Learning Differential Operators for Interpretable Time Series Modeling

Yingtao Luo[1], Chang Xu[2], Yang Liu[2], Weiqing Liu[2], Shun Zheng[2], Jiang Bian[2]

[1]Carnegie Mellon University [2]Microsoft Research Asia

Background

Partial Differential Equations (PDEs) are widely used to model the highly nonlinear sequential patterns. They can reveal the inherent differential relations of variables (functions) and their rates of change (derivatives) in any local area of the dynamical system.

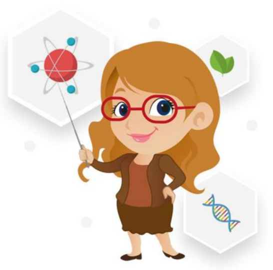
With assumptions, economists use equations to predict options:

Assume that the differentiation of stock price S is

$$dS = \mu S dt + \sigma S dz$$

Assume f is the call options of S as a function of S and t, we have Black–Scholes equation

$$df = \left(\frac{\partial f}{\partial S}\mu S + \frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)dt + \frac{\partial f}{\partial S}\sigma S dz$$



Based on observations, scientists explain electromagnetism by:

$$\nabla \times \boldsymbol{H} = \boldsymbol{J} + \frac{\partial \boldsymbol{D}}{\partial t}$$
$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$$
$$\nabla \cdot \boldsymbol{B} = 0$$
$$\nabla \cdot \boldsymbol{D} = \rho$$

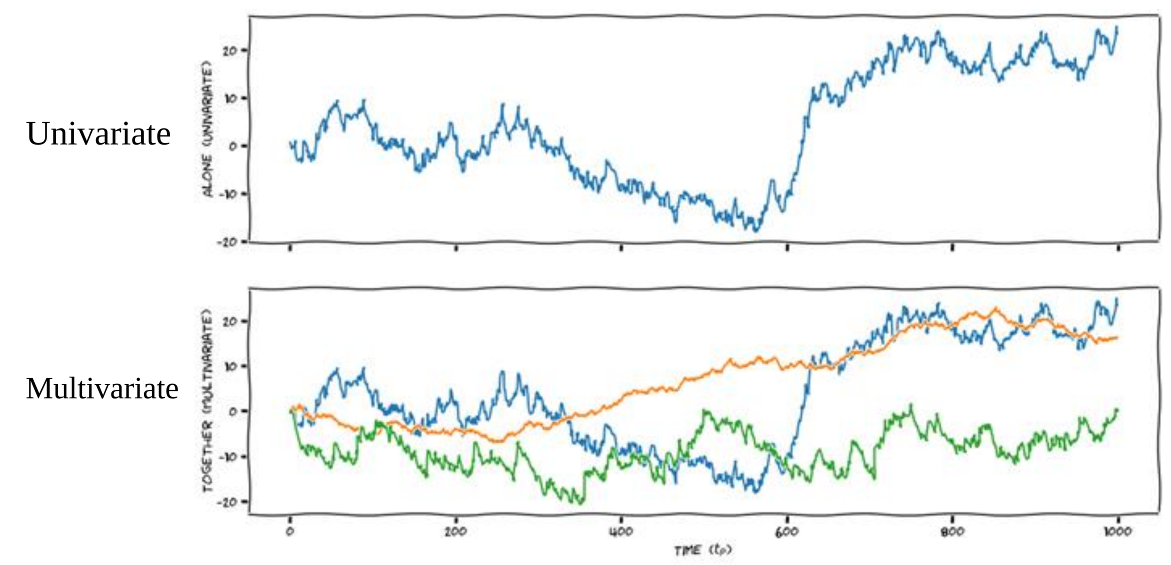
Challenges

1. Interpretability versus Accuracy

Deep learning models excel in model accuracy, but the calculations are conducted in the latent space, making it increasingly hard for people to understand its mechanism and trust its prediction.

2. Multivariate Time Series

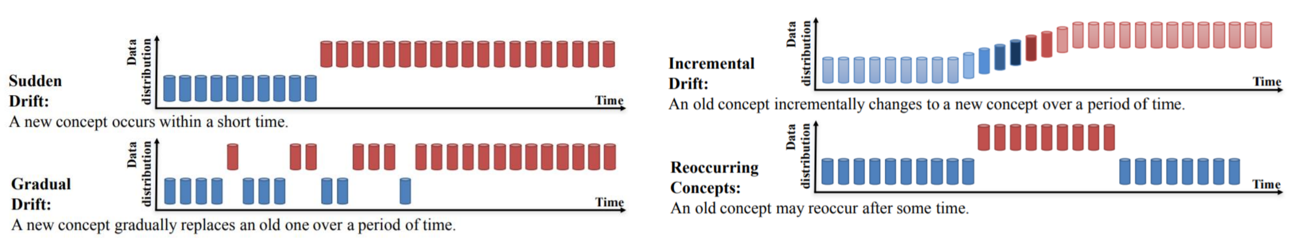
Neural ODE can only model univariate time series, and it is only a black-box model without interpretable mathematical expression.



Multivariate TS involve high-dimensional differential relations. It is challenging to understand what variables dominate the dynamics and how do they depend on each other.

3. Concept Drift/Distribution Shift

Many unobserved & unpredictable factors can change the pattern. A single PDE is unlikely to adapt to the ever-changing dynamics.



Methodology

PDE-Net proposes convolutional kernel as learnable differential method. However, it can hardly handle higher-dimensional data.

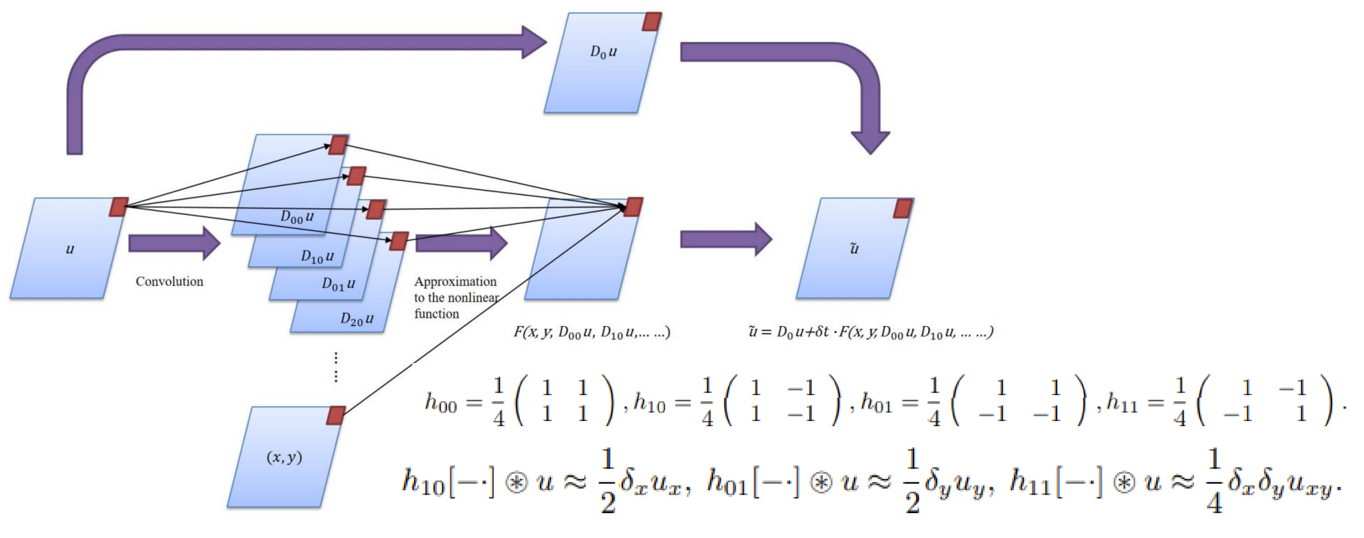


Figure 1: The schematic diagram of a δt-block.

The various combinations of different variables of the multivariate time series can lead to an exponentially growing number of differential terms. It is computationally intractable to include all possible terms.

Therefore, we propose the P-block to adaptatively learn the differential operators that explain the mechanism of multivariate time series.

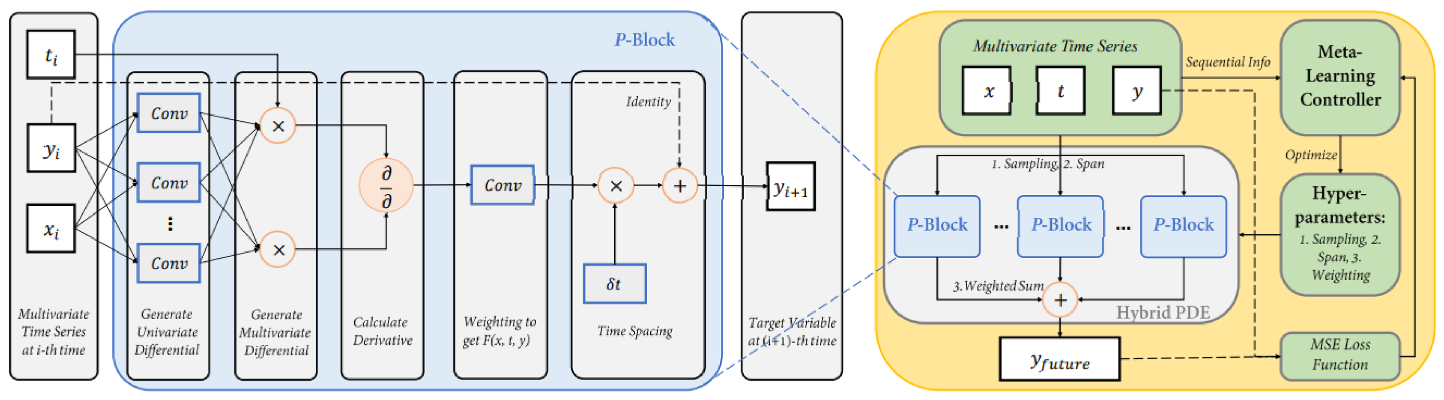


Figure 1: The learning framework of P-block (left) and the dynamic PDE model (right). The P-block can take the multivariate time series as input to generate the function F that parameterizes the time-evolving system dynamics, which can iteratively forecast future points via time spacing. The meta-learning controller takes various combinations of time series and hyperparameters as inputs to learn to capture the current dynamics and predict the loss. It guides us to search for the optimal hyperparameters.

Our P-block is optimized by sparse regression, where each term contains highly nonlinear differential representations.

Goal: learn a close-form PDE $[\frac{\partial y}{\partial t} = a \times \frac{\partial y}{\partial x} + b \times \frac{\partial^2 y}{\partial x^2} + c \times \frac{\partial^3 y}{\partial x^3} + \cdots]$

Requirement: ***Concise (Interpretable), Good Fitting (Performance)***

Sparse regression ensures the optimization

$$W^* = \underset{W}{\operatorname{argmin}} \left\| \frac{\partial y}{\partial t} - XW \right\|_2^2 + \lambda \|W\|_1,$$

A new ***P-block*** is proposed with additive and multiplicative representation of Conv operators:

$$F(x,t,y) = \underbrace{\sum_{c=1}^{N_c} f_c}_{\text{Weight Sum}} \cdot [\underbrace{\prod_{l=1}^{N_l}}_{\text{Higher-order}} \underbrace{\prod_{j=1}^{k}}_{\text{Multi-variate}} (\underbrace{K_{(c,l)} \circledast y}_{\text{Derivative}}) / (\underbrace{K_{(c,l)} \circledast x_j}_{\text{Differential}}) \underbrace{T}_{\text{Time info}}],$$

Once we obtain the PDE, the ***inference*** can be very fast with *Euler Time Stepping*

$$y_{i+1} = y_i + \hat{F}(y;\omega)\delta t,$$

First, the target variable y and the correlated variables x are processed by differential Conv operators $K_{(c,l)}$ to create terms such as ∂y and ∂x_j.
Second, these terms are processed by element-wise product operators to create multivariate polynomials of differentials like ∂y^2 and $\partial x_1 \partial x_2$.
Third, by division, we create derivatives such as $\frac{\partial y^2}{\partial x_1 \partial x_2}$.
Last, by weighting, we obtain the additive combination of these terms.

We use meta-learning of a hybrid PDE model of multiple PDEs.
We adjust the sampling rates, weights, and the time span of each PDE to capture the concept drift and adapt to the ever-changing dynamics.

Theoretical Analysis

For $N \times 1$ Conv filter q, define the vector of q as

$$V(q) = (v_i)_{N \times 1}, \tag{9}$$

where

$$v_i = \frac{1}{i!} \sum_{g=-\frac{N-1}{2}}^{\frac{N-1}{2}} g^i q[g], i = 0, 1, \ldots, N-1 \tag{10}$$

By Taylor's expansion, we can formulate the convolution on any smooth function f, i.e. the differential operator on f, as

$$\begin{aligned} L(y) &= \sum_{g=-\frac{N-1}{2}}^{\frac{N-1}{2}} q[g] f(x + g\delta x) \\ &= \sum_{g=-\frac{N-1}{2}}^{\frac{N-1}{2}} q[g] \sum_{i=0}^{N-1} \prod_{j=0}^{k} \frac{\partial^i f}{\partial^i x_j}\bigg|_x \frac{g^i}{i!} \delta x^i + o(\sum_{j=0}^{N-1} |\delta x_j|^{N-1}) \\ &= \sum_{i=0}^{N-1} v_i \delta x_i \cdot \prod_{j=0}^{k} \frac{\partial^i f}{\partial^i x}\bigg|_x + o(\sum_{j=0}^{N-1} |\delta x_j|^{N-1}). \end{aligned} \tag{11}$$

We can conclude that filter q can approximate any differential operator with the prescribed order of accuracy, as a "universal polynomial approximator" for TS. The error is bounded by the minimal terms related to variable difference δx. Therefore, for time series that are generally smooth, P-block can work well.

Results

P-block is comparable to state-of-the-art models and provides understandable mathematical expressions.

Table 1: Performances on different models for multi-step time series forecasting in test RMSE. Best performances are indicated by bold fonts and the strongest baselines are underlined. RMSE times the modifier is authentic RMSE.

Dataset	Modifier	LSTM	ARIMA	Prophet	Neural ODE	ODE-RNN	ConvTrans	DeepAR	Ours
Synthetic	$\times 10^{-8}$	5.0148	5.1317	5.0192	4.9622	4.9454	4.8586	<u>4.8149</u>	**4.7933**
Orderbook	$\times 10^{-6}$	2.8960	6.0201	4.0005	2.4719	2.5523	<u>0.9690</u>	1.6672	**0.8795**
MISO	$\times 10^{-2}$	5.4411	5.2085	5.1413	5.2085	4.7990	<u>4.0583</u>	4.2171	4.1489
PhysioNet	$\times 10^{-3}$	3.6438	3.5597	3.2567	2.4400	2.3551	<u>2.3291</u>	2.4673	**2.3266**

Table 2: Performances on different models for single-step time series forecasting in test RMSE. Best performances are indicated by bold fonts and the strongest baselines are underlined. RMSE times the modifier is authentic RMSE.

Dataset	Modifier	LSTM	ARIMA	Prophet	Neural ODE	ODE-RNN	ConvTrans	DeepAR	Ours
Synthetic	$\times 10^{-8}$	4.9727	5.0254	5.0124	4.9030	4.8862	4.6472	<u>4.6036</u>	**4.4829**
Orderbook	$\times 10^{-9}$	3.2354	3.4782	4.0104	2.6025	3.0146	<u>2.4399</u>	2.7490	**1.5495**
MISO	$\times 10^{-2}$	5.2237	5.0837	4.9920	4.7274	4.5962	<u>3.8489</u>	4.0311	**3.6263**
PhysioNet	$\times 10^{-3}$	3.2853	3.2161	3.1566	2.3555	2.3282	<u>2.1460</u>	2.1805	**2.0260**

Table 3: Performances on our model for the multi-step time series forecasting in test RMSE w/o meta-learning controller.

Dataset	Modifier	Meta	Hybrid	Single
Synthetic	$\times 10^{-8}$	**4.7933**	4.9284	4.9581
Orderbook	$\times 10^{-6}$	**0.8795**	1.6982	1.9346
MISO	$\times 10^{-2}$	**4.1489**	5.8392	6.1908
PhysioNet	$\times 10^{-3}$	**2.3266**	2.9804	3.4947

Table 5: The different time series dynamics at different times of Orderbook dataset captured by the meta-learning model.

Stages	Partial Differential Equations
First	$\frac{\partial y}{\partial t} = 4.224 \times 10^{-7} y - 0.186 + 1.632 \frac{\partial x}{\partial t} - 0.023 \frac{\partial x}{\partial t} \cdot t$
Second	$\frac{\partial y}{\partial t} = 2.134 \frac{\partial x}{\partial t} - 1.562 \times 10^{-6} \frac{\partial x}{\partial t} \cdot x - 7.720 \times 10^{-7} \frac{\partial x}{\partial t} \cdot t$
Third	$\frac{\partial y}{\partial t} = -0.067 \frac{\partial x}{\partial t} \cdot x + 0.044 \frac{\partial x}{\partial t} + -0.007t + 0.003 \frac{\partial y}{\partial x} \cdot x$

Table 4: The mathematical form of the major trend captured by PDE for the last multi-step prediction of each sequence.

Dataset	Partial Differential Equations
Synthetic	$\frac{\partial^2 y}{\partial t^2} = 1.02 \frac{\partial^2 y}{\partial x_1^2} + 0.99 \frac{\partial^2 y}{\partial x_2^2}$
Orderbook	$\frac{\partial y}{\partial t} = -0.067 \frac{\partial x}{\partial t} \cdot x + 0.044 \frac{\partial x}{\partial t} + -0.007t + 0.003 \frac{\partial y}{\partial x} \cdot x$
MISO	$\frac{\partial y}{\partial t} = -1.157 \frac{\partial^2 y}{\partial x^2} + 1.144 \frac{\partial^2 y}{\partial x^2} \cdot t + 0.062 \frac{\partial x}{\partial t} \cdot t - 0.032 \frac{\partial y}{\partial x} \cdot x$
PhysioNet	$\frac{\partial y}{\partial t} = -0.274 \frac{\partial x_1}{\partial t} + 0.091 \frac{\partial x_2}{\partial t} \cdot t + 0.088 \frac{\partial y}{\partial x_2} - 0.062 \frac{\partial y}{\partial x_3} \cdot x$