A BENCHMARK STUDY FOR LIMIT ORDER BOOK (LOB) MODELS AND TIME SERIES FORECASTING MODELS ON LOB DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a comprehensive benchmark to evaluate the performance of deep learning models on limit order book (LOB) data. Our work makes four significant contributions: (i) We evaluate existing LOB models on a proprietary futures LOB dataset to examine the transferability of LOB model performance between various assets; (ii) We are the first to benchmark existing LOB models on the mid-price return forecasting (MPRF) task. (iii) We present the first benchmark study to evaluate SOTA time series forecasting models on the MPRF task to bridge the two fields of general-purpose time series forecasting and LOB time series forecasting; and (iv) we propose an architecture of convolutional **C**ross-**V**ariate **M**ixing **L**ayers (**CVML**) as an add-on to any deep learning multivariate time series model to significantly enhance MPRF performance on LOB data. Our empirical results highlight the value of our benchmark results on our proprietary futures LOB dataset, demonstrating a performance gap between the commonly used open-source stock LOB dataset and our futures dataset. Furthermore, the results demonstrate that LOB-aware model design is essential for achieving optimal prediction performance on LOB datasets. Most importantly, our results show that our proposed CVML architecture brings about an average improvement of 244.9% to various time series models' mid-price return forecasting performance.

1 INTRODUCTION

The Limit Order Book (LOB) serves as the order-matching engine for all exchanges, providing the most granular market time series data for analysis (Weber, 1999; Ntakaris et al., 2018). As a universal data format across markets and assets (e.g., stocks and futures), LOB contains high-resolution macroeconomic information crucial for asset price predictions (Chan et al., 2005; Harris & Panchapagesan, 2005; Large, 2007; Avellaneda & Stoikov, 2008; Roşu, 2009; Eisler et al., 2012).

Two primary research tracks in LOB data analysis are **M**id-**P**rice **T**rend **P**rediction (**MPTP**) and **M**id-**P**rice **R**eturn **F**orecasting (**MPRF**). Inspired by the success of deep learning in domains like natural language processing (Vaswani et al., 2023) and computer vision (He et al., 2016), researchers have proposed deep learning models for LOB analysis and general time series predictions (Zeng et al., 2023; Liu et al., 2023; Nie et al., 2022; Wang et al., 2024).

Despite these advancements, the field lacks comprehensive benchmark studies comparing model performance. While Prata et al. (2024) conducted a benchmark study for MPTP, it was limited to stock datasets, excluding other asset types. For MPRF, which encompasses both LOB modeling and time series prediction, no benchmark exists that evaluates deep learning-based LOB models and state-of-the-art time series forecasting models on LOB data. This work aims to address these gaps in both MPTP and MPRF tasks.

To address these limitations, we conduct a comprehensive study on both MPTP and MPRF tasks. For MPTP, we evaluate state-of-the-art LOB models using the open-source FI-2010 stock dataset and our proprietary futures dataset, CHF-2023, measuring each model's F1 scores across five different prediction horizons. We also conduct an ablation study to assess the predictive power of various LOB feature types, including basic LOB features, time-insensitive features, and time-sensitive features.

For the MPRF task, we evaluate both LOB-specific models and state-of-the-art time series forecasting models on FI-2010. We assess their forecasting performance using three metrics: Mean-Square Error (MSE), Coefficient of Determination (R^2), and Pearson Correlation (Corr). Additionally, we propose a novel neural architecture featuring cross-variate mixing layers, called **CVML**, designed to enhance the forecasting performance of existing time series models on LOB data.

Our experiments reveal several key findings. For MPTP, we observe inconsistent model performance rankings between the stock and futures LOB datasets, with models generally performing worse on the futures data. This suggests limited generalizability of current LOB model architectures and highlights the distinct underlying characteristics of LOB data from different assets. Our ablation study confirms that each feature subset contributes unique predictive power, underscoring the importance of comprehensive feature selection in LOB modeling. For MPRF, our results demonstrate that LOB models incorporating LOB-specific inductive bias significantly outperform general-purpose time series forecasting models. The latter, lacking LOB-specific design considerations, show minimal forecasting power when directly applied to LOB datasets. Our proposed CVML module significantly improves the forecasting capabilities of all benchmarked time series models, extending their predictive power to the more complex and noisy LOB time series data.

Contributions. We summarize our contributions as follows:

- We evaluate existing LOB models on the MPTP task using a proprietary futures LOB dataset, CHF-2023, to assess the transferability of models designed for stock LOB data across asset classes.
- We present the first benchmark on the MPRF task in the literature.
- We pioneer benchmarking state-of-the-art time series forecasting models on the MPRF task, bridging the gap between general-purpose and LOB-specific time series forecasting.
- We propose a novel Cross-Variate Mixing Layer (CVML) as an add-on to existing time series models, enhancing their MPRF performance by an average of 244.9%.

These contributions advance LOB modeling across different assets and tasks while providing a new tool to enhance time series model performance on LOB data.

Organization. Section 2 includes related work. Section 3 discusses background information on the MPTP and MPRF tasks. Section 4 details our benchmark studies and our proposed CVML architecture. Section 5 discusses the prediction performance gap between the stock and futures datasets in MPTP. Section 6 includes concluding remarks.

2 Related Works

Price Trend Prediction Surveys. Several comprehensive benchmark surveys have examined deep learning applications in price trend prediction (Ozbayoglu et al., 2020; Sezer et al., 2020; Jiang, 2021), each with a distinct focus. Jiang (2021) emphasize reproducibility, analyzing model architectures, evaluation metrics, and implementations in stock price and market index prediction studies from 2017 to 2019. A follow-up study by Kumbure et al. (2022) extend this analysis to datasets and input variables commonly used in stock market predictions. Hu et al. (2021) review 86 papers on stock and foreign exchange price prediction, while other surveys (Rundo et al., 2019; Mintarya et al., 2023) compare machine learning and deep learning methods in stock market prediction, concluding that deep learning approaches generally offer superior accuracy. Nti et al. (2020) broaden the scope beyond technical analysis, reviewing 122 papers from 2007 to 2018 covering technical, fundamental, and combined analyses. Additionally, Shah et al. (2019) evaluate the real-world applicability of models through backtesting performance. Notably, these surveys do not include benchmarks for prediction models on Limit Order Book (LOB) data. The most relevant work is a benchmark study by Prata et al. (2024) on mid-price trend prediction models using LOB data. However, our work addresses three key limitations of their study: We use a proprietary dataset with a time range 200 times larger than the open-source dataset they employed. We benchmark models on both stock and futures datasets, whereas they focused solely on stocks. We extend our analysis to include the mid-price forecasting problem (a regression task), benchmarking both LOB models and state-of-the-art time series forecasting models, in addition to the mid-price trend prediction task (a classification problem) they addressed. These enhancements allow our study to provide a more comprehensive and diverse evaluation of LOB-based prediction models across different assets and problem types.

Time Series Forecasting Models. The success of deep learning in natural language processing and computer vision has significantly influenced time series forecasting, with deep learning models

becoming predominant in this field. Transformer-based architectures, in particular, have emerged as the leading approach for multivariate time series forecasting (Nie et al., 2022; Liu et al., 2023). However, recent developments have shown that models based on linear layers (Zeng et al., 2023; Wang et al., 2024; Chen et al., 2023; Oreshkin et al., 2020; Challu et al., 2023; Zhang et al., 2022) can achieve comparable performance to transformer-based models. While convolutional neural networks (O'shea & Nash, 2015; Wu et al., 2023; Franceschi et al., 2019) and recurrent networks (Hochreiter & Schmidhuber, 1997; Lai et al., 2018; Franceschi et al., 2019) have also been applied to time series forecasting, their performance generally lags behind that of transformers and linear-based architectures. Notably, there has been limited overlap between general time series forecasting and Limit Order Book (LOB) time series analysis. To date, no comprehensive benchmarking of state-of-the-art time series forecasting models on the complex LOB time series data has been conducted. To address this gap, our paper selects four state-of-the-art time series forecasting models, encompassing both transformer-based and linear architectures. Our aim is to bridge the divide between general-purpose time series forecasting and the more specialized field of LOB time series forecasting, providing insights into the applicability and performance of these models on LOB data.

3 BACKGROUND

Limit Order Book (LOB). Global exchanges use matching engines to pair orders from bid and ask sides of market participants. The Limit Order Book is the essential data structure organizing these orders, reflecting market supply and demand. Three common order types exist: **1)** *Market orders* are requests to buy or sell a specified number of shares at the best available price, usually executed immediately. **2)** *Limit orders* are requests to buy or sell a specified number of shares at a specified price, often queued for matching due to price constraints. **3)** *Cancel orders* are requests to withdraw previously submitted limit orders.

We model the Limit Order Book (LOB) as a time series $\mathbb{L} \in \mathbb{R}^{4\times L\times T}$, where $\mathbb{L}(t) \in \mathbb{R}^{4\times L}$ represents the LOB at time step $t \in [0,T]$. Specifically, $\mathbb{L}(t) = \{P_i^{\text{bid}}(t), Q_i^{\text{bid}}(t), P_i^{\text{ask}}(t), Q_i^{\text{ask}}(t)\}_{i\in[0,L]}$, with T observed time steps and L levels on each side of the order book. $P^{\text{bid/ask}}i(t)$ and $Q_i^{\text{bid/ask}}(t)$ denote the price and quantity at level i at time t, respectively. Levels are ordered by price, representing order priority in matching. The best bid price is the highest bid, while the best ask is the lowest ask. The mid-price, $\text{mp}(t)$, is the average of these best prices. Execution of more bid (ask) orders decreases (increases) the mid-price.

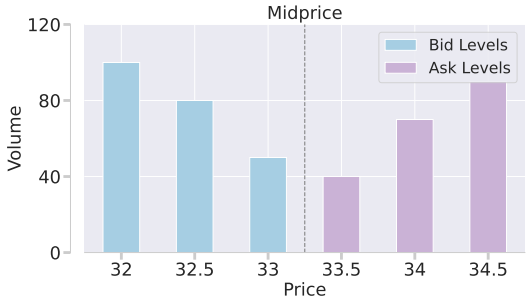


Figure 1: **Visualizing Limit Order Book Data:** three *bid* and three *ask* levels of varying price and volume are shown, as well as the *mid-price*.

Mid-Price Return Forecasting (MPRF). To address mid-price volatility and non-stationarity, we model the problem as forecasting the mid-price *return*. At time t, our target is:

$$\text{target}_h(t) = \text{mp}(t+h)/\text{mp}(t) - 1, \quad \text{where } h \text{ is the forecasting horizon.}$$

Mid-Price Trend Prediction (MPTP). An alternative approach models mid-price prediction as a classification problem, categorizing trends into three classes: **U** (upward), **D** (downward), and **S** (stable). Following (Ntakaris et al., 2018), we generate labels from raw LOB data by comparing the current mid-price to the average future mid-price:

$$\begin{cases} \mathbf{U}, & \text{if} \quad \text{avg_mp}(k,t) > \text{mp}(t)\times(1+\alpha) \\ \mathbf{D}, & \text{if} \quad \text{avg_mp}(k,t) < \text{mp}(t)\times(1-\alpha) \\ \mathbf{S}, & \text{if} \quad \text{Otherwise,} \end{cases} \quad \text{where } \text{avg_mp}(k,t) = (\Sigma_{i=1}^{k}\text{mp}(t+i))/k.$$

Using $\alpha = 0.002\%$ yields approximately equal distribution (33%) for each label.

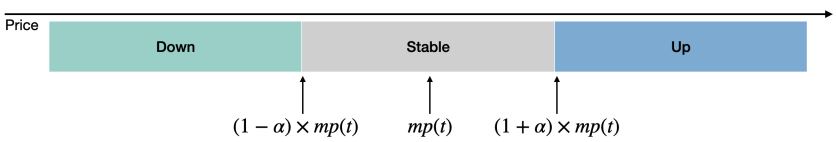


Figure 2: **Trends:** By comparing the average mid-price with an interval defined around the current mid-price, the trend can fall in one of three possible ranges: *Down*, *Stable*, and *Up*

Table 1: **CHF-2023 Features.** CHF-2023 consists of three feature sets: a set of 20 *Basic LOB* features, a set of 26 *Time-insensitive* features, and a set of 20 *Time-sensitive* features.

Feature Set	Definitions	Details
Basic LOB	$u_1 = \{P_i^{\text{bid}}, V_i^{\text{bid}}, P_i^{\text{ask}}, V_i^{\text{ask}}\}_{i=1}^{n}$	5-level LOB Data
Time-insensitive	$u_2 = \{(P_i^{\text{bid}} - P_i^{\text{ask}}), (P_i^{\text{bid}} + P_i^{\text{ask}})/2\}_{i=1}^{n}$	Spread & Mid-Price
	$u_3 = \{P_n^{\text{ask}} - P_1^{\text{ask}}, P_1^{\text{bid}} - P_n^{\text{bid}}, \lvert P_{i+1}^{\text{ask}} - P_i^{\text{ask}}\rvert, \lvert P_{i+1}^{\text{bid}} - P_i^{\text{bid}}\rvert\}_{i=1}^{n-1}$	Price Differences
	$u_4 = \left\{\frac{1}{n}\sum_{i=1}^{n} P_i^{\text{ask}}, \frac{1}{n}\sum_{i=1}^{n} P_i^{\text{bid}}, \frac{1}{n}\sum_{i=1}^{n} V_i^{\text{ask}}, \frac{1}{n}\sum_{i=1}^{n} V_i^{\text{bid}}\right\}$	Price & Volume Means
	$u_5 = \left\{\sum_{i=1}^{n}(P_i^{\text{ask}} - P_i^{\text{bid}}), \sum_{i=1}^{n}(V_i^{\text{ask}} - V_i^{\text{bid}})\right\}$	Accumulated Differences
Time-sensitive	$u_6 = \left\{\mathrm{d}P_i^{\text{ask}}/\mathrm{d}t, \mathrm{d}P_i^{\text{bid}}/\mathrm{d}t, \mathrm{d}V_i^{\text{ask}}/\mathrm{d}t, \mathrm{d}V_i^{\text{bid}}/\mathrm{d}t\right\}_{i=1}^{n}$	Price & Volume Derivation

4 EXPERIMENTS

We conduct all experiments using 3 seeds to minimize the effect of random initialization.

Datasets: **1) FI-2010** (Ntakaris et al., 2018)[1]: This Limit Order Book (LOB) dataset includes 10 trading days of data from five Finnish companies on the NASDAQ Nordic stock market. It was designed to evaluate machine learning models' performance on stock price trend prediction. **2) CHF-2023**: To address FI-2010's limitations (single asset type and short time span), we use this proprietary futures LOB dataset. It covers 5 years of LOB data for SC (Crude Oil), one of China's most liquid futures contracts. The raw data[2] is the most granular LOB dataset available for the Chinese futures market, offering 500ms resolution snapshots with five-level bid and ask information. We use both datasets in MPTP to study LOB models' generalizability to different assets. We only use FI-2010 in MPRF for efficient experimentation.

Evaluation Metrics: For MPTP, we use the F1 scores. For MPRF, we use Mean Squared Error (MSE), Pearson Correlation Coefficient (Corr), and Coefficient of Determination (R^2). MSE quantifies the average squared difference between predicted and actual returns, providing a measure of prediction accuracy. The Pearson Correlation Coefficient assesses the linear relationship between predicted and actual returns, indicating the direction and strength of their association. Lastly, R^2 represents the proportion of variance in the target variable explained by our model. The implementation of MSE and R^2 are from Scikit-learn (Pedregosa et al., 2011). The implementation of Pearson Correlation Coefficient is from SciPy (Virtanen et al., 2020).

Hyperparameter Search. For MPTP, we use the hyperparameters from the original paper of the models. For MPRF, we perform a grid search. Details are in Appendix B.

Table 2: **Input lookback size and number of features for MPTP Models.** (Tsantekidis et al., 2017b;b;a; Tran et al., 2018; Zhang et al., 2019; Passalis et al., 2019; Tsantekidis et al., 2020; Wallbridge, 2020; Passalis et al., 2020; Zhang & Zohren, 2021; Guo & Chen, 2022) The number of features is formatted as [basic]/[basic + time-insensitive]/[basic + time-insensitive + time-sensitive]

	MLP	LSTM	CNN1	CTABL	DEEPLOB	DAIN	CNNLSTM	CNN2	TRANSLOB	TLONBoF	BINCTABL	DEEPLOBATT	DLA
Lookback size	100	100	100	10	100	15	300	300	100	15	10	50	5
Features (FI-2010)	40/86/144	40/86/144	40/86/144	40/86/144	40/86/144	40/86/144	40/86/144	40/86/144	40/86/144	40/86/144	40/86/144	40/86/144	40/86/144
Features (CHF-2023)	20/46/66	20/46/66	20/46/66	20/46/66	20/46/66	20/46/66	20/46/66	20/46/66	20/46/66	20/46/66	20/46/66	20/46/66	20/46/66

Table 3: **Input lookback size and number of features for MPRF Models.** (Nie et al., 2022; Zeng et al., 2023; Liu et al., 2023; Wang et al., 2024) The number of features is formatted as [basic]/[basic + time-insensitive]/[basic + time-insensitive + time-sensitive]

	MLP	LSTM	CNN1	BINCTABL	DAIN	TRANSLOB	PatchTST	DLinear	iTransformer	TimeMixer
Lookback size	100	100	100	10	15	100	100	100	100	100
Features (FI-2010)	41/86/144	41/86/144	41/86/144	41/86/144	41/86/144	41/87/145	41/87/145	41/87/145	41/87/145	41/87/145

4.1 MODELS

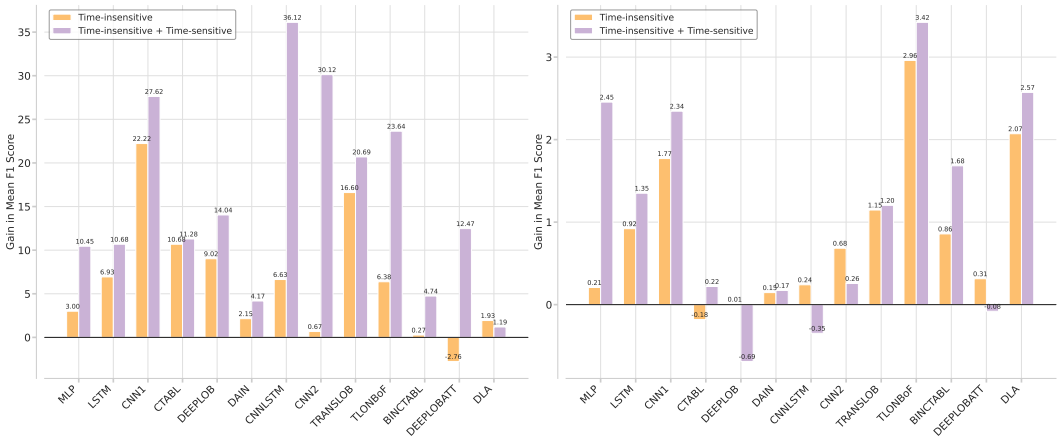
Our benchmark includes models for two tasks: MPTP and MPRF. For MPTP, we select 13 state-of-the-art models. The input and output of each mid-price trend prediction model follow the same protocol

[1]License: Creative Commons Attribution 4.0 International (CC BY 4.0)

[2]http://www.cffex.com.cn/u/cms/www/202201/20211342wucd.pdf

Table 4: **Mid-price Trend Prediction F1 Scores (Mean&Standard Deviation) on Basic LOB data + time-insensitive features + time-sensitive features**. We provide the F1 scores on mid-price trend predictions across horizons {1,2,3,5,10} on for the FI-2010 and CHF-2023 datasets. The model performance ranking is not consistent between two datasets, indicating that models' prediction power for one asset is not automatically transferable to another asset.

	FI-2010						CHF-2023					
Model	K=1	K=2	K=3	K=5	K=10	avg	K=1	K=2	K=3	K=5	K=10	avg
MLP	54.0 (3.7)	52.6 (2.0)	55.6 (0.3)	55.0 (0.4)	52.3 (2.8)	53.9	42.4 (1.3)	47.0 (0.5)	47.8 (0.7)	46.3 (0.3)	41.3 (1.0)	45.0
LSTM	76.3 (0.3)	75.4 (0.1)	76.9 (0.9)	75.6 (1.1)	62.9 (1.3)	73.4	49.7 (0.4)	50.9 (0.4)	51.4 (0.3)	**49.2 (0.3)**	45.8 (0.3)	49.4
CNN1	73.6 (0.3)	72.0 (0.8)	75.3 (0.5)	78.4 (0.8)	78.9 (1.1)	75.6	48.5 (0.9)	49.7 (1.0)	50.6 (0.6)	48.2 (0.5)	45.5 (0.8)	48.5
CTABL	76.6 (0.2)	72.6 (0.4)	77.9 (0.2)	82.1 (0.5)	83.3 (0.5)	78.5	45.2 (2.1)	46.4 (1.0)	46.7 (0.9)	45.6 (0.7)	43.8 (0.5)	45.5
DEEPLOB	**81.2 (0.3)**	82.4 (0.7)	86.1 (0.5)	88.1 (0.2)	88.7 (0.2)	85.3	43.2 (5.3)	47.9 (2.4)	49.5 (1.8)	46.6 (1.3)	41.4 (1.3)	45.7
DAIN	80.8 (0.1)	79.7 (0.1)	85.8 (0.1)	90.0 (0.0)	93.2 (0.0)	85.9	45.7 (0.3)	48.4 (0.5)	48.7 (0.5)	47.1 (0.2)	41.8 (0.8)	46.3
CNNLSTM	74.1 (0.4)	68.1 (0.3)	73.7 (0.7)	77.3 (1.1)	77.8 (0.9)	74.2	45.4 (0.6)	47.0 (1.3)	48.1 (0.9)	46.7 (1.0)	46.0 (0.6)	46.6
CNN2	73.1 (0.2)	67.5 (1.0)	70.8 (4.2)	74.4 (3.4)	65.7 (1.2)	70.3	45.5 (0.6)	46.2 (0.6)	47.9 (0.5)	46.4 (0.5)	46.2 (0.3)	46.4
TRANSLOB	74.5 (1.1)	70.5 (0.3)	75.2 (0.3)	78.4 (0.2)	68.2 (3.2)	73.4	**52.2 (0.4)**	**52.1 (0.3)**	**51.4 (0.4)**	49.0 (0.4)	**46.2 (0.4)**	50.2
TLONBoF	61.3 (6.1)	66.9 (0.3)	72.6 (0.7)	75.2 (4.0)	71.2 (2.7)	69.4	50.4 (0.4)	50.3 (0.7)	50.3 (0.3)	47.1 (1.2)	45.0 (0.3)	48.6
BINCTABL	80.3 (0.1)	**83.4 (0.3)**	**87.9 (0.2)**	**91.3 (0.1)**	**93.2 (0.5)**	87.2	47.2 (0.8)	47.8 (1.1)	48.2 (0.6)	47.0 (0.6)	44.9 (0.4)	47.0
DEEPLOBATT	77.9 (0.0)	78.5 (0.0)	82.5 (0.0)	82.9 (0.0)	82.1 (0.0)	80.8	47.4 (0.8)	49.9 (0.6)	50.6 (0.4)	48.8 (0.2)	44.8 (0.7)	48.3
DLA	71.8 (0.0)	71.1 (0.0)	76.4 (0.0)	85.0 (0.0)	59.3 (0.0)	72.7	48.4 (2.1)	50.5 (0.6)	50.4 (0.5)	47.8 (0.6)	41.1 (0.8)	47.6
Mean	73.5	72.4	76.7	79.5	75.1	47.0	48.8	49.3	47.4	44.1		



(a) Gain on FI-2010 (b) Gain on CHF-2023

Figure 3: **Gains in mean F1 scores.** These plots illustrate the incremental prediction power gained from time-insensitive and time-sensitive features on FI-2010 (a) and CHF-2023 (b) datasets. Yellow bars indicate the improvement in F1 scores when adding time-insensitive features to basic LOB features for most models. Purple bars, compared to yellow, demonstrate the further enhancement in F1 scores when incorporating time-sensitive features alongside basic and time-insensitive features.

in their original paper. For MPRF, we choose 10 models with two goals in mind: benchmarking diverse neural architectures (MLP, CNN, LSTM, and Transformer) and evaluating the importance of domain-specific inductive bias for LOB data. We include some models from the MPTP list that have LOB-specific adaptations, as well as high-performing general-purpose time series forecasters. This mix allows us to compare specialized LOB models against successful general-purpose forecasters. For time series forecasting models including PatchTST, DLinear, iTransformer and TimeMixer, the LOB data as well as the history mid-price return are input as a multivariate time series. The output for all MPRF models is a scaler representing the return of horizon h. Each input is $\mathbb{R}^{T\times 4L}$, output is $\mathbb{R}$. Detailed model and input information is in Appendix A. Table 2 and Table 3 include the input lookback size and feature dimension for each model.

4.2 Mid-price Trend Prediction Results

Table 4 reveals inconsistencies in the top-performing models between the stock FI-2010 and futures CHF-2023 datasets. While BINCTABL, DAIN, and DEEPLOB perform significantly better than other models on the FI-2010 dataset, this advantage is not present in the CHF dataset. This discrepancy suggests that models that exhibit strong performance on the stock LOB dataset are not robust to the futures LOB datasets. To investigate the predictive power of different feature types (basic, time-insensitive, time-sensitive), we evaluate the models on two feature subsets: one with only basic features, and another with basic and time-insensitive features. Figure 3 illustrates the gains in

Table 5: **Mid-price Return Forecasting Results (Mean) on Basic LOB data**. 10 LOB models and time series forecasting models are benchmarked to compare their Mean Square Error (MSE), Pearson correlation (Corr), and Coefficient of Determination (R^2) on mid-price return forecasting across 5 horizons {1,2,3,5,10} on the FI-2010 dataset. LOB models perform much better than general-purpose time series models, indicating that it is essential to include LOB-relevant inductive bias into the model design to achieve good forecasting power on LOB datasets. For each horizon, the best model is bolded, and the next best model is underlined.

		FI-2010				
Model	Metric	K=1	K=2	K=3	K=5	K=10
MLP	MSE	0.659 (0.005)	1.096 (0.021)	1.431 (0.020)	1.972 (0.008)	2.963 (0.162)
	Corr	0.084 (0.002)	0.101 (0.001)	0.102 (0.011)	0.106 (0.006)	0.125 (0.019)
	R^2	-0.004 (0.007)	-0.016 (0.020)	-0.016 (0.014)	-0.008 (0.004)	-0.060 (0.058)
LSTM	MSE	**0.638 (0.005)**	**1.035 (0.009)**	**1.328 (0.007)**	**1.824 (0.013)**	**2.571 (0.024)**
	Corr	**0.173 (0.020)**	**0.211 (0.024)**	**0.244 (0.007)**	**0.274 (0.014)**	0.298 (0.009)
	R^2	**0.027 (0.007)**	**0.041 (0.008)**	**0.057 (0.005)**	**0.067 (0.006)**	**0.081 (0.009)**
CNN1	MSE	0.665 (0.008)	1.058 (0.007)	1.379 (0.015)	1.879 (0.023)	2.681 (0.017)
	Corr	0.129 (0.006)	0.185 (0.013)	0.210 (0.004)	0.248 (0.016)	**0.298 (0.007)**
	R^2	-0.013 (0.012)	0.020 (0.007)	0.021 (0.011)	0.039 (0.012)	0.041 (0.006)
BINCTABL	MSE	0.650 (0.000)	1.047 (0.001)	1.347 (0.008)	1.838 (0.016)	2.612 (0.012)
	Corr	0.106 (0.004)	0.176 (0.003)	0.215 (0.015)	0.249 (0.016)	0.278 (0.003)
	R^2	0.010 (0.000)	0.029 (0.001)	0.044 (0.006)	0.061 (0.008)	0.069 (0.004)
DAIN	MSE	0.693 (0.011)	1.114 (0.014)	1.436 (0.004)	1.977 (0.004)	2.824 (0.014)
	Corr	0.038 (0.004)	0.068 (0.007)	0.085 (0.002)	0.107 (0.003)	0.127 (0.001)
	R^2	-0.057 (0.016)	-0.032 (0.013)	-0.019 (0.003)	-0.011 (0.002)	-0.007 (0.005)
TRANSLOB	MSE	0.659 (0.005)	1.088 (0.002)	1.395 (0.003)	1.904 (0.015)	2.704 (0.029)
	Corr	0.079 (0.010)	0.090 (0.019)	0.150 (0.018)	0.210 (0.004)	0.267 (0.009)
	R^2	-0.005 (0.008)	-0.008 (0.002)	0.009 (0.002)	0.027 (0.008)	0.033 (0.010)
PatchTST	MSE	0.654 (0.000)	1.077 (0.000)	1.406 (0.001)	1.949 (0.001)	2.795 (0.002)
	Corr	0.081 (0.002)	0.082 (0.001)	0.079 (0.004)	0.092 (0.003)	0.085 (0.003)
	R^2	0.003 (0.001)	0.002 (0.000)	0.002 (0.001)	0.004 (0.000)	0.001 (0.001)
DLinear	MSE	0.652 (0.000)	1.073 (0.000)	1.402 (0.000)	1.945 (0.001)	2.782 (0.000)
	Corr	0.080 (0.002)	0.081 (0.001)	0.074 (0.001)	0.083 (0.002)	0.084 (0.002)
	R^2	0.006 (0.000)	0.006 (0.000)	0.005 (0.000)	0.006 (0.001)	0.005 (0.000)
iTransformer	MSE	0.683 (0.008)	1.183 (0.031)	1.582 (0.016)	2.279 (0.095)	3.401 (0.076)
	Corr	0.045 (0.004)	0.045 (0.007)	0.033 (0.005)	0.063 (0.004)	0.056 (0.004)
	R^2	-0.041 (0.012)	-0.096 (0.028)	-0.123 (0.012)	-0.165 (0.048)	-0.216 (0.027)
TimeMixer	MSE	0.657 (0.001)	1.075 (0.002)	1.394 (0.002)	1.888 (0.018)	2.643 (0.017)
	Corr	0.083 (0.005)	0.110 (0.001)	0.135 (0.005)	0.201 (0.025)	0.271 (0.014)
	R^2	-0.001 (0.002)	0.004 (0.001)	0.011 (0.001)	0.035 (0.009)	0.055 (0.006)

prediction performance based on average F1 scores across 5 horizons for each feature set (full results are available in Appendix E). We define feature set gains as the difference between its average F1 scores and those of the basic features. Consequently, basic features have zero gain, serving as the baseline. Positive gains for time-insensitive and time-sensitive features indicate additional predictive power beyond raw LOB readings. Figure 3 demonstrates that both time-insensitive and time-sensitive features contribute positive gains on top of basic LOB features for both FI-2010 and CHF-2023 datasets. This finding affirms the universal effectiveness of these feature sets across stock and futures datasets, underscoring their value in enhancing model performance regardless of the asset type.

4.3 Mid-price Return Forecasting Results

Table 5 shows that LOB models with LOB-specific inductive bias significantly outperform general-purpose time series prediction models. This highlights the importance of incorporating LOB-related inductive bias in model design. Additionally, the results indicate that general-purpose models lack the generalization needed for strong forecasting performance on LOB datasets. This performance gap underscores the specialized nature of LOB data and the need for tailored models in financial forecasting.

Prediction Performance Gap Between LOB Models and Time Series Models. In the MPRF task, a significant performance gap exists between LOB-specific models and general time series models.

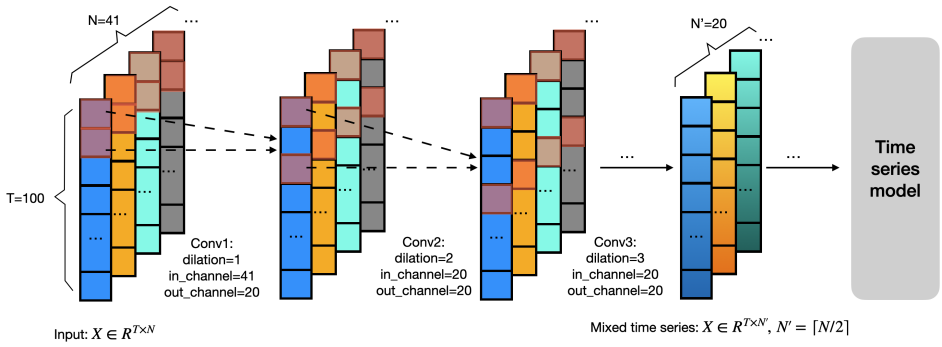


Figure 4: **Cross-Variate Mixing Layers (CVML) Architecture.** CVML consists of 5 Conv1D layers, with the first 3 illustrated here. Each Conv1D layer employs a kernel size of 2 and matches its input channels to the number of variates in the input time series. The architecture features increasing dilation across successive layers to expand the receptive field. Input $X \in \mathbb{R}^{T \times N}$ is transformed into a mixed time series $X' \in \mathbb{R}^{T \times N'}$, $N' = \lceil N/2 \rceil$, before feeding into the subsequent time series model.

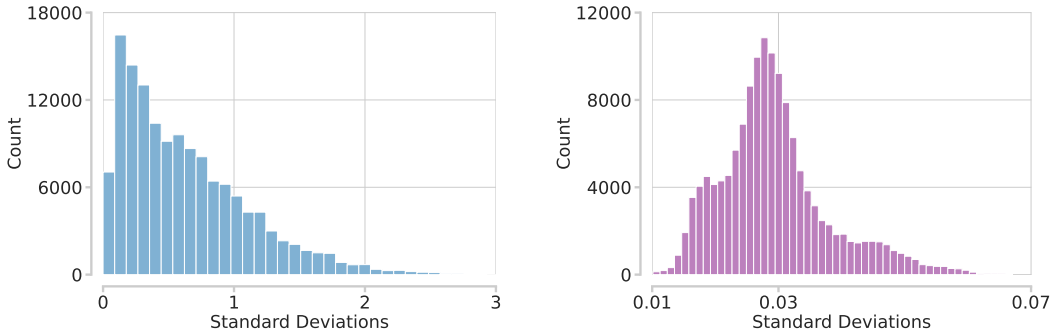
Table 6: **Time Series Model Performance with and without CVML** on the FI-2010 Dataset using basic LOB features. The % column indicates the percentage improvement from adding CVML.

	MSE (↓)						Corr (↑)						R^2 (↑)					
Model	K=1	K=2	K=3	K=5	K=10	%	K=1	K=2	K=3	K=5	K=10	%	K=1	K=2	K=3	K=5	K=10	%
PatchTST-CVML	**0.653**	**1.071**	**1.370**	**1.893**	**2.646**	3.1	**0.070**	**0.113**	**0.165**	**0.191**	**0.241**	86.2	**0.005**	**0.007**	**0.028**	**0.033**	**0.054**	958.3
PatchTST	0.654	1.077	1.406	1.949	2.795		0.081	0.082	0.079	0.092	0.085		0.003	0.002	0.002	0.004	0.001	
DLinear-CVML	**0.650**	**1.042**	**1.352**	**1.796**	**2.548**	5.9	**0.104**	**0.192**	**0.205**	**0.291**	**0.313**	174.9	**0.010**	**0.035**	**0.040**	**0.082**	**0.089**	814.3
DLinear	0.652	1.073	1.402	1.945	2.782		0.080	0.081	0.074	0.083	0.084		0.006	0.006	0.005	0.006	0.005	
iTransformer-CVML	**0.654**	1.084	**1.402**	**2.002**	**2.649**	14.6	**0.054**	**0.070**	**0.088**	**0.121**	**0.249**	140.5	**0.002**	**-0.005**	**0.005**	**-0.024**	**0.053**	104.8
iTransformer	0.683	**1.183**	1.582	2.279	3.401		0.045	0.045	0.033	0.063	0.056		-0.041	-0.096	-0.123	-0.165	-0.216	
TimeMixer-CVML	**0.642**	**1.033**	**1.329**	**1.807**	**2.494**	4.6	**0.160**	**0.221**	**0.257**	**0.298**	**0.353**	61.1	**0.022**	**0.043**	**0.056**	**0.076**	**0.109**	194.2
TimeMixer	0.657	1.075	1.394	1.888	2.643		0.083	0.110	0.135	0.201	0.271		-0.001	0.004	0.011	0.035	0.055	

Notably, complex Transformer-based models including PatchTST and iTransformer underperform compared to simpler LOB-specific architectures based-on MLPs or LSTMs. This discrepancy suggests that without LOB-aware architectural design, conventional time series models struggle to generate accurate predictions on LOB data due to its low signal-to-noise ratio. This observation indicates that the sophisticated temporal modeling capabilities of state-of-the-art time series models may be impeded by the noisy temporal dynamics and intricate cross-variate correlations inherent in LOB data. Based on this hypothesis, we propose a novel module called Cross-Variate Mixing Layers (CVML) to enhance the signal-to-noise ratio of LOB time series. CVML serves as an add-on layer preceding the time series modeling layers in standard time series prediction models. It accepts raw LOB data, mixes the features (or different variates from a time series perspective), and outputs an intermediate multivariate time series as input for subsequent modeling layers. CVML integrates seamlessly with existing time series models without requiring further modifications and can be trained end-to-end. CVML has five Conv1D layers, each with a kernel size of 2 and $\lceil N/2 \rceil$ output channels, where N is the number of input features. This design leverages convolution kernels to extract cross-variate features and capture correlations. Additionally, we implement increasing dilation in the kernel for each successive layer to enhance temporal signal extraction.

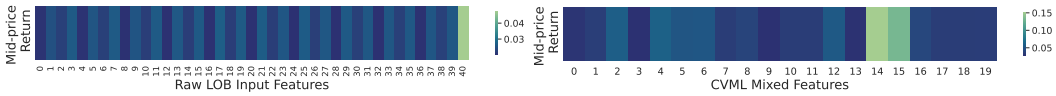
To demonstrate CVML's efficacy, we prepend it to four time series models without altering their core architectures. Results show significant improvements in forecast performance across all metrics (MSE, Corr, and R^2), as shown in Table 6. Averaging across four models and five horizons, CVML achieves a 7.4% improvement in MSE, 101.6% in Pearson Correlation (Corr), and 244.9% in R^2. This highlights CVML's potential as a powerful add-on for enhancing general time series models on complex LOB data, bridging the gap between specialized LOB models and general-purpose forecasting approaches.

To illustrate CVML's effects, we analyze the standard deviation (std) of the time steps in each mid-price return input, then plot the histogram of these std values across all inputs. Figure 5 demonstrates that CVML significantly reduces std values of the LOB time series, indicating its effective smoothing effect. Furthermore, the distribution in Figure 5b more closely resembles a normal distribution. These transformations collectively enhance the time series models' ability to capture temporal signals by presenting them with more structured and less noisy data. This visualization provides evidence of

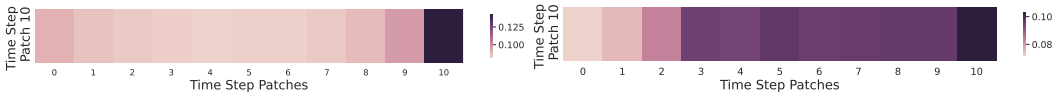


(a) Raw Inputs Standard Deviations (b) Outputs of CVML's Standard Deviations

Figure 5: **Standard Deviation Distribution (std) of Inputs Before and After CVML Processing.** This histogram compares the std of raw inputs (a) and CVML outputs (b) for the FI-2010 test set, using TimeMixer with CVML add-on. The CVML outputs exhibit lower std values and a distribution closer to normal, suggesting noise reduction.



(a) iTransformer Cross-Variate Attentions (w/o CVML) (b) iTransformer Cross-Variate Attentions (w/ CVML)



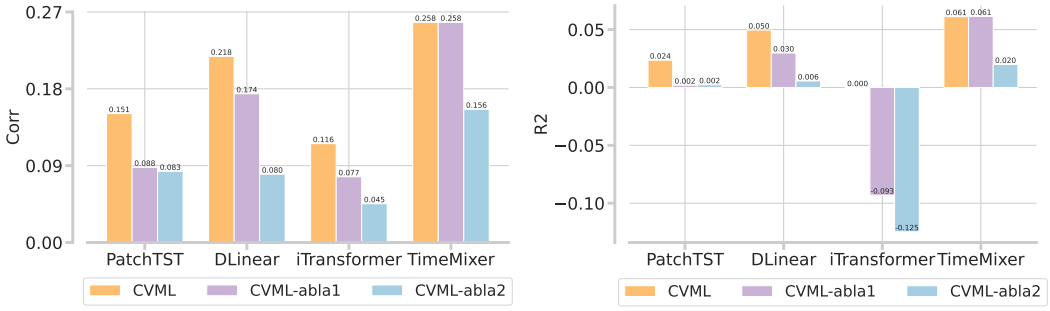
(c) PatchTST Temporal Attentions (w/o CVML) (d) PatchTST Temporal Attentions (w/ CVML)

Figure 6: **Impact of CVML on Cross-Variate and Temporal Attention Patterns.** This figure compares attention scores from iTransformer and PatchTST. (a, b) Cross-variate attention from mid-price return (id: 40) to all variates. (c, d) Temporal attention from the latest time step patch (id: 10) to all historical patches. iTransformer w/ CVML (b) captures more nuanced cross-variate correlations compared to (a). PatchTST w/ CVML (d) reveals clearer temporal dependencies than (c).

CVML's role in improving the signal-to-noise ratio of LOB data, thereby facilitating more accurate predictions by subsequent time series models.

We demonstrate that CVML enhances the ability of time series models to capture cross-variate and temporal correlations, using iTransformer and PatchTST as examples. We chose them for their interpretable attention mechanisms and their representative modeling on different correlation types: iTransformer focuses on cross-variate correlations through self-attention on the variate dimension, while PatchTST emphasizes temporal correlations via self-attention on time dimension patches. We train both models on FI-2010 and analyze the average attention scores from their final attention layers across the test set. For iTransformer, we visualize attention scores from the mid-price return (target variate, id: 40) to all other variates and itself. For PatchTST, we examine average attention scores from the last time step patch (id: 10) to all other patches and itself.

Cross-Variate Correlations: Figure 6a reveals that without CVML, iTransformer's mid-price return attention is predominantly self-focused, with a uniform pattern corresponding to the LOB input feature layout ($\{V_i^{\text{bid}}, P_i^{\text{bid}}, V_i^{\text{ask}}, P_i^{\text{ask}}\}_{i=1}^{10}$). This uniform attention suggests only surface-level capture of cross-variate correlations, failing to differentiate between LOB levels. Notably, it doesn't reflect the expected stronger correlation of mid-price to the best bid and ask prices. In contrast, Figure 6b shows that with CVML, iTransformer exhibits varied attention across variates, indicating a more nuanced modeling of LOB-level correlations. **Temporal Correlations:** Figure 6c demonstrates PatchTST's attention w/o CVML, showing strong self-attention for the latest time patch but no discernible temporal pattern with other patches. Conversely, Figure 6d demonstrates that with CVML, PatchTST captures a clear decaying temporal pattern and stronger attention to immediate past neighbors, indicating improved modeling of temporal dependencies.



(a) Correlation (b) R^2

Figure 7: **Average Correlation and R^2 Scores Across Prediction Horizons.** This figure compares the performance of CVML and its two ablated versions (CVML-abla1 and CVML-abla2) across four time series models. (a) shows average Correlation scores and (b) shows average R^2 scores, both calculated across five prediction horizons. The complete CVML consistently outperforms its ablated counterparts. Notably, CVML-abla2, which lacks cross-variate information aggregation, performs the worst, highlighting the critical importance of cross-variate mixing in CVML's effectiveness.

4.4 Ablation Study

CVML is designed to capture two types of correlations: cross-variate and temporal. To demonstrate the efficacy of this design, we conduct an ablation study using two modified versions of CVML. The first variant (CVML-abla1) reduces the kernel size to 1, focusing solely on cross-variate correlations while eliminating temporal correlations. The second variant (CVML-abla2) maintains the original kernel size of 2 but employs depthwise convolution (Chollet, 2017; Pandey, 2024), which processes each variate independently without cross-variate information aggregation. We replicate the experiments from Table 6 using these ablated versions and compare their average forecast performance across the five prediction horizons. Figure 7 shows that both ablated versions underperform the original CVML. These results underscore the importance of CVML's dual-correlation design, highlighting its ability to effectively capture both cross-variate and temporal dependencies in LOB data. The full results of the two ablated CVMLs including MSE, Corr, and R^2 are in Table 18.

To verify that CVML's performance gain does not come model size increase, we examine the model size of each model before and after adding CVML. Table 7 shows the number of learnable parameters. The percentage indicates the size of the vanilla model compared to the counterpart with CVML. Except TimeMixer, all other models are of more than 90% size of the counterpart with CVML. Thus, we increase TimeMixer's number of layers to increase its learnable parameters to 14156, about 109% of the CVML version and test its performance.

To ensure that CVML's performance gains are not solely attributable to increased model complexity, we compare the model sizes before and after incorporating CVML. Table 7 presents the number of learnable parameters for each model, with percentages indicating the size of the vanilla model relative to its CVML-enhanced counterpart. All models except TimeMixer is over 90% of the size of the version with CVML integration. We augment TimeMixer's architecture by increasing its number of layers, resulting in 14,156 learnable parameters, approximately 109% of its CVML version's size. We then evaluate this enlarged TimeMixer and it still significantly underperforms TimeMixer-CVML, proving that CVML's gains are not solely attributable to increased model complexity.

Table 7: Model Size Comparison

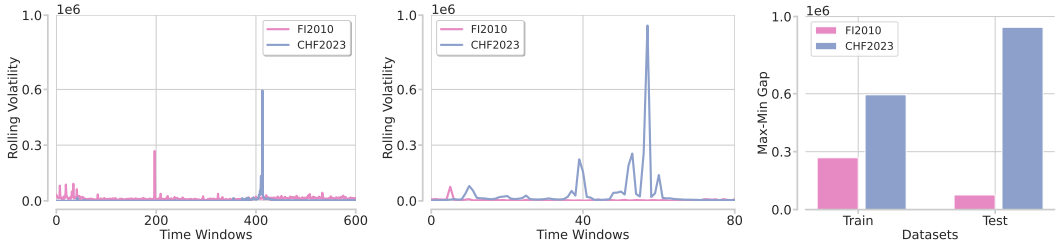
	PatchTST	DLinear	iTransformer	TimeMixer
Vanilla	33766 (92.65%)	8282 (148%)	6358017 (99.96%)	9471 (72.71%)
w/ CVML	36444	5614	6360803	13025

Table 8: Enlarged TimeMixer (109% of TimeMixer+CVML size)

	K=1	K=2	K=3	K=5	K=10
Corr	0.072	0.110	0.146	0.206	0.268
R^2	-0.010	-0.001	0.011	0.030	0.053

5 Discussions

Prediction Performance Gap Between FI-2010 and CHF-2023 in MPTP. Overall, models demonstrate superior performance on the FI-2010 dataset but not on the CHF-2023 dataset. To understand this discrepancy, we analyze the rolling volatility of both datasets using the formula: $\sigma_S = \text{std}(S) \times \sqrt{\text{annualized}}$ where σ_S is the standard deviation of mid-price returns in a history window of size S, and $\sqrt{\text{annualized}}$ is a normalization factor. The annualized term equals the number of time steps in a 252-day trading year at the dataset's resolution. We calculate rolling



(a) FI-2010/CHF-2023 Training Data (b) FI-2010/CHF-2023 Testing Data (c) FI-2010/CHF-2023 Max-Min Gap in Rolling Volatility

Figure 8: **Rolling Volatility Comparison: FI-2010 vs. CHF-2023.** The CHF-2023 dataset exhibits sharp volatility spikes compared to the more stable FI-2010 dataset, making it more challenging for model fitting and prediction. (a) and (b) display the maximum volatility point for every 500 time steps to enhance clarity and interpretability without compromising the overall trend. (c) illustrates the max-min gap in rolling volatility, further highlighting the disparity between the two datasets.

volatility using a sliding window with $S = 20$ and a step size of 1. Figure 8 reveals that FI-2010's rolling volatility is relatively consistent across the entire time series. In contrast, CHF-2023 exhibits several extreme volatility spikes, presenting significant challenges for model fitting and prediction. Two notable spike periods are identified: September 2021 to January 2022: Coinciding with crude oil price increases due to supply disruptions and production constraints (United States International Trade Commission, 2021). May 2023 to June 2023: Corresponding to crude oil price fluctuations following the EU's import ban on Russian crude oil and products (French, 2024). This analysis highlights the unique characteristics and challenges inherent in the futures dataset compared to FI-2010. The inclusion of futures data in our benchmark provides a more comprehensive evaluation, capturing market dynamics not present in stock-only datasets. This diversity in data sources enhances the robustness and applicability of our benchmark study to real-world financial forecasting scenarios.

6 Conclusion

We present a comprehensive benchmark of neural network architectures' predictive performance on limit order book (LOB) datasets. Our analysis encompasses two critical LOB prediction tasks: mid-price trend prediction (MPTP) and mid-price return forecasting (MPRF). We evaluate models using both an open-source stock LOB dataset and a proprietary futures LOB dataset, comparing specialized LOB models against state-of-the-art general-purpose time series forecasting models. Our research yields four conclusions:

- **Feature Importance.** All three sets of LOB features, basic, time-insensitive, and time-sensitive, demonstrate significant predictive power for the MPTP task, underscoring the importance of comprehensive feature selection in LOB modeling.
- **Asset-Specific Challenges.** The futures LOB dataset exhibits unique characteristics that pose novel challenges for LOB models initially designed on stock data. It highlights the need for asset-specific considerations in model development.
- **Model Specialization.** LOB-specific models significantly outperform general-purpose state-of-the-art time series models on LOB data for the MPRF task. It shows a performance degradation of general-purpose state-of-the-art time series models on the low signal-to-noise ratio LOB time series data.
- **CVML Enhancement.** Our proposed convolution-based cross-variate mixing layers (CVML) substantially improve the predictive performance of general-purpose time series models (by 101.6% in Correlation and 244.9% in R^2) without requiring modifications to the core time series model architectures. These findings underscore the complexity of LOB data analysis and the potential for architectural innovations to bridge the performance gap between specialized and general-purpose models, enabling more effective modeling of the complex dynamics present in LOB data.

REFERENCES

Marco Avellaneda and Sasha Stoikov. High-frequency trading in a limit order book. *Quantitative Finance*, 8(3):217–224, 2008.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023.

Soon Huat Chan, Kenneth A Kim, and S Ghon Rhee. Price limit performance: evidence from transactions data and the limit order book. *Journal of Empirical Finance*, 12(2):269–290, 2005.

SA Chen, CL Li, N Yoder, SO Arik, and T Pfister. Tsmixer: An all-mlp architecture for time series forecasting. arxiv 2023. *arXiv preprint arXiv:2303.06053*, 2023.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Zoltan Eisler, Jean-Philippe Bouchaud, and Julien Kockelkoren. The price impact of order book events: market orders, limit orders and cancellations. *Quantitative Finance*, 12(9):1395–1419, 2012.

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

Matthew French. Brent crude oil prices averaged $19 per barrel less in 2023 than 2022. https://www.eia.gov/todayinenergy/detail.php?id=61142, 2024. Accessed: 08.09.2024.

Yanhong Guo and Xinxin Chen. Forecasting the mid-price movements with high-frequency lob: A dual-stage temporal attention-based deep learning architecture. *Arabian Journal for Science and Engineering*, 48(8):9597–9618, 2022.

Lawrence E Harris and Venkatesh Panchapagesan. The information content of the limit order book: evidence from nyse specialist trading decisions. *Journal of Financial Markets*, 8(1):25–67, 2005.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Zexin Hu, Yiqi Zhao, and Matloob Khushi. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, 4(1):9, 2021.

Weiwei Jiang. Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications*, 184:115537, 2021.

Mahinda Mailagaha Kumbure, Christoph Lohrmann, Pasi Luukka, and Jari Porras. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197:116659, 2022.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Jeremy Large. Measuring the resiliency of an electronic limit order book. *Journal of Financial Markets*, 10(1):1–25, 2007.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation, 2015.

Latrisha N Mintarya, Jeta NM Halim, Callista Angie, Said Achmad, and Aditya Kurniawan. Machine learning approaches in stock market prediction: A systematic literature review. *Procedia Computer Science*, 216:96–102, 2023.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Adamantios Ntakaris, Martin Magris, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, 37(8):852–866, 2018.

Isaac Kofi Nti, Adebayo Felix Adekoya, and Benjamin Asubam Weyori. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, 53 (4):3007–3057, 2020.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

Keiron O'shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. *Applied soft computing*, 93:106384, 2020.

Atul Pandey. Depth-wise convolution and depth-wise separable convolution. https://medium.com/@zurister/depth-wise-convolution-and-depth-wise-separable-convolution-37346565d4ec, 2024. Accessed: 08.09.2024.

Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Deep adaptive input normalization for time series forecasting. *IEEE transactions on neural networks and learning systems*, 31(9):3760–3765, 2019.

Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Temporal logistic neural bag-of-features for financial time series forecasting leveraging limit order book data. *Pattern Recognition Letters*, 136:183–189, 2020.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Matteo Prata, Giuseppe Masi, Leonardo Berti, Viviana Arrigoni, Andrea Coletta, Irene Cannistraci, Svitlana Vyetrenko, Paola Velardi, and Novella Bartolini. Lob-based deep learning models for stock price trend prediction: a benchmark study. *Artificial Intelligence Review*, 57(5):1–45, 2024.

Ioanid Roşu. A dynamic model of the limit order book. *The Review of Financial Studies*, 22(11): 4601–4641, 2009.

Francesco Rundo, Francesca Trenta, Agatino Luigi Di Stallo, and Sebastiano Battiato. Machine learning for quantitative finance applications: A survey. *Applied Sciences*, 9(24):5574, 2019.

Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181, 2020.

Dev Shah, Haruna Isah, and Farhana Zulkernine. Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies*, 7(2):26, 2019.

Dat Thanh Tran, Alexandros Iosifidis, Juho Kanniainen, and Moncef 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.

Dat Thanh Tran, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Data normalization for bilinear structures in high-frequency financial time-series. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7287–7292. IEEE, 2021.

Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th conference on business informatics (CBI)*, volume 1, pp. 7–12. IEEE, 2017a.

Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Using deep learning to detect price change indications in financial markets. In *2017 25th European signal processing conference (EUSIPCO)*, pp. 2511–2515. IEEE, 2017b.

Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Using deep learning for price prediction by exploiting stationary limit order book features. *Applied Soft Computing*, 93:106401, 2020.

United States International Trade Commission. The 2021 commodity price surge: Causes and impacts on trade flows. https://www.usitc.gov/research_and_analysis/tradeshifts/2021/special_topic, 2021. Accessed: 08.09.2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

James Wallbridge. Transformers for limit order books. *arXiv preprint arXiv:2003.00130*, 2020.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

Bruce W Weber. Next-generation trading in futures markets: A comparison of open outcry and order matching systems. *Journal of Management Information Systems*, 16(2):29–45, 1999.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*, 2022.

Zihao Zhang and Stefan Zohren. Multi-horizon forecasting for limit order books: Novel deep learning approaches and hardware acceleration using intelligent processing units. *arXiv preprint arXiv:2105.10430*, 2021.

Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, 2019.

Appendix

A Detailed Information of Models in Benchmark

A.1 Models for Mid-price Trend Prediction

- **MLP, LSTM, CNN1, CNN2, CNNLSTM.** Tsantekidis et al. (2017b) used Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) to predict future mid-price movements. The same authors (Tsantekidis et al., 2017a) also proposed a Convolutional Neural Network (CNN) model (CNN1), which employs a standard CNN architecture with convolutional layers followed by fully connected layers. In 2020, the same research team (Tsantekidis et al., 2020) proposed another CNN model (CNN2) and a CNNLSTM model based on the described LSTM and CNN2. The key improvement of CNN2 is the use of causal convolutions with "full" padding, ensuring all convolutional layers produce outputs with the same number of time steps. This matches labels to their correct time steps and prevents future data from influencing past predictions. The CNNLSTM model merges CNN2 and LSTM, using CNN2 for feature extraction and passing the features to the LSTM module for classification. In our experiments, we adjusted the CNN kernel size to match the different number of features in each dataset.
- **DAIN.** Passalis et al. (2019) introduce Deep Adaptive Input Normalization (DAIN) for time series forecasting. The main innovation of the method is learning to adaptively normalize input data during model training via two feed-forward layers, an adaptive shifting layer and scaling layer, instead of using fixed schemes like z-score normalization. Besides, they use a gating layer to suppress irrelevant features. DAIN explores three possible neural network architectures: MLP, CNN, and RNN. We choose the MLP architecture as it demonstrates the highest empirical performance.
- **CTABL, BINCTABL.** Tran et al. (2018) propose the Temporal-Attention-Augmented Bilinear Layer (TABL) model. The Bilinear Layer (BL) performs two linear transformations on the input data along the feature and temporal dimensions to capture how stock prices interact at a given point in time and how prices at an index progress over time. However, within a BL, it is unclear how different time instance representations interact. TABL addresses this by incorporating a temporal attention mechanism into the BL model to learn the importance of each time instance relative to others. The input time series is first passed through the feature dimension transformation, then through the temporal attention mechanism, and finally through the temporal dimension transformation. In our experiments, we use the C(TABL) variant of TABL for its superior empirical performance over A(TABL) and B(TABL). Tran et al. (2021) improve upon TABL with BINCTABL, which incorporates a Bilinear Normalization (BiN) strategy that normalizes the data along the temporal and feature dimensions with learnable parameters to scale the normalization. The authors compare BiN as a simpler and more intuitive approach compared to DAIN when using TABL networks.
- **DEEPLOB.** Zhang et al. (2019) propose Deep Convolutional Neural Networks for Limit Order Books (DeepLOB). The authors employ a mid-price-based smooth data labeling method to reduce noise and eliminate minor oscillations. DEEPLOB combines convolutional layers and an Inception Module to extract features from the noisy financial data, then uses a Long Short-Term Memory (LSTM) layer to capture longer time dependencies among the extracted features. In our experiments, we adjust the kernel size of the last convolution layer for different datasets.
- **DEEPLOBATT.** Zhang & Zohren (2021) introduce DeepLOB-Attention, an encoder-decoder model built upon their previous work, DEEPLOB. DEEPLOBATT utilizes DEEPLOB as the encoder and an Attention model (Luong et al., 2015) as the decoder. The authors investigated both Attention and Seq2Seq as the decoder in their paper. Since DEEPLOBATT outperforms DEEPLOB-Seq2Seq in nearly all experiments, we choose the DEEPLOBATT model in our experiments and adjust the kernel size of the convolutional block for different datasets according to the feature dimensions.
- **TLONBoF.** Passalis et al. (2020) propose Temporal Logistic Neural Bag-of-Features (TLo-NBoF) (2020), a refined Bag-of-Features (BoF) formulation to better capture the dynamics of time series data compared to existing BoF methods. They achieved this by introducing a novel adaptive scaling mechanism and replacing the Gaussian density estimation of regular BoF with a logistic kernel. The input time series is first passed through a 1-D convolution layer to extract relationships

between time instances. Then, the TLONBoF formulation uses a learnable logistic kernel and codebook to aggregate the feature vectors into short-term, mid-term, and long-term histograms that capture the overall behaviors of the input time series. This adaptive method removes the need for sophisticated initialization methods and facilitates smoother hyperparameter tuning.

- **DLA.** Guo & Chen (2022) propose a Dual-Stage Temporal Attention-Based Deep Learning Architecture (DLA). It uses a dual-stage temporal attention mechanism to repeatedly highlight the most important time instances in input time series data. In the first stage, temporal attention is performed on the input data to learn the importance of each time instance relative to the others. The output of the attention is passed through a stacked Gated Recurrent Unit (GRU) network to further enhance the representational state of the input. In the second stage, temporal attention weights are adaptively assigned to the hidden states of the GRU network. The model's performance is benchmarked against other models in the literature, such as CTABL (Tran et al., 2018), DEEPLOB (Zhang et al., 2019), and TLONBoF (Passalis et al., 2020).
- **TRANSLOB.** Wallbridge (2020) proposes TransLOB, a new model architecture for LOB data that uses Transformer blocks (Vaswani et al., 2023). The TRANSLOB architecture consists of a causal convolution module and a causal transformer block to stay consistent with the nature of time-series data. The convolutional module comprises 5 1-D convolution layers with increasing dilation to capture relationships between short-term and long-term time instances. The transformer block includes 2 masked self-attention encoders to determine important time instance representations. The performance of TRANSLOB is benchmarked against other state-of-the-art models such as CTABL, DEEPLOB, and CNN-LSTM.

A.2 Models for Mid-price Forecasting

Besides the LOB Models including MLP, CNN1, LSTM, BINCTABL, DAIN, and TRANSLOB mentioned above, we also include 4 state-of-the-art time series forecasting models.

- **PatchTST.** Nie et al. proposes PatchTST (Nie et al., 2022), a multivariate time series forecasting model based on vanilla Transformer architecture. It has two major novelties. First, it uses the same model to predict each variate of a multivariate time series model independently, making it technically a univariate time series model. Second, to better capture time series' local pattern, it models the time series in patch-wise (each patch includes multiple consecutive time steps) rather than timestep-wise.
- **iTransformer.** To better capture variate-centric information rather than only temporal dimension, Liu et al. (2023) proposes iTransformer. It uses the vanilla transformer components in a novel way where it first embeds each series of a multivariate time series into a token and then applies attention across all the tokens representing different series.
- **DLinear.** Besides transformer-based models, linear models are also showing top performance in time series forecasting. Zeng et al. (2023) questions Transformer architecture's capability to properly capture temporal correlations in time series data. They propose a simple baseline architecture, DLinear. It is a one-layer linear model with a decomposition component. It first decomposes the raw time series input into a trend component and a seasonal component. Then, two separate one-layer linear layers are applied on these two components. Lastly, two components sum up in to the final prediction. DLinear outperformans most of the Transformer-based time series prediction model and achieve top-tier time series forecasting performance.
- **TimeMixer.** Along the line of better modeling temporal correlations from the time series data, Wang et al. (2024) proposes TimeMixer. It is a linear-layer-based model. Its major novelty is to first preprocess the raw input time series into multiple time series of different resolutions by downsampling. Then, they mix pairs of time series of different resolutions by adding one of the time series to a transformation of the other. The transformation consists of two linear layers and an intermediate GELU activation function. Lastly, they use a linear layer to regress from the mixed time series to produce the final prediction.

B Hyperparameters

Table 9, Table 10, and Table 11 include the hyperparameters we use in MPTP and MPRF. For Mid-Price Trend Prediction (MPTP), we adhere to the hyperparameters specified in the original papers for the models. We apply a patience of 8 for MPTP on the FI dataset and a patience of 3 for the CHF dataset. For Mid-Price Return Forecasting (MPRF), we perform a grid search for the

batch size, including {32, 64, 128} and the learning rate, including {0.01, 0.001, 0.0001, 0.00001} on horizon 5. We employ a patience of 15 for the FI dataset and a patience of 2 for the CHF dataset.

Table 9: Hyperparameters for Mid-Price Trend Prediction on the FI-2010 Dataset. Dashes mean the corresponding module does not exist in the model architecture.

	FI						
Model	**Learning Rate**	**Optimizer**	**Batch Size**	**Epochs**	**Dropout**	**MLP Hidden**	**RNN Hidden**
LSTM	0.001	Adam	32	100	0	64	40
MLP	0.001	Adam	64	100	0.1	128	-
CNN1	0.0001	Adam	64	100	-	32	-
CTABL	0.01	Adam	256	200	0.1	-	-
DAIN	0.0001	RMSprop	32	100	0.5	512	-
DEEPLOB	0.01	Adam	32	100	-	-	64
CNNLSTM	0.001	RMSprop	32	100	0.1	32	32
CNN2	0.001	RMSprop	32	100	-	32	-
TRANSLOB	0.0001	Adam	32	150	0.1	64	-
TLONBoF	0.0001	Adam	128	100	0.5	512	-
BINCTABL	0.001	Adam	128	200	0.1	-	-
DEEPLOBATT	0.001	Adam	32	100	-	-	64
DLA	0.01	Adam	256	100	0.5	-	100

Table 10: Hyperparameters for Mid-Price Trend Prediction on CHF-2023 Dataset

	CHF						
Model	**Learning Rate**	**Optimizer**	**Batch Size**	**Epochs**	**Dropout**	**MLP Hidden**	**RNN Hidden**
LSTM	0.001	Adam	1024	100	0	64	40
MLP	0.001	Adam	2048	100	0.1	128	-
CNN1	0.0001	Adam	1024	100	-	32	-
CTABL	0.01	Adam	2048	200	0.1	-	-
DAIN	0.0001	RMSprop	2048	100	0.5	512	-
DEEPLOB	0.01	Adam	1024	100	-	-	64
CNNLSTM	0.001	RMSprop	2048	100	0.1	32	32
CNN2	0.001	RMSprop	1024	100	32	-	-
TRANSLOB	0.001	Adam	2048	150	0.1	64	-
TLONBoF	0.0001	Adam	1024	100	0.5	512	-
BINCTABL	0.001	Adam	1024	200	0.1	-	-
DEEPLOBATT	0.001	Adam	1024	100	-	-	64
DLA	0.001	Adam	1024	100	0.5	-	100

Table 11: Hyperparameters for Mid-Price Return Forecasting on the FI-2010 Dataset

	FI							
Model	**Learning Rate**	**Optimizer**	**Batch Size**	**Epochs**	**Dropout**	**MLP Hidden**	**RNN Hidden**	**Transformer Hidden**
MLP	0.0001	Adam	32	50	0.1	128	-	-
LSTM	0.01	Adam	64	50	0.2	64	32	-
CNN1	0.0001	Adam	32	50	-	-	32	-
BINCTABL	0.0001	Adam	64	50	0.1	128	32	-
DAIN	0.0001	Adam	128	50	0.5	-	-	-
TRANSLOB	0.0001	Adam	128	50	0.1	64	-	60
PATCHTST	0.0001	Adam	128	50	0.3	-	-	128
DLINEAR	0.0001	Adam	128	50	-	-	-	-
ITRANSFORMER	0.0001	Adam	128	50	0.1	-	-	512
TIMEMIXER	0.0001	Adam	128	50	0.1	16	-	-

C Additional MPRF Results on Multivariate Synthetic Datasets

To further demonstrate CVML's ability to capture cross-variate correlation, we generate a synthetic dataset and test CVML's performance on it. Compared to real datasets with latent cross-variate correlations such as FI-2010 and CHF-2023, the synthetic dataset has well-defined cross-variate correlations specified in the data generation code. The synthetic dataset's target is a synthetic electricity price, the other variates are electricity load, electricity production and temperature. We generate the data in a way that the temperature affects the electricity load (e.g. cold and hot

Table 12: **Time Series Model Performance with and without CVML** on the Synthetic Dataset using basic LOB features.

Model	MSE (↓)			Corr (↑)			R^2 (↑)		
	K=1	K=5	K=10	K=1	K=5	K=10	K=1	K=5	K=10
PatchTST-CVML	**0.807**	**0.7538**	**0.7589**	**0.4298**	**0.485**	**0.48**	**0.1818**	**0.2346**	**0.2296**
PatchTST	1.0508	1.0441	1.0382	0.3907	0.3915	0.3991	-0.0654	-0.0602	-0.054
DLinear-CVML	**0.7825**	**0.7733**	**0.8743**	**0.4585**	**0.4639**	**0.3364**	**0.2067**	**0.2147**	**0.1124**
DLinear	0.8861	0.8847	0.8868	0.3188	0.3209	0.3159	0.1016	0.1017	0.0997
iTransformer-CVML	**0.8544**	**0.8183**	**0.8179**	**0.4011**	**0.4235**	**0.4317**	**0.1337**	**0.1691**	**0.1697**
iTransformer	1.1667	1.161	1.1706	0.339	0.3627	0.3628	-0.1829	-0.1789	-0.1884
TimeMixer-CVML	**0.7469**	0.7704	**0.7539**	**0.493**	0.4693	**0.4865**	**0.2427**	0.2177	**0.2347**
TimeMixer	0.7622	**0.7596**	0.7781	0.4774	**0.4782**	0.4586	0.2272	**0.2287**	0.2101

temperature increase the electricity load), the electricity load and the production affect the electricity price (e.g. higher load increases the price and higher production lower the price).

C.1 Synthetic Time Series Data Generation

The synthetic dataset consists of multiple time series components with complex relationships and non-linear interactions. The data generation process follows a hierarchical structure where intermediate variables influence the final target variable (electricity price). There are totally six varieties: electricity price, load, production, temperature, supply_margin, price_volatility. These varieties have cross-variate correlations. For example, an increasing load of electricity leads to an increasing electricity price. An increasing production of electricity leads to a decreasing electricity price. When the temperature is too high or too cold, the load will increase. We introduce the detailed generation process of each variate as follows.

Temperature Generation: Let $t \in \{0, 1, ..., n-1\}$ represent the time index for n hourly observations. The temperature time series T_t is generated as:

$$T_t = 20 + 10\sin(\text{seasonal_cycle}) + 5\sin(\text{daily_cycle}) + \epsilon_T + \delta_H - \delta_C \tag{1}$$

where:

- seasonal_cycle $= \frac{2\pi t}{24 \times 365.25}$
- daily_cycle $= \frac{2\pi t}{24}$
- $\epsilon_T \sim \mathcal{N}(0, 2^2)$ represents random fluctuations
- $\delta_H \sim \text{Bernoulli}(0.05) \times 8$ represents heat waves
- $\delta_C \sim \text{Bernoulli}(0.05) \times 8$ represents cold snaps

Load Generation: The electricity load L_t is modeled as (including base load, daily pattern, AC usage, heating, seasonal pattern, weekday effect, and a random noise):

$$\begin{aligned} L_t = 1000 + 200\sin(\text{daily_cycle}) + 150\mathbb{1}_{T_t>25} + 100\mathbb{1}_{T_t<10} \\ + 100\sin(\text{seasonal_cycle}) + 50\mathbb{1}_{\text{weekday}} + \epsilon_L \end{aligned} \tag{2}$$

where $\epsilon_L \sim \mathcal{N}(0, 30^2)$ and $\mathbb{1}$ represents the indicator function.

Production Generation: The production capacity P_t is generated through multiple components:

$$P_t = (1.1L_t + 100\sin(\text{daily_cycle}) + \epsilon_P) \times M_t \times O_t \tag{3}$$

where:

- $\epsilon_P \sim \mathcal{N}(0, 30^2)$
- $M_t \sim \text{Categorical}([1.0, 0.7], [0.9, 0.1])$ represents maintenance periods
- $O_t \sim \text{Categorical}([1.0, 0.3], [0.98, 0.02])$ represents outages

Table 13: **Time Series Model Performance with and without CVML** on the synthetic electricity price dataset. The % column indicates the percentage improvement from adding CVML.

Model	MSE (↓) K=1	K=2	K=3	K=5	K=10	%	Corr (↑) K=1	K=2	K=3	K=5	K=10	%	R^2 (↑) K=1	K=2	K=3	K=5	K=10	%
PatchTST-CVML	**0.653**	**1.071**	**1.370**	**1.893**	**2.646**	3.1	**0.070**	**0.113**	**0.165**	**0.191**	**0.241**	86.2	**0.005**	**0.007**	**0.028**	**0.033**	**0.054**	958.3
PatchTST	0.654	1.077	1.406	1.949	2.795		0.081	0.082	0.079	0.092	0.085		0.003	0.002	0.002	0.004	0.001	
DLinear-CVML	**0.650**	**1.042**	**1.352**	**1.796**	**2.548**	5.9	**0.104**	**0.192**	**0.205**	**0.291**	**0.313**	174.9	**0.010**	**0.035**	**0.040**	**0.082**	**0.089**	814.3
DLinear	0.652	1.073	1.402	1.945	2.782		0.080	0.081	0.074	0.083	0.084		0.006	0.006	0.005	0.006	0.005	
iTransformer-CVML	**0.654**	1.084	**1.402**	**2.002**	**2.649**	14.6	**0.054**	**0.070**	**0.088**	**0.121**	**0.249**	140.5	**0.002**	**-0.005**	**0.005**	**-0.024**	**0.053**	104.8
iTransformer	0.683	**1.183**	1.582	2.279	3.401		0.045	0.045	0.033	0.063	0.056		-0.041	-0.096	-0.123	-0.165	-0.216	
TimeMixer-CVML	**0.642**	**1.033**	**1.329**	**1.807**	**2.494**	4.6	**0.160**	**0.221**	**0.257**	**0.298**	**0.353**	61.1	**0.022**	**0.043**	**0.056**	**0.076**	**0.109**	194.2
TimeMixer	0.657	1.075	1.394	1.888	2.643		0.083	0.110	0.135	0.201	0.271		-0.001	0.004	0.011	0.035	0.055	

Electricity Price Generation (Target Variable): The final price Y_t is generated through a complex interaction of components:

$$\begin{aligned} Y_t = ((&50 + 0.08L_t - 0.04P_t + 0.7(T_t - 20)^2 \\ &+ 15\sin(\text{daily_cycle}) + 10\sin(\text{seasonal_cycle})) \\ &\times R_t \times S_t \times D_t + \epsilon_Y) \end{aligned} \tag{4}$$

where:

- $R_t \sim \text{Categorical}([1.0, 1.5, 0.7], [0.7, 0.15, 0.15])$ represents regime changes
- S_t represents price spikes triggered by conditions:

$$S_t = \begin{cases} \sim \text{Categorical}([1.0, 2.5], [0.7, 0.3]) & \text{if } C_t = 1 \\ 1.0 & \text{otherwise} \end{cases}$$

  where $C_t = 1$ if any of the following conditions are met:
  - $L_t/P_t > 0.9$ (high demand relative to production)
  - $T_t > 30$ (very hot weather)
  - $T_t < 0$ (very cold weather)
  - $P_t/P_{\text{base},t} < 0.5$ (significant production issues)
- $D_t \sim \text{Categorical}([1.0, 0.4], [0.97, 0.03])$ represents sudden price drops
- $\epsilon_Y \sim \mathcal{N}(0, (0.3 \times 100)^2)$ represents price noise

C.1.1 Derived Features

Additional features are computed from the primary variables:

$$\text{supply_margin}_t = \frac{P_t - L_t}{L_t}$$

$$\text{price_volatility}_t = \sigma(\{Y_{t-23}, ..., Y_t\})$$

where σ represents the rolling standard deviation over a 24-hour window.

As shown in Table 13, this synthetic time series dataset, although it was defined in a specific domain (electricity) for better interpretability, includes generic patterns that could be found across different domains of multivariate time series. Specifically, it includes the following patterns: multiple seasonality (e.g. weekly/monthly sales cycles, weekday/weekend differences in web traffic), non-linear relationships, temporary anomalies and recoveries (e.g. electricity outage), periods of high volatility followed by calmer periods (e.g. viral content spread on social media), multi-factor interactions (e.g. inventory-price-demand relationships in supply chain) and stochastic components that mirror real-world randomness. CVML's good performance on this synthetic dataset shows meaningful values for the broader time series forecasting field.

D Additional MPTP results on Bitcoin Dataset

We conduct further MPTP benchmark experiments on a public Bitcoin LOB datasets [3]. Our conclusion from benchmarking on the FI and CHF datasets is still valid on the crypto dataset, which is that there is limited generalizability of current LOB model architectures and the underlying characteristics of LOB data from different assets are different. As shown in Table 14, the ranking of the LOB model performance is also different from the one on the FI-2010 dataset and CHF-2023 dataset.

[3] https://www.kaggle.com/datasets/siavashraz/bitcoin%2Dperpetualbtcusdtp%2Dlimit%2Dorder%2Dbook%2Ddata/data

Table 14: **Mid-price Trend Prediction F1 Scores (Mean&Standard Deviation) on Basic LOB data + time-insensitive features + time-sensitive features**. We provide the F1 scores on mid-price trend predictions across horizons {1,2,3,5,10} on for the Bitcoin LOB dataset. The model performance ranking is not consistent with that on the FI-2010 and CHF-2023 datasets , further confirming that models' prediction power for one asset is not automatically transferable to another asset.

Model	K=1	K=2	K=3	K=5	K=10	Avg
MLP	92.4 (0.2)	93.2 (0.1)	93.8 (0.1)	94.3 (0.1)	95.3 (0.0)	93.8
LSTM	86.2 (2.8)	94.7 (0.2)	95.2 (0.5)	96.3 (0.1)	97.2 (0.1)	94.0
CNN1	94.5 (0.7)	96.3 (0.2)	96.9 (0.2)	96.9 (0.1)	97.7 (0.1)	96.5
CTABL	53.9 (0.2)	64.3 (0.0)	72.1 (0.2)	80.9 (0.2)	92.3 (0.2)	72.7
DeepLOB	**97.9** (0.1)	**98.4** (0.0)	98.5 (0.1)	98.1 (0.0)	98.3 (0.0)	**98.3**
DAIN	34.7 (0.3)	53.3 (0.4)	66.9 (0.6)	79.8 (0.1)	87.1 (0.1)	64.4
CNN-LSTM	97.2 (0.1)	97.7 (0.2)	97.9 (0.1)	97.7 (0.1)	98.1 (0.0)	97.7
CNN2	97.1 (0.1)	98.1 (0.0)	98.2 (0.1)	97.9 (0.1)	98.1 (0.0)	97.9
TransLOB	95.9 (0.8)	98.2 (0.2)	98.3 (0.2)	98.0 (0.2)	98.4 (0.1)	97.8
TLONBOF	56.5 (1.0)	70.0 (0.9)	78.1 (0.7)	88.2 (0.3)	94.8 (0.1)	77.5
BinCTABL	50.5 (0.4)	60.3 (0.5)	65.1 (1.7)	73.1 (0.2)	90.4 (0.4)	67.9
DeepLOBAtt	97.6 (0.2)	98.1 (0.5)	**98.7** (0.2)	**98.4** (0.1)	**98.5** (0.1)	**98.3**
DLA	57.1 (2.0)	69.2 (0.3)	74.4 (0.4)	81.0 (0.2)	89.0 (0.2)	74.2

E FULL MPTP RESULTS ON DIFFERENT FEATURE SETS

Table 15 and Table 16 include the full results for LOB models on the MPTP task on the basic feature set and the basic+time_insensitive feature set. They support Figure 3.

Table 15: **Mid-price Trend Prediction F1 Scores (Mean&Standard Deviation) on Basic LOB data**. 13 models relevant in the literature are benchmarked to compare their F1 scores on across horizons {1,2,3,5,10} on the basic LOB feature set of the FI-2010 and CHF-2023 datasets. For each horizon, the best model is bolded, and the next best model is underlined.

	FI-2010					CHF-2023				
Model	K=1	K=2	K=3	K=5	K=10	K=1	K=2	K=3	K=5	K=10
MLP	35.014 (4.545)	42.242 (2.038)	46.759 (0.918)	46.061 (0.904)	47.210 (2.887)	38.788 (1.487)	44.328 (0.760)	45.958 (0.518)	45.650 (0.210)	37.952 (1.522)
LSTM	64.809 (1.377)	57.882 (0.613)	65.205 (0.198)	66.898 (0.747)	58.850 (0.926)	47.280 (0.459)	49.762 (0.210)	50.710 (0.265)	48.411 (0.202)	43.972 (0.522)
CNN1	27.608 (0.000)	30.815 (0.096)	54.783 (4.891)	62.882 (0.828)	63.955 (0.705)	42.357 (1.899)	47.936 (0.912)	49.203 (0.587)	47.676 (0.390)	43.661 (1.056)
CTABL	67.353 (0.585)	60.531 (0.213)	66.186 (0.198)	70.736 (0.367)	71.244 (1.092)	43.902 (2.157)	45.986 (0.675)	46.429 (1.193)	46.432 (0.417)	43.867 (0.879)
DEEPLOB	70.018 (1.160)	62.357 (0.577)	70.403 (1.010)	75.924 (0.089)	77.551 (0.285)	45.926 (2.140)	48.324 (1.646)	49.166 (1.854)	47.465 (1.230)	41.233 (1.290)
DAIN	79.767 (0.050)	70.202 (0.110)	79.851 (0.036)	87.041 (0.029)	91.816 (0.075)	45.130 (0.759)	49.014 (0.274)	49.290 (0.211)	46.928 (0.211)	40.445 (0.529)
CNNLSTM	27.620 (0.000)	29.656 (1.210)	34.060 (2.43)	44.248 (10.898)	54.872 (6.768)	45.067 (3.705)	47.342 (1.175)	48.556 (2.505)	47.740 (1.173)	**46.209 (0.198)**
CNN2	27.620 (0.000)	27.914 (1.978)	33.315 (1.909)	50.086 (11.537)	61.930 (5.501)	42.357 (1.899)	47.936 (0.912)	49.203 (0.587)	47.676 (0.390)	43.661 (1.056)
TRANSLOB	51.020 (12.632)	40.976 (8.987)	50.876 (10.709)	60.748 (1.541)	59.715 (0.859)	**49.189 (1.454)**	**50.379 (0.618)**	50.739 (0.323)	48.690 (0.426)	45.931 (0.273)
TLONBoF	37.549 (1.759)	40.181 (3.560)	41.551 (2.348)	48.991 (1.371)	60.702 (6.665)	43.961 (1.370)	45.935 (0.839)	46.556 (0.382)	45.935 (0.970)	43.544 (1.084)
BINCTABL	**80.985 (0.055)**	**71.168 (0.371)**	**80.734 (0.044)**	**87.553 (0.037)**	**92.074 (0.042)**	44.800 (0.608)	46.041 (0.647)	46.063 (0.479)	45.759 (0.182)	44.124 (0.263)
DEEPLOBATT	69.435 (0.011)	62.936 (0.015)	59.100 (0.203)	73.083 (0.007)	77.028 (0.020)	47.955 (1.213)	50.176 (0.993)	**50.744 (0.332)**	**49.104 (0.231)**	43.938 (1.4981)
DLA	76.410 (0.028)	65.966 (0.012)	77.858 (0.006)	85.713 (0.005)	51.617 (0.010)	44.136 (5.060)	48.060 (3.583)	47.797 (3.055)	46.992 (0.337)	38.347 (1.026)
Mean	55.016	58.290	65.311	69.695	72.525	46.660	48.685	49.246	47.398	43.357

Table 16: **Mid-price Trend Prediction F1 Scores (Mean&Standard Deviation) on Basic LOB data + time-insensitive features**. The F1 scores across horizons {1,2,3,5,10} for the FI-2010 and CHF-2023 datasets on the basic LOB + time-insensitive feature set of the FI-2010 and CHF-2023 datasets. For each horizon, the best model is bolded, and the next best model is underlined.

	FI-2010					CHF-2023				
Model	K=1	K=2	K=3	K=5	K=10	K=1	K=2	K=3	K=5	K=10
MLP	45.849 (1.995)	44.475 (1.324)	47.855 (0.570)	44.487 (1.209)	49.629 (0.564)	40.343 (1.641)	44.818 (2.101)	45.975 (0.924)	45.336 (0.712)	37.242 (0.679)
LSTM	74.261 (0.051)	65.072 (0.223)	72.295 (0.579)	76.537 (1.496)	60.141 (1.992)	48.779 (0.556)	50.464 (0.495)	51.178 (0.154)	48.805 (0.108)	45.511 (1.654)
CNN1	60.554 (13.194)	61.928 (0.308)	70.664 (0.174)	77.906 (0.520)	80.107 (1.099)	47.367 (1.303)	49.236 (0.847)	50.400 (0.461)	48.316 (0.376)	44.365 (1.459)
CTABL	77.336 (0.107)	68.265 (0.483)	76.910 (0.217)	82.573 (0.388)	84.356 (0.203)	44.012 (2.707)	46.187 (1.791)	45.916 (0.567)	45.667 (0.490)	43.930 (0.634)
DEEPLOB	79.047 (0.075)	69.773 (0.216)	78.797 (0.087)	85.249 (0.090	88.471 (0.177)	46.316 (2.756)	47.581 (1.547)	49.692 (1.162)	47.435 (1.381)	41.116 (2.251)
DAIN	79.935 (0.028)	**80.190 (0.102)**	79.905 (0.072)	87.151 (0.022)	92.222 (0.024)	45.773 (0.391)	48.475 (0.214)	48.607 (0.281)	46.767 (0.231)	41.913 (1.216)
CNNLSTM	38.279 (12.537)	36.006 (1.015)	36.181 (0.456)	36.845 (0.342)	76.312 (1.819)	44.696 (0.908)	48.517 (0.877)	49.729 (0.439)	47.392 (0.448)	45.782 (0.610)
CNN2	28.558 (0.666)	33.529 (1.818)	36.469 (0.541)	39.401 (2.606)	66.267 (7.428)	45.365 (1.145)	48.162 (0.672)	49.027 (0.372)	46.843 (1.082)	44.855 (0.766)
TRANSLOB	69.436 (5.014)	59.600 (3.084)	70.478 (0.758)	75.294 (2.101)	71.545 (6.861)	**51.213(0.655)**	**52.294 (0.202)**	**51.752 (0.293)**	48.984 (0.269)	**46.418 (0.540)**
TLONBoF	43.105 (8.616)	40.792 (1.035)	51.367 (1.609)	59.992 (3.051)	65.607 (3.224)	49.888 (0.896)	49.506 (0.680)	49.426 (0.288)	47.120 (0.375)	44.761 (0.195)
BINCTABL	**81.024 (0.034)**	71.547 (0.265)	**80.896 (0.016)**	**87.849 (0.055)**	**92.541 (0.081)**	46.309 (1.214)	46.844 (0.981)	47.218 (1.409)	46.388 (0.567)	44.313 (0.295)
DEEPLOBATT	75.652 (0.013)	58.878 (0.111)	68.990 (0.103)	67.337 (0.119)	56.923 (0.034)	48.388 (0.578)	50.737 (0.260)	51.192 (0.567)	**49.229 (0.214)**	43.939 (0.886)
DLA	77.143 (0.004)	67.721 (0.007)	78.241 (0.002)	85.417 (0.002)	58.703 (0.005)	48.128 (0.516)	50.080 (0.309)	50.092 (0.398)	47.888 (0.734)	39.499 (0.818)
Mean	63.860	58.290	65.311	69.695	72.525	46.660	48.685	49.254	47.398	43.357

F COMPUTATIONAL RESOURCES

For all the experiments, we use a computation cluster with 1 node of 8 Nvidia A100 GPUs and 3 nodes of 8 Nvidia 2080Ti GPUs. The RAM is 1TB per node and there are 96 CPU cores per node.

G FULL CVML RESULTS ON MPRF

Table 17 includes the full mean and standard deviation results for the time series models using CVML. Table 18 contains ablation results on CVML using its two ablated variants, CVML-abla1 and CVML-abla2. CVML-abla1 focuses exclusively on cross-variate correlations ang ignores temporal relationships by performing convolution with a kernel size of 1. CVML-abla2 uses depthwise convolution. We set the number of groups in the convolution equal to the number of input channels, which allows the model to focus exclusively on temporal correlations and ignore cross-variate correlations. The results demonstrate how these ablated versions compare to the original CVML design, highlighting the architecture's dual-correlation approach and its impact on model performance.

Table 17: **Full MPRF Results for Table 6 with Mean and Std on Basic LOB data with CVML**

Model	Metric	K=1	K=2	K=3	K=5	K=10
PatchTST	MSE	0.653 (0.004)	1.071 (0.016)	1.370 (0.024)	1.893 (0.043)	2.646 (0.018)
	Corr	0.070 (0.036)	0.113 (0.037)	0.165 (0.061)	0.191 (0.045)	0.241 (0.011)
	R^2	0.005 (0.005)	0.007 (0.015)	0.028 (0.017)	0.033 (0.022)	0.054 (0.007)
DLinear	MSE	0.650 (0.003)	1.042 (0.004)	1.352 (0.011)	1.796 (0.008)	2.548 (0.028)
	Corr	0.104 (0.021)	0.192 (0.011)	0.205 (0.021)	0.291 (0.007)	0.313 (0.024)
	R^2	0.010 (0.004)	0.035 (0.004)	0.040 (0.007)	0.082 (0.004)	0.089 (0.010)
iTransformer	MSE	0.654 (0.002)	1.084 (0.014)	1.402 (0.007)	2.002 (0.046)	2.649 (0.014)
	Corr	0.054 (0.039)	0.070 (0.007)	0.088 (0.014)	0.121 (0.027)	0.249 (0.018)
	R^2	0.002 (0.024)	-0.005 (0.013)	0.005 (0.005)	-0.024 (0.024)	0.053 (0.005)
TimeMixer	MSE	0.642 (0.004)	1.033 (0.006)	1.329 (0.007)	1.807 (0.029)	2.494 (0.056)
	Corr	0.160 (0.009)	0.221 (0.009)	0.257 (0.007)	0.298 (0.009)	0.353 (0.008)
	R^2	0.022 (0.006)	0.043 (0.005)	0.056 (0.005)	0.076 (0.015)	0.109 (0.020)

Table 18: **MPRF Results on Ablated CVMLs**. The comparison to CVML's results in Table 6 demonstrates CVML's ability to capture cross-variate correlations and temporal correlations.

		CVML-abla1 (Cross-variate)					CVML-abla2 (Temporal)				
Model	Metric	K=1	K=2	K=3	K=5	K=10	K=1	K=2	K=3	K=5	K=10
PatchTST	MSE	0.6565	1.1168	1.3936	1.9292	2.7402	0.6531	1.0763	1.4063	1.9501	2.7983
	Corr	0.0102	0.0572	0.1080	0.1204	0.1443	0.0826	0.0813	0.0784	0.0909	0.0835
	R^2	-0.0005	-0.0349	0.0107	0.0138	0.0206	0.0046	0.0026	0.0016	0.0031	-0.0002
DLinear	MSE	0.6501	1.0681	1.3680	1.842	0.25945	0.6526	1.0727	1.4016	1.9446	2.7819
	Corr	0.0789	0.1031	0.1750	0.243	0.2710	0.0773	0.0810	0.0744	0.0814	0.0860
	R^2	0.009	0.0102	0.0288	0.0283	0.0726	0.0055	0.0060	0.0048	0.0059	0.0057
iTransformer	MSE	0.7341	1.272	1.488	1.9398	3.1331	0.6967	1.1605	1.5865	2.2183	3.4300
	Corr	0.0372	0.061	0.069	0.1178	0.1000	0.0368	0.0460	0.0280	0.0646	0.0517
	R^2	-0.1188	-0.179	-0.056	0.0084	-0.1199	-0.0617	-0.0754	-0.1263	-0.1340	-0.2260
TimeMixer	MSE	0.637	1.038	1.329	1.806	2.508	0.6558	1.0791	1.4006	1.9031	2.6729
	Corr	0.180	0.2142	0.257	0.296	0.342	0.0837	0.1047	0.1186	0.2114	0.2617
	R^2	0.030	0.039	0.057	0.077	0.104	0.0006	0.0000	0.0271	0.0271	0.0446