent repetitions, and the two paths always belong to the same candidate.

The single pair of (G.9) is the minimal instance of a repeated-pair estimator. Let $\omega_b = (\xi_b^{(1)}, \xi_b^{(2)})$ denote the independent input pair of repetition $b = 1, \ldots, n_{\mathrm{rep}}$, and

$$\ell(X;\omega_b) = \sum_{a=1}^{A} \nu_a \left(\Delta_{a,b}^{(1)}\right)^\top W \Delta_{a,b}^{(2)}, \qquad \widehat{J}_{n_{\mathrm{rep}}}(X) = \frac{1}{n_{\mathrm{rep}}} \sum_{b=1}^{n_{\mathrm{rep}}} \ell(X;\omega_b), \tag{G.10}$$

where $\Delta_{a,b}^{(r)} = m_a(X;\xi_b^{(r)}) - m_a^{\mathrm{obs}}$. Evaluating n_{rep} pairs costs $2n_{\mathrm{rep}}$ complete simulations of the same candidate; the pairs are disjoint and independent by design, so the pair scores are independent of one another. If the per-pair score has finite variance, $\mathrm{Var}\big(\widehat{J}_{n_{\mathrm{rep}}}(X)\big) = \mathrm{Var}\big(\ell(X;\omega)\big)/n_{\mathrm{rep}}$: additional pairs improve the stability of an already unbiased score. The operational main search uses exactly one pair per new unique candidate, $\widehat{J} = \widehat{J}_1$; averaging facts across paths before squaring, or squaring each path before averaging, define different statistics and are reported only as diagnostics.

G.5 Realized scores, invalid candidates, and common random inputs

The theoretical target J is nonnegative, but a realized cross score can be negative when the two paths of a pair land on opposite sides of the observed facts. Scores are therefore used as returned—clipping at zero or taking absolute values would destroy the expectation property of Appendix I.4—and the optimizer ranks candidates by the real-valued score.

If a required fact is undefined or non-finite, or the simulation fails, the evaluation has no valid score; failure handling inside the search follows the declared rules of Appendix H.

Comparability across candidates rests on three rules. First, the same pair of random input streams is held fixed across candidates, so that differences in scores reflect populations rather than resampled luck. Second, every candidate is simulated with its own independent simulator state—accounts, order books, and generator state are not shared between candidates, even when their input streams coincide. Third, the random numbers driving the optimization procedure are drawn from a separate stream and do not enter the simulation inputs. Common inputs induce positive covariance between the scores of two candidates evaluated on the same streams, which reduces the variance of their difference and sharpens comparisons, although the reduction is encouraged rather than guaranteed under every response relation (Glasserman & Yao, 1992).

G.6 CALIBRATION-SEARCH SCORE, TEMPORAL HELD-OUT MONITORING, AND FRESH-SEED ROBUSTNESS

Three empirical questions are distinct.

Calibration-search score is the realized common-random-input score $\widehat{J}(X)$ on the frozen pair of simulation streams used by the search; it is the quantity optimized during the search.

Temporal held-out monitoring evaluates a calibration-side candidate on market data immediately following the calibration window. Appendix H.8 defines this evaluation as observe-only: it does not alter proposals, branch selection, or the calibration winner.

Fresh-seed robustness asks whether the selected population remains compatible with the target facts under simulation randomness not used to construct the search trajectory. This is a post-search empirical check, not part of the default two-path calibration objective; if reported, its random inputs, number of repetitions, and aggregation rule are stated separately.

Accordingly, "empirically compatible" is always read relative to the checks actually reported: a low search score evidences close agreement with the declared facts on the frozen search surface, while temporal held-out behavior and fresh-seed robustness provide different evidence.

H PROGRESSIVE HIERARCHICAL INFERENCE

The structured search is organized into rounds, stages, and studies. This appendix states the complete procedure: the outer lifecycle, the fixed-map invariant of one study, the temporary search map and its legality checks, the scheduling of local groups, the stage ordering, the construction and inheritance of directions, the warm-started Sobol-to-TPE transition, the paired local policy of the second-level stage, radius adaptation, cache and failure semantics, and the multi-start and held-out protocol. Numerical settings enter as configuration constants; this appendix fixes the rules, and the profile's key quantities are reported in Appendix J.

H.1 OUTER SEARCH LIFECYCLE AND STAGE ORDERING

Algorithm 1 states the complete outer lifecycle. Everything the score depends on is frozen before the run: the simulator and decoder, the baseline X^0, the metric W, the simulation-input pair, and the architecture limits.

The baseline is evaluated once and shared by all starts; without a valid baseline score the run does not proceed. Structural comparison happens only at declared round intervals and only between rounds. The final selection is taken from the entire evaluated archive of the calibration search, with ties broken by complexity and then stable identifiers; no additional simulations are run for final selection.

Odd rounds execute the eight stages in their canonical order:

$$z_1 \to z_2^{\mathrm{local}} \to H_3 \to H_\theta \to z_2^{\mathrm{global}} \to U_3 \to U_\theta \to \mathrm{Cross},$$

and even rounds execute the fixed permutation

$$z_1 \to H_\theta \to H_3 \to z_2^{\mathrm{global}} \to U_\theta \to U_3 \to z_2^{\mathrm{local}} \to \mathrm{Cross}.$$

The stage order alternates so that over successive rounds each level adapts to current versions of the others. In both orders the stages are sequential within a start: each study anchors on the complete candidate committed by the preceding stage, and the eight independently optimized blocks are never concatenated into an unevaluated configuration. Starts run in parallel with one another; stages within a start do not.

H.2 ONE FIXED-MAP STUDY AND ITS TEMPORARY SEARCH MAP

One study explores the parent through the temporary search map of Appendix H.2:

Algorithm 1 Outer lifecycle of one structured-search run

Require: Frozen simulator and decoder, baseline X^0, metric W, input pair, N_{start} starts, R rounds, structural interval n_{chk}

Create the exact-identity cache and the evaluated archive
Evaluate the baseline once; abort the run if it has no valid score
Initialize each start once by joint proposals around X^0
for $r = 1, \ldots, R$ **do**
  for all starts, in parallel **do**
    for the eight stages in this round's order **do**
      Select the stage's local unit; freeze its study map
      Run one fixed-map study (Algorithm 2)
      Commit the strictly better winner, if any, as the new parent
      Record the parent-to-winner displacement as an inheritable direction; update unit records and next radius
    end for
  end for
  if r is a multiple of n_{chk} and $r < R$ **then**
    for all starts, in parallel **do**
      Construct the available Keep, Grow, and Prune branches
      Score any Prune anchor not yet evaluated
      Run branch studies with identical settings from scored anchors
      Select among the qualified branches and the parent
      Transfer geometry state to the winning architecture
    end for
  end if
  Monitor each start's calibration-best candidate on held-out data (observe-only)
end for
return The best qualified candidate of the entire evaluated archive, together with its architecture, scores, and run record

Algorithm 2 One fixed-map study

Require: Scored parent, frozen stage map, proposal schedule

Import the parent's validated evaluation as a warm-start observation at $u = 0$ (Appendix H.4)
best $\leftarrow$ parent; evaluate the initial Sobol block
for each feedback batch **do**
  if the number of distinct decoded candidates with valid scores in this study is below the declared threshold **then**
    Propose the next batch from the continuing Sobol sequence
  else
    Propose the next batch from the local policy and TPE in the same frozen u-space (Appendix H.5)
  end if
  Evaluate the batch in parallel and return terminal feedback in stable order
  Update best if a strictly better valid candidate appears
end for
return best and study diagnostics

The key invariant is that within one study the four components of the map are frozen:

$$Z_c, \qquad \lambda, \qquad D, \qquad \rho \tag{H.1}$$

do not change while the study runs, even if a better proposal appears midway. A better candidate is tracked as the incumbent, but state changes are committed only at the study boundary. This is what keeps the temporary coordinates interpretable: if the anchor, directions, scales, or radius changed mid-study, the same u would point to different candidates at different times and all returned feedback would lose its meaning.

With parent coordinates Z_c, per-coordinate scales λ, direction matrix D with d_u columns, and radius ρ, proposals are

$$Z(u) = Z_c + \rho \, \mathrm{diag}(\lambda) \, Du, \qquad u \in [-1, 1]^{d_u}, \tag{H.2}$$

Each row of the assembled D is divided by the larger of one and its absolute row sum, so that sweeping u over the box moves no coordinate by more than ρ scale units; for the second level this normalization is applied to the contrast coordinates before multiplication by $C_A^\top$, not to the decoded asset scores. After normalization the columns of D are checked for effective rank; dependent directions are regenerated up to a declared attempt limit, and if the set is still rank-deficient d_u is reduced to the actual number of independent directions.

Because the searched coordinates multiply into factor products, coordinate movement and actual change in the decoded population are not the same thing. The map therefore evaluates the exact products, including cross terms: for any residual $E = UH$,

$$\Delta E = (U + \Delta U)(H + \Delta H) - UH = \Delta U \, H + U \, \Delta H + \Delta U \, \Delta H, \tag{H.3}$$

and dropping the last term would understate joint core–loading moves. The pre-materialization change relative to the anchor is measured by

$$d_{\mathrm{pre}} = \max\big\{ \|\Delta z_1\|_\infty, \ \|\Delta E_2\|_\infty, \ \|\Delta E_3\|_\infty / c_3, \ \|\Delta E_\theta\|_\infty / c_\theta \big\}, \tag{H.4}$$

where $\| \cdot \|_\infty$ is the largest absolute entry, the first two blocks use their natural logit units, and $c_3, c_\theta > 0$ are the declared residual scales shared with Appendix H.9. A proposal is legal only if $d_{\mathrm{pre}} \le 1$, all coordinates are finite, and the complete typed population decodes legally under the rules of Appendix F. If the raw step violates the bound, the same u is retried with its increment multiplied by 2^{-k} for $k = 0, \dots, k_{\max}$ in order, and the first legal result is used; if none is legal, the position is recorded as an invalid proposal without simulation. This deterministic backtracking is part of the fixed map—the sampler sees a fixed function from u to score—and consumes no market simulations.

H.3 Stage units and local-group scheduling

Each stage owns a family of local units, defined on stable identities: groups of endowment coordinates by trader; groups of trader-side second-level coordinates—rows of B_2 in full-contrast mode or of U_2 in low-rank mode; single-asset slices of the strategy core H_3; asset–strategy blocks of the parameter core H_θ retaining all fields of one strategy type; and groups of trader rows of the loadings U_3 and U_θ. The selected unit determines which coordinate directions dominate the study's direction set; family-wide random directions still give unselected coordinates a chance to move, so an unselected group is not frozen.

Units are scheduled by two alternating criteria with stable tie-breaking: odd rounds select the least-visited unit, even rounds select the unit with the highest stored improvement-per-proposal record, and ties are broken by visit count and then by stable unit identity. After a study on unit g, its record is updated as an exponential moving average,

$$I_g \leftarrow \tfrac{1}{2} \, I_g + \tfrac{1}{2} \, \frac{\max\big(0, \widehat{J}_{\mathrm{entry}} - \widehat{J}_{\mathrm{best}}\big)}{n_{\mathrm{proposed}}}, \tag{H.5}$$

where the denominator counts all proposed positions of the study, including duplicates and cache hits, so that heavy cache reuse cannot inflate the record. The global second-level stage and Cross keep their own stage state and do not attribute their gains to any local unit. This record is a scheduling heuristic under one fixed input pair, and a finite run need not visit all units.

H.4 DIRECTION CONSTRUCTION AND CROSS-STUDY INHERITANCE

The direction matrix of a study mixes three sources in declared proportions, with the recipe depending on the stage. *Local coordinate directions* move coordinates inside the unit selected for the current stage, drawn without replacement under the optimization random stream. *Inherited improvement directions* reuse what earlier studies have learned: whenever a stage commits a strictly better winner, the parent-to-winner displacement, normalized by the per-coordinate scales λ and to unit length, is recorded under its variable family, and each stage retains its most recent accepted directions up to a declared depth. When a new map is assembled, candidate history is examined in a fixed recency order and admitted up to a declared count, subject to a compatibility filter: a direction is used only if its shape and rank match the current architecture, its stored coordinate prefix covers the currently open coordinates, it remains finite and nonzero after projection into the current map, and it is responsive and linearly independent of the already admitted directions. Directions may also be borrowed across compatible stages—between the local and global second-level stages, between the core and loading stages of one factorized level, and between any ordinary stage and Cross—while a borrowed direction never records an improvement for the borrowing stage: only a stage's own parent-to-winner displacement enters its history. *Fresh random directions* cover the whole variable family of the stage, so that exploration is not restricted to the selected unit. The local second-level stage is the one exception to inheritance: to preserve its strict sparse-axis semantics of Appendix H.5, it admits no dense inherited directions, although it still warm-starts from the parent, keeps its group records and radius, and contributes its own improvements to the shared history.

Two stages have distinct recipes. The global second-level stage uses no local coordinate block: its directions are inherited improvement directions plus fresh market-wide directions in the second-level coordinate block z_2. The Cross stage combines per-level inherited directions, one fresh direction per level, and joint directions that move all four levels together; joint directions are normalized layer by layer before the common legality check of Appendix H.2. When a level's core is zero or its loading directions have degenerate residual response—as at initialization or after a Grow expansion—loading studies construct fixed joint core–loading activation directions before the map is frozen, so that candidates can produce nonzero products at all; whether the decoded population actually changes is decided by the identity check of Appendix H.7.

Each study opens by importing one observation it already owns: the parent's validated evaluation, registered as a completed warm-start observation at $u = 0$, the center of the temporary search space. The import is conditional on provenance: the parent must carry a finite score from an eligible evaluation source, and decoding $u = 0$ must reproduce the complete physical identity of that evaluation—the same decoded population under the same initialization, simulator, metric, and input pair. The warm-start observation creates no new candidate and consumes no simulation; it lets the new study begin from one real observation whose coordinates it understands. Initialization studies, the Keep/Grow/Prune structural studies of Appendix H.9, and held-out evaluations do not import warm starts.

The study then opens with a block of scrambled Sobol positions (Owen, 1997), which covers the box evenly without using any feedback; the scramble seed is fixed per start so that runs are reproducible and starts differ. Batches continue along the same Sobol sequence until the study contains the declared minimum number of distinct valid observations, *including the warm-start observation*; only then do subsequent batches come from the local policy and TPE. The switch condition counts distinct decoded identities, not proposals: duplicate identities do not advance the threshold.

TPE uses the scored observations from the current fixed map to propose subsequent positions (Bergstra et al., 2011). Only positions actually proposed within the current fixed map train the sampler, together with the warm-start observation at the origin; observations from a different anchor, basis, or rank are not imported, since their u coordinates would not agree with the current map. Within a batch, positions are asked sequentially with pending positions marked for the declared constant-liar mechanism, evaluated in parallel, and reported back in proposal order rather than completion order. Failed or invalid results are recorded as such. The parent is retained on a score tie, and ties among better proposals are broken by stable proposal order.

H.5 PAIRED LOCAL SEARCH IN THE SECOND-LEVEL STAGE

Within a mature batch, a declared quota of positions is reserved for local proposals around the current best point; the remaining positions come from TPE. For most stages the local policy proposes bounded coordinate moves around the incumbent within the frozen map, with a shared step size adapted from batch-level feedback. The local second-level stage uses a stricter paired policy with three distinctive components.

First, the entire study is restricted to the selected trader group: the map's support contains only the second-level coordinates of the chosen group—rows of B_2, plus the cash-fraction coordinates when that sub-block is calibrated—and a declared number of sparse coordinate axes is drawn without replacement from this support under a seed bound to the study identity. The restriction applies to Sobol and TPE positions as well, so no proposal of the study modifies another trader's allocation; in low-rank mode the shared asset loadings stay fixed, which confines the change to the selected rows without confining it to a single asset.

Second, local proposals come in signed pairs around the current best center c: along axis j with requested step h_j, the pair is $u_\pm = c \pm h\, e_j$ with $h = \min(h_j,\, 1 - |c_j|)$, so both members share one box-feasible amplitude. Before any trial is created, a bounded decode-only pre-check decodes the center and both members and requires successful decoding, an equal map-backoff count for all three points, distinct physical identities not already scored in this study, and a bounded physical change: the decoded asset-allocation and cash-fraction rows may move by at most a declared absolute amount per entry. The pre-check runs no simulation and creates no candidate; axes that exhaust their planning attempts fall back to TPE.

Third, step-size feedback is attributed per axis and only to complete pairs. When both members become valid observations, the pair is compared against the center's score: a significant improvement keeps the step and advances a consecutive-success counter that expands the axis step after repeated successes; a complete but non-improving pair halves the axis step and retries the same axis once; a pair with a failed or duplicate member is uninformative and moves the cursor to the next axis without changing any step. All valid observations—including individual pair members—still update the incumbent and train TPE.

H.6 RADIUS ADAPTATION

Each stage of each start keeps its own radius state, updated only at study boundaries and always clipped to a declared range. The rules, in priority order, are:

1. *Strict improvement*: the next radius of the stage is widened by a fixed factor, and the non-improvement counter is reset.
2. *Decode-back-to-parent*: if at least half of the proposed positions were legal but decoded to the parent, the directions are regenerated and the radius is widened rather than shrunk—a discrete plateau of the quantization or clipping maps is not evidence that the neighborhood is exhausted.
3. *Evaluated but not improving*: if the study produced novel, legal, successfully scored candidates but no strict improvement, a non-improvement counter is incremented; after a declared number of such visits the radius is shrunk by a fixed factor and the counter resets.
4. *No usable evidence*: otherwise—mostly invalid, degenerate, or failed evaluations—the radius and counter are held, and the reason is reported separately.

Structural branch comparisons start from one common declared radius, so no branch is advantaged by an inherited step size. An accepted rank change resets the radii of the affected stages and the geometry state of Cross; visit counts defined on stable trader or asset identities remain meaningful and are kept, while records tied to old latent-axis numbering are rebuilt. The widening factors, bounds, and thresholds are fixed configuration constants of the profile.

H.7 CANDIDATE IDENTITY, DUPLICATES, AND FAILURES

Candidate equality is defined by the complete decoded population under the same initialization, simulator, fact metric and simulation-input pair. Different representations of that population reuse

its evaluation; approximate equality is not a duplicate criterion. A duplicate consumes a proposal position without adding a new simulation pair or advancing the distinct-valid-candidate threshold of Appendix H.4. A failed evaluation remains a failure: it receives no surrogate score and is not retried with replacement seeds. A resumed run continues the declared evaluation rather than starting an additional pair. The implementation identities and replay records are specified in Appendix J.7.

Three counters are kept distinct throughout. The *proposal budget* counts the new positions a study may propose, split between the opening Sobol block and the mature batches. *Sampler trials* additionally include the warm-start observation, which occupies a trial record but no proposal position and no evaluation. *Realized simulations* count the candidate evaluations actually run, which fall short of the proposal count by cache hits, duplicates, decode-only rejections, and failures. The numerical budget of each study type is reported in Appendix J.

H.8 Multi-start and held-out monitoring

All starts share the same legal baseline X^0, and each initializes exactly once: its factor blocks are drawn so that the initial coordinates decode to X^0—nonzero orthonormal loadings paired with zero cores, which also motivates the activation directions of Appendix H.4—and a fixed number of joint Sobol proposals, including core directions, is evaluated around the baseline. Each start then keeps the best of the baseline and its own proposals, with ties retaining the baseline; a start whose proposals all fail simply begins from the baseline. Candidate construction reads no scores, so the baseline and all initialization proposals can be scheduled together.

Thereafter the starts are fully independent trajectories. Each maintains its own parent, direction history, radii, unit records, and proposal state; starts do not exchange incumbents or searcher state, though two starts that converge to the same decoded population share its cached score. The optimization random streams are independent across starts, while the simulation-input pair is shared by the whole run. Across rounds, each start continues from everything it has learned—accepted candidates extend the relevant direction history, and unit visit and improvement records accumulate—subject to the rank-change resets of Appendix H.6.

Held-out monitoring is attached to this loop but outside the search. At the end of every round, each start's current calibration-best candidate is evaluated on held-out data immediately following the calibration window: each of the two calibration paths continues from its calibration-terminal market and account state with the population parameters frozen, and the next calibration round does not inherit the held-out terminal state. This evaluation is observe-only: held-out results never feed into proposals, branch comparison, structural selection, or the final calibration choice, which remains the best candidate retained by the calibration search on calibration-window scores.

H.9 Structural branch comparison protocol

At fixed round intervals the search compares three branches—Keep, Grow, and Prune—from the same scored parent, and the winning branch becomes the parent of the next round. Structural comparison asks whether the representation capacity of the factorized third and fourth levels should change, and it compares realized scores after a finite search, not theoretical optima of the competing architectures. At each declared checkpoint, all available branches start from the same scored parent, receive identical study settings and one common initial structural radius, and are run to completion before any verdict; branches do not inherit one another's results within a check, and the parent itself remains eligible throughout.

Availability is explicit: Grow is unavailable when both adaptable ranks are at their declared maxima, and Prune is unavailable when both are at their declared minima; an unavailable branch is skipped, reducing the planned positions of the check. If the Prune anchor fails to decode legally or to score, only Prune is skipped—Keep and Grow still run.

H.10 Structural branches: Keep, Grow, and Prune

Keep preserves the current ranks and runs its branch study from the parent anchor using cross-layer directions, jointly refining the existing coordinates. Keep is therefore an active competitor that invests the same search effort in the current architecture, and its best qualified candidate competes

with those of the other branches. The original parent remains in the common final comparison regardless.

Grow increases one rank of one factorized level. The target level alternates deterministically: at structural check j of start m, even $j+m$ first targets the third level and odd $j+m$ first targets the fourth; if the favored level is already at its declared maximum, the other level is tried instead. The selected rank grows by the fixed proposal rule $r^+ = \min(2r,\, r_{\max})$, doubling toward the declared cap $r_{\max}$ of Appendix J.

The enlarged representation is constructed to reproduce the parent exactly. With current loadings U and core H of the selected level, new loading columns U_{new} are drawn from the orthogonal complement of the current loading span, the old columns are kept entry-wise, and the new core slices are zero:

$$U^+ = [U,\, U_{\text{new}}], \qquad H^+ = \begin{bmatrix} H \\ 0 \end{bmatrix}, \qquad U^+H^+ = UH. \tag{H.6}$$

The embedding is verified at the decoded population: the expanded candidate must materialize to exactly the parent's X, and if it does not, the branch stops. During the Grow study the new loadings are frozen and the directions are split between the new core slices and cross-layer refinement of the old coordinates, so the added capacity gets a direct opportunity to contribute; the new loadings become trainable by the ordinary loading stages only after the branch is accepted.

A Grow candidate is qualified for structural selection and for the global archive only if its decoded population differs from the parent's and its new contribution is nontrivial,

$$\|E_{\text{new}}\|_F > \epsilon_{\text{act}}\, \max\!\big\{c_v \sqrt{D_v^{\text{res}}},\, \|E_{\text{parent}}\|_F\big\}, \qquad E_{\text{new}} = U_{\text{new}} H_{\text{new}}, \tag{H.7}$$

where $D_3^{\text{res}} = NA(S-1)$ and $D_\theta^{\text{res}} = NAP$ are the residual-table sizes of the two factorized levels, c_v the declared residual scale of level $v \in \{3, \theta\}$, and ϵ_{act} a small declared threshold. Unactivated expansions remain diagnostics: their score improvements, achieved through the old coordinates, are not attributed to the larger rank. If the empty columns can be removed while the decoded population stays exactly unchanged, the candidate is recoded and its score reused under the lower architecture's identity.

Prune tests whether a simpler representation suffices. For each eligible level, the trader-unfolded residual matrix is truncated from rank r to $r-1$ by a singular value decomposition, keeping the leading $r-1$ terms; by the Eckart–Young theorem this is the best rank-$(r-1)$ approximation of the current residual in Frobenius norm, a statement about the residual matrix rather than the market score. When both levels are eligible, the branch nominates the level with the smaller normalized discarded-tail cost

$$A_v = \frac{\tau_v}{D_v^{\text{res}}\, c_v^2\, \Delta d_v}, \tag{H.8}$$

where τ_v is the discarded squared singular-value tail of level $v \in \{3, \theta\}$ and Δd_v the persistent coordinates the truncation would save—$\Delta d_3 = N + A(S-1)$ and $\Delta d_\theta = N + AP$; exact ties nominate the third level. The truncated factors define a deterministic lower-rank anchor, with all other levels unchanged.

The anchor is then fully legalized and scored on its own two paths, and—unless invalid or failed—it receives its full branch study even when its anchor score is worse than the parent's, since branches are compared by what their search achieves, not by how attractive their starting score looks. Pruning removes latent directions only; traders, strategies, accounts, cash, and holdings are untouched. Whether the simpler candidate is accepted is decided by the market score, not by the tail energy that nominated it.

H.11 Structural verdict, state transfer, and final output

The structural verdict compares the parent and the qualified best candidates of all completed branches lexicographically by

$$\big(\widehat{J},\ d_Z,\ \mathrm{id}_X\big), \tag{H.9}$$

where d_Z is the number of persistent coordinates and id_X a stable candidate identity. Complexity enters only to break exact score ties: an equal-score Prune can displace the parent, while an equal-score Grow cannot, and no complexity preference trades worse fit for a smaller representation. Because the parent is always retained in the comparison, the accepted realized score is non-increasing across checks.

An accepted branch change transfers state according to what still means the same thing. The decoded parent, its evaluation, and the entire evaluated archive are always retained under their complete-population identities. Within the winning architecture, the affected level's factor coordinates, direction history, and improvement records are rebuilt, because they are tied to the old latent-axis numbering, and the Cross stage's geometry state is reset; unit visit counts defined on stable trader or asset identities remain meaningful and are preserved. Levels whose ranks did not change carry their state over untouched. The next round's eight stages then proceed on the selected architecture, refining the actually evaluated winning candidate.

The run retains every evaluated candidate as an auditable archive, with its architecture, coordinates, two path-fact records, signed cross score, diagnostics, and the proposal-to-identity ledger of how it arose. The final result is the best qualified candidate of the entire archive under the ordering of (H.9)—not merely the terminal endpoints of the starts—and the held-out scores of Appendix H.8 play no role in this selection. Final ranking and serialization reuse the cache and launch no new simulations.

I Analytical Properties and Scope

This appendix states properties of the residual representation, the continuous decoding maps, and the calibration score. Appendices I.1 and I.2 concern approximation of a specified target residual; Appendix I.3 transfers residual errors through the continuous decoder; and Appendix I.4 establishes the fixed-candidate expectation property of the cross score.

I.1 Centered matrix approximation

Proposition (centered matrix approximation). Let $E\mathbf{1} = 0$ be a centered residual matrix with singular values $\sigma_1 \geq \sigma_2 \geq \cdots$, and let E_r be its rank-r truncated singular value decomposition. Then

$$E_r\mathbf{1} = 0, \qquad \|E - E_r\|_F^2 = \sum_{j>r} \sigma_j(E)^2 = \min_{\mathrm{rank}(B)\leq r} \|E - B\|_F^2. \tag{I.1}$$

The error identity is the Eckart–Young theorem (Eckart & Young, 1936). Compatibility with the second-level decoder is direct: every right singular vector with nonzero singular value lies in the row space of E and is therefore orthogonal to $\mathbf{1}$, so the truncated residual remains centered and can be factorized in the contrast coordinates of Appendix F.3. This is a reachability statement about a given residual; it says nothing about the residual rank of an unknown market population.

I.2 Partial Tucker approximation

Proposition (partial Tucker approximation). For a residual tensor $E \in \mathbb{R}^{N\times A\times d_{\mathrm{last}}}$, where d_{last} is the final-mode size, let P_N and P_A be the orthogonal projectors onto the leading left singular subspaces of its trader- and asset-mode unfoldings with ranks r_N and r_A, and set $\widetilde{E} = E \times_1 P_N \times_2 P_A$. Then

$$\|E - \widetilde{E}\|_F^2 \leq \underbrace{\sum_{j>r_N} \sigma_j\big(E_{(1)}\big)^2}_{\tau_N} + \underbrace{\sum_{j>r_A} \sigma_j\big(E_{(2)}\big)^2}_{\tau_A} \leq 2\,\|E - E_{\mathrm{best}}\|_F^2, \tag{I.2}$$

where E_{best} is the best approximation with the same multilinear rank bounds. This is the partial HOSVD bound of De Lathauwer et al. (2000): the error decomposes into the orthogonal terms $(I - P_N)E$ and $P_N E(I - P_A)$, each projection is non-expansive, and each tail is bounded by the corresponding squared approximation error via (I.1). The set of tensors with these unfolding-rank bounds is closed, so a best approximation exists, though the projected approximation need not itself attain it.

Fixing the asset projector to the identity—the profile of Appendices F.6–F.7—removes the asset tail τ_A and reduces the bound to exact trader-unfolding matrix truncation under (I.1). The bound concerns a residual known before projection; it says nothing about the spectral tail of an unknown true market residual or about global optimality of the factor search.

I.3 Decoder stability

Proposition (decoder stability). The softmax map $p(x)$ is $\frac{1}{2}$-Lipschitz: for score vectors x, y,

$$\|p(x) - p(y)\|_2 \le \tfrac{1}{2}\|x - y\|_2, \qquad D_{\mathrm{KL}}\big(p(x) \,\|\, p(y)\big) \le \tfrac{1}{4}\|x - y\|_2^2. \tag{I.3}$$

Its Jacobian $\mathrm{diag}(p) - pp^\top$ has maximum absolute row sum $\max_i 2p_i(1 - p_i) \le \frac{1}{2}$, and integration along the score segment proves the first bound; the second follows from the same Hessian bound applied to the log-partition identity $D_{\mathrm{KL}}(p(x)\|p(y)) = A(y) - A(x) - \nabla A(x)^\top (y - x)$ with $A(x) = \log \sum_i e^{x_i}$. Appending a zero reference score preserves reference-logit difference norms, so the bound applies to the first- and third-level decoders; the sigmoid map of optional cash coordinates is $\frac{1}{4}$-Lipschitz by its scalar derivative bound.

For the fourth level, clipping is non-expansive, so before field quantization the restored parameters satisfy

$$|\theta_{iap} - \theta'_{iap}| \le (b_p - a_p)\,|E_{\theta,iap} - E'_{\theta,iap}|, \tag{I.4}$$

which converts normalized-field residual error back into physical units. These continuous bounds do not extend through discretization: integer or lot quantization is not globally Lipschitz, since for a floor quantizer with step q only $|q\lfloor x/q\rfloor - q\lfloor y/q\rfloor| \le |x - y| + q$ holds in general, and arbitrarily close inputs can cross a threshold. Combining (I.1)–(I.4) therefore controls decoded continuous quantities in terms of residual tails, but not typed, quantized executions.

I.4 Fixed-candidate expectation property

Proposition (fixed-candidate target). Suppose the facts are defined almost surely with finite second moments, and the candidate X is fixed independently of the evaluation inputs. Then

$$\mathbb{E}\,\widehat{J}_{n_{\mathrm{rep}}}(X) = J(X), \tag{I.5}$$

and in particular the equality already holds at $n_{\mathrm{rep}} = 1$.

The derivation is a four-term expansion. Write $e_a = \mu_a(X) - m_a^{\mathrm{obs}}$ and $\eta_a^{(r)} = m_a(X; \xi^{(r)}) - \mu_a(X)$, so that $\Delta_a^{(r)} = e_a + \eta_a^{(r)}$ with $\mathbb{E}\,\eta_a^{(r)} = 0$. Then $(\Delta_a^{(1)})^\top W \Delta_a^{(2)} = e_a^\top W e_a + e_a^\top W \eta_a^{(2)} + (\eta_a^{(1)})^\top W e_a + (\eta_a^{(1)})^\top W \eta_a^{(2)}$; the middle terms vanish in expectation because the fluctuations are centered, and the last vanishes because the two paths are independent. No independence between assets within a path is required. This is independent cross-product estimation of a quadratic mean discrepancy, a principle also used in unbiased discrepancy estimators (Gretton et al., 2012).

The property explains why the ordinary squared score is not the objective: for a single path,

$$\mathbb{E}\,\big\|m_a(X;\xi) - m_a^{\mathrm{obs}}\big\|_W^2 = \big\|\mu_a(X) - m_a^{\mathrm{obs}}\big\|_W^2 + \mathrm{tr}\big\{W\ \mathrm{Cov}[m_a(X;\xi)]\big\}, \tag{I.6}$$

and the added variance term generally varies with the candidate, so a squared single-path score can reorder candidates simply because one of them is noisier. Squaring the average of two paths retains half of this extra term. Equivalently, the cross score decomposes as

$$\widehat{J}(X) = \sum_{a=1}^{A} \nu_a \left[\left\| \frac{\Delta_a^{(1)} + \Delta_a^{(2)}}{2} \right\|_W^2 - \frac{1}{4} \left\| \Delta_a^{(1)} - \Delta_a^{(2)} \right\|_W^2 \right], \tag{I.7}$$

in which the pathwise squares and the disagreement term are diagnostics only.

The property is pointwise: it concerns a candidate fixed independently of the evaluation inputs, and does not assert unbiasedness or any error bound for the adaptively selected winner of the search, which minimizes a realized random surface and tends to select luckily low scores.

I.5 Scope

These properties concern a specified residual and the continuous decoder, not the identification of a unique population or the fit of its simulated market: discrete decisions, quantization, and order matching can change the trajectory even when decoded perturbations are small. The search compares realized scores; parent retention makes accepted scores non-increasing under the frozen input pair, which is a protocol property rather than monotonic improvement in J. Post-selection evidence is assessed through the distinct evaluation channels of Appendix G.6.

J Experimental Binding and Reproducibility

This appendix records the simulation setting, the calibration profile and their reproducibility information. The coordinate and schedule calculations are arithmetic consequences of the stated numerical search profile.

J.1 Simulation and population configuration

The simulation binding identifies the market institution and broader setting κ: assets and strategy catalogue, clock and sessions, order types and price rules, fees, reservations, settlement, aggregation, lot rounding, matching priorities, initialization and random-input distribution. The strategy-external profile identifies the selected information sources and the operators of Appendix D.

The subscripted ρ_{seis} and λ_{seis} are the information-mechanism scalars; the unsubscripted ρ and λ in the paper retain the search meanings of Appendix H.

For the Branch B profile, X^0 and the cash-fraction table c_2^0 are fixed before calibration. Initial accounts are materialized by the rules of Appendix B.1; these inputs do not vary between candidates. The strategy-field schema supplies the ranges, types and joint constraints used by the decoder. The value $P = 52$ in Table J.2 belongs to the stated numerical profile.

Table J.1: Simulation configuration for the specified realization.

Simulation component	Setting
Simulation clock	09:14:00–15:10:00 on each simulated trading day.
Information realization	Endogenous mode: lagged simulation behavior is the information source.
Relation	Cross-asset Euclidean distance, exponential affinity, tie-inclusive Top-K; $K = 16$ ($N \geq 17$ for this profile).
Diffusion	Source and retained-belief propagation; $\rho_{\text{seis}} = 0.5$.
Projection	Eligible-mass-preserving mixture; $\lambda_{\text{seis}} = 1$.
Projection clip and strength	$\delta_{\max} = 0.25$, $\eta_\pi = 0.10$.
Initial information state	$U_0^{\text{beh}} = B_0^{\text{seis}} = 0$.
Signal freshness bound	`signal_max_age_sim_seconds` $= 30$ s of simulation time; older signals abstain (Appendix C.7).

J.2 Empirical data and price-grid conventions

The empirical extraction uses the following conventions:

1. Keep valid positive-price, positive-integer-quantity trades, excluding the source's cancellation code C. Order records by time, trade identifier, then original row order.
2. Use 242 minute price points: 09:30–11:30 and 13:00–15:00. Take the last valid trade price and total quantity in each minute. At 11:30 and 15:00 the current convention includes only trades at the exact endpoint. Empty minutes carry the previous known price and have zero volume; a missing opening price is not filled from the future.
3. Form 241 adjacent within-day returns, including the 11:30-to-13:00 change. Overnight returns and returns crossing two trading days are excluded.
4. Infer the day's reference from the latest available earlier closing trade summary, taken from 09:15 through 15:00:59.999. Process reference lookup before updating the current day's summary. Up to 20 preceding file dates initialize this state without contributing reference-window facts.
5. Exclude a stock-day with no reference, a reference older than 30 calendar days, an incomplete price grid, insufficient traded minutes, or non-finite daily facts. All five dimensions share the same validity decision. The current minimum traded-minute count is one with an observed opening minute. Early closing-reference times and opening gaps above 15% are quality flags rather than automatic removals.

J.3 Scoring targets and evaluation periods

Observed targets and simulated facts use the daily definitions and period aggregation of Appendix G.2. The calibration assets are the five A-share stocks 000002 (Vanke A), 000063 (ZTE), 600028 (Sinopec), 600519 (Kweichow Moutai) and 601318 (Ping An Insurance). The empirical reference pool used to construct W covers 2022–2025.

The calibration horizon comprises five complete trading days, 2024-01-02 through 2024-01-08. Held-out monitoring continues the calibration-terminal state for three further trading days under the same two-path protocol. Asset weights ν_a are initially equal across the five assets and are held fixed across candidate evaluations.

J.4 Empirical metric artifact and source binding

The metric uses $\alpha_f = 0.30$, $\epsilon_f = 10^{-6}$ and $Q_f = I_{K_f}$ in (G.7). The variance fallback of the metric construction was not activated.

The adopted metric construction uses Shanghai main-board trade records with code prefixes 600, 601, 603 and 605 over 2022–2025. It contains $n_{\mathrm{hist}} = 969$ trading dates and $h_{\mathrm{ref}} = 5$; the 20-date pre-read is separate. Of 1,762 candidate codes, 1,623 satisfy the declared day/window coverage criteria. A fixed hash ordering with stock-sampling seed 20260913 selects $A_{\mathrm{ref}} = 1{,}500$ assets. Eligibility requires at least

$$n_a^{\mathrm{valid}} \geq \max\{60, \lceil 0.8 n_{\mathrm{hist}} \rceil, h_{\mathrm{ref}} + 1\} = 776,$$
$$n_a^{\mathrm{win}} \geq \max\{60, \lceil 0.8(n_{\mathrm{hist}} - h_{\mathrm{ref}} + 1) \rceil\} = 772.$$

The selected pool supplies 1,438,648 complete five-day windows. Reference-pool weighting is $1/1500$ and does not determine ν_a.

The metric used in the score is the full-precision matrix stored in the frozen metric artifact, with rows and columns ordered as $(f_1, \ldots, f_5)$ from (G.3); the artifact specifies no horizon rescaling. The trading calendar is reconstructed from the trade records themselves rather than from an official exchange calendar. Official historical non-ST status and reference prices were unavailable in this construction, so the fifth fact uses a band proxy computed from inferred closing prices in place of the official reference price. These source conventions delimit the interpretation of W.

Table J.2: Numerical binding of the calibration profile. Braced values list the experiment grid; the sensitivity profile changes only the entries marked †.

Quantity	Value
Traders N	$\{256,\ 512,\ 1024\}$
Assets A	5
Strategy types S	7
Parameters per strategy instance P	52
Initialization branch (Appendix B.1)	Branch B: wealth split, frozen cash-fraction table c_2^0
Initial trader ranks (r_3, r_θ)	$(8, 4)$
Allowed ranks	$2 \le r_3 \le 32$, $2 \le r_\theta \le 32$
Axis modes	second level full contrast; third/fourth identity asset axis
Starts N_{start}	$\{2,\ 4\}$
Rounds R	$\{2,\ 3,\ 4\}$
Initialization proposals per start n_{init}	8, in addition to the shared baseline
Ordinary study n_0, Q, b	$8,\ 6,\ 4$ ($Q = 4$ †)
Structural branch n_{0s}, Q_s, b_s	$8,\ 6,\ 4$ ($Q_s = 4$ †)
Structure-check interval n_{chk}	1, excluding the final round
Temporary direction count d_u	16 (geometry sensitivity: 8)
Warm start (ordinary studies)	parent evaluation imported at $u = 0$ after provenance and physical-identity verification
TPE switch threshold	8 distinct valid observations in the current study, including the warm-start observation
Local quota per mature batch	$\min(2, \lfloor b/2 \rfloor)$; remaining positions from TPE
Metric regularization	correlation shrinkage $\alpha_f = 0.30$; eigenvalue floor 10^{-6}; $Q_f = I$
Paths per unique candidate	2, fixed for the run
Held-out monitoring	enabled; 3 continuation days; inherits the two-path scoring

J.5 Calibration configuration

The baseline population and initial account inputs are held fixed across candidate evaluations. The reported run starts at the declared initial ranks $(r_3, r_\theta) = (8, 4)$, and no rank change is accepted during the run: the searched representation remains the $5{,}631$-coordinate profile of Table J.3 throughout. The metric W is the full-precision matrix identified in Appendix J.4. The coordinate scales of Appendix H.2 are: 1 for endowment and contrast coordinates, $N^{-1/2}$ for trader loadings, $\sqrt{N/r_3}$ for strategy-core entries, and $0.1\sqrt{N/r_\theta}$ for parameter-core entries; these precede the exact residual trust check and do not by themselves bound final account changes.

Local-unit sizes, the per-stage direction allocation at $d_u = 16$, and the sparse-axis policy of the local second-level stage follow the stage definitions of Appendix H.

J.6 Coordinate, path, and computational accounting

The materialized population stores

$$d_X^{\text{slots}} = N + NA + NAS + NAP \tag{J.1}$$

typed slots. Discounting only the simplex redundancies, the dense free-coordinate count under the fixed-cash profile is

$$d_{\text{dense}} = (N-1) + N(A-1) + NA(S-1) + NAP. \tag{J.2}$$

Under the identity-asset profile of Appendix F, the searched persistent coordinates number

$$d_Z = (N-1) + N(A-1) + r_3\big(N + A(S-1)\big) + r_\theta(N + AP), \tag{J.3}$$

while the optional low-rank second-level representation of Appendix F.4 and learned third/fourth-level asset axes would give the general form of the technical design; the calibrated-cash variant of Branch C adds NA direct logit coordinates to the second level. Fixed baselines and contrast bases require storage but are not searched, and factor-gauge redundancy is not subtracted from these explicit-coordinate accounts. Each additional strategy rank costs $N + A(S-1) = 286$ coordinates and each additional parameter rank costs $N + AP = 516$.

For scale, the dense count (J.2) at this size is $75{,}519$ and the materialized slots (J.1) number $77{,}056$. As an accounting illustration of how the representation scales with the asset dimension, at $A = 1000$ with N, S, P unchanged the dense count would be $15{,}103{,}999$; the main profile at the reported ranks $(r_3, r_\theta) = (8, 4)$ would search $d_Z = 515{,}071$ coordinates, about 3.4% of dense, and enabling the optional scalable second level of Appendix F.4 at $r_2 = 8$ would further reduce this to $269{,}375$, about 1.8%. These are deterministic accounting identities at the stated ranks; adaptive rank changes move the searched count within the ranges of Table J.2.

One candidate evaluation means one unique decoded identity scored on two complete simulation paths. Write $n = n_0 + Qb$ for the positions of one ordinary study, $n_s = n_{0s} + Q_s b_s$ for a structural branch study, and $J_c = \lfloor (R-1)/n_{\mathrm{chk}} \rfloor$ for the number of structure checks. With b_j available branch studies and a_j Prune anchors at check j, the scheduled maximum number of candidate positions is

$$N_{\mathrm{pos}} = 1 + N_{\mathrm{start}} n_{\mathrm{init}} + N_{\mathrm{start}} RLn + \sum_{j=1}^{J_c} \big(b_j n_s + a_j\big), \tag{J.4}$$

where $L = 8$ is the number of stages. With all three branches available and one Prune anchor per start, this is $1 + N_{\mathrm{start}} n_{\mathrm{init}} + N_{\mathrm{start}} RLn + N_{\mathrm{start}} J_c (3n_s + 1)$. The warm-start observation of each ordinary study occupies a sampler trial but adds no position and no path to this ledger. The no-cache, no-failure plan is $N_{\mathrm{sim}}^{\mathrm{plan}} = 2N_{\mathrm{pos}}$ complete paths, and the actual launched-path count satisfies

$$N_{\mathrm{sim}}^{\mathrm{actual}} = \sum_{x \in \mathcal{U}} \sum_{r=1}^{2} \mathbf{1}\{\text{path } (x, r) \text{ launched}\} \le 2\,|\mathcal{U}| \le 2N_{\mathrm{pos}}, \tag{J.5}$$

where $\mathcal{U}$ is the set of identities reserved in the run. Invalid pre-decode positions and cache hits consume positions but launch no paths; a launched path that fails still counts; and no confirmation, held-out, or repeated-parent path is hidden in this ledger.

The sensitivity profile $Q = Q_s = 4$ gives 24 positions per study, 931 positions, and 1,862 planned paths.

Held-out monitoring adds $N_{\mathrm{start}} \cdot R = 4$ continuation evaluations at the reported $N_{\mathrm{start}} = 2$, $R = 2$—one per start at the end of every round—each continuing that start's two calibration paths for three further trading days from their calibration-terminal states under the same two-path protocol.

Table J.3: Persistent-coordinate counts at $N = 256$, $A = 5$, fixed cash fraction. The calibrated-cash variant adds $NA = 1{,}280$ to every row. The ratios use the same-size dense count $75{,}519$. The $(8, 4)$ row is the reported configuration; the remaining rows are accounting references only.

(r_3, r_θ)	d_1	d_2	d_3	d_θ	Total d_Z	d_Z/d_{dense}
$(4, 2)$	255	1,024	1,144	1,032	3,455	4.57%
$(8, 2)$	255	1,024	2,288	1,032	4,599	6.09%
$(4, 4)$	255	1,024	1,144	2,064	4,487	5.94%
$(8, 4)$ (reported)	255	1,024	2,288	2,064	5,631	7.46%

Table J.4: Schedule arithmetic of the calibration profile at $N_{\text{start}} = 2$, $R = 2$ ($Q = Q_s = 6$, all branches available). Every path contains all traders, assets, and simulated days.

Part	Positions	Planned full paths
Baseline and initialization	$1 + 2 \cdot 8 = 17$	34
Round 1, sixteen ordinary studies	512	1,024
Structure check	$2\,(3 \cdot 32 + 1) = 194$	388
Round 2, sixteen ordinary studies	512	1,024
Total	**1,235**	**2,470**

This contributes 8 continuation-path segments totaling 24 simulated path-days. These quantities are reported separately from the calibration ledger of Table J.4, since held-out evaluations are observe-only and play no role in candidate selection.

J.7 Evaluation identity, replay, and reproducibility manifest

Evaluation reuse identifies the complete typed population, including field ordering and initialized accounts, within a namespace fixing the decoder, baseline, initialization policy, simulator/data versions, metric and input pair. A hash is a lookup aid; exact equality determines reuse. Concurrent requests for one identity share its reserved evaluation, and both completed scores and terminal failures are retained.

A simulation checkpoint stores the market and account state, generator states and tick identity. For the information mechanism it additionally stores U_t^{beh} and B_t^{seis}, the population and base-mixture identity, static eligibility and slot ordering, and the relation, diffusion and projection configuration. A checkpoint within a tick also identifies its frozen capacity snapshot. The search checkpoint retains the population, architecture, stage and radius state, proposal order, completed evaluations and in-flight work needed to continue the declared schedule, including the Sobol cursor, the sampler state, and the per-axis step records of the paired local policy.

To reproduce a reported comparison, retain its simulator configuration, baseline and schema, the metric W, the scored data and periods, the search profile, and the random inputs used for optimization and simulation. When SEIS is used, its graph, diffusion and projection settings belong to the simulator configuration. Checkpoints and run identifiers support execution and recovery but are not method parameters.

Optional reported diagnostics. Graph connectivity, belief summaries, eligibility frequencies and runtime-mixture changes can describe the mechanism when an experiment uses them; if reported, their units, aggregation and denominators are stated. Likewise, any fresh-seed validation specifies its independent input streams, repetition count and score aggregation separately.

The coordinate and path counts above apply to the settings in Table J.2; a different search profile recomputes them from its actual settings.

J.8 Finite-window precision and drift diagnostics

The robustness audit in Section 4.4 freezes the baseline population, simulation environment, empirical target, and metric. For a nondegenerate asset–fact pair (a, j), its serial-correlation-adjusted normalized standard error and the corresponding observation requirement are

$$\mathrm{SE}_{a,j}(n) = \frac{\widehat{\sigma}_{a,j}}{s_{a,j}} \sqrt{\frac{\widehat{\tau}_{a,j}}{nS}}, \qquad N_\gamma = \max_{a,j} \left\lceil \widehat{\tau}_{a,j} \left(\frac{\widehat{\sigma}_{a,j}}{\gamma s_{a,j}} \right)^2 \right\rceil,$$

where $S = 4$ is the number of independent paths, n is the number of post-regime days per path, $s_{a,j}$ is the frozen empirical daily dispersion, and $\widehat{\tau}_{a,j}$ is an AR(1) serial-dependence adjustment.

Figure J.1 shows that normalized uncertainty declines with the available window but at fact-specific rates. At 36 days per path, the worst case is 0.096; the simultaneous requirements are $N_{0.25} = 22$

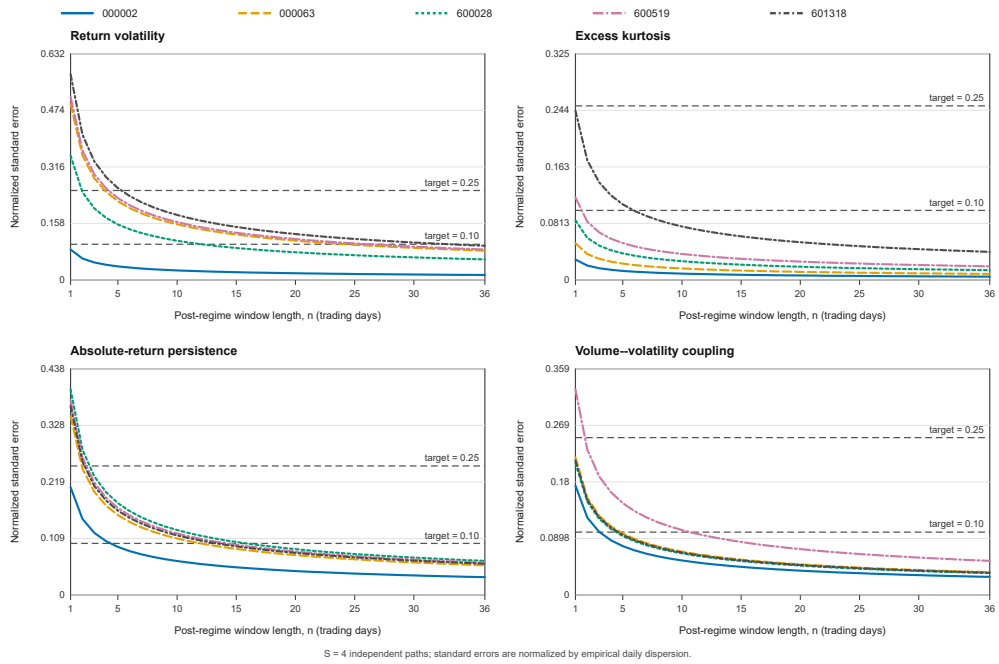


Figure J.1: Monte Carlo precision of window-averaged market facts on the frozen baseline population. Standard errors are normalized by the per-asset empirical daily dispersion and plotted against the number of post-regime trading days per path. Horizontal lines mark $\gamma = 0.25$ and 0.10. Day 1 is excluded as a configuration-defined execution-regime boundary.

and $N_{0.10} = 133$ day observations. These calculations bound Monte Carlo measurement error and do not assert stationarity.

J.9 Population size and volatility scaling

This appendix reports the full population-size grid behind Section 4.5. Total wealth and the strategy mix are fixed; population size varies over $N \in \{64, 128, 256, 512\}$ with three preregistered random paths per size, and each trader's resources are rescaled by $256/N$ so that aggregate wealth is invariant. Evaluation uses post-formation days 2–10 under the same observation code as the main experiments. Filled markers are lot-consistent estimates; the $N = 512$ configuration is constrained by the round-lot rule and is shown only as a sensitivity point, excluded from the slope.

J.10 Stylized-fact profile of the calibrated market

This subsection reports the per-dimension stylized-fact profile referenced by Section 4.5.

K SEIS Ablation

Section 4.3 summarizes the main effect of the Strategy-External Information System (SEIS); this appendix reports the complete fact-level comparison. The calibrated MarketPop population of Table 1 is held fixed and SEIS is toggled without recalibration, under the same simulator, random-input protocol, and evaluation schedule. The SEIS On evaluation coincides with MarketPop (Ours) in Table 1.

Disabling SEIS changes d_{cal} from 1.47 to 1.95 and d_{HO} from 4.89 to 5.23.

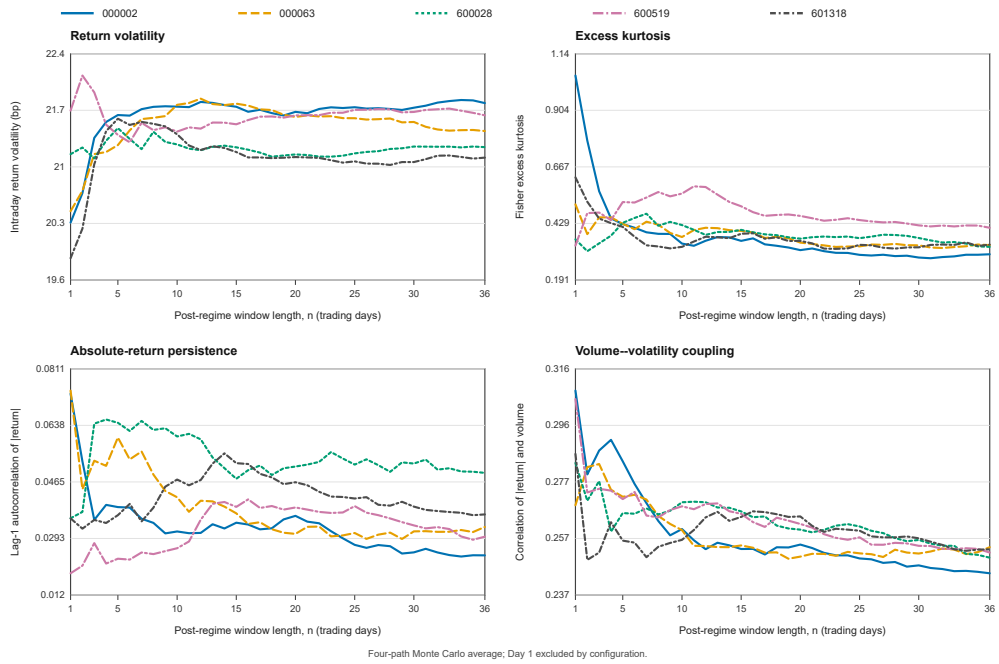


Figure J.2: Window means of the four nondegenerate market facts on the same frozen paths. Curves are reported by asset as the post-regime prefix expands; their residual movement separates slow state evolution from the sampling uncertainty summarized in Figure J.1.

Table K.1: Ablation of the Strategy-External Information System. The calibrated population is held fixed and SEIS is toggled without recalibration. Bold values in Q_1–Q_5 fall within one empirical standard deviation; bold discrepancy values indicate the better result.

Configuration	Q_1	Q_2	Q_3	Q_4	Q_5	$d_{\mathrm{cal}} \downarrow$	$d_{\mathrm{HO}} \downarrow$
SEIS Off	**12.83**	**0.526**	**0.167**	**0.227**	**0**	1.95	5.23
SEIS On	**13.35**	**0.551**	**0.158**	**0.260**	**0**	1.47	4.89

L Calibrated Population Anatomy

Section 4.3 summarizes the main deviation pattern of the calibrated population $\widehat{X}$ from the initial configuration X^0; this appendix reports the complete level-by-level comparison, with Figure L.1 providing the corresponding visual summary.

Calibration replaces the uniform initial endowment of $39{,}062{,}500$ cents per trader with a skewed allocation ranging from $14{,}054{,}363$ to $28{,}668{,}029$ cents, with the Gini coefficient rising from 0 to 0.073 and the wealthiest decile holding 12.6% of the endowment. The remaining levels move only marginally: asset-allocation weights stay centered at 0.2 with cross-trader dispersion of order 0.01; strategy-influence weights remain centered at $1/6$ with dispersion of order 0.006 and maximum deviation 0.05; and the behavioral parameters remain essentially at their template defaults, with median absolute normalized movement of 0.0024 and a maximum of 0.0055 over all 47 fields. One possible explanation for the concentration in π_1 is that a more unequal endowment changes the concentration of trading capacity and thereby the coupling between volume and volatility (Q_4).

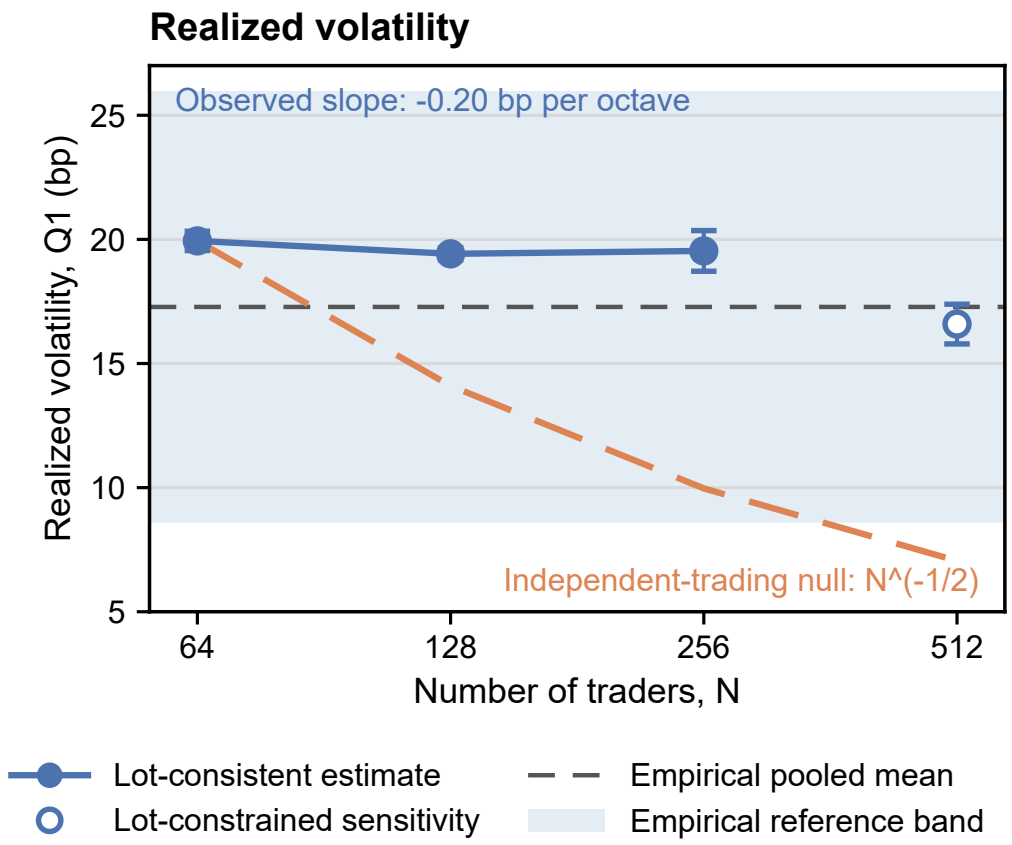


Figure J.3: Population size and realized volatility under the fixed pilot configuration. Filled markers are lot-consistent estimates; the hollow, round-lot-constrained $N = 512$ point is excluded from the slope. Bars are seed standard deviations. Shading and the gray dashed line mark the empirical band and pooled mean; the orange dashed curve is the $N^{-1/2}$ null.

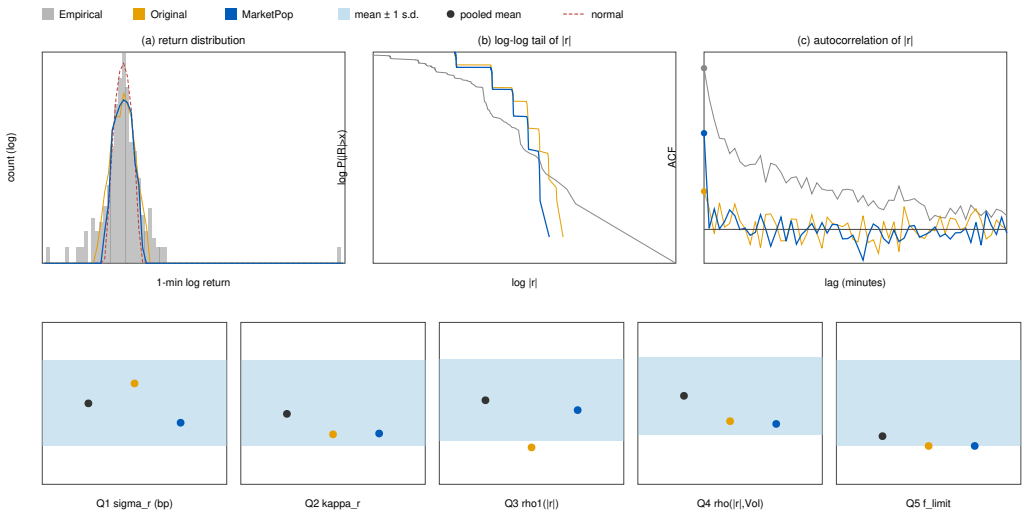


Figure J.4: Stylized-fact profile of the calibrated market. Top row: one-minute log returns pooled over assets and registered paths for the empirical data (gray), the uncalibrated baseline Original (orange), and the calibrated market MarketPop (blue) — (a) return distribution on a log-count scale with a normal overlay, (b) log–log tail of $|r|$, (c) autocorrelation of $|r|$; calibration moves each dynamical signature toward the empirical reference, with residual gaps visible rather than claimed away. Bottom row: the five facts Q_1–Q_5 against the pooled empirical mean (dot) $\pm$ one standard deviation (shaded) of Section 4.1; markers show Original and MarketPop as in Table 1. The panels display dynamical properties and reference coverage; they do not constitute a trajectory-level fitting claim.

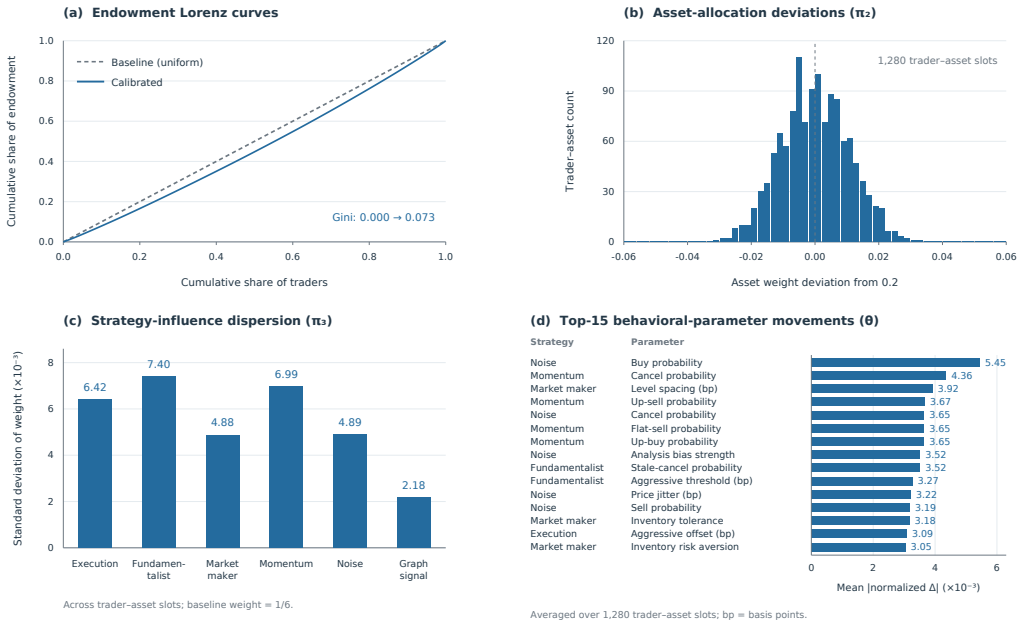


Figure L.1: Calibrated population anatomy: deviation of the calibrated population $\widehat{X}$ from the initial configuration X^0, level by level. (a) Lorenz curves of the initial endowment, baseline vs. calibrated; (b) histogram of calibrated asset-weight deviations from the uniform 0.2; (c) per-strategy dispersion of calibrated strategy-influence weights around $1/6$; (d) top-15 behavioral-parameter movements, mean absolute normalized change over all trader–asset slots.