BEYOND MARKET CONSENSUS: MODELING TRADER HETEROGENEITY THROUGH BEHAVIORAL RESIDUALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Realistic market simulation requires agents to preserve persistent behavioral differences, yet large language model (LLM) agents often converge toward average behavior when exposed to similar market conditions. We argue that trader heterogeneity can be characterized by how individuals systematically deviate from shared market consensus. Based on this insight, we propose Consensus-Guided Residual Behavior Modeling (CRBM), which decomposes trading behavior into a consensus component shared across traders and a behavioral residual capturing trader-specific deviations. CRBM derives consensus signals from local market contexts, estimates persistent residual patterns from individual transaction histories, and integrates them through a residual-conditioned expert mechanism to generate structured trading orders. Using real transaction records from Polymarket, we evaluate CRBM at three levels: individual behavioral fidelity, cross-trader heterogeneity, and emergent market dynamics. CRBM consistently improves transaction-level prediction and joint action–outcome matching over competitive baselines, while ablations demonstrate the complementary roles of consensus modeling and behavioral residuals. In Monte Carlo market replays, CRBM preserves wallet-specific behavioral differences and produces aggregate price dynamics consistent with observed market evolution. Zero-shot evaluation on Manifold further provides preliminary evidence of cross-platform generalization. These results suggest that modeling individual behavior as structured deviations from shared consensus provides an effective approach to preserving agent heterogeneity in LLM-based market simulation.

1 INTRODUCTION

Prediction markets aggregate information from distributed participants through real-world trading, with market prices commonly interpreted as collective estimates of future event probabilities Wolfers & Zitzewitz (2004); Berg et al. (2008). Meanwhile, large language models (LLMs) are used to construct agents that simulate human decision-making and interaction in complex environments Park et al. (2023); Horton (2023). Public prediction-market platforms such as Polymarket provide transaction-level trajectories that make it possible to learn trading agents from observed behavior rather than manually specified rules or profiles.

However, existing LLM agents typically represent such differences through manually designed roles, textual profiles, or preference instructions Park et al. (2023); Santurkar et al. (2023); Lee et al. (2024). They often overlook the fact that these outcomes emerge from traders with persistent behavioral differences rather than from a single average participant. Under similar market conditions, traders may differ in their directional choices and trading scales. Existing LLM agents typically represent such differences through manually designed roles, textual profiles, or preference instructions, which lack direct supervision from real trading trajectories. We therefore study heterogeneous trading behavior generation rather than event-outcome or price forecasting. Unlike aggregate forecasting, this task requires a model to reproduce both the shared response associated with a market state and systematic variation across wallets. Accurate isolated orders are insufficient if repeated generations collapse toward an average strategy. Behavioral fidelity must therefore be examined

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.

across individual transactions, persistent wallet tendencies, emergent trader groups, and their aggregate market consequences.

The central challenge is that heterogeneity cannot be inferred from isolated orders. The same action–outcome decision may reflect the local market state or a persistent wallet-specific tendency, and these effects remain confounded without a common behavioral reference. Conversely, manually written personas provide no evidence that an agent reproduces deviations that persist across repeated real decisions. A useful model must therefore compare traders under similar contexts and use their historical trajectories to distinguish persistent wallet-level patterns from context-specific variation.

We formulate heterogeneous trading behavior through an interpretable consensus–residual decomposition. Market consensus provides a shared behavioral reference under the current market conditions, while wallet-level residuals summarize how a wallet's earlier trading behavior deviates from corresponding market references. Based on this decomposition, we introduce *Consensus-Guided Residual Behavior Modeling* (CRBM), which constructs market consensus from strictly preceding other-wallet transactions, estimates a minimal behavioral residual from a fixed wallet profile segment, and conditions structured order generation on both components. The generated orders contain trading actions, outcome selections, and trade sizes. In our implementation, a residual-conditioned expert mechanism adapts the generation pathway to the profile residual. This mechanism is one implementation of CRBM rather than the definition of the broader consensus–residual framework.

We collect real Polymarket transaction records and evaluate CRBM at the transaction, trader, and market levels. The transaction-level experiments assess behavioral fidelity and the contributions of individual model components. Monte Carlo replays examine whether generated behavior preserves wallet-level tendencies, forms distinguishable trader populations, and produces aggregate price dynamics consistent with the observed event progression. A zero-shot Manifold case study further evaluates cross-platform transfer. CRBM achieves the best performance on outcome selection and joint action–outcome matching among the evaluated methods, while the multi-level results show that it preserves persistent trader differences and generates meaningful collective market outcomes.

Our contributions are summarized as follows:

- We formulate heterogeneous trading behavior generation using an interpretable consensus–residual framework under a strictly past-only information boundary, where consensus is constructed exclusively from preceding transactions of other wallets.
- We derive a minimal and interpretable wallet profile residual from real transaction trajectories using economic-direction deviation and relative trading scale, and instantiate residual-aware generation with a profile-conditioned MoE-LoRA mechanism.
- We develop a profile-to-held-out evaluation protocol and conduct comprehensive experiments on real Polymarket trajectories, including baseline comparisons, component and residual-feature ablations, wallet-level behavioral preservation, population-level structure, and constrained historical market replay. The results validate CRBM's effectiveness across transaction, wallet, population, and market levels, while a zero-shot Manifold case study provides preliminary evidence of cross-platform transferability.

2 Problem Formulation

We study heterogeneous trading behavior generation from real prediction-market trajectories. For target transaction i, let w_i, $\mathcal{M}_i$, and t_i denote its wallet, market, and execution time. Let x_i be the market state immediately before t_i, and let $y_i = (a_i, o_i, s_i)$ denote the observed action, outcome, and trade size. Our objective is to generate the wallet's structured order, rather than to predict the eventual event outcome or future market price.

For each wallet w, we chronologically divide its transactions into an earlier profile segment $\mathcal{H}_w^p$ and a later held-out segment $\mathcal{H}_w^h$, containing approximately 70% and 30% of its transactions, respectively. The profile segment is used to estimate the input residual r_w^p. Transactions in the held-out segment are never used to construct this representation. A separate held-out residual r_w^h is computed only after generation for wallet-level evaluation and is never provided to the model.

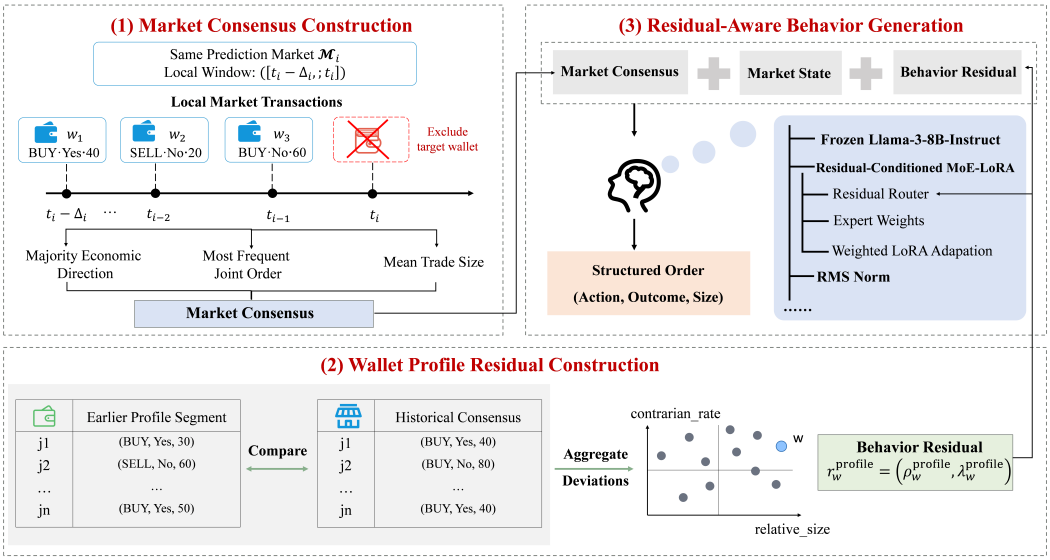


Figure 1: Overview of CRBM. For each target transaction, market consensus is constructed from strictly preceding transactions of other wallets in the same market, while the wallet profile residual is computed from a fixed earlier profile segment. The pre-trade market state, causal market consensus, and profile residual condition structured order generation. The held-out residual is computed only for evaluation and is never supplied to the model.

For target transaction i, the consensus reference set contains only earlier transactions from other wallets in the same market:

$$\mathcal{G}_i^- = \{j \mid \mathcal{M}_j = \mathcal{M}_i,\ t_i - \Delta_i \le t_j < t_i,\ w_j \neq w_i\}, \qquad c_i = \mathcal{C}(\mathcal{G}_i^-), \tag{1}$$

where Δ_i is an backward-looking window and $\mathcal{C}$ denotes the consensus aggregation function. Thus, neither the target transaction nor any transaction at or after t_i is available when constructing c_i.

CRBM formulates structured order generation as

$$p_\theta\left(y_i \mid x_i, c_i, r^p_{w_i}\right), \tag{2}$$

where the market state, causal consensus, and wallet profile jointly condition the generated order. This past-only information boundary prevents look-ahead leakage; it does not imply causal identification or support interventional claims.

3 Methodology

Figure 1 summarizes the three stages of CRBM. First, for target transaction i, the model constructs a past-only market consensus c_i from earlier transactions of other wallets in the same market. The target wallet and all transactions at or after t_i are excluded. Second, CRBM computes the profile residual $r^p_{w_i}$ from the wallet's earlier profile segment, without using held-out transactions. Third, the pre-trade market state x_i, market consensus c_i, and profile residual $r^p_{w_i}$ condition order generation. The profile residual is included in the prompt and determines the LoRA-expert mixture weights.

The held-out residual $r^h_{w_i}$ is computed from real held-out transactions solely for wallet-level evaluation and is never supplied to the prompt, router, or generation model. The residual-conditioned MoE-LoRA architecture operationalizes CRBM in this study but does not define the broader consensus–residual framework.

3.1 Market Consensus Construction

Market consensus c_i summarizes the behavior of other wallets in the same market before target transaction i. We use the causal reference set $\mathcal{G}_i^-$ defined in Equation 1, which excludes the target wallet and all transactions executed at or after t_i.

To obtain sufficient reference activity without using future information, we consider windows of 10 minutes, 1 hour, 6 hours, and 24 hours. We use the shortest window containing at least 10 valid transactions. A target is excluded if fewer than 10 valid references are available within 24 hours.

We define economic direction according to the exposure of an order to the YES outcome. BUY Yes and SELL No are treated as BULLISH-YES, whereas BUY No and SELL Yes are treated as BEARISH-YES. The consensus direction d_i^c is the majority economic direction in $\mathcal{G}_i^-$. Within that direction, we select the most frequent observed action–outcome pair (a_i^c, o_i^c). This avoids constructing an order from independently aggregated action and outcome labels.

We also compute the mean reference size $\bar{s}_i^c$. The resulting market consensus is

$$c_i = (d_i^c, a_i^c, o_i^c, \bar{s}_i^c)\,.$$

It therefore contains the prevailing economic direction, a representative order realization, and the local trade-size scale. Additional validity, window-selection, and tie-handling details are provided in Appendix A.1.

3.2 WALLET-LEVEL BEHAVIORAL RESIDUAL MODELING

CRBM represents persistent wallet differences using transactions from the wallet's earlier profile segment. Let $\mathcal{V}_w$ denote the valid profile transactions for which a past-only market consensus is available, and let $N_w = |\mathcal{V}_w|$. For transaction j, d_j and d_j^c denote the wallet and consensus economic directions, while s_j and $\bar{s}_j^c$ denote the wallet and consensus trade sizes.

We characterize wallet w using two quantities:

$$\rho_w^p = \frac{1}{N_w} \sum_{j \in \mathcal{V}_w} \mathbb{I}[d_j \neq d_j^c], \qquad \lambda_w^p = \frac{\mathrm{mean}_{j \in \mathcal{V}_w}\, s_j}{\mathrm{mean}_{j \in \mathcal{V}_w}\, \bar{s}_j^c + \epsilon}. \tag{3}$$

Here, ρ_w^p is the fraction of profile transactions that oppose the market consensus, and λ_w^p measures the wallet's average trade size relative to the local market reference. Orders are considered consistent when they express the same economic direction, even if they use different BUY/SELL realizations.

The profile residual supplied to the model is $r_w^p = (\rho_w^p, \lambda_w^p)$. It is a minimal behavioral summary rather than a complete trader profile. For wallet-level evaluation, we apply the same calculation to the held-out transactions to obtain r_w^h. The held-out residual is used only as an evaluation reference and is never provided to the model.

3.3 RESIDUAL-AWARE BEHAVIOR GENERATION

CRBM uses the profile residual for both prompt conditioning and parameter adaptation. For target transaction i, the prompt P_i contains the pre-trade market state x_i, the past-only market consensus c_i, and the wallet profile residual $r_{w_i}^p$. The model then generates the structured order $\hat{y}_i = (\hat{a}_i, \hat{o}_i, \hat{s}_i)$. The held-out residual $r_{w_i}^h$ is never included in the prompt or otherwise provided to the model.

We implement residual-aware generation using a frozen Llama-3-8B-Instruct backbone Dubey et al. (2024) and K LoRA experts Hu et al. (2022). A trainable router converts the profile residual into expert weights. For a frozen linear layer with input h, the adapted output is

$$\boldsymbol{\pi}_i = \mathrm{softmax}\big(g_\phi(r_{w_i}^p)\big)\,, \qquad h' = Wh + \frac{\alpha}{R} \sum_{k=1}^{K} \pi_{ik} B_k A_k h. \tag{4}$$

Here, W remains frozen, while the LoRA parameters $\{A_k, B_k\}_{k=1}^K$ and router g_ϕ are trainable. Consequently, wallets with different profile residuals receive different mixtures of expert adaptations without requiring separate foundation models.

The trainable parameters are optimized using standard supervised fine-tuning:

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_{i=1}^{N} \log p_\theta(y_i \mid P_i). \tag{5}$$

The target order y_i is used only as training supervision and is not included in the input prompt. Architectural and training details are provided in Appendix A.3 and Appendix B.2.

Category	Method	Parse ↑	Act. Acc. ↑	Act. F1 ↑	Out. Acc. ↑	Out. F1 ↑	Strict ↑	Size Log-MAE ↓
Rule-based	Historical Profile	1.0000	0.8445	0.4665	0.5011	0.4908	0.4449	1.0981
	Majority Following	1.0000	0.8524	0.5355	0.4295	0.4253	0.3612	1.6694
Structured ML	LightGBM	1.0000	0.7159	0.5414	0.5121	0.4975	0.3866	**0.9258**
	MLP	1.0000	0.6927	0.5348	0.5286	0.5197	0.3535	0.9443
General LLMs	Llama-3 (No FT)	1.0000	0.7445	0.5044	0.5683	0.3693	0.4604	0.9446
	GPT-4o Zero-Shot	1.0000	0.5779	0.5027	0.4772	0.3269	0.2954	1.1800
Pers./Pref.	JANUS	1.0000	0.8359	0.5190	0.4934	0.4902	0.4174	1.1838
	P-Soups	1.0000	0.8445	0.4665	0.5319	0.5052	0.4637	0.9756
	CoPL-style	1.0000	0.8645	0.4637	0.4791	0.4788	0.4174	1.2033
Ours	**CRBM**	1.0000	**0.8722**	**0.5477**	**0.5769**	**0.5521**	**0.4707**	1.1140

Table 1: Transaction-level trading behavior generation results under the causal wallet-disjoint v16 validation protocol. F1 scores are macro-averaged; lower is better for Size Log-MAE.

4 EXPERIMENTS

Research Questions. Our experiments address three research questions derived from the consensus–residual modeling framework.

RQ1: Under a strictly past-only information boundary, how accurately does CRBM generate held-out structured trading behavior, and which components contribute to its performance? We evaluate action, outcome, joint action–outcome, and trade-size fidelity on a wallet-disjoint validation set, together with component ablations, residual-feature ablations, expert-count sensitivity, and a shuffled-residual control.

RQ2: Does CRBM preserve heterogeneous trading behavior beyond individual transactions? We examine whether residuals estimated from an earlier wallet profile predict subsequent unseen behavior by comparing generated residuals with those computed from held-out transactions. We then evaluate population-level behavioral structure and aggregate price-path reconstruction under constrained historical replay. Agreement between generated residuals and the input profile residual is reported separately as residual adherence rather than evidence of held-out behavioral preservation.

RQ3: Does the consensus–residual modeling paradigm exhibit preliminary transferability across prediction-market platforms? We evaluate the Polymarket-trained model on Manifold without platform-specific fine-tuning and analyze performance across varying user histories.

4.1 EXPERIMENTAL SETUP

Data and Splits. We construct a structured trading-behavior dataset from real Polymarket transaction trajectories. Each sample contains the pre-trade market state x_i, a past-only market consensus c_i constructed from preceding transactions of other wallets, the target wallet's profile residual $r_{w_i}^{\text{profile}}$, and the observed order $y_i = (a_i, o_i, s_i)$. For each wallet, the earliest 70% of its chronologically ordered transactions form the profile segment, while the latest 30% form the held-out target segment. After past-only feature construction and validity filtering, the dataset contains 128 wallets, which are partitioned into wallet-disjoint training, validation, and test sets containing 89, 13, and 26 wallets, respectively. Their target segments contain $10{,}806$, 908, and $2{,}316$ samples. The training data may be resampled to address class imbalance, whereas the validation and test sets retain their natural distributions. The validation set is used for checkpoint selection and transaction-level model comparison, whereas the test set is used for the wallet-level and multi-level analyses reported in Section 4.3. Detailed construction, filtering, split statistics, and temporal-integrity checks are provided in Appendix A.1, Appendix A.2, and Appendix B.1.

Baselines and Implementation. We compare CRBM with rule-based methods (Historical Profile and Majority Following), structured models (LightGBM and MLP), general-purpose language models (Llama-3-8B-Instruct without task-specific fine-tuning and GPT-4o zero-shot), the personalized alignment method JANUS, and preference-based methods (P-Soups and CoPL) Ke et al. (2017); Dubey et al. (2024); OpenAI (2024); Lee et al. (2024); Jang et al. (2024); Choi et al. (2025). All methods use the same wallet-disjoint splits and a common structured output schema. No method re-

ceives transactions from the target wallet's held-out segment as input. CRBM uses a frozen Llama-3-8B-Instruct backbone with two LoRA experts and a total LoRA rank budget of 16. Complete baseline inputs, hyperparameters, training configurations, and computing infrastructure are reported in Appendix B.2, Appendix B.3, and Appendix B.5. Unless otherwise specified, all stochastic training experiments use a fixed random seed of 42.

Evaluation. At the transaction level, we report Action and Outcome Accuracy and F1, Strict Accuracy requiring both fields to be correct, Size Log-MAE, and Parse Rate as a format-validity diagnostic. At the wallet level, we compare the generated residual r_w^g with the real held-out residual r_w^h, separately for the directional and scale components, using Spearman rank correlation and mean absolute error. We also compare r_w^g with the input profile residual r_w^p as a secondary measure of residual adherence rather than held-out preservation. At the population level, we examine the generated residual distribution and use clustering and silhouette analysis as descriptive measures of behavioral structure. At the market level, Monte Carlo historical replay is evaluated using price-path RMSE, dynamic time warping distance, price-change correlation, and final-price error. Complete definitions and statistical procedures are provided in Appendix B.4.

4.2 Transaction-Level Behavioral Fidelity

Overall Comparison. To evaluate whether CRBM more faithfully generates observed order-placement behavior, we compare it with all baselines on the same wallet-disjoint validation set. Table 1 reports performance on trading action, outcome selection, joint action–outcome matching, and trade size.

CRBM achieves the best Action Macro-F1, Outcome Accuracy, Outcome Macro-F1, and Strict Accuracy, obtaining 0.5477, 0.5769, 0.5521, and 0.4707, respectively. Majority Following remains competitive on Action Macro-F1, indicating that market consensus contains behavioral signals. Compared with MLP, CRBM improves Outcome Macro-F1 and Strict Accuracy by 0.0324 and 0.1172, respectively, supporting the use of residual-aware generation for joint order behavior. CRBM does not achieve the lowest Size Log-MAE; LightGBM performs best on this metric, suggesting that trade size may depend on additional information such as capital, position, and liquidity.

Mechanism Ablation. To identify which components contribute to heterogeneous behavior generation, we ablate its group-level consensus, wallet-level residual representation, and residual-aware parameter adaptation. Table 2 combines the component and behavioral-feature ablations.

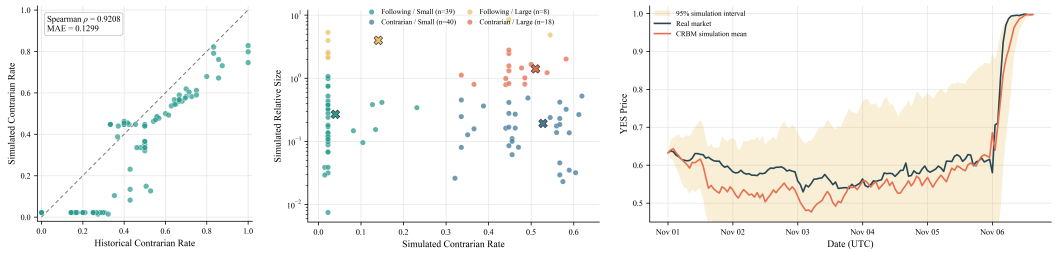
Setting	Act. F1 ↑	Out. F1 ↑	Strict ↑	Size Log-MAE ↓
w/o Consensus	0.4665	0.5088	0.4449	1.3389
w/o Residuals	0.5198	0.4960	0.4229	1.7624
Single LoRA	0.4665	0.5399	0.4319	1.2379
w/o Gating	0.4662	0.5472	0.4468	1.2079
Contrarian Only	0.4987	0.5093	0.4306	1.7454
Relative Size Only	0.5303	0.4829	0.4141	1.4110
CRBM	**0.5477**	**0.5521**	**0.4707**	**1.1140**

Table 2: Ablation results for CRBM.

Removing the behavioral residual substantially degrades performance. Outcome Macro-F1 decreases from 0.5521 to 0.4960, Strict Accuracy decreases from 0.4707 to 0.4229, and Size Log-MAE increases from 1.1140 to 1.7624. This indicates that market context and consensus alone are insufficient to represent wallet-level behavioral differences.

Removing market consensus also reduces Outcome Macro-F1 to 0.5088 and increases Size Log-MAE to 1.3389, supporting the role of consensus as a shared reference against which wallet-specific deviations are defined. Contrarian Only obtains higher Outcome Macro-F1 than Relative Size Only (0.5093 versus 0.4829), whereas Relative Size Only performs better on Action Macro-F1 and Size Log-MAE. Combining both residual dimensions provides the best overall balance.

Replacing the multi-expert architecture with a single LoRA reduces Outcome Macro-F1 to 0.5399, Strict Accuracy to 0.4319, and increases Size Log-MAE to 1.2379. Disabling residual-aware gating



(a) **(b)** **(c)**

Figure 2: Multi-level evaluation of CRBM-generated trading behavior. (a) Wallet-level residual evaluation. (b) Population structure in the generated behavioral space. (c) Observed and historically replayed price paths; the shaded region denotes the 2.5th–97.5th percentile interval across Monte Carlo runs.

similarly reduces Strict Accuracy to 0.4468 and increases Size Log-MAE to 1.2079. These results indicate that both the multi-expert adaptation space and residual-conditioned routing contribute to CRBM's performance.

4.3 Multi-Level Behavioral Evaluation

Transaction-level accuracy alone cannot establish whether generated agents preserve persistent wallet-level differences or whether their decisions form population- and market-level patterns. We therefore complement transaction-level evaluation with analyses at three levels: held-out wallet-level preservation, population-level behavioral structure, and constrained historical market replay.

Evaluation Protocol. For each test wallet w, let r_w^g denote the residual estimated from generated target-segment transactions. Our primary wallet-level evaluation compares r_w^g with the residual r_w^h computed from the wallet's real held-out transactions. This comparison measures preservation of unseen wallet-level behavior without supplying the evaluation target to the model. As a secondary diagnostic, we compare r_w^g with the input profile residual r_w^p. This measures adherence to the conditioning signal rather than preservation of unseen behavior, because r_w^p is supplied to the model.

At the population level, we examine whether generated behavior collapses toward a common pattern or retains variation in direction and trade size. At the market level, we aggregate generated orders in a historical replay and compare the resulting price path with the observed trajectory. Because the replay fixes the historical market contexts, wallet-activation process, and price-update rule, it measures controlled price-path reconstruction rather than fully endogenous agent–market emergence.

Held-Out Wallet-Level Preservation. For each test wallet, we compare the directional and scale-related residuals recovered from generated target-segment transactions with the residuals computed from the real held-out transactions. We use rank correlation to evaluate whether the model preserves cross-wallet ordering and absolute error to measure the magnitude of the discrepancy.

Residual Adherence. As a secondary diagnostic, we compare the residual recovered from generated target-segment transactions with the profile residual supplied to the model. This comparison measures whether generated behavior responds to the conditioning signal; it does not evaluate whether the profile residual predicts the wallet's unseen held-out behavior. Complete residual-adherence results are reported in Appendix C.2.

Population-Level Structure. We evaluate whether CRBM collapses wallets toward an average behavioral pattern or preserves population-level heterogeneity. After generating target-segment transactions, we recompute the directional and scale-related residuals for each wallet and analyze their distribution in the resulting behavioral space. Clustering is used as a descriptive tool for examining population structure rather than as direct evidence that the discovered groups correspond to ground-truth trader types.

Method	RMSE ↓	MAE ↓	Up/Down ↑	Flow Corr. ↑	Change Corr. ↑	DTW ↓
Historical Profile	0.1256	0.0708	0.4297	-0.3172	-0.0475	0.0311
Majority Following	0.0484	0.0367	0.5156	0.1353	0.1626	0.0183
Llama-3 (No FT)	0.2237	0.1128	0.5169	-0.1541	-0.1133	0.0618
P-Soups	0.1333	0.0731	0.5014	0.1479	0.1344	0.0375
CRBM	**0.0441**	**0.0362**	**0.5234**	**0.1499**	**0.2133**	**0.0095**

Table 3: Price-path reconstruction results under the controlled historical replay protocol. Lower is better for RMSE, MAE, and DTW.

K-means clustering with $K = 4$ identifies four descriptive groups along the contrarian-rate and relative-size dimensions, with a silhouette score of 0.5770.

Historical Market Replay. We aggregate generated orders to evaluate whether they reconstruct the observed price trajectory under controlled historical conditions. The replay retains the historical sequence of market contexts and uses a predefined wallet-activation and price-update process. It therefore evaluates aggregate consistency with an observed event path rather than fully endogenous market emergence or deployable price forecasting.

The historical replay uses the Polymarket market "Will Donald Trump Win the 2024 U.S. Presidential Election?" The observed sequence contained 135 hourly prices and 134 consecutive decision scenarios. A total of 105 wallets produced one decision for each scenario, resulting in $105 \times 134 = 14{,}070$ generated decisions per method.

The replay was divided into pre-election, vote-counting, and result-largely-certain phases, with wallet activation probabilities of 0.10, 0.50, and 0.85, respectively. BUY YES and SELL NO generated positive pressure on the YES price, whereas SELL YES and BUY NO generated negative pressure. Signed orders were weighted by trade size and aggregated into standardized net order flow.

4.4 Cross-Platform Evaluation on Manifold

We conduct a zero-shot cross-platform evaluation by applying the Polymarket-trained CRBM to Manifold without platform-specific fine-tuning. After filtering, the evaluation set contains 949 target trades from 349 users across 54 binary markets, with all behavioral features constructed only from transactions preceding each target. CRBM obtains an Outcome Accuracy of 0.6101, an Outcome F1 of 0.6078, and a Size Log-MAE of 1.2133. Because Manifold uses different trading mechanisms and action semantics, and because no Manifold-specific fine-tuning or retrained cross-platform baselines are included, these results provide preliminary evidence of transferability rather than conclusive evidence of domain generalization. Data processing, action mapping, and history-stratified results are reported in Appendix C.4.

5 Discussion

Behavior Modeling Rather Than Market Forecasting. CRBM models how traders place orders under observed market contexts; it does not forecast future outcomes or prices. Market consensus is only a behavioral reference, not a rational or correct signal, and price replay tests the aggregation of generated behavior rather than deployable forecasting performance.

Scope of the Residual Representation. Our two-dimensional residual—contrarian tendency and relative trade size—is a minimal, interpretable representation rather than a trader profile. Richer histories could capture positions, capital, trading frequency, price sensitivity, and liquidity responses; the Manifold results also show that reliable residual estimation requires sufficient user history.

Implementation and Simulation Scope. Residual-conditioned MoE-LoRA is one implementation of CRBM, and alternative conditional adaptation mechanisms remain possible. Our Monte Carlo replay uses historical contexts and fixed update rules rather than fully endogenous

agent–market feedback, motivating future evaluation with alternative routing methods and closed-loop market simulations.

6 Related Work

6.1 Prediction Markets and Heterogeneous Trading Behavior

Prediction markets aggregate dispersed information through trading, and their prices are commonly interpreted as collective estimates of event probabilities Wolfers & Zitzewitz (2004); Berg et al. (2008). Prior research has focused primarily on aggregate properties such as price accuracy, information efficiency, trading volume, and final market outcomes Hanson (2003); Chen & Pennock (2010). Market-microstructure models explain how these aggregate signals arise from traders with heterogeneous information, incentives, and strategies Kyle (1985); Glosten & Milgrom (1985); De Long et al. (1990). Empirical and theoretical studies have further documented persistent differences in trading direction, scale, and reaction intensity across participants Barber & Odean (2000); O'Hara (1995). Public transaction-level records from platforms such as Polymarket make it possible to study such differences directly. Rather than forecasting event outcomes or modeling an average market participant, we use real transaction histories to model heterogeneous order-placement behavior as the combination of a shared market response and trader-specific deviations.

6.2 LLM Agents and Personalized Behavior Modeling

LLMs have been used to simulate individual behavior, social interaction, and economic decision-making Park et al. (2023); Horton (2023). Alignment and personalization methods condition model behavior on human feedback, explicit principles, group opinions, or user preferences Ouyang et al. (2022); Bai et al. (2022); Santurkar et al. (2023); Lee et al. (2024); Aher et al. (2023); Argyle et al. (2023). Although these approaches can produce differentiated outputs Bisbee et al. (2024); Santurkar et al. (2023), they typically rely on textual profiles, surveys, or preference annotations. CRBM instead derives wallet-level behavioral conditions from repeated real transactions and represents them as deviations from market consensus.

LoRA enables parameter-efficient adaptation, while mixture-of-experts and instance-wise LoRA methods support input-dependent combinations of specialized updates Hu et al. (2022); Shazeer et al. (2017); Kong et al. (2024). CRBM uses wallet profile residuals to condition this adaptation pathway. The MoE-LoRA architecture is an implementation mechanism; our focus is the past-only consensus–residual representation connecting earlier wallet histories with held-out trading behavior.

7 Conclusion

This paper presents Consensus-Guided Residual Behavior Modeling (CRBM), an interpretable framework for modeling heterogeneous trading behavior under a past-only information boundary. CRBM represents wallet-level behavior as systematic deviations from market consensus and uses these residuals to condition structured order generation. Experiments on Polymarket show improved outcome selection and joint action–outcome fidelity, while ablations support the contributions of consensus, behavioral residuals, and residual-conditioned adaptation. Multi-level analyses further indicate that CRBM preserves wallet-level behavioral differences, retains descriptive population structure, and reconstructs aggregate price paths under constrained historical replay. A zero-shot Manifold case study provides preliminary evidence of cross-platform transferability. Together, these findings support consensus–residual modeling as a practical and interpretable approach to heterogeneous agent generation. Future work will investigate richer residual representations, alternative adaptation mechanisms, and closed-loop market simulations.

AI Use Statement

Generative AI tools were used to assist with language editing, manuscript organization, LaTeX revision, and code development and debugging. All AI-assisted outputs were reviewed and revised by the authors. The authors verified the mathematical definitions, experimental protocols, implementa-

tion behavior, reported results, and cited literature, and take full responsibility for the final content of the paper and supplementary materials.

Ethics Statement

This study uses publicly observable, pseudonymous prediction-market transaction records and models behavior at the wallet level. We do not attempt to identify the natural persons associated with individual wallets. Wallet-level behavioral regularities should not be interpreted as psychological traits, because a wallet may represent multiple users, automated trading strategies, or changing decision processes. CRBM is intended for behavioral modeling and controlled simulation rather than market manipulation, personalized financial targeting, or deployable price forecasting. We acknowledge the privacy and misuse risks associated with behavioral profiling from public transaction histories and report aggregate evaluations where possible.

Reproducibility Statement. We provide anonymized source code, data-processing scripts, training and evaluation configurations, baseline implementations, and ablation protocols at `https://anonymous.4open.science/r/crbm_code-21D8`. Additional implementation and evaluation details are provided in the appendix.

References

Gati V. Aher, Rosa I. Arriaga, and Adam Tauman Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 337–371. PMLR, 2023.

Lisa P. Argyle, Ethan C. Busby, Nancy Fulda, Joshua R. Gubler, Christopher Rytting, and David Wingate. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351, 2023. doi: 10.1017/pan.2023.2.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022. doi: 10.48550/arXiv.2212.08073.

Brad M. Barber and Terrance Odean. Trading is hazardous to your wealth: The common stock investment performance of individual investors. *The Journal of Finance*, 55(2):773–806, 2000. doi: 10.1111/0022-1082.00226.

Joyce E. Berg, Forrest D. Nelson, and Thomas A. Rietz. Prediction market accuracy in the long run. *International Journal of Forecasting*, 24(2):285–300, 2008. doi: 10.1016/j.ijforecast.2008.03.007.

James Bisbee, Joshua D. Clinton, Cassy Dorff, Brenton Kenkel, and Jennifer M. Larson. Synthetic replacements for human survey data? the perils of large language models. *Political Analysis*, 32 (4):401–416, 2024. doi: 10.1017/pan.2024.5.

Yiling Chen and David M. Pennock. Designing markets for prediction. *AI Magazine*, 31(4):42–52, 2010. doi: 10.1609/aimag.v31i4.2313.

Youngbin Choi, Seunghyuk Cho, Minjong Lee, MoonJeong Park, Yesong Ko, Jungseul Ok, and Dongwoo Kim. CoPL: Collaborative preference learning for personalizing LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12875–12893, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.650.

J. Bradford De Long, Andrei Shleifer, Lawrence H. Summers, and Robert J. Waldmann. Noise trader risk in financial markets. *Journal of Political Economy*, 98(4):703–738, 1990. doi: 10.1086/261703.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783.

Lawrence R. Glosten and Paul R. Milgrom. Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics*, 14(1):71–100, 1985. doi: 10.1016/0304-405X(85)90044-3.

Robin Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1): 107–119, 2003. doi: 10.1023/A:1022058209073.

John J. Horton. Large language models as simulated economic agents: What can we learn from Homo Silicus? Working Paper 31122, National Bureau of Economic Research, Cambridge, MA, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=EMrnoPRvxe.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Xiaoyu Kong, Jiancan Wu, An Zhang, Leheng Sheng, Hui Lin, Xiang Wang, and Xiangnan He. Customizing language models with instance-wise LoRA for sequential recommendation. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2024. doi: 10.52202/079017-3593.

Albert S. Kyle. Continuous auctions and insider trading. *Econometrica*, 53(6):1315–1335, 1985. doi: 10.2307/1913210.

Seongyun Lee, Sue Hyun Park, Seungone Kim, and Minjoon Seo. Aligning to thousands of preferences via system message generalization. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2024. doi: 10.52202/079017-2347.

Maureen O'Hara. *Market Microstructure Theory*. Blackwell Publishers, Cambridge, MA, 1995.

OpenAI. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. doi: 10.48550/arXiv.2410.21276.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022.

Joon Sung Park, Joseph C. O'Brien, Carrie J. 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*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. doi: 10.1145/3586183.3606763.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 29971–30004. PMLR, 2023.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

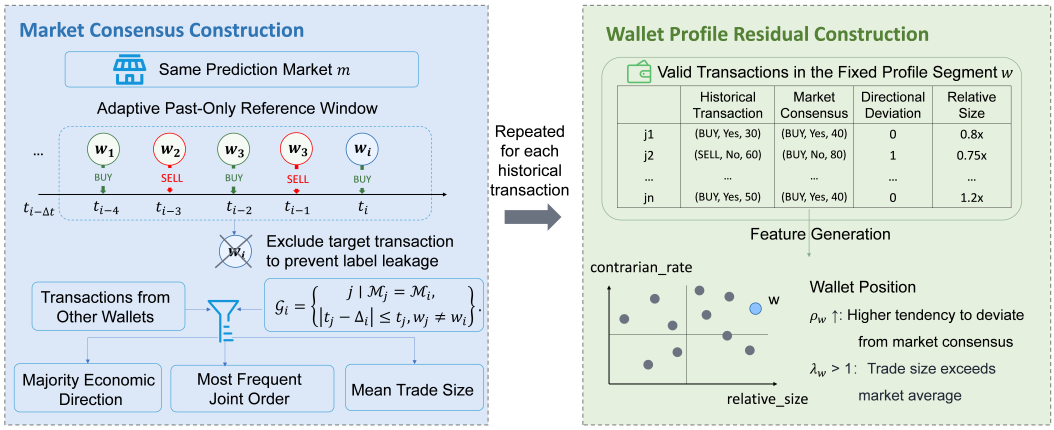


Figure 3: Detailed construction of past-only market consensus and wallet-level behavioral residuals. Consensus is constructed from strictly preceding transactions of other wallets, while the profile residual is estimated only from the wallet's fixed earlier profile segment.

Justin Wolfers and Eric Zitzewitz. Prediction markets. *Journal of Economic Perspectives*, 18(2): 107–126, 2004. doi: 10.1257/0895330041371321.

A Additional Methodological Details

A.1 Causal Data Construction

For each wallet w, we chronologically divide its transactions into an earlier profile segment $\mathcal{H}_w^{\text{profile}}$ and a later target segment $\mathcal{H}_w^{\text{heldout}}$. The profile segment contains the earliest 70% of the wallet's transactions, whereas the target segment contains the latest 30%. Transactions sharing the same timestamp are retained on the same side of the split boundary.

For target transaction i, the market-consensus reference set is restricted to valid transactions from other wallets in the same market that occur strictly before t_i:

$$\mathcal{G}_i^-(\Delta) = \{ j \mid \mathcal{M}_j = \mathcal{M}_i, \ t_i - \Delta \le t_j < t_i, \ w_j \neq w_i \}. \tag{6}$$

The lookback length is selected from

$$\{10 \text{ minutes}, 1 \text{ hour}, 6 \text{ hours}, 24 \text{ hours}\}.$$

We use the shortest window containing at least 10 valid preceding transactions. A target transaction is excluded when fewer than 10 valid reference transactions are available within 24 hours.

The target wallet is excluded from every consensus window. Furthermore, the pre-trade market state excludes information produced by the target order itself or by transactions executed at or after t_i. Here, causal refers only to the temporal information boundary and does not imply identification of causal effects, interventions, or counterfactual relationships.

We verify the temporal and wallet-level information boundaries for every retained target transaction. Specifically, the wallet profile must end before the target transaction, every consensus-reference transaction must precede the target time, and the target wallet must be absent from the consensus set:

$$\max_{j \in \mathcal{H}_{w_i}^{\text{profile}}} t_j < t_i, \qquad \max_{j \in \mathcal{G}_i^-} t_j < t_i, \qquad w_j \neq w_i \quad \forall j \in \mathcal{G}_i^-. \tag{7}$$

These conditions are checked after feature construction and before the final train–validation–test partition is produced.

Tie Handling. If the two economic directions receive the same number of votes, the corresponding target transaction is excluded because a unique prevailing direction cannot be identified. After the

consensus direction has been determined, if multiple joint action–outcome pairs within that direction have the same maximum frequency, the target transaction is likewise excluded. This rule avoids introducing an arbitrary ordering among economically equivalent tied candidates.

A.2 Valid Transactions for Residual Estimation

Not every transaction admits a valid consensus reference. For segment $S \in \{\text{profile}, \text{heldout}\}$, we define

$$\widetilde{\mathcal{H}}_w^S = \left\{ j \in \mathcal{H}_w^S \,\middle|\, \mathcal{G}_j^- \text{ exists and } |\mathcal{G}_j^-| \geq 10 \right\}. \tag{8}$$

Only transactions in $\widetilde{\mathcal{H}}_w^S$ contribute to wallet-level residuals. Transactions without a valid consensus are excluded from both the numerator and denominator.

Before consensus matching, a wallet must have at least five profile transactions and at least three target transactions. After consensus matching, we retain a wallet only when both $\widetilde{\mathcal{H}}_w^{\text{profile}}$ and $\widetilde{\mathcal{H}}_w^{\text{heldout}}$ contain at least three valid transactions. For computational efficiency, at most 300 approximately uniformly spaced transactions from each segment are used for residual estimation.

The profile-level directional residual is

$$\rho_w^{\text{profile}} = \frac{1}{\left|\widetilde{\mathcal{H}}_w^{\text{profile}}\right|} \sum_{j \in \widetilde{\mathcal{H}}_w^{\text{profile}}} \mathbb{I}[d_j \neq d_j^c]. \tag{9}$$

The profile-level scale residual is

$$\lambda_w^{\text{profile}} = \frac{\dfrac{1}{\left|\widetilde{\mathcal{H}}_w^{\text{profile}}\right|} \displaystyle\sum_{j \in \widetilde{\mathcal{H}}_w^{\text{profile}}} s_j}{\dfrac{1}{\left|\widetilde{\mathcal{H}}_w^{\text{profile}}\right|} \displaystyle\sum_{j \in \widetilde{\mathcal{H}}_w^{\text{profile}}} \bar{s}_j^c + \epsilon}. \tag{10}$$

The same definitions are applied to $\widetilde{\mathcal{H}}_w^{\text{heldout}}$ to obtain the evaluation-only held-out residual.

A.3 Residual-Conditioned Expert Architecture

CRBM uses the profile residual through two complementary pathways. In the semantic-conditioning pathway, the pre-trade market state x_i, market consensus c_i, and profile residual $r_{w_i}^{\text{profile}}$ are serialized into the structured prompt

$$P_i = \text{Prompt}\left(x_i, c_i, r_{w_i}^{\text{profile}}\right). \tag{11}$$

In the parameter-adaptation pathway, the same profile residual is passed to a gating network that produces sample-specific LoRA-expert weights:

$$\boldsymbol{\pi}_i = \text{softmax}\left(g_\phi\left(r_{w_i}^{\text{profile}}\right)\right), \qquad \sum_{k=1}^{K} \pi_{ik} = 1. \tag{12}$$

For frozen foundation-model weight W, the adapted output is

$$h' = Wh + \sum_{k=1}^{K} \pi_{ik} \frac{\alpha_k}{R_k} B_k A_k h. \tag{13}$$

Here, A_k and B_k are the trainable low-rank matrices of expert k, while R_k and α_k denote the corresponding rank and scaling coefficient.

Expert and Router Configuration. The main configuration uses $K = 2$ LoRA experts with a total rank budget of 16, assigning rank $R_k = 8$ to each expert. The foundation-model parameters remain frozen, while the LoRA matrices and router parameters are optimized jointly using the supervised objective in Equation 5. LoRA updates are applied to the `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj` transformations. Complete optimization settings are reported in Appendix B.2.

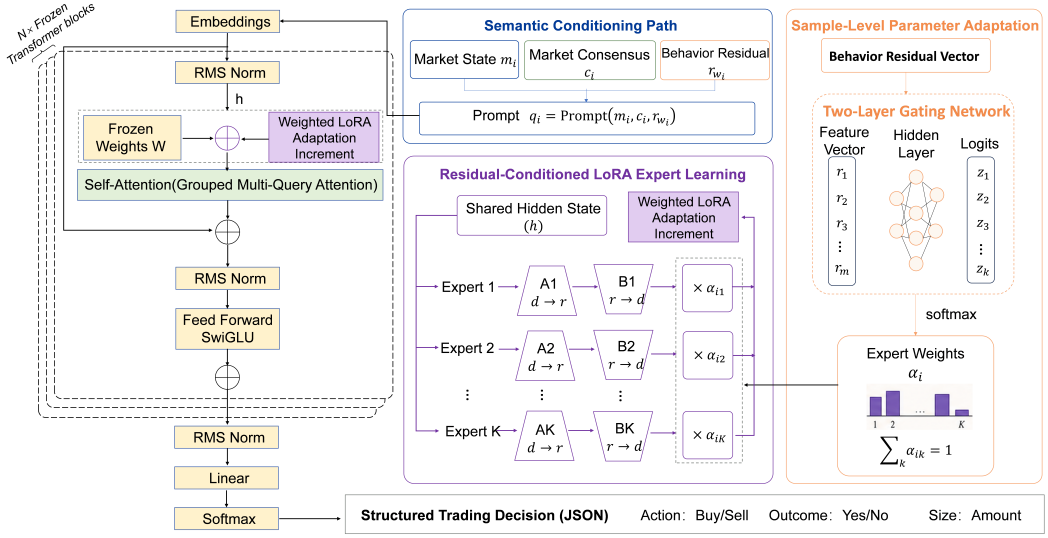


Figure 4: Residual-conditioned expert architecture. The wallet profile residual conditions both the structured prompt and the LoRA-expert routing pathway. The held-out residual is not included in either pathway and is used only for evaluation.

Split	Wallets	Target Samples
Training	89	10,806
Validation	13	908
Test	26	2,316
Total	128	14,030

Table 4: Wallet-disjoint dataset splits after past-only feature construction and validity filtering.

B Dataset and Experimental Details

B.1 Dataset Statistics and Wallet-Disjoint Splits

After past-only feature construction and wallet-validity filtering, the dataset contains 128 wallets. We randomly partition wallets, rather than individual transactions, into mutually exclusive training, validation, and test sets using a $70\%/10\%/20\%$ split. Consequently, no wallet appears in more than one data split. Table 4 reports the resulting numbers of wallets and held-out target samples.

The training set may be resampled to address class imbalance. Resampling is performed only after the wallet-disjoint split has been fixed and is never applied to the validation or test sets. The validation set is used for checkpoint selection and transaction-level model comparison, whereas the test set is used for the wallet-level and multi-level analyses reported in Section 4.3.

B.2 Training Configuration

LoRA experts are inserted into the `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj` linear transformations. During evaluation, generated text is parsed into action, outcome, and size fields. Parse Rate is computed before the remaining metrics, while action, outcome, and size metrics are computed from valid parsed outputs.

B.3 Baseline Settings and Input Fairness

All baselines use the same wallet-disjoint splits. Rule-based baselines are deterministic. LightGBM uses 300 trees for the action and outcome classifiers and 400 trees for the size regressor, with learning rate 0.03, 31 leaves, feature and row subsampling rates of 0.9, and random seed 42. The MLP uses

Item	Setting
Foundation model	Llama-3-8B-Instruct
Number of experts K	2
Total LoRA rank	16
Rank per expert	8
LoRA scaling α	32
LoRA dropout	0.05
Residual dimension	2
LoRA learning rate	1×10^{-4}
Gating learning rate	1×10^{-3}
Epochs	1
Per-device batch size	1
Gradient accumulation	16
Effective batch size	16
Maximum sequence length	512
Warm-up ratio	0.05
Weight decay	0.01
Learning-rate schedule	Cosine
Precision	bfloat16
Gradient clipping	1.0
Random seed	42
Checkpoint interval	200 steps
Selected checkpoint	Step 800

Table 5: Training configuration used for the causal v16 CRBM experiment.

two hidden layers of width 128, dropout 0.15, batch size 128, learning rate 10^{-3}, weight decay 10^{-4}, and 20 epochs.

For language-model baselines, the same structured output schema is used whenever supported. P-Soups uses rank 16, LoRA scaling 32, dropout 0.05, learning rate 2×10^{-4}, and maximum sequence length 512. CoPL uses rank 8, four experts, bfloat16 precision, and maximum sequence length 1024.

B.4 Evaluation Metrics and Statistical Reporting

Transaction-Level Metrics. We evaluate the fidelity of generated structured orders along the action, outcome, joint-decision, size, and format-validity dimensions. Action Accuracy and Action F1 measure BUY/SELL prediction, while Outcome Accuracy and Outcome F1 measure YES/NO selection. Strict Accuracy is the proportion of target transactions for which both the action and outcome are predicted correctly.

Trade-size fidelity is evaluated using Size Log-MAE:

$$\mathrm{LogMAE} = \frac{1}{N} \sum_{i=1}^{N} \left| \log(1 + \hat{s}_i) - \log(1 + s_i) \right|, \tag{14}$$

where $\hat{s}_i$ and s_i denote the generated and observed trade sizes. We additionally report Parse Rate, defined as the proportion of generated outputs that can be converted into the required action, outcome, and size fields. Parse Rate is treated as a format-validity diagnostic rather than a primary measure of behavioral fidelity.

Wallet-Level Metrics. For each test wallet, we compute $r_w^{\mathrm{generated}}$ from its generated target-segment transactions and r_w^{heldout} from the corresponding real held-out transactions. Held-out preservation is evaluated separately for the directional and scale components using rank correlation and absolute error across test wallets. This is the primary wallet-level evaluation because r_w^{heldout} is never supplied to the model.

We additionally compare $r_w^{\mathrm{generated}}$ with r_w^{profile}. This comparison measures residual adherence, namely whether generated behavior follows the supplied conditioning signal. Because the profile residual is provided as model input, residual adherence is reported as a conditioning diagnostic and is not interpreted as recovery of unseen wallet behavior.

K	Setting	Act. F1	Out. F1	Strict	Size Log-MAE
1	Single LoRA	0.4665	0.5399	0.4319	1.2379
2	MoE-LoRA	**0.5477**	**0.5521**	**0.4707**	**1.1140**
4	MoE-LoRA	0.5365	0.5364	0.4448	1.3581

Table 6: Sensitivity to the number of LoRA experts.

Method	Spearman ↑	MAE ↓
Llama-3 (No FT)	0.8454	0.4138
MLP	0.7835	0.2778
LightGBM	0.6561	0.2644
P-Soups	0.4679	0.2061
Historical Profile	0.2684	0.2653
Majority Following	—	0.4230
CRBM	**0.9208**	**0.1299**

Table 7: Residual-adherence results under the causal wallet-disjoint v16 evaluation protocol. Spearman correlation and MAE compare generated contrarian tendencies with the profile residual supplied as model input. These results measure adherence to the conditioning signal rather than preservation of unseen held-out behavior.

Statistical Reporting. Due to computational cost, each learned model is trained once using a fixed random seed of 42. The reported results are therefore point estimates and do not characterize variation across independent training runs. Model selection and hyperparameter configuration are performed exclusively on the validation set.

B.5 Computing Infrastructure

The experiments were conducted on an Ubuntu 22.04 LTS server with two NVIDIA L40 GPUs and one NVIDIA A100 80GB PCIe GPU. The server contained two Intel Xeon Silver 4314 processors and 251 GiB of system memory. The software environment used Python 3.9.23, PyTorch 2.5.1 with CUDA 12.1 and cuDNN 9.1, Transformers 4.57.6, PEFT 0.17.1, scikit-learn 1.6.1, and LightGBM 4.6.0.

C Additional Evaluation Details

C.1 Expert-Count Sensitivity

We evaluate whether increasing the number of residual-conditioned LoRA experts improves structured behavior generation. Table 6 compares a single LoRA adapter with two- and four-expert configurations under the same total evaluation protocol.

Moving from a single LoRA adapter to two residual-conditioned experts improves all four metrics. Increasing the number of experts from $K = 2$ to $K = 4$ reduces Action F1, Outcome F1, and Strict Accuracy, while also increasing Size Log-MAE. Thus, $K = 2$ provides the best overall performance for the two-dimensional residual representation used in this study.

C.2 Residual-Adherence Analysis

Residual adherence measures whether generated behavior reflects the profile residual supplied as a conditioning signal. For each test wallet w, we compare the residual recovered from generated target-segment transactions, $r_w^{\text{generated}}$, with the input profile residual r_w^{profile}. This comparison is distinct from held-out preservation, which compares $r_w^{\text{generated}}$ with the evaluation-only residual r_w^{heldout}.

CRBM obtains the highest rank correlation and the lowest absolute error among the evaluated methods, indicating that its generated behavior responds strongly to the supplied profile residual. This

Descriptive Group	Share	Contrarian Rate	Relative Size
Small-Scale Contrarian	38.10%	0.5280	0.1927
Small-Scale Consensus	37.14%	0.0398	0.2682
Large-Scale Contrarian	17.14%	0.5104	1.4061
Large-Scale Consensus	7.62%	0.1409	3.9987

Table 8: Descriptive groups identified from CRBM-generated trading trajectories. The groups are post-hoc interpretations rather than ground-truth trader types.

History	N	Outcome Acc. ↑	Outcome F1 ↑	Size Log-MAE ↓
Overall	949	0.5101	0.5278	1.2133
Long history	712	0.5322	0.5385	1.1980
Medium history	164	0.5076	0.5045	1.2375
Limited history	73	0.4805	0.5125	1.3079

Table 9: Zero-shot cross-platform evaluation of the Polymarket-trained CRBM on Manifold, reported overall and by the amount of available user history.

result should be interpreted only as conditioning adherence. It does not establish that the profile residual predicts the wallet's subsequently observed held-out behavior.

C.3 Population-Level Structure

Generated wallet behaviors are represented using contrarian rate and relative size. Because relative size is right-skewed, the analysis applied $\log(1 + \text{relative_size})$ before standardization. K-means clustering used $K = 4$, random seed 42, and 30 initializations. Cluster names were assigned post hoc as descriptive interpretations rather than supervised trader-type labels.

C.4 Manifold Case-Study Details

The Manifold case study initially collects 100 candidate markets and retains 54 valid binary markets. The resulting data contain $1{,}474$ candidate target trades, 349 users, and $114{,}272$ historical transactions. A target sample requires at least three preceding transactions from the target user and at least three preceding transactions in the target market. After filtering, 949 target samples remain.

All histories are truncated strictly before the target timestamp. Manifold transactions are primarily represented as purchases of YES or NO shares and are therefore mapped to BUY YES or BUY NO. Because this representation does not preserve the BUY/SELL distinction used by Polymarket, we report Outcome Accuracy, Outcome F1, and Size Log-MAE, but not action or strict joint-order accuracy.

To examine sensitivity to the amount of available user history, we define three groups:

- **Long history**: at least 100 preceding transactions;
- **Medium history**: 20–99 preceding transactions;
- **Limited history**: fewer than 20 preceding transactions while satisfying the minimum-history requirement.

Performance varies with the amount of available user history. Outcome F1 decreases from 0.5385 for users with long histories to 0.5125 for users with limited histories, while Size Log-MAE increases from 1.1980 to 1.3079. This pattern is consistent with CRBM's reliance on historical transactions to estimate stable wallet-level residuals, although it does not by itself establish that history length causes the performance difference.

This case study should be interpreted as preliminary evidence of cross-platform transferability rather than comprehensive domain generalization. It includes no Manifold-specific fine-tuning, considers only one target platform, and does not compare against baselines retrained on Manifold.

D PROMPT AND OUTPUT FORMATS

The model receives a structured prompt containing the pre-trade market state, past-only market consensus, and wallet profile residual. The required output is a JSON object:

```
{
  "action": "BUY or SELL",
  "outcome": "Yes or No",
  "size": positive number
}
```

Outputs are parsed before metric computation. Parse Rate measures the proportion of outputs conforming to the required schema. For parsed outputs, action and outcome metrics are computed from the corresponding categorical fields, while trade-size error is computed from the positive numerical size field.

E DATA AND CODE AVAILABILITY

The supplementary materials include data-processing scripts, model-training code, baseline implementations, evaluation scripts, and historical-replay code. They also provide the wallet-disjoint split identifiers, prompt and output schemas, configuration files, and scripts used to reproduce the reported tables. Public wallet identifiers are pseudonymized while preserving consistent identifiers required for wallet-level splitting and evaluation.