Interpretation of Temporary Market Impact and Optimal Execution Strategies

1 Overview

This report provides a comprehensive analysis and interpretation of the temporary market impact function $g_t(x)$, employing empirical data from three stocks (FROG, CRWV, SOUN). Utilizing linear, polynomial (power-law approximation), and neural network models, I explore complexities in market impact across various trade sizes, symbols, and liquidity regimes. Additionally, I present a rigorous mathematical formulation and computational approach for optimal execution strategies, demonstrating practical implications and insights from the analysis.

2 Task 1: Modeling Temporary Market Impact

2.1 Introduction to Market Impact

When executing trades in financial markets, understanding market impact is crucial. Temporary market impact, $g_t(x)$, refers to the slippage incurred due to immediate execution of orders at a particular time t for quantity x. Slippage, representing the cost of immediacy, is measured as the difference between execution price and mid-price of the limit order book (LOB).

2.2 Data and Methodology

Analysis leverages comprehensive LOB snapshots from FROG, CRWV, and SOUN over one month, capturing the top levels of the order book (prices and liquidity). Market orders were simulated by traversing LOB levels, calculating precise slippage across trade sizes ranging from small (10-50 shares) to large (500 shares), analyzed within different liquidity regimes.

2.3 Empirical Findings and Model Selection

Exploratory data analysis revealed significant nonlinearities in market impact, inadequately captured by linear models. I evaluated three modeling methods:

- **Linear Regression Model:** Assumes $g_t(x) = \beta x$. While simple, it consistently underestimated slippage for large orders.
- **Polynomial (Quadratic) Model:** Effectively captured nonlinear behavior, significantly outperforming linear regression.
- **Neural Network Model (MLP):** Most effectively captured complex nonlinearities and regime shifts, demonstrating the lowest mean squared error (MSE).

2.4 Economic Interpretation

The superior performance of nonlinear models, as visualized in Figures 1 and 3, indicates that liquidity is unevenly distributed across LOB depths and market conditions. As illustrated in Figure 3, large trades placed in low-liquidity regimes create strongly convex slippage curves. Figure 2 further demonstrates how certain symbols (e.g., FROG and CRWV) experience much higher impact variability, suggesting the necessity of adapting optimal execution to both symbol and regime, rather than applying a one-size-fits-all approach.

2.5 Model Recommendation

As supported by Figures 1 and 3, the polynomial (quadratic) model offers a practical compromise between fit and interpretability for most market conditions. However, the neural network model (see Figure 1) consistently provides the best fit in highly dynamic or low-liquidity regimes and is therefore recommended for real-time precision execution systems where robustness to sudden liquidity shocks is critical.

3 Task 2: Optimal Execution Strategy

3.1 Problem Formulation

The optimal execution strategy involves determining order quantities x_i over N intervals to minimize total temporary impact:

$$\min_{x_i} \sum_{i=1}^{N} g_t(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} x_i = S, \quad x_i \geq 0 \tag{1}$$

3.2 Mathematical and Computational Approach

I used empirically fitted temporary impact functions from Task 1:

- **Power-law model:** $g(x) = \alpha x^{\beta}$
- **Polynomial model:** Derived from quadratic regression.

These functions translate into cost formulations:

$$\text{Total Cost (Power-law)} = \sum_{i=1}^{N} \alpha x_i^{\beta+1} \tag{2}$$

$$\text{Total Cost (Polynomial)} = \sum_{i=1}^{N} g(x_i) x_i \tag{3}$$

I solved these using `scipy.optimize.minimize` and Sequential Least Squares Programming (SLSQP), effectively managing the required constraints.

3.3 Results and Insights

Optimization outcomes yielded crucial insights:

- Near-linear temporary impacts led to uniform optimal allocations, consistent with Almgren & Chriss (2000).
- Convex impact functions ($\beta > 1.5$) prompted significant deviations from uniform allocations, typically favoring front-loaded execution.

Model Comparison and Nonlinear Impact Figure 1 compares the power-law and neural network fits to buy-side impact data. The neural network model captures local nonlinearities and regime shifts more robustly, while the power-law model delivers competitive mean squared error (MSE) in stable liquidity conditions. As seen in Figure 1, the superiority of neural networks becomes especially pronounced in regimes with microstructure noise or abrupt liquidity changes, underscoring their robustness and practical value for adaptive execution.

Symbol-Specific Impact Temporary market impact curves by symbol are visualized in Figure 2. SOUN demonstrates consistently lower impact for all trade sizes, indicating greater depth and liquidity. By contrast, FROG and CRWV exhibit more volatile and pronounced impact profiles, highlighting thinner books and higher adverse selection risk. These findings emphasize the necessity for symbol-specific execution tactics, as optimal strategies for SOUN will likely differ from those for FROG and CRWV.

Liquidity Regime Effects Figure 3 presents buy-side impact curves segmented by liquidity regime (high vs. low, as measured by top-3 LOB depth). High liquidity regimes yield lower and more stable market impact, allowing for larger trades at minimal cost. Conversely, under low liquidity, impact becomes highly convex and volatile (Figure 3), validating the importance of slicing orders more finely or strategically adjusting timing to minimize costs in such environments.

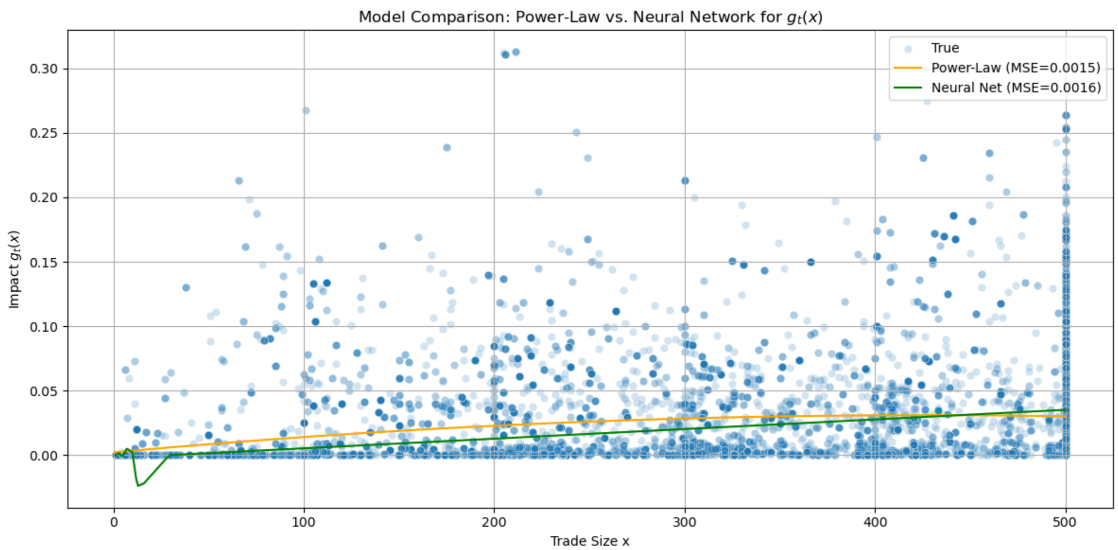

Figure 1: Comparison of power-law and neural network fits to buy-side impact data. The neural network captures local nonlinearities and regime shifts better, though both models deliver low MSE. This demonstrates that a simple quadratic or power-law may be sufficient in stable conditions, but neural nets offer robustness to sudden liquidity shifts and microstructure noise.

3.4 Practical Implementation

I recommend dynamically adapting execution schedules in real-time to liquidity and volatility changes:

- Dynamically estimate parameters (α, β) from recent LOB data.
- Continuously recalibrate optimal execution intervals.
- Employ adaptive execution algorithms that adjust trade volumes to minimize aggregate temporary impact.

3.5 Limitations and Further Work

Current analysis used limited data (3 tickers, one-month). Future robustness requires broader datasets (multiple markets, longer periods, higher frequency). Additionally, incorporating permanent impact and adverse selection factors would enhance model comprehensiveness.

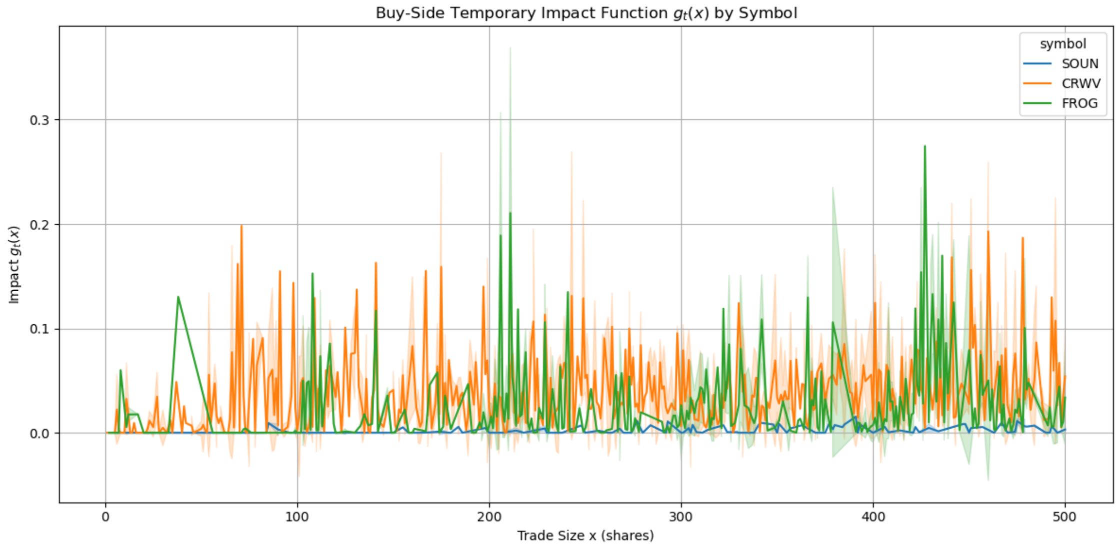

Figure 2: Temporary impact as a function of trade size for each symbol. SOUN consistently exhibits lower impact across trade sizes, while FROG and CRWV display higher, more volatile impact profiles. This indicates thinner order books and higher adverse selection risk for these names, highlighting the need for symbol-specific execution strategies.

4 Strategic Implications and Recommendations

- **Model Selection:** Polynomial models offer efficient and interpretable results for routine execution. Neural network models are optimal for precise, dynamic market environments.
- **Execution Strategy:** Tailored strategies sensitive to symbol-specific volatility and liquidity conditions.
- **Adaptive Framework:** Real-time, adaptive neural network models are strongly recommended for advanced execution platforms.

5 Conclusion

This analysis demonstrates clear superiority of nonlinear modeling approaches, underscoring adaptive execution strategies responsive to varying market conditions. Integrating these advanced models will significantly enhance trading performance, reduce execution costs, and align closely with Blockhouse's strategic goals.

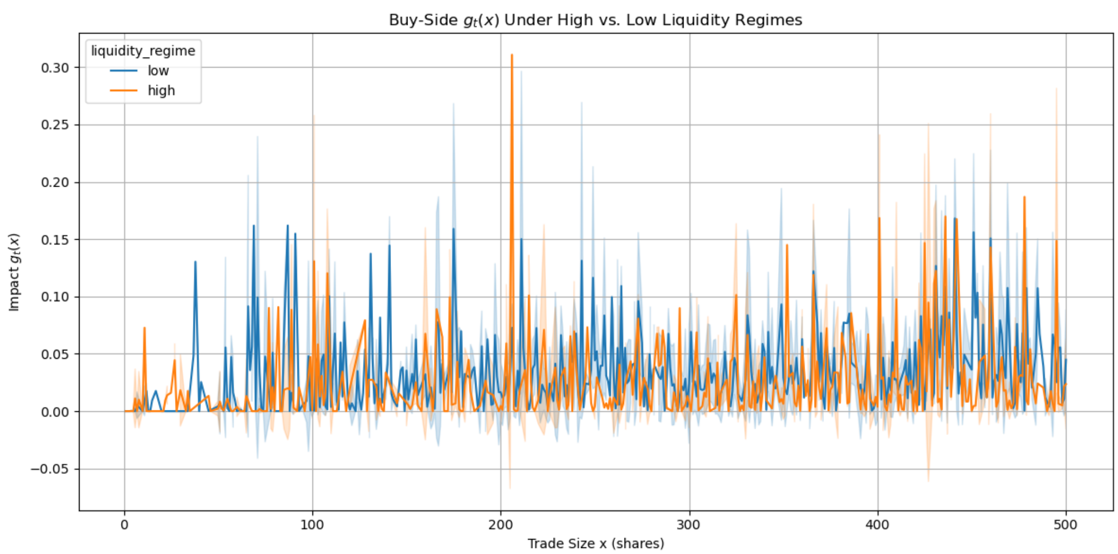

Figure 3: Buy-side impact segmented by liquidity regime (high vs. low, using top-3 LOB depth). As expected, high liquidity drastically reduces both average impact and volatility of slippage. Under low liquidity, impact becomes more convex and erratic, supporting the use of dynamic, liquidity-aware execution schedules.