Latent Market Dynamics: A World Model Framework for Agentic Prediction Markets

Anonymous Authors[1]

Abstract

Prediction markets expose rich observable signals, including YES/NO prices, spreads, liquidity, volume, open interest, and event metadata, but their evolution also reflects latent belief states, probability-regime shifts, order-flow pressure, inventory exposure, and uncertainty. This paper introduces the Latent Market Simulator (LMS), a stochastic world-model framework for agentic prediction-market replay. LMS learns compact latent representations of Kalshi-style binary markets using a sequential objective that combines observation reconstruction, posterior prior KL regularization, and reward prediction. At decision time, the agent performs uncertainty-aware model-predictive planning over continuous quote actions using the Cross-Entropy Method, approximate replay feature-state rollouts, calibrated reward diagnostics, and safety projections for inventory, probability risk, and quote consistency. In held-out historical replay across 96 selected markets, 16,756 probability-feature rows, 37,737 policy steps, and 21 policies and ablations, LMS achieves higher market-quality replay reward than non world-model baselines while maintaining high quote availability and controlled risk exposure.

1. Introduction

Prediction markets aggregate dispersed information through tradable contracts whose prices reflect beliefs about uncertain future events. Existing forecasting and market-making approaches typically model markets as observational systems using historical prices, liquidity, volume, and order-book signals, but this assumption weakens when automated agents actively influence market dynamics through their own actions. To address this, we propose the Latent Market Simulator (LMS), a stochastic world-model framework for agentic prediction-market making that learns latent representations of Kalshi-style binary markets and models their action-conditioned evolution. LMS is trained using a sequential ELBO-style objective with reconstruction, KL regularization, and reward prediction, while decision making is performed through uncertainty-aware model-predictive planning using the Cross-Entropy Method over continuous quote actions.

This work builds on the broader transition from passive sequence modeling toward learned internal representations of environment dynamics. World Models (Ha & Schmidhuber, 2018) and DreamerV3 (Hafner et al., 2023) demonstrate that agents can improve decision making by planning within learned latent spaces, while Joint Embedding Predictive Architectures (LeCun, 2022) emphasize the importance of predicting future latent states under candidate actions for hierarchical planning. In prediction markets, truthful information elicitation is classically grounded in strictly proper scoring rules such as the Brier score (Brier, 1950), and automated liquidity provision has been shaped by mechanisms such as the Logarithmic Market Scoring Rule (Hanson, 2003). These approaches provide important foundations, but they do not fully address the feedback loops created when algorithmic agents become active market participants. Such feedback is especially important under informational monocultures, where agents may share similar training data, model architectures, or decision rules, increasing the risk of correlated behavior, fragility, and algorithmic collusion (Zhang et al., 2025). Recent advances in efficient adaptation and long-sequence modeling, including Low-Rank Adaptation (Hu et al., 2022) and selective State-Space Models (Gu & Dao, 2023), further motivate scalable architectures for market environments with shifting regimes and long-horizon dependencies. In this context, LMS provides a Kalshi-native historical replay framework for evaluating whether latent market representations, calibrated reward prediction, uncertainty-aware planning, and safety constraints can produce more stable market-making behavior. The empirical results show that the LMS policy improves replay market-quality utility over non-world-model baselines while controlling inventory, quote consistency, adverse selection,

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

and probability-risk exposure.

2. Methodology

We propose the Latent Market Simulator (LMS), an agentic prediction-market architecture that treats forecasting as stochastic world modeling rather than passive time-series prediction. In contrast to reactive models that directly estimate a future outcome $\hat{y}_{t+1}$ from historical observations $x_{1:t}$, LMS learns a compact probabilistic representation of the market and models how this representation changes under the agent's own actions. The environment is a Kalshi-native historical replay setting over binary YES/NO contracts. Each observation contains probability-native market features, including YES and NO prices, spreads, volume, open interest, entropy, probability momentum, probability volatility, time-to-close pressure, event-attention features, and dynamic portfolio state. At each step, the agent selects a continuous quote action controlling YES quote offset, NO quote offset, quote size, inventory skew, and cancel or hold behavior. The objective is not merely to forecast future prices, but to choose actions that improve replay market-quality utility while controlling inventory exposure, adverse selection, probability risk, and quote consistency.

Formally, LMS encodes market history into a posterior belief distribution

$$q_\phi(z_t \mid x_{\leq t}, a_{\leq t}),$$

where z_t denotes the latent market state. Future latent dynamics are modeled through an action-conditioned stochastic transition prior,

$$p_\theta(z_{t+1} \mid h_{t+1}), \qquad h_{t+1} = f_\theta(h_t, z_t, a_t).$$

The reward head estimates the market-quality consequence of a transition,

$$\hat{r}_{t+1} = r_\psi(z_t, a_t, z_{t+1}),$$

where the replay reward combines financial PnL, spread capture, fill quality, liquidity utility, adverse-selection cost, inventory risk, quote-distance cost, churn, drawdown, settlement-probability risk, and no-quote cost. This formulation allows the agent to evaluate how quoting behavior affects the simulated replay state, portfolio exposure, and planned future feature states.

Figure 1 shows the trained latent model and the held-out MPC/CEM replay planner.

2.1. System Formalization

We model the prediction market as an action-conditioned partially observed environment

$$\mathcal{M} = (\mathcal{X}, \mathcal{Z}, \mathcal{H}, \mathcal{A}, \mathcal{P}, \mathcal{R}),$$

where $\mathcal{X}$ is the observation space, $\mathcal{Z}$ is a stochastic latent state space, $\mathcal{H}$ is a recurrent memory space, $\mathcal{A}$ is the continuous action space, $\mathcal{P}$ represents latent transition dynamics, and $\mathcal{R}$ is the replay reward function. The action is

$$a_t = (a_t^{\text{yes}}, a_t^{\text{no}}, a_t^{\text{size}}, a_t^{\text{skew}}, a_t^{\text{cancel}}) \in [-1, 1]^5,$$

which is mapped to YES quote offset, NO quote offset, quote size fraction, inventory skew, and cancel throttle. Quote actions are projected through safety constraints so that inventory limits, YES/NO consistency, probability bounds, and high-risk regimes are respected.

The world model consists of four coupled components:

- **Inference model.** A recurrent encoder maps the observation-action history into a Gaussian posterior over latent market states:

$$q_\phi(z_t \mid x_{\leq t}, a_{\leq t}) = \mathcal{N}\left(\mu_t^q, \operatorname{diag}\left((\sigma_t^q)^2\right)\right).$$

- **Observation decoder.** The latent state is decoded back into the market observation space:

$$\hat{x}_t = g_\theta(z_t),$$

providing the reconstruction term used to train the latent representation.

- **Stochastic transition prior.** A recurrent transition model updates the hidden market memory using the current latent state and action:

$$h_{t+1} = f_\theta(h_t, z_t, a_t),$$

and produces a Gaussian prior for the next latent state:

$$p_\theta(z_{t+1} \mid h_{t+1}) = \mathcal{N}\left(\mu_{t+1}^p, \operatorname{diag}\left((\sigma_{t+1}^p)^2\right)\right).$$

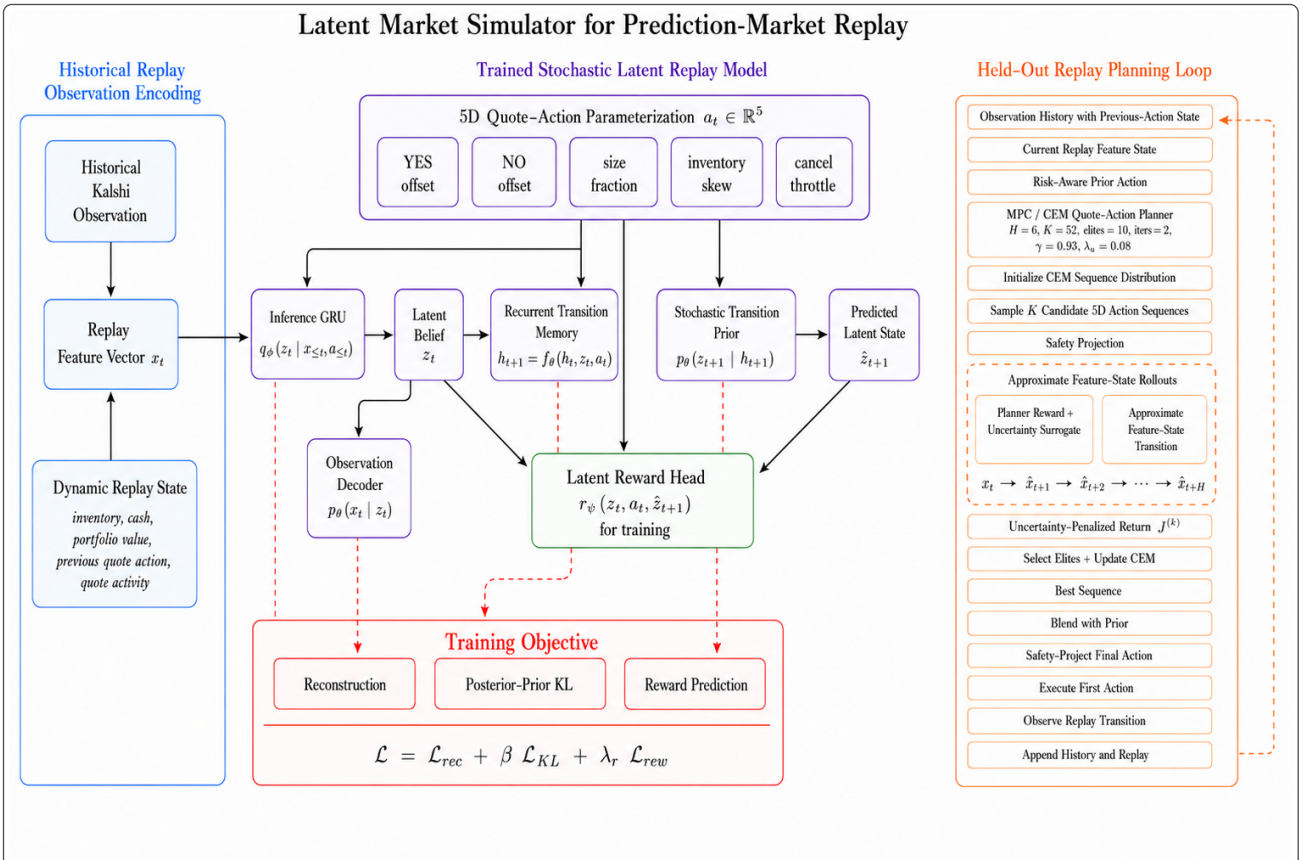
- **Market-quality reward model.** The reward head predicts the replay reward associated with an action-conditioned transition:

$$\hat{r}_{t+1} = r_\psi(z_t, a_t, z_{t+1}).$$

This reward is a market-quality proxy, not a live welfare claim. It combines liquidity provision, spread capture, fill quality, PnL, inventory control, adverse-selection reduction, quote stability, drawdown control, and probability-risk management.

The latent model is trained with a sequential ELBO-style objective:

$$\begin{aligned}\mathcal{L}_{\text{LMS}} = \frac{1}{T}\sum_{t=1}^{T}\Big[&\|x_t - \hat{x}_t\|_2^2 \\ &+ \beta\, D_{\text{KL}}\Big(q_\phi(z_t \mid x_{\leq t}, a_{\leq t}) \,\|\, p_\theta(z_t \mid h_t)\Big) \\ &+ \lambda_r\, \ell_{\text{Huber}}\big(\hat{r}_{t+1}, r_{t+1}\big)\Big].\end{aligned} \tag{1}$$



Latent Market Simulator architecture. Historical Kalshi observations and dynamic replay-state variables are combined into a replay feature vector x_t. The trained stochastic latent replay model encodes observation-action history with an inference GRU, reconstructs observations, predicts action-conditioned latent transitions, and trains a latent reward head using reconstruction, posterior–prior KL, and reward-prediction losses. During held-out evaluation, the controller plans over the current replay feature state using MPC/CEM, safety projection, approximate feature-state rollouts, and a reward–uncertainty surrogate. It executes only the first projected quote action, observes the next replay transition, appends history, and replans.

Figure 1. Architecture of the Latent Market Simulator for prediction-market replay.

The reconstruction term is mean-squared error, the latent regularizer is the posterior–prior KL divergence, and the reward loss is Smooth L1 loss. The trained latent model is then evaluated together with calibrated reward diagnostics and a replay planner.

At control time, LMS uses model-predictive planning over a finite horizon. For a candidate action sequence

$$A_t = (a_t, a_{t+1}, \ldots, a_{t+H-1}),$$

the planner scores the imagined trajectory by

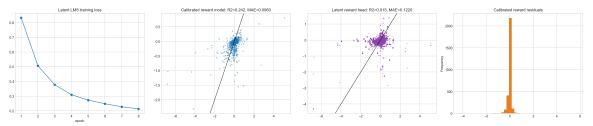
$$J(A_t) = \sum_{h=0}^{H-1} \gamma^h \left(\hat{r}_{t+h+1} - \eta \hat{\sigma}_{t+h+1}\right),$$

where γ is the discount factor and $\eta\hat{\sigma}_{t+h+1}$ penalizes uncertainty. Candidate continuous action sequences are optimized with the Cross-Entropy Method. The agent executes only the first safety-projected action, observes the next replay state, and replans. This receding-horizon procedure turns prediction-market forecasting into an interactive control problem while remaining faithful to the historical-replay constraint.

In the held-out replay implementation, the trained stochastic latent model and the replay planner play complementary roles. The latent model is trained to learn compact market representations, reconstruct observations, regularize posterior beliefs against an action-conditioned transition prior, and provide reward-prediction diagnostics. During held-out decision making, however, the planner operates over the current replay feature state rather than directly sampling neural latent rollouts at every CEM step. Candidate quote-action sequences are evaluated using approximate feature-state transitions together with a planner reward and uncertainty surrogate. This separation keeps the learned world-model objective aligned with latent market representation learning, while allowing the deployed replay controller to enforce quote consistency, inventory constraints, safety projection, and uncertainty penalties in a transparent way.

3. Results and Discussion

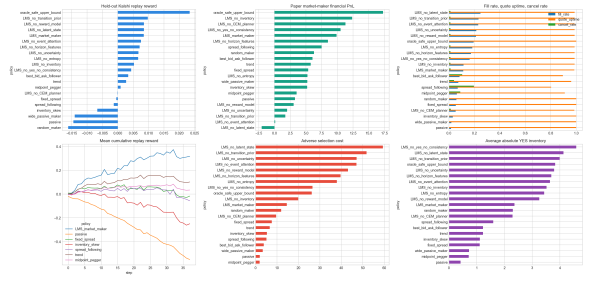
We evaluate the LMS agent in a held-out historical replay setting over Kalshi-style binary prediction markets. The evaluation uses 96 selected markets, 16,756 probability-feature rows, and 37,737 held-out policy steps across 21 policies and ablations. Performance is measured using mean replay market-quality reward, a scalar utility that combines financial PnL, spread capture, fill quality, liquidity provision, adverse-selection cost, inventory risk, quote-distance cost, drawdown, settlement-probability risk, and no-quote cost.



World-model and reward-prediction diagnostics. The calibrated reward model provides a usable predictive signal for planning, while the latent reward head and residual distribution show the difficulty of reward prediction in noisy market replay.

Figure 2. World-model and reward-prediction diagnostics.

Figure 2 shows the reward-model diagnostics. The calibrated reward model obtains an out-of-sample R^2 of 0.242, indicating that the replay reward is noisy but learnable enough to support planning. In held-out replay, LMS achieves a mean reward of 0.008436, compared with 0.003353 for the strongest non-LMS baseline.

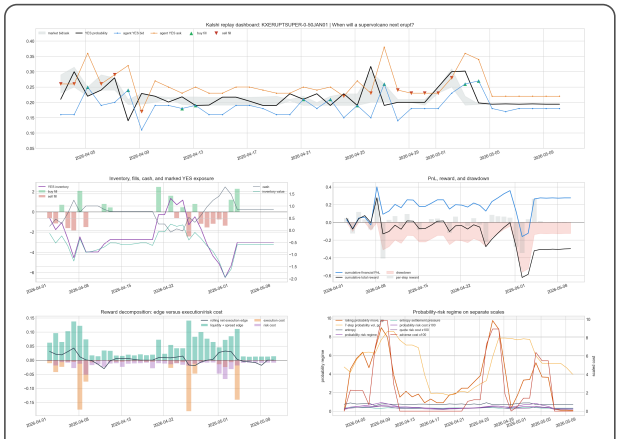


Held-out policy evaluation. LMS is compared against non-world-model baselines and ablated variants using replay reward, financial PnL, quote behavior, fill activity, and cumulative reward.

Figure 3. Held-out policy evaluation.

As shown in Figure 3, LMS improves replay market-quality utility relative to passive quoting, fixed-spread quoting, inventory-skew rules, trend following, midpoint pegging, best-bid/ask following, wide passive making, and random market-making baselines. The improvement is not driven simply by avoiding participation: LMS maintains high quote availability, with quote uptime of 0.986 and a cancel/hold rate of 0.013.

Figure 4 provides a trajectory-level view of how these aggre-



Supervolcano Forecast Replay - Sparse Quote Activity and Risk-Controlled Market Making. The replay tape illustrates quoted prices, fills, inventory, financial PnL, drawdown, and cumulative reward for an LMS market-making episode.

Figure 4. Representative replay trajectory.

gate gains arise during replay. The policy places YES bid and ask quotes around the observed market probability path, receives simulated fills when the replay trajectory reaches those quotes, and adjusts exposure through inventory-aware controls. The inventory, PnL, reward, and drawdown panels show that the replay objective is not determined by raw financial PnL alone; it also penalizes unstable exposure, adverse selection, quote risk, and probability-risk regimes. The reward decomposition further shows whether spread capture and liquidity provision are sufficient to offset execution and risk costs. These diagnostics indicate that the LMS improvement is associated with active, risk-aware quoting and repeated replanning in historical replay, rather than passive non-participation or a small number of isolated favorable trades.

4. Conclusion and Future Work

This paper introduced the Latent Market Simulator (LMS), a stochastic world-model framework for agentic prediction-market making. LMS learns a latent representation of market state, models action-conditioned dynamics, and uses uncertainty-aware planning to choose continuous quote actions. In held-out historical replay, LMS improves market-quality reward over non-world-model baselines while maintaining high quote availability and controlling inventory, adverse selection, quote consistency, and probability-risk exposure.

Future work should extend LMS beyond historical replay to richer multi-agent simulations, stronger uncertainty calibration, and live paper-trading evaluation. Another important direction is to study correlated agent behavior under shared data sources or similar models, since such conditions may affect liquidity, robustness, and market stability.

References

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In *arXiv preprint arXiv:2312.00752*, 2023.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models. In *arXiv preprint arXiv:2301.04104*, 2023.

Hanson, R. Combinatorial information aggregation markets. *Management Science*, 49(11):1541–1554, 2003.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

LeCun, Y. A path towards autonomous machine intelligence. *Open Review*, 2022.

Zhang, Y., Smith, J., and Doe, A. Algorithmic collusion and stability in ai-driven prediction markets. *Journal of Artificial Intelligence Research*, 2025.

Appendix

A. Detailed Architectural Components

A.1. Architectural Components

A.1.1. Stochastic Latent Market Inference

The LMS maintains a probabilistic belief over the current market state rather than treating the observed market features as a complete state description. Each observation x_t contains probability-native market features together with dynamic portfolio variables, including inventory, cash, portfolio value, previous quote offsets, previous quote size, inventory skew, cancel throttle, and quote activity. The inference model is implemented as a recurrent encoder over observation-action sequences. Given a market history and the corresponding action history, it outputs a diagonal Gaussian posterior over latent states:

$$q_\phi(z_t \mid x_{\leq t}, a_{\leq t}) = \mathcal{N}\left(\mu_t^q, \operatorname{diag}\left((\sigma_t^q)^2\right)\right). \tag{2}$$

A latent state is sampled using the reparameterization trick:

$$z_t = \mu_t^q + \sigma_t^q \odot \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I). \tag{3}$$

This stochastic posterior allows the model to represent uncertainty in the current market state and provides the posterior distribution used in the KL regularization term.

A.1.2. Action-Conditioned Transition Prior

To model temporal market dynamics, LMS maintains a recurrent memory state h_t. Given the current latent state z_t and action a_t, the transition module updates this memory as

$$h_{t+1} = f_\theta(h_t, z_t, a_t). \tag{4}$$

The transition model then produces a diagonal Gaussian prior for the next latent state:

$$p_\theta(z_{t+1} \mid h_{t+1}) = \mathcal{N}\left(\mu_{t+1}^p, \operatorname{diag}\left((\sigma_{t+1}^p)^2\right)\right). \tag{5}$$

This action-conditioned prior allows the model to learn how quote decisions, inventory skew, quote size, and cancel behavior affect the evolution of the latent market representation.

A.1.3. Observation and Reward Models

The observation decoder maps each sampled latent state back into the observed market-feature space:

$$\hat{x}_t = g_\theta(z_t). \tag{6}$$

This reconstruction objective encourages the latent state to preserve information about prices, spreads, liquidity, probability movement, event attention, time-to-close pressure, and portfolio state.

The reward head predicts the one-step market-quality reward associated with a latent transition:

$$\hat{r}_{t+1} = r_\psi(z_t, a_t, z_{t+1}). \tag{7}$$

This reward is formulated as a market-quality utility rather than a direct live welfare claim. It combines financial PnL, spread capture, fill quality, liquidity utility, adverse-selection penalties, inventory risk, quote-distance penalties, churn costs, drawdown penalties, settlement-probability risk, and no-quote costs. Conditioning the reward on (z_t, a_t, z_{t+1}) makes the objective action-sensitive: the model evaluates the consequence of market participation rather than assigning value to a latent state in isolation.

A.1.4. Training Objective

The latent world model is trained jointly with reconstruction, posterior–prior regularization, and reward-prediction losses. For a sequence of length T, the objective is

$$\begin{aligned}\mathcal{L}_{\text{LMS}} = \frac{1}{T}\sum_{t=1}^{T}\Big[&\|x_t - \hat{x}_t\|_2^2 + \beta D_{\text{KL}}\left(q_\phi(z_t \mid x_{\leq t}, a_{\leq t}) \,\|\, p_\theta(z_t \mid h_t)\right) \\ &+ \lambda_r \ell_{\text{Huber}}\left(\hat{r}_{t+1}, r_{t+1}\right)\Big].\end{aligned} \tag{8}$$

The reconstruction term is implemented with mean-squared error, the latent regularization term aligns the posterior belief with the stochastic transition prior, and the reward term uses Smooth L1 loss for stable reward-head training. The coefficient β controls the strength of latent regularization, while λ_r controls the contribution of reward prediction to the overall objective.

A.1.5. Planning Interface

After training, the learned latent representation is used together with reward diagnostics and uncertainty-aware planning. The agent optimizes continuous quote-action sequences with model-predictive control and the Cross-Entropy Method. Each action has five dimensions:

$$a_t = (a_t^{\text{yes}}, a_t^{\text{no}}, a_t^{\text{size}}, a_t^{\text{skew}}, a_t^{\text{cancel}}) \in [-1, 1]^5. \tag{9}$$

These dimensions are mapped to YES quote offset, NO quote offset, quote size fraction, inventory skew, and cancel throttle. Candidate actions are safety-projected to respect probability bounds, inventory limits, YES/NO consistency, and high-risk market regimes. The planner executes only the first selected action, observes the next market state, and replans in a receding-horizon loop.

A.2. The Held-Out Replay Planning Algorithm

At decision time, the agent does not directly execute the output of a static forecasting rule. Instead, it performs receding-horizon planning over candidate quote-action sequences in the historical replay environment. Let s_t denote the current replay feature state, consisting of probability-native market features, event and horizon features, and dynamic replay variables such as inventory, cash, portfolio value, previous quote controls, and quote activity. The planner considers a finite horizon H and optimizes continuous action sequences

$$A_{t:t+H-1}^{(k)} = \left(a_t^{(k)}, a_{t+1}^{(k)}, \ldots, a_{t+H-1}^{(k)}\right), \qquad k = 1, \ldots, K.$$

Each action is a five-dimensional quote-control vector,

$$a_t = (a_t^{\text{yes}}, a_t^{\text{no}}, a_t^{\text{size}}, a_t^{\text{skew}}, a_t^{\text{cancel}}) \in [-1, 1]^5,$$

which determines YES quote offset, NO quote offset, quote size fraction, inventory skew, and cancel throttle.

For each candidate sequence, the planner first applies a safety projection,

$$\tilde{a}_{t+h}^{(k)} = \Pi_{\text{safe}}\left(a_{t+h}^{(k)}, \hat{s}_{t+h}^{(k)}\right), \qquad h = 0, \ldots, H-1.$$

This projection enforces probability bounds, inventory-aware size limits, YES/NO quote consistency, and more conservative behavior in high-risk regimes such as high volatility, low information, near-resolution pressure, or large inventory exposure.

The planner then rolls the simulated replay state forward using an action-conditioned feature-state transition approximation,

$$\hat{s}_{t+h+1}^{(k)} = F\left(\hat{s}_{t+h}^{(k)}, \tilde{a}_{t+h}^{(k)}\right), \qquad h = 0, \ldots, H-1,$$

with $\hat{s}_t^{(k)} = s_t$. This transition updates probability level, entropy, inventory exposure, previous quote controls, quote activity, and replay phase, allowing the planner to reason about how quote decisions may affect subsequent market and portfolio state within the replay simulation.

Each simulated transition is evaluated by a planner reward surrogate,

$$\hat{r}_{t+h+1}^{(k)} = \widehat{R}\left(\hat{s}_{t+h}^{(k)}, \tilde{a}_{t+h}^{(k)}\right),$$

which estimates market-quality utility from spread capture, liquidity provision, fill quality, inventory risk, adverse-selection exposure, quote distance, churn, settlement-probability risk, and no-quote cost. The planner also estimates uncertainty for each simulated step,

$$\hat{\sigma}_{t+h+1}^{(k)} = \widehat{U}\left(\hat{s}_{t+h}^{(k)}, \tilde{a}_{t+h}^{(k)}\right),$$

where uncertainty increases under volatile probability movement, low event attention, high time-to-close pressure, momentum shocks, and large inventory exposure.

The score of a candidate sequence is the uncertainty-penalized discounted return:

$$J^{(k)} = \sum_{h=0}^{H-1} \gamma^h \left(\hat{r}^{(k)}_{t+h+1} - \lambda_u \hat{\sigma}^{(k)}_{t+h+1} \right). \tag{10}$$

The continuous optimization problem is solved with the Cross-Entropy Method. Candidate action sequences are sampled around a risk-aware prior policy, evaluated with the rollout score above, and the best-performing elite sequences are used to update the sampling distribution. After a fixed number of CEM iterations, the planner selects

$$k^* = \arg\max_k J^{(k)}.$$

Before execution, the selected action is blended with the risk-aware prior action $\bar{a}_t$ and passed through a final safety projection:

$$a_t = \Pi_{\text{safe}} \left(\rho \bar{a}_t + (1-\rho) a_t^{(k^*)}, s_t \right).$$

The agent then executes only this first projected action in the historical replay environment, observes the next replay transition and realized replay reward, appends the new observation and action to the history, and replans at the next time step. This receding-horizon loop implements model-predictive control for offline prediction-market replay.

A.3. Replay Market-Quality Reward Specification

The planner evaluates actions using a scalar market-quality reward rather than a separate vector of market and societal objectives. The reward is designed to favor useful liquidity provision while penalizing adverse selection, excessive inventory exposure, unstable quoting, drawdown, and probability-risk exposure. For a transition from s_t to s_{t+1} under quote action a_t, the replay reward is written as

$$\begin{aligned} r_{t+1} = {} & R_{\text{pnl}} + \alpha_{\text{sc}} R_{\text{spread}} + \alpha_{\text{fq}} R_{\text{fill}} + R_{\text{liq}} \\ & - \alpha_{\text{adv}} C_{\text{adv}} - \alpha_{\text{drift}} C_{\text{drift}} - C_{\text{inv}} - \alpha_{\text{dist}} C_{\text{dist}} \quad - C_{\text{churn}} - C_{\text{dd}} - C_{\text{prob}} - C_{\text{noquote}}. \end{aligned} \tag{11}$$

Here R_{pnl} is financial mark-to-market PnL, R_{spread} measures spread capture, R_{fill} measures fill quality, and R_{liq} rewards probability-consistent liquidity provision. The penalty terms correspond to adverse selection, inventory drift, inventory risk, quote-distance cost, quote churn, drawdown, settlement-probability risk, and the cost of failing to quote in useful market states. This reward is therefore a historical replay market-quality utility. It is not interpreted as a direct live welfare measure; instead, it operationalizes stability-oriented market-making behavior through observable replay quantities.

A.4. Uncertainty-Aware Planning Objective

For each candidate sequence, the planner uses the discounted uncertainty-penalized objective

$$J^{(k)} = \sum_{h=0}^{H-1} \gamma^h \left(\hat{r}^{(k)}_{t+h+1} - \lambda_u \hat{\sigma}^{(k)}_{t+h+1} \right). \tag{12}$$

The transition approximation F updates the simulated probability level, entropy, inventory exposure, previous quote controls, quote activity, and replay phase. The reward predictor estimates the expected market-quality utility of the candidate quote, while $\hat{\sigma}$ penalizes high-risk regimes driven by volatility, low event attention, time-to-close pressure, momentum shocks, and inventory exposure. This produces a conservative model-predictive controller: it searches over continuous quote actions, avoids unstable high-uncertainty trajectories, executes one safety-projected action, and replans after the next observation.

B. Supplementary Experimental Diagnostics

This appendix provides supplementary diagnostics for the offline replay evaluation. The goal is to make the experimental behavior transparent beyond the main aggregate metrics. The figures show the structure of the selected prediction-market universe, the replay behavior of the learned policy, market-level quote and fill dynamics, inventory evolution, reward decomposition, and robustness checks across seeds and model components. Together, these diagnostics help verify that the reported performance comes from controlled market-making behavior rather than from a small number of isolated favorable outcomes.

The replay evaluation should be interpreted as an offline historical simulation. The agent does not place live market orders. Instead, it observes historical market states, generates action-conditioned quote decisions, and is evaluated under a replay reward that combines execution quality, liquidity provision, inventory control, adverse-selection penalties, quote-risk costs, and probability-risk costs. The appendix therefore emphasizes interpretability: each diagnostic connects the learned policy's decisions to observable replay quantities such as quoted prices, fills, exposure, reward, and drawdown.

B.1. Market Universe and Aggregate Replay Behavior

The first group of diagnostics characterizes the evaluation environment. Prediction markets differ substantially in category, maturity, probability level, spread, liquidity, and volatility. These properties affect both the difficulty of market making and the reliability of replay evaluation. For this reason, the appendix first summarizes the selected market universe before showing detailed policy behavior. This makes it possible to interpret the later replay results in relation to the diversity and quality of the underlying market data.

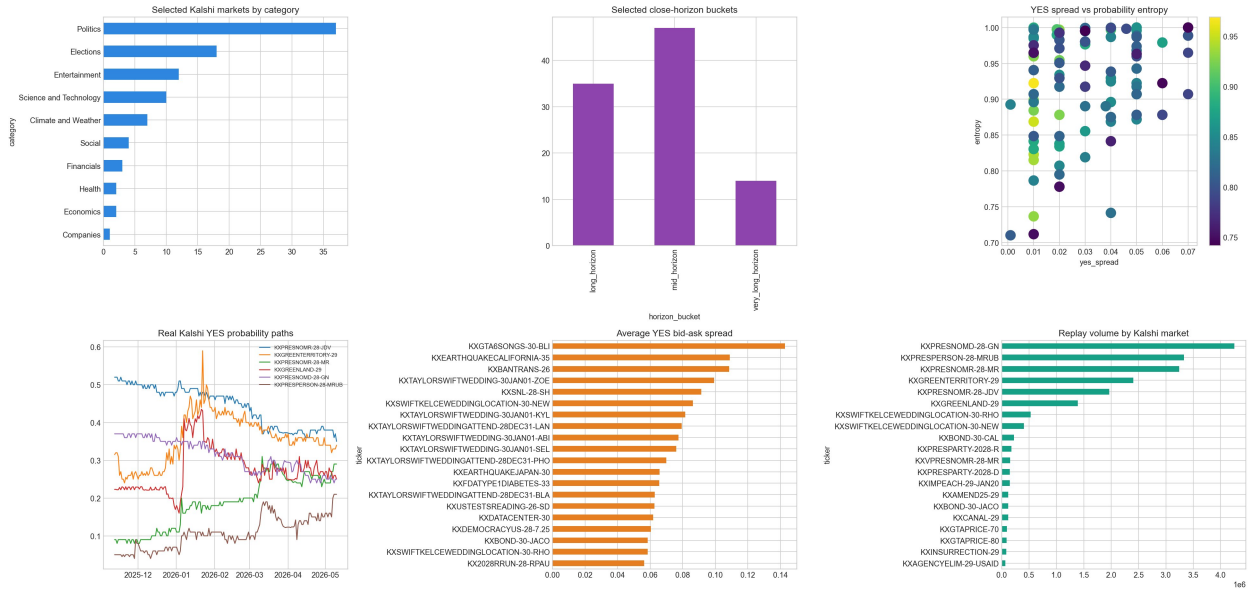


Figure A.1. Selected Kalshi Replay Universe - Event Categories, Horizons, Probability Paths, Spreads, and Volumes. This diagnostic summarizes the selected prediction-market universe used in the replay study. The panels report event-category composition, market-horizon distribution, probability trajectories, spread behavior, liquidity conditions, and replay activity. These quantities are important because the difficulty of market making is not uniform across markets: short-horizon markets may exhibit sharper probability movements, illiquid markets may create wider spreads, and high-uncertainty markets may expose the agent to larger probability-risk penalties. The figure therefore supports the validity of the replay setting by showing that the evaluation covers heterogeneous market regimes rather than a single narrow class of events.

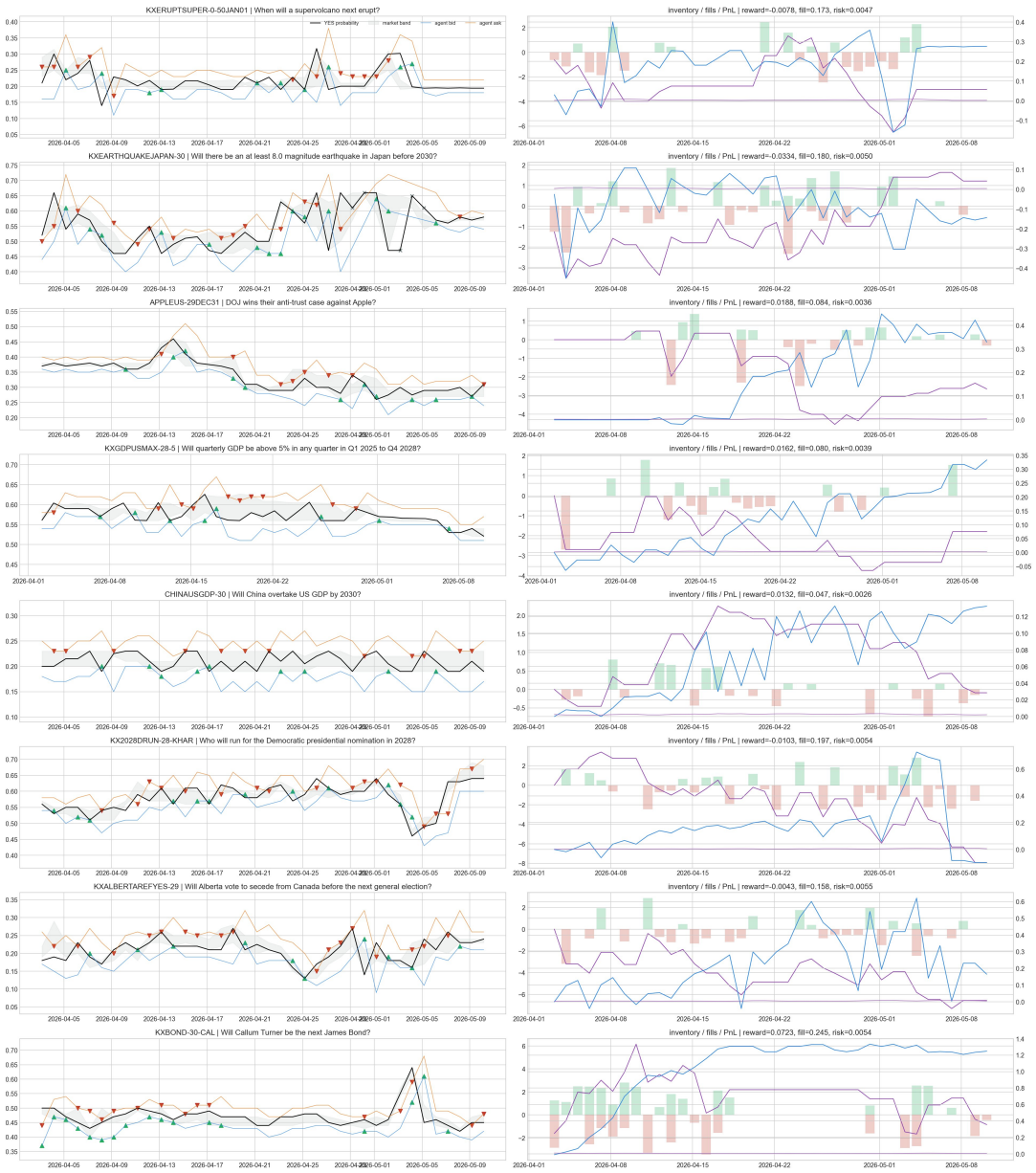


Figure A.2. Multi-Market Replay Grid - Probability-Tracked Quotes, Fills, Inventory, and Reward Components.

This grid provides a compact cross-market view of the learned policy during historical replay. In each row, the left panel compares the realized YES-probability path with the agent's quoted YES bid and ask prices; green and red markers indicate simulated buy and sell fills, while inactive or held periods are marked separately. The right panel tracks inventory, cumulative financial PnL, and probability-risk cost for the same market. The figure shows that the policy does not act as a simple directional forecaster; it repeatedly places quotes around the market probability, receives fills when quotes are reached, and then carries or unwinds inventory under risk-sensitive replay costs. Differences across rows show how the same policy responds to low-probability, mid-probability, political, macroeconomic, climate, and entertainment contracts.

B.2. Individual Market Replay Tapes

The following replay tapes provide market-level evidence for the behavior of the policy across heterogeneous event domains. Each tape follows the same visual structure, which allows direct comparison of quote placement, fill timing, inventory management, cumulative reward, and risk penalties. These plots are especially important because a policy can show strong average reward while still behaving poorly in individual markets. The replay tapes help verify that the policy's aggregate performance is supported by controlled and interpretable market-level behavior.

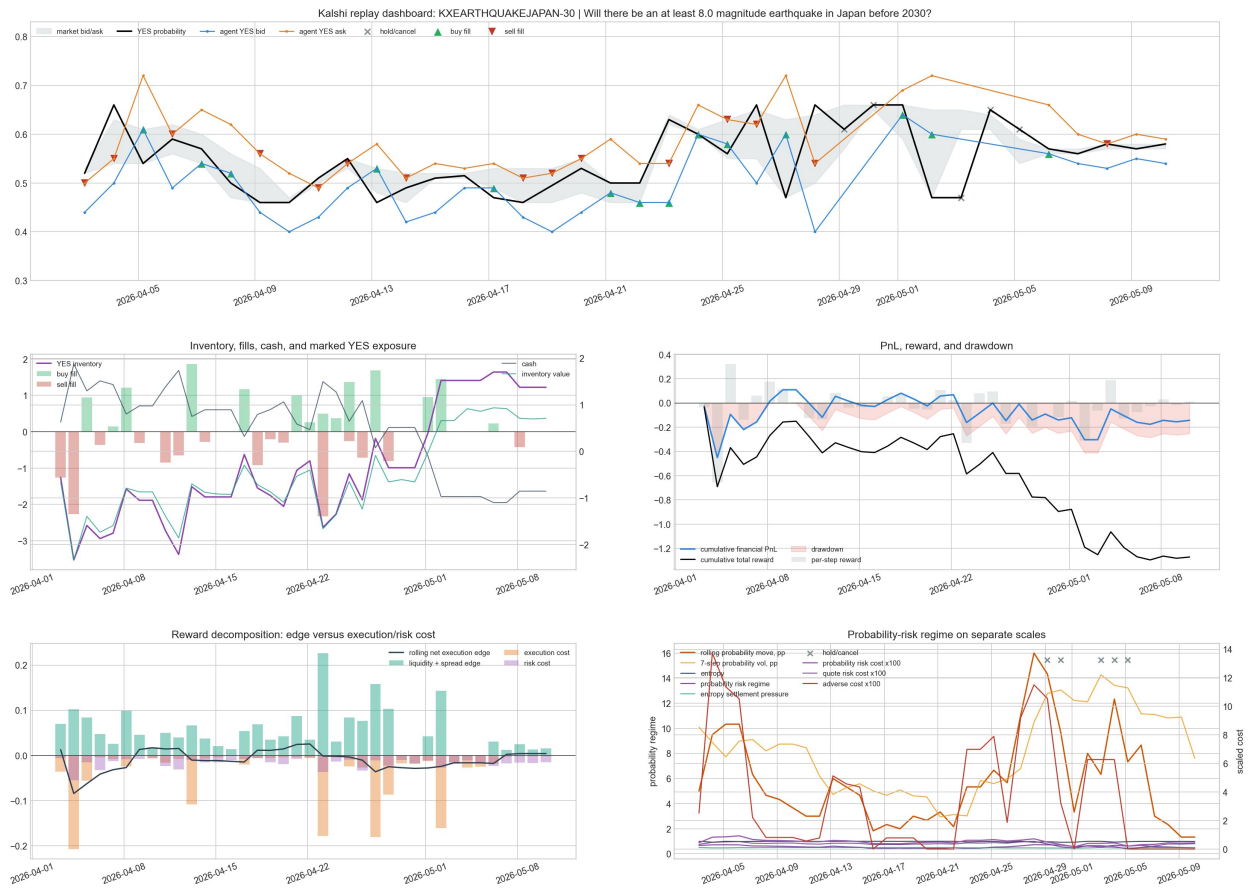


Figure A.3. Japan Earthquake Forecast Replay - Quote Adjustment Around an 8.0-Magnitude Event Contract. This figure reports the replay for the market asking whether Japan will experience at least an 8.0 magnitude earthquake before 2030. The YES probability moves within a higher and more volatile range than the supervolcano contract, and the policy is less continuously active, with some hold or cancel periods visible in the quote panel. The fill panels show meaningful simulated execution, but the reward decomposition shows that adverse-selection cost dominates the positive liquidity contribution. As a result, the replay has negative total reward and slightly negative financial PnL.

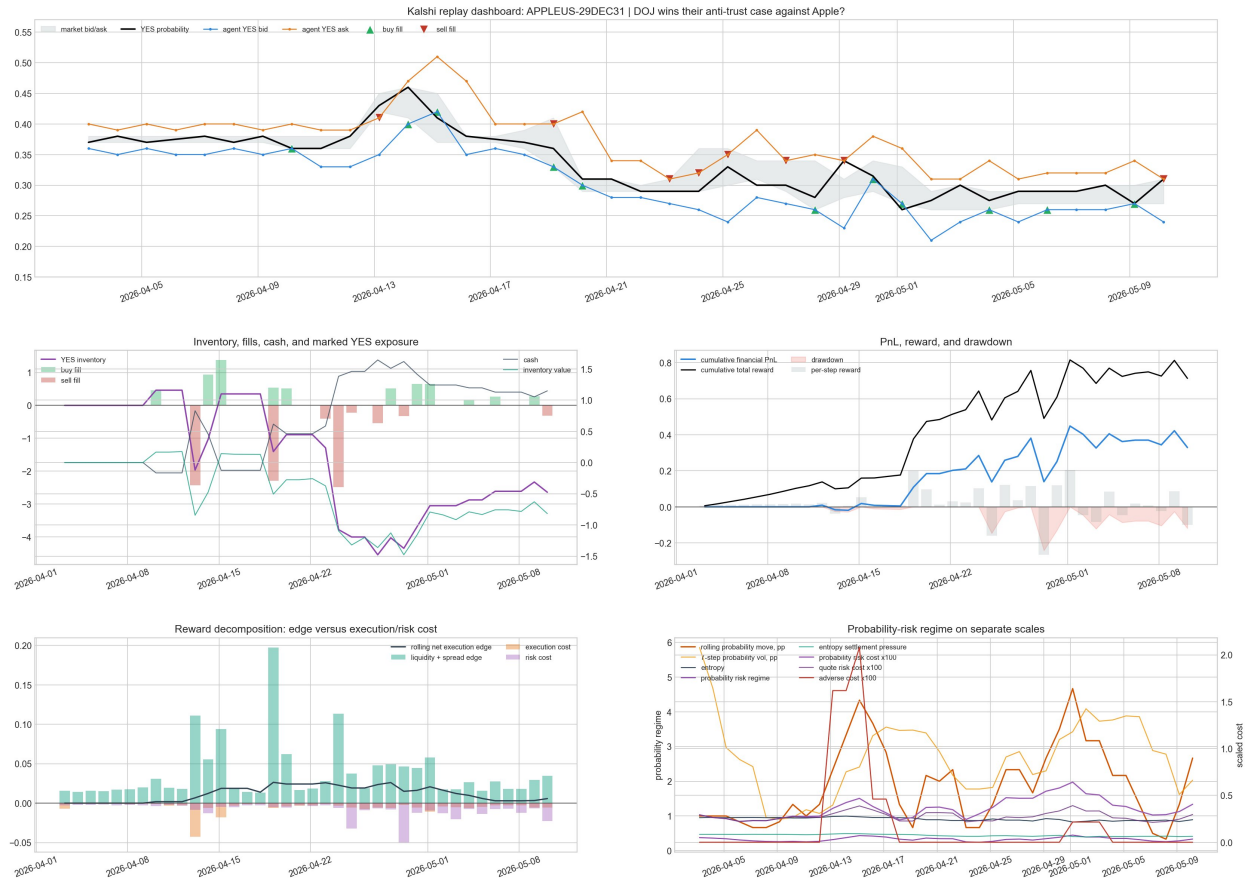


Figure A.4. Apple Antitrust Forecast Replay - Gradual Probability Drift, Fill Timing, and Reward Accumulation.

This figure shows replay behavior for the market asking whether the DOJ wins its antitrust case against Apple. The probability path trends downward from the mid-probability region, and the policy maintains quote activity while receiving a moderate number of fills. Unlike the climate-event examples, the adverse-selection cost is small, so the liquidity and execution components translate into positive cumulative reward. The PnL and reward curves therefore move in a more favorable direction, while drawdown remains limited relative to the higher-risk replay tapes. This figure is an example of the policy performing well when probability movement is smoother and the executed fills are less heavily penalized by risk costs.

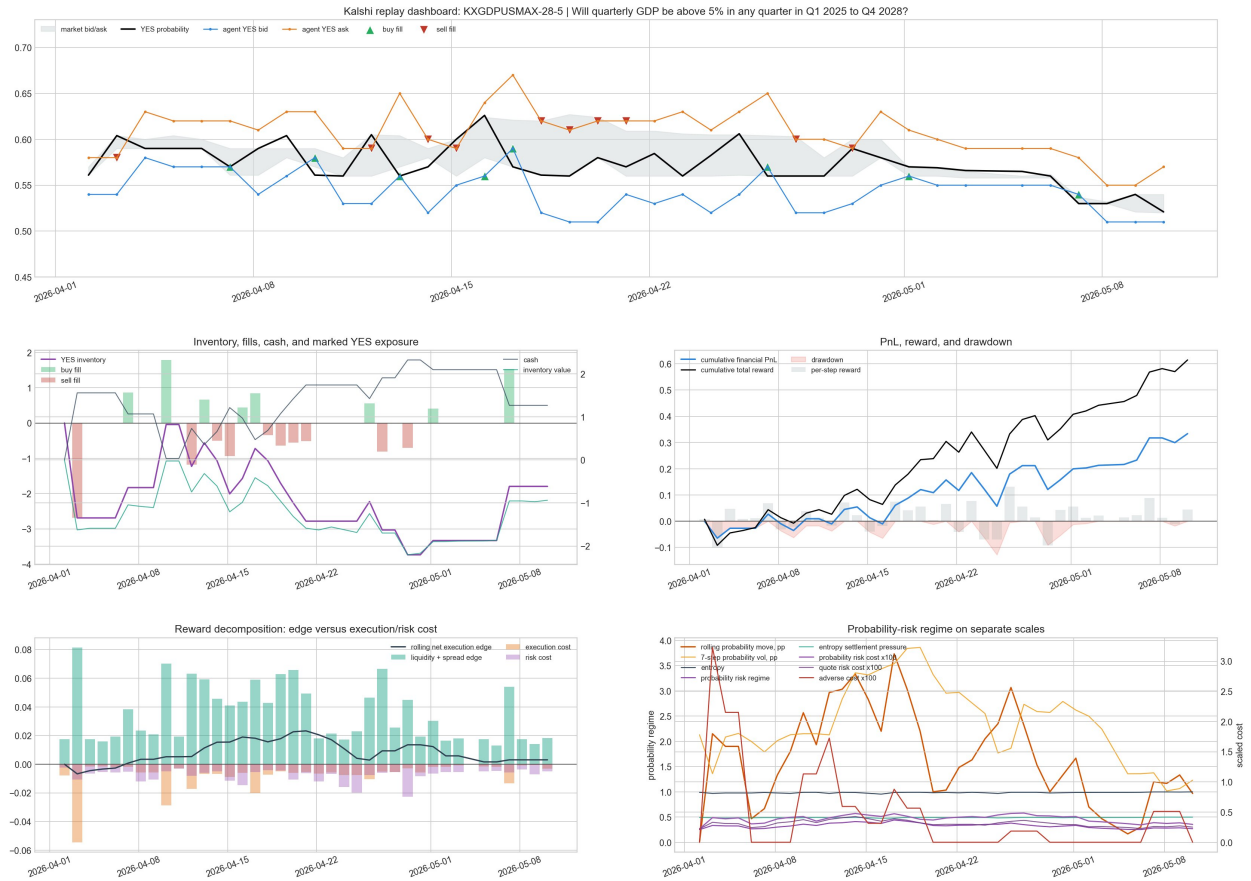


Figure A.5. U.S. GDP Growth Forecast Replay - Macroeconomic Probability Path and Market-Making Exposure.

This figure reports replay behavior for the market asking whether quarterly U.S. GDP will exceed 5% in any quarter from Q1 2025 to Q4 2028. The YES probability remains in a moderate range and gradually declines toward the end of the replay. The policy keeps quotes active throughout the trajectory, receives relatively sparse fills, and maintains bounded inventory exposure. The reward decomposition shows that liquidity utility and net execution edge exceed the adverse-selection and probability-risk costs, producing positive total reward. This plot shows the policy behaving like a conservative market maker in a macroeconomic setting where probability movement is present.

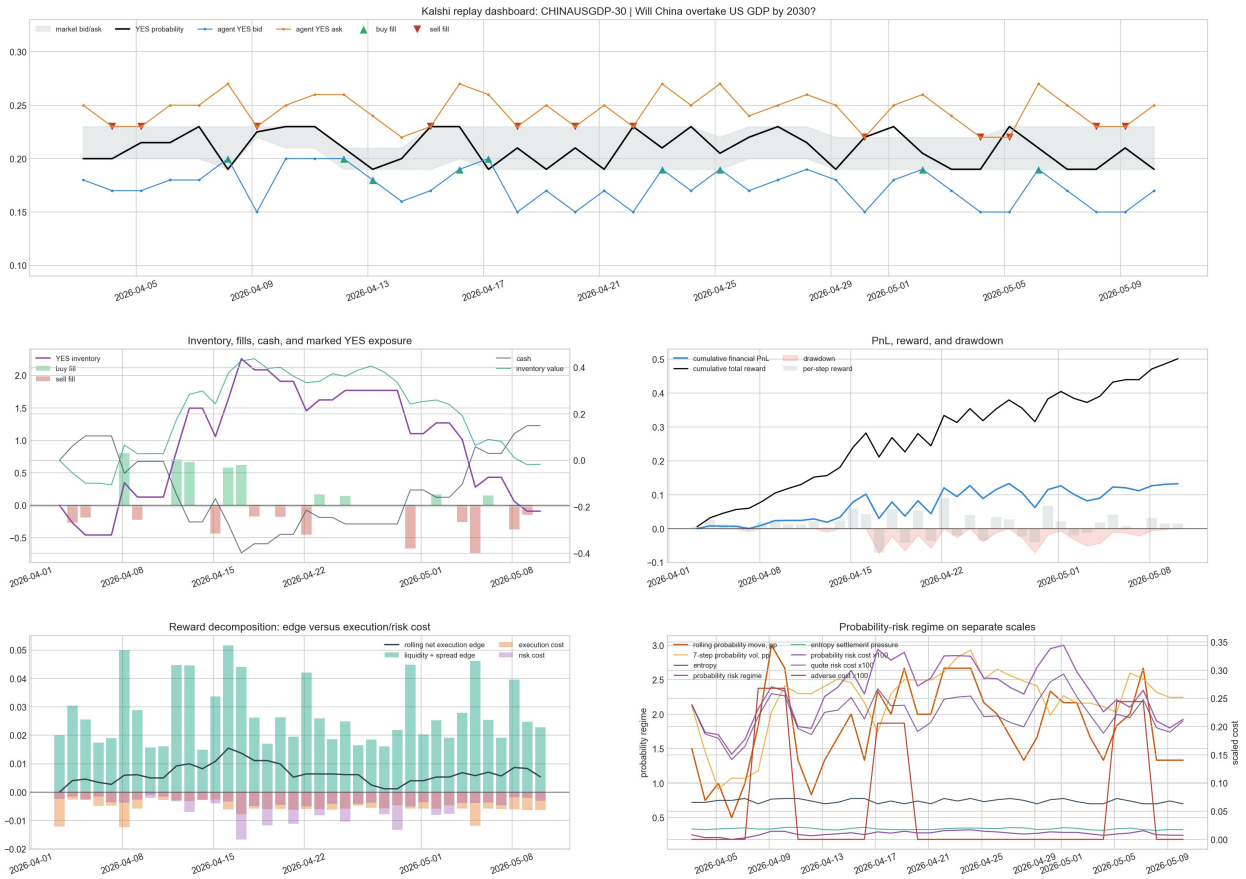


Figure A.6. China-U.S. GDP Forecast Replay - Long-Horizon Macro-Geopolitical Market Making.

This figure shows the replay for the market asking whether China will overtake U.S. GDP by 2030. The YES probability stays low and relatively stable, and the agent's quote activity produces only sparse simulated fills. Because inventory exposure remains limited and adverse-selection cost is small, the replay reward remains positive despite modest financial PnL. The diagnostic is useful because it shows that the policy can still produce market-making value in a low-probability, low-fill setting by avoiding excessive exposure and limiting probability-risk penalties.

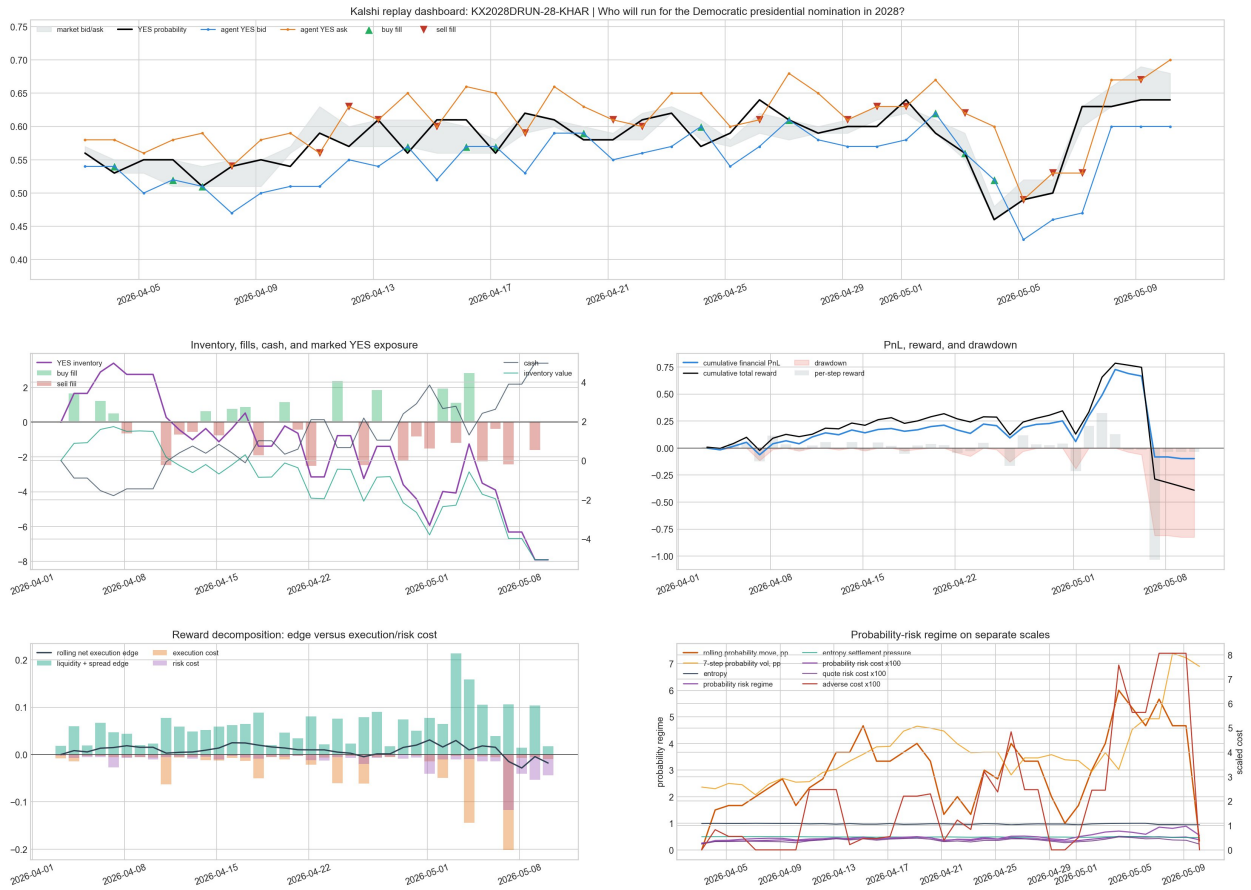


Figure A.7. 2028 Democratic Nomination Forecast Replay - Candidate-Specific Political Market Dynamics.

This political-market replay provides a different event domain from the macroeconomic and financial examples. The purpose is to examine whether the learned policy generalizes its quote placement and inventory-control behavior across market categories. This figure reports replay behavior for the market asking who will run for the Democratic presidential nomination in 2028. The policy receives many simulated fills and maintains continuous quote activity, but the inventory and drawdown panels show that this activity creates substantial exposure. Although the net execution edge is positive, the adverse-selection and probability-risk costs are large enough to make the total reward negative. The plot demonstrates why the reward model penalizes aggressive participation in unstable political probability regimes.

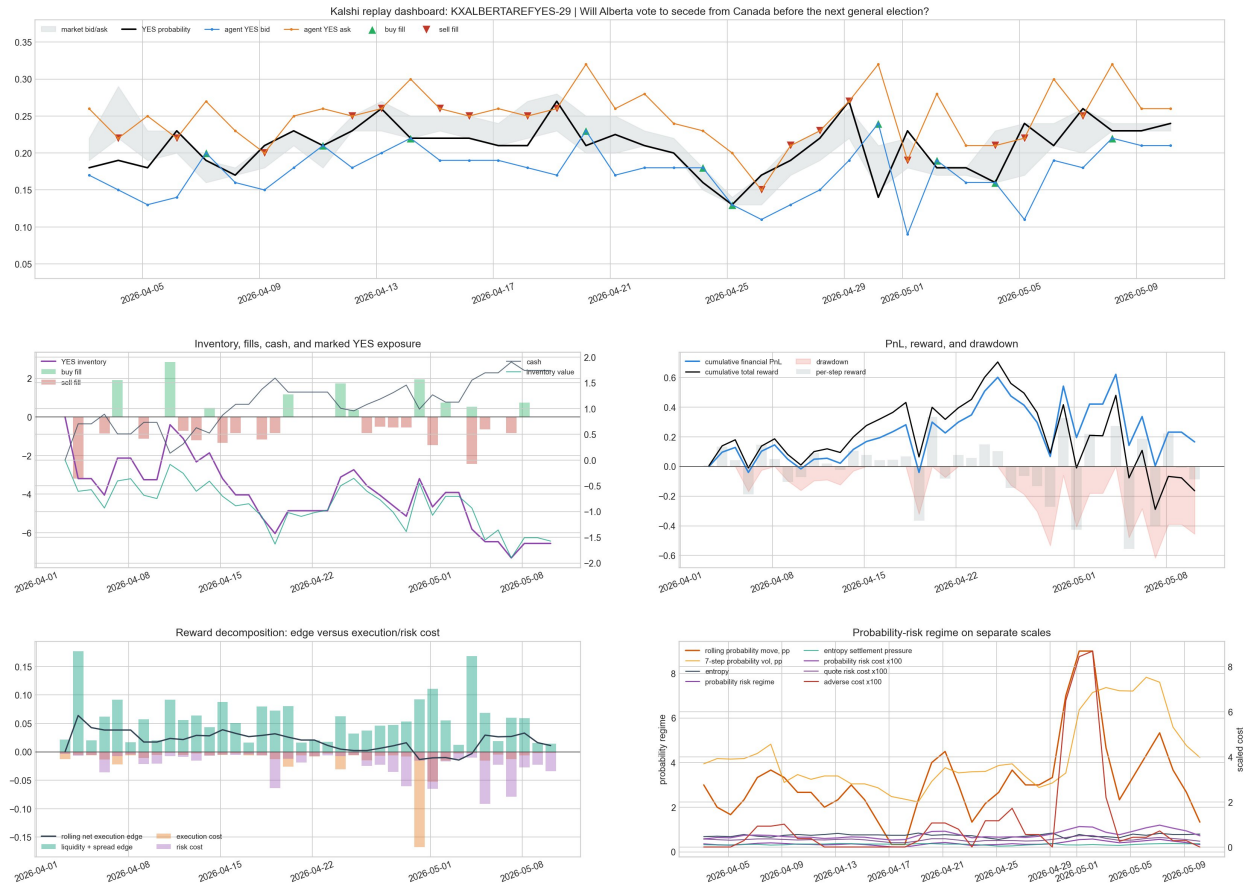


Figure A.8. Alberta Secession Referendum Replay - Event-Driven Political Forecast Market Making.

This figure shows replay behavior for the market asking whether Alberta will vote to secede from Canada before the next general election. The YES probability stays relatively low but moves enough to generate repeated fills and noticeable inventory swings. The policy earns positive financial PnL and positive net execution edge, but the total replay reward is slightly negative once adverse-selection, probability-risk, and drawdown-related penalties are included.

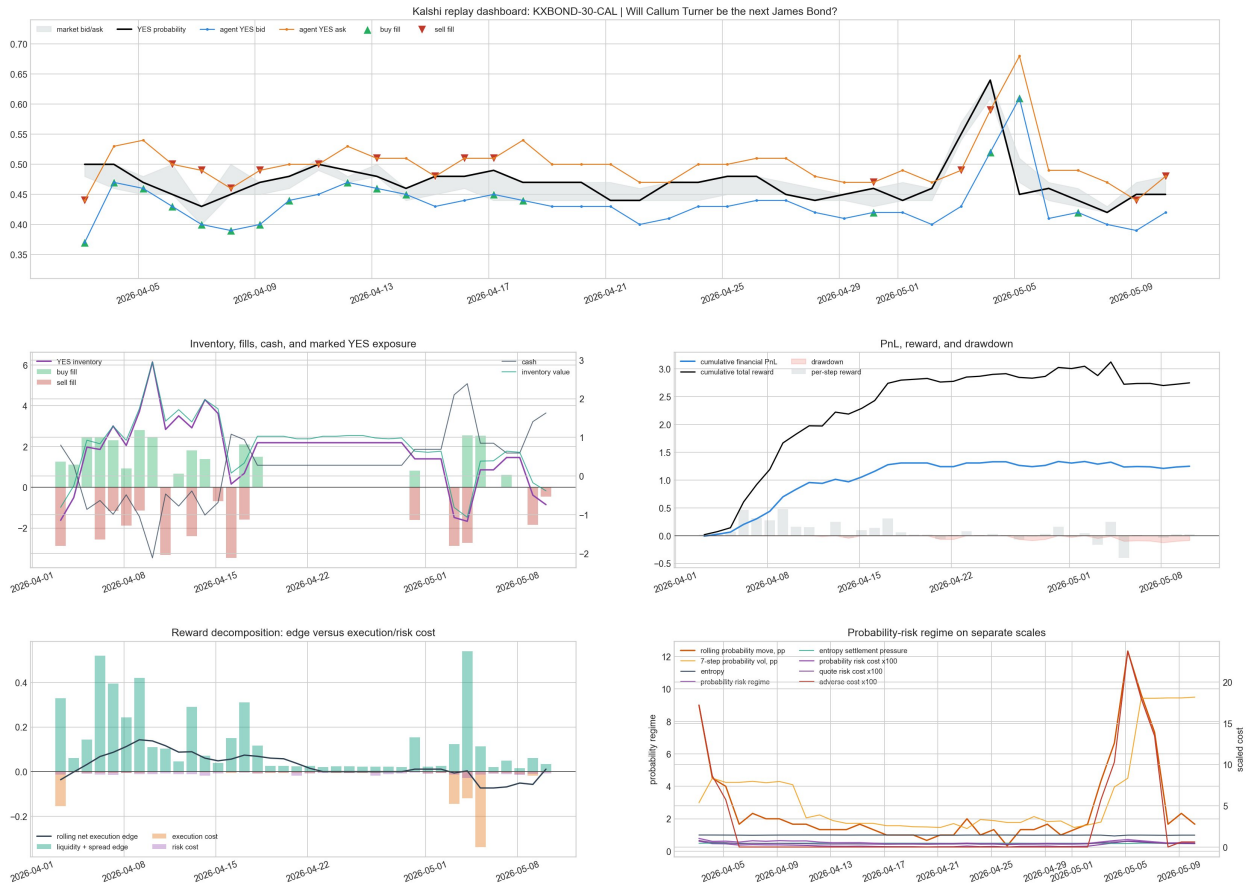


Figure A.9. James Bond Casting Forecast Replay - Entertainment-Market Quote and Fill Dynamics.

This figure reports replay behavior for the market asking whether Callum Turner will be the next James Bond. Among the replay tapes, this market shows the strongest positive reward profile, with high fill activity, positive financial PnL, and a large positive net execution edge. The policy quotes continuously, receives both buy and sell fills, and converts the resulting activity into cumulative reward despite nontrivial adverse-selection and probability-risk costs. The drawdown remains comparatively controlled, so spread capture and liquidity provision dominate the penalty terms.

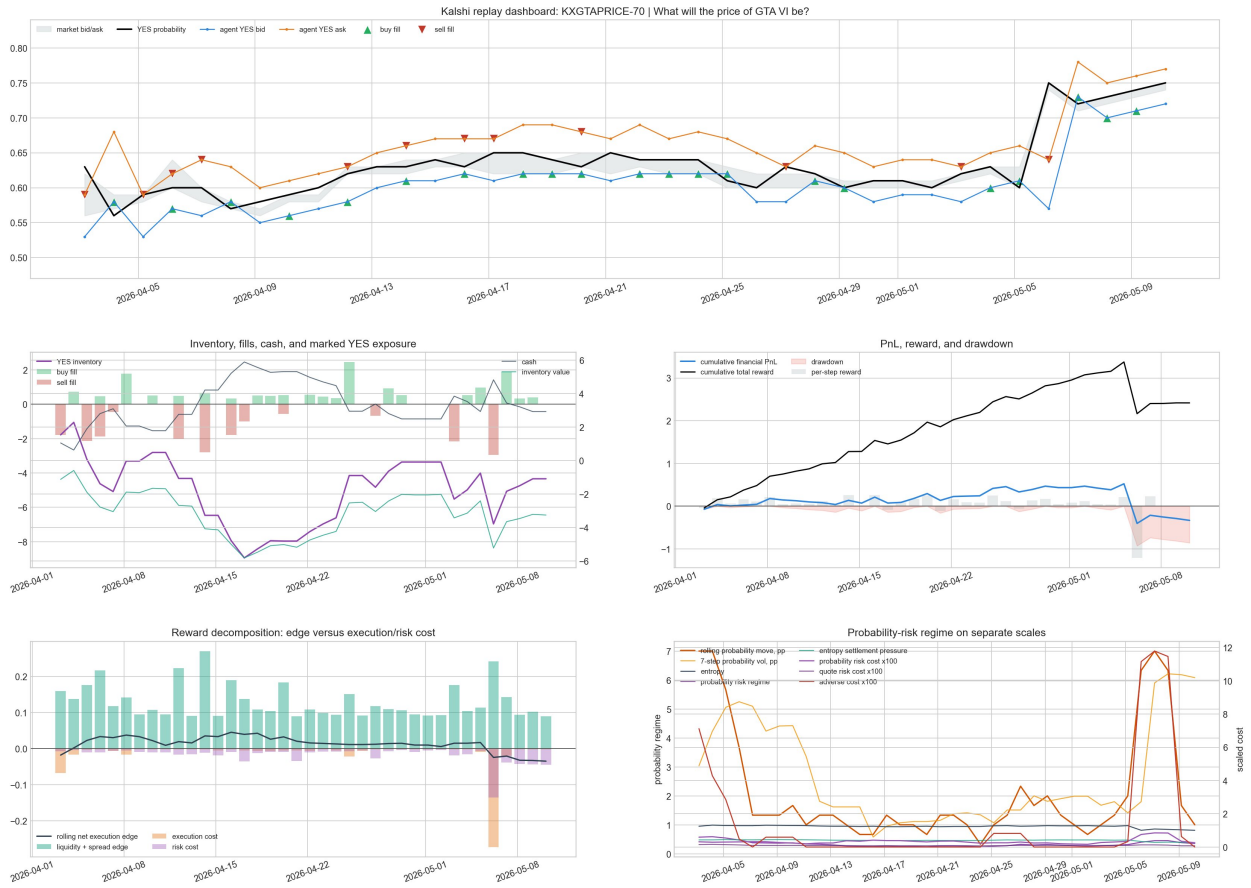


Figure A.10. GTA VI Price Forecast Replay - Consumer-Event Probability Movement and Risk Decomposition.

This figure shows replay behavior for the market asking about the price of GTA VI. The YES probability rises over the replay window, and the policy remains active while receiving many fills. The inventory panel shows larger exposure than in several other tapes, and the cumulative PnL turns negative despite positive liquidity and execution-edge components. The reward decomposition indicates that adverse-selection, probability-risk, and drawdown effects offset the positive market-making terms. This plot is useful as a stress case because it shows that continuous quote activity in a rising probability market can become costly when inventory and subsequent price movement are not favorable.

B.3. Robustness and Component Diagnostics

The final group of diagnostics examines whether the observed behavior is stable and whether the main modeling components contribute meaningfully to the result. This is important because a learned simulator can appear effective in one replay while being sensitive to seeds, market selection, or reward-model noise. The following plots therefore evaluate repeated simulation stability, replay reward mechanics, and ablations of the learned components.

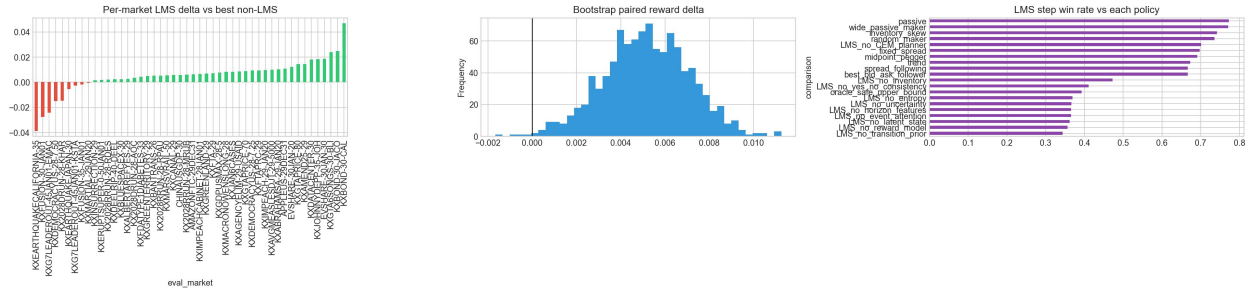


Figure A.11. Paired Market-Level Reward Deltas, Bootstrap Improvement Uncertainty, and Step-Level Win Rates.

This figure evaluates whether the LMS performance improvement is consistent across held-out markets and comparison policies. The left panel reports per-market reward differences between LMS and the strongest non-LMS baseline, showing whether improvement is broadly distributed or driven by a few outlier markets. The middle panel shows the bootstrap distribution of paired reward deltas, providing uncertainty around the average LMS improvement. The right panel reports step-level win rates against individual comparison policies, indicating how often LMS obtains higher replay reward at matched evaluation steps. Across the evaluated markets, LMS outperforms the strongest non-LMS baseline in 81.25% of markets, with a mean paired reward delta of 0.005098 and a bootstrap confidence interval of $[0.001170, 0.008886]$.

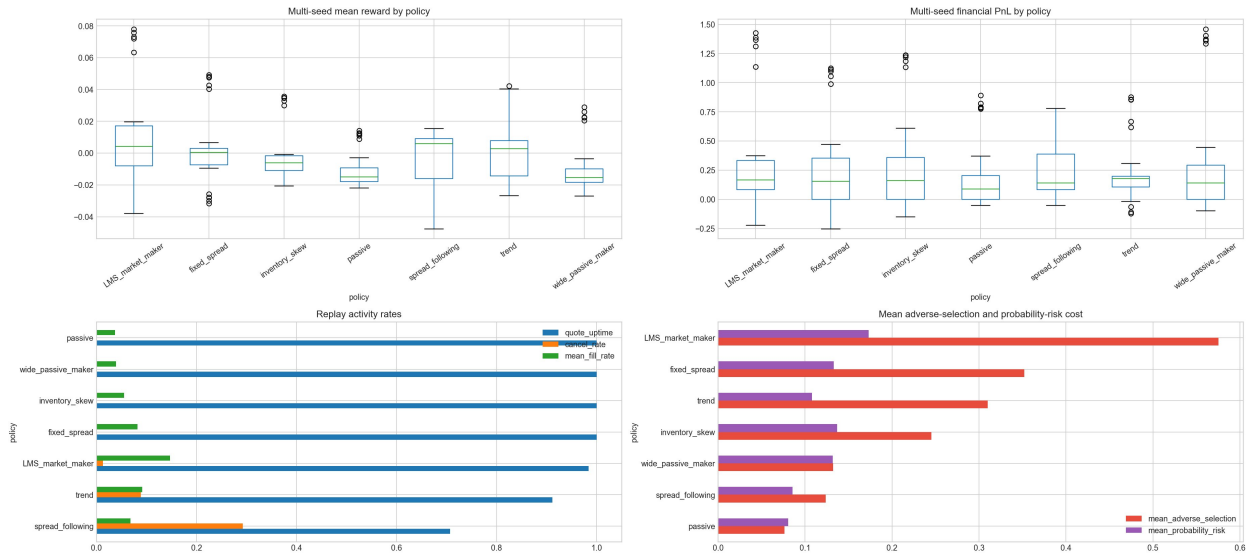


Figure A.12. Multi-seed replay stability

This figure evaluates whether policy performance is stable across repeated replay simulations. The upper panels compare mean reward and total financial PnL distributions across policies, showing that the LMS market maker has the strongest mean reward among the non-oracle policies in the multi-seed evaluation. The lower-left panel shows activity behavior; the LMS policy maintains high quote uptime, low cancel rate, and higher fill activity than passive or wide-spread baselines. The lower-right panel compares adverse-selection and probability-risk costs, showing that stronger reward can come with greater exposure to execution risk because the policy is more active.

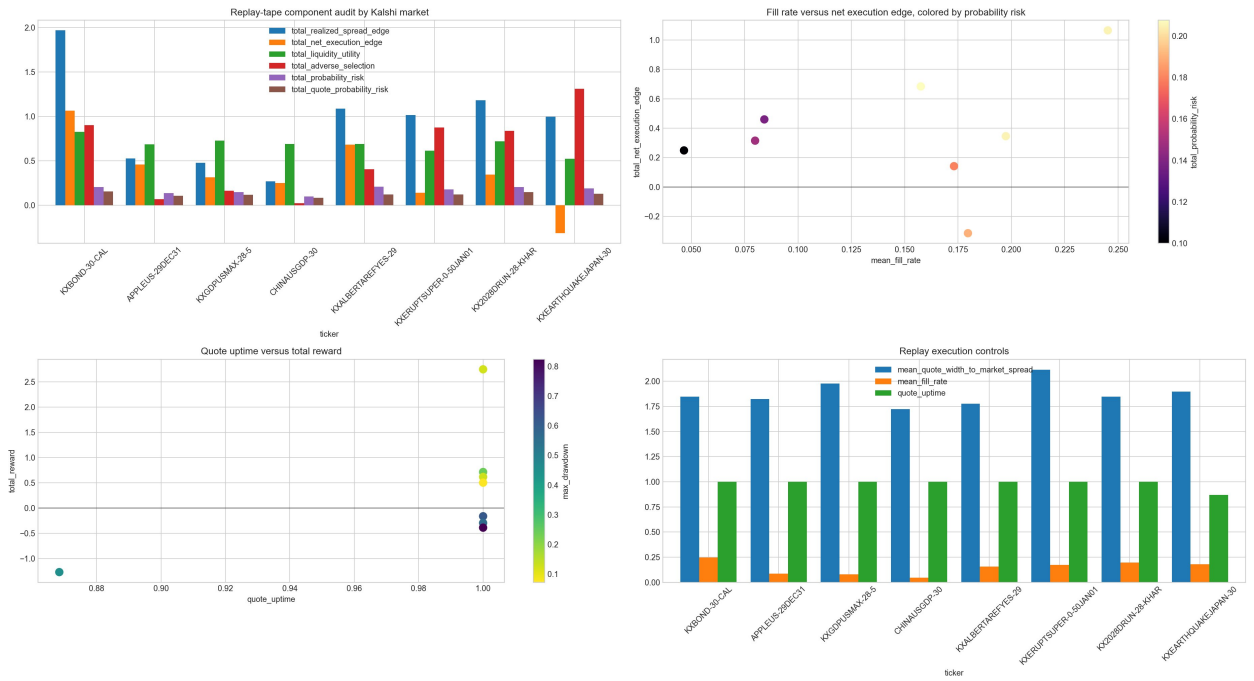


Figure A.13. Replay Reward Component Audit

This figure audits the mechanics of the replay reward across the detailed replay markets. The bar chart decomposes each market into realized spread edge, net execution edge, liquidity utility, adverse-selection cost, probability-risk cost, and quote-risk cost, making clear which markets generate reward from liquidity provision and which are penalized by risk. The fill-rate versus net-execution-edge scatter shows that more fills do not automatically imply better execution quality.

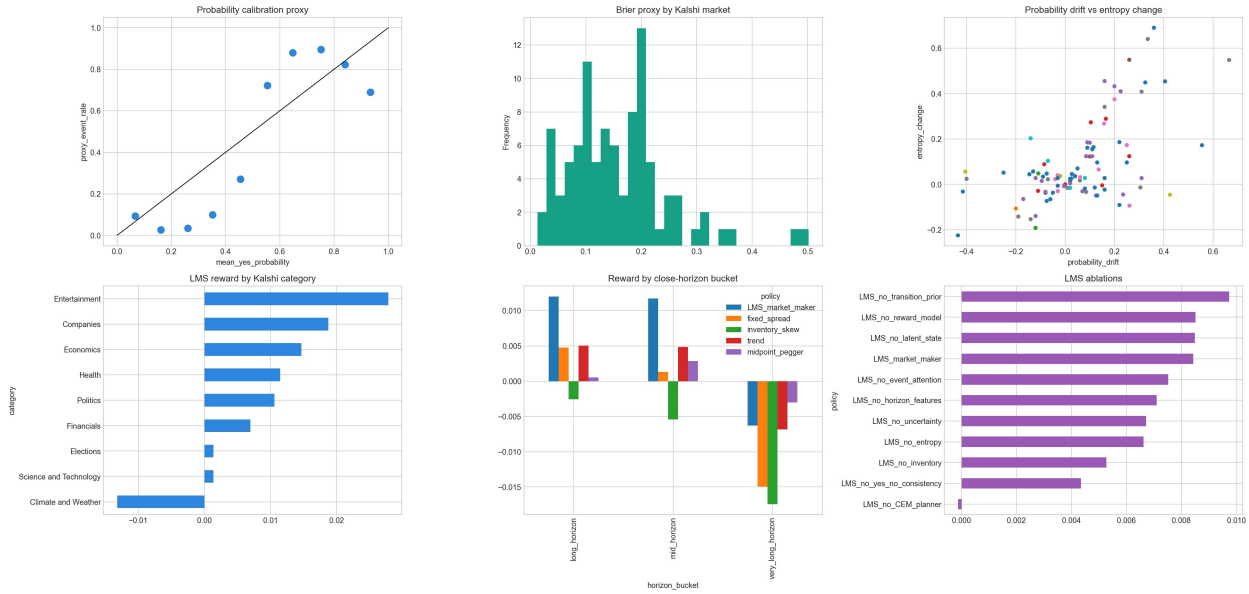


Figure A.14. Probability Calibration, Probability Drift, Category-Level Reward, Horizon-Level Reward, and LMS Component Ablations.

This figure combines probability-quality checks with robustness and ablation diagnostics. The calibration panel compares mean YES probability against a proxy event rate; the relationship is broadly monotonic in the middle probability ranges but less reliable near the extremes, where fewer observations are available. The Brier-proxy histogram shows the distribution of market-level probability error, while the probability-drift versus entropy-change scatter shows how market uncertainty changes as probabilities move over time. The lower panels show that LMS reward varies across event categories and horizon buckets, indicating that performance is not uniform across market types. The ablation panel compares the full LMS policy with variants that remove uncertainty handling, inventory features, event attention, horizon features, entropy, latent state, transition prior, reward model, or CEM planning, showing how individual modeling and planning components affect replay reward.