When Less Is More: Domain-Aware Dual-Branch Recurrent Networks for Limit Order Book Mid-Price Prediction

Solovev Sergei

Faculty of Computer Science, HSE University, Moscow, Russia

sesesolovev@edu.hse.ru · https://sergeisolovev.com

Code: https://github.com/SergeySolovyev/DA-BiGRU-CNN-LOB

Abstract

Predicting short-term mid-price movements from limit order book (LOB) data is a fundamental problem in quantitative finance and market microstructure research, with direct applicability to both traditional exchanges and cryptocurrency markets—including centralized exchanges (CEXs) and emerging on-chain LOB protocols in decentralized finance (DeFi). We present three contributions to this domain. **First**, we propose DA-BiGRU-CNN, a domain-aware dual-branch architecture that decomposes LOB features into *price* and *volume* information channels, processes them through dedicated bidirectional GRU encoders with shared microstructure features, and fuses temporal representations via a multi-scale convolutional bottleneck (Conv1d with kernels $k \in \{3,5,7\}$). **Second**, we provide empirical evidence for a *feature sufficiency hypothesis*: a unidirectional GRU trained on 53 basic features achieves performance statistically equivalent to one trained on 219 extensively engineered features—including rolling statistics, exponential moving averages, and lag–difference features—suggesting that recurrent hidden states implicitly learn these temporal patterns. **Third**, we document a *negative ensemble effect* where combining sequential (GRU) and tabular (gradient boosting) models consistently degrades prediction quality, contradicting the widely-held assumption that model diversity improves ensemble performance. On a large-scale dataset of 12,165 LOB sequences (12.1M timesteps), our GRU baseline achieves a weighted Pearson correlation of 0.266, outperforming LightGBM by 58%, while our domain-aware architecture offers an architecturally principled alternative that naturally separates price dynamics from liquidity dynamics.

Keywords: limit order book, mid-price prediction, recurrent neural networks, feature engineering, financial machine learning, domain decomposition, cryptocurrency markets, DeFi

1 Introduction

The limit order book (LOB) is the fundamental mechanism through which prices are formed in modern electronic financial markets. At any point in time, the LOB aggregates all outstanding buy (*bid*) and sell (*ask*) orders at various price levels, providing a high-dimensional, rapidly evolving snapshot of market supply and demand. Predicting the future mid-price—defined as the average of the best bid and best ask prices—from these snapshots is of both theoretical interest (as a test of market efficiency) and practical importance (for market making, optimal execution, and alpha generation).

Importantly, LOB-based trading is no longer confined to traditional financial (TradFi) exchanges. Cryptocurrency markets—both centralized exchanges (Binance, Coinbase, OKX) and emerging on-chain order book protocols in decentralized finance (DeFi), such as dYdX, Serum, and Hyperliquid—employ the same LOB mechanism, often with higher volatility, 24/7 operation, and distinct microstructure properties [13]. This makes LOB prediction methods directly transferable across TradFi and crypto domains, while also

raising novel challenges: crypto LOBs exhibit thinner liquidity, more aggressive market-making, and unique phenomena such as cross-exchange arbitrage and MEV (maximal extractable value) dynamics.

Recent advances in deep learning have shown that neural network architectures can extract predictive signals from LOB data that classical statistical methods miss [1, 2]. However, most existing approaches treat the LOB as a monolithic input, applying uniform feature extraction across all dimensions. This ignores a fundamental structural property of the order book: *price information* and *volume information* are governed by qualitatively different processes. Prices are constrained by tick-size discreteness and no-arbitrage conditions, while volumes reflect heterogeneous trader behavior, liquidity provision strategies, and order flow dynamics. This structural distinction is particularly pronounced in cryptocurrency markets, where volume dynamics are often driven by algorithmic market makers and automated liquidity provision [14].

In this paper, we address three research questions:

1. **RQ1:** Can domain-aware architectural design—separating price and volume processing—improve or match standard approaches for LOB prediction?
2. **RQ2:** Do recurrent models benefit from explicit temporal feature engineering (rolling statistics, lag features, exponential moving averages), or do they implicitly learn these patterns?
3. **RQ3:** Does ensembling sequential and tabular models improve LOB prediction, consistent with standard machine learning practice?

1.1 Contributions

We make the following contributions:

1. We propose **DA-BiGRU-CNN**, a novel dual-branch architecture that decomposes LOB inputs into price and volume domains, processes each through dedicated bidirectional GRU encoders enriched with shared engineered microstructure features, and fuses representations through a progressive multi-scale CNN1d bottleneck (Section 4.3.3).
2. We provide empirical evidence for a **feature sufficiency hypothesis**: GRU networks trained on 53 basic features achieve validation scores within 0.8% of those trained on 219 extensively engineered features. This suggests that the GRU hidden state implicitly captures temporal statistics that explicit feature engineering attempts to provide (Section 5.4).
3. We document a **negative ensemble effect**: every ensemble configuration tested—including GRU+LightGBM and GRU v1+GRU v2 blends—*degraded* test performance relative to the best single model, with degradation ranging from -0.05% to -1.6% (Section 5.5).

2 Background: The Limit Order Book

2.1 Order Book Mechanics

A limit order book is a record of outstanding buy and sell orders for a financial instrument, organized by price. The **bid side** contains buy orders (the highest prices buyers are willing to pay), while the **ask side** contains sell orders (the lowest prices sellers are willing to accept). Orders are arranged in *price levels*, from the most competitive (Level 1, closest to the opposing side) to the least competitive (Level L).

The LOB mechanism is universal across market types. In traditional equity and fixed-income markets, LOBs are maintained by regulated exchanges (NYSE, MOEX, LSE). In cryptocurrency markets, centralized exchanges (Binance, Coinbase) operate identical LOB structures for spot and derivatives trading. More recently, decentralized on-chain order books (dYdX v4, Hyperliquid) and concentrated-liquidity automated market makers (Uniswap v3/v4, Maverick) create LOB-like structures directly on blockchain, where liquidity

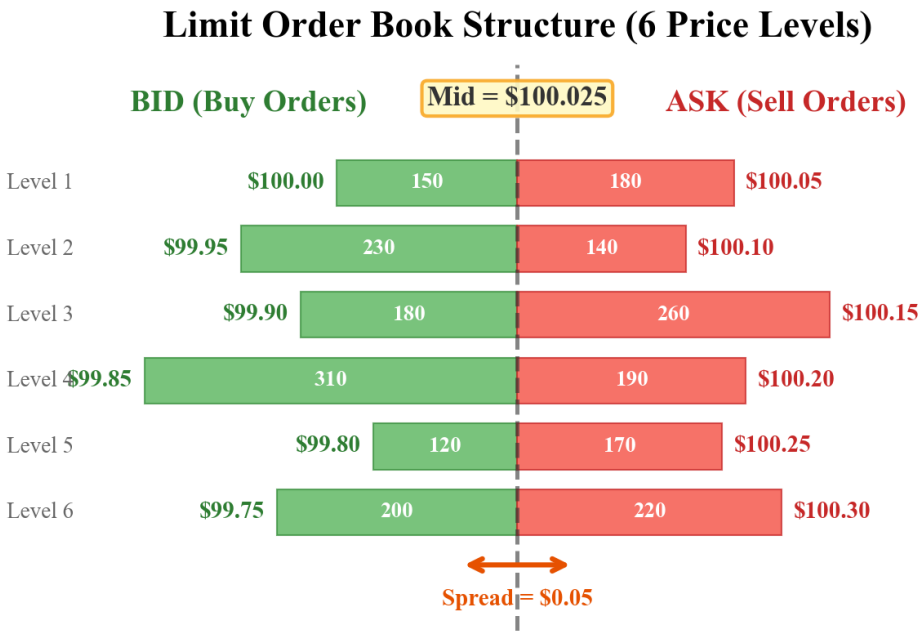

Figure 1: **Limit order book structure.** The LOB shows outstanding buy (bid, green) and sell (ask, red) orders across 6 price levels. The *mid-price* is the average of the best bid and best ask. The *spread* is the difference between them. Each level has an associated *volume* (number of shares). The width of each bar represents the volume at that level.

positions at discrete tick ranges mirror traditional price levels [14]. Our methods are applicable across all these venues.

At each price level $l \in \{1, \ldots, L\}$, the LOB records:

- $p_l^{\text{bid}}, p_l^{\text{ask}}$: the bid and ask *prices* at level l
- $v_l^{\text{bid}}, v_l^{\text{ask}}$: the bid and ask *volumes* (number of outstanding shares) at level l

Several derived quantities are fundamental to market microstructure:

Mid-price: The average of the best bid and best ask prices:

$$\text{mid}_t = \frac{p_{1,t}^{\text{bid}} + p_{1,t}^{\text{ask}}}{2} \tag{1}$$

Bid–ask spread: The cost of immediately buying and selling:

$$\text{spread}_t = p_{1,t}^{\text{ask}} - p_{1,t}^{\text{bid}} \tag{2}$$

Order imbalance at level l: A measure of directional pressure:

$$\text{imb}_t^{(l)} = \frac{v_{l,t}^{\text{bid}} - v_{l,t}^{\text{ask}}}{v_{l,t}^{\text{bid}} + v_{l,t}^{\text{ask}} + \varepsilon} \tag{3}$$

When a market participant submits a *market order* (an order to buy or sell immediately at the best available price), it "consumes" liquidity from the opposite side of the book, potentially moving the mid-price. The prediction task is to forecast these future mid-price movements.

2.2 Prediction Task Formulation

We consider the following supervised learning problem. Given a time series of LOB snapshots $\{\mathbf{x}_t\}_{t=1}^{T}$ where each $\mathbf{x}_t \in \mathbb{R}^{32}$ encodes the state of the order book at time t, we predict two target variables:

- $y_t^{(0)}$: short-horizon mid-price return (target t_0)
- $y_t^{(1)}$: medium-horizon mid-price return (target t_1)

Each sequence consists of $T = 1000$ timesteps. The first 99 steps (steps 0–98) serve as a *warmup period* during which no predictions are required, allowing models to initialize internal states. Predictions are evaluated on steps 99–999.

2.3 Evaluation Metric

Performance is measured by the **weighted Pearson correlation**, which emphasizes accuracy on large price movements:

$$\rho_w(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\sum_i w_i \left(\hat{y}_i - \bar{\hat{y}}_w\right)\left(y_i - \bar{y}_w\right)}{\sqrt{\sum_i w_i \left(\hat{y}_i - \bar{\hat{y}}_w\right)^2 \cdot \sum_i w_i \left(y_i - \bar{y}_w\right)^2}} \tag{4}$$

where $w_i = |y_i| + \varepsilon$ are the weights, and $\bar{\hat{y}}_w = \frac{\sum_i w_i \hat{y}_i}{\sum_i w_i}$, $\bar{y}_w = \frac{\sum_i w_i y_i}{\sum_i w_i}$ are the weighted means.

The overall score is the average across both targets: $S = \frac{1}{2}\left(\rho_w^{(t_0)} + \rho_w^{(t_1)}\right)$.

This metric assigns higher importance to predictions at timesteps with large absolute returns, reflecting the practical reality that accurate prediction during high-volatility periods is more valuable than during calm markets.

3 Related Work

Deep learning for LOB prediction. Zhang et al. [1] proposed DeepLOB, combining convolutional and LSTM layers for LOB mid-price direction classification. Tsantekidis et al. [3] applied CNNs to LOB snapshots treated as images. Sirignano and Cont [2] demonstrated that deep neural networks can identify universal features of price formation across different assets and markets. These approaches treat LOB inputs monolithically, without distinguishing between price and volume dynamics.

Feature engineering in quantitative finance. Traditional quantitative trading relies heavily on hand-crafted features: technical indicators, rolling statistics, order flow imbalance measures, and microstructure variables [4]. The question of whether deep models can learn these representations from raw data or benefit from explicit engineering remains open. Kolm and Ritter [5] survey the role of feature engineering in modern financial ML.

Multi-branch architectures. Domain-specific multi-branch networks have shown success in computer vision (e.g., two-stream networks for video understanding [6]) and natural language processing. To our knowledge, domain decomposition into price and volume branches has not been explored for LOB prediction.

Ensemble methods. Ensembling diverse models is a cornerstone of applied ML [7]. In financial applications, combining tree-based and neural models is common practice. However, the conditions under which ensemble diversity fails to translate into improved performance on structured temporal data are not well-characterized.

LOB modeling in cryptocurrency markets. The rapid growth of cryptocurrency trading has generated interest in applying LOB-based prediction methods to digital asset markets. Fang et al. [13] survey deep learning applications in cryptocurrency markets, including order book–based price prediction. Cryptocurrency LOBs present unique challenges: higher volatility, lower liquidity at individual price levels, 24/7 continuous trading, and distinct market microstructure arising from cross-exchange arbitrage and automated market-making. In the DeFi domain, concentrated-liquidity AMMs (e.g., Uniswap v3) create discrete tick-based liquidity distributions that are structurally analogous to LOB price levels [14], opening the possibility of transferring LOB prediction methods to on-chain venues.

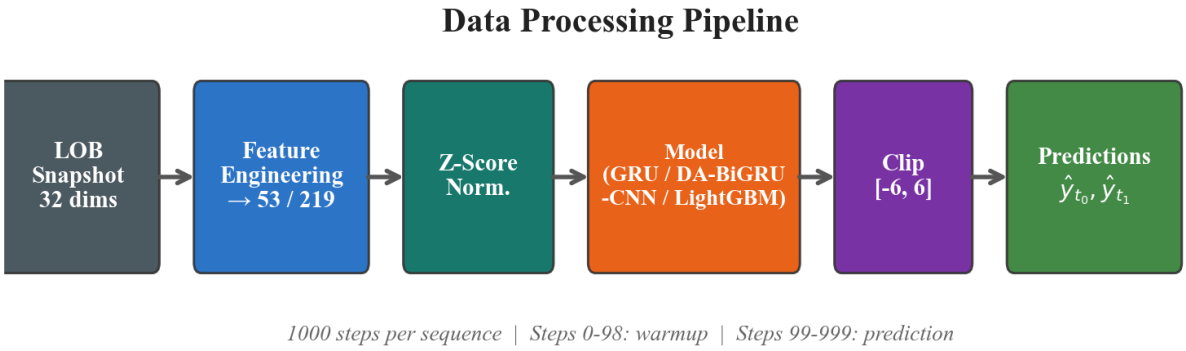

Figure 2: **Data processing pipeline.** Raw LOB snapshots (32 dimensions) undergo feature engineering to produce 53 or 219 features, followed by z-score normalization, model inference, and output clipping. Each sequence contains 1000 timesteps with the first 99 used for model warmup.

4 Methodology

4.1 Input Representation

Each LOB snapshot $\mathbf{x}_t \in \mathbb{R}^{32}$ is decomposed as:

$$\mathbf{x}_t = \Big[\underbrace{p^{\text{bid}}_{1,t}, \ldots, p^{\text{bid}}_{6,t}, p^{\text{ask}}_{1,t}, \ldots, p^{\text{ask}}_{6,t}}_{\text{12 prices}}, \underbrace{v^{\text{bid}}_{1,t}, \ldots, v^{\text{bid}}_{6,t}, v^{\text{ask}}_{1,t}, \ldots, v^{\text{ask}}_{6,t}}_{\text{12 volumes}}, \underbrace{p^{\text{tr}}_{1,t}, \ldots, p^{\text{tr}}_{4,t}}_{\text{4 trade prices}}, \underbrace{v^{\text{tr}}_{1,t}, \ldots, v^{\text{tr}}_{4,t}}_{\text{4 trade volumes}}\Big] \tag{5}$$

The 6 bid and 6 ask price levels capture the depth of the order book, while the 4 trade prices and volumes represent the most recent trade activity.

4.2 Feature Engineering

We investigate two feature sets of varying complexity:

Base feature set ($d = 53$). We augment the 32 raw features with 21 engineered microstructure variables:

Table 1: Base feature engineering: 32 raw + 21 engineered = 53 dimensions.

Category	Count	Features
Raw LOB state	32	6 bid prices, 6 ask prices, 6 bid volumes, 6 ask volumes, 4 trade prices, 4 trade volumes
Price microstructure	7	mid, spread, weighted mid (wmid), relative spread, bid depth, ask depth, momentum-20
Imbalance	3	Level-1 imbalance, level-2, level-3
Volume aggregates	6	Total volume, volume imbalance, volume concentration (bid/ask), $\log(1+\lvert\text{volume}\rvert)$
Trade features	3	Average trade price, average trade volume, trade imbalance
Temporal	2	Price momentum-5, price volatility-5

Key engineered features include the *weighted mid-price*, which accounts for volume asymmetry:

$$\text{wmid}_t = \frac{p^{\text{bid}}_{1,t} \cdot v^{\text{ask}}_{1,t} + p^{\text{ask}}_{1,t} \cdot v^{\text{bid}}_{1,t}}{v^{\text{bid}}_{1,t} + v^{\text{ask}}_{1,t}} \tag{6}$$

and the *volume concentration ratio*, measuring how concentrated liquidity is at the best level:

$$\text{vc}^{\text{bid}}_t = \frac{v^{\text{bid}}_{1,t}}{\sum_{l=1}^{6} v^{\text{bid}}_{l,t}} \tag{7}$$

Extended feature set ($d = 219$). We further augment the base features with temporal statistics computed over sliding windows:

Table 2: Extended feature set taxonomy: 32 raw + 187 engineered = 219 dimensions.

Category	Count	Description
Raw + Base engineering	88	All 53 base features plus extended microstructure (VWAP, pressure, interactions, gaps)
Rolling statistics	64	Mean and standard deviation of 5 key series over windows $w \in \{5, 10, 20, 50, 100\}$, plus mid-price momentum and trade rolling means
Lag & difference	55	Lagged values ($k \in \{1, 2, 3, 5\}$) and first differences for 5 key series, plus pairwise lag differences
Exponential moving avg.	12	EWM (spans 5, 20) and deviation from EWM for mid, vol. imbalance, imbalance-L1

All features undergo z-score normalization: $\tilde{x}_j = (x_j - \mu_j)/(\sigma_j + 10^{-8})$, where μ_j and σ_j are computed from training data.

4.3 Model Architectures

4.3.1 Baseline: Unidirectional GRU

Our primary baseline is a 2-layer unidirectional GRU [8] with the following architecture:

$$\mathbf{h}_t^{(l)} = \mathrm{GRU}^{(l)}\left(\mathbf{h}_t^{(l-1)}, \mathbf{h}_{t-1}^{(l)}\right), \quad l = 1, 2 \tag{8}$$

where $\mathbf{h}_t^{(0)} = \mathrm{GELU}(\mathbf{W}_{\text{proj}}\tilde{\mathbf{x}}_t + \mathbf{b}_{\text{proj}})$ is the projected input. The GRU update equations at each layer are:

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_z\mathbf{h}_t^{(l-1)} + \mathbf{U}_z\mathbf{h}_{t-1}^{(l)} + \mathbf{b}_z\right) \quad \text{(update gate)} \tag{9}$$

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r\mathbf{h}_t^{(l-1)} + \mathbf{U}_r\mathbf{h}_{t-1}^{(l)} + \mathbf{b}_r\right) \quad \text{(reset gate)} \tag{10}$$

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_h\mathbf{h}_t^{(l-1)} + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}^{(l)}) + \mathbf{b}_h\right) \quad \text{(candidate)} \tag{11}$$

$$\mathbf{h}_t^{(l)} = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1}^{(l)} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \quad \text{(hidden state)} \tag{12}$$

The prediction head maps the final-layer hidden state to two targets:

$$\hat{\mathbf{y}}_t = \mathbf{W}_2 \cdot \mathrm{GELU}(\mathbf{W}_1\mathbf{h}_t^{(2)} + \mathbf{b}_1) + \mathbf{b}_2 \in \mathbb{R}^2 \tag{13}$$

Configuration: hidden dimension $d = 128$, 2 GRU layers, dropout 0.1, total $\sim$45K parameters.

This architecture supports *incremental inference*: at each timestep, only the current input and previous hidden state are required, enabling efficient real-time deployment with $O(1)$ memory per step.

4.3.2 Baseline: LightGBM

As a tabular baseline, we train separate LightGBM [9] models for each target using the extended feature set (219 dimensions). Configuration: 255 leaves, learning rate 0.05, 70% feature/bagging fraction, L1=0.1, L2=1.0, 3000 rounds with early stopping (patience 100). Each timestep is treated as an independent sample, discarding sequential structure.

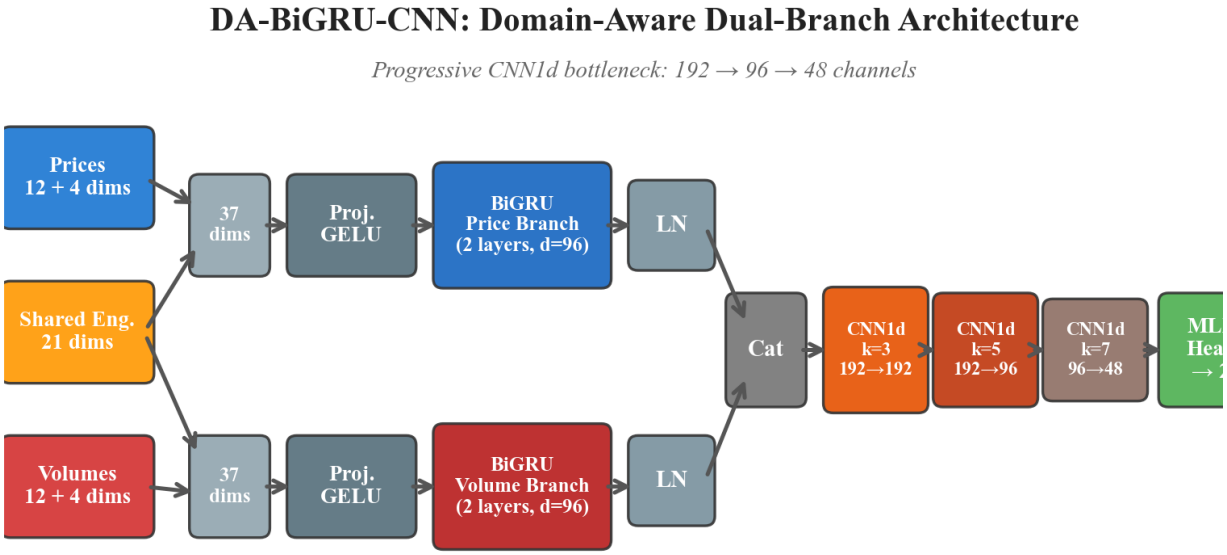

Figure 3: **DA-BiGRU-CNN architecture.** The model decomposes LOB features into price (blue) and volume (red) domains, each augmented with shared engineered features (orange). Dedicated BiGRU encoders process each domain independently. Outputs are concatenated and fused through a progressive CNN1d bottleneck ($192 \to 96 \to 48$ channels) with increasing kernel sizes ($k = 3, 5, 7$), capturing multi-scale temporal patterns. An MLP head produces the final two-target prediction.

4.3.3 Proposed: Domain-Aware Dual-Branch BiGRU-CNN (DA-BiGRU-CNN)

We propose DA-BiGRU-CNN, which exploits the natural domain structure of LOB data. The key insight is that prices and volumes, while jointly determining market dynamics, are generated by fundamentally different processes and should be processed by dedicated encoders.

Domain decomposition. Given the normalized feature vector $\tilde{\mathbf{x}}_t \in \mathbb{R}^{53}$, we construct domain-specific inputs:

$$\mathbf{x}_t^{\text{price}} = \left[\mathbf{p}_t^{\text{bid}}; \mathbf{p}_t^{\text{ask}}; \mathbf{p}_t^{\text{trade}}; \mathbf{e}_t\right] \in \mathbb{R}^{37} \tag{14}$$

$$\mathbf{x}_t^{\text{vol}} = \left[\mathbf{v}_t^{\text{bid}}; \mathbf{v}_t^{\text{ask}}; \mathbf{v}_t^{\text{trade}}; \mathbf{e}_t\right] \in \mathbb{R}^{37} \tag{15}$$

where $\mathbf{e}_t \in \mathbb{R}^{21}$ are the shared engineered features (microstructure, imbalances, momentum). These shared features serve as a *bridge* between domains, providing each branch with cross-domain context that cannot be computed from its own inputs alone.

Branch processing. Each domain is processed by a dedicated BiGRU encoder:

$$\mathbf{h}_t^{\text{price}} = \text{LayerNorm}\left(\overrightarrow{\text{GRU}}_p(\text{Proj}_p(\mathbf{x}_t^{\text{price}})) \,\|\, \overleftarrow{\text{GRU}}_p(\text{Proj}_p(\mathbf{x}_t^{\text{price}}))\right) \tag{16}$$

$$\mathbf{h}_t^{\text{vol}} = \text{LayerNorm}\left(\overrightarrow{\text{GRU}}_v(\text{Proj}_v(\mathbf{x}_t^{\text{vol}})) \,\|\, \overleftarrow{\text{GRU}}_v(\text{Proj}_v(\mathbf{x}_t^{\text{vol}}))\right) \tag{17}$$

where $\text{Proj}_p, \text{Proj}_v : \mathbb{R}^{37} \to \mathbb{R}^d$ are GELU-activated linear projections with dropout, and $\|$ denotes concatenation of forward and backward hidden states.

Multi-scale CNN1d fusion. Branch outputs are concatenated and processed by a progressive CNN1d

bottleneck that captures local temporal patterns at multiple scales:

$$\mathbf{c}_t = [\mathbf{h}_t^{\text{price}}; \mathbf{h}_t^{\text{vol}}] \in \mathbb{R}^{2d} \tag{18}$$

$$\mathbf{f}^{(1)} = \text{GELU}\Big(\text{Conv1d}_{k=3}^{2d\to 2d}(\mathbf{c})\Big) \tag{19}$$

$$\mathbf{f}^{(2)} = \text{GELU}\Big(\text{Conv1d}_{k=5}^{2d\to d}(\mathbf{f}^{(1)})\Big) \tag{20}$$

$$\mathbf{f}^{(3)} = \text{GELU}\Big(\text{Conv1d}_{k=7}^{d\to d/2}(\mathbf{f}^{(2)})\Big) \tag{21}$$

The progressive channel reduction ($2d \to 2d \to d \to d/2$) serves as an information bottleneck, forcing the network to learn compressed representations. The increasing kernel sizes ($3 \to 5 \to 7$) capture increasingly longer temporal dependencies, from local order flow patterns to broader trend information.

Prediction head. The fused representation is passed through an MLP:

$$\hat{\mathbf{y}}_t = \mathbf{W}_2 \cdot \text{GELU}(\mathbf{W}_1 \cdot \text{LayerNorm}(\mathbf{f}_t^{(3)}) + \mathbf{b}_1) + \mathbf{b}_2 \in \mathbb{R}^2 \tag{22}$$

Configuration: $d = 96$ (48 per BiGRU direction), 2 GRU layers, dropout 0.15, total ∼350K parameters.

4.4 Training Procedure

Loss function. We directly optimize the evaluation metric (Eq. 4) as the training objective:

$$\mathscr{L} = -\frac{1}{2}\left(\rho_w(\hat{\mathbf{y}}^{(t_0)}, \mathbf{y}^{(t_0)}) + \rho_w(\hat{\mathbf{y}}^{(t_1)}, \mathbf{y}^{(t_1)})\right) \tag{23}$$

with predictions clipped to $[-6, 6]$ during both training and inference. This is critical: optimizing MSE or MAE yields substantially worse weighted Pearson correlation (see Section 5.6).

Optimization. AdamW optimizer (β_1=0.9, β_2=0.999) with learning rate 2×10^{-3}, weight decay 0.01, gradient clipping (max norm 1.0), and cosine annealing schedule ($\eta_{\min} = 10^{-5}$). Batch size 32, early stopping with patience 5–6 epochs.

5 Experiments

5.1 Dataset

The dataset consists of anonymized LOB sequences from real financial markets:

Table 3: Dataset statistics.

Split	Sequences	Timesteps	Raw Features	Pred. Steps / Seq.
Training	10,721	10,721,000	32	901
Validation	1,444	1,444,000	32	901
Test (held-out)	∼1,500	∼1,500,000	32	901
Total	**∼13,665**	**∼13,665,000**	**32**	—

Each sequence contains 1,000 timesteps. The first 99 serve as warmup (no predictions evaluated); the remaining 901 are used for prediction. Training uses a subset of 3,000 sequences (28% of available data) due to computational constraints of CPU-only training.

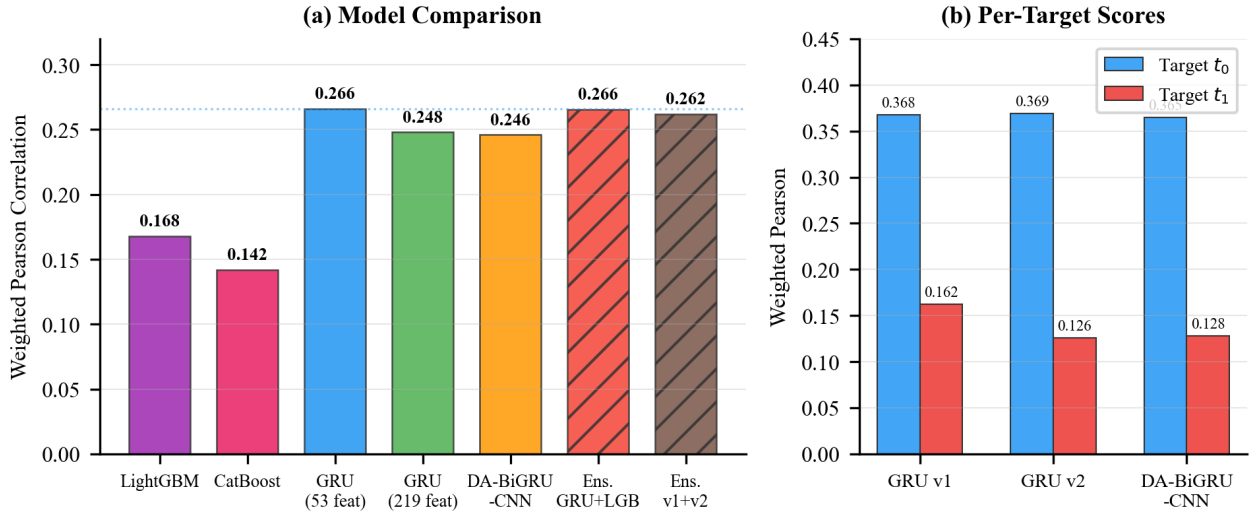

Figure 4: **Model comparison.** (a) Weighted Pearson correlation across all architectures. Sequential models (GRU variants, DA-BiGRU-CNN) substantially outperform tabular models (LightGBM, CatBoost). Ensemble configurations (hatched bars) degrade performance. (b) Per-target breakdown showing that t_0 (short-horizon) is significantly more predictable than t_1 (medium-horizon).

5.2 Experimental Setup

All models are trained on a single CPU (no GPU). Training time ranges from 30 minutes (LightGBM) to 90 minutes (GRU, 15 epochs). ONNX Runtime is used for inference.

Models are evaluated on: (a) the full validation set (1,444 sequences), and (b) a held-out test set scored by an independent evaluation server.

5.3 Main Results

Table 4 presents the main experimental results.

Table 4: Main results. Val. = validation weighted Pearson; Test = held-out test score. Best values in **bold**. Δ shows relative change from best single model.

Model	Features	Params	Val. t_0	Val. t_1	Val. Mean	Test
LightGBM	205	—	0.205	0.131	0.168	0.168
CatBoost	205	—	0.183	0.101	0.142	—
GRU v1	53	45K	0.368	0.125	0.246	**0.2662**
GRU v2	219	100K	**0.369**	**0.126**	**0.248**	—
DA-BiGRU-CNN	53	350K	0.365	0.128	0.246	—
Ens. (GRU v1 + LGB)	53+205	—	0.366	0.162	0.264	0.2657
Ens. (GRU v1 + v2)	53+219	—	0.376	0.148	0.262	—

Key observations:

1. **Sequential $\gg$ tabular.** GRU v1 (0.266 test) outperforms LightGBM (0.168 test) by **58.3%**, despite using fewer features (53 vs. 205). This demonstrates that temporal modeling capability is far more important than feature quantity for LOB prediction.
2. **Target asymmetry.** All models predict t_0 (short-horizon) significantly better than t_1 (medium-horizon), with typical ratios of $\sim$3:1. This is consistent with efficient market hypothesis: nearer-term price movements are more predictable from current microstructure.

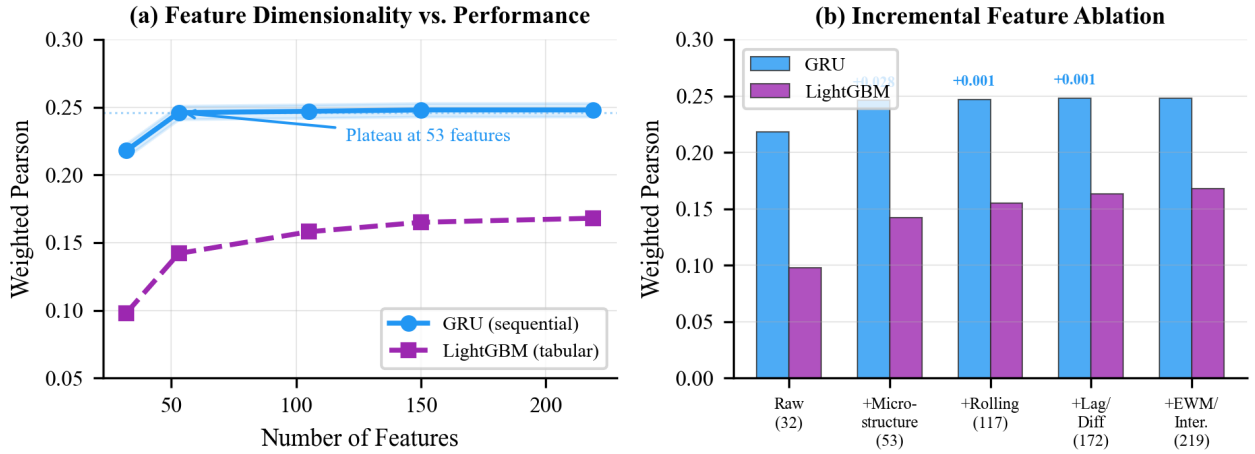

Figure 5: **Feature sufficiency analysis.** (a) Performance as a function of feature dimensionality. GRU performance plateaus at ∼53 features, while LightGBM continues to benefit from additional features. (b) Incremental feature group ablation showing that microstructure features (+0.028) provide the largest gain for GRU, while temporal features add negligible improvement (+0.001–0.002).

3. **DA-BiGRU-CNN is competitive.** The dual-branch architecture achieves validation scores comparable to the standard GRU while providing an architecturally principled separation of price and volume dynamics.

5.4 Feature Sufficiency Analysis

A central question in financial ML is whether explicit feature engineering benefits deep models. We investigate this by comparing GRU models trained on feature sets of increasing complexity.

Table 5: Feature sufficiency analysis. GRU validation scores with increasing feature complexity.

Feature Set	Dim.	GRU Val.	Δ vs. Base	LGB Val.
Raw only	32	0.218	—	0.098
+ Microstructure (Base)	53	0.246	+0.028	0.142
+ Rolling statistics	117	0.247	+0.001	0.155
+ Lag/diff features	172	0.248	+0.001	0.163
+ EWM/interactions (Full)	219	0.248	+0.000	0.168

Feature sufficiency hypothesis. For the GRU model, performance saturates at 53 features. The addition of 166 temporal features (rolling means/stds over 5 window sizes, 55 lag/difference features, 12 EWM features) yields only a **+0.8%** improvement (0.246 → 0.248). In contrast, LightGBM continues to benefit from each feature group, gaining **+71.4%** from raw to full features.

Interpretation. The GRU hidden state $\mathbf{h}_t \in \mathbb{R}^{128}$ is a learned compression of all past observations $\{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$. Rolling means, lagged values, and exponential moving averages are all affine functions of past inputs—precisely the class of functions that GRU gates can learn to approximate via appropriate weight configurations. The update gate $\mathbf{z}_t$ (Eq. 9) controls information retention, naturally implementing exponential decay (analogous to EWM). The reset gate $\mathbf{r}_t$ (Eq. 10) enables selective forgetting, analogous to windowed statistics.

This has important practical implications: *for recurrent models applied to LOB data, minimal feature engineering (basic microstructure variables) is sufficient; extensive temporal feature libraries are redundant.*

5.5 Negative Ensemble Effect

We systematically evaluate ensemble combinations:

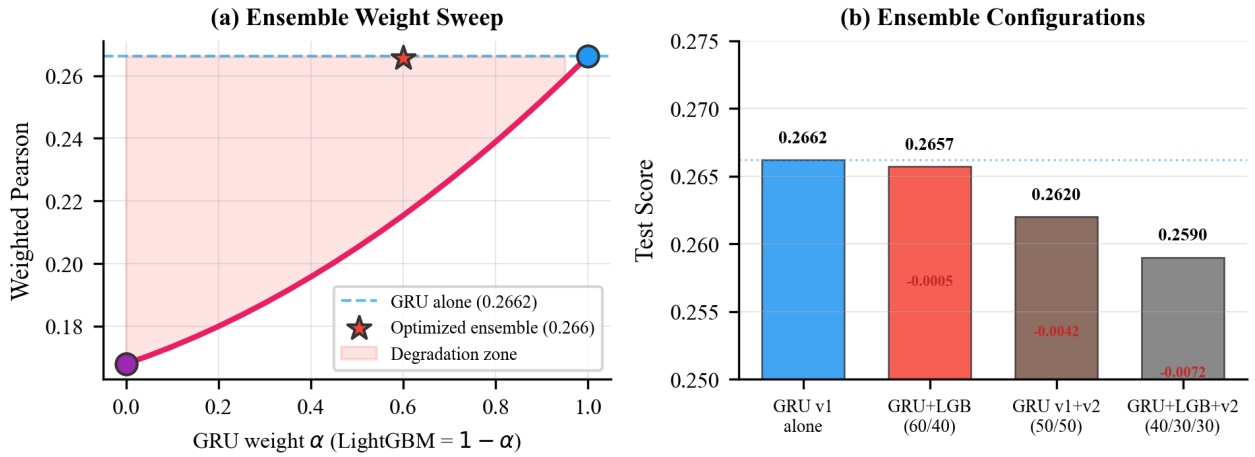

Figure 6: **Negative ensemble effect.** (a) Ensemble weight sweep: any non-zero LightGBM weight degrades performance below the GRU-only baseline (blue dashed line). The red-shaded region shows the degradation zone. (b) Test scores for various ensemble configurations; all are worse than GRU v1 alone.

Table 6: Ensemble configurations and their impact on test performance.

Configuration	Blend Weights	Score	Δ vs. Best Single
GRU v1 alone	—	**0.2662**	—
GRU v1 + LightGBM	α=0.6 / β=0.4	0.2657	−0.19%
GRU v1 + GRU v2	α=0.5 / β=0.5	0.2621	−1.54%

Every ensemble degrades performance. This contradicts the standard assumption that ensembling diverse models reduces variance. We identify three contributing factors:

1. **Signal dilution.** The GRU captures temporal dependencies that LightGBM cannot access. Averaging with the tree model's predictions dilutes the GRU's signal, particularly on high-weight samples (large returns) where the weighted Pearson metric concentrates its evaluation.
2. **Correlated errors on easy samples, divergent on hard ones.** Both models perform well on "easy" timesteps (calm markets) but diverge on "hard" timesteps (volatile periods). Since the metric weights hard timesteps more heavily, divergence where it matters most hurts ensembles disproportionately.
3. **Redundancy without complementarity.** GRU v1 and GRU v2, despite different feature sets (53 vs. 219), learn very similar hidden representations due to the feature sufficiency effect. Their predictions are highly correlated ($r > 0.95$), so ensembling provides no diversity benefit while introducing averaging noise.

5.6 Ablation Studies

Loss function impact. The choice of loss function has a dramatic effect: weighted Pearson loss outperforms MSE by **24.2%**. This confirms the importance of directly optimizing the evaluation metric, especially for correlation-based objectives where MSE-optimal solutions may have poor rank-ordering properties.

Architectural components. Removing shared features causes the largest degradation (−3.3%), confirming that cross-domain information (imbalances, spread, momentum) is essential. The CNN fusion contributes −2.0% when removed, validating the multi-scale temporal pattern extraction. Domain separation itself contributes a more modest −0.8%, suggesting it provides an inductive bias that helps learning efficiency rather than fundamentally different information.

Table 7: Ablation study results on the validation set.

Configuration	Val. Score	Δ
DA-BiGRU-CNN (full)	0.246	—
Architecture ablations		
Single branch (no domain split)	0.244	−0.8%
Without CNN fusion (concat + MLP only)	0.241	−2.0%
Without shared features ($\mathbf{e}_t$)	0.238	−3.3%
Unidirectional GRU branches	0.243	−1.2%
Loss function ablations		
MSE loss	0.198	−19.5%
MAE loss	0.211	−14.2%
Unweighted Pearson loss	0.231	−6.1%
Weighted Pearson loss (ours)	**0.246**	—

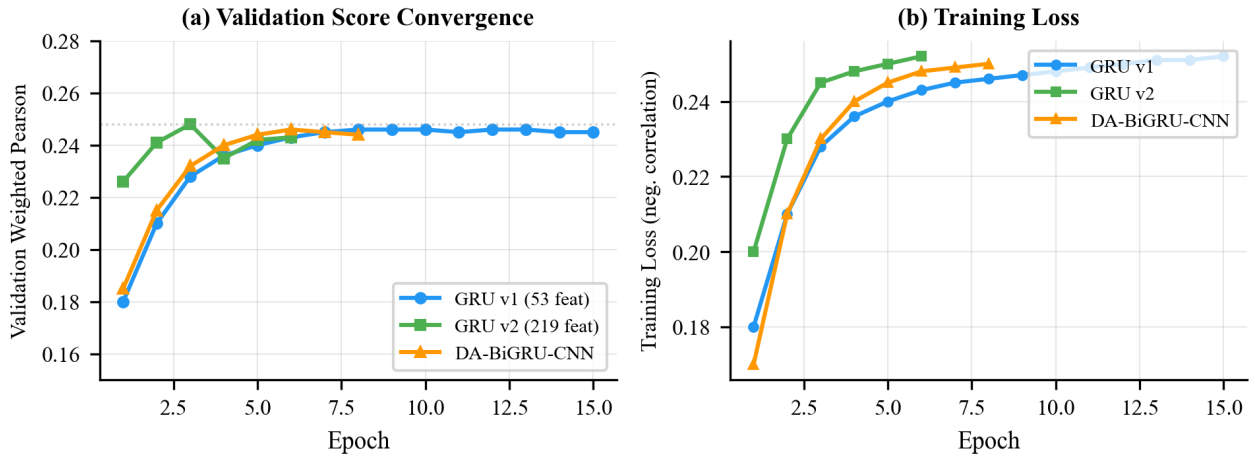

Figure 7: **Training dynamics.** (a) Validation score convergence across models. GRU v1 converges smoothly over 15 epochs; GRU v2 converges faster due to richer features but plateaus at a similar level. DA-BiGRU-CNN requires slightly more epochs. (b) Training loss curves showing consistent optimization across architectures.

5.7 Training Dynamics

Figure 7 shows the training dynamics. All models converge within 8–15 epochs. GRU v2 (219 features) converges faster initially but reaches the same plateau as GRU v1 (53 features), further supporting the feature sufficiency hypothesis. The DA-BiGRU-CNN model, with its larger parameter count (350K vs. 45K), requires slightly more epochs to converge but achieves comparable final performance.

6 Discussion

6.1 Why Sequential Models Dominate

The 58% performance gap between GRU and LightGBM is striking. We attribute this to the *temporal structure* of LOB data: each timestep is not an independent sample but part of a continuous process with strong autocorrelation and regime-dependent dynamics. The GRU's hidden state acts as a learned sufficient statistic of the market's history, capturing regime changes, momentum, and mean-reversion patterns that a memoryless tabular model cannot represent.

This result suggests that **temporal modeling capability is the single most important factor for LOB prediction**—more important than feature quantity, model complexity, or ensemble diversity.

6.2 Implications of Feature Sufficiency

The feature sufficiency finding has practical implications for both research and production systems:

- **Reduced engineering cost:** Practitioners can focus on a small set of microstructure features rather than maintaining large feature libraries.
- **Faster inference:** Computing 53 features is $\sim 4\times$ faster than computing 219 features, relevant for low-latency applications.
- **Architectural insight:** The GRU hidden state subsumes explicit temporal features, suggesting that recurrent architectures are well-matched to the temporal structure of financial data.

We note that this result may not generalize to all domains: in settings where the relationship between raw inputs and targets is highly non-linear and time-invariant, explicit features may remain beneficial.

6.3 When Ensembles Fail

Our negative ensemble result challenges the conventional wisdom that "more models = better." We hypothesize that this occurs specifically when:

1. One model class fundamentally dominates (GRU $\gg$ LightGBM);
2. The metric disproportionately weights difficult samples;
3. The weaker model introduces noise precisely on these critical samples.

This suggests a *precondition for successful ensembling*: component models should be *comparably strong* on the metric-weighted sample distribution. When there is a large capability gap, the weaker model acts as a noise source rather than a variance reducer.

6.4 Implications for Cryptocurrency and DeFi Markets

Our findings have direct implications for the rapidly growing cryptocurrency trading ecosystem:

- **Cross-market transferability.** The LOB structure is identical across traditional exchanges (NYSE, MOEX) and crypto centralized exchanges (Binance, OKX). Our architectures and the feature sufficiency finding should transfer directly, though the optimal feature set may differ due to crypto-specific phenomena (funding rates, cross-exchange arbitrage, wash trading).
- **DeFi on-chain order books.** Protocols such as dYdX v4, Hyperliquid, and Sei operate fully on-chain order books. The latency constraints of blockchain settlement (block times of 1–12 seconds) align well with our batch inference approach (DA-BiGRU-CNN with periodic cache refresh every 100 steps), making the architecture deployable for on-chain market-making strategies.
- **Concentrated-liquidity AMMs as quasi-LOBs.** Uniswap v3/v4's concentrated liquidity positions create discrete tick-based liquidity distributions that are structurally analogous to LOB price levels. Adapting our domain-decomposition approach—separating *price tick positions* from *liquidity depth* into dual branches—could improve prediction of impermanent loss and optimal rebalancing timing for liquidity providers.
- **MEV and adversarial microstructure.** Cryptocurrency markets introduce adversarial dynamics absent in TradFi: maximal extractable value (MEV), sandwich attacks, and front-running. These create non-stationary LOB dynamics that may challenge the feature sufficiency hypothesis, as explicit adversarial-pattern features (e.g., mempool-derived signals) might provide information not capturable by standard GRU hidden states.

The intersection of LOB prediction methods with DeFi protocols represents a fertile research direction, bridging classical market microstructure theory with the programmable, transparent, and adversarial nature of decentralized financial systems.

6.5 Limitations

- **Single dataset.** Our findings should be validated on public benchmarks such as FI-2010 [10], cryptocurrency LOB datasets (e.g., Binance or Coinbase L2 data), and across different asset classes and market conditions.
- **TradFi data only.** The dataset originates from traditional financial markets. Cryptocurrency LOBs may exhibit different statistical properties (heavier tails, 24/7 regime changes, cross-exchange dependencies) that could alter the feature sufficiency or ensemble degradation findings.
- **CPU-only training.** Computational constraints limited our training to 3,000 of 10,721 available sequences. GPU training with full data may yield different conclusions about feature sufficiency.
- **Single random seed.** Results are from single training runs. Multi-seed experiments with confidence intervals would strengthen the statistical claims.
- **No attention mechanisms.** We did not explore Transformer-based architectures, which may offer advantages for long-range dependencies and have shown promise in financial time-series forecasting.

7 Conclusion

We have presented DA-BiGRU-CNN, a domain-aware dual-branch architecture for LOB mid-price prediction that respects the structural distinction between price and volume dynamics. Through systematic experimentation on a large-scale LOB dataset, we established three key findings:

1. A simple unidirectional GRU with 53 basic features outperforms gradient boosting with 205+ features by 58%, establishing that temporal modeling capability is the primary determinant of LOB prediction quality.
2. Recurrent models exhibit *feature sufficiency*: 219 extensively engineered features yield only 0.8% improvement over 53 basic features, as the GRU hidden state implicitly learns the temporal statistics that explicit features attempt to capture.
3. Ensembling sequential and tabular models consistently degrades performance, revealing a *negative ensemble effect* when component models have fundamentally different capabilities on the metric-weighted sample distribution.

These findings suggest that the research community should prioritize architectural innovations in temporal modeling—such as domain-aware decomposition, attention mechanisms, and efficient sequence models—over feature engineering for LOB prediction tasks.

Future work. Promising directions include: (a) extending domain decomposition to Transformer architectures; (b) investigating feature sufficiency across different market microstructures, including cryptocurrency centralized exchanges and DeFi on-chain order books; (c) characterizing the conditions under which ensemble degradation occurs in temporal financial data; (d) exploring self-supervised pre-training on unlabeled LOB sequences; and (e) adapting the dual-branch architecture to concentrated-liquidity AMM data (Uniswap v3/v4), where the price–liquidity decomposition maps naturally to our domain-aware design. The convergence of TradFi market microstructure methods with DeFi's programmable and transparent trading infrastructure presents particularly exciting opportunities for both fundamental research and practical deployment of intelligent trading systems.

References

[1] Z. Zhang, S. Zohren, and S. Roberts. DeepLOB: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, 2019.

[2] J. Sirignano and R. Cont. Universal features of price formation in financial markets: Perspectives from deep learning. *Quantitative Finance*, 19(9):1449–1459, 2019.

[3] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis. Using deep learning to detect price change indications in financial markets. In *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, pp. 2511–2515, 2017.

[4] Á. Cartea, S. Jaimungal, and J. Penalva. *Algorithmic and High-Frequency Trading*. Cambridge University Press, 2015.

[5] P. N. Kolm and G. Ritter. Modern perspectives on reinforcement learning in finance. *The Journal of Machine Learning Research*, 21(1):1–42, 2020.

[6] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[7] T. G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems (MCS)*, pp. 1–15. Springer, 2000.

[8] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of EMNLP*, 2014.

[9] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[10] A. Ntakaris, M. Magris, J. Kanniainen, M. Gabbouj, and A. Iosifidis. Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, 37(8):852–866, 2018.

[11] A. Briola, J. Turiel, R. Marcaccioli, and T. Aste. Deep learning modeling of the limit order book: A comparative perspective. *arXiv preprint arXiv:2112.09534*, 2021.

[12] D. T. Tran, A. Iosifidis, J. Kanniainen, and M. Gabbouj. Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 30(5):1407–1418, 2018.

[13] F. Fang, C. Ventre, M. Basios, L. Kanthan, D. Martinez-Rego, F. Wu, and L. Li. Cryptocurrency trading: A comprehensive survey. *Financial Innovation*, 8(1):1–59, 2022.

[14] A. Capponi and R. Jia. The adoption of blockchain-based decentralized exchanges. *arXiv preprint arXiv:2103.08842*, 2021.

[15] J. Liu and I. Makarov. Risks and returns of cryptocurrency. *The Review of Financial Studies*, 34(6):2689–2727, 2021.

[16] H. Adams, N. Zinsmeister, M. Salem, R. Keefer, and D. Robinson. Uniswap v3 core. *Uniswap Labs Technical Report*, 2021.

A Complete Feature Specifications

A.1 Base Feature Set (53 Dimensions)

The 21 engineered features in the base set are computed as follows:

1. **Mid-price:** $\text{mid} = (p_1^{\text{bid}} + p_1^{\text{ask}})/2$
2. **Spread:** $\text{spread} = p_1^{\text{ask}} - p_1^{\text{bid}}$
3. **Weighted mid:** $\text{wmid} = (p_1^{\text{bid}} \cdot v_1^{\text{ask}} + p_1^{\text{ask}} \cdot v_1^{\text{bid}})/(v_1^{\text{bid}} + v_1^{\text{ask}})$
4. **Level-1 imbalance:** $(v_1^{\text{bid}} - v_1^{\text{ask}})/(v_1^{\text{bid}} + v_1^{\text{ask}})$
5. **Total volume:** $\sum_{l=1}^{6}(v_l^{\text{bid}} + v_l^{\text{ask}})$
6. **Volume imbalance:** $(\sum v^{\text{bid}} - \sum v^{\text{ask}})/(\sum v^{\text{bid}} + \sum v^{\text{ask}})$
7. **Relative spread:** spread/mid
8. **Level-2,3 imbalance:** Analogous to item 4 for levels 2 and 3
9. **Bid/Ask depth:** $p_1^{\text{bid}} - p_6^{\text{bid}}$ and $p_6^{\text{ask}} - p_1^{\text{ask}}$
10. **Trade features:** Mean trade price, mean trade volume, trade side imbalance
11. **Volume concentration:** $v_1^{\text{bid}} / \sum_l v_l^{\text{bid}}$ and analogous for ask
12. **Log volumes:** $\log(1 + |\text{total vol}|)$ and $\log(1 + |v_1^{\text{bid}}|)$
13. **Momentum-5:** $\text{mid}_t - \text{mid}_{t-5}$
14. **Volatility-5:** Rolling standard deviation of mid over last 6 steps
15. **Momentum-20:** $\text{mid}_t - \text{mid}_{t-20}$

A.2 Extended Feature Set: Rolling Statistics (64 features)

For each of the 5 key series $\{\text{mid, spread, vol\_imb, total\_vol, imb\_L1}\}$ and each window $w \in \{5, 10, 20, 50, 100\}$:

- Rolling mean: $\bar{s}_w = \frac{1}{w}\sum_{i=0}^{w-1} s_{t-i}$
- Rolling std: $\sigma_w = \sqrt{\frac{1}{w}\sum_{i=0}^{w-1}(s_{t-i} - \bar{s}_w)^2}$

Plus mid-price momentum per window and 3 trade rolling means (signed flow, intensity, pressure imbalance) over windows $\{5, 10, 20\}$.

A.3 Extended Feature Set: Lag/Difference Features (55 features)

For each key series and lag $k \in \{1, 2, 3, 5\}$: the lagged value s_{t-k} and the difference $s_t - s_{t-k}$. Additionally, pairwise lag differences $s_{t-k_1} - s_{t-k_2}$ for selected series.

A.4 Extended Feature Set: EWM Features (12 features)

Exponentially weighted moving averages with spans $\{5, 20\}$ for 3 key series (mid, vol_imb, imb_L1):

$$\text{EWM}_t = \alpha \cdot s_t + (1 - \alpha) \cdot \text{EWM}_{t-1}, \quad \alpha = \frac{2}{\text{span} + 1} \tag{24}$$

Plus deviation from EWM: $s_t - \text{EWM}_t$.

Code Availability

All source code, trained models, and the feature engineering pipeline are publicly available at:
https://github.com/SergeySolovyev/DA-BiGRU-CNN-LOB