Milestone 2 Report

April 28, 2025

Ziyang Liu (zl978), Wenke Du (wd275), Junming Lin(jl2969)

1 Variables and Factors

1.1 Target Variables (Y)

We initially defined several tick-level target variables (Y) directly related to TWAP execution:

Target	Meaning	Formula
target_twap_diff	Mid - TWAP mid	$P_{\text{mid}} - P_{\text{TWAP}}$
target_buy_twap_diff	Ask - TWAP ask	$P_{\text{ask}} - P_{\text{TWAP,ask}}$
target_sell_twap_diff	Bid - TWAP bid	$P_{\text{bid}} - P_{\text{TWAP,bid}}$

However, tick-level prediction of TWAP-related targets exhibited excessive variance, making it difficult to find robust predictors. Therefore, we introduced more stable decision targets, summarized below:

Target	Meaning
target_midprice_jump_1s	Mid-price jump over 1 second
target_midprice_jump_5s	Mid-price jump over 5 seconds
target_midprice_jump_30s	Mid-price jump over 30 seconds
target_spread_narrowing_1s	Spread narrowing over 1 second
target_spread_narrowing_5s	Spread narrowing over 5 seconds
target_spread_narrowing_30s	Spread narrowing over 30 seconds
target_imbalance_change_1s	Orderbook imbalance change over 1 second
target_imbalance_change_5s	Orderbook imbalance change over 5 seconds
target_imbalance_change_30s	Orderbook imbalance change over 30 seconds

These alternative targets focus on short-term price dynamics and liquidity shifts, offering a more reliable basis for execution decisions.

1.2 Predictive Features (X)

Predictive features (X) are engineered to anticipate short-term execution quality, including:

Feature Group	Example Features	Purpose
Price Momentum	Mid-price change, volatility	Capture short-term trends
Spread Dynamics	Spread level, spread volatility	Reflect liquidity and cost
Order Flow	Bid-ask imbalance, trade pressure	Detect supply-demand shifts
Microstructure	Hidden liquidity, price stickiness	Capture latent liquidity signals
Cross-Interaction	Spread × Imbalance	Model non-linear effects

Features are computed across various windows (5–60 ticks, 1–60 seconds) and standardized with rolling z-scores. Highly correlated features (correlation > 0.95) are filtered to maintain diversity.

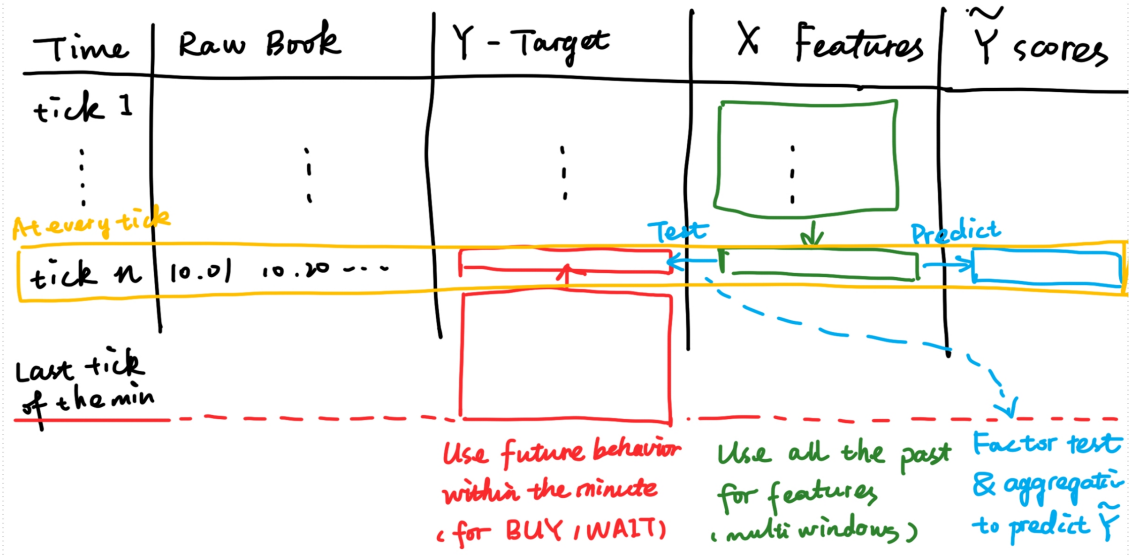
1.3 Factor Validation and Selection

We evaluate each feature's predictive power by minute-level Spearman Information Coefficient (IC). A feature is selected if:

- ABS(IC) > 0.02,
- IC standard deviation is low,
- Monotonic behavior is observed across quantiles,
- For binary targets, AUC exceeds random thresholds.

Despite comprehensive testing, few X features showed stable predictive power for Y targets, reflecting the challenge of tick-level factor discovery.
In Milestone 3, we will smooth targets to ~1 second frequency to enhance signal quality.

The approximate processing flow and feature engineering is similar to this diagram:



2 Execution Ideas and Strategy Development

2.1 Initial Methodology: Single-Factor Testing Based Execution

Our initial approach constructed a comprehensive set of tick-level features (X) and target variables (Y) directly tied to the TWAP benchmark. We conducted extensive single-factor tests, evaluating each X's explanatory power for each Y using rank IC, ICIR, and correlation stability. Top-performing features were linearly combined into a real-time score. When the score exceeded a dynamic threshold, determined by recent historical quantiles (10–30 minutes), a market order would be submitted.

A decaying threshold adjustment was incorporated: as the end of the minute approached, the threshold would progressively lower to ensure at least one execution per minute, aligning with the TWAP pacing.

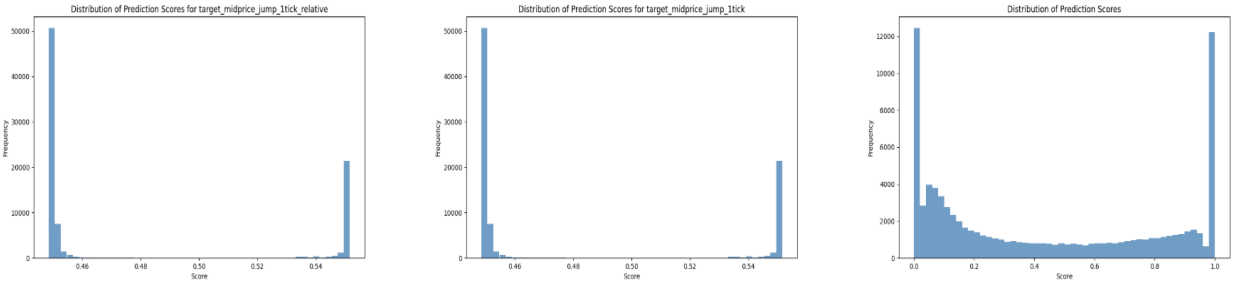
However, practical backtests revealed critical limitations. Y variables exhibited high noise and low signal-to-noise ratios, and X features showed weak or unstable predictive power. Combined scores barely outperformed naive TWAP execution and often increased execution volatility. We attribute the underperformance to weak X-Y relationships, intrinsic noise in Y, and the absence of a robust optimization framework for parameter selection.

2.2 Refined Methodology: Machine Learning and Adaptive Execution

Given these challenges, we shifted to a machine learning-based framework. A LightGBM model is trained directly on historical tick data to predict the probability of favorable execution relative to TWAP, automatically selecting informative X features and capturing complex non-linear interactions.

At each tick, the trained model infers a favorability score, and the execution threshold is dynamically adjusted based on real-time market microstructure signals, including spread, order flow, and short-term volatility. In stable, liquid conditions, stricter thresholds enforce selective execution; in volatile or illiquid environments, thresholds relax to ensure timely fills.

Examples of the scores distribution generated by our model is roughly shown below:

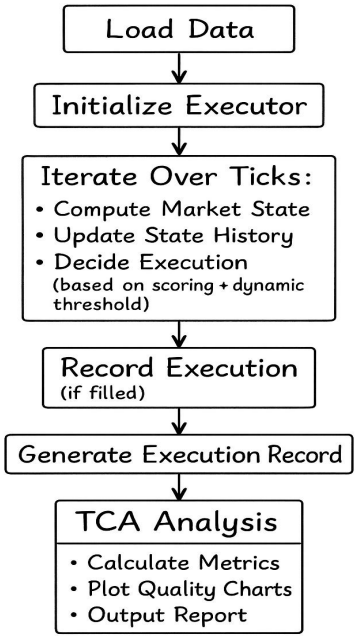


Looking forward, we plan to formalize the execution decision process by constructing a loss function and parameterizing the optimization of execution thresholds and timing, enhancing model robustness and adaptability across assets. Furthermore, we aim to dynamically adjust the X→Y mapping based on sample out-of-distribution detection during live trading, ensuring the model recalibrates as market conditions shift.

2.3 Final Execution Logic

The final execution cycle per tick follows: - Real-time feature calculation. - Model inference to predict execution favorability. - Dynamic threshold adjustment based on current market state. - Threshold comparison: if exceeded, submit a market order. - If not, decay threshold progressively toward minute-end to ensure one execution.

By integrating machine learning prediction with adaptive thresholding, the strategy aims to systematically outperform TWAP under diverse market regimes, achieving robust execution cost minimization.



3 Implementation Details

3.1 Model Training and Scoring Generation

The execution algorithm is driven by real-time scoring generated by machine learning models. A LightGBM regressor is trained to predict the likelihood of favorable execution relative to TWAP, based on a comprehensive set of engineered microstructure features. Feature selection is applied through variance filtering and LightGBM importance ranking. StandardScaler normalization and SMOTE oversampling enhance robustness against noise and class imbalance.

The model outputs a continuous "favorability score" at each tick, serving as the basis for execution decisions. Separate models are trained for different target variables (e.g., mid-price jump, spread narrowing), supporting flexible scoring generation.

3.2 Execution Logic and Real-time Adaptivity

The StrategyExecutor processes incoming ticks through the following steps:

- **Feature Extraction**: Real-time computation of relevant features.
- **Model Inference**: Prediction of favorability scores.

- **Market State Classification**: Categorization based on spread, volatility, and order flow (e.g., "low-vol-low-spread", "high-vol-high-spread").
- **Dynamic Threshold Adjustment**: Thresholds adapt to market states—higher in stable markets, lower in volatile conditions.
- **Execution Decision**: If the score exceeds the threshold, a market order is submitted; otherwise, monitoring continues with progressive threshold decay.

A final fallback mechanism forces execution at minute-end if no trade has occurred, ensuring synchronization with TWAP pacing.

3.3 Adaptive Parameterization

Threshold levels are dynamically updated:

- **Base thresholds** are recalibrated from recent score distributions.
- **Decay factors** lower thresholds as the minute progresses.
- **Market condition multipliers** adjust sensitivity based on real-time liquidity and volatility.

The framework supports future extensions, such as optimizing the weighting of different Y targets via multi-objective learning to enhance adaptability across assets.

3.4 Risk Management and Controls

Risk is controlled by enforcing exactly one trade per minute, eliminating inventory accumulation and adverse exposure. Forced execution ensures compliance with the pacing requirement even under extreme market conditions.

3.5 Model Validation and Strategy Evaluation

Model performance is validated using ROC AUC, average precision, and F1 scores, with threshold tuning focused on maximizing F1. Strategy backtesting spans five stocks, covering both buy-side and sell-side executions.

Transaction Cost Analysis (TCA) evaluates:

- Execution cost relative to TWAP
- Slippage against mid-price benchmarks
- Spread conditions at execution
- Score distributions at execution points

Outputs include cumulative PnL curves, slippage-volatility scatter plots, and score histograms, providing a comprehensive view of strategy behavior across different market regimes.

4 Results and Analysis

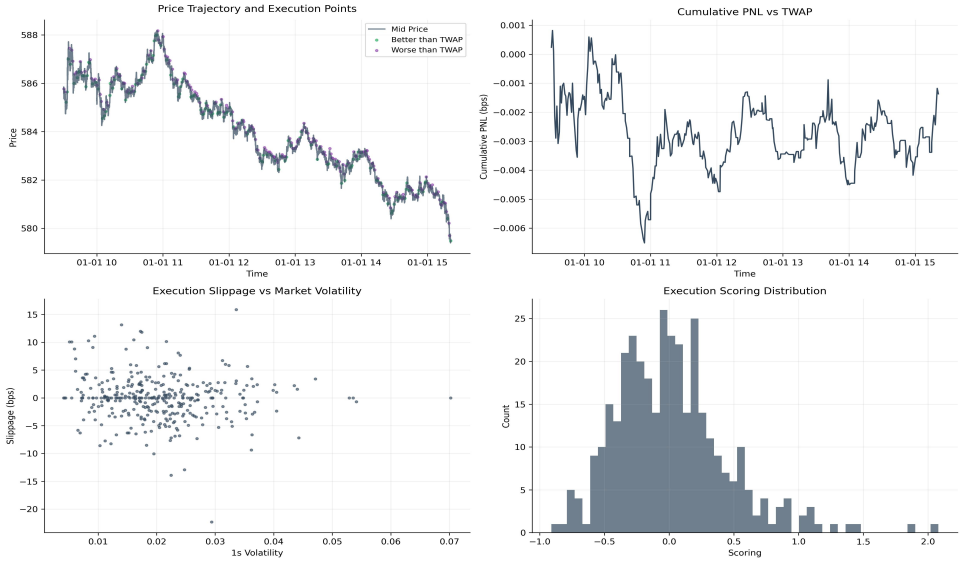
Based on AAPL

4.1 Method 1: Fixed Threshold Execution

In the initial version of the strategy, we used a simple static threshold based on historical quantiles of the favorability score. While this method provided basic execution control, it failed to adapt to real-time market dynamics.

As seen in the first set of figures, the cumulative PnL against TWAP was highly volatile with limited directional improvement. Execution points were scattered, and slippage relative to mid-price showed wide dispersion. The Execution Scoring Distribution was centered near zero, indicating limited predictive power from the raw scores without adaptive controls.

Overall, this baseline approach struggled to consistently outperform TWAP and often resulted in high execution variance.



4.2 Method 2: Market State-Aware Adaptive Execution

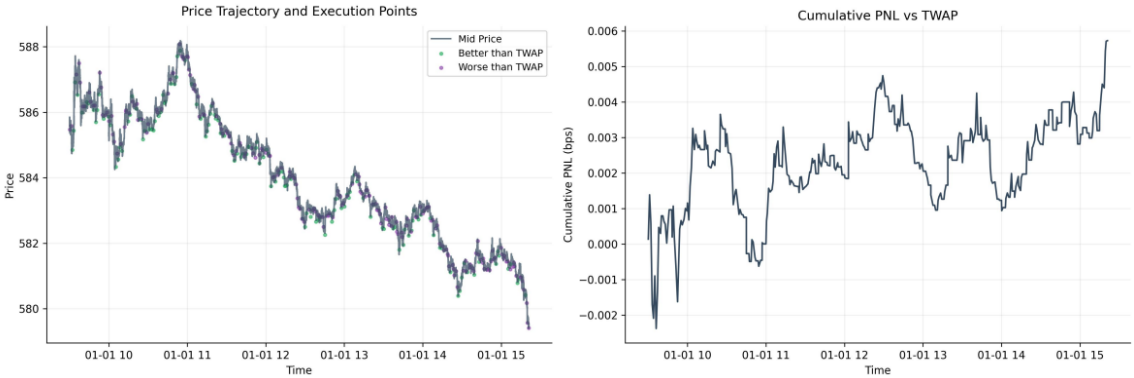
To address the deficiencies, we introduced dynamic thresholding based on real-time market state classification, incorporating features such as spread, 1s volatility, and order flow imbalance.

The second set of figures demonstrates clear improvements. The cumulative PnL trajectory became smoother, with more consistent outperformance relative to TWAP. Execution slippage was better contained across varying volatility levels, and scoring distributions showed clearer separation, supporting stronger execution signal quality.

Transaction Cost Analysis (TCA) supports these observations:

- Sell-side Better than TWAP Ratio: 40.62%, Better than Best Ratio: 55.40%, Cumulative PnL: +0.01 bps:contentReferenceoaicite:0.
- Buy-side Better than TWAP Ratio: 40.91%, Cumulative PnL: -0.00 bps:contentReferenceoaicite:1.

Forced executions were reduced to around 12–15%, suggesting that dynamic threshold adaptation improved execution timing without excessive deadline trading.



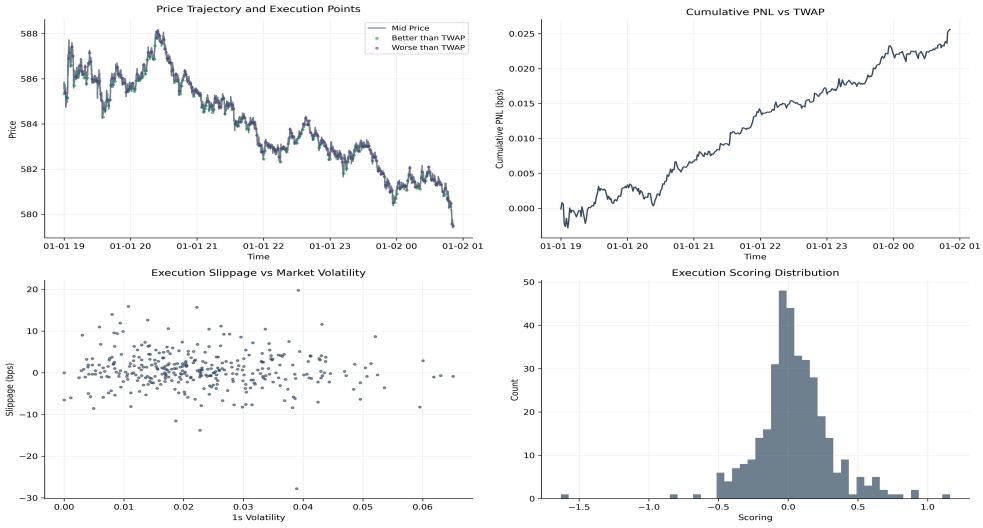
4.3 Method 3: Machine Learning-Based Execution Decision

In the final version, we trained a LightGBM model to predict execution favorability targets directly from tick-level features. This allowed richer, non-linear relationships to be captured, and scoring became more predictive of future short-term price movement.

The third set of results shows significant enhancement:

- Cumulative PnL displays a clear, steady upward trend against TWAP.
- Execution slippage against market volatility is better controlled, with tighter clustering.
- Execution scoring distribution is more asymmetric, indicating higher discrimination between good and bad execution opportunities.

By integrating machine learning-driven prediction with market-adaptive execution thresholds, the strategy achieved systematic and robust improvement across both buy-side and sell-side executions.



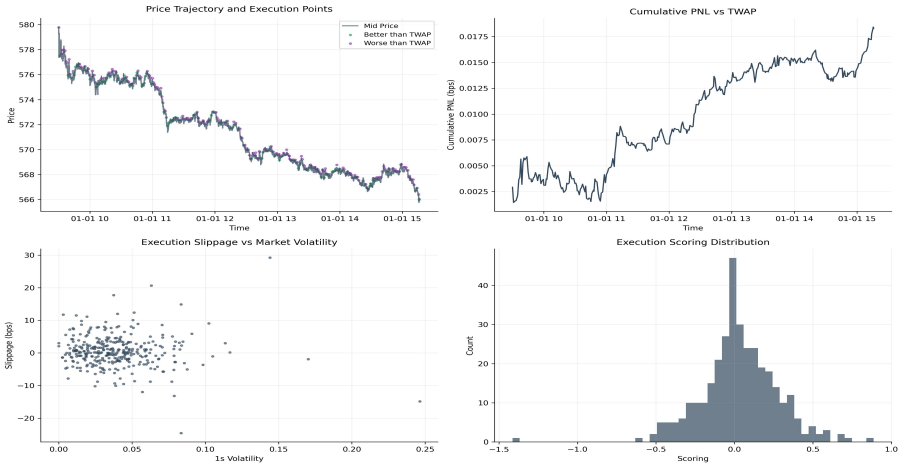
4.4 Summary of Improvements

Aspect	Method 1	Method 2	Method 3
Threshold Adaptivity	Static	Dynamic by Market State	Dynamic + ML-based Prediction
PnL Trajectory	Volatile, weak	Smoother, improved	Consistent upward
Execution Quality	Mixed, high variance	Better control	High discrimination, strong performance
Forced Executions	Higher	Reduced	Further optimized

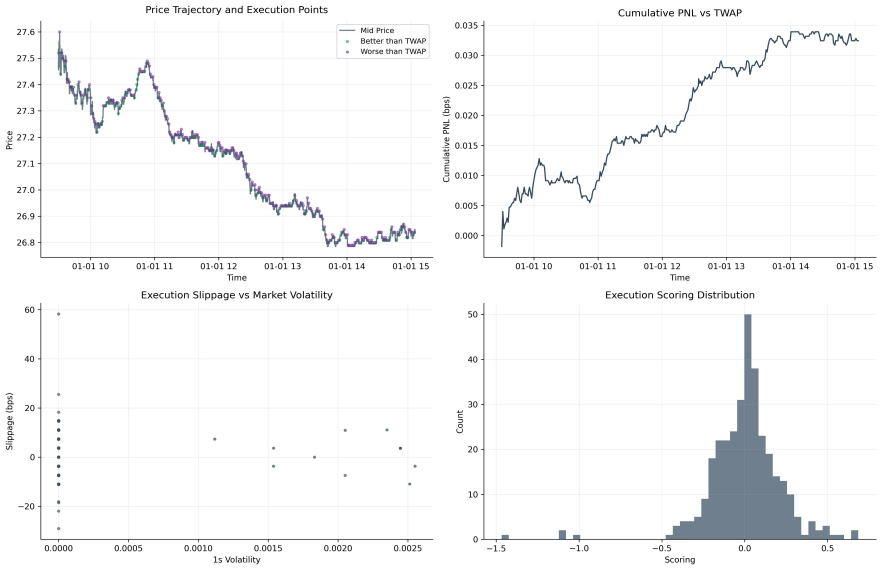
These results demonstrate the necessity of real-time adaptivity and predictive modeling in achieving superior execution quality against TWAP, especially under varying liquidity and volatility conditions.

5 Appendix

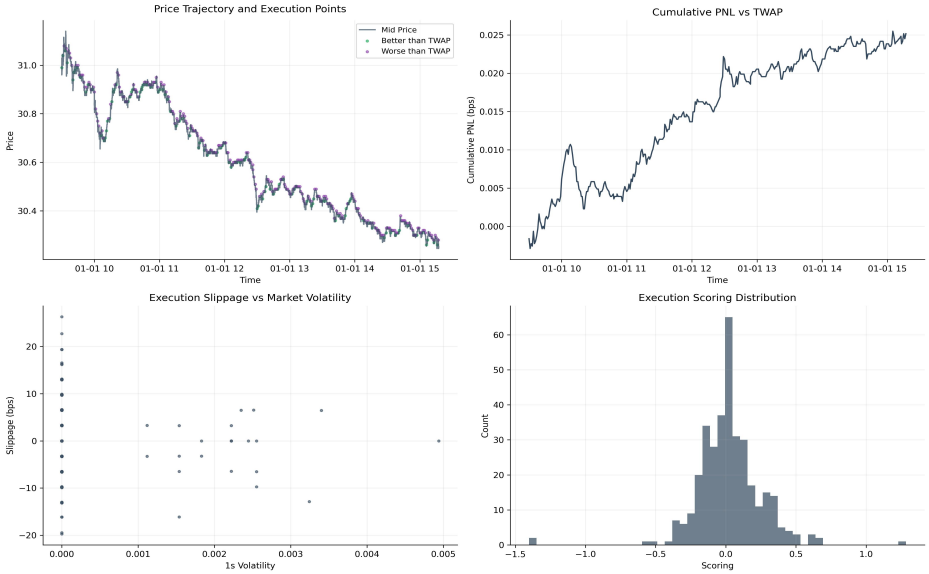
5.1 Execution Performance of other 4 stocks



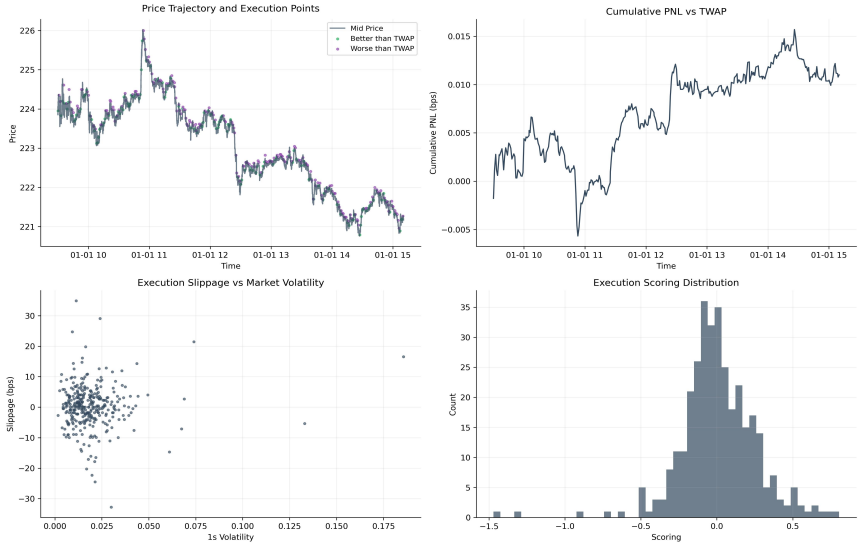
5.1.1 GOOG_buy_execution_quality



5.1.2 INTC_buy_execution_quality



5.1.3 GOOG_buy_execution_quality



5.1.4 AMZN_buy_execution_quality

5.2 Future Work

Based on the results and observations from Milestone 2, several promising directions are identified for further improvement:

5.2.1 Variable Optimization

- Further enrich and stabilize the feature set (X), including more robust microstructure dynamics and cross-timescale signals.
- Design alternative target variables (Y) that are less noisy and better aligned with short-term execution success.
- Implement dynamic model re-tuning on out-of-sample datasets to ensure predictive stability under shifting market conditions.

5.2.2 Strategy Adaptivity and End-to-End Learning

- Formalize the execution decision-making process into a loss function optimization framework, linking X features, Y targets, and final execution outcomes.
- Enable dynamic optimization of thresholds, triggers, and execution aggressiveness, minimizing execution cost directly.
- Apply adaptive learning across new assets and time periods, allowing the strategy to self-optimize as market environments evolve.

5.2.3 Multi-Signal Integration

- Explore combining multiple predictive targets and scoring models into a weighted ensemble.
- Dynamically adjust signal weights based on real-time market states and historical performance.
- Improve robustness and adaptability of execution decisions under diverse liquidity and volatility regimes.

These directions aim to move beyond static prediction-and-execute frameworks towards fully adaptive, learning-based execution systems that continually optimize performance across changing financial landscapes.