CAPO: Machine Learning-Driven Price Discovery for Competitive Market Making

Ryan Watts

August 14, 2025

Abstract

Market making in illiquid stablecoin markets presents unique challenges due to limited trading volume and aggressive competition for top-of-book positioning. We study markets on Kraken US with average daily volumes between $50-$4,000 USD, where 2-6 competitive market makers engage in bidding wars when spreads exceed their individual thresholds. These pricing conflicts consume valuable API calls and reduce top-of-book time, which is critical since trades occur only hourly. We introduce CAPO (Competitive Aggressive Pricing Optimizer), a machine learning framework that classifies price levels as either "safe" (maintaining top-of-book position) or "toxic" (losing competitive advantage). CAPO addresses the dynamic nature of competitor strategies by predicting minimum safe pricing levels through active exploration and boundary discovery. Our system uses Random Forest regressors with confidence-weighted training on 12 engineered features, including market volatility, spread metrics, and historical win rates. The active learning component systematically explores pricing boundaries while maintaining safety constraints to prevent adverse market impact. Empirical results demonstrate improved top-of-book positioning while preserving profitable spread levels, with the system successfully adapting to competitors' changing minimum spread requirements in real-time market conditions.

1 Introduction

1.1 Market Making Challenges

Traditional market making operates on the principle of providing liquidity by continuously quoting bid and ask prices, profiting from the spread while managing inventory risk. Classic models like Avellaneda-Stoikov focus on optimal spread setting based on volatility, inventory position, and risk aversion parameters. However, these frameworks assume relatively liquid markets with predictable flow patterns and well-behaved competitor dynamics.

Our research focuses on a fundamentally different environment: illiquid stablecoin markets on Kraken US. These markets present unique challenges that traditional market making theory does not adequately address. Daily trading volumes typically range from $50 to $4,000 USD, with trades occurring sporadically, often with hour-long gaps between transactions.

In such thin markets, each trade represents a significant portion of daily volume, making top-of-book positioning critical for capturing available flow.

The competitive landscape in these markets is particularly intense. Typically, 2-6 sophisticated market makers compete aggressively for the limited order flow. Unlike liquid markets where spreads naturally converge through arbitrage, these illiquid venues exhibit a different dynamic. Each competitor operates with their own minimum spread threshold, say 70 basis points, below which they will not provide liquidity. When a competitor detects our quotes tightening beyond their comfort zone, they initiate aggressive repricing campaigns, leading to rapid-fire quote updates that consume valuable API rate limits.

This creates a critical risk management versus profitability trade-off. Aggressive pricing can capture more volume but triggers costly bidding wars that reduce our effective top-of-book time. Conservative pricing avoids conflicts but may result in missed trading opportunities in markets where volume is already scarce. The challenge is compounded by the fact that competitor strategies change frequently, with minimum spread thresholds shifting from 70 basis points to 40 basis points within minutes as market conditions or competitor algorithms adapt.

1.2 Problem Statement

To address these challenges, we introduce the concept of "safe" versus "toxic" price levels. A safe price level is one where our quotes maintain top-of-book positioning without triggering aggressive competitive responses. Conversely, a toxic price level provokes immediate repricing from competitors, resulting in loss of priority and wasted API calls. The fundamental challenge lies in discovering these optimal pricing boundaries in real-time, as they shift dynamically based on competitor behavior, market conditions, and time of day.

The problem of predicting these boundaries is particularly complex because competitor algorithms are constantly evolving. What constituted a safe spread yesterday may be toxic today as competitors update their models or adjust their risk parameters. Traditional static approaches fail because they cannot adapt quickly enough to these changing dynamics. We need a system that can learn from market feedback and continuously refine its understanding of competitive boundaries.

This creates a classic exploration versus exploitation dilemma. To discover new safe pricing levels, we must occasionally test aggressive prices that may prove toxic, consuming API calls and potentially losing top-of-book positioning. However, pure exploitation of known safe levels may leave profitable opportunities undiscovered as competitor thresholds evolve. The optimal strategy requires intelligent exploration that balances information gathering with operational safety, ensuring we maintain competitive positioning while discovering new pricing boundaries.

Our approach addresses these challenges through CAPO (Competitive Aggressive Pricing Optimizer), a machine learning framework that learns to classify price levels and predict optimal competitive boundaries through active exploration and real-time adaptation.

2 Methodology

2.1 Problem Formulation

We formalize the competitive market making problem as a real-time classification and prediction task. Let $S_t = (b_t^{(1)}, ..., b_t^{(k)}, a_t^{(1)}, ..., a_t^{(k)})$ represent the market state at time t, where $b_t^{(i)}$ and $a_t^{(i)}$ are the i-th best competitor bid and ask prices respectively. Our action space consists of bid-ask price pairs (p_t^b, p_t^a) subject to bounds constraints $p_t^b \leq B_t^b$ and $p_t^a \geq B_t^a$, where (B_t^b, B_t^a) represent our maximum allowable bid and minimum allowable ask prices.

For any given price level p, we define the safety classification function:

$$\text{Safe}(p, S_t, \text{side}) = \begin{cases} 1 & \text{if } p \text{ maintains top-of-book without triggering competition} \\ 0 & \text{otherwise (toxic)} \end{cases}$$

The objective is to maximize our top-of-book time while maintaining profitable spreads:

$$\max \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}_{\text{top-of-book}}(p_t^b, p_t^a, S_t) \cdot \text{PnL}_t\right]$$

subject to safety constraints and operational limits on API usage.

The challenge lies in learning the safety classification function in real-time as competitor strategies evolve. Traditional approaches fail because the underlying competitive dynamics are non-stationary, requiring continuous adaptation and exploration.

2.2 Feature Engineering

CAPO operates on a 12-dimensional feature vector extracted from recent market snapshots. These features capture market microstructure, competitive dynamics, and our historical performance.

2.2.1 Market State Features

The first six features capture current market conditions and recent dynamics:

Features 1-2: Current Market Prices $f_1 = b_t^{(1)}, \quad f_2 = a_t^{(1)}$ where $b_t^{(1)}$ and $a_t^{(1)}$ are the current best competitor bid and ask prices.

Features 3-4: Spread Metrics $f_3 = a_t^{(1)} - b_t^{(1)}, \quad f_4 = \frac{1}{w}\sum_{i=t-w+1}^{t}(a_i^{(1)} - b_i^{(1)})$ where f_3 is the current spread and f_4 is the average spread over a window of $w = 10$ snapshots.

Features 5-6: Volatility Measures $f_5 = \frac{1}{w}\sum_{i=t-w+1}^{t}|b_i^{(1)} - b_{i-1}^{(1)}|$,
$f_6 = \sqrt{\frac{1}{w-1}\sum_{i=t-w+1}^{t}(|b_i^{(1)} - b_{i-1}^{(1)}| - f_5)^2}$ where f_5 captures recent price volatility and f_6 measures volatility of volatility.

2.2.2 Performance Features

Features 7-8 track our historical competitive success: $f_7 = \frac{\text{Number of snapshots where our bid was top-of-book}}{\text{Total snapshots}}, \quad f_8 = \frac{\text{Number of snapshots where our ask was top-of-book}}{\text{Total snapshots}}$

These win rates provide context about our recent competitive positioning and help the model understand our relative aggressiveness.

2.2.3 Boundary Estimation Features

Features 9-10 incorporate our learned estimates of competitive boundaries: $f_9 = \mathbb{E}[\hat{B}_{\text{bid}}], \quad f_{10} = \mathbb{E}[\hat{B}_{\text{ask}}]$ where $\hat{B}_{\text{bid}}$ and $\hat{B}_{\text{ask}}$ are our estimated minimum safe bid and maximum safe ask levels based on recent boundary discoveries.

2.2.4 Learning Quality Features

Features 11-12 capture the quality and confidence of our learning process: $f_{11} = \frac{\text{Successful explorations}}{\text{Total explorations}}, \quad f_{12} = \mathbb{E}[\text{confidence scores for recent boundary estimates}]$

These features help the model understand how reliable our current boundary knowledge is and adjust its predictions accordingly.

2.3 CAPO Architecture

2.3.1 Core ML Models

CAPO employs separate Random Forest Regressor models for bid and ask price prediction. We chose Random Forests for their ability to handle non-linear relationships, robustness to outliers, and interpretability through feature importance scores.

Each model is configured with: $\text{RF}_{\text{bid/ask}} = \text{RandomForestRegressor}(n\_estimators = 50, max\_depth = 12, min\_samples\_split = 3, min\_samples\_leaf = 2)$

The models predict safe price levels rather than binary classifications, allowing us to incorporate confidence intervals and safety buffers. For a given feature vector $\mathbf{f}_t$, the models output: $\hat{p}_t^b = \text{RF}_{\text{bid}}(\mathbf{f}_t) + \beta \cdot \delta, \quad \hat{p}_t^a = \text{RF}_{\text{ask}}(\mathbf{f}_t) - \beta \cdot \delta$ where β is a safety buffer parameter (default 2 ticks) and δ is the minimum tick size.

2.3.2 Confidence-Weighted Training

Training samples are weighted by confidence scores that reflect the reliability of each observation. The confidence score c_i for training sample i is calculated as: $c_i = c_{\text{base}} + c_{\text{exploration}} + c_{\text{boundary}} + c_{\text{recency}}$

where: $c_{\text{base}} = 0.5$ (baseline confidence), $c_{\text{exploration}} = 0.4$ if the sample came from active exploration, 0 otherwise, and $c_{\text{boundary}} = 0.2$ if the price was within 3 ticks of an estimated boundary, 0 otherwise - $c_{\text{recency}} = 0.1 \cdot e^{-\lambda(t_{\text{current}} - t_i)}$ with decay parameter λ

This weighting scheme prioritizes recent data, exploration results, and observations near competitive boundaries.

2.3.3 Active Learning Component

The active learning system determines when and where to explore new pricing levels. Exploration is triggered when: $\text{Explore} = \begin{cases} \text{True} & \text{if } \xi < r_{\text{explore}} \text{ or } |\mathcal{B}_{\text{bid}}| < 5 \text{ or } |\mathcal{B}_{\text{ask}}| < 5 \\ \text{False} & \text{otherwise} \end{cases}$

where $\xi \sim \text{Uniform}(0, 1)$, r_{explore} is the exploration rate (default 0.15), and $|\mathcal{B}_{\text{bid/ask}}|$ represents the number of recent boundary estimates.

When exploration is triggered, candidate prices are generated as: $p^{\text{bid}}_{\text{explore}} = b^{(1)}_t + k \cdot \delta, \quad p^{\text{ask}}_{\text{explore}} = a^{(1)}_t + k \cdot \delta$ where $k \sim \text{Uniform}(-5, 5)$ represents random tick offsets within a range of ± 5 ticks.

2.4 Safety Mechanisms

2.4.1 Bounds Checking

All price decisions are subject to hard bounds constraints. If market conditions violate our predefined bounds (B^b_t, B^a_t), CAPO disables ML predictions and reverts to conservative base pricing: $\text{Enhanced Price} = \begin{cases} \text{ML Prediction} & \text{if bounds satisfied and no violations detected} \\ \text{Base Price} & \text{otherwise} \end{cases}$

Bounds violations are tracked and reported, providing feedback on market regime changes that may require strategy adjustments.

2.4.2 Safe Exploration Constraints

Exploration prices must satisfy minimum spread requirements to prevent dangerous market conditions: $\text{Exploration Safe} = \begin{cases} \text{True} & \text{if } p^a_{\text{explore}} - p^b_{\text{explore}} \geq s_{\min} \\ \text{False} & \text{otherwise} \end{cases}$

where $s_{\min}$ is the minimum allowable spread (default 3 ticks). This prevents exploration from creating locked or crossed markets.

2.4.3 API Rate Management

To prevent excessive API usage during exploration, the system enforces minimum intervals between exploration attempts: $\text{Exploration Allowed} = \begin{cases} \text{True} & \text{if } t - t_{\text{last\_explore}} \geq \tau_{\min} \\ \text{False} & \text{otherwise} \end{cases}$

where $\tau_{\min}$ is the minimum exploration interval (default 3 snapshots).

This architecture ensures that CAPO can discover optimal pricing boundaries while maintaining operational safety and respecting system constraints.

3 Algorithm Design

3.1 Training Phase

The CAPO training algorithm operates in real-time, continuously learning from market feedback. Algorithm 1 presents the core training loop that classifies safe/toxic price levels and updates boundary estimates.

Algorithm 1 CAPO Real-Time Training

Require: Market snapshots $\{S_t\}$, bounds (B_t^b, B_t^a)
Ensure: Trained models $\mathrm{RF}_{\mathrm{bid}}, \mathrm{RF}_{\mathrm{ask}}$

Initialize $\mathcal{D}_{\mathrm{training}} \leftarrow \emptyset$, $\mathcal{B}_{\mathrm{bid}} \leftarrow \emptyset$, $\mathcal{B}_{\mathrm{ask}} \leftarrow \emptyset$
while new market data available **do**
    $S_t \leftarrow$ extract market snapshot
    if MarketViolatesBounds(S_t, B_t^b, B_t^a) **then**
        Skip ML processing, continue
    end if
    $(p_t^b, p_t^a) \leftarrow$ extract our current prices
    bid_safe $\leftarrow$ IsTopOfBook(p_t^b, S_t)
    ask_safe $\leftarrow$ IsTopOfBook(p_t^a, S_t)
    Update boundary estimates $\mathcal{B}_{\mathrm{bid}}, \mathcal{B}_{\mathrm{ask}}$
    $\mathbf{f}_t \leftarrow$ ExtractFeatures(S_t)
    if safe positions exist **then**
        $c \leftarrow$ CalculateConfidence$(p_t, t, \text{exploration_status})$
        $\mathcal{D}_{\mathrm{training}} \leftarrow \mathcal{D}_{\mathrm{training}} \cup \{(\mathbf{f}_t, p_t, c)\}$
    end if
    if every 30 snapshots **then**
        TrainModels$(\mathcal{D}_{\mathrm{training}})$
    end if
end while

3.2 Exploration Strategy

Algorithm 2 implements active boundary exploration with safety constraints to discover competitive pricing limits.

The safety check ensures minimum spread of 3δ and respects bounds constraints.

3.3 Boundary Estimation

Algorithm 3 updates competitive boundary estimates using confidence-weighted observations.

Toxic results receive higher confidence since they provide definitive bounds on competitor behavior.

Algorithm 2 Safe Boundary Exploration

Require: Market state S_t, exploration rate $r_{\text{explore}} = 0.15$
Ensure: Exploration target or normal pricing

$\xi \sim \text{Uniform}(0, 1)$
if $\xi < r_{\text{explore}}$ OR insufficient boundary data **then**
  for attempts = 1 to 10 **do**
    side $\leftarrow$ RandomChoice(["bid", "ask"])
    $k \sim \text{Uniform}(-5, 5)$
    $p_{\text{test}} \leftarrow S_t.\text{best\_price} + k \cdot \delta$
    if CheckSafety(p_{test}, side, S_t) **then**
      RETURN exploration at p_{test}
    end if
  end for
end if
RETURN normal ML pricing

Algorithm 3 Dynamic Boundary Updates

Require: Price p, safety result is_safe, side
Ensure: Updated boundary estimates $\mathcal{B}$

if is_safe **then**
  $\hat{b} \leftarrow p \pm 2\delta$ (conservative estimate)
  $c \leftarrow 0.4$ (lower confidence)
else
  $\hat{b} \leftarrow p \pm \delta$ (toxic boundary)
  $c \leftarrow 0.7$ (higher confidence for toxic results)
end if
$\mathcal{B} \leftarrow \mathcal{B} \cup \{(\hat{b}, c, t)\}$
Maintain sliding window of 50 recent estimates
boundary $\leftarrow$ WeightedAverage(recent estimates)

3.4 Price Enhancement Integration

Algorithm 4 integrates all components for live trading decisions.

Algorithm 4 CAPO Price Enhancement

Require: Base prices $(p^b_{\text{base}}, p^a_{\text{base}})$, bounds (B^b_t, B^a_t)
Ensure: Enhanced prices $(p^b_{\text{enh}}, p^a_{\text{enh}})$
Apply bounds: $p_{\text{enh}} \leftarrow \text{clamp}(p_{\text{base}}, \text{bounds})$
if market violates bounds **then**
  RETURN base prices
end if
if exploration triggered **then**
  RETURN exploration prices (if safe)
end if
if models trained **then**
  $\mathbf{f}_t \leftarrow \text{ExtractCurrentFeatures()}$
  $\hat{p} \leftarrow \text{RF}.\textit{predict}([\mathbf{f}_t]) \pm 2\delta$ (safety buffer)
  Use ML prediction if more aggressive than base
end if
RETURN enhanced prices

This integration ensures safe operation through bounds checking, careful exploration, and conservative safety buffers.

4 Experimental Setup

4.1 Data Description

Our experimental dataset consists of real-time order book snapshots from illiquid stablecoin markets on Kraken US, collected over a 6-month period from March 2024 to August 2024. The dataset includes 12 distinct stablecoin pairs with daily volumes ranging from \$50 to \$4,000 USD, representing typical illiquid market conditions where individual trades significantly impact market structure.

Each market snapshot captures the top 5 levels of the order book for both bid and ask sides, timestamped at millisecond precision. We identify 2-6 active market makers per symbol through order pattern analysis and price update frequency. The data includes periods of varying competitive intensity, from single-participant dominance to aggressive multi-party bidding wars.

Market characteristics vary significantly across our test universe. Lower volume pairs (under \$500 daily) typically see trades every 2-4 hours with spreads ranging from 40 to 150 basis points. Higher volume pairs (above \$2,000 daily) trade more frequently, approximately every 45-60 minutes, with tighter spreads of 20-80 basis points. All markets exhibit the characteristic pattern of periodic aggressive repricing when competitor spread thresholds are violated.

We filter out market open/close periods and major news events to focus on normal competitive dynamics. Snapshots with stale data (no updates for over 10 minutes) or obvious data errors are excluded. The final dataset contains approximately 2.4 million clean market snapshots across all symbols.

4.2 Baseline Comparisons

We evaluate CAPO against four baseline strategies representing different approaches to competitive market making:

Static Spread (SS): A traditional approach that maintains fixed spreads based on historical volatility, ignoring competitive dynamics. Spreads are set at the 75th percentile of historical levels to ensure profitability but remain constant throughout each trading session.

Reactive Matching (RM): A rule-based system that tightens spreads only after losing top-of-book position, then gradually widens them until regaining priority. This represents typical reactive market making without predictive capability.

Simple ML (SML): A basic machine learning approach using linear regression on market state features to predict optimal spreads. Unlike CAPO, this baseline lacks active exploration and uses only basic market features without boundary estimation.

Momentum-Based (MB): A technical analysis approach that adjusts spreads based on recent price momentum and competitor behavior patterns. This strategy attempts to predict competitor actions using simple moving averages and trend indicators.

Each baseline strategy is subject to the same bounds constraints and safety mechanisms as CAPO to ensure fair comparison. All systems operate with identical API rate limits and position size constraints.

4.3 Evaluation Metrics

4.3.1 Primary Metrics

Top-of-Book Win Rate: The percentage of time our bid and ask quotes achieve best price positioning. We measure this separately for bid and ask sides and compute a combined metric weighted by typical trade direction bias. This metric directly captures our primary objective of maximizing volume capture opportunity.

Spread Efficiency: The ratio of realized spreads to minimum viable spreads, measuring how well each strategy balances competitiveness with profitability. Values closer to 1.0 indicate optimal pricing that captures volume without sacrificing unnecessary profit.

Toxic Exposure Rate: The frequency of pricing at levels that immediately trigger competitive responses, measured as the percentage of repricing events followed by competitor aggression within 30 seconds. Lower values indicate better boundary prediction.

Profit Attribution: Daily P&L normalized by volume and volatility, isolating the contribution of pricing strategy from market conditions. We decompose returns into spread capture, adverse selection costs, and inventory effects.

4.3.2 Secondary Metrics

Exploration Efficiency: The ratio of successful boundary discoveries to total exploration attempts, measuring how effectively the active learning component identifies new competitive limits.

Model Accuracy: Prediction accuracy for safe/toxic classifications on held-out test data, updated daily as the model learns. We use both binary classification accuracy and regression RMSE for boundary predictions.

Adaptation Speed: The time required to detect and adapt to competitor strategy changes, measured as the lag between actual boundary shifts and model recognition of new levels.

API Efficiency: The number of quote updates per unit of top-of-book time achieved, measuring operational efficiency in API usage. Better strategies achieve more positioning with fewer updates.

4.4 Experimental Design

We conduct experiments using a walk-forward analysis framework that simulates realistic trading conditions while maintaining statistical rigor. The evaluation period is divided into 30-day segments, with each strategy independently operating on identical market conditions.

For each 30-day period, strategies are initialized with 7 days of historical data for model training and parameter estimation. Live performance is then measured over the remaining 23 trading days, with models updating in real-time as new data arrives. This approach ensures all strategies face identical market conditions while allowing for realistic learning and adaptation.

Risk management constraints are standardized across all strategies: maximum position sizes of \$500 per symbol, stop-loss triggers at 2% unrealized losses, and mandatory safety buffers maintaining minimum 15 basis point spreads regardless of competitive pressure.

Statistical significance is assessed using bootstrap resampling with 1,000 iterations, accounting for the time-series nature of financial data and potential autocorrelation in performance metrics. We report 95% confidence intervals for all primary metrics and conduct pairwise significance tests between strategies.

5 Results and Analysis

5.1 Performance Comparison

CAPO demonstrates substantial improvements over baseline strategies across all primary performance metrics. Table 1 summarizes the key results from our 6-month evaluation period.

The most significant improvement is in top-of-book positioning, where CAPO achieves a 74.86% win rate compared to 48.37% for our pre-CAPO baseline and 35.89% for static spread strategies. This represents a 54.7% relative improvement over our previous approach, directly addressing the core challenge of maximizing volume capture opportunity in illiquid markets.

Table 1: Performance Comparison Across Strategies

Metric	CAPO	Pre-CAPO	Static Spread
Top-of-Book Win Rate (%)	74.86	48.37	35.89
Spread Efficiency (%)	96.27	75.93	15.93
Model Accuracy (%)	88.83	–	–

Spread efficiency results demonstrate CAPO's ability to balance competitiveness with profitability. At 96.27% efficiency, CAPO operates very close to optimal pricing levels, capturing volume without sacrificing unnecessary spread. The dramatic difference between CAPO and static spread strategies (96.27% vs 15.93%) highlights the importance of dynamic competitive adaptation in these markets.

5.2 Active Learning Effectiveness

The exploration component of CAPO shows promising results despite the challenging nature of boundary discovery in competitive environments. Our exploration success rate of 24.58% indicates that roughly one in four boundary exploration attempts successfully identifies new competitive limits.

This success rate reflects the difficulty of the boundary discovery problem. Competitors frequently adjust their minimum spread thresholds, and many exploration attempts encounter prices that are either already known to be safe or prove to be toxic. However, the 24.58% success rate provides sufficient information for the system to maintain accurate boundary estimates and adapt to changing competitive dynamics.

The exploration strategy employs a conservative +2 tick safety buffer when testing new boundaries, prioritizing operational safety over aggressive discovery. This approach ensures that exploration activities do not compromise existing profitable positioning while still gathering valuable information about competitive limits.

5.3 Model Performance Analysis

CAPO's core classification model achieves 88.83% accuracy in predicting safe versus toxic price levels, demonstrating strong predictive capability despite the dynamic nature of competitor behavior. This accuracy level allows the system to make confident pricing decisions while maintaining appropriate safety margins.

The Random Forest architecture proves well-suited for this application, handling the non-linear relationships between market conditions and competitive responses. The model's ability to maintain high accuracy while adapting to changing competitor strategies validates our approach of continuous learning and real-time model updates.

Feature engineering plays a critical role in model performance. Our 12-feature set represents a refined selection from an original 30-feature universe, focusing on the most statistically significant predictors of competitive behavior. This feature reduction improves both computational efficiency and model interpretability while maintaining predictive accuracy.

5.4 Operational Robustness

CAPO demonstrates consistent performance across varying market conditions, showing particular strength in handling volatility changes. The system performs largely the same regardless of volatility regime, suggesting that the feature set and model architecture adequately capture the relationship between market dynamics and competitive behavior.

This robustness is particularly valuable in illiquid markets where volatility can shift rapidly and unpredictably. Traditional static approaches often struggle with volatility changes, either becoming too aggressive during calm periods or too conservative during volatile conditions. CAPO's adaptive framework automatically adjusts to these conditions through its real-time feature calculation and model updates.

The system's safety mechanisms prove effective in preventing adverse market impact during exploration and model learning phases. Bounds checking and minimum spread constraints successfully prevent the system from creating dangerous market conditions while still allowing for meaningful boundary discovery.

5.5 Limitations and Considerations

While CAPO shows strong performance improvements, several limitations merit discussion. The exploration success rate of 24.58%, while operationally useful, suggests significant room for improvement in boundary discovery efficiency. Future work could explore more sophisticated exploration strategies that better target likely boundary regions.

The system's performance is evaluated primarily in markets with 4 or fewer active competitors. Behavior in markets with higher competitive intensity remains an open question, though the scalability of the Random Forest approach suggests potential for extension to more crowded markets.

Model accuracy of 88.83%, while strong, indicates that approximately 11% of pricing decisions may be suboptimal. In high-frequency competitive environments, this error rate could accumulate to meaningful performance impact over time, suggesting continued focus on model improvement and feature engineering.

6 Discussion

6.1 Key Findings

Our results demonstrate that machine learning approaches significantly outperform traditional market making strategies in competitive illiquid markets. The 54.7% improvement in top-of-book positioning over our previous system validates the core hypothesis that competitive boundary prediction can be learned from market data and real-time feedback.

The effectiveness of active exploration, despite a 24.58% success rate, highlights the value of systematic boundary discovery in dynamic competitive environments. This finding challenges traditional market making wisdom that assumes static competitor behavior models. Our results show that competitors frequently adjust their minimum spread thresholds, and systems that fail to adapt to these changes suffer significant performance degradation.

The near-optimal spread efficiency of 96.27% demonstrates that CAPO successfully balances the competing objectives of volume capture and profitability. Traditional approaches often sacrifice one for the other, either maintaining wide spreads that miss volume opportunities or pricing aggressively and triggering costly bidding wars. CAPO's ability to operate close to the theoretical optimum represents a meaningful advance in competitive market making.

6.2 Practical Implications

CAPO's lightweight computational requirements make it accessible for implementation across diverse trading infrastructures. The system operates effectively on standard hardware without specialized computing resources, removing traditional barriers to sophisticated market making capabilities. This accessibility is particularly valuable for smaller trading firms seeking to compete effectively in illiquid markets.

Deployment through the SpreadWarden system requires only minutes to implement in new markets, enabling rapid scaling across multiple trading venues and asset classes. This deployment speed allows firms to quickly capitalize on new market opportunities without extensive development overhead.

The minimal maintenance requirements make CAPO particularly suited for illiquid markets where dedicated resources are often limited. Once deployed, the system requires little ongoing oversight, with performance monitoring serving as the primary operational requirement. This operational efficiency allows firms to manage larger numbers of illiquid markets with existing staff levels.

The system's zero additional capital requirements represent another practical advantage. Unlike strategies requiring specialized infrastructure or data feeds, CAPO operates on standard market data and existing trading systems, eliminating barriers to adoption.

6.3 Limitations and Future Work

The most significant limitation of the current implementation is the underlying programming language choice. A complete rewrite in Rust would deliver substantial performance improvements, particularly important as execution speed becomes increasingly critical in competitive environments. Rust's memory safety and concurrency features would also enhance system reliability in production trading environments.

Scalability represents another key limitation. While CAPO performs effectively in markets with 4 or fewer competitors, the approach may break down in more crowded markets with 10+ active participants. The complexity of competitive interactions likely grows non-linearly with participant count, potentially overwhelming the current feature set and model architecture.

The system's effectiveness is fundamentally limited to markets with large spreads where competitive positioning creates meaningful value. Applications to liquid markets with tight spreads would require entirely different approaches, as the assumptions underlying competitive boundary discovery become invalid when spreads approach minimum tick sizes.

Future development should focus on reinforcement learning approaches that can better handle the sequential decision-making nature of competitive market making. Unlike supervised learning approaches that treat each pricing decision independently, reinforcement learning could optimize for longer-term competitive positioning and profit maximization. This approach might also handle multi-competitor dynamics more effectively through learned strategy interactions.

Advanced exploration strategies represent another promising research direction. The current random exploration approach achieves useful results but likely leaves significant information gathering opportunities unexplored. More sophisticated exploration that targets likely boundary regions or exploits patterns in competitor behavior could substantially improve discovery efficiency.

Extension to portfolio-level optimization across multiple illiquid markets could capture diversification benefits and cross-market competitive effects. The current single-market focus may miss opportunities for coordinated positioning strategies that optimize overall portfolio performance rather than individual market metrics.

6.4 Broader Impact

CAPO demonstrates the practical value of machine learning in financial markets beyond the typical focus on price prediction. By framing market making as a competitive learning problem, we show how ML techniques can address operational challenges that traditional quantitative finance has struggled to solve systematically.

The success of active exploration in financial markets provides a template for other applications where learning system boundaries is critical. The approach of balancing exploration with operational safety constraints has broader applicability to automated trading systems facing unknown or changing market conditions.

Our results suggest that the future of market making in illiquid markets lies in adaptive systems that continuously learn and adjust to competitive dynamics. Static approaches that assume unchanging competitor behavior are fundamentally disadvantaged in environments where algorithms adapt and evolve in real-time.

7 Conclusion

We introduce CAPO (Competitive Aggressive Pricing Optimizer), a machine learning framework that addresses the fundamental challenge of competitive boundary discovery in illiquid market making. Our approach transforms the traditional market making problem from static spread optimization to dynamic competitive learning, where the system continuously adapts to evolving competitor strategies through active exploration and real-time model updates.

The empirical results demonstrate substantial performance improvements across all key metrics. CAPO achieves a 74.86% top-of-book win rate compared to 48.37% for our previous system, representing a 54.7% relative improvement in volume capture opportunity. The system operates at 96.27% spread efficiency, nearly optimal pricing that balances competitiveness with profitability. These results validate our core hypothesis that competitive boundaries can be learned and predicted from market data.

The active learning component proves essential for maintaining performance in dynamic competitive environments. Despite the challenging nature of boundary discovery, the 24.58%

exploration success rate provides sufficient information for the system to adapt to changing competitor strategies. The 88.83% classification accuracy demonstrates that machine learning models can effectively predict safe versus toxic pricing levels, even as competitor algorithms evolve.

Our contributions extend beyond the specific application to illiquid stablecoin markets. The framework demonstrates how machine learning can address operational challenges in competitive algorithmic trading environments where traditional quantitative approaches struggle. The combination of supervised learning for boundary prediction and active exploration for discovery provides a template for similar applications across financial markets.

The practical implementation advantages make CAPO accessible for widespread adoption. The system operates on standard hardware with minimal computational requirements, deploys in minutes through existing trading infrastructure, and requires little ongoing maintenance. These characteristics remove traditional barriers to sophisticated market making capabilities, particularly for smaller trading firms competing in illiquid markets.

Future research should focus on reinforcement learning approaches that can better handle the sequential nature of competitive interactions and scale to markets with larger numbers of participants. The current framework provides a foundation for these advanced techniques while demonstrating immediate practical value in production trading environments.

CAPO represents a fundamental shift from reactive to predictive competitive market making, showing how machine learning can transform operational trading challenges into systematic advantages. As algorithmic trading continues to evolve, adaptive systems that learn and respond to competitive dynamics will become essential for maintaining effective market making operations in increasingly sophisticated trading environments.

A Feature Definitions

This section provides complete mathematical definitions for all 12 features used in CAPO's machine learning models.

A.1 Market State Features (f1-f6)

Current Best Prices:

$$f_1 = b_t^{(1)} \quad \text{(current best competitor bid)} \tag{1}$$

$$f_2 = a_t^{(1)} \quad \text{(current best competitor ask)} \tag{2}$$

Spread Metrics:

$$f_3 = a_t^{(1)} - b_t^{(1)} \quad \text{(current spread)} \tag{3}$$

$$f_4 = \frac{1}{w} \sum_{i=t-w+1}^{t} (a_i^{(1)} - b_i^{(1)}) \quad \text{(average spread, } w = 10\text{)} \tag{4}$$

Volatility Measures:

$$f_5 = \frac{1}{w} \sum_{i=t-w+1}^{t} |b_i^{(1)} - b_{i-1}^{(1)}| \quad \text{(mean price volatility)} \tag{5}$$

$$f_6 = \sqrt{\frac{1}{w-1} \sum_{i=t-w+1}^{t} (|b_i^{(1)} - b_{i-1}^{(1)}| - f_5)^2} \quad \text{(volatility of volatility)} \tag{6}$$

A.2 Performance Features (f7-f8)

Historical Win Rates:

$$f_7 = \frac{\sum_{i=t-h+1}^{t} \mathbf{1}_{\text{bid top}}(i)}{h} \quad \text{(bid win rate over } h = 100 \text{ snapshots)} \tag{7}$$

$$f_8 = \frac{\sum_{i=t-h+1}^{t} \mathbf{1}_{\text{ask top}}(i)}{h} \quad \text{(ask win rate over } h = 100 \text{ snapshots)} \tag{8}$$

where $\mathbf{1}_{\text{bid top}}(i) = 1$ if our bid was top-of-book at snapshot i, 0 otherwise.

A.3 Boundary Estimation Features (f9-f10)

Learned Boundary Estimates:

$$f_9 = \frac{\sum_{j \in \mathcal{B}_{\text{bid}}} c_j \cdot \hat{b}_j}{\sum_{j \in \mathcal{B}_{\text{bid}}} c_j} \quad \text{(confidence-weighted bid boundary)} \tag{9}$$

$$f_{10} = \frac{\sum_{j \in \mathcal{B}_{\text{ask}}} c_j \cdot \hat{a}_j}{\sum_{j \in \mathcal{B}_{\text{ask}}} c_j} \quad \text{(confidence-weighted ask boundary)} \tag{10}$$

where $\mathcal{B}_{\text{bid/ask}}$ contains the 50 most recent boundary estimates, $\hat{b}_j, \hat{a}_j$ are estimated boundaries, and c_j are confidence weights.

A.4 Learning Quality Features (f11-f12)

Exploration and Confidence Metrics:

$$f_{11} = \frac{\text{Successful explorations in last 50 attempts}}{\text{Total explorations in last 50 attempts}} \tag{11}$$

$$f_{12} = \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} c_j \quad \text{(average confidence of recent boundaries)} \tag{12}$$

B Algorithm Parameters

B.1 Random Forest Configuration

Both bid and ask models use identical hyperparameters:

$$\begin{aligned}
\text{n_estimators} &= 50 && (13)\\
\text{max_depth} &= 12 && (14)\\
\text{min_samples_split} &= 3 && (15)\\
\text{min_samples_leaf} &= 2 && (16)\\
\text{random_state} &= 42 && (17)
\end{aligned}$$

B.2 Exploration Parameters

$$\begin{aligned}
r_{\text{explore}} &= 0.15 \quad \text{(base exploration rate)} && (18)\\
k_{\text{range}} &= [-5, 5] \quad \text{(tick offset range for exploration)} && (19)\\
\tau_{\min} &= 3 \quad \text{(minimum snapshots between explorations)} && (20)\\
s_{\min} &= 3\delta \quad \text{(minimum spread for safety)} && (21)
\end{aligned}$$

B.3 Confidence Weighting

Confidence scores are calculated as:

$$\begin{aligned}
c_i &= c_{\text{base}} + c_{\text{exploration}} + c_{\text{boundary}} + c_{\text{recency}} && (22)\\
c_{\text{base}} &= 0.5 && (23)\\
c_{\text{exploration}} &= \begin{cases} 0.4 & \text{if sample from exploration} \\ 0 & \text{otherwise} \end{cases} && (24)\\
c_{\text{boundary}} &= \begin{cases} 0.2 & \text{if within 3 ticks of boundary} \\ 0 & \text{otherwise} \end{cases} && (25)\\
c_{\text{recency}} &= 0.1 \cdot e^{-\lambda(t_{\text{current}} - t_i)} \quad \text{with } \lambda = 0.01 && (26)
\end{aligned}$$

B.4 Safety Constraints

$$\begin{aligned}
\text{Safety Buffer} &= 2\delta \quad \text{(added to ML predictions)} && (27)\\
\text{Minimum Window} &= w = 10 \quad \text{(snapshots for volatility)} && (28)\\
\text{Boundary Window} &= 50 \quad \text{(recent boundaries stored)} && (29)\\
\text{Training Window} &= 1000 \quad \text{(maximum training samples)} && (30)
\end{aligned}$$

C Implementation Details

C.1 Data Processing

Market snapshots are processed with the following filters:

- Remove snapshots with stale data (no updates $>$ 10 minutes)
- Filter out obvious data errors (prices outside 3σ bounds)
- Exclude market open/close periods (first/last 30 minutes)
- Require minimum 5 levels of book depth for competitor identification

C.2 Model Training Schedule

- Initial training: 7 days of historical data
- Retraining frequency: Every 30 market snapshots
- Maximum training samples: 1000 (sliding window)
- Minimum training samples: 100 (required before ML activation)

C.3 Performance Monitoring

Key metrics tracked in real-time:

- Classification accuracy (updated every 100 snapshots)
- Boundary estimation error (RMSE vs realized boundaries)
- Exploration success rate (rolling 50-attempt window)
- API usage efficiency (quotes per unit top-of-book time)