FROM ANONYMOUS WALLETS TO BEHAVIORAL AGENTS: A CALIBRATED POLICY TEST BED FOR DAO GOVERNANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Unlike traditional corporate hierarchies, Decentralized Autonomous Organizations (DAOs) manage a multi-billion-dollar economy by utilizing token-voting mechanisms that aim to distribute governance power. While DAOs envision democratic governance, their complex economic dynamics can produce unforeseen issues, such as token concentration and runaway inflation. Navigating this complexity requires scalable simulation testbeds that construct heterogeneous behavioral agents from large volumes of on-chain transaction data. We propose DAO-SIM, the first realistic data-driven, LLM-based token-flow simulator that uses inverse behavioral profiling through event-chain clustering and two-stage LLM interpretation to reconstruct heterogeneous participant profiles under the anonymity of DAOs. Then a hierarchical agent architecture separates probabilistic decisions by profile-level super agents from independent execution by wallet-level child agents, enabling flexible simulation scaling-up while reducing redundant LLM inference. Built on over 3 million on-chain transactions collected from Uniswap, DAO-SIM simulates decision-making for an initial population of 5,000 agents and up to 10,000 at peak. Compared to existing agent-based baselines, our simulation aligns much more closely with observed empirical data and faithfully reproduces 3 key stylized facts of token economics: whale effect, path dependence, and the-rich-get-richer effect. Leveraging this validated fidelity, we deploy DAO-SIM as a robust testbed for counterfactual policy experiments. Our findings reveal: modest benefits from perfectly equal initial allocations, substantial inequality reduction from equal inflation issuance, and – contrary to prior studies – little redistributive effect from staking. Consequently, we recommend combining fairer initial allocations with equally issued inflation to improve DAO governance.

1 INTRODUCTION

Decentralized autonomous organizations (DAOs) represent a distinct organizational form adopted across the blockchain ecosystem, including decentralized exchanges (Allen et al., 2023), lending protocols (Falk et al., 2024), and blockchain infrastructure (Fritsch et al., 2024). Unlike conventional corporations, DAOs let token holders govern directly rather than through a board, and their votes and execution are publicly verifiable on blockchains (shown in Figure 1). By early 2025, DAOs had become economically important: over 10,000 active DAOs with their treasury assets reaching US$22.5 billion (Cong et al., 2025). However, DAO governance faces challenges, including runaway inflation (Jermann & Xiang, 2025) and unequal token holdings (Fernandez et al., 2024).

Among these challenges, token-holding inequality is particularly important because voting power is typically tied to token ownership, which has motivated research into ways to mitigate token concentration. Agent-based modeling (ABM) has shown potential for studying concentration mechanisms and evaluating policies (Fernandez et al., 2024; Cong et al., 2024). However, ABMs are often criticized for oversimplifying human behavior through fixed agent types and decision rules. Recent advances in LLM agents offer new opportunities to improve the realism of such simulators, with successful applications spanning economic decision-making (Li et al., 2024), social interactions (Piao et al., 2025), and financial markets (Yang et al., 2025; Piao et al., 2026). These advances motivate integrating LLM agents into DAO simulators for more realistic simulation. A distinctive opportunity

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.

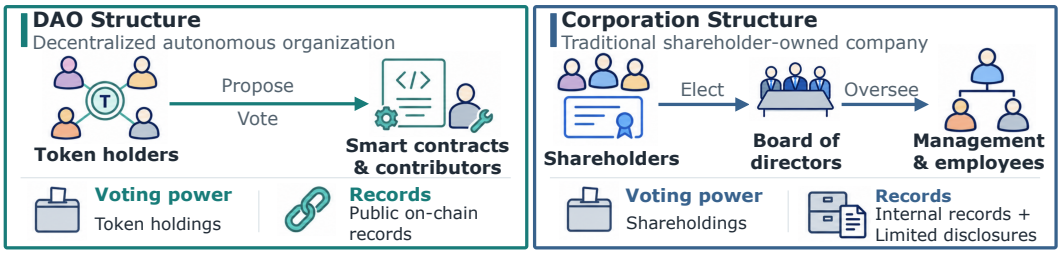


Figure 1: Governance structure comparison between DAOs and conventional corporations.

in DAOs is the public availability of on-chain wallet-level transfers and holdings, whereas comparable participant-level records in conventional financial markets are often private. Using these on-chain records to inform agent behavior is therefore central to simulation realism.

In this study, we propose DAO-SIM, a realistic data-driven simulator that, to the best of our knowledge, is the first to apply LLM agents to DAO token flow dynamics simulation. We address two critical methodological challenges in its domain. The first challenge is *profiling from raw trading records*: constructing behavioral profiles requires extracting and interpreting complex patterns from these raw trading records. Stock-market simulators often rely on individual or institutional attributes from external platforms, as individual-level trading records are generally private. We therefore take advantage of the public availability of on-chain trading records to perform *inverse behavioral profiling*. Specifically, we cluster wallets through event-chain features (Wang et al., 2016), consisting of transfer direction, counterparty type, relative size, and time gap. An LLM then performs two-step profile interpretation: deriving trading profiles from each cluster's characteristics, then incorporating DAO-related participant and governance participant attributes into final profiles. Grounded in real on-chain data, these profiles combine both behavioral heterogeneity and realism.

The second challenge is a *scale–cost dilemma* arising from clustering-based inverse behavioral profiling: the resulting profiles are far fewer than the participants needed to simulate token flows at scale. Specifically, assigning one agent per profile limits simulation scale, while assigning multiple independent LLM agents to each profile incurs redundant inference costs. We therefore introduce a *hierarchical agent architecture*, where *super agents* make profile-level probabilistic decisions, while a large population of *child agents* independently execute these decisions through Bernoulli sampling. An order book matches their execution orders at exogenous observed token prices. Executed trades update child wealth positions and profit and loss (PnL), feeding back into subsequent super agent decisions. Compared to prior simulators, this architecture allows us to flexibly scale up the simulation, improving realism while controlling costs (Comparison in Table 1).

To validate simulation realism, we apply DAO-SIM to Uniswap (UNI), a major DAO with over 380,000 holders and treasury assets of US$1.64 billion (Tally, 2026). We collect over three million records spanning September 17, 2020, to September 20, 2021, combining UNI transfers extracted using Reth with governance proposals and votes extracted from Snapshot. Through inverse behavioral profiling, we initialize 48 super agents that guide 5,000 child agents, with the child agent population reaching 10,000 at its peak. Then we test whether DAO-SIM simulations reproduce three documented stylized facts: *whale effect*, *path dependence*, and *the-rich-get-richer effect*. The results show that DAO-SIM reproduces the empirical UNI trends in all three stylized facts, and the metric values align more closely with empirical observations than those of ABM simulations, indicating greater fidelity in capturing token holding inequality and flow concentration.

Table 1: Comparison of financial simulators: profiling, agent architecture, and population scale.

Simulator	Target	Profile construction	Decision and execution	Population*
TwinMarket	Stock	Xueqiu attributes	LLMs make decisions and executions	100 (max 1,000)
MarketSim	Stock	Institution styles	LLMs give weights, traders execute policies	2,950 traders
DAO-SIM	**DAO**	**Transfer clustering + LLM interpretation**	**Super agents output probabilities; child agents execute by sampling**	**5,000 initially; 10,000 at peak**

Note: TwinMarket: Yang et al., 2025; MarketSim: Piao et al., 2026. * LLM-guided population excludes background agents.

Finally, we use DAO-SIM as a robust policy testbed to compare the effects of initial token allocation, inflation, and staking in a Uniswap-calibrated environment with previously reported effects (shown in Table 2). First, we extend prior work on fair launch (Barbereau et al., 2023) by testing perfectly equal initial allocations, which yield modest reductions in final inequality. Second, we refine prior inflation findings (Cong et al., 2024): equal issuance of inflation substantially reduces inequality, whereas proportional issuance offers little mitigation. Third, staking provides little redistribution, challenging the concentration-reducing effects reported by prior ABMs (Cong et al., 2024). Derived from more trustable simulations calibrated to empirical Uniswap data, these findings offer new perspectives on policy effectiveness. They also recommend promoting fairer initial allocations and distributing newly issued tokens through inflation equally across all holders to improve DAO governance. Overall, our study makes three key contributions:

Table 2: Policy effects on reducing concentration.

Policy	Empirical	ABM	DAO-SIM
Initial	Useless[a]	Useless[b]	**Modest**
Inflation	Effective[c]	Effective[d]	**Effective**
Staking	Centralizing[e]	Effective[d]	**Useless**

Sources: [a]Barbereau et al., 2023; [b]Fernandez et al., 2024; [c]Han et al., 2026; [d]Cong et al., 2024; [e]Pahari et al., 2026 on voting power.

- We introduce DAO-SIM, an LLM-based DAO token-flow simulator calibrated to empirical data, and validate its realism through closer alignment with empirical stylized facts than ABMs.
- We provide a robust testbed calibrated to empirical data for designing and evaluating DAO policies, offering new perspectives on the effectiveness of initial allocation, inflation, and staking.
- We combine *inverse behavioral profiling* with a *hierarchical agent architecture* to derive heterogeneous profiles from on-chain records and scale up simulations while controlling LLM costs.

2 Related Work

2.1 LLM for Social and Economic Simulation

Motivated by the reasoning capability of LLMs, LLM-based social simulations model human-like behavior through carefully designed agent-level reasoning architectures, spanning small communities (Park et al., 2023) and large urban societies (Piao et al., 2025). Built on these advances, economic simulations use adaptive agents to study AI labor market (Chiu et al., 2025) and macroeconomic system (Li et al., 2024), providing controlled testbeds for aggregate dynamics and policy interventions. Financial simulations then incorporate heterogeneous investors (Yang et al., 2025) and institutional behavior (Piao et al., 2026) to study emergent stock market dynamics. However, these financial simulations primarily focus on stock markets, with limited attention to cryptocurrency ecosystems and DAO governance including token holding concentration.

2.2 DAO Governance Challenges and Corresponding Policies

Empirical research consistently finds highly concentrated token-based voting power across major DAO protocols, despite their decentralized governance goals (Barbereau et al., 2023; Fritsch et al., 2024; Sun et al., 2024). Research on mitigating token concentration finds that airdrops can reduce concentration (Han et al., 2026), while initial fair-launch allocations may not prevent subsequent concentration (Fernandez et al., 2024). Inflation has been shown to reduce token concentration in simulations (Cong et al., 2024), while the effects of staking remain contested (Cong et al., 2024; Pahari et al., 2026). Evaluating unimplemented policies remains difficult with empirical data, while existing DAO simulators rely on oversimplified fixed rules. These limitations highlight the need to integrate LLM agents into DAO simulations for more reliable policy evaluation.

3 DAO Simulator Design

As illustrated in Figure 2, DAO-SIM consists of three components: i) behavioral profiling, which derives heterogeneous participant profiles from raw on-chain transfer and governance records; ii) market simulation, which specifies agent decision and execution rules and the market matching mechanism; and iii) policy testing, which evaluates how DAO governance policies affect token holding concentration. We describe each component in the following sections.

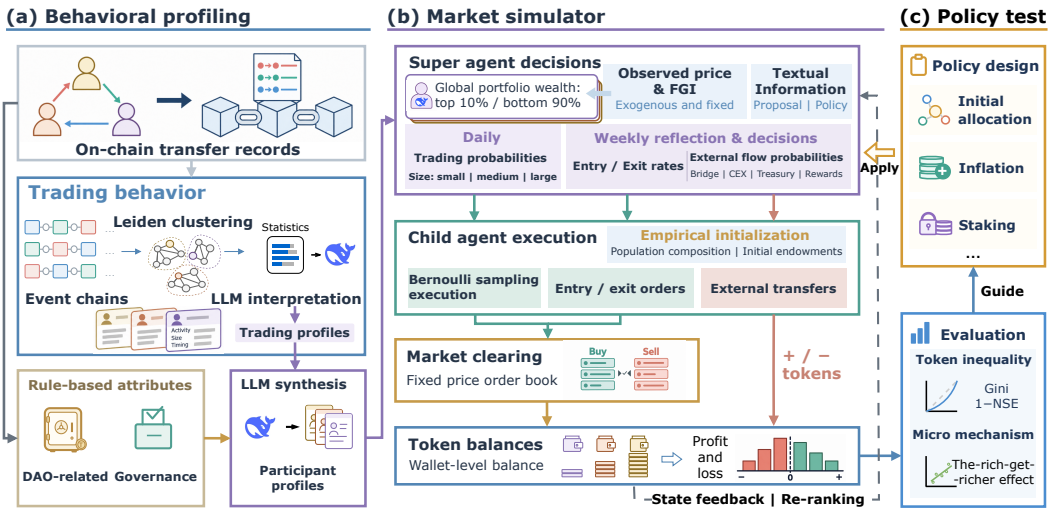


Figure 2: Overview of DAO-SIM. In Behavioral profiling, we infer heterogeneous profiles from on-chain transfer records. In Market simulator, we establish a DAO token market to simulate flow dynamics. In Policy test, we evaluate how counterfactual policies affect token inequality.

3.1 Inverse Behavioral Profiling

Existing simulators construct profiles from platform-specific user attributes and trading records covering a limited participant sample (Yang et al., 2025) or institutional investment styles (Piao et al., 2026). DAO on-chain transfer records, on the contrary, provide publicly observable traces of individual-level behavior. We therefore infer heterogeneous participant profiles inversely from these on-chain records. The procedure contains four steps: transfer-action encoding, event-chain feature construction, behavioral clustering, and two-stage LLM profile interpretation.

Transfer-action encoding. We first transform each raw governance-token transfer into a discrete action event a_t. Each action records three dimensions: transfer direction, counterparty type, and transfer size relative to the wallet's current token balance. Counterparties distinguish ordinary users from behaviorally meaningful entities such as CEXs, bridges, DAO-related sources, and reward distributors, while transfer amounts are discretized into small, medium, and large categories. The complete action vocabulary and discretization thresholds are reported in Appendix A.2.

Event-chain construction. A single transfer action provides limited behavioral information, so we represent each wallet through recurring local action patterns. To preserve temporal spacing, a discretized time-gap event g_t is inserted between every two consecutive transfer actions. For an odd chain length k, an event chain is therefore an alternating sequence of actions and time gaps,

$$c_{i,t}^{(k)} = (a_t, g_t, a_{t+1}, \ldots, g_{t+(k-3)/2}, a_{t+(k-1)/2}). \tag{1}$$

We use $k = 5$ as the default setting in simulations. Each distinct event-chain c is treated as a behavioral feature, and $x_{i,c}$ denotes its occurrence frequency in the transaction history of wallet i. Rare chains are removed, while chains involving CEXs, bridges, or reward distributors receive additional weight because they provide more specific and heterogeneous information about token-flow roles. We normalize the weighted event-chain counts within each wallet, producing a feature vector $\widetilde{\mathbf{x}}_i$ that captures the composition of recurring behavioral patterns rather than absolute transaction activity.

Leiden clustering. Following Wang et al. (2016), behavioral dissimilarity between wallets i and j is measured by normalized polar distance,

$$d(i,j) = \frac{1}{\pi} \arccos\left(\frac{\widetilde{\mathbf{x}}_i^\top \widetilde{\mathbf{x}}_j}{\|\widetilde{\mathbf{x}}_i\|_2 \|\widetilde{\mathbf{x}}_j\|_2} \right). \tag{2}$$

An approximate nearest-neighbor graph is then constructed from these distances and partitioned using the Leiden community-detection algorithm (Traag et al., 2019). Wallets are therefore grouped

according to recurring similarities in transfer direction, counterparties, amounts, and timing rather than externally assigned personas. The resulting clusters are described in Appendix A.3.

Two-stage LLM profile interpretation. We convert behavioral clusters into simulation-ready participant profiles through two-stage LLM interpretation, with cluster assignments and observable DAO attributes determined independently of the LLM. Details of LLM usage and prompts are provided in Appendix A.4 and A.5, with the final profile composition summarized in Appendix A.6.

Stage 1: Trading-profile interpretation. For each cluster, we select representative event chains using statistics describing how frequently they occur within the cluster and how strongly they distinguish it from other clusters. Given these chains and statistics, the LLM generates a compact trading profile describing characteristic transfer triggers, flow roles, timing, transaction sizes, and holding tendencies. When the LLM judges profiles from different clusters to be too similar, it refines them using event chains that distinguish those clusters, reducing semantic overlap.

Stage 2: Participant-profile synthesis. DAO-related participants and governance participants are sparse and may contribute little to clustering. We therefore identify these attributes using deterministic rules: an address is labeled a *DAO-related participant* if it has at least one observed incoming governance-token transfer from designated treasury, vesting, protocol-contract, or governance-contract addresses during the observation period, and a *governance participant* if it has an observed voting, proposal-creation, or delegation event. For wallet i, the LLM synthesizes the structured evidence $E_i = (T_{z_i}, A_i, G_i)$ into a final participant profile, where T_{z_i} is its cluster's trading profile and A_i and G_i indicate DAO-related-participation and governance-participation status, respectively.

3.2 Hierarchical Agent Architecture for Market Simulation

Hierarchical agent architecture. Mapping inferred profiles to trading agents creates a trade-off: one agent per profile limits market participation and risks liquidity depletion, while multiple agents per profile waste LLM tokens on nearly identical prompts. We therefore introduce a hierarchical agent architecture, separating profile-level reasoning from wallet-level execution. For each participant profile, two *super agents* represent the current top 10% and bottom 90% wealth segments and generate group-level decisions, while *child agents* represent individual executable wallets that realize these decisions independently. Child agents are dynamically reassigned to the corresponding super agent within their profile based on their wealth rank each day (as shown in Figure 3). This design preserves profile-level behavioral heterogeneity while allowing the simulated population to scale up flexibly.

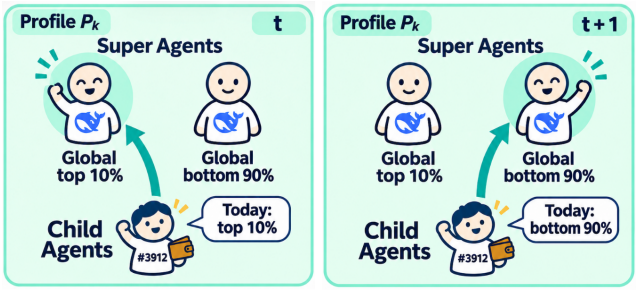


Figure 3: Demonstration on hierarchical agent architecture.

Population initialization. The population is initialized from empirical on-chain distributions. Within each wealth segment, wallets are allocated across participant profiles according to their empirical shares, preserving the observed association between behavioral type and wealth position. Initial token balances are sampled from empirically calibrated distributions: lognormal for the bottom 90% and Pareto for the top 10%. Unlike the prescribed fiat-endowment distributions in Fernandez et al. (2024), USD trading capacity is sampled from empirically estimated distributions conditional on token holdings and wealth segment. New entrants draw their token balances and USD trading capacity from the same distributions used at initialization. Calibration details are in Appendix A.7.

Decision process. Super agents receive their profiles and wealth segments, protocol policies, token prices and the Fear and Greed Index (FGI), group size, PnL and trading capacity, and past decisions and executions. Daily decisions also use recent market history to set probabilities for small, medium, and large purchases and sales. Weekly population reflection adds weekly market trends and sets entry and exit rates. A separate weekly decision sets external-transfer probabilities. Entry and exit change the population size, which remains fixed in our ABM baseline. New entrants draw endowments from the same distributions used at initialization. We primarily use the DeepSeek-V4-Pro API for super-agent decisions, with further details provided in Appendix A.8.

Execution loop. Child agents sample feasible actions through Bernoulli trials independently. Daily orders are matched at observed token prices, while external transfers bypass the order book. Entry and exit orders first match against each other. The remaining net orders enter the order book at a fixed daily fraction over the next seven steps, with priority for matching against daily orders. Executed trades and external transfers update wallet balances. We then compute token inequality and group PnL statistics, with the latter informing subsequent super-agent decisions.

3.3 Policy Interface and Evaluation

DAO-SIM provides a testbed to support counterfactual policy interventions (*e.g.*, initial allocation, token inflation and staking). Policies can modify the corresponding market or token-distribution mechanisms and are simultaneously described to super agents, allowing participant behavior to respond to the intervention rather than remaining fixed. We evaluate policy effects on both token inequality, measured by the Gini coefficient and $1-\text{NSE}$, and its underlying micro-level mechanisms. Grounded in high-fidelity simulations, these results inform subsequent policy redesign.

4 Validate DAO Simulation

In this section, we deploy DAO-SIM on Uniswap, validate it by three documented stylized facts: *whale effect, path dependence, the-rich-get-richer effect*, and compare its results with empirical data and an ABM baseline. Uniswap is Ethereum's leading decentralized exchange (DEX), with 60.1% of 30-day DEX trading volume (DefiLlama, 2026), over 380,000 token-holding addresses, and a US$1.64 billion treasury (Tally, 2026). Following the steps in Section 3, we employ 48 super agents and 5000 initial child agents for simulations on Uniswap. *Empirical* denotes observed Uniswap on-chain statistics, *DAO-SIM* denotes our simulation results, and *ABM* denotes the Fernandez et al. (2024)-style baseline with parameters selected by grid search. We conduct three runs using random seeds 101, 202, and 303, and the results shown below are averaged across all seeds. Details of the empirical data sources and collection procedures are provided in Appendix A.1.

4.1 Stylized Fact 1: Whale Effect

The first stylized fact is the *whale effect*: governance tokens are persistently concentrated among a small fraction of addresses (Sun et al., 2024; Fritsch et al., 2024; Born et al., 2026). Since token holdings determine voting power, realistic DAO simulators should reproduce their concentration.

Table 3: Errors in simulated token concentration relative to empirical data. Lower values indicate better agreement across all evaluated metrics.

	DAO-SIM		ABM	
Metric	**RMSE**	**MAPE(%)**	**RMSE**	**MAPE(%)**
Gini	0.0189	1.84	0.1230	12.30
$1-\text{NSE}$	0.0388	4.49	0.4183	62.38

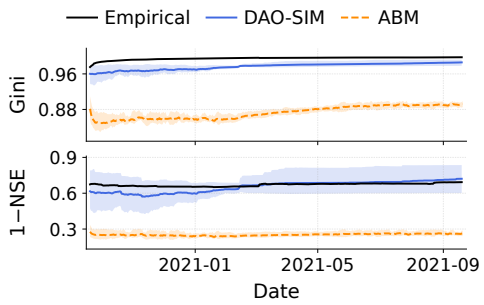


Figure 4: Temporal trajectories of token concentration measured by Gini and $1-\text{NSE}$.

Table 4: Token-flow share comparison of the top 1% and 5% of addresses, ranked separately by their total token received and sent volumes.

Flow	**Top** q	**Empirical**	**DAO-SIM**	**ABM**
Receive	Top 1%	93.5%	76.1%	54.8%
	Top 5%	97.0%	90.9%	79.4%
Send	Top 1%	94.0%	70.3%	57.0%
	Top 5%	97.3%	86.2%	81.0%

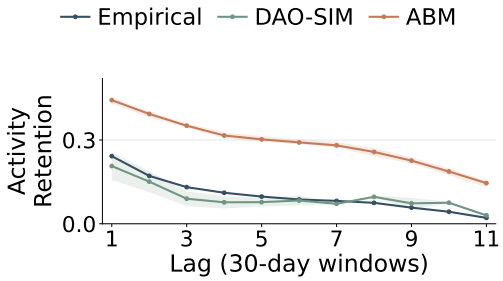


Figure 5: Activity retention of top-1% active addresses, measured by Jaccard overlap at lags, with each lag unit representing 30 days.

We evaluate the whale effect using the Gini coefficient and $1-$NSE, where higher values indicate greater concentration. As shown in Figure 4, the empirical distribution remains highly concentrated throughout the observation period. DAO-SIM closely reproduces this regime and its gradual upward trend, whereas the ABM remains substantially less concentrated, especially for $1-$NSE. Quantitatively (shown in Table 3), DAO-SIM achieves lower RMSEs and MAPEs for both Gini coefficient and $1-$NSE compared to ABM, indicating substantially closer empirical alignment.

4.2 Stylized Fact 2: Path Dependence

The second stylized fact is *path dependence*: a small set of highly active addresses repeatedly dominates token flows. These active addresses can reinforce token concentration by consistently receiving more tokens than they send (Victor & Lüders, 2019; De Collibus et al., 2025).

We examine *flow concentration*, the shares of tokens received and sent by addresses ranked by each volume, and *activity retention*, the extent to which the same addresses remain active. Retention is measured by the Jaccard overlap $J(A, B) = |A \cap B|/|A \cup B|$, where A and B are the top-1% active-address sets in two observation windows separated by certain units, with each unit representing 30 days. Table 4 shows that DAO-SIM more closely matches the empirical top-1% shares of tokens received/sent ($93.5\%/94.0\%$), producing $76.1\%/70.3\%$ compared with the ABM's $54.8\%/57.0\%$. The same ordering holds for the top 5%. For activity retention (Figure 5), DAO-SIM matches the empirical one-window overlap (0.21 versus 0.24), with both declining to zero at longer lags. The ABM starts higher (0.44) and declines more slowly. Thus, DAO-SIM achieves higher fidelity.

4.3 Stylized Fact 3: The-rich-get-richer Effect

The third stylized fact is *the-rich-get-richer effect*: wealthier holders tend to accumulate more tokens over time (Di Francesco Maesa et al., 2018; Kondor et al., 2021). This pattern provides a micro-level explanation for token concentration and is essential for a high-fidelity DAO simulator to capture.

We use a *two-part accumulation test* to examine whether window-start holdings predict the probability and amount of positive net accumulation, controlling for common differences across windows. For each wallet i and rolling window $[t, t+\Delta]$, we estimate:

$$p_{i,t} = \Pr(\Delta B_{i,t} > 0 \mid x_{i,t}, t) = \text{logit}^{-1}(\alpha_1 + \beta_1 x_{i,t} + \lambda_{1,t}), \tag{3}$$

$$\log(1 + \Delta B_{i,t}) = \alpha_2 + \beta_2 x_{i,t} + \lambda_{2,t} + \varepsilon_{i,t}, \quad \Delta B_{i,t} > 0, \tag{4}$$

where $B_{i,t}$ denotes wallet i's token balance at time t, $x_{i,t} = \log(1 + B_{i,t})$ is its log-transformed initial balance, and $\Delta B_{i,t} = B_{i,t+\Delta} - B_{i,t}$ is its net token accumulation over the window. The window fixed effects $\lambda_{1,t}$ and $\lambda_{2,t}$ allow the baseline accumulation probability and conditional magnitude to vary across windows. We summarize the two parts using the *average marginal effect* (AME) and *conditional accumulation magnitude* (CAM): $\text{AME} = \frac{1}{N}\sum_{i,t} \frac{\partial \hat{p}_{i,t}}{\partial x_{i,t}}$ and $\text{CAM} = \beta_2$, where N counts wallet-window observations. Positive AME and CAM indicate that wealthier wallets are more likely to accumulate tokens and accumulate larger amounts when they do, respectively. Figure 6 shows that DAO-SIM reproduces the positive empirical AME (0.011 versus 0.003), whereas the ABM reverses its sign (-0.027), potentially due to its fixed share of net-buying agents. Both models approximate the empirical CAM (0.429), with DAO-SIM's estimate (0.497) closer than the ABM's (0.543). These results support DAO-SIM's higher fidelity in capturing micro-level mechanisms while reproducing macro-level concentration, demonstrating consistency across both scales.

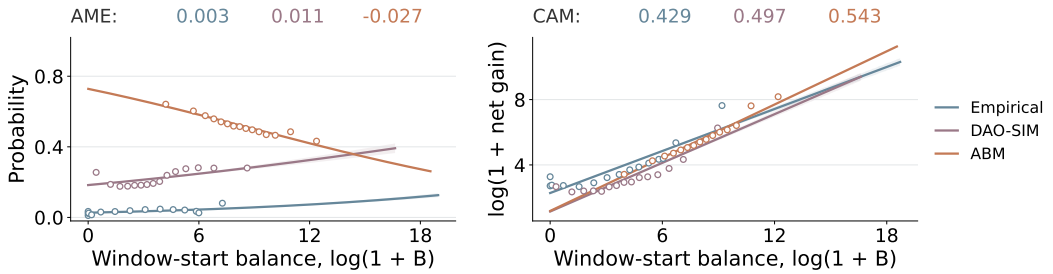


Figure 6: Two-part accumulation test: probability of positive accumulation and its magnitude.

5 POLICY TESTBED

We use our simulator to evaluate three policy approaches commonly studied in the DAO literature: i) *initial token allocation*, ii) *inflation*, and iii) *staking*. We examine their implications for token-holding inequality and explore the mechanisms underlying the observed outcomes.

Table 5: Token-holding inequality at day 365 under initial allocation, inflation, and staking policies.

Metric	Default	Initial allocation				Inflation				Staking		
		0.75	0.50	0.25	0.00	E1	E7	E30	P7	30d/1%	60d/3%	90d/5%
Gini	0.986	0.975	0.942	0.855	0.803	0.698	0.604	0.706	0.981	0.965	0.987	0.979
$1 - \text{NSE}$	0.721	0.640	0.472	0.349	0.319	0.176	0.128	0.206	0.589	0.517	0.699	0.618

5.1 INITIAL TOKEN ALLOCATION

We vary initial allocation through a balancing parameter $\gamma \in \{0.75, 0.50, 0.25, 0.00\}$, using $b_i(\gamma) = S b_i^\gamma / \sum_j b_j^\gamma$, where b_i is wallet i's baseline balance and S is the total supply. Intuitively, a smaller γ produces a more equal allocation, and $\gamma = 0$ assigns perfectly equal balances.

Table 5 shows that lower γ reduces both final Gini and $1 - \text{NSE}$, from 0.986 to 0.803 and from 0.721 to 0.319, respectively, although concentration remains substantial. Two patterns may explain this improvement (Figure 7). First, flow concentration generally declines as γ decreases, suggesting weaker dominance by leading addresses and potentially weaker path dependence. Second, AME rises while CAM declines overall, particularly at low γ: wealthier wallets have a stronger advantage in whether they accumulate tokens, but a weaker advantage in how much they accumulate. The reduced advantage in accumulation amounts may partly offset the increased probability advantage, weakening the overall rich-get-richer effect. These results suggest that fairer initial allocations modestly reduce final concentration, indicating some policy benefits despite the reconcentration trend also observed in prior studies (Barbereau et al., 2023; Fernandez et al., 2024; Cong et al., 2024).

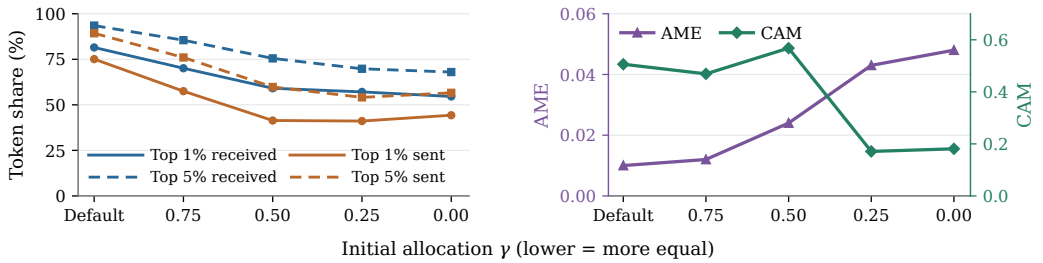


Figure 7: Initial allocation. Left: flow concentration. Right: AME (left axis) and CAM (right axis).

5.2 INFLATION POLICY

We implement inflation by periodically issuing new tokens, allocated either equally across all wallets (E) or proportionally to their current holdings (P). The number following each label denotes the issuance interval in days. All treatments use an annual inflation rate of 10%.

Equal issuance of inflation substantially reduces final inequality (with similar results for E1, E7, and E30), whereas P7 offers little improvement as shown in Table 5. Under equal issuance, AME rises sharply while CAM declines (shown in the left panel of Figure 8), which probably spreads accumulation opportunities across more wallets: positive rates achieve 51–58%, compared with 23% under Default. E7 also produces larger typical gains but smaller extreme gains than Default and P7 (shown in the middle panel of Figure 8), suggesting that accumulation becomes more broadly distributed and less dominated by large gains among wealthy holders. Consequently, initially wealthy holders lose ownership share substantially (the initial top 1% illustrates this in the right panel of Figure 8), leading to lower inequality. These findings complement empirical evidence on airdrops (Han et al., 2026) and ABM findings on inflation (Cong et al., 2024), supporting equal issuance as a promising policy for reducing inequality while keeping the overall inflation rate controlled.

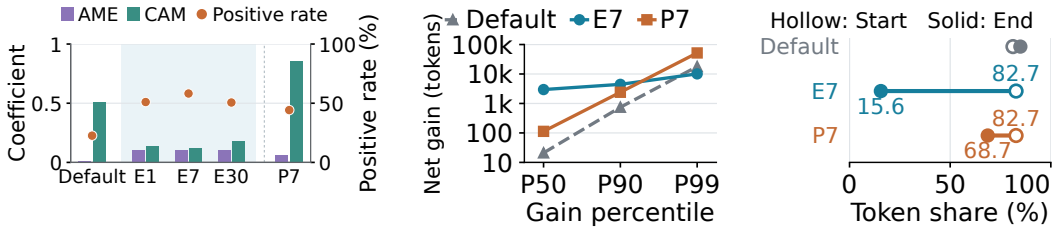


Figure 8: Inflation effects. Left: AME, CAM, and positive rate, representing the proportion of observations with net token gains; shading groups equal-issuance policies. Middle: positive 30-day net-gain quantiles. Right: initial and final ownership shares of the initially richest 1%.

5.3 Staking Policy

Staking resembles a fixed-term deposit. We compare 30-, 60-, and 90-day staking policies with maturity rewards of 1%, 3%, and 5%, respectively. Super agents determine execution probabilities based on profiles and current context. Staked tokens cannot be transferred until maturity.

As shown in Table 5, staking policies offer little mitigation of inequality for all 30d/1%, 60d/3%, and 90d/5% settings, with only a slight reduction under 30d/1%. The left panel of Figure 9 may explain this limitation, as staking rewards concentrate among already wealthy holders. Empirical research identifies voting-power concentration through staking intermediaries (Pahari et al., 2026), while our results highlight a distinct limitation arising from rewards tied to existing holdings. The adapted ABM likewise fails to reduce inequality in our Uniswap setting, unlike previously reported findings (Cong et al., 2024), suggesting that policy benefits depend on the simulated environment and behavioral assumptions. Its nearly unchanged AME and similarly increased CAM across policies may be attributable to its fixed behavioral rules (Figure 9, right). By contrast, DAO-SIM responds differently to staking terms: 30d/1% raises AME and lowers CAM, while longer terms produce smaller changes. These responses nevertheless deliver little overall inequality reduction. Together with our inflation experiments, these findings suggest that encouraging token retention is insufficient: broader reward distribution may be needed to weaken existing wealth advantages.

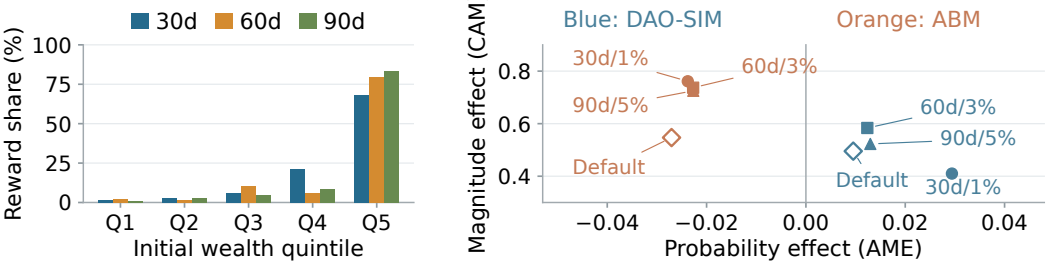


Figure 9: Staking effects. Left: reward shares across five initial wealth quintiles; Q5 denotes the initially wealthiest 20% of addresses. Right: AME and CAM for both DAO-SIM and ABM.

6 Conclusion

In this study, we introduce DAO-SIM, a realistic data-driven LLM simulator for studying governance challenges in DAOs. Inverse behavioral profiling reconstructs heterogeneous participant profiles from transparent on-chain records, while hierarchical agent architecture enables flexible simulation scaling-up and controlled LLM token costs. Validation on Uniswap shows greater fidelity to macro-level empirical concentration and micro-level mechanisms than a traditional rule-based ABM. Policy experiments show that more equal initial allocations alleviate but do not eliminate concentration, equal issuance substantially reduces inequality, and staking offers little redistribution as rewards accrue predominantly to wealthy holders. These findings support combining fairer initial allocations with continued equal token issuance and highlight the importance of reward allocation and behavioral responses in policy assessment. Together, our results establish DAO-SIM as a counterfactual testbed for examining the mechanisms and limits of DAO token-distribution policies.

AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity and academic style of the manuscript, assist with preliminary searches for literature on behavioral clustering and agent-based simulation, and support Python code writing and development. We reviewed the suggested revisions and verified the cited sources against the original publications. The use of LLMs for two-stage behavioral profile interpretation and super-agent decision-making is described in the methodology and appendix. The authors take full responsibility for the final manuscript, including its claims, references, code, and reported results.

REPRODUCIBILITY STATEMENT

The simulator design is described in Section 3, while Appendix A provides the transfer-action encoding, initialization calibration, and LLM decision prompt templates. Section 4 specifies the random seeds used for the main validation experiments, whose results are averaged over three runs. Both LLM responses and wallet-level actions introduce stochasticity, so individual simulation trajectories may differ across repeated runs. We report aggregate outcomes across runs to facilitate comparison with empirical patterns.

REFERENCES

Darcy W. E. Allen, Chris Berg, Aaron M Lane, Trent MacDonald, and Jason Potts. The exchange theory of web3 governance. *Kyklos*, 76(4):659–675, 2023. doi: 10.1111/kykl.12345.

Tom Barbereau, Reilly Smethurst, Orestis Papageorgiou, Johannes Sedlmeir, and Gilbert Fridgen. Decentralised finance's timocratic governance: The distribution and exercise of tokenised voting rights. *Technology in Society*, 73:102251, 2023. doi: 10.1016/j.techsoc.2023.102251.

Alexandra Born, Zakaria Gati, Claudia Lambert, Mahvish Naeem, and Antonella Pellicani. Who to regulate? identifying actors within DeFi's governance. *SSRN Electronic Journal*, 2026. doi: 10.2139/ssrn.6478108.

Christopher Chiu, Simpson Zhang, and Mihaela van der Schaar. When ai agents compete for jobs: Strategic capabilities and economic dynamics of ai labour markets. *arXiv preprint arXiv:2512.04988*, 2025. URL https://arxiv.org/pdf/2512.04988.

Lin William Cong, Yilei Dong, Yunbo Lu, Qingsong Ruan, and Guojun Wang. Agent-based modeling for daos and defi. *Available at SSRN 5171559*, 2024. URL https://ssrn.com/abstract=5171559.

Lin William Cong, Daniel Rabetti, Charles C. Y. Wang, and Yu Yan. Centralized governance in decentralized organizations. *SSRN Electronic Journal*, 2025. doi: 10.2139/ssrn.5168660.

Francesco Maria De Collibus, Carlo Campajola, and Claudio J Tessone. The microvelocity of money in ethereum. *EPJ Data Science*, 14(1):11, 2025. doi: 10.1140/epjds/s13688-024-00518-6.

DefiLlama. Ethereum DEX Volume Rankings, 2026. URL https://defillama.com/dexs/chain/ethereum. Rolling 30-day trading volume. Accessed September 11, 2026.

Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. Data-driven analysis of bitcoin properties: exploiting the users graph. *International Journal of Data Science and Analytics*, 6(1): 63–80, 2018. doi: 10.1007/s41060-017-0074-x.

Brett Falk, Tasneem Pathan, Andrew Rigas, and Gerry Tsoukalas. Blockchain governance: an empirical analysis of user engagement on daos. *arXiv preprint arXiv:2407.10945*, 2024. URL https://arxiv.org/pdf/2407.10945.

Joaquin Delgado Fernandez, Tom Barbereau, and Orestis Papageorgiou. Agent-based model of initial token allocations: Simulating distributions post fair launch. *ACM Transactions on Management Information Systems*, 15(1):1–22, 2024. doi: 10.1145/3649318.

Robin Fritsch, Marino Müller, and Roger Wattenhofer. Analyzing voting power in decentralized governance: Who controls daos? *Blockchain: Research and Applications*, 5(3):100208, 2024. doi: 10.1016/j.bcra.2024.100208.

Jungsuk Han, Jongsub Lee, and Tao Li. Decentralized autonomous organization governance. *Management Science*, 2026. doi: 10.1287/mnsc.2024.07033.

Urban Jermann and Haotian Xiang. Tokenomics: Optimal monetary and fee policies. *Journal of Monetary Economics*, 155:103808, 2025. doi: 10.1016/j.jmoneco.2025.103808.

Dániel Kondor, Nikola Bulatovic, József Stéger, István Csabai, and Gábor Vattay. The rich still get richer: Empirical comparison of preferential attachment via linking statistics in bitcoin and ethereum. *Frontiers in Blockchain*, 4:668510, 2021. doi: 10.3389/fbloc.2021.668510.

Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. EconAgent: large language model-empowered agents for simulating macroeconomic activities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15523–15536, 2024. doi: 10.18653/v1/2024.acl-long.829.

Vabuk Pahari, Balakrishnan Chandrasekaran, Johnnatan Messias, Krishna P Gummadi, and Abhisek Dash. On the centralization of governance power in decentralized autonomous organizations. In *2026 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–10. IEEE, 2026. doi: 10.1109/ICBC67748.2026.11575466.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023. doi: 10.1145/3586183.3606763.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, et al. AgentSociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. 2025. URL https://arxiv.org/pdf/2502.08691.

Jinghua Piao, Zhentao Liu, Cheng Huang, Jiarui Huang, Songwei Li, Ranran Wang, and Yong Li. MarketSim: Simulating stock markets with large-scale generative agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/pdf?id=EzpJxPDqXB.

Xiaotong Sun, Charalampos Stasinakis, and Georgios Sermpinis. Decentralization illusion in decentralized finance: Evidence from tokenized voting in makerdao polls. *Journal of Financial Stability*, 73:101286, 2024. ISSN 1572-3089. doi: 10.1016/j.jfs.2024.101286.

Tally. Uniswap Governance, 2026. URL https://www.tally.xyz/gov/uniswap. Accessed September 11, 2026.

Vincent A Traag, Ludo Waltman, and Nees Jan van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019. doi: 10.1038/s41598-019-41695-z.

Friedhelm Victor and Bianca Katharina Lüders. Measuring ethereum-based erc20 token networks. In *International conference on financial cryptography and data security*, pp. 113–129. Springer, 2019. doi: 10.1007/978-3-030-32101-7_8.

Gang Wang, Xinyi Zhang, Shiliang Tang, Haitao Zheng, and Ben Y Zhao. Unsupervised clickstream clustering for user behavior analysis. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pp. 225–236, 2016. doi: 10.1145/2858036.2858107.

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 *Advances in Neural Information Processing Systems*, volume 38, pp. 63469–63519, 2025. doi: 10.52202/085713-2132.

A APPENDIX

APPENDIX CONTENTS

A.1 Raw Data Source 12
A.2 Event Chain Construction 12
A.3 Trading Clustering Results 12
A.4 Trading Profile Prompts and Inference Settings 13
A.4.1 Trading Profile Interpretation 13
A.4.2 Profile Similarity Audit 15
A.4.3 Evidence-Based Profile Refinement 15
A.5 Final Profile Construction Prompts 16
A.6 Final Profile Composition 17
A.7 Initialization Calibration 18
A.7.1 Initial Token-Balance Distributions 18
A.7.2 Conditional USD-Capacity Model 18
A.8 Simulation Prompts and Inference Settings 19
A.8.1 Daily Transfer Decisions 19
A.8.2 Weekly Population Reflection 20
A.8.3 Weekly External-Transfer Decisions 22

A.1 RAW DATA SOURCE

Governance proposals and voting records are obtained from Snapshot[1]. On-chain UNI token-transfer records are collected through a locally operated Ethereum node running Reth[2]. The observed prices we use in the market are UNI/USDT spot prices obtained from Binance[3], while Fear and Greed Index (FGI) data are obtained from CoinMarketCap[4].

A.2 EVENT CHAIN CONSTRUCTION

We represent each wallet's transfer history as a sequence of transfer actions and time-gap events. Each transfer action combines its direction, counterparty type, and relative size. Table 6 defines the action categories, with received and sent flow measured relative to the focal wallet. Table 7 discretizes the transferred amount relative to the wallet's token holdings into three size categories. Table 8 defines the time-gap categories used to distinguish transfers occurring in rapid succession from those separated by longer periods.

A.3 TRADING CLUSTERING RESULTS

Using event chains of length $k = 5$, we obtain 15 trading clusters from 156,957 addresses. Table 9 summarizes their sizes and characteristic transfer patterns.

[1] https://snapshot.box/
[2] https://github.com/paradigmxyz/reth
[3] https://www.binance.com/en/trade/UNI_USDT
[4] https://coinmarketcap.com/charts/fear-and-greed-index/

Table 6: Definition of blockchain transfer action categories

Action Category	Direction	Counterparty or Interpretation
IN_FROM_DAO	Receive	DAO treasury, vesting, protocol, or governance contracts.
IN_FROM_REWARD	Receive	Reward-distribution contracts.
IN_FROM_CEX	Receive	Centralized exchanges; typically exchange withdrawals.
OUT_TO_CEX	Send	Centralized exchanges; deposits or potential selling.
IN_FROM_BRIDGE	Receive	Cross-chain bridge addresses.
OUT_TO_BRIDGE	Send	Cross-chain bridge addresses.
IN_FROM_USER	Receive	Other user-controlled addresses.
OUT_TO_USER	Send	Other user-controlled addresses.

Table 7: Transfer-size categories based on the transferred share of holdings

Size Category	Share of Holdings
SMALL	$r < 10\%$
MEDIUM	$10\% \leq r < 30\%$
LARGE	$r \geq 30\%$

A.4 Trading Profile Prompts and Inference Settings

Trading profiles are constructed from event-chain statistics, rather than the textual summaries in Table 9. The procedure comprises three tasks: trading profile interpretation, profile similarity audit, and evidence-based profile refinement. Together, these tasks translate statistical patterns into standalone behavioral descriptions for subsequent final profiling. The boxes below present condensed prompts, with instance-specific JSON inputs replaced by placeholders.

API Configuration. We use DeepSeek-V4-Pro via API for trading-profile construction and subsequent final-profile synthesis. Requests are processed sequentially with a temperature of 0.2. Maximum output budgets are 6,000 tokens for trading profile interpretation, 8,000 tokens each for profile similarity audit and evidence-based profile refinement, and 2,500 tokens for final-profile synthesis. The application allows up to three retries for request failures, with linear backoff. Prompts require JSON responses, which are parsed and normalized into task-specific schemas. Input evidence, instantiated prompts, raw responses, and generated profiles are saved for inspection.

A.4.1 Trading Profile Interpretation

For each cluster, the LLM interprets the top 20 ranked event chains, their mean feature values, coverage, χ^2 scores, and behavioral summaries to construct an initial trading profile. Let N and N_c denote the total and within-cluster address counts, and let $n(s)$ and $n_c(s)$ count addresses containing chain s at least once in these respective sets. Within- and outside-cluster coverage are defined as

$$\mathrm{Cov}_{\mathrm{in}}(s,c) = \frac{n_c(s)}{N_c}, \qquad \mathrm{Cov}_{\mathrm{out}}(s,c) = \frac{n(s) - n_c(s)}{N - N_c}. \tag{5}$$

Mean feature values are averaged over the corresponding address sets. Positive evidence requires both higher mean feature values and higher coverage inside the cluster; within-cluster coverage of at least 5% additionally qualifies it as high-coverage evidence. Evidence with either metric lower inside the cluster is classified as weaker or negative. The prompt prioritizes high-coverage positive evidence, uses low-coverage patterns for secondary details, and treats weaker or negative evidence as constraints on interpretation. The output describes trigger, flow role, timing, relative transfer size, and holding meaning, together with simulation guidance and supporting evidence, without inferring identity or governance attributes.

System prompt for trading profile interpretation

```
You are an on-chain transfer-behavior analyst for DAO token simulation. Convert the supplied
     cluster evidence into a compact, standalone trading profile that can be understood
    without access to other profiles.
```

Table 8: Time-gap events between consecutive transfer actions

Time-Gap Event	Interval
GAP_LT_1D	$\Delta t < 1$ day
GAP_1D_7D	$1 \leq \Delta t < 7$ days
GAP_7D_30D	$7 \leq \Delta t \leq 30$ days
GAP_GT_30D	$\Delta t > 30$ days

Table 9: Trading clusters obtained with $k = 5$. Shares are calculated over all 156,957 addresses. "Rapid" denotes an inter-transfer gap below one day.

Cluster	Addresses	Share	Characteristic transfer patterns
C1	61,245	39.02%	Mixed transfers between users, including rapid outgoing sequences and delayed follow-up transfers.
C2	57,623	36.71%	Rapid forwarding of tokens received from users, followed by further receipts from users after 1–30 days.
C3	12,687	8.08%	DAO-related incoming transfers interleaved with transfers between users, including rapid subsequent outgoing transfers.
C4	5,290	3.37%	Repeated incoming transfers from users, followed by outgoing transfers after more than 30 days.
C5	3,388	2.16%	Incoming transfers from users followed by successive rapid outgoing transfers.
C6	2,740	1.75%	Rapid outgoing transfers to users, followed by further outgoing transfers after 7–30 days.
C7	2,739	1.75%	Rapid forwarding of tokens received from users, followed by another outgoing transfer after more than 30 days.
C8	2,330	1.48%	Rapid outgoing transfers to users, followed by further outgoing transfers after 1–7 days.
C9	1,925	1.23%	Repeated incoming transfers from users, followed by outgoing transfers after 1–7 days.
C10	1,758	1.12%	Rapid sequences of outgoing transfers to users, followed by further outgoing transfers after more than 30 days.
C11	1,699	1.08%	Successive incoming transfers from users followed by outgoing transfers, with gaps between adjacent transfers of less than one day.
C12	511	0.33%	Rapid forwarding of reward tokens, followed by an incoming transfer from a user after more than 30 days.
C13	184	0.12%	Receipt of reward tokens, followed after more than 30 days by a small-share outgoing transfer and then a rapid large-share outgoing transfer.
C14	95	0.06%	Receipt of reward tokens followed by successive outgoing transfers to users, with intervening gaps exceeding 30 days.
C15	2,743	1.75%	Low-coverage distinguishing patterns involving incoming transfers from CEX or reward addresses and delayed outgoing transfers to users.

```
Describe five behavioral dimensions: trigger, flow role, timing, amount pattern, and holding
      meaning.

Evidence Rules:
- Use positive evidence to characterize observed behavior.
- Strong defining characteristics must be supported by high-coverage positive evidence.
- Low-coverage patterns may provide secondary nuance.
- Use weaker or negative evidence only to avoid overclaiming.
- Do not infer DAO identity, governance participation, investment motivation, profit-taking,
      arbitrage, or official operational status.
- Do not make unsupported claims of uniqueness or absence.
- Do not compare the profile with other clusters.
```

User prompt for trading profile interpretation

```
Analyze the following cluster evidence:
{TRADING_CLUSTER_EVIDENCE_JSON}

Required Output:
Return only a JSON object containing:
- cluster_id
```

```
- TradingTypeName
- CoreBehaviorSummary
- BehaviorPattern: trigger, flow_role, timing, amount_pattern, holding_meaning
- SimulationImplication
- EvidenceBasis

Provide a two-to-four-sentence summary, one sentence describing the simulation implication,
     and two to four supporting event chains with coverage or chi-squared values when
     available.
```

A.4.2 PROFILE SIMILARITY AUDIT

The audit jointly reviews all initial profiles to identify overlapping behavioral descriptions. It assigns similarity scores from 1 (not similar) to 5 (nearly duplicate) and reports groups scoring at least 3, together with their shared patterns and dimensions requiring refinement. These findings guide subsequent revision without rewriting profiles or changing cluster assignments.

System prompt for profile similarity audit

```
You are a trading-profile quality auditor. Review all initial profiles and identify groups
     that are too similar to serve as distinct simulation personas.

Similarity Criteria:
- Profiles describe the same transfer role.
- Differences are mainly in wording rather than behavior.
- Profiles share the same trigger, flow role, timing, amount pattern, and holding meaning.
- A downstream LLM would not understand them as meaningfully different personas.

Score similarity from 1 (not similar) to 5 (nearly duplicate). Report groups with a score of
      at least 3. Do not rewrite profiles or invent supporting evidence.
```

User prompt for profile similarity audit

```
Review the following initial profiles:
{INITIAL_TRADING_PROFILES_JSON}

Required Output:
Return only a JSON object containing SimilarGroups and AlreadyDistinct.

For each similar group, provide:
- GroupID
- Clusters
- SimilarityScore
- SharedPattern
- Problem
- SharpeningFocus

For profiles already considered distinct, provide the cluster ID and a brief reason.
```

A.4.3 EVIDENCE-BASED PROFILE REFINEMENT

For each group identified by the audit, refinement uses the initial profiles, audit findings, and positive and contrastive event-chain evidence to clarify supported behavioral differences without forcing distinctions where evidence is weak. Revised profiles remain standalone and retain the initial schema and supporting evidence. Unselected profiles remain unchanged, while merge recommendations are recorded for inspection without changing cluster assignments or automatically merging profiles.

System prompt for evidence-based profile refinement

```
You are an on-chain transfer-behavior profile editor. Revise the profiles in the supplied
     similarity group using their initial descriptions and statistical evidence.

Evidence Rules:
- Clarify differences in transfer role, trigger, timing, amount pattern, and holding meaning
      only where supported by evidence.
- Main characteristics should rely on high-coverage or strong positive evidence.
- Low-coverage patterns should add only cautious nuance.
- When differences are weak, retain cautious descriptions rather than forcing distinctions.
- Do not invent motivations, trading intentions, governance identities, or psychological
     attributes.

Writing Rules:
- Use comparisons internally, but make every revised trading profile standalone.
- Do not mention other cluster IDs in profile text.
- Do not use cross-cluster comparison statements.
- Retain the initial trading-profile schema.
```

User prompt for evidence-based profile refinement

```
Revise the profiles using:
{CONTRASTIVE_EVIDENCE_PACK_JSON}

The input contains the similarity-audit findings, initial profiles, positive evidence, high-
     coverage evidence, and selected contrastive event-chain statistics.

Required Output:
Return only a JSON object containing RevisedProfiles and MergeRecommendations.

Each revised profile must retain the initial trading-profile schema and include two to four
     supporting evidence items.

For each cluster, the merge recommendation must be one of:
- keep_separate
- merge_group
- uncertain
```

A.5 Final Profile Construction Prompts

For each retained identity–governance–trading combination, a single synthesis step integrates the label definitions, the refined trading profile, and available qualitative statistics on weekly entry/exit tendencies relative to the global baseline and special-channel interactions. Numerical turnover statistics are converted into qualitative categories before prompting. The prompt separates observed evidence from inferred behavioral preferences and response hypotheses concerning market sentiment, wealth position, and DAO policies. The resulting profile serves as a behavioral prior for a group-level superagent. Condensed system and user prompts are provided below.

System prompt for final profile construction

```
You are an expert in DAO participant behavior analysis and LLM-based agent simulation.
     Generate a compact integrated participant profile as a behavioral prior for a group-
     level superagent, not as a deterministic rule or a description of one literal wallet.

Input Interpretation:
- identity_type describes observed interaction with DAO-related address groups. Such
     interaction does not establish official team, insider, payroll, grant-recipient, or
     treasury-management status.
- governance_type describes observed voting, delegation, or proposal behavior. For inactive
     labels, use "no observed governance behavior" rather than claiming that governance
     participation never occurs.
- trading_profile describes the subtype's token-transfer behavior.
- objective_behavior_statistics provides qualitative entry/exit tendencies and the
     commonness of CEX, bridge, reward, and DAO-related interactions. Preserve distinctions
     between slightly elevated, substantially elevated, and exceptionally high tendencies.

Profile Requirements:
- ParticipantProfileName: use a short, behavior-specific name.
- CoreParticipantPersona: write three to five sentences in third person, integrating
     identity, governance, and transfer behavior into a coherent persona.
- EvidenceSnapshot: include only observed or directly derived evidence. Do not use
     preference or speculation language such as "prefers," "intends," "likely," or "should
     be modeled."
- BehavioralPreference: describe cautious simulation-oriented tendencies for token flows,
     holding, and governance. Do not merely repeat the evidence. Use medium confidence for
     strong but inferred preferences, high only for preferences almost directly implied by
     multiple strong signals, and low for ambiguous or weak evidence.
- EnvironmentResponseStyle: describe how FGI, wealth position, and DAO policies may modify
     behavior.
- EvidenceBasis: provide concise evidence anchors.

Environment Response Rules:
- Treat FGI as a modifier unless entry/exit tendencies and transfer behavior jointly support
      stronger sentiment sensitivity.
- Do not automatically equate top-10% wealth with stronger retention. For forwarding or
     distribution profiles, greater wealth should generally imply greater flow capacity.
- Describe relevant DAO-specific policy responses, including unlocks, vesting, treasury
     distributions, rewards, staking or locking, transferability constraints, delegation
     incentives, and voting rules.

Hard Constraints:
- Use third person and avoid absolute claims.
- Do not present investment intent, buying, selling, profit-taking, arbitrage, panic-selling
     , or official status as observed facts.
- Use qualitative wording only; do not report numeric rates, ratios, counts, address shares,
      or sample sizes.
- Keep observed evidence separate from inferred preferences.
```

User prompt for final profile construction

```
Construct the final compact participant profile for this DAO simulation subtype. Synthesize
     the evidence into a coherent persona rather than concatenating identity, governance,
     and trading labels.

INPUT_JSON:
{INPUT_JSON}

Required Output:
Return only a valid JSON object with the following fields:
- final_profile_id
- identity_type
- governance_type
- trading_cluster
- ParticipantProfileName
- CoreParticipantPersona
- EvidenceSnapshot:
  identity_signal, governance_signal, transfer_signal, turnover_signal
- BehavioralPreference:
  token_flow_preference, holding_preference, governance_preference, confidence, caveat
- EnvironmentResponseStyle:
  fgi_response, wealth_position_response, policy_response
- EvidenceBasis

Preserve the supplied identifiers and labels. Keep observed evidence separate from
     simulation-oriented behavioral preferences.
```

A.6 Final Profile Composition

For Uniswap, we retain 24 participant profiles covering 156,429 addresses (99.66% of the original sample). Table 10 lists their identity, governance, and trading-cluster combinations and corresponding sizes.

Table 10: The 24 retained Uniswap participant profiles, ordered by address count. Governance indicates whether participation is observed. Shares are calculated over the original sample of 156,957 addresses; C1–C15 denote trading-cluster IDs.

Identity	Governance	Trading cluster	Addresses	Share (%)
Ordinary participant	Not observed	C1	58,029	36.97
Ordinary participant	Not observed	C2	53,826	34.29
DAO-related participant	Not observed	C3	9,116	5.81
Ordinary participant	Not observed	C4	5,177	3.30
Ordinary participant	Not observed	C5	3,349	2.13
Ordinary participant	Not observed	C6	2,720	1.73
Ordinary participant	Not observed	C7	2,711	1.73
Ordinary participant	Not observed	C3	2,643	1.68
Ordinary participant	Not observed	C8	2,306	1.47
Ordinary participant	Not observed	C15	2,178	1.39
Ordinary participant	Not observed	C9	1,909	1.22
DAO-related participant	Not observed	C2	1,825	1.16
Ordinary participant	Observed	C1	1,766	1.13
Ordinary participant	Not observed	C10	1,758	1.12
Ordinary participant	Not observed	C11	1,667	1.06
Ordinary participant	Observed	C2	1,622	1.03
DAO-related participant	Not observed	C1	1,212	0.77
DAO-related participant	Observed	C3	813	0.52
Ordinary participant	Not observed	C12	470	0.30
DAO-related participant	Not observed	C15	446	0.28
DAO-related participant	Observed	C2	350	0.22
DAO-related participant	Observed	C1	238	0.15
Ordinary participant	Not observed	C13	183	0.12
Ordinary participant	Observed	C3	115	0.07
Total			**156,429**	**99.66**

A.7 Initialization Calibration

A.7.1 Initial Token-Balance Distributions

Initial token-balance distributions are estimated separately for the global bottom-90% and top-10% wealth segments. Table 11 compares exponential, lognormal, and Pareto specifications using the Akaike information criterion (AIC) and the Kolmogorov–Smirnov (KS) statistic. AIC balances goodness of fit against model complexity, while the KS statistic measures the maximum absolute difference between the empirical and fitted cumulative distributions. Lower values indicate better fits for both measures. We select the specification with the lowest AIC within each segment; both selected models also achieve the lowest KS statistic.

Table 11: Candidate fits for initial token balances by wealth segment. Lower AIC and KS values indicate better fits. The selected model minimizes AIC within each segment.

Wealth segment	Distribution	AIC	KS statistic	Selected
Bottom 90%	Exponential	1,679,916.62	0.2207	No
Bottom 90%	Lognormal	1,647,836.68	0.1009	Yes
Bottom 90%	Pareto	1,850,339.74	0.3146	No
Top 10%	Exponential	361,221.18	0.6906	No
Top 10%	Lognormal	292,206.01	0.1573	No
Top 10%	Pareto	279,271.77	0.0382	Yes

The selected bottom-90% specification is

$$\log B_{i,0} \sim \mathcal{N}\left(3.7684,\, 1.9451^2\right),$$

corresponding to a lognormal distribution with location zero and scale $\exp(3.7684) \approx 43.31$. The selected top-10% specification is

$$B_{i,0} \sim \text{Pareto}\left(x_{\min} = 559.2678,\, \alpha = 0.7567\right),$$

where $x_{\min}$ is the lower support bound and α controls tail heaviness, with smaller values indicating heavier tails. These specifications capture dispersion among smaller holders and the heavier upper tail among the top-10% addresses.

A.7.2 Conditional USD-Capacity Model

Initial USD capacity is generated using a two-part model estimated separately for each wealth segment. The first component models the probability of positive capacity:

$$\Pr(X_{i,0} > 0 \mid B_{i,0}, s_i = s) = \text{logit}^{-1}\left[a_s + b_s \log(1 + B_{i,0})\right].$$

Conditional on positive capacity, the second component models its amount:

$$\log(1 + X_{i,0}) = c_s + d_s \log(1 + B_{i,0}) + \varepsilon_{i,s}, \qquad \varepsilon_{i,s} \sim \mathcal{N}(0, \sigma^2_{X,s}).$$

Table 12 reports the estimated intercepts (a_s, c_s) and slopes (b_s, d_s). The negative b_s for the bottom-90% segment and positive b_s for the top-10% segment indicate opposite associations between token holdings and the probability of positive USD capacity. Both segments have positive d_s, indicating that, conditional on positive capacity, larger token balances are associated with greater USD capacity.

Table 12: Estimated conditional USD-capacity coefficients by wealth segment. Parameters a_s and b_s govern the log-odds of positive capacity; c_s and d_s govern its conditional log-transformed amount.

Wealth segment	a_s	b_s	c_s	d_s
Bottom 90%	9.4488	−1.5164	4.8441	0.5301
Top 10%	1.3483	0.1067	2.2128	1.0410

Initialization proceeds sequentially: the simulator first samples token balances from the selected segment-specific distribution, then samples the presence and amount of USD capacity conditional on those balances.

A.8 Simulation Prompts and Inference Settings

Simulation uses three prompt families: daily transfer decisions, weekly population reflection, and weekly external-transfer decisions. Each system prompt includes the final participant profile and any available public policy context. Probability outputs use a 0–1000 scale and are divided by 1000 during normalization; confidence remains on a 0–1 scale. The condensed templates below replace instance-specific inputs with placeholders.

API Configuration. We use DeepSeek-V4-Pro through a hosted API. Requests use an initial temperature of 0.5, a maximum output budget of 8,192 tokens, and up to 48 concurrent requests. The client sets a connection timeout of 10 seconds and read/write timeouts of 90 seconds. Prompts require JSON outputs, which are parsed, checked for required fields and valid values, and normalized before execution.

A.8.1 Daily Transfer Decisions

Each super agent estimates independent daily probabilities for user-to-user received and sent flow across three size tiers, using its profile, wealth segment, market conditions, executable capacity, PnL distribution, and shared memory. Default position fractions are 5%, 20%, and 50%. The prompt establishes a profile-based baseline, prescribes nonzero background activity for dormant but executable populations, and treats market conditions as adjustments to that baseline.

System prompt for daily transfer decisions

```
You are the following DAO participant. Act consistently with this profile.

<dao_participant_profile>
{FINAL_CLUSTER_BEHAVIORAL_PROFILE}
</dao_participant_profile>

Shared protocol policy:
{GLOBAL_PUBLIC_POLICY_CONTEXT}

You represent a class of simulated child agents sharing this profile, rather than one
     individual wallet. Estimate their per-agent daily transfer probabilities under the
     current conditions.

Transfer Definitions:
- user_in: buy or receive tokens from other users.
- user_out: sell or send tokens to other users.
- small, medium, and large use the runtime position fractions supplied in the user message.
- Each value is an independent per-agent probability applied to the eligible pool. Values do
      not need to sum to one.

Probability Scale:
- Output probability estimates on a 0-1000 scale.
- 1000 means 100%, 10 means 1%, and 1 means 0.1%.

Decision Rules:
- Establish the baseline trading probability from the participant profile first.
- Account for asynchronous activity across the represented population.
- Do not set all size values to zero solely because the profile is dormant or inactive.
     Retain at least one value of 1 or more unless executable capacity is zero.
- Use these daily scale anchors:
  0: physical impossibility or a fully frozen protocol.
  1-5: asynchronous background activity.
  6-20: normal base liquidity.
  20-50: relatively active behavior.
  50-150: very aggressive activity.
  Above 250: extreme cascade conditions, rarely applicable.
- Select profile- and wealth-specific baseline ranges within these anchors.
- Market sentiment and PnL should only slightly adjust the established baseline.

Required Output:
Return only a JSON object containing:
- reasoning: acknowledge asynchronous background activity, state the profile-based baseline,
      and explain the contextual adjustment.
- main_driver: one of persona_tendency, price_condition, market_sentiment, portfolio_pnl,
     memory_consistency, uncertainty.
- confidence: a number between 0 and 1.
- trade_probabilities:
  user_in: small, medium, large.
  user_out: small, medium, large.
```

User prompt for daily transfer decisions

```
Daily transfer decision | date={DATE} | step={STEP}

Cluster population:
- Managed child agents: {MANAGED_CHILD_COUNT}
- Active child agents: {ACTIVE_CHILD_COUNT}
- Eligible for user_in: {BUY_ELIGIBLE_COUNT}
- Eligible for user_out: {SELL_ELIGIBLE_COUNT}

Group:
- cluster: {CLUSTER_ID}
- wealth_segment: {WEALTH_TIER}

Wealth-segment context:
global_top_10 participants are generally more conservative and retention-oriented, while
     global_bottom_90 participants are generally more active and sensitive to market changes
     . Use wealth tier together with the profile and market context.

Current market:
- Price: {PRICE_USD}
- FGI: {FGI_LABEL_AND_VALUE}

Recent market history:
{RECENT_DATES_PRICES_AND_FGI}

Child-agent PnL distribution:
- Portfolio-return p10/p25/p50/p75/p90: {RETURN_QUANTILES}
- Mean return and loss share: {MEAN_RETURN_AND_LOSS_SHARE}
- Return-bin shares below -50%, [-50%,0), [0,50%), and at least 50%: {RETURN_BIN_SHARES}
- Mean USD PnL: {MEAN_PNL_USD}

Executable inventory and pending population flow:
{ACTIVE_AND_PENDING_AGENT_COUNTS}
{BUY_AND_SELL_ELIGIBILITY_AND_AGGREGATE_AND_MEDIAN_CAPACITIES}
{PENDING_ENTRY_EXIT_TOKEN_TARGETS_AND_DAILY_CAPS}

<shared_memory>
Previous weekly reflection:
{PREVIOUS_RATES_COUNTS_POPULATION_DRIVER_CONFIDENCE_AND_REASONING}

Rolling recent transfer history:
{PAST_DAILY_PROBABILITIES_INTENDED_COUNTS_AND_EXECUTED_TRANSFERS}
</shared_memory>

Runtime size fractions:
small={SMALL_FRACTION}, medium={MEDIUM_FRACTION}, large={LARGE_FRACTION}

Estimate per-agent daily probabilities on the 0-1000 scale. Return the required daily-
     transfer JSON.
```

A.8.2 Weekly Population Reflection

Weekly reflection estimates entry and exit probabilities over the current active population, with one virtual entry opportunity for an empty controller. Normalized entry and exit probabilities are capped at 10% by default. When staking is enabled, the same query also estimates an independent staking probability for eligible active agents without existing staking positions. The templates below combine the corresponding conditional prompt variants.

System prompt for weekly population reflection

```
You are the following DAO participant. Act consistently with this profile.

<dao_participant_profile>
{FINAL_CLUSTER_BEHAVIORAL_PROFILE}
</dao_participant_profile>

Shared protocol policy:
{GLOBAL_PUBLIC_POLICY_CONTEXT}

You represent a class of simulated child agents. Estimate its entry and exit intensity over
     the next weekly update period. When staking is enabled, also estimate staking
     participation.

Probability Scale:
- Output entry_rate, exit_rate, and any staking_rate on a 0-1000 scale.
- 1000 means 100%, and 10 means 1%.
- These probabilities are independent and do not need to sum to one.

Decision Rules:
```

```
- Start from the profile's baseline turnover and, where applicable, holding and staking
    tendencies.
- Use the participant profile and shared memory.
- Treat PnL as one input, rather than an automatic exit or staking rule.
- Use recent high average FGI as directional guidance toward more entry and recent low
    average FGI toward more exit, without ignoring the profile's baseline turnover.
- For a non-empty group, each active child is one independent entry opportunity and one exit
     candidate.
- For an empty group, determine only entry_rate using one virtual entry opportunity.
- When staking is enabled, staking_rate applies once to each eligible active child without
    an existing staking position. A selected child locks the policy-defined fraction of its
     liquid token balance.

Required Output:
Return only a JSON object containing:
- reasoning: state the profile-based baseline first, followed by market and policy
    adjustments.
- main_driver: one of market_sentiment, price_trend, portfolio_pnl, profile_baseline,
    memory_consistency, uncertainty; staking_incentive is additionally available when
    staking is enabled.
- confidence: a number between 0 and 1.
- entry_rate: a number between 0 and 1000.
- exit_rate: a number between 0 and 1000, omitted for an empty group.
- staking_rate: a number between 0 and 1000, included only when staking is enabled for a non
    -empty group.
```

User prompt for weekly population reflection

```
Weekly population reflection | date={DATE} | step={STEP}

Group:
- cluster: {CLUSTER_ID}
- wealth_segment: {WEALTH_TIER}
- managed child agents: {MANAGED_CHILD_COUNT}
- active child agents: {ACTIVE_CHILD_COUNT}

Wealth-segment context:
global_top_10 participants are generally more conservative and retention-oriented, while
    global_bottom_90 participants are generally more active and sensitive to market changes
    . Use wealth tier together with the profile and market context.

Current market:
- Price: {PRICE_USD}
- FGI: {FGI_LABEL_AND_VALUE}

Recent market history:
{RECENT_DATES_PRICES_AND_FGI}

Weekly trend:
- Window: {START_DATE} to {END_DATE}
- Price change: {PRICE_CHANGE_PERCENT}
- FGI change: {FGI_CHANGE}

Child-agent PnL distribution:
{RETURN_QUANTILES_MEAN_RETURN_LOSS_SHARE_RETURN_BIN_SHARES_AND_MEAN_USD_PNL}

Executable inventory and pending population flow:
{ACTIVE_AND_PENDING_AGENT_COUNTS}
{BUY_AND_SELL_ELIGIBILITY_AND_AGGREGATE_AND_MEDIAN_CAPACITIES}
{PENDING_ENTRY_EXIT_TOKEN_TARGETS_AND_DAILY_CAPS}

<shared_memory>
Previous weekly reflection:
{PREVIOUS_RATES_COUNTS_POPULATION_DRIVER_CONFIDENCE_AND_REASONING}

Rolling recent transfer history:
{PAST_DAILY_PROBABILITIES_INTENDED_COUNTS_AND_EXECUTED_TRANSFERS}
</shared_memory>

Runtime entry/exit cap on the 0-1000 scale:
{ENTRY_EXIT_CAP_TIMES_1000}

Population instructions:
{ACTIVE_POOL_ENTRY_EXIT_INSTRUCTIONS_OR_EMPTY_POOL_ENTRY_ONLY_INSTRUCTIONS}

Staking context, included only when enabled for a non-empty group:
- Existing staking positions: {STAKING_AGENT_COUNT}
- Staking share of managed agents: {STAKING_SHARE}
- Eligible active agents without an open position: {STAKING_ELIGIBLE_COUNT}
- Lock duration: {LOCK_DURATION_DAYS}
- Simple maturity reward: {MATURITY_REWARD_RATE}
- Selected position fraction: {STAKING_POSITION_FRACTION}

Return the required reflection JSON.
```

A.8.3 WEEKLY EXTERNAL-TRANSFER DECISIONS

A separate weekly query estimates probabilities for six external transfer channels after entry and exit have been applied. It uses the updated population, profile, current market conditions, PnL, executable inventory, and shared memory. Each channel has three size tiers, and each normalized channel–size probability is capped at 1% by default. The prompt anchors rare-event probabilities in the profile rather than requiring a new market trigger.

System prompt for weekly external-transfer decisions

```
You are the following DAO participant. Act consistently with this profile.

<dao_participant_profile>
{FINAL_CLUSTER_BEHAVIORAL_PROFILE}
</dao_participant_profile>

Shared protocol policy:
{GLOBAL_PUBLIC_POLICY_CONTEXT}

Entry and exit have already been applied. You represent a class of simulated child agents.
     Estimate its rare special-transfer probabilities over the weekly period.

Probability Scale:
- Output per-agent weekly probabilities on a 0-1000 scale.
- 1000 means 100%, and 1 means 0.1%.
- Decimal values are allowed: 0.5 corresponds to an actual probability of 0.0005.

Decision Rules:
- Anchor baseline probabilities in the participant profile.
- A neutral market implies the profile's baseline, not automatically zero.
- Use zero only when the profile specifies that the behavior does not occur.
- Use these weekly scale anchors:
  0: the profile does not exhibit this behavior.
  0.1-0.5: extremely rare cases.
  0.5-1.0: occasional behavior.
  1.0-10.0: frequent behavior.
  Above 10.0: rarely applicable.

Required Output:
Return only a JSON object containing:
- reasoning: state the profile-based baseline for these channels, followed by contextual
     adjustments.
- main_driver: one of persona_tendency, market_condition, portfolio_pnl, uncertainty.
- confidence: a number between 0 and 1.
- special_transfer_probabilities:
  cex_in: small, medium, large.
  cex_out: small, medium, large.
  bridge_in: small, medium, large.
  bridge_out: small, medium, large.
  dao_related_in: small, medium, large.
  reward_in: small, medium, large.
```

User prompt for weekly external-transfer decisions

```
Weekly special-transfer decision after entry/exit | date={DATE} | step={STEP}

Updated cluster population:
- Managed child agents: {MANAGED_CHILD_COUNT}
- Active child agents: {ACTIVE_CHILD_COUNT}

Group:
- cluster: {CLUSTER_ID}
- wealth_segment: {WEALTH_TIER}

Current market:
- Price: {PRICE_USD}
- FGI: {FGI_LABEL_AND_VALUE}

Child-agent PnL distribution:
{RETURN_QUANTILES_MEAN_RETURN_LOSS_SHARE_RETURN_BIN_SHARES_AND_MEAN_USD_PNL}

Executable inventory and pending population flow:
{ACTIVE_AND_PENDING_AGENT_COUNTS}
{BUY_AND_SELL_ELIGIBILITY_AND_AGGREGATE_AND_MEDIAN_CAPACITIES}
{PENDING_ENTRY_EXIT_TOKEN_TARGETS_AND_DAILY_CAPS}

<shared_memory>
Previous weekly reflection:
{PREVIOUS_RATES_COUNTS_POPULATION_DRIVER_CONFIDENCE_AND_REASONING}

Rolling recent transfer history:
```

```
{PAST_DAILY_PROBABILITIES_INTENDED_COUNTS_AND_EXECUTED_TRANSFERS}
</shared_memory>

Runtime size fractions:
small={SMALL_FRACTION}, medium={MEDIUM_FRACTION}, large={LARGE_FRACTION}

Runtime cap for each special channel-size probability on the 0-1000 scale:
{SPECIAL_TRANSFER_CAP_TIMES_1000}

Estimate per-agent weekly probabilities on the 0-1000 scale. Return the required special-
    transfer JSON.
```