Deep Hedging the Volume-Weighted Average Price Risk in Order-Driven Markets

Master Thesis
Master of Science UZH ETH in Quantitative Finance

Michael Geiser
Student number: 22-738-645

Advisor: Prof. Dr. Josef Teichmann
Department of Mathematics, ETH Zurich

December 10, 2025

Contents

Hedging Problem

Problem

Buying Q_0 units of a certain asset S over the time interval $[0, T]$ (one trading day), where our average purchase price is compared to the market **volume-weighted average price** (VWAP) during the same time interval. We focus on the buy side; the sell case is symmetric.

The asset trades in a **limit order book** (LOB) market, where prices and volumes emerge from two order types and a matching mechanism:

- **Limit order** (LO): instruction to buy or sell a specific quantity at a specific price, which executes only if a counterparty agrees to trade at that price.
- **Market order** (MO): instruction to buy or sell a specific quantity immediately at the best available prices.
- **Price-time priority rule:** MOs are matched with available LOs taking better prices first, and within a price level, earlier orders.

No market impact: we do not interact with the market. Instead, we receive MO prices and LO fills based on exogenous rules derived from the current LOB state and its evolution (e.g., we trade through a liquidity provider).

Same-side execution: no intraday round-trip trading (e.g., selling during a buy program).

Goal

Building a trading policy (i.e., a program to buy in the market via MOs and LOs) to hedge the downside risk of our wealth (i.e., minimize potential large losses).

Hedging Problem

We present a simple example to show the relevance of the trading policy in the shape of the wealth distribution.

Terminal wealth:

$$W^{\delta} = Q_0(\mathrm{VWAP} - P^{\delta}), \tag{1}$$

where P is our average purchase price, which depends on the trading policy δ.

Setup: we use the Xiu & Palomar (2023) decomposition of price volatility and traded volume

$$\sigma_{t,n} = d_t \times s_n \times I_{t,n} \times u_{t,n} \times \varepsilon_{t,n}, \qquad V_{t,n} = d_t^* \times s_n^* \times I_{t,n} \times \varepsilon_{t,n}^*. \tag{2}$$

Parameter	Meaning	Value for simulations
$d_t,\ d_t^*$	Daily volatility / volume level	$d_t = 0.1,\ d_t^* = 1000$
$s_n,\ s_n^*$	Intraday seasonal patterns	$s_n = 1,\ \ \log s_n^* = 3(\frac{n}{N} - 0.5)^2 - \bar{s}$
$I_{t,n}$	Shared information-flow factor	$\log I_{t,n+1} = \phi_I \log I_{t,n} + \sigma_I Z_{t,n+1},\ \ \phi_I = 0.95,\ \ \sigma_I = 0.1$
$u_{t,n}$	Idiosyncratic volatility shock	$u_{t,n} = 1$
$\varepsilon_{t,n},\ \varepsilon_{t,n}^*$	Log-normal noise in volatility / volume	$\sigma_\varepsilon = 0.1,\ \ \sigma_{\varepsilon^*} = 0.3$

Price dynamics:

$$S_{t,n+1} = S_{t,n}\ \exp\left\{-\tfrac{1}{2}\sigma_{t,n}^2 \Delta t + \sigma_{t,n}\sqrt{\Delta t}\, Z_{t,n+1}\right\}, \qquad Z_{t,n+1} \sim \mathcal{N}(0,1). \tag{3}$$

Hedging Problem

We generate 100,000 price and volume paths on the interval $[0, 1]$ with 100 time steps.

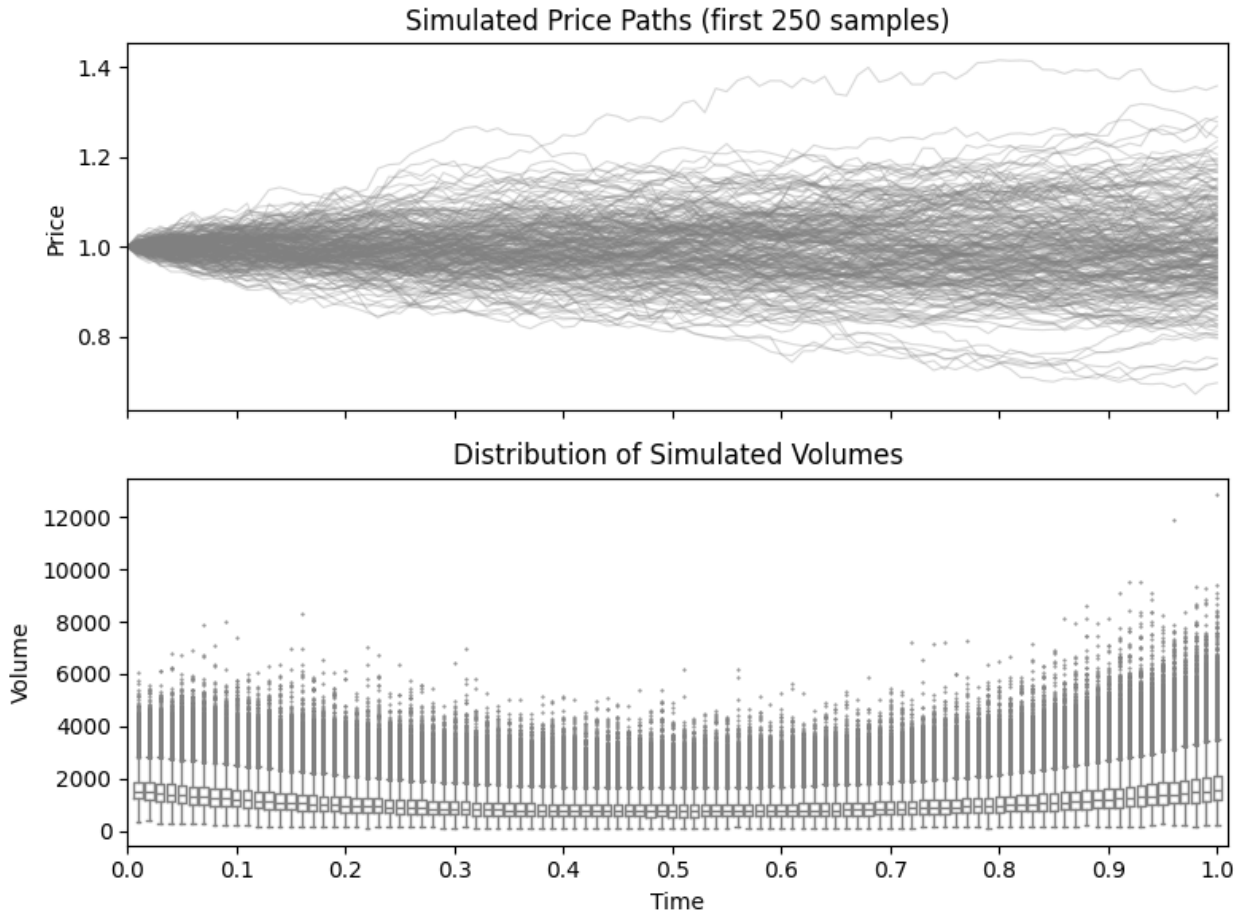

Figure: Simulation of intraday trajectories using the Xiu–Palomar model.

We compare four trading policies for buying $Q_0 = 25{,}000$ units, evaluating their terminal wealth on simulated market paths and measuring risk using the 99% Expected Shortfall.

Hedging Problem

Trading policies: δ_n denotes the hedge amount at time t_n, $n \in \{0, \ldots, N-1\}$

- **Random policy.** draw a random point $(w_0, \ldots, w_{N-1})$ from the N-simplex and allocate trades proportionally: $\delta_n = Q_0 w_n$.
- **Uniform policy.** execute evenly across all time steps: $\delta_n = Q_0 \frac{1}{N}$.
- **Expectation policy.** allocate trades proportionally to the (unconditional) expected market volume: $\delta_n = Q_0 \frac{\mathbb{E}[V_n]}{\sum_{i=0}^{N-1} \mathbb{E}[V_i]}$.
- **Neural policy.** learn a trading policy by mapping market features and past actions to the next allocation using a neural network trained to minimize 99% Expected Shortfall on simulated paths: $\delta_n = f_n(\theta_n; S_n, V_n, \delta_{n-1})$, $\delta_{-1} = 0$.

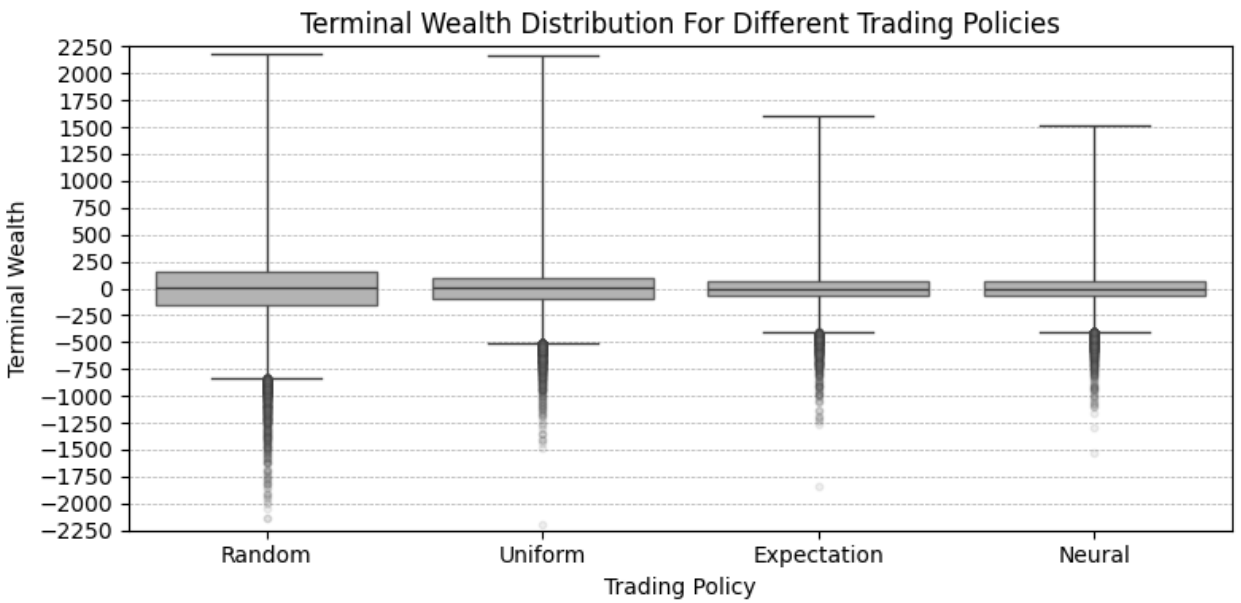

Figure: Wealth distribution of all trading policies.

Policy	99% ES
Random	1,062.9
TWAP	656.3
Expectation	543.0
Neural	522.2

Table: Estimated 99% ES.

Deep Hedging Approach: General Formulation

Goal

Modeling a market with frictions and restrictions, and use deep learning to discover the trading policy that best hedges a liability by minimizing a convex risk measure of the resulting wealth.

Time window partition: we divide $[0, T]$ into $N \in \mathbb{N}$ sub-intervals specified by the sequence

$$0 = t_0 < t_1 < \cdots < t_N = T. \tag{4}$$

Probability space: we work on a filtered probability space $(\Omega, \mathbb{F}, \mathbb{P})$ with $\mathbb{F} = \{\mathcal{F}_n\}_{n=0}^N$. Let $\mathcal{F}_n = \sigma(\mathbf{I}_0, \ldots, \mathbf{I}_n)$, where $\mathbf{I}_n \in \mathbb{R}^r$ denotes the market information at time t_n. Denote the set of all real-valued r.v.s over Ω by $\mathcal{X} = \{X : \Omega \mapsto \mathbb{R}\}$.

Tradable assets: assume $d \in \mathbb{N}$ tradable assets with mid-price process

$$X = \{\mathbf{X}_n\}_{n=0}^N, \quad \mathbf{X}_n = (X_n^{(1)}, \ldots, X_n^{(d)}) \in \mathbb{R}^d_{>0}. \tag{5}$$

Trading policy: the *proposed changes in asset holdings* is represented by the process

$$\delta = \{\boldsymbol{\delta}_n\}_{n=0}^{N-1}, \qquad \boldsymbol{\delta}_n = (\delta_n^{(1)}, \ldots, \delta_n^{(d)}) \in \mathbb{R}^d. \tag{6}$$

Let $\mathcal{H}$ denote the space of all such (policy) processes.

Restrictions: trading at t_n is subject to a function $H_n : \mathbb{R}^{d(n+1)} \to \mathbb{R}^d$, mapping the policy history $(\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_n)$ to a traded quantity $H_n(\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_n)$. The actual position is then

$$\tilde{\boldsymbol{\delta}}_n = \sum_{i=0}^{n} H_i(\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_i), \qquad \tilde{\boldsymbol{\delta}}_{-1} := \mathbf{0}. \tag{7}$$

Deep Hedging Approach: General Formulation

Profits: trading profits (or losses) are calculated via the stochastic integral

$$(H \circ \delta \bullet X)_N = \sum_{n=0}^{N-1} \tilde{\boldsymbol{\delta}}_n \cdot (\mathbf{X}_{n+1} - \mathbf{X}_n). \tag{8}$$

Costs: trading costs are modeled by functions $c_n : \mathbb{R}^d \to \mathbb{R}$ that assign to the traded quantities $\Delta\tilde{\boldsymbol{\delta}}_n$ a per-period cost $c_n\big(\Delta\tilde{\boldsymbol{\delta}}_n\big)$. The total cost becomes

$$C_N(H \circ \delta) = \sum_{n=0}^{N-1} c_n\big(\Delta\tilde{\boldsymbol{\delta}}_n\big), \qquad \Delta\tilde{\boldsymbol{\delta}}_n = \tilde{\boldsymbol{\delta}}_n - \tilde{\boldsymbol{\delta}}_{n-1} = H_n(\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_n). \tag{9}$$

Liability: at time T, we face a $\mathcal{F}_N$-measurable liability with payoff Z (e.g., Call option).

Wealth: Terminal wealth equals after-cost profits minus the liability payoff:

$$W^\delta(Z) = -Z \; + \; (H \circ \delta \bullet X)_N \; - \; C_N(H \circ \delta). \tag{10}$$

To assess wealth distribution risk, we use a **convex risk measure** $\rho : \mathcal{X} \to \mathbb{R}$ satisfying:

- *Monotonicity* (a wealth distribution with greater risk has a higher risk measure).
- *Convexity* (diversification works).
- *Translation invariance* (holding risk-free cash reduces risk exactly in that cash amount).

Optimization objective

$$\pi(Z) = \inf_{\delta \in \mathcal{H}} \; \rho\Big(W^\delta(Z)\Big). \tag{11}$$

Deep Hedging Approach: Solution Via NNs

It can be shown that there exists a map $g_n : \mathbb{R}^{r(n+1)} \to \mathbb{R}^d$ such that $\boldsymbol{\delta}_n = g_n(\mathbf{I}_0, \ldots, \mathbf{I}_n)$. Therefore, solving (11) requires optimizing over all such maps $g_n(\cdot)$, which is an infinite-dimensional problem.

Instead, we parameterize $\boldsymbol{\delta}_n$ using a neural network $f_n(\theta_n; \cdot)$ and approximate a minimizer of (11) by solving the finite-dimensional problem in the parameters θ_n (Buehler et al.).

Feedforward Neural Network

Given input and output dimensions H_0 and H_{L+1}, layer widths $(H_1, \ldots, H_L)$, parameters $A^{(\ell)} \in \mathbb{R}^{H_\ell \times H_{\ell-1}}$, $b^{(\ell)} \in \mathbb{R}^{H_\ell}$, and activation σ, a **neural network** (NN) is the composition

$$f(\theta; x) = A^{(L+1)}\sigma\big(A^{(L)}\sigma(\cdots\sigma(A^{(1)}x + b^{(1)})\cdots) + b^{(L)}\big) + b^{(L+1)}, \tag{12}$$

where $\theta = \{(A^{(\ell)}, b^{(\ell)})\}_{\ell=1}^{L+1}$. We refer to hidden layers and layer widths as **architecture**.

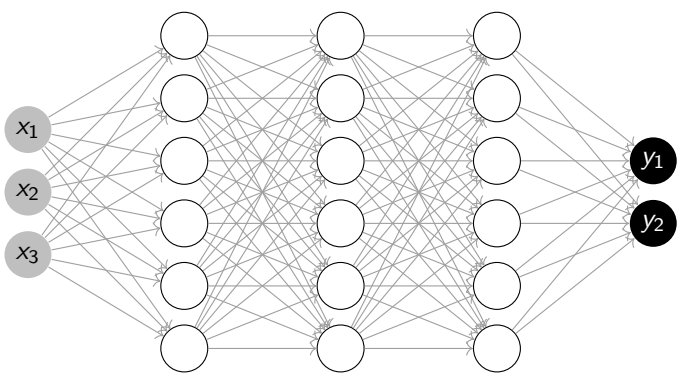

Figure: Neural network example: three inputs, three hidden layers each having width six, and two outputs.

Deep Hedging Approach: Solution Via NNs

We consider a nested family of NN architectures whose depths $L^{(M)}$ and layer widths $H_\ell^{(M)}$ grow with M. For each architecture, the parameter space for a single network with input dimension d_{in} and output dimension d_{out} is

$$\Theta_M(d_{\text{in}}, d_{\text{out}}) = \left\{ \{(A^{(\ell)}, b^{(\ell)})\}_{\ell=1}^{L^{(M)}+1} : A^{(\ell)} \in \mathbb{R}^{H_\ell^{(M)} \times H_{\ell-1}^{(M)}},\ b^{(\ell)} \in \mathbb{R}^{H_\ell^{(M)}} \right\}. \tag{13}$$

For time step $n \in \{0, \ldots, N-1\}$, take $d_{\text{in}} = r(n+1) + d$ and $d_{\text{out}} = d$, and define the **total parameter space** as the product

$$\Theta_M = \prod_{n=0}^{N-1} \Theta_M\big(r(n+1) + d, d\big). \tag{14}$$

The next result shows that, for sufficiently large architectures, a recursive policy network approximates the optimal policy that solves (11).

Proposition 1 (Buehler et al.)

Let $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_{N-1}) \in \Theta_M$, and take NNs $\{f_n(\theta_n; \cdot)\}_{n=0}^{N-1}$. Define the policy $\delta^{\boldsymbol{\theta}} = \{\boldsymbol{\delta}_n\}_{n=0}^{N-1}$ by

$$\boldsymbol{\delta}_{-1} = \mathbf{0}; \qquad \boldsymbol{\delta}_n = f_n\big(\theta_n; \mathbf{I}_0, \ldots, \mathbf{I}_n, \boldsymbol{\delta}_{n-1}\big), \quad n = 0, \ldots, N-1. \tag{15}$$

Consider the optimization objective $\pi_M(Z) = \inf_{\boldsymbol{\theta} \in \Theta_M} \ \rho\Big(W^{\delta^{\boldsymbol{\theta}}}(Z)\Big)$. Then, for any $Z \in \mathcal{X}$,

$$\lim_{M \to \infty} \pi_M(Z) = \pi(Z). \tag{16}$$

Deep Hedging Approach: Practical Application

Strategy To Solve The VWAP Risk Hedging Problem

Applying Deep Hedging to learn an execution policy δ that hedges the downside risk of VWAP-based wealth.

In practice, we follow the next steps:

1. **Market model**
   - Build a stochastic LOB model.
   - Simulate trajectories that act as training scenarios.
2. **Trading policy**
   - Model wealth as the sum of an after-cost profit and a liability which depend on a trading policy.
   - Parameterize the trading policy using neural networks.
   - Define an architecture for the neural network.
3. **Risk-based training**
   - Choose a (empirical) convex risk measure on terminal wealth.
   - Minimize the empirical risk over wealth samples using an optimization algorithm.

Outcome

A trained network policy δ^* that maps (real-)time LOB features to VWAP risk-hedging trading decisions.

Market Model: State Representation

Time window: market opens at time 0 and closes at time $T > 0$ (one trading day).

Tick size: represents the minimum price increase allowed, denoted by $\epsilon > 0$.

Relative prices: at any time $t \in [0, T]$, each side of the LOB is described by $K \in \mathbb{N}$ relative prices, indexed 1 to K ticks away from the best *opposite* quote.

Queues: let $\mathbf{a}_t = (\mathrm{a}_t^{(1)}, \ldots, \mathrm{a}_t^{(K)})$ denote the **ask queues** at time t, where $\mathrm{a}_t^{(k)} \geq 0$ is the volume offered k ticks from the best opposite quote. Likewise, $\mathbf{b}_t = (\mathrm{b}_t^{(1)}, \ldots, \mathrm{b}_t^{(K)})$ denotes the **bid queues**. To prevent queue depletion, we impose constant bounds outside the $2K$ grid (whenever levels shift and a new outermost level appears, its queue is initialized to $c^\infty > 0$).

Mid price and bid-ask spread:

$$S_t = \tfrac{1}{2}(A_t + B_t), \qquad \varepsilon_t = A_t - B_t, \tag{17}$$

where the **best ask price** $A_t \in \epsilon\mathbb{N}$ and the **best bid price** $B_t \in \epsilon\mathbb{N}$ are the prices associated with the first non-zero ask queue and bid queue, respectively.

We use the following notation to represent the market state:

LOB State

$$\mathbf{L}_t = (S_t, \varepsilon_t, \mathbf{a}_t, \mathbf{b}_t). \tag{18}$$

Market Model: State Representation

In the example below, the number of visible relative prices is $K = 9$. The ask and bid queues are $\mathbf{a}_t = (0, 0, 0, 0, 1, 3, 5, 4, 0)$ and $\mathbf{b}_t = (0, 0, 0, 0, 1, 0, 4, 6, 3)$. The bound is $c^\infty = 5$.

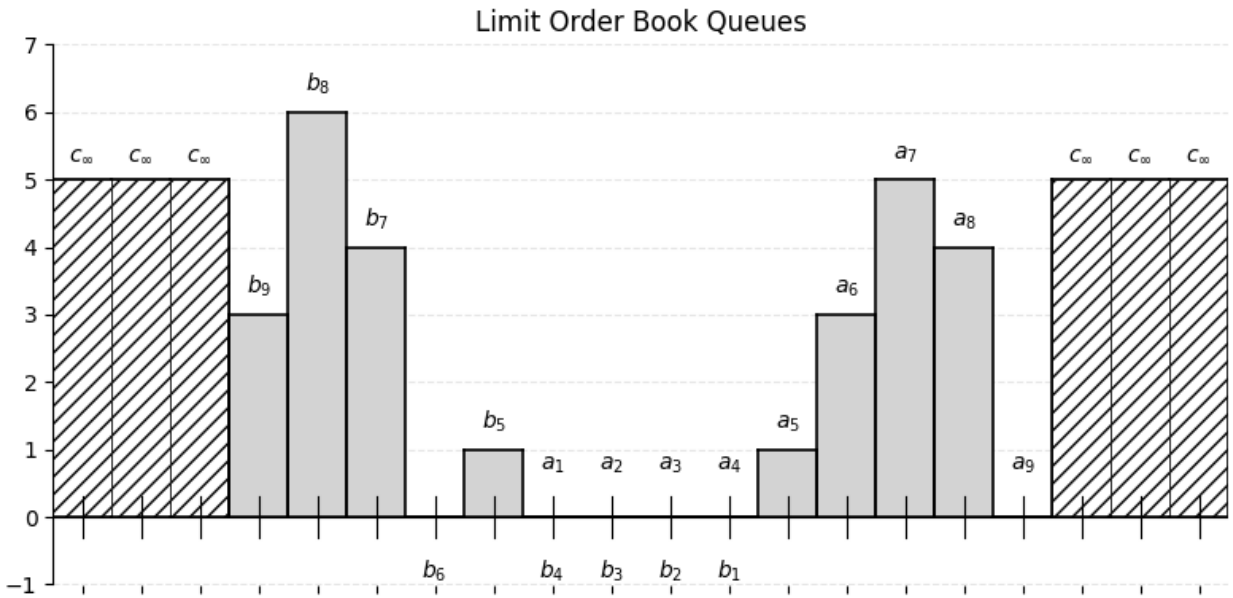

Figure: LOB representation.

Market Model: State Update

LOB dynamics: changes in state $\mathbf{L}_t$ are driven by market participants' actions. Each action is represented as an event.

Events: **arrival of buy or sell MOs**, **arrival of buy or sell LOs**, and full or partial **cancellation of existing buy or sell LOs**.

Given a current state $\mathbf{L}_t$, we apply some rules to update the LOB state when a new event occurs at time $t + \Delta t$.

Example (Buy market order of size $q > 0$)

$$S_t = 100, \quad \varepsilon_t = 0.50 \text{ (tick} = 0.1), \quad \mathbf{a}_t = (0, 0, 0, 0, 1, 3, 5, 4, 0), \quad \mathbf{b}_t = (0, 0, 0, 0, 1, 0, 4, 6, 3).$$

Consider a buy market order of size $q = 6$. Then, the number of ask queues fully consumed is $i = 2$ (the third queue is partially consumed).

Mid-price and bid-ask spread updates:

$$S_{t+\Delta t} = S_t + \tfrac{1}{2} i\, \epsilon = 100 + \tfrac{1}{2} \cdot 2 \cdot 0.1 = 100.1, \qquad \varepsilon_{t+\Delta t} = \varepsilon_t + i\, \epsilon = 0.50 + 2 \cdot 0.1 = 0.70.$$

Ask queue update:

$$\mathbf{a}_{t+\Delta t} = (0, 0, 0, 0, 0, 0, 3, 4, 0).$$

Bid queue update:

$$\mathbf{b}_{t+\Delta t} = (0, 0, 0, 0, 0, 0, 1, 0, 4).$$

Market Model: Simulation

Santa Fe model: event *times* are driven by independent Poisson processes; event *volumes* are sampled from independent random variables:

Event Type	Intensity Parameter	Volume Law
Market orders	γ ($\times 2$)	$\mathcal{V}^{\mathrm{M}}$
Limit orders arrivals at level k	$\lambda^{(k)}$ ($\times 2$)	$\mathcal{V}^{\mathrm{L}}$
Limit order cancellations at level k	$\rho^{(k)}\,\mathrm{b}_t^{(k)}$ (buy), $\rho^{(k)}\,\mathrm{a}_t^{(k)}$ (sell)	$\mathcal{V}^{\mathrm{C}}$

Aggregate and Total Intensities

The agg. intensity of each event category equals the sum of its subtypes' intensities:

$$\lambda = \sum_{k=1}^{K} \lambda^{(k)}, \qquad \rho_t^{\mathrm{b}} = \sum_{k=1}^{K} \rho^{(k)} \mathrm{b}_t^{(k)}, \qquad \rho_t^{\mathrm{a}} = \sum_{k=1}^{K} \rho^{(k)} \mathrm{a}_t^{(k)}. \tag{19}$$

The total intensity governs the waiting time to the next event, regardless of event type:

$$\Lambda_t = 2(\gamma + \lambda) + \rho_t^{\mathrm{b}} + \rho_t^{\mathrm{a}}. \tag{20}$$

Market Model: Simulation

Algorithm 1: Simulate LOB path

inputs: order intensities γ, $\{\lambda^{(k)}\}_{k=1}^{K}$, $\{\rho^{(k)}\}_{k=1}^{K}$; volume laws $\mathcal{V}^{\mathrm{M}}$, $\mathcal{V}^{\mathrm{L}}$, $\mathcal{V}^{\mathrm{C}}$; start/end times s, t; initial state $\mathbf{L}_s = (S_s, \varepsilon_s, \mathbf{a}_s, \mathbf{b}_s)$
initialize: $\mathbf{L} \leftarrow \mathbf{L}_s$, $\tau \leftarrow s$, $\lambda \leftarrow \sum_{k=1}^{K} \lambda^{(k)}$
while $\tau \leq t$ **do**
    Update cancellation intensities: $\rho_\tau^{\mathrm{b}} \leftarrow \sum_{k=1}^{K} \rho^{(k)}\, \mathrm{b}_\tau^{(k)}$, $\rho_\tau^{\mathrm{a}} \leftarrow \sum_{k=1}^{K} \rho^{(k)}\, \mathrm{a}_\tau^{(k)}$.
    Update total intensity: $\Lambda_\tau \leftarrow 2(\gamma + \lambda) + \rho_\tau^{\mathrm{b}} + \rho_\tau^{\mathrm{a}}$
    Simulate next-event time: draw $\Delta\tau \sim \mathrm{Exp}(\Lambda_\tau)$
    if $\tau + \Delta\tau > t$ **then break while loop**
    Simulate event type: draw $\mathtt{e} \sim \mathrm{Cat}\Big((\gamma, \gamma, \lambda, \lambda, \rho_\tau^{\mathrm{a}}, \rho_\tau^{\mathrm{b}})/\Lambda_\tau\Big)$
    Simulate quantity: draw $\mathtt{q} \sim \mathcal{V}$ depending on $\mathtt{e}$
    Simulate relative price:
        if $\mathtt{e} \in \{\text{sell LO},\ \text{buy LO}\}$ **then** draw $\mathtt{k} \sim \mathrm{Cat}\Big((\lambda^{(1)}, \ldots, \lambda^{(K)})/\lambda\Big)$
        else if $\mathtt{e} = $ CXL sell LO **then** draw $\mathtt{k} \sim \mathrm{Cat}\Big((\rho^{(1)}\mathrm{a}_\tau^{(1)}, \ldots, \rho^{(K)}\mathrm{a}_\tau^{(K)})/\rho_\tau^{\mathrm{a}}\Big)$
        else if $\mathtt{e} = $ CXL buy LO **then** draw $\mathtt{k} \sim \mathrm{Cat}\Big((\rho^{(1)}\mathrm{b}_\tau^{(1)}, \ldots, \rho^{(K)}\mathrm{b}_\tau^{(K)})/\rho_\tau^{\mathrm{b}}\Big)$
        end if
    Update time: $\tau \leftarrow \tau + \Delta\tau$
    Update state: $\mathbf{L} \leftarrow \mathbf{L} + (\mathtt{e}, \mathtt{q}, \mathtt{k})$
end while
output: $\mathbf{L}$

Market Model: Simulation

The following figure shows a simulated LOB trajectory using **Algorithm 1** with Santa Fe parameters calibrated on real data.

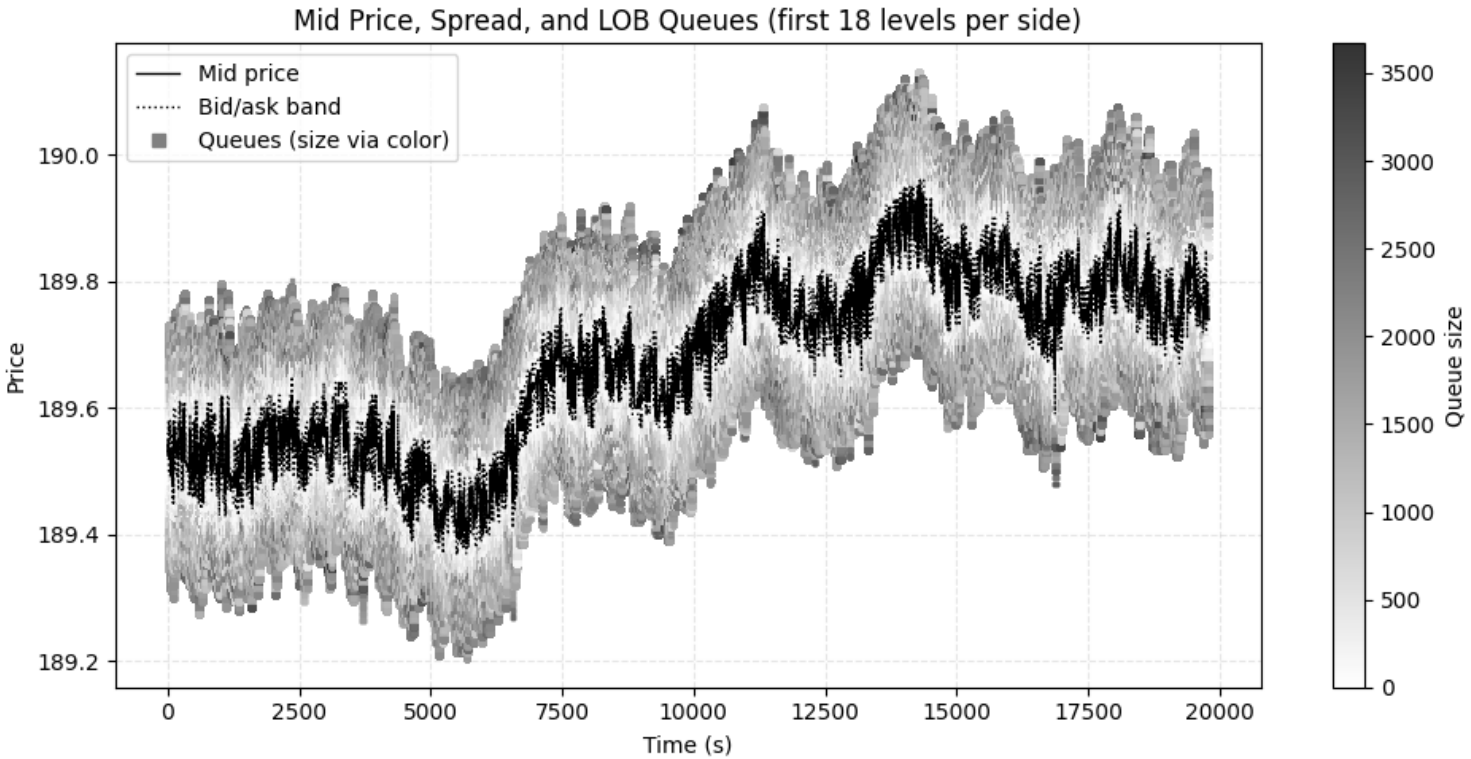

Figure: Simulated LOB.

Trading Policy: Wealth Parametrization

Algorithm 1 provides a procedure to simulate a continuous-time LOB state $\mathbf{L}_t$.

If we take snapshots of this process at each time t_n, we obtain a new discrete-time, $\mathbb{F}$-adapted process $\widetilde{\mathrm{L}} = \{\widetilde{\mathbf{L}}_n\}_{n=0}^{N}$ by setting

$$\widetilde{\mathbf{L}}_n := (\widetilde{S}_n, \widetilde{\varepsilon}_n, \widetilde{\mathbf{a}}_n, \widetilde{\mathbf{b}}_n) = (S_{t_n}, \varepsilon_{t_n}, \mathbf{a}_{t_n}, \mathbf{b}_{t_n}), \qquad n = 1, \ldots, N. \tag{21}$$

Trading Policy

Fix $L \in \mathbb{N}$ such that $L \leq K$. Define the $(L+1)$-dimensional, $\mathbb{F}$-adapted stochastic process

$$(\varphi, \vartheta) = \{(\varphi_n, \boldsymbol{\vartheta}_n)\}_{n=0}^{N-1}, \tag{22}$$

where $(\varphi_n, \boldsymbol{\vartheta}_n) = (\varphi_n, \vartheta_n^{(1)}, \ldots, \vartheta_n^{(L)})$ represents our orders at time t_n:

- a buy MO with size $\varphi_n \geq 0$;
- L buy LOs, each with a relative price $\ell \in \{1, \ldots, L\}$ and a size $\vartheta_n^{(\ell)} \geq 0$.

We impose $(\varphi_{-1}, \boldsymbol{\vartheta}_{-1}) = \mathbf{0}$ (i.e., no prior positions).

At time t_n, the **execution quantity** $Q_n^{(\varphi,\vartheta)}$ is the number of units executed via MOs and filled LOs during $[t_{n-1}, t_n)$. Similarly, the **execution value** $V_n^{(\varphi,\vartheta)}$ is the total cost (executed quantities times execution prices) over the same sub-interval.

Trading Policy: Wealth Parametrization

1) Contribution of market orders:

We define the **ask-price ladder**

$$A_n^{(k)} = \widetilde{S}_n - \tfrac{1}{2}\widetilde{\varepsilon}_n + k\,\epsilon, \qquad k \in \{1, \ldots, K\}. \tag{23}$$

When buying φ_n units against the ask side, the **depleted liquidity** from queue $\widetilde{\mathrm{a}}_n^{(k)}$ is given by

$$d^{(k)}(\varphi_n) = \min\big\{\, \widetilde{\mathrm{a}}_n^{(k)},\ (\varphi_n - \Sigma_{k-1}(\widetilde{\mathbf{a}}_n))_+ \big\}. \tag{24}$$

Formula (24) simply breaks down φ_n units into k amounts that do not exceed the ask queues. If we denote the **actual executed quantity** by

$$m_n(\varphi_n) = \sum_{k=1}^{K} d^{(k)}(\varphi_n), \tag{25}$$

then the **average price** $A_n^{(0)}(\varphi_n)$ for the φ_n units purchased is given by

$$A_n^{(0)}(\varphi_n) \;=\; \begin{cases} \dfrac{\sum_{k=1}^{K} d^{(k)}(\varphi_n) A_n^{(k)}}{m_n(x)}, & \text{if } m_n(\varphi_n) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

Contribution of Market Orders

The quantity and value accruals from the MO are:

$$Q_n^{(\varphi,\vartheta)} \leftarrow Q_n^{(\varphi,\vartheta)} + m_n(\varphi_n), \qquad V_n^{(\varphi,\vartheta)} \leftarrow V_n^{(\varphi,\vartheta)} + m_n(\varphi_n) A_n^{(0)}(\varphi_n). \tag{27}$$

Trading Policy: Wealth Parametrization

2) Contribution of Previous Limit Orders:

We define the **bid-price ladder** as

$$B_n^{(k)} = \widetilde{S}_n + \tfrac{1}{2}\widetilde{\varepsilon}_n - k\,\epsilon, \qquad k \in \{1, \dots, K\}. \tag{28}$$

When submitting $\vartheta_{n-1}^{(\ell)}$ LOs at price $B_{n-1}^{(\ell)}$ at time t_{n-1}, $\ell \in \{1, \dots, L\}$, the **filled quantity** at time t_n is given by

$$r_n^{(\ell)}(\vartheta_{n-1}^{(\ell)}) = e_n^{(\ell)}\,\vartheta_{n-1}^{(\ell)}, \tag{29}$$

where $e_n^{(\ell)} \in \{0, 0.5, 1\}$ reflects realized activity on $[t_{n-1}, t_n)$:

- 1 if the market traded below $B_{n-1}^{(\ell)}$,
- 0.5 if the lowest trade was exactly $B_{n-1}^{(\ell)}$,
- and 0 otherwise.

Contribution of Previous Limit Orders

The quantity and value accruals from filled LOs are:

$$Q_n^{(\varphi,\vartheta)} \leftarrow Q_n^{(\varphi,\vartheta)} + \sum_{\ell=1}^{L} r_n^{(\ell)}(\vartheta_{n-1}^{(\ell)}), \qquad V_n^{(\varphi,\vartheta)} \leftarrow V_n^{(\varphi,\vartheta)} + \sum_{\ell=1}^{L} r_n^{(\ell)}(\vartheta_{n-1}^{(\ell)})\,B_{n-1}^{(\ell)}. \tag{30}$$

Trading Policy: Wealth Parametrization

Adding up all contributions at time t_n:

$$Q_n^{(\varphi,\vartheta)} = m_n(\varphi_n) \; + \; \sum_{\ell=1}^{L} r_n^{(\ell)}(\vartheta_{n-1}^{(\ell)}), \tag{31}$$

$$V_n^{(\varphi,\vartheta)} = m_n(\varphi_n) A_n^{(0)}(\varphi_n) \; + \; \sum_{\ell=1}^{L} r_n^{(\ell)}(\vartheta_{n-1}^{(\ell)}) \, B_n^{(\ell)}. \tag{32}$$

Unexecuted LOs are canceled at each period end (we do not forgo execution priority under this setup). For the last time step, we assume no LOs are submitted.

Summing over all time steps, we have:

Total Execution Quantity And Average Execution Price

$$Q^{(\varphi,\vartheta)} = \sum_{n=0}^{N-1} Q_n^{(\varphi,\vartheta)}, \qquad P^{(\varphi,\vartheta)} = \begin{cases} \dfrac{\sum_{n=0}^{N-1} V_n^{(\varphi,\vartheta)}}{Q^{(\varphi,\vartheta)}}, & \text{if } Q^{(\varphi,\vartheta)} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

Trading Policy: Wealth Parametrization

Let $\{(\tau_m, q_m, p_m)\}_{m=1}^{M_T}$ be the MOs over $[0, T]$, with $q_m > 0$. For $n \in \{0, \dots, N\}$, define the set

$$M_n = \{\, m : \tau_m \in [t_{n-1}, t_n) \,\}. \tag{34}$$

The **market's execution quantity** and **execution value** at time t_n are

$$Q_n^{\mathrm{mkt}} = \sum_{m \in M_n} q_m, \qquad V_n^{\mathrm{mkt}} = \sum_{m \in M_n} q_m\, p_m. \tag{35}$$

Market's Volume-Weighted Average Price (VWAP)

$$\mathrm{VWAP} \;=\; \frac{\sum_{n=0}^{N} V_n^{\mathrm{mkt}}}{\sum_{n=0}^{N} Q_n^{\mathrm{mkt}}}. \tag{36}$$

Risk Hedging Problem Parametrization

The terminal wealth is defined as

$$W_{\mathrm{VWAP}}^{(\varphi,\vartheta)} = Q_0\big(\mathrm{VWAP} - \widetilde{S}_N\big) - Q^{(\varphi,\vartheta)}\big(P^{(\varphi,\vartheta)} - \widetilde{S}_N\big) - \tfrac{1}{2}\widetilde{\varepsilon}_N \left|Q^{(\varphi,\vartheta)} - Q_0\right|, \tag{37}$$

where we include a terminal penalty $\frac{1}{2}\widetilde{\varepsilon}_N \left|Q^{(\varphi,\vartheta)} - Q_0\right|$ to discourage mandate deviations.
Let $\mathcal{H}$ represent the space of all policy processes such as (22), and let ρ be a convex risk measure. The optimization problem is

$$\pi = \inf_{(\varphi,\vartheta)\in\mathcal{H}} \rho\Big(W_{\mathrm{VWAP}}^{(\varphi,\vartheta)}\Big). \tag{38}$$

Trading Policy: Deep-Hedging Modeling

By modeling the components in

$$W^{\delta}(Z) = -Z \; + \; (H \circ \delta \bullet X)_N \; - \; C_N(H \circ \delta), \tag{39}$$

we can express

$$W^{(\varphi,\vartheta)}_{\mathrm{VWAP}} = Q_0\big(\mathrm{VWAP} - \widetilde{S}_N\big) - Q^{(\varphi,\vartheta)}\big(P^{(\varphi,\vartheta)} - \widetilde{S}_N\big) - \tfrac{1}{2}\widetilde{\varepsilon}_N\,\big|Q^{(\varphi,\vartheta)} - Q_0\big| \tag{40}$$

as the sum of a liability, transaction costs, and trading gains, justifying the DH approach.

To this end, we choose the following:

Deep Hedging Modeling

$$X_n^{(\ell)} = \widetilde{S}_n, \quad \ell = 1, \ldots, L+1, \tag{41}$$

$$\boldsymbol{\delta}_n = \big(\varphi_n, \vartheta_n^{(1)}, \ldots, \vartheta_n^{(L)}\big), \tag{42}$$

$$H_n(\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_n) = \Big(m_n(\varphi_n),\, r_n^{(1)}(\vartheta_{n-1}^{(1)}),\, \ldots,\, r_n^{(L)}(\vartheta_{n-1}^{(L)})\Big), \tag{43}$$

$$c_n(H_n) = m_n(\varphi_n)\Big(A_n^{(0)}(\varphi_n) - \widetilde{S}_n\Big) + \sum_{\ell=1}^{L} r_n^{(\ell)}(\vartheta_{n-1}^{(\ell)})\big(B_{n-1}^{(\ell)} - \widetilde{S}_n\big), \tag{44}$$

$$Z = Q_0\big(\widetilde{S}_N - \mathrm{VWAP}\big) + \tfrac{1}{2}\widetilde{\varepsilon}_N\,\big|Q^{(\varphi,\vartheta)} - Q_0\big|. \tag{45}$$

Trading Policy: Parametrized Wealth Samples

At time t_n, the relevant market information includes:

- The LOB state: $\widetilde{\mathbf{L}}_n = (\widetilde{S}_n, \widetilde{\varepsilon}_n, \widetilde{\mathbf{a}}_n, \widetilde{\mathbf{b}}_n)$.
- The realized LO fill coefficients from the previous interval: $e_n^{(1)}, \ldots, e_n^{(L)}$.
- The cumulative market's execution quantity and value: $\sum_{i=0}^{n} Q_i^{\mathrm{mkt}}$, $\sum_{i=0}^{n} V_i^{\mathrm{mkt}}$.

The value of L is determined empirically by measuring how far MOs typically penetrate the book and taking the 99th percentile of those depths.

To reduce input dimension, we truncate the observable depth K by passing only the first L queues per side. The observable information at time t_n is then gathered into the feature vector

Market Features at Time t_n

$$\mathbf{I}_n = \Big(\widetilde{S}_n, \widetilde{\varepsilon}_n, \widetilde{\mathrm{a}}_n^{(1)}, \ldots, \widetilde{\mathrm{a}}_n^{(L)}, \widetilde{\mathrm{b}}_n^{(1)}, \ldots, \widetilde{\mathrm{b}}_n^{(L)}, e_n^{(1)}, \ldots, e_n^{(L)}, \sum_{i=0}^{n} Q_i^{\mathrm{mkt}}, \sum_{i=0}^{n} V_i^{\mathrm{mkt}}\Big). \tag{46}$$

Remark. Under the Santa Fe model, the LOB state process $\{\mathrm{L}_t\}_{t\in[0,T]}$ is Markovian. Therefore, as pointed out in *Buehler et al.*, we may simplify the recursive network policy as follows:

$$(\varphi_n, \vartheta_n^{(1)}, \ldots, \vartheta_n^{(L)}) = f_n\big(\theta_n; \mathbf{I}_n, \varphi_{n-1}, \vartheta_{n-1}^{(1)}, \ldots, \vartheta_{n-1}^{(L)}\big). \tag{47}$$

Trading Policy: Parametrized Wealth Samples

To parametrize wealth samples, we use **Algorithm 2**:

Algorithm 2: Compute wealth samples

inputs: simulated market features $\{(\mathbf{I}_0^{(j)}, \mathbf{I}_1^{(j)}, \ldots, \mathbf{I}_N^{(j)})\}_{j=1}^{J}$; neural network parameters $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_{N-1})$; target quantity Q_0

for $j = 1$ to J **do**

  initialize: $(\varphi_{-1}, \vartheta_{-1}^{(1)}, \ldots, \vartheta_{-1}^{(L)}) \leftarrow \mathbf{0}$, $Q \leftarrow 0$, $G \leftarrow 0$, $C \leftarrow 0$

  for $n = 0$ to $N - 1$ **do**

    Compute policy: $(\varphi, \vartheta^{(1)}, \ldots, \vartheta^{(L)}) \leftarrow f_n(\theta_n; \mathbf{I}_n^{(j)}, \varphi_{-1}, \vartheta_{-1}^{(1)}, \ldots, \vartheta_{-1}^{(L)})$

    Update inventory: $Q \leftarrow Q + Q_n^{(\varphi,\vartheta),j}$

    Update profits: $G \leftarrow G + Q(\widetilde{S}_{n+1}^{(j)} - \widetilde{S}_n^{(j)})$

    Update costs: $C \leftarrow C + V_n^{(\varphi,\vartheta),j} - Q_n^{(\varphi,\vartheta),j}\widetilde{S}_n^{(j)}$

    Update previous policy: $(\varphi_{-1}, \vartheta_{-1}^{(1)}, \ldots, \vartheta_{-1}^{(L)}) \leftarrow (\varphi, \vartheta^{(1)}, \ldots, \vartheta^{(L)})$

  end for

  Compute VWAP: $\text{VWAP} \leftarrow \sum_{i=0}^{N} V_i^{\text{mkt},j} / \sum_{i=0}^{N} Q_i^{\text{mkt},j}$

  Compute liability: $Z \leftarrow Q_0(\widetilde{S}_N^{(j)} - \text{VWAP}) - \frac{1}{2}\widetilde{\varepsilon}_N^{(j)}|Q - Q_0|$

  Compute wealth sample: $W^{(j)}(\boldsymbol{\theta}) \leftarrow -Z + G - C$

end for

output: $\{W^{(1)}(\boldsymbol{\theta}), \ldots, W^{(J)}(\boldsymbol{\theta})\}$

Trading Policy: Neural Network Architecture

Network map: for each t_n, we employ a neural network with two hidden layers of width 16:

$$H_0 = 3L + 5, \qquad H_\ell = 16 \ \ (\ell = 1, 2), \qquad H_3 = L + 1. \tag{48}$$

Taking weights $A_n^{(\ell)} \in \mathbb{R}^{H_\ell \times H_{\ell-1}}$ and biases $b_n^{(\ell)} \in \mathbb{R}^{H_\ell}$, $\ell \in \{1, 2, 3\}$, the input layer is

$$h_n^{(0)} = (\mathbf{I}_n, \varphi_{n-1}, \vartheta_{n-1}^{(1)}, \ldots, \vartheta_{n-1}^{(L)}), \tag{49}$$

the hidden layers are

$$h_n^{(\ell)} = \tanh\big(A_n^{(\ell)} h_n^{(\ell-1)} + b_n^{(\ell)}\big), \qquad \ell = 1, 2, \tag{50}$$

and the output layer is

$$h_n^{(3)} = \mathrm{ReLU}\big(A_n^{(3)} h_n^{(2)} + b_n^{(3)}\big) = (\varphi_n, \vartheta_n^{(1)}, \ldots, \vartheta_n^{(L)}). \tag{51}$$

The tanh nonlinearity supplies smooth saturating hidden features, while the output ReLU enforces the constraint of nonnegative order sizes.

Time discretization: we fix $N = 128$ trading nodes and adopt an equidistant grid on $[0, T]$, $t_n = \frac{n}{N} T$, $n \in \{0, \ldots, N\}$, so that we rebalance at uniform intervals.

Trading Policy: Neural Network Architecture

Schematic of the full neural network system:

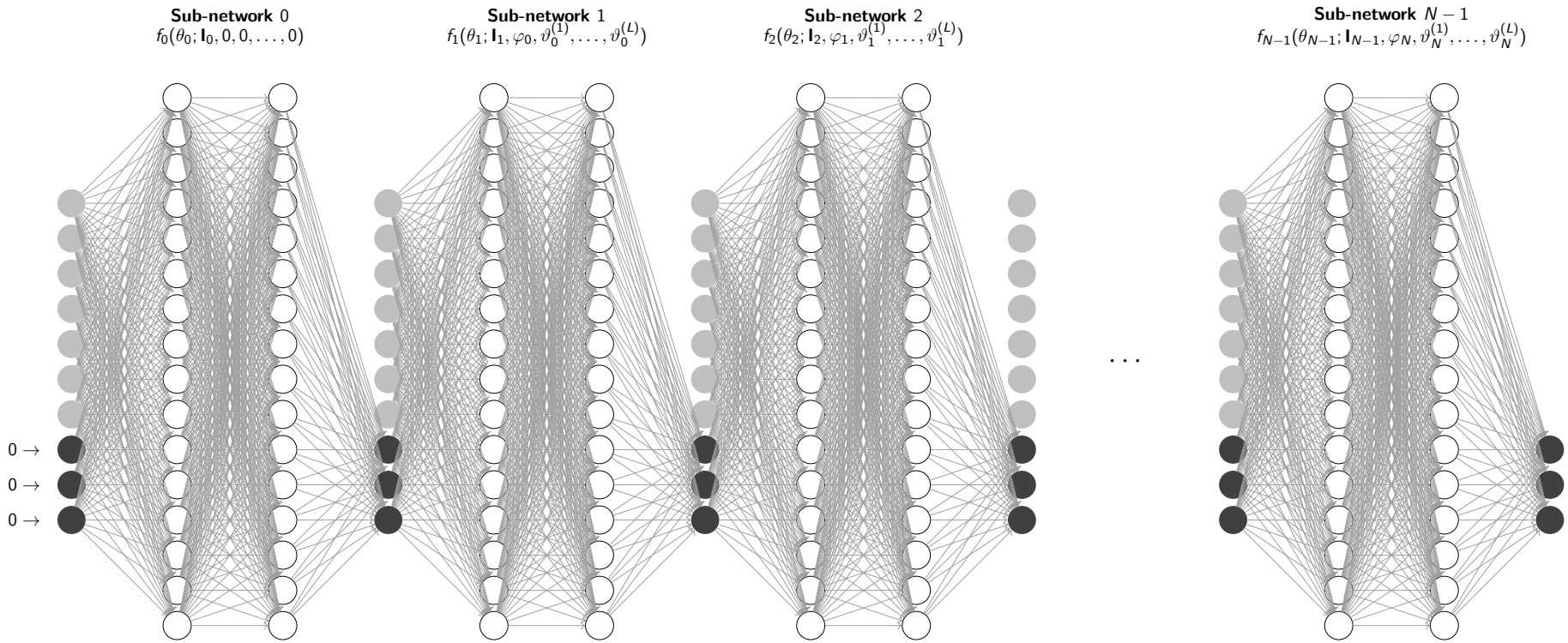

Figure: System with N sub-networks. Each sub-network ingests $3L + 5$ inputs ($2L + 4$ gray market features and $L + 1$ black previous-policy features), has two hidden layers of width 16, and outputs $L + 1$ non-negative actions that feed the next time step. The initial black inputs are zeros (flat start).

Loss Function: Convex Risk Measure

Convex risk measure: Expected Shortfall at confidence level $\alpha \in (0,1)$.

Ideally, we would use SGD to find the optimal neural network parameters; however, mini-batch gradients for loss functions that are convex risk measures are generally biased estimators of their full-batch version.

As a work-around, we extend the search space by one dimension and perform mini-batch optimization of an equivalent representation of Expected Shortfall.

Take parameters $\boldsymbol{\theta} \in \widetilde{\Theta}_M$, and denote by $\{W^{(1)}(\boldsymbol{\theta}), \ldots, W^{(J)}(\boldsymbol{\theta})\}$ the wealth samples. Define the losses $L^{(j)}(\boldsymbol{\theta}) = -W^{(j)}(\boldsymbol{\theta})$, $j \in \{1, \ldots, J\}$, and let $\ell^{(1)}(\boldsymbol{\theta}) \leq \cdots \leq \ell^{(J)}(\boldsymbol{\theta})$ be the sorted statistic.

Let $\alpha \in (0,1)$ (confidence level) and $\tau \in \mathbb{R}$ (threshold), and define $k = \lceil \alpha J \rceil$, $r = k - \alpha J$.

(Sample) Expected Shortfall

$$\widehat{\mathrm{ES}}_\alpha(\boldsymbol{\theta}) = \frac{1}{(1-\alpha)\,J} \sum_{j=k+1}^{J} \left(\ell^{(j)}(\boldsymbol{\theta}) + r\,\ell^{(k)}(\boldsymbol{\theta}) \right). \tag{52}$$

(Sample) Rockafellar–Uryasev

$$\widehat{\mathrm{RU}}_{\alpha,\tau}(\boldsymbol{\theta}) = \tau + \frac{1}{(1-\alpha)\,J} \sum_{j=1}^{J} \left(L^{(j)}(\boldsymbol{\theta}) - \tau \right)_+. \tag{53}$$

Loss Function: Convex Risk Measure

The following result links both metrics:

Theorem (Rockafellar & Uryasev)

For any $\alpha \in (0,1)$, *we have*

$$\widehat{\mathrm{ES}}_\alpha(\boldsymbol{\theta}) = \inf_{\tau \in \mathbb{R}} \widehat{\mathrm{RU}}_{\alpha,\tau}(\boldsymbol{\theta}), \tag{54}$$

with minimizer

$$\tau^*(\boldsymbol{\theta}) \in \arg\inf_{\tau \in \mathbb{R}} \widehat{\mathrm{RU}}_{\alpha,\tau}(\boldsymbol{\theta}) = \begin{cases} \ell^{(k)}(\boldsymbol{\theta}), & \text{if } \alpha J \notin \mathbb{N}, \\ [\ell^{(k)}(\boldsymbol{\theta}), \ell^{(k+1)}(\boldsymbol{\theta})], & \text{otherwise .} \end{cases} \tag{55}$$

By the theorem and the fact that the minimization of $\widehat{\mathrm{RU}}_{\alpha,\tau}(\boldsymbol{\theta})$ with respect to $(\tau, \boldsymbol{\theta}) \in \mathbb{R} \times \Theta_M$ can be carried out by first minimizing over $\tau \in \mathbb{R}$ for fixed $\boldsymbol{\theta}$ and then minimizing the result over $\boldsymbol{\theta} \in \Theta_M$ (see Rockafellar & Uryasev), we have

$$\inf_{\boldsymbol{\theta} \in \Theta_M} \widehat{\mathrm{ES}}_\alpha(\boldsymbol{\theta}) = \inf_{(\tau,\boldsymbol{\theta}) \in \mathbb{R} \times \Theta_M} \widehat{\mathrm{RU}}_{\alpha,\tau}(\boldsymbol{\theta}). \tag{56}$$

Equation (56) allows us to minimize ES via the equivalent RU objective, the advantage being that the latter is suitable for mini-batch training (assuming the map $\boldsymbol{\theta} \mapsto W^{(j)}(\boldsymbol{\theta})$ is (piecewise) differentiable and that the mini-batch (sub)gradients are unbiased).

Experiments: Estimated Market Model Parameters

The numerical experiments are based on **LOBSTER** (Limit Order Book Reconstruction System) data, which reconstructs NASDAQ's full depth-of-book.

For our simulation, we selected four stocks:

Name	Ticker	Start date	End date	Start time	End time
Cisco	CSCO	01.01.2015	31.03.2015	09:30	16:00
Intel	INTC	01.01.2015	31.03.2015	09:30	16:00
Priceline	PCLN	01.01.2015	31.03.2015	09:30	16:00
Tesla	TSLA	01.01.2015	31.03.2015	09:30	16:00

Table: Dataset description: ticker symbols, sample period, and daily trading window.

To calibrate the Santa Fe model, we estimate order-flow intensities and volume distribution parameters for each stock using the 61 business days reported. Order volumes are modeled as log-normal.

Effective time window: analysis restricted to 10:00–15:30 to avoid open and close noise effects. This yields an effective horizon of $T = 19{,}800$ seconds (5.5 trading hours per day).

Experiments: Estimated Market Model Parameters

Estimated Santa Fe model parameters:

Parameter		CSCO	INTC	PCLN	TSLA
Tick	ϵ	0.01	0.01	0.01	0.01
LOB depth	K	10	10	295	75
Relative price levels	L	6	5	78	18
LOB bounds	c^{∞}	7,764	4,377	178	6,494
MO intensity	2γ	0.6067	0.8508	0.0936	0.2730
LO total intensity	2λ	9.7078	13.2461	2.2086	4.6423
CXL total intensity (avg.)	$\rho^{\mathrm{a}}+\rho^{\mathrm{b}}$	9.5266	12.6604	2.1232	4.4019
Total event intensity (avg.)	Λ	19.8411	26.7573	4.4254	9.3172
MO log-volume location	μ_{M}	5.6911	5.2753	3.6789	3.9220
MO log-volume scale	σ_{M}	1.0924	0.9487	1.0477	1.3312
LO log-volume location	μ_{L}	5.0833	4.8582	3.1401	4.3454
LO log-volume scale	σ_{L}	1.1052	1.0253	1.4952	0.5664
CXL log-volume location	μ_{C}	5.5645	5.2079	3.6754	4.3453
CXL log-volume scale	σ_{C}	1.3179	1.0834	1.0271	0.5161

Table: Santa Fe model parameter estimates for all stocks.

Highlights:

- **Activity.** Total event intensity is highest for INTC/CSCO, lowest for PCLN, with TSLA in between.
- **Effective depth.** CSCO/INTC require shallow books ($L = 6/L = 5$), while PCLN and TSLA need much deeper ones ($L = 78/L = 18$).
- **Passive vs. aggressive flow.** MOs represent only 2–3% of events across all stocks.

Experiments: Estimated Market Model Parameters

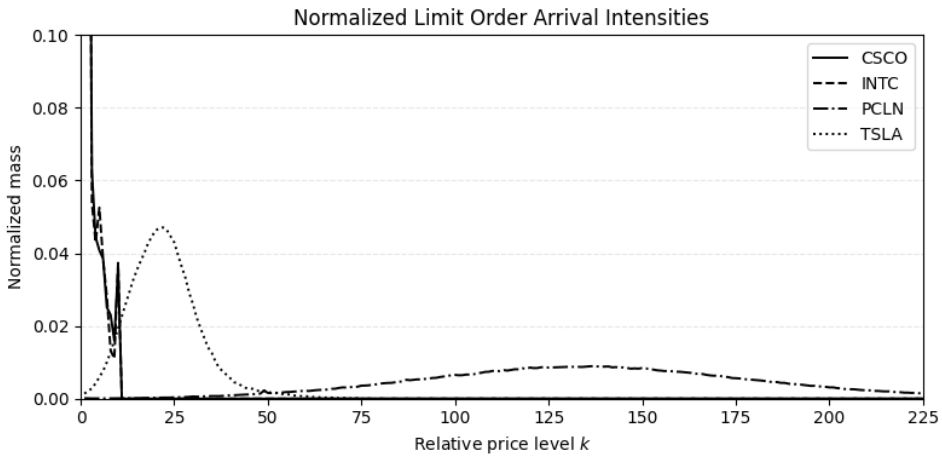

Figure: Normalized LO arrival intensities.

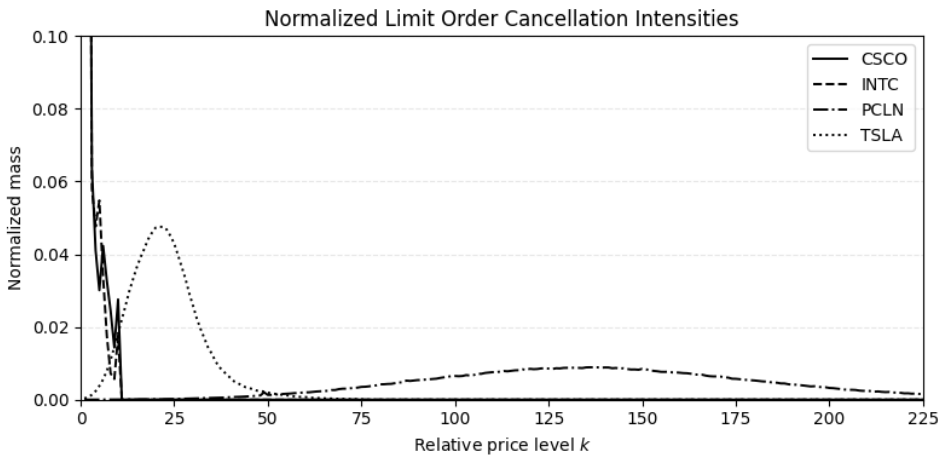

Figure: Normalized LO cancellation intensities.

Highlights:

- **Arrival–cancellation alignment.** Arrival and cancellation profiles closely match for every stock.
- **Front vs. deep book.** CSCO/INTC concentrate activity near the top of the book; PCLN spreads activity broadly; TSLA increases with depth, peaks, then tapers.

Experiments: Optimized Trading Policy

Initial LOB state: queues are initialized to the average estimated queue sizes; the mid-price is set to the closing mid-price from the 2015-03-31 15:30 snapshot; the bid-ask spread is set to the average queues' spread.

Comparability of results: hedge (execution quantities) and order allocations are divided by the target quantity. The training loss and wealth are divided by the target quantity and initial mid-price.

Target quantity: Q_0 is set to 25% of the average daily traded volume over, which can be estimated as

$$\mathbb{E}\big[\text{MO volume over } [0, T]\big] = \underbrace{2\gamma\, T}_{\text{avg. no. MOs}} \times \underbrace{e^{\mu_{\mathrm{M}} + \frac{1}{2}\sigma_{\mathrm{M}}^2}}_{\text{avg. MO size}}. \tag{57}$$

The next table shows the estimated average MO daily volume and the target quantity Q_0 for each stock.

Parameter	CSCO	INTC	PCLN	TSLA
Avrg. traded vlm.	6,462,018	5,163,794	127,056	662,124
Target quantity	1,615,505	1,290,949	31,764	165,531

Table: Average daily traded volume and target quantity per stock.

Experiments: Optimized Trading Policy

Training protocol: the training set consists of $J = 250{,}000$ simulated trajectories. We use early stopping with patience 10 (budget 1,000 epochs) and a halve-on-plateau learning-rate schedule (initial rate 10^{-1}, minimum 2^{-10}). Optimization is performed with a Adam and mini-batches of size $B = 25{,}000$. The network parameters are initialized at random.

The training loss per epoch shows two phases:

- **Phase 1 (pre-elbow):** rapid reduction of obvious errors (e.g., avoiding costly orders and limiting inventory exposure).
- **Phase 2 (post-elbow):** slower, incremental improvements.

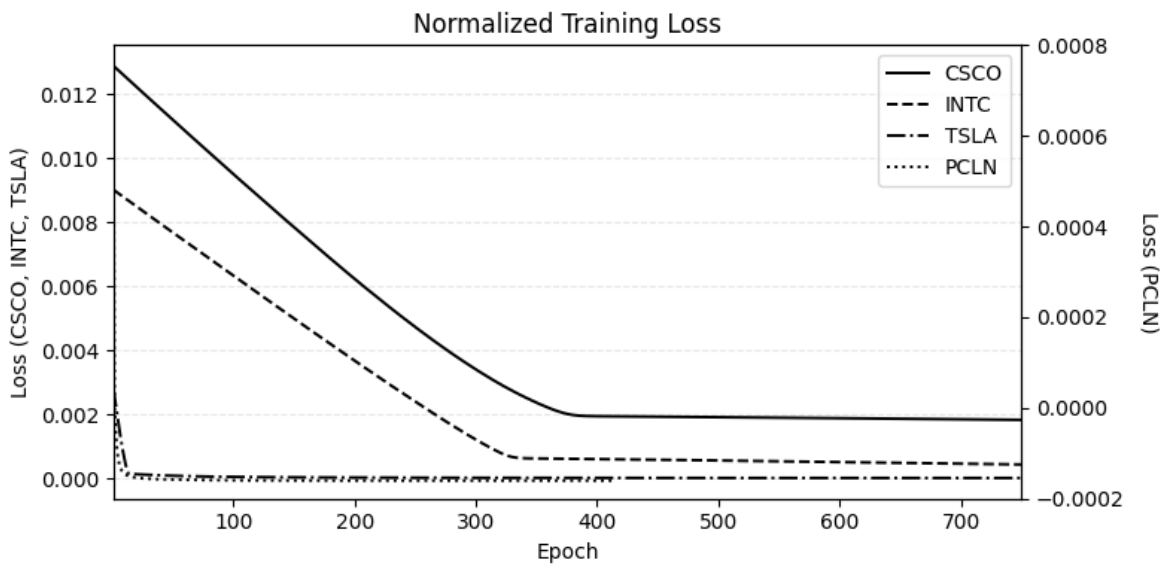

Figure: Normalized loss.

Experiments: Optimized Trading Policy

Applying the trained policy to the training set yields wealth distributions with slightly positive sample means. This is consistent with agents' tendency to accumulate inventory near the market close (see the discussion below on inventory evolution).

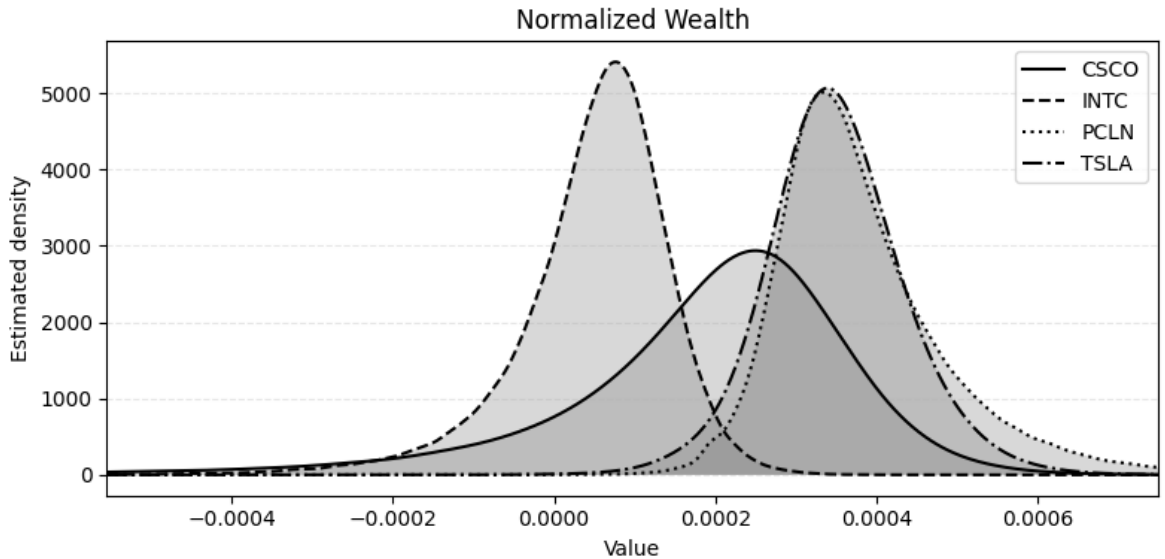

Figure: Normalized wealth.

Parameter	CSCO	INTC	PCLN	TSLA
Expctd. val.	1.7555	0.4868	3.8162	3.4728
Stndrd. dev.	3.4777	0.9708	1.0965	3.6483
RU measure	17.5836	3.6367	-1.6038	0.1863

Table: Wealth distribution results for all stock (scaled by 10^4).

Experiments: Optimized Trading Policy

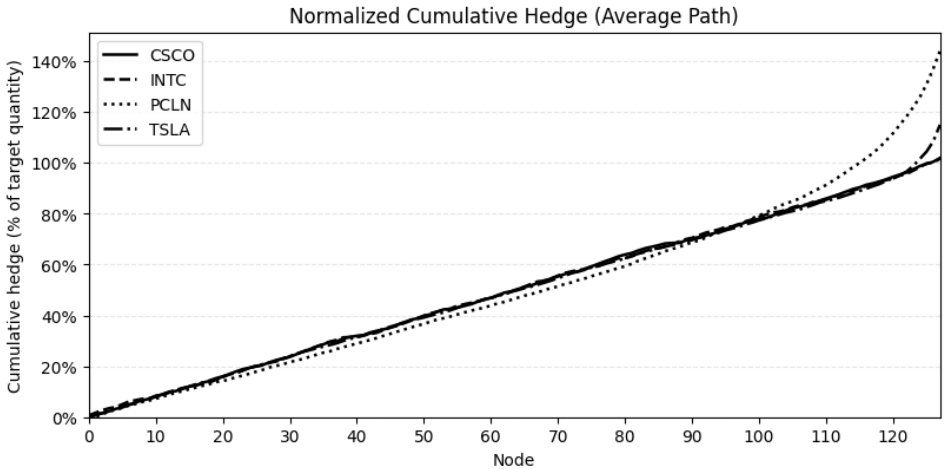

Figure: Cumulative normalized hedge (path).

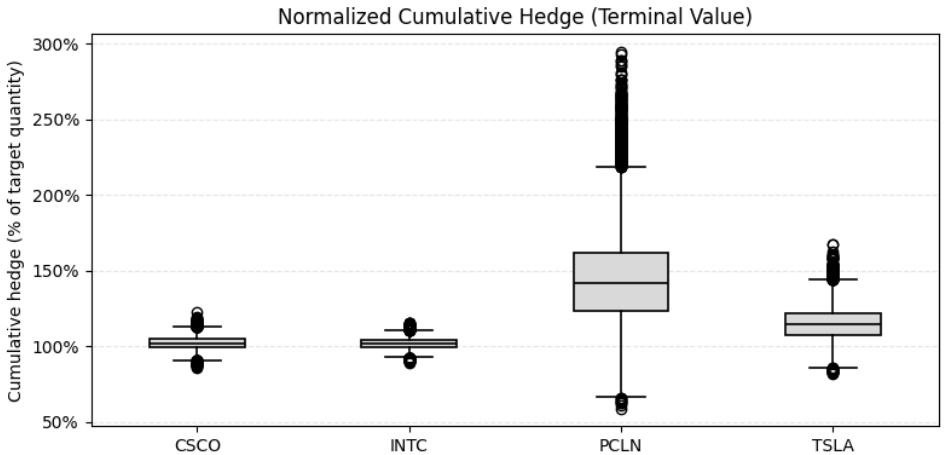

Figure: Cumulative normalized hedge (terminal).

All stocks except PCLN build inventory uniformly throughout the session and finish with a mean terminal inventory close to the target.

This result is consistent with the Santa Fe model's treatment of MOs. PCLN differs for two reasons:

- Late-session LOs at lower prices improve terminal mark-to-market, causing the policy to add size near the close and push mean inventory above target.
- PCLN's wider LO profile yields more dispersed fill levels and, consequently, more dispersed final inventories.

Remark. To avoid larger deviations from the target, one may increase the penalty factor. However, this might force a suboptimal MO allocation.

Experiments: Optimized Trading Policy

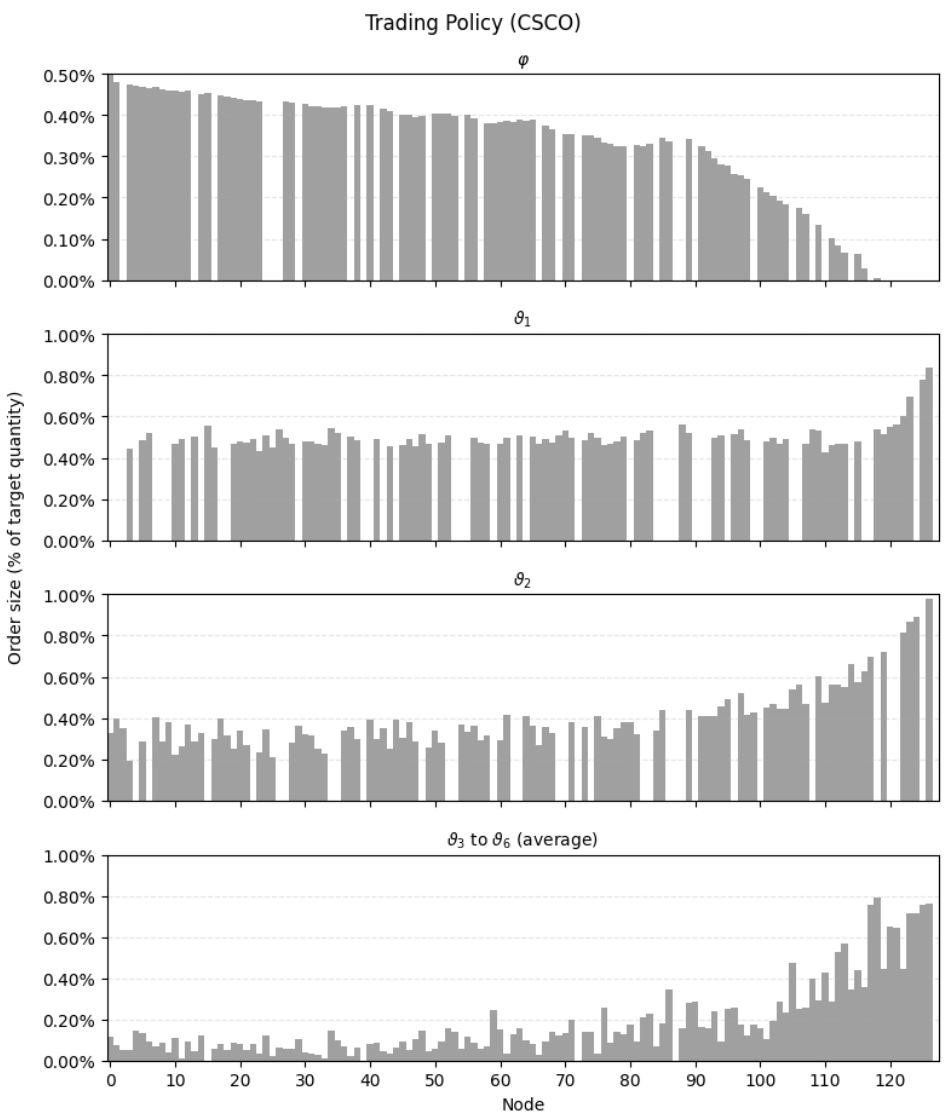

Figure: CSCO.

CSCO. The learned policy for CSCO begins with a balanced mix of MOs and LOs and shifts near the end toward an exclusively passive strategy:

- **Early phase ($n \approx 0$–90).** Small MO use ($0.5\% \to 0.3\%$); shallow LOs stable ($0.4\% - 0.5\%$); deeper placements begin rising ($0.1\% \to 0.3\%$).
- **Middle phase ($n \approx 90$–115).** MOs vanish; level-1 LOs stay near 0.5%; level-2 grow to 0.7%; deeper postings strengthen to $\sim 0.5\%$.
- **Late phase ($n \approx 115$–127).** No MOs; passive placements surge across all levels, reaching 0.8%–1.0%.

Experiments: Optimized Trading Policy

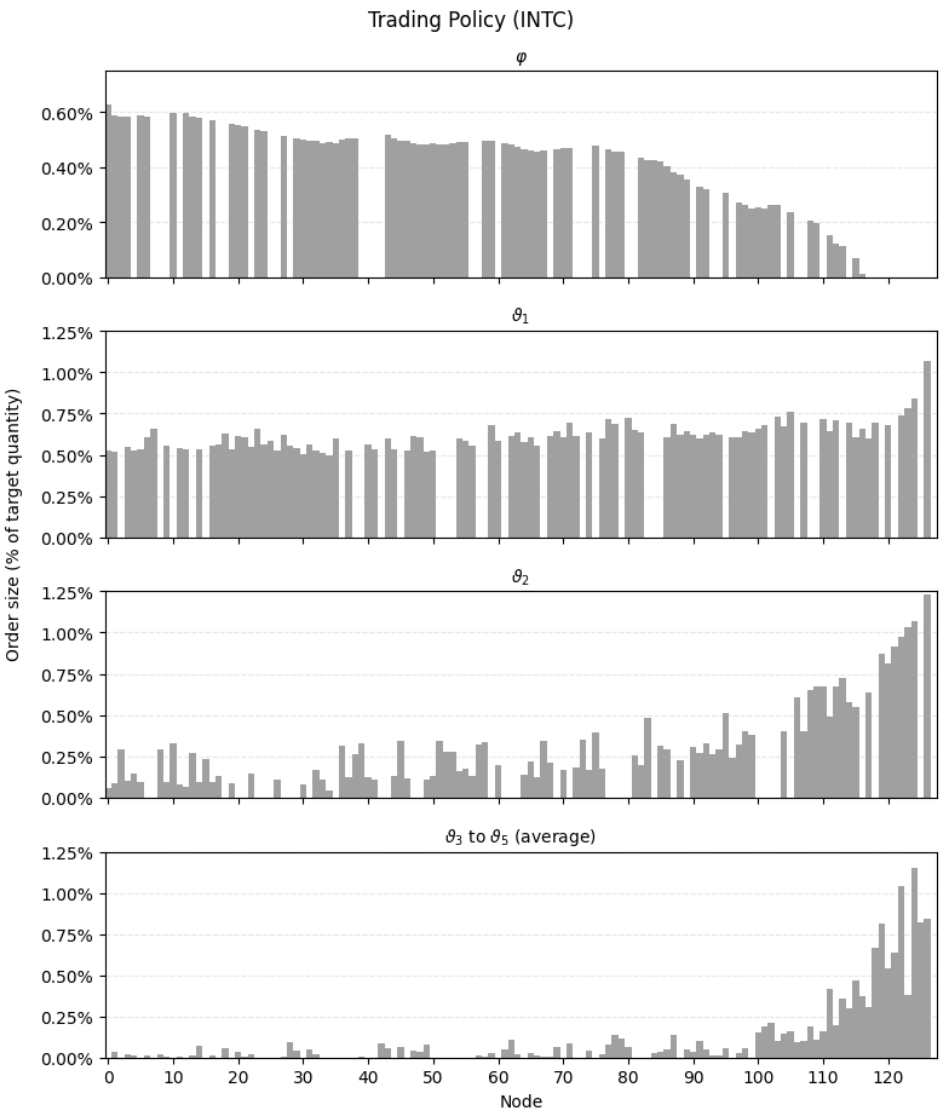

Figure: INTC.

INTC. The learned policy for INTC is characterized by early use of MOs and shallow LOs, followed by a gradual shift toward deeper-book posting:

- **Early phase ($n \approx 0$–80).** MOs stay in the $0.4\% - 0.6\%$ range; level-1 LOs rise ($0.5\% \to 0.75\%$); level-2 appears intermittently (up to 0.35%); deeper levels negligible.
- **Middle phase ($n \approx 50$–115).** MOs taper to zero; level-1 LOs remain near 0.75%; deeper-book allocations grow to about 0.75%.
- **Late phase ($n \approx 115$–127).** No MOs; level-1 LOs increase to $\sim 1.0\%$; deeper-level posting expands strongly to 1.0%–1.1%.

Experiments: Optimized Trading Policy

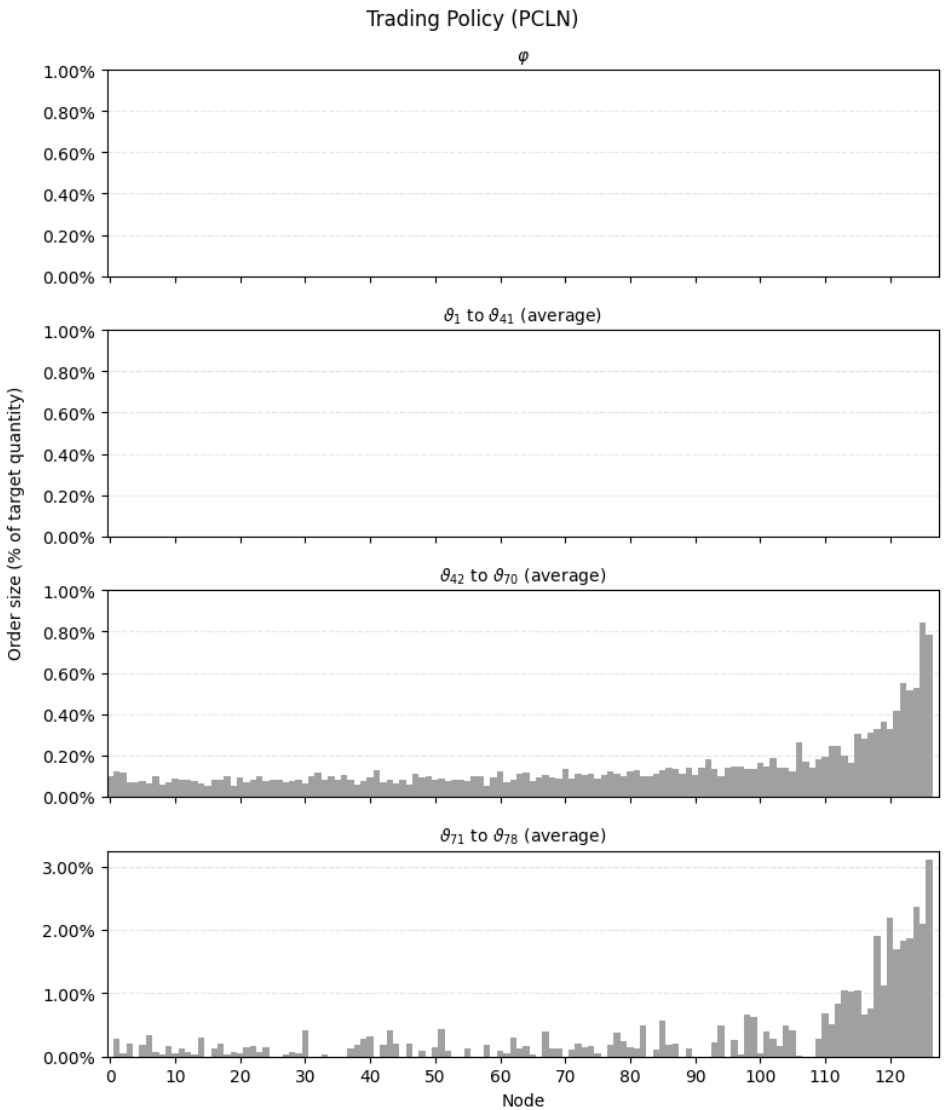

Figure: PCLN.

PCLN. The learned policy for PCLN is purely passive and allocated predominantly at deeper book levels, with order sizes increasing over time:

- **Early–mid phase ($n \approx 0$–110).** No MOs or shallow LOs; mid-book posting grows from $\sim 0.1\%$ to $\sim 0.2\%$, with deep levels used only sporadically.
- **Late phase ($n \approx 110$–127).** Still no MOs or shallow LOs; activity shifts deep into the book—mid-level posting rises to $\sim 0.8\%$ and deepest levels dominate, reaching $\sim 3.25\%$.

Experiments: Optimized Trading Policy

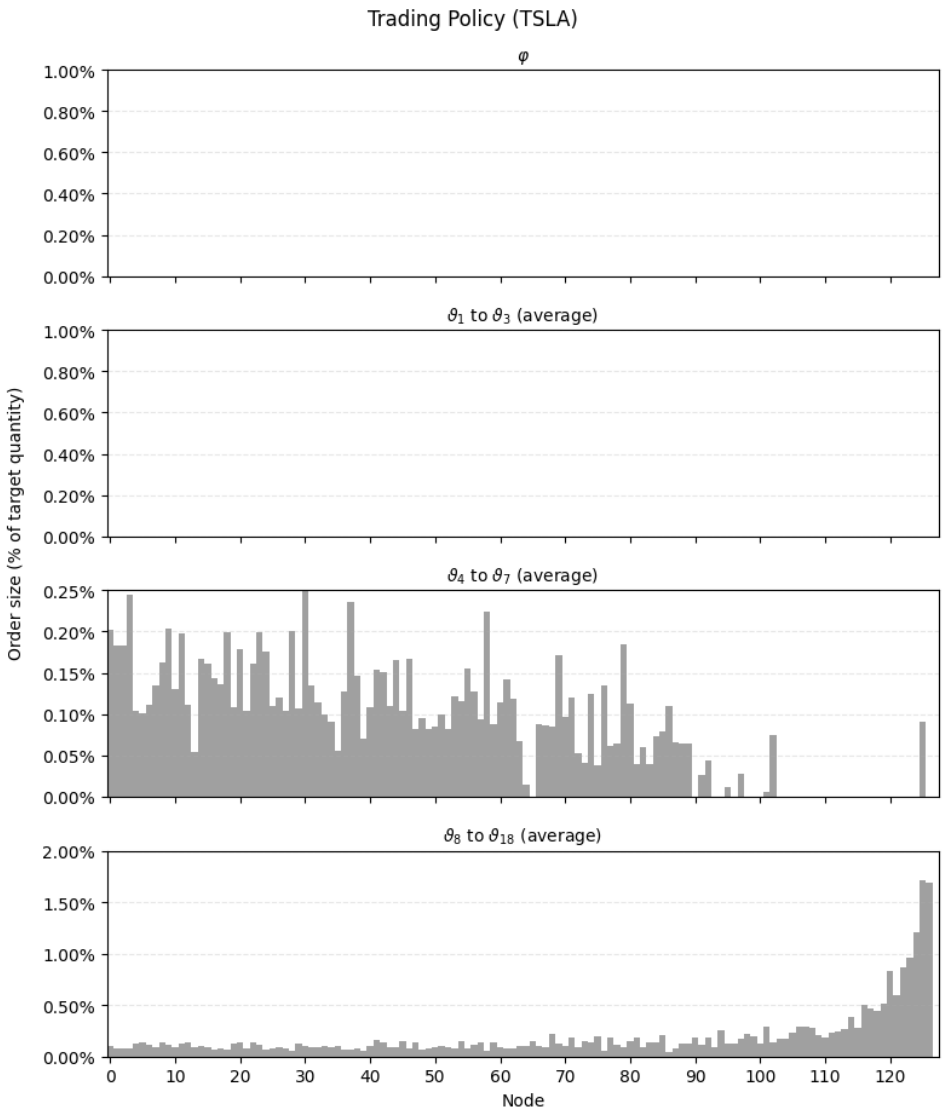

Figure: TSLA.

TSLA. The learned TSLA policy is purely passive, with a mid-book bias:

- **Early phase ($n \approx 0$–80).** No MOs or level-1 LOs; mid-book placement ranges from 0.05% to 0.25% of Q_0, while deep-book posting stays near 0.1%.
- **Middle phase ($n \approx 80$–100).** Still no MOs or level-1 activity; mid-book posting tapers off, and deep-book allocation increases to about 0.2% per step.
- **Late phase ($n \approx 100$–127).** Strategy remains fully passive; deep-book posting escalates sharply, reaching nearly 1.75% of Q_0 per step near the end.

Findings and Future Work

The estimated Santa Fe parameters explain cross-asset differences in the learned policies.

The location of mass in $\{\lambda^{(k)}\}$ and $\{\rho^{(k)}\}$ predicts the order-type mix: shallow concentration supports shallow LO posting and MO usage, while concentration at deeper levels favors purely passive, deep LO placement:

- **Tight-spread books suggest early aggressiveness:** CSCO and INTC exhibit high activity near the top of the book $\Rightarrow$ smaller cost differential between MOs and shallow LOs $\Rightarrow$ the learned policies begin with meaningful MO and shallow LO posting.
- **Wide books suggest purely passive deep placement:** PCLN exhibits activity concentrated deep in the book $\Rightarrow$ high costs for MO and shallow LO $\Rightarrow$ the learned strategy allocates to deep LOs.
- **Intermediate-structure books suggest passive allocation at mid levels:** TSLA sits between the extremes $\Rightarrow$ the policy allocates LOs only, with an initial mid-book bias.

In all cases, a **late-session ramp-up in LO posting** emerges, consistent with a higher probability of execution below the closing mid-price.

Remark. The full implementation, together with the calibrated Santa Fe parameter set, is publicly available at `https://github.com/mikegeiser/deep_hedging_VWAP`.

Findings and Future Work

Market model realism: improve intraday realism by allowing state-dependent order intensities and time/price-dependent volume distributions, or by using non-parametric, data-driven LOB simulators.

Compressed input features: a compact summary of raw per-side queue vectors could reduce dimensionality while preserving shape information. Options include:

- Parametric fits to arrival/cancellation profiles.
- Curve signatures that encode the book's geometry.

Additional state variables (e.g., posted/canceled liquidity) could further improve the information stream.

Endogenous market impact: within the DH framework, market impact can be implemented by simulating exogenous shocks and propagating the state with control-dependent transitions. For instance, suppose that we model prices via a GBM. On a grid $t_n = n\Delta t$ with i.i.d. $Z_{n+1} \sim \mathcal{N}(0,1)$, a control-dependent GBM takes the form

$$S_{n+1} = S_n \exp\Big\{ \big(\mu(\delta_n) - \tfrac{1}{2}\sigma^2(\delta_n)\big)\Delta t \; + \; \sigma(\delta_n)\sqrt{\Delta t}\, Z_{n+1} \Big\}, \tag{58}$$

where $\mu(\delta_n)$ and $\sigma(\delta_n)$ encode drift/volatility changes due to the agent's trade at step n. The simulated shocks $\{Z_n\}$ are used as training data; the model maps shocks and controls to terminal wealth. Similar techniques might be applied to a LOB parametric model.

Takeaways

- Deep Hedging is a flexible and powerful approach to hedge (optimize) a risk portfolio:

$$\underbrace{W^{\delta^{\boldsymbol{\theta}}}(Z)}_{\text{wealth}} = \underbrace{-Z}_{\text{liability}} + \underbrace{(H \circ \delta^{\boldsymbol{\theta}} \bullet X)_N}_{\text{trading}} - \underbrace{C_N(H \circ \delta^{\boldsymbol{\theta}})}_{\text{costs}}, \tag{59}$$

  with $\delta^{\boldsymbol{\theta}} = \{\boldsymbol{\delta}_n\}_{n=0}^{N-1}$ parametrized by neural networks

$$\underbrace{\boldsymbol{\delta}_n}_{\text{new action}} = f_n\Big(\theta_n;\ \underbrace{\mathbf{I}_0, \ldots, \mathbf{I}_n}_{\text{mkt information}},\ \underbrace{\boldsymbol{\delta}_{n-1}}_{\text{past action}}\Big). \tag{60}$$

  By choosing an appropriate Z, X, H, and C, we can model a wide range of practical applications and find an optimal trading policy $\delta^{\boldsymbol{\theta}^*}$ by solving

$$\pi_M(Z) = \inf_{\boldsymbol{\theta} \in \Theta_M} \rho\Big(W^{\delta^{\boldsymbol{\theta}}}(Z)\Big). \tag{61}$$

- In particular, we can hedge the VWAP execution risk in (Santa Fe) LOB markets using Deep Hedging. The learned policies are stock-specific and economically interpretable, with microstructure conditions shaping optimal market and limit order allocation.
- We can easily switch to more complex / realistic models by changing the market simulator and generating new training data.

Thank you for attending!

Any questions?

Appendix: Unbiasedness of RU Mini-Batches

We analyze the unbiasedness of the mini-batch (sub)gradients of the RU objective.

RU mini-batch (sub)gradients

Let $B \subseteq \{1, \ldots, J\}$ be a mini-batch sampled uniformly.

$$g_{\boldsymbol{\theta}}^{B}(\tau, \boldsymbol{\theta}) = \frac{1}{(1-\alpha)\,|B|} \sum_{b \in B} \mathbf{1}\{L^{(b)}(\boldsymbol{\theta}) > \tau\}\, \nabla_{\boldsymbol{\theta}} L^{(b)}(\boldsymbol{\theta}), \tag{62}$$

$$g_{\tau}^{B}(\tau, \boldsymbol{\theta}) = 1 - \frac{1}{(1-\alpha)\,|B|} \sum_{b \in B} \mathbf{1}\{L^{(b)}(\boldsymbol{\theta}) > \tau\}, \tag{63}$$

where $\nabla_{\boldsymbol{\theta}} L^{(b)}(\boldsymbol{\theta}) = -\nabla_{\boldsymbol{\theta}} W^{(b)}(\boldsymbol{\theta})$.

Note that in the particular case where B is the whole training set, we recover the **RU full-batch (sub)gradients**, denoted by $g_{\boldsymbol{\theta}}^{J}(\tau, \boldsymbol{\theta})$ and $g_{\tau}^{J}(\tau, \boldsymbol{\theta})$.

Conditioning on the training dataset, a standard calculation yields

$$\mathbb{E}\Big[g_{\boldsymbol{\theta}}^{B}(\tau, \boldsymbol{\theta})\Big] = g_{\boldsymbol{\theta}}^{J}(\tau, \boldsymbol{\theta}), \qquad \mathbb{E}\Big[g_{\tau}^{B}(\tau, \boldsymbol{\theta})\Big] = g_{\tau}^{J}(\tau, \boldsymbol{\theta}), \tag{64}$$

which means that SGD on the RU objective (jointly in $(\tau, \boldsymbol{\theta})$) is unbiased and converges under the usual conditions.

Appendix: Wealth Gradients

We analyze the differentiability of the map $\boldsymbol{\theta} \mapsto W^{(j)}(\boldsymbol{\theta})$, assuming the nonlinearity σ is (piecewise) C^1.

Because φ_n and $\vartheta_n^{(\ell)}$ are neural network outputs, each is a composition of affine maps and σ (element-wise). Since σ is (piecewise) C^1, it follows that $\nabla_{\boldsymbol{\theta}} \varphi_n$ and $\nabla_{\boldsymbol{\theta}} \vartheta_n^{(\ell)}$ are valid (sub)gradients, justifying their use in the next formulas:

$$\nabla_{\boldsymbol{\theta}} Q^{(j)}(\boldsymbol{\theta}) = \sum_{n=0}^{N-1} \left(\mathbf{1}_{\{\varphi_n < \Sigma_K(\widetilde{\mathbf{a}}_n)\}} \nabla_{\boldsymbol{\theta}} \varphi_n + \sum_{\ell=1}^{L} e_n^{(\ell)} \nabla_{\boldsymbol{\theta}} \vartheta_{n-1}^{(\ell)} \right), \tag{65}$$

$$\nabla_{\boldsymbol{\theta}} V^{(j)}(\boldsymbol{\theta}) = \sum_{n=0}^{N-1} \left(\sum_{k=1}^{K} \mathbf{1}_{\{\Sigma_{k-1}(\widetilde{\mathbf{a}}_n) < \varphi_n < \Sigma_k(\widetilde{\mathbf{a}}_n)\}} A_n^{(k)} \nabla_{\boldsymbol{\theta}} \varphi_n + \sum_{\ell=1}^{L} e_n^{(\ell)} B_{n-1}^{(\ell)} \nabla_{\boldsymbol{\theta}} \vartheta_{n-1}^{(\ell)} \right). \tag{66}$$

From (40) and the fact that $Q^{(\varphi,\vartheta)} P^{(\varphi,\vartheta)} = V^{(\varphi,\vartheta)}$, we have

$$\nabla_{\boldsymbol{\theta}} W^{(j)}(\boldsymbol{\theta}) = \left(\widetilde{S}_N^{(j)} - \frac{\widetilde{\varepsilon}_N^{(j)}}{2} \operatorname{sign}(Q^{(j)}(\boldsymbol{\theta}) - Q_0) \right) \nabla_{\boldsymbol{\theta}} Q^{(j)}(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} V^{(j)}(\boldsymbol{\theta}). \tag{67}$$

Therefore, the map $\boldsymbol{\theta} \mapsto W^{(j)}(\boldsymbol{\theta})$ is (piecewise) differentiable.

Appendix: Santa Fe Parameters Estimation

We describe the estimation formulas for the Santa Fe intensities.

Let M denote the number of trading days in the sample, and let $\mathrm{MO}^{\pm}$, $\mathrm{LO}_k^{\pm}$, and $\mathrm{CXL}_k^{\pm}$ denote, respectively, MOs submissions, LO arrivals at relative price $k \in \{1, \dots, K\}$, and LO cancellations at relative price k, on the buy $(+)$ and sell $(-)$ sides. Denote by $N_T^m(\cdot)$ the total count of a given event type on day $m \in \{1, \dots, M\}$ over the window $[0, T]$. The intensity estimators are

$$\gamma = \frac{1}{M} \frac{\sum_{m=1}^{M} \left(N_T^m(\mathrm{MO}^-) + N_T^m(\mathrm{MO}^+) \right)}{2T}, \tag{68}$$

$$\lambda^{(k)} = \frac{1}{M} \frac{\sum_{m=1}^{M} \left(N_T^m(\mathrm{LO}_k^-) + N_T^m(\mathrm{LO}_k^+) \right)}{2T}, \tag{69}$$

$$\rho^{(k)} = \frac{1}{M} \frac{\sum_{m=1}^{M} \left(N_T^m(\mathrm{CXL}_k^-) + N_T^m(\mathrm{CXL}_k^+) \right)}{2V^{(k)}T}, \tag{70}$$

where

$$V^{(k)} = \frac{1}{M} \frac{\sum_{m=1}^{M} \int_0^T \left(a_{t,m}^{(k)} + b_{t,m}^{(k)} \right) dt}{2T} \tag{71}$$

is the time–weighted average queue size at relative level k (per side).

Appendix: LOB State Update Rules

Given a current state $\mathbf{L}_t$, we apply the following rules to update the LOB state when a new event occurs at time $t + \Delta t$.

Remark. We show the update logic for the buy case. The update formulas for sell MOs, sell LOs, and cancellations of sell LOs mirror the buy case rules, with all price adjustments occurring in the opposite direction.

Buy market order of size $q > 0$

Let $i = \Sigma^{-1}(q, \mathbf{a}_t) - \Sigma^{-1}(0, \mathbf{a}_t)$. Then,

$$S_{t+\Delta t} = S_t + \tfrac{1}{2} i\,\epsilon, \qquad \varepsilon_{t+\Delta t} = \varepsilon_t + i\,\epsilon, \tag{72}$$

$$\mathbf{a}_{t+\Delta t} = \big(\min\{\mathrm{a}_t^{(1)}, (\Sigma_1(\mathbf{a}_t) - q)_+\}, \ldots, \min\{\mathrm{a}_t^{(K)}, (\Sigma_K(\mathbf{a}_t) - q)_+\}\big), \tag{73}$$

$$\mathbf{b}_{t+\Delta t} = \big(\underbrace{0, \ldots, 0}_{i \text{ times}}, \mathrm{b}_t^{(1)}, \ldots, \mathrm{b}_t^{(K-i)}\big). \tag{74}$$

Observe that $i > 0$ only if the order size fully consumes the best available liquidity.

Appendix: LOB State Update Rules

Buy limit order of size $q > 0$ at relative price $k \in \{1, \dots, K\}$

Let $i = (\Sigma^{-1}(0, \mathbf{a}_t) - k)_+$. Then,

$$S_{t+\Delta t} = S_t + \tfrac{1}{2} i\, \epsilon, \qquad \varepsilon_{t+\Delta t} = \varepsilon_t - i\, \epsilon \tag{75}$$

$$\mathbf{a}_{t+\Delta t} = \big(\mathrm{a}_t^{(1+i)}, \dots, \mathrm{a}_t^{(K)}, \underbrace{c^\infty, \dots, c^\infty}_{i \text{ times}}\big) \tag{76}$$

$$\mathbf{b}_{t+\Delta t} = \big(\mathrm{b}_t^{(1)}, \dots, \mathrm{b}_t^{(k-1)}, \mathrm{b}_t^{(k)} + q, \mathrm{b}_t^{(k+1)}, \dots, \mathrm{b}_t^{(K)}\big). \tag{77}$$

Note that $i > 0$ only when the order is priced within the bid-ask spread.

Cancellation of buy limit order of size $q > 0$ at relative price $k \in \{1, \dots, K\}$

Let $q' = \min\{q, \mathrm{b}_t^{(k)}\}$. Define $i = \Sigma^{-1}(q', \mathbf{b}_t) - \Sigma^{-1}(0, \mathbf{a}_t)$ if $k = \Sigma^{-1}(0, \mathbf{a}_t)$ and $i = 0$ otherwise. Then

$$S_{t+\Delta t} = S_t - \tfrac{1}{2} i\, \epsilon, \qquad \varepsilon_{t+\Delta t} = \varepsilon_t + i\, \epsilon, \tag{78}$$

$$\mathbf{a}_{t+\Delta t} = \big(\underbrace{0, \dots, 0}_{i \text{ times}}, \mathrm{a}_t^{(1)}, \dots, \mathrm{a}_t^{(K-i)}\big), \tag{79}$$

$$\mathbf{b}_{t+\Delta t} = \big(\mathrm{b}_t^{(1)}, \dots, \mathrm{b}_t^{(k-1)}, \mathrm{b}_t^{(k)} - q', \mathrm{b}_t^{(k+1)}, \dots, \mathrm{b}_t^{(K)}\big). \tag{80}$$

Observe that $i > 0$ whenever the best available liquidity is fully canceled.