VALUE LIFTING FOR MULTIVARIATE TIME-SERIES CLASSIFICATION: FROM FOURIER-TAYLOR STRUCTURE TO TRAINABLE HIGH-ORDER REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing multivariate time-series classification (MTSC) models mainly improve temporal dependency modeling and cross-variable interaction, but they leave underexplored whether the scalar values these encoders produce benefit from an explicit, trainable transformation before classification. Here, we study this question through *value lifting* and instantiate it with Fourier-Taylor High-Order Lifting (FT-Lift), which converts intermediate values into a structured collection of odd- and even-order responses and learns how to combine them. Specifically, after coefficient relaxation, the Fourier-Taylor derivation provides a structured route to the high-order representation. The results show that FT-Lift improves the complete MTSC system, transfers to conventional MTSC backbones with architecture-dependent gains, achieves the best average rank among the compared lifting formulations, and provides task-dependent benefits from high-order value transformation. Overall, the evidence supports FT-Lift as a useful trainable value-lifting component for MTSC rather than a uniquely Fourier-specific function family.

1 INTRODUCTION

Multivariate time series classification (MTSC) is prevalent and significant in applications such as smart factories, gesture recognition, and healthcare (Mohammadi Foumani et al., 2024). Existing MTSC models have substantially improved the learning of temporal patterns and relationships among variables. For example, convolutional models enlarge temporal receptive fields (Zhou et al., 2026; Luo & Wang, 2024), and Transformer-based models capture long-range dependencies and variable interaction (Zerveas et al., 2021; Nie et al., 2022; Foumani et al., 2024; Chen et al., 2024). Despite these advances, most existing designs primarily answer *where* information should interact and *how* those interactions should be aggregated.

Among them, a question is underexplored: *how should the value carried by an intermediate feature be represented before classification?* Consider a univariate component x_t of an MTS. Two classifiers may require similar temporal receptive fields but features with substantially different class-relevant nonlinear geometry: one may depend mainly on signed amplitudes, whereas another may emphasize magnitude- or energy-like responses such as x_t^2, or more selective high-order responses such as x_t^4. For instance, three-axis accelerometer observations can be transformed into semantically meaningful quantities (*e.g.*, roll and pitch angles) through nonlinear trigonometric mappings for predicting badminton receiving techniques. Similar effects can arise in multivariate motion, physiological, and spectral signals, where discriminative information depends jointly on how their amplitudes are nonlinearly represented.

We call this component *value lifting*: an element-wise map $\mathbf{h} \mapsto \phi(\mathbf{h})$ that exposes multiple nonlinear responses of the same intermediate value before temporal/channel features are pooled and classified. In this paper, our goal is to ask whether an explicit trainable value lifting pathway can be a useful part of an MTSC architecture.

Inspired by the complementary roles of Fourier and Taylor expansions in organizing nonlinear responses into harmonic components and explicit high-order terms, we instantiate value lifting from a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

finite Fourier series followed by a truncated Taylor expansion. To obtain a trainable neural representation, the analytically prescribed coefficients in this literal implementation (Strict-FT) are further relaxed; we call this Fourier-Taylor High-Order Lifting (FT-Lift) or Relaxed-FT.

Comprehensive experiments are conducted to investigate the effectiveness of FT-Lift in MTSC using the UEA benchmark and complementary synthetic datasets. Our results reveal a consistent but bounded role for value lifting in MTSC. VLNet, an MTSC architecture equipped with FT-Lift, reaches a mean accuracy of 0.79 on 26 UEA datasets, while system ablations and cross-architecture transfer show that its benefit is not confined to a single architecture. Under matched controls, Relaxed FT-Lift achieves the best average rank and improves over Strict FT-Lift on average, but remains statistically comparable to No Lift and parameter-matched polynomial lifting. Controlled high-order tasks show that classification becomes more difficult as the target order increases, whereas UEA sensitivity analyses reveal that the best-performing lifting order, factorization width, and temporal kernel size vary across tasks. Taken together, these findings support FT-Lift as a useful but task-dependent MTSC component whose effectiveness depends on both data characteristics and architectural configuration.

To summarize, our study made the following contributions: (1) We formulate *value lifting* as an explicit nonlinear representation stage in MTSC, positioned between temporal/channel feature extraction and classification, and instantiate it with a high-order pathway. (2) We derive FT-Lift from a Fourier-Taylor expansion while explicitly characterizing what coefficient relaxation changes. This prevents the relaxed layer from being misinterpreted as an exact Fourier approximation or as a function family larger than polynomial lifting. (3) Through same-protocol lifting controls, cross-architecture transfer, controlled high-order tasks, and 26-dataset sensitivity analyses, we characterize the empirical scope of value lifting: when it helps, when additional order does not help, and which conclusions are not supported by the current evidence.

2 Related Work

Multivariate Time-Series Classification. Traditional MTSC methods predict class labels by extracting discriminative patterns, distances, or features (Bagnall et al., 2018; Schäfer & Leser, 2017; Dempster et al., 2021). As MTS datasets grow in scale, these methods face increasing difficulty in handling the large number of candidate features. Inspired by the success of 2D convolutional networks (ConvNets) in computer vision, many studies have developed effective 1D ConvNets for MTSC through carefully designed learning or modeling strategies, including the squeeze-and-excitation paradigm (Karim et al., 2019), prototype learning (Zhang et al., 2020), shapelet extraction (Li et al., 2021), kernel size tuning (Tang et al., 2020; Zhou et al., 2026), and graph-based dependency modeling (Liu et al., 2024; Wang et al., 2024; Liu et al., 2026). The strong representation-learning capability of Transformer architectures (Lin et al., 2022) has further motivated a broad line of Transformer-based MTSC models (Zerveas et al., 2021; Liu et al., 2021; Chen et al., 2022; Zuo et al., 2023; Foumani et al., 2024; Chen et al., 2024; Xiao et al., 2024b; Le et al., 2024; Xiao et al., 2025; Wang et al., 2025c; Zhou et al., 2022; Nie et al., 2022; Zhang & Yan, 2023), while recent frequency-aware methods use spectral information to improve temporal representation (Eldele et al., 2024; Mu et al., 2025; Yu et al., 2025). These studies mainly improve temporal or variable dependency modeling. To emphasize the significance of value lifting, our study is complementary to temporal-backbone design; it focuses on an explicit nonlinear map applied to intermediate values.

Channel Independence and Channel Dependence. Channel Independence (CI) and Channel Dependence (CD) have been actively discussed in recent time series forecasting (TSF) studies. Since PatchTST (Nie et al., 2022) demonstrated superior performance on TSF, more CI-based models have been developed to better model channel-wise information without introducing distributional shift and noise from other channels (Das et al., 2023; Chi et al., 2024; Peiwen & Changsheng, 2023). Later, some researchers advocated that CD-based models can capture channel dependencies and bring more information to the downstream applications (Zhao & Shen, 2024; Abdelmalak et al., 2025). More recent studies indicate that hybrid designs can advance incremental effects on robustness, capacity, and convergence (Wang et al., 2025b;a; Chi et al., 2024; Li et al., 2023; Li & Xie). The incorporation can be sequential (Wang et al., 2025b;a), parallel (Chi et al., 2024; Li et al., 2023), and multi-step (Li & Xie). Inspired by the hybrid studies and the recent lightweight model EdgeMTSC (Zhou et al., 2026), we retain simple-but-effective depthwise convolutions as the

CI-based temporal implementation and treat them as a backbone component rather than the main contribution of this paper.

Fourier and Polynomial Representations. Fourier analysis has been widely studied in signal processing and machine learning for discovering periodic structures and frequency-domain characteristics. Classical Fourier theory demonstrates that sufficiently regular functions can be approximated by finite sums of sinusoidal components, enabling compact spectral representations of complex signals (Tolstov, 2012). In recent deep learning studies, Fourier-related representations have been adopted for sequence mixing, operator learning, periodic kernels, and spectral representation learning (Lee-Thorp et al., 2022; Li et al., 2020; Tompkins & Ramos, 2018; Liu et al., 2025a). Fourier features and sinusoidal representations can also change the frequency behavior of neural networks (Tancik et al., 2020; Sitzmann et al., 2020). On the other hand, Polynomial and higher-order networks (Chen et al., 2019; Chrysos et al., 2020; Xiao et al., 2024a; Xu et al., 2024) explicitly introduce products or polynomial terms to increase nonlinear mapping capacity, while learnable activation functions and KAN-style layers make scalar nonlinear transformations trainable (Agostinelli et al., 2014; Liu et al., 2025b). Our use of Fourier analysis is different from frequency-domain MTSC: the Fourier-Taylor expansion is used to expose a structured high-order dictionary. After relaxation, the resulting layer belongs to a polynomial family, which we state explicitly and test against direct polynomial controls.

3 Value Lifting for MTSC

3.1 Problem Formulation & Architecture

A multivariate time series is denoted by $X = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(d)}\} \in \mathbb{R}^{d\times l}$, where d and l are the number of variables and sequence length, respectively. Given labeled samples $\{(X_j, y_j)\}_{j=1}^{m}$, the goal is to learn a classifier $f(X)$ over M classes. Let $\mathbf{h}$ denote an intermediate representation produced by the temporal/channel encoder. We study an explicit value transformation

$$\phi : \mathbf{h} \mapsto \phi(\mathbf{h}) \tag{1}$$

before pooling and classification.

The backbone, as shown in Figure 4, follows the existing convolutional system. A pointwise projection is followed by a channel-specific depthwise convolution and a channel-shared depthwise convolution. The former learns variable-specific temporal filters, while the latter applies a shared temporal kernel across variables. Their intermediate responses are passed through the high-order lifting pathway and fused with a residual temporal branch before adaptive pooling and classification. Thus, the architecture separates temporal/channel extraction from value-domain transformation: the backbone determines *which temporal patterns are present*, while FT-Lift determines *how their scalar responses are nonlinearly represented*. For ease of presentation, we refer to this backbone as Value Lifting Network (VLNet) hereafter. More details related to VLNet are described in Appendix C.

3.2 Strict Fourier-Taylor Expansion

By definition, a finite Fourier series with N harmonics is

$$s_N(r) = a_0 + \sum_{n=1}^{N}\left[a_n \cos\left(\frac{2\pi n r}{P}\right) + b_n \sin\left(\frac{2\pi n r}{P}\right)\right], \tag{2}$$

where P denotes the period and a_n, b_n are Fourier coefficients. The Maclaurin expansions are

$$\cos z = \sum_{i=0}^{\infty}(-1)^i \frac{z^{2i}}{(2i)!}, \qquad \sin z = \sum_{i=0}^{\infty}(-1)^i \frac{z^{2i+1}}{(2i+1)!}. \tag{3}$$

Truncating the two series at order index I gives

$$\tilde{s}_{N,I}(r) = a_0 + \sum_{n=1}^{N}\sum_{i=0}^{I} a_n d_i^c(P) n^{2i} r^{2i} + \sum_{n=1}^{N}\sum_{i=0}^{I} b_n d_i^s(P) n^{2i+1} r^{2i+1}, \tag{4}$$

where

$$d_i^c(P) = (-1)^i \frac{(2\pi/P)^{2i}}{(2i)!}, \qquad d_i^s(P) = (-1)^i \frac{(2\pi/P)^{2i+1}}{(2i+1)!}. \tag{5}$$

Equation 4 is the strict truncated Fourier-Taylor expression. It preserves the sign, factorial, period scaling, odd/even parity, and the coupling caused by sharing a_n or b_n across Taylor orders. The full Fourier-Taylor derivation is detailed in Appendix B.

3.3 Relaxed Fourier-Taylor Lifting

For a neural layer, define the unscaled high-order dictionary

$$\mathbf{\Phi}_R(r) = \mathrm{Concat}_{n=1}^{N} \mathrm{Concat}_{i=0}^{I} \left[n^{2i} r^{2i}, n^{2i+1} r^{2i+1} \right]. \tag{6}$$

The deterministic analytical factors in Eq. 5 can be collected in a diagonal matrix D_P, such that the corresponding analytically scaled dictionary is

$$\mathbf{\Phi}_{FT}(r) = D_P \mathbf{\Phi}_R(r). \tag{7}$$

Proposition 1 (Diagonal scaling under learnable aggregation)**.** *Assume every retained entry of D_P is non-zero and the dictionary is followed by an unrestricted learnable linear map. Then*

$$\{W D_P \mathbf{\Phi}_R(r) : W\} = \{\Theta \mathbf{\Phi}_R(r) : \Theta\}. \tag{8}$$

The proof is direct by setting $\Theta = W D_P$ and is provided in Appendix A. Proposition 1 primarily states that deterministic non-zero basis scaling can be absorbed once every basis term is given a free aggregation coefficient.

The second change is more important. In Eq. 4, a_n and b_n couple all even or odd orders belonging to the same harmonic. The implemented neural layer releases this coupling and learns an effective matrix for each harmonic-order pair:

$$\phi_R(\mathbf{h}) = \mathbf{b}_0 + \sum_{n=1}^{N} \sum_{i=0}^{I} \left[\Theta_{n,i}^c n^{2i} \mathbf{h}^{\odot 2i} + \Theta_{n,i}^s n^{2i+1} \mathbf{h}^{\odot (2i+1)} \right]. \tag{9}$$

We call this form Relaxed-FT or FT-Lift. The notation $\Theta_{n,i}$ is intentionally order-dependent. It replaces the earlier notation in which a single relaxed coefficient was written for all i, which did not accurately describe the implemented degrees of freedom.

3.4 What the Fourier-Taylor Derivation Does Not Claim

After full relaxation, the harmonic branches can be collapsed order by order:

$$\sum_{n=1}^{N} \Theta_{n,k} n^k r^k = \left(\sum_{n=1}^{N} \Theta_{n,k} n^k \right) r^k = \bar{\Theta}_k r^k. \tag{10}$$

Proposition 2 (Function-family relation)**.** *Under the fully relaxed formulation in Eq. 9, the scalar function family is contained in the polynomial family up to the maximum retained degree. Multiple harmonic branches provide a factorization of the effective polynomial coefficients rather than additional exact Fourier frequencies.*

The harmonic index therefore becomes a factorization index rather than an additional sinusoidal frequency. This observation is central to the interpretation of our experiments: if Relaxed-FT and parameter-matched polynomial lifting perform similarly, the evidence supports trainable high-order value lifting, not a uniquely Fourier-specific function class. The proof is given in Appendix A.

4 Experimental Study

4.1 Datasets, Protocol, and Evaluation

For comparison with 13 general time-series frameworks, we follow Mu et al. (2025); Zhou et al. (2026) and evaluate on 10 representative datasets. These datasets are also used for the controlled

Table 1: Performance comparison with the recent advanced general time-series analysis frameworks on 10 UEA datasets.

	LSTNet	LSSL	FED-former	Flow-former	SCINet	Dlinear	Patch-TST	MICN	Times-Net	Cross-former	Modern-TCN	Time-Mixer++	TVNet	VLNet
	SIGIR'18	*ICLR'22*	*ICLR'22*	*ICLR'22*	*NeurIPS'22*	*AAAI'23*	*ICLR'23*	*ICLR'23*	*ICLR'23*	*ICLR'23*	*ICLR'24*	*ICLR'25*	*ICLR'25*	
EC	0.399	0.311	0.312	0.338	0.344	0.362	0.328	0.353	0.357	0.380	0.363	0.399	0.356	0.357
FD	0.657	0.667	0.660	0.676	0.689	0.680	0.683	0.652	0.686	0.687	0.708	0.718	0.712	0.692
HW	0.258	0.246	0.280	0.338	0.236	0.270	0.296	0.255	0.321	0.288	0.306	0.265	0.327	0.629
HB	0.771	0.727	0.737	0.776	0.775	0.751	0.749	0.747	0.780	0.776	0.772	0.791	0.781	0.805
JV	0.981	0.984	0.984	0.989	0.960	0.962	0.975	0.946	0.984	0.991	0.988	0.979	0.989	0.997
PEMS	0.867	0.861	0.809	0.860	0.838	0.751	0.893	0.855	0.896	0.859	0.891	0.910	0.889	0.896
SRS1	0.840	0.908	0.887	0.925	0.925	0.873	0.907	0.860	0.918	0.921	0.934	0.931	0.937	0.874
SRS2	0.528	0.522	0.544	0.561	0.572	0.505	0.578	0.536	0.572	0.583	0.603	0.656	0.605	0.572
SAD	1.000	1.000	1.000	0.988	0.981	0.814	0.983	0.971	0.990	0.979	0.987	0.998	0.994	0.999
UW	0.878	0.859	0.853	0.866	0.851	0.821	0.858	0.828	0.853	0.853	0.867	0.882	0.866	0.925
Avg. ACC	0.718	0.709	0.707	0.732	0.717	0.679	0.725	0.7	0.736	0.732	0.742	0.753	0.746	0.775
#Top-1	2	1	1	0	0	0	0	0	0	0	0	4	1	4
#Top-2	2	1	1	1	0	0	0	0	1	2	1	6	3	6
#Top-3	3	1	1	2	0	0	1	0	1	2	4	8	6	6
Avg. Rank	7.2	8.6	8.6	5.9	8.2	9.9	7.3	10.7	5.2	5.8	4.2	3.1	3.4	3.3

Table 2: Performance comparison with the recent advanced MTSC-dedicated models on 26 UEA datasets.

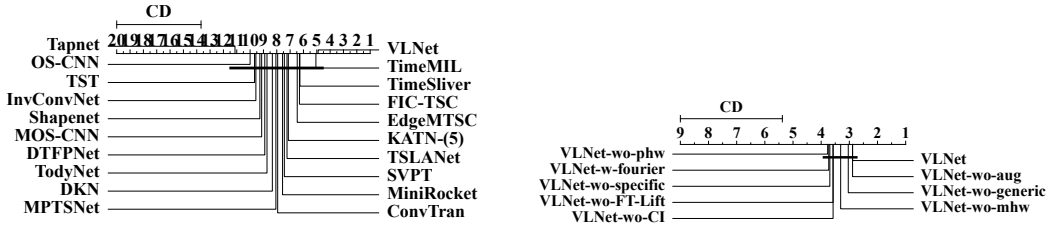
	MiniRocket	TodyNet	EdgeMTSC	InvConvNet	TimeSliver	TST	SVPT	ConvTran	TimeMIL	DKN	KATN-(5)	TSLANet	MPTSNet	DTFPNet	FIC-TSC	VLNet
	KDD'21	*Info.'24*	*AAAI'26*	*ICML'25*	*ICLR'26*	*KDD'21*	*AAAI'23*	*DMKD'24*	*ICML'24*	*TSMCS'24*	*TBD'25*	*ICML'24*	*AAAI'25*	*PR'26*	*ICML'25*	
AWR	0.992	0.987	0.983	0.990	0.993	0.983	0.993	0.983	0.990	0.993	0.993	0.990	0.977	0.991	0.993	0.993
CT	0.993	N/A	0.991	N/A	N/A	0.000	0.990	0.992	N/A	0.986	0.986	N/A	N/A	N/A	N/A	0.998
CR	0.986	1.000	1.000	0.986	0.986	0.958	1.000	1.000	1.000	0.951	1.000	0.986	0.944	1.000	1.000	1.000
ER	0.981	0.915	0.893	N/A	0.841	0.933	0.937	0.963	0.956	0.933	0.968	N/A	0.944	N/A	N/A	0.919
EW	0.962	0.840	0.542	N/A	0.841	N/A	0.925	0.593	0.823	0.628	0.733	N/A	N/A	N/A	N/A	0.748
EP	1.000	0.971	0.993	0.959	0.985	0.920	0.986	0.986	1.000	0.979	0.913	0.986	0.971	1.000	0.989	0.993
EC	0.468	0.350	0.354	0.260	0.437	0.337	0.331	0.361	0.407	0.372	0.411	0.304	0.433	0.376	0.392	0.357
FD	0.620	0.627	0.679	0.647	0.700	0.681	0.512	0.672	0.698	0.631	0.664	0.668	0.698	0.687	0.684	0.692
FM	0.550	0.676	0.680	0.563	0.650	0.776	0.600	0.560	0.670	0.600	0.610	0.610	0.640	0.570	0.650	0.660
HMD	0.392	0.649	0.608	0.410	0.460	0.608	0.392	0.405	0.487	0.662	0.688	0.527	0.635	N/A	0.493	0.446
HW	0.507	0.436	0.386	0.531	0.571	0.305	0.433	0.375	0.482	0.231	0.320	0.579	0.344	N/A	0.616	0.629
HB	0.771	0.756	0.771	0.774	0.805	0.712	0.790	0.375	0.815	0.765	0.762	0.776	0.756	0.770	0.810	0.805
IW	0.595	N/A	0.588	N/A	N/A	0.684	0.184	0.713	N/A	0.362	0.567	0.100	N/A	0.160	0.711	0.681
JV	0.989	N/A	0.995	0.977	N/A	0.994	0.978	0.989	N/A	0.930	0.968	0.992	N/A	0.959	0.991	0.997
LSST	0.643	0.615	0.412	0.550	0.685	0.381	0.666	0.616	0.690	0.347	0.407	0.663	0.604	0.590	0.653	0.614
LIB	0.922	0.850	0.928	0.887	0.833	0.844	0.883	0.928	0.972	0.900	0.839	0.928	0.872	0.867	0.794	0.928
MI	0.550	0.640	0.620	0.497	0.619	N/A	0.650	0.560	0.720	0.620	0.630	0.620	0.650	0.650	0.650	0.630
NATO	0.928	0.972	0.989	0.957	0.989	0.900	0.906	0.944	0.994	0.872	0.889	0.956	0.944	0.961	0.989	0.983
PEMS	0.522	0.780	0.884	0.804	0.948	0.919	0.867	0.828	0.931	0.930	0.953	0.838	0.942	0.821	0.792	0.896
PD	N/A	0.987	0.989	0.988	0.981	0.974	0.983	0.987	0.600	0.948	0.982	0.989	0.989	0.987	0.976	0.991
PS	0.292	0.309	0.256	0.298	0.290	0.088	0.176	0.306	0.311	0.525	0.466	0.178	0.144	0.298	0.313	0.287
RS	0.868	0.803	0.914	0.877	0.921	0.829	0.842	0.862	0.908	0.879	0.875	0.908	0.875	0.861	0.898	0.934
SRS1	0.925	0.898	0.898	0.861	0.900	0.925	0.884	0.918	0.898	0.913	0.908	0.918	0.928	0.910	0.901	0.874
SRS2	0.522	0.550	0.594	0.544	0.620	0.589	0.600	0.583	0.639	0.600	0.600	0.617	0.572	0.550	0.594	0.572
SAD	0.620	N/A	0.998	0.995	N/A	0.993	0.986	N/A	N/A	0.963	0.983	0.999	N/A	0.997	0.999	0.999
UW	0.938	0.850	0.866	0.929	0.910	0.903	0.941	0.891	0.900	0.893	0.897	0.913	0.881	0.862	0.902	0.925
Avg. ACC	0.741	0.748	0.762	0.74	0.771	0.718	0.748	0.736	0.768	0.747	0.77	0.741	0.75	0.756	0.773	0.79
#Top-1	4	1	1	0	2	1	3	2	8	2	4	1	1	2	3	8
#Top-2	9	1	8	1	9	2	5	3	9	3	6	4	4	3	8	10
#Top-3	10	4	9	3	11	5	6	6	9	3	6	7	6	5	10	13
Avg. Rank	6.5	8.231	6.115	8.308	5.654	8.038	6.923	7.115	5.0	7.423	6.692	6.154	7.385	7.962	5.538	4.5
#Wins	18	14	18	20	10	19	17	17	5	18	16	20	18	20	10	-
#Draws	0	1	3	0	2	0	2	2	1	1	3	2	1	1	3	-
#Loses	8	7	5	2	10	7	7	7	16	7	7	4	7	5	10	-
p-value	0.148	0.038	0.023	0.0	0.822	0.006	0.014	0.02	0.179	0.02	0.091	0.109	0.172	0.022	0.778	-

comparison of different lifting strategies. We further employ 26 MTSC datasets from the UEA archive (Bagnall et al., 2018) to compare VLNet with 19 advanced MTSC-dedicated models, assess sensitivity to key hyperparameters, and evaluate FT-Lift when integrated into conventional MTSC architectures. Additional details on the UEA datasets are provided in Appendix D.1. Dataset names are abbreviated following Xiao et al. (2024b). To isolate the effect of lifting order, we first conduct controlled experiments on a synthetic dataset, as detailed in Appendix F.

In the same-protocol comparison, variants share the same temporal backbone, optimizer, training budget, and model-selection rule. Normalization statistics are estimated from training data. During training, we use $I = 1$, $N = 3$, $P = 1$, and set the depthwise convolution kernel size κ to 47 by default. More details related to metrics, hyperparameter settings, and optimization can be found in Appendix D.2 and D.3. All experiments are conducted using Python 3.8.20 with PyTorch 2.3.0 and CUDA 12.6 support on an NVIDIA GeForce RTX 3090 GPU. The source code and experimental configurations are available at `https://anonymous.4open.science/r/FT-Lift-796E/`.

4.2 Broad MTSC Performance

We adopted the performance of general time-series frameworks and MTSC-dedicated models from their studies for comparison. More details about the baseline models can be found in Appendix D.4.



(a) Average ranks of the 26-dataset setting.

(b) Average ranks of the structural ablations.

Figure 1: Broad MTSC Performance. (1) Average accuracy ranks of our system and 19 MTSC-dedicated baselines. The CD score is 6.321. (2) Average accuracy ranks of our system and its variants, showing the structural advancement. The CD score is 3.622.

Table 1 presents the performance comparison with the recent 13 general time-series frameworks on 10 UEA datasets. VLNet with FT-Lift achieves an average accuracy of 77.5%. Compared with TimeMixer++, our model has competitive ranking results, but with a significant accuracy improvement of 2.2%.

Due to page-width limitations, we hide the results of TapNet (0.669) (Zhang et al., 2020), ShapeNet (0.704) (Li et al., 2021), OS-CNN (0.696) (Tang et al., 2020), and MOS-CNN (0.719) (Tang et al., 2020) in Table 2. The 'N/A' indicates unreported performance of the corresponding methods. We also rank their performance using a critical difference (CD) diagram in Figure 1a. The performance comparison demonstrates the superiority of VLNet across 6 different kinds of models. It achieves strong results, with the best average accuracy (0.79) and the best ranking results (Top-1/2/3: 8/10/13). By the definition of the Wilcoxon Test p-value, the MTSC system is statistically significant except for MiniRocket, TimeSliver, TimeMIL, KATN-(5), TSLANet, MPTSNet, and FIC-TSC ($p \geq 0.05$). A separate repeated-run evaluation of the existing implementation over ten random seeds reports an average ACC of 0.787 across the 26 datasets, together with average balanced accuracy 0.767 and Macro-F1 0.769. Full per-dataset values are provided in Appendix G.

4.3 Does the High-Order Pathway Contribute to VLNet?

The structural ablation of the existing architecture provides the first piece of evidence. Figure 1b demonstrates their average ranks and CD score. Specifically, removing the polynomial lifting block (VLNet-wo-FT-Lift) lowers the average by roughly 0.8%, replacing it with the original sine-cosine Fourier series (VLNet-w-fourier) lowers the average by roughly 1.3%, and removing the polynomial highway lowers the average by roughly 1.8%. Other ablations are detailed in Appendix E. These ablations consistently indicate that the value-lifting pathway contributes to the predictive performance of the complete architecture. The sine-cosine replacement also suggests that the useful component is not simply the presence of a Fourier transform; the trainable high-order pathway is more compatible with VLNet.

4.4 Does FT-Lift Improve the Other Architectures?

To evaluate whether FT-Lift transfers beyond VLNet, we integrate it into two conventional CNN architectures, *i.e.*, ResNet and FCN, introduced by Wang et al. (2017). For both models, FT-Lift is inserted between the backbone and the pooling layer, and the resulting variants are evaluated on all 26 UEA datasets. The results demonstrate that the vanilla FCN and ResNet achieve mean accuracies of 0.677 and 0.742, respectively. When FT-Lift is inserted between the backbone and the global average pooling layer, ResNet yields a 2.83% improvement in mean accuracy, while FCN achieves a 0.3% improvement. These results show that FT-Lift can transfer beyond VLNet, and the magnitude of improvement is architecture-dependent.

4.5 What Form of Value Lifting Is Supported by Matched Controls?

To separate the lifting mechanism from the rest of the architecture, we compare six variants using the same backbone and training protocol: No Lift, Exact Fourier, Strict-FT, Direct Polynomial, a parameter-matched polynomial control, and Relaxed-FT (FT-Lift). Direct Polynomial uses one

Table 3: Value lifting form comparison on 10 representative UEA datasets under the same protocol. "Std." is the dispersion across dataset-level mean accuracies. W/T/L is reported from the perspective of Relaxed-FT. Wilcoxon p-values are uncorrected pairwise values; Holm-adjusted values for the prespecified comparisons are discussed in the text.

Method	Mean ACC	Std.	Avg. Rank	Relaxed-FT W/T/L	p vs. Relaxed-FT
Relaxed-FT	**0.7807**	0.2081	**2.30**	–	–
No Lift	0.7786	0.2090	3.10	6/0/4	0.3750
Param.-Matched Polynomial	0.7773	0.2046	3.90	8/0/2	0.2754
Exact Fourier	0.7773	0.2105	3.50	6/2/2	0.1614
Strict-FT	0.7761	0.2077	3.90	8/0/2	0.0488
Direct Polynomial	0.7754	0.2070	4.30	8/0/2	0.1309

Table 4: Test accuracy on controlled value-order tasks over four repeated runs. Values are mean±standard deviation.

Method	$p = 1$	$p = 2$	$p = 4$	$p = 8$
Direct Polynomial	0.8252 ± 0.0155	0.7959 ± 0.0160	0.6709 ± 0.0148	0.5425 ± 0.0131
Param.-Matched Polynomial	$\mathbf{0.8350} \pm 0.0091$	0.7954 ± 0.0262	0.6587 ± 0.0070	$\mathbf{0.5557} \pm 0.0241$
Strict-FT	0.8311 ± 0.0213	0.7940 ± 0.0330	$\mathbf{0.6738} \pm 0.0214$	0.5474 ± 0.0176
Relaxed-FT	0.8306 ± 0.0056	$\mathbf{0.8008} \pm 0.0241$	0.6543 ± 0.0395	0.5523 ± 0.0130

transformation per polynomial order, whereas the parameter-matched control retains the same number of harmonic-order parameter blocks as Relaxed-FT but removes the n^k factorization.

Three observations matter for the paper's claim: (1) Relaxed-FT has the highest mean accuracy and best average rank in this matched study; (2) it improves on Strict-FT from 0.7761 to 0.7807 and has higher dataset-level mean accuracy on 8 out of 10 datasets (The uncorrected paired Wilcoxon comparison between Relaxed-FT and Strict-FT gives $p = 0.0488$, but this difference does not remain significant after Holm correction ($p_{\text{adj}} \approx 0.244$). We therefore interpret the result as directional evidence that relaxation is beneficial); and (3) the gap to No Lift and parameter-matched polynomial lifting is small and statistically inconclusive. This does not support the interpretation that the Relaxed-FT owes its usefulness to a uniquely larger scalar function family. More conservatively, trainable high-order value transformation can be useful, while preserving the strict analytical coefficient constraints of the Fourier-Taylor expansion appears restrictive for the tasks considered here.

4.6 Does More Value Order Automatically Help?

A common intuition behind explicit high-order lifting is that increasing the available polynomial order should make increasingly high-order class evidence easier to represent. We first test this intuition using controlled binary tasks in which the class-defining statistic is constructed from normalized probabilists' Hermite polynomials of target order $p \in \{1, 2, 4, 8\}$ (Weisstein, 2002). The input distribution and sample-generation procedure are otherwise fixed across target orders. More details about the synthetic dataset generation are provided in Appendix F. We compare four high-order parameterizations under the same training protocol. The result is negative but informative, as shown in Figure 3 (a). Accuracy decreases from roughly 0.83 at $p = 1$ to roughly 0.54–0.56 at $p = 8$ for every parameterization, while the best-performing method varies across target orders. We further tuned $I \in \{1, 2, 3, 4\}$ across all 26 datasets, with results reported in Table 8. The aggregate performance is also non-monotonic with respect to I; $I = 1$ and $I = 3$ both achieve a mean accuracy of 0.790, whereas $I = 2$ and $I = 4$ obtain 0.785 and 0.776, respectively. More importantly, the best order varies across individual datasets. Taken together, these results indicate that the useful lifting order is task-dependent rather than monotonically beneficial; higher orders do not systematically ease high-order targets, yet the best practical performance does not always occur at the lowest order. We therefore view value lifting as a trainable nonlinear expansion whose effective order should be selected according to the task, rather than maximized by default.

4.7 How Sensitive is VLNet to the Key Hyperparameters?

We further examine the factorization width N inherited from the Fourier-Taylor construction and the depthwise-convolution kernel size k. Across the 26 UEA datasets, mean accuracy peaks at 0.792 with $N = 5$ and drops to 0.785 at $N = 6$; for κ, it peaks at 0.790 with $\kappa = 47$ and decreases to

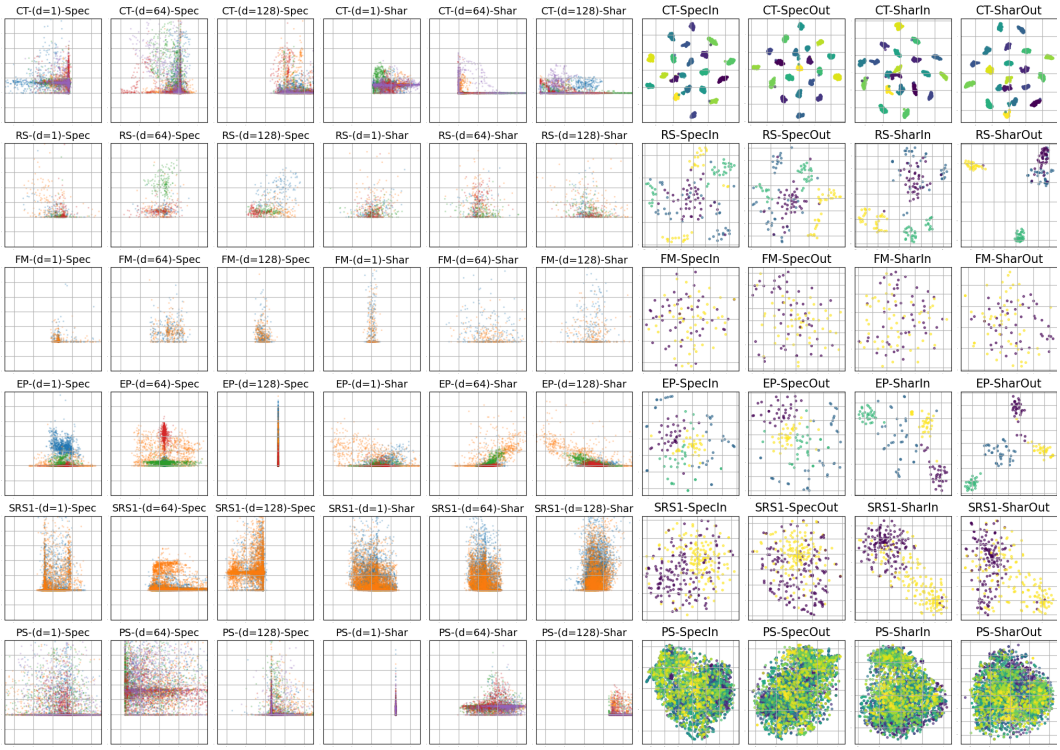


Figure 2: Analysis of intermediate values and FT-Lifted values on the test datasets. Each row corresponds to a dataset. The first six plots illustrate the mappings from intermediate values to the corresponding lifted features along the d-th latent variable, using one randomly selected sample from each category. The last four plots are the t-SNE visualizations of the input/output of the FT-Lift. Spec denotes the channel-specific FT-Lift branch, with SpecIn and SpecOut indicating the input and output of the Spec branch, respectively; Shar denotes the channel-shared FT-Lift branch, with SharIn/SharOut indicating the input/output of the Shar branch.

0.784 at $\kappa = 67$. Dataset-level optima also vary across both hyperparameters. Together with the order analysis in Section 4.6, these results show that lifting order, factorization width, and temporal receptive field are task-dependent rather than monotonically improved by increasing model capacity.

5 Discussion

What the Existing Results Support? The evidence supports value lifting as a useful but task-dependent MTSC component. VLNet benefits from its lifting pathway, and FT-Lift also transfers to conventional CNN backbones. However, its matched advantage over No Lift and parameter-matched polynomial lifting is small, while both controlled and real-data experiments show that higher lifting order is not uniformly beneficial. We therefore frame the central contribution as the introduction and study of an explicit trainable value-transformation pathway for MTSC. Beyond this work, future work will investigate whether stronger Fourier-specific advantages emerge under other settings and how far useful lifting order can be extended while preserving trainability and generalization.

Why Keep the Fourier-Taylor Derivation? Given the polynomial function-family equivalence induced by full relaxation, the role of the Fourier-Taylor derivation is constructive rather than to define a distinct function class. Starting from the sine and cosine expansions provides a systematic organization of odd- and even-order terms, makes the analytical coefficient tying explicit, and establishes a transparent path from the strict analytical series to the trainable layer used in practice. The derivation also clarifies which structural constraints originate from the Fourier-Taylor formulation and which are removed or absorbed under learnable aggregation. However, it does *not* imply that the fully relaxed layer preserves exact periodic semantics.

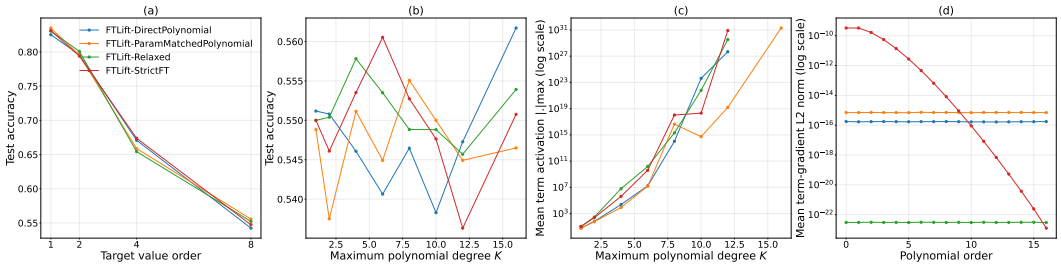


Figure 3: Controlled value-order and numerical-conditioning results. (a) Test accuracy as the target value order increases. (b) Test accuracy vs. maximum polynomial degree K. (c) Mean maximum term activation vs. polynomial degree. (d) Mean term-gradient L_2 norm across polynomial orders. The vertical axes in (c) and (d) are logarithmic.

How Value Lifting Reshapes Intermediate Values? Figure 2 provides a qualitative view of how FT-Lift transforms the intermediate representations learned by the temporal branches on selected datasets for which VLNet shows different performance. The first six plots of each row visualize the input-output mappings of selected latent variables in the channel-specific and channel-shared FT-Lift branches. Rather than acting as a simple rescaling, FT-Lift induces visibly nonlinear, order-dependent transformations of the intermediate values, with the geometry of the transformed representation varying across datasets and latent dimensions. This observation is consistent with the role of FT-Lift as an explicit value-domain transformation applied after temporal feature extraction. Besides, the four t-SNE plots on the right provide a complementary representation-level view. On datasets where VLNet achieves relatively strong performance, such as CT, RS, and EP, the lifted representations often exhibit more compact or more clearly organized class-dependent structures than their corresponding inputs, at least in one of the two branches. In contrast, on datasets where VLNet shows more modest performance, such as FM and PS, substantial class overlap remains after lifting, and the effect also differs between the channel-specific and channel-shared branches. These examples therefore suggest that FT-Lift reshapes intermediate feature geometry in a dataset- and branch-dependent manner. We interpret the visualization as qualitative evidence of representation transformation, rather than as evidence that value lifting universally improves class separability.

Secondary Numerical Observation. Although optimization is not the main focus of this paper, the existing diagnostics from the controlled experiments explain why very high polynomial degree should be treated cautiously. In the degree K sweep using the same protocol, raw monomial activations grow by many orders of magnitude (Figure 3 (c)), and the strict analytical parameterization shows a highly nonuniform order-wise raw-gradient profile (Figure 3 (d)). According to Figure 3 (b), we treat them as implementation guidance since these diagnostics do not translate into a consistent predictive ordering

Limitations. (1) The lifting comparison currently covers 10 UEA datasets rather than all 26. The broad 26-dataset table provides coverage but combines literature-reported baselines trained under heterogeneous protocols. (2) FT-Lift is element-wise and does not directly enumerate cross-variable monomials such as $x_i x_j$; cross-variable effects arise through the surrounding mixing operations. (3) Raw monomial features become numerically fragile at high degree, so the present evidence supports lifting with lower orders. (4) Because the parameter-matched polynomial performs similarly to Relaxed-FT, future work will further investigate the high-order value-lifting principle and the particular factorization.

6 Conclusion

We study explicit trainable value lifting for MTSC through FT-Lift, a high-order transformation of intermediate features. VLNet achieves 0.79 average accuracy on 26 UEA datasets, while FT-Lift also improves ResNet and FCN. Relaxed-FT performs best in the matched comparison, but its advantage over No Lift and parameter-matched polynomial lifting is small. Further analyses show that lifting order, factorization width, and kernel size are task-dependent rather than monotonically beneficial. Overall, we interpret FT-Lift as a useful value-transformation component, not a uniquely superior Fourier-specific function family.

AI USE STATEMENT

The authors note that generative AI tools (specifically ChatGPT by OpenAI) were used for detecting typographical or grammatical errors only.

REFERENCES

Ibram Abdelmalak, Kiran Madhusudhanan, Jungmin Choi, Christian Kloetergens, Vijaya Krishna Yalavarit, Maximilian Stubbemann, and Lars Schmidt-Thieme. Channel dependence, limited lookback windows, and the simplicity of datasets: How biased is time series forecasting? *arXiv preprint arXiv:2502.09683*, 2025.

Forest Agostinelli, Matthew Hoffman, Peter Sadowski, and Pierre Baldi. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830*, 2014.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Binghui Chen, Weihong Deng, and Jiani Hu. Mixed high-order attention network for person re-identification. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019.

Xiwen Chen, Peijie Qiu, Wenhui Zhu, Huayu Li, Hao Wang, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. Timemil: Advancing multivariate time series classification via a time-aware multiple instance learning. 2024.

Xiwen Chen, Wenhui Zhu, Peijie Qiu, Hao Wang, Huayu Li, Zihan Li, Yalin Wang, Aristeidis Sotiras, and Abolfazl Razi. Fic-tsc: Learning time series classification with fisher information constraint. *arXiv preprint arXiv:2505.06114*, 2025.

Yijiang Chen, Xiangdong Zhou, Zhen Xing, Zhidan Liu, and Minyang Xu. Cass: A channel-aware self-supervised representation learning framework for multivariate time series classification. In *International Conference on Database Systems for Advanced Applications*, pp. 375–390. Springer, 2022.

Ce Chi, Xing Wang, Kexin Yang, Zhiyan Song, Di Jin, Lin Zhu, Chao Deng, and Junlan Feng. Injecttst: a transformer method of injecting global information into independent channels for long time series forecasting. *arXiv preprint arXiv:2403.02814*, 2024.

Grigorios G Chrysos, Stylianos Moschoglou, Giorgos Bouritsas, Yannis Panagakis, Jiankang Deng, and Stefanos Zafeiriou. P–nets: Deep polynomial neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

Angus Dempster, Daniel F Schmidt, and Geoffrey I Webb. Minirocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 248–257, 2021.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, and Xiaoli Li. Tslanet: Rethinking transformers for time series representation learning. *arXiv preprint arXiv:2404.08472*, 2024.

Navid Mohammadi Foumani, Chang Wei Tan, Geoffrey I Webb, and Mahsa Salehi. Improving position encoding of transformers for multivariate time series classification. *Data Mining and Knowledge Discovery*, 38(1):22–48, 2024.

Thibaut Germain, Chrysoula Kosma, and Laurent Oudre. Time series representations with hard-coded invariances. In *Forty-second International Conference on Machine Learning*, 2025.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate lstm-fcns for time series classification. *Neural networks*, 116:237–245, 2019.

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.

Xuan-May Le, Ling Luo, Uwe Aickelin, and Minh-Tuan Tran. Shapeformer: Shapelet transformer for multivariate time series classification. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1484–1494, 2024.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. In *Proceedings of the 2022 Conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, pp. 4296–4313, 2022.

Chenghan Li, Mingchen Li, and Ruisheng Diao. Tvnet: A novel time series analysis method based on dynamic convolution and 3d-variation. *arXiv preprint arXiv:2503.07674*, 2025.

Guozhong Li, Byron Choi, Jianliang Xu, Sourav S Bhowmick, Kwok-Pan Chun, and Grace Lai-Hung Wong. Shapenet: A shapelet-neural network approach for multivariate time series classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8375–8383, 2021.

Hangtian Li and Xiaolei Xie. Dims: Channel-dependent and seasonal-trend independent transformer using multi-stage training for time series forecasting.

Wei Li, Xiangxu Meng, Chuhao Chen, and Jianing Chen. Mlinear: Rethink the linear model for time-series forecasting. *arXiv preprint arXiv:2305.04800*, 2023.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI open*, 3: 111–132, 2022.

Huaiyuan Liu, Donghua Yang, Xianzhang Liu, Xinglei Chen, Zhiyu Liang, Hongzhi Wang, Yong Cui, and Jun Gu. Todynet: temporal dynamic graph neural network for multivariate time series classification. *Information Sciences*, 677:120914, 2024.

Minghao Liu, Shengqi Ren, Siyuan Ma, Jiahui Jiao, Yizhou Chen, Zhiguang Wang, and Wei Song. Gated transformer networks for multivariate time series classification. *arXiv preprint arXiv:2103.14438*, 2021.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

Shikang Liu, Chuyang Wei, Xiren Zhou, and Huanhuan Chen. Spectral-aware reservoir computing for fast and accurate time series classification. In *Forty-second International Conference on Machine Learning*, 2025a.

Shikang Liu, Ziyu Tang, Xiren Zhou, and Huanhuan Chen. Svgl: Scale-variable graph learning in model space for multivariate time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23801–23809, 2026.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. In *International conference on learning representations*, 2025b.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*, 2024.

Navid Mohammadi Foumani, Lynn Miller, Chang Wei Tan, Geoffrey I Webb, Germain Forestier, and Mahsa Salehi. Deep learning for time series classification and extrinsic regression: A current survey. *ACM Computing Surveys*, 56(9):1–45, 2024.

Yang Mu, Muhammad Shahzad, and Xiao Xiang Zhu. Mptsnet: Integrating multiscale periodic local patterns and global dependencies for multivariate time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19572–19580, 2025.

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.

Akash Pandey, Payal Mohapatra, Wei Chen, Qi Zhu, and Sinan Keten. Timesliver: Symbolic-linear decomposition for explainable time series classification. *arXiv preprint arXiv:2601.21289*, 2026.

Yuan Peiwen and Zhu Changsheng. Is channel independent strategy optimal for time series forecasting? *arXiv preprint arXiv:2310.17658*, 2023.

Patrick Schäfer and Ulf Leser. Multivariate time series classification with weasel+ muse. *arXiv preprint arXiv:1711.11343*, 2017.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Wensi Tang, Guodong Long, Lu Liu, Tianyi Zhou, Michael Blumenstein, and Jing Jiang. Omni-scale cnns: a simple and effective kernel size configuration for time series classification. *arXiv preprint arXiv:2002.10061*, 2020.

Georgi P Tolstov. *Fourier series*. Courier Corporation, 2012.

Anthony Tompkins and Fabio Ramos. Fourier feature approximations for periodic kernels in time-series modelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Haoxin Wang, Yipeng Mo, Kunlan Xiang, Nan Yin, Honghe Dai, Bixiong Li, and Songhai Fan. Csformer: Combining channel independence and mixing for robust multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21090–21098, 2025a.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*, 2023.

Jie Wang, Zhongguang Zheng, Chaoliang Zhong, and Jun Sun. Channel-independent refiner for multivariate time series forecasting. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 5304–5308, 2025b.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, volume 2025, pp. 16980–17016, 2025c.

Yucheng Wang, Yuecong Xu, Jianfei Yang, Min Wu, Xiaoli Li, Lihua Xie, and Zhenghua Chen. Graph-aware contrasting for multivariate time-series classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 15725–15734, 2024.

Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pp. 1578–1585. IEEE, 2017.

Eric W Weisstein. Hermite polynomial. *https://mathworld. wolfram. com/*, 2002.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022a.

Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Flowformer: Linearizing transformers with conservation flows. *arXiv preprint arXiv:2202.06258*, 2022b.

Tingxiong Xiao, Weihang Zhang, Yuxiao Cheng, and Jinli Suo. Hope: high-order polynomial expansion of black-box neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024a.

Zhiwen Xiao, Huanlai Xing, Rong Qu, Li Feng, Shouxi Luo, Penglin Dai, Bowen Zhao, and Yuanshun Dai. Densely knowledge-aware network for multivariate time series classification. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(4):2192–2204, 2024b.

Zhiwen Xiao, Huanlai Xing, Rong Qu, Hui Li, Huagang Tong, Shouxi Luo, Jing Song, Li Feng, and Qian Wan. Knowledge aggregation transformer network for multivariate time series classification. *IEEE Transactions on Big Data*, 2025.

Chenhui Xu, Fuxun Yu, Maoliang Li, Zihao Zheng, Zirui Xu, Jinjun Xiong, and Xiang Chen. Infinite-dimensional feature interaction. *Advances in Neural Information Processing Systems*, 2024.

Wuqing Yu, Bowen Fu, Zhan Li, Jian Zhou, Junhao Wang, and Jiacai Zhang. Dtfpnet: Temporal and frequency dynamic graph neural network for time series classification. *Pattern Recognition*, 2025.

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.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2114–2124, 2021.

Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6845–6852, 2020.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Lifan Zhao and Yanyan Shen. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. *arXiv preprint arXiv:2401.17548*, 2024.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

Xueyi Zhou, Zhenyu Li, and Dong-Kyu Chae. Edgemtsc: A lightweight large-kernel convnet for multivariate time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Rundong Zuo, Guozhong Li, Byron Choi, Sourav S Bhowmick, Daphne Ngar-yin Mah, and Grace LH Wong. Svp-t: a shape-level variable-position transformer for multivariate time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11497–11505, 2023.

A PROOFS

A.1 PROOF OF PROPOSITION 1

Let $D_P \in \mathbb{R}^{q\times q}$ be diagonal and assume every retained diagonal entry is non-zero. For any $W \in \mathbb{R}^{o\times q}$, define $\Theta = WD_P$. Then

$$WD_P\mathbf{\Phi}_R(r) = \Theta\mathbf{\Phi}_R(r). \tag{11}$$

Conversely, for any Θ, define $W = \Theta D_P^{-1}$. Therefore, both parameterizations generate the same set of outputs after an unrestricted linear map. This argument only concerns deterministic basis scaling. The Fourier coefficient tying in Eq. 4 is a separate structural constraint and is explicitly released in Eq. 9.

A.2 PROOF OF PROPOSITION 2

For each degree k, Eq. 9 contains a sum of terms of the form $\Theta_{n,k}n^k\mathbf{h}^{\odot k}$. Since n^k is a scalar constant,

$$\sum_{n=1}^{N}\Theta_{n,k}n^k\mathbf{h}^{\odot k} = \bar{\Theta}_k\mathbf{h}^{\odot k}, \quad \bar{\Theta}_k := \sum_{n=1}^{N}\Theta_{n,k}n^k. \tag{12}$$

Summing over k therefore yields a polynomial map up to the largest retained degree. Multiple harmonic branches provide an overparameterized factorization of $\bar{\Theta}_k$ but do not introduce a new scalar basis function after complete relaxation and summation.

B FULL FOURIER-TAYLOR DERIVATION

Substituting Eq. 3 into Eq. 2 gives

$$\tilde{s}_{N,I}(r) = a_0 + \sum_{n=1}^{N}\sum_{i=0}^{I}\left[(-1)^i a_n\frac{(2\pi n)^{2i}}{(2i)!P^{2i}}r^{2i} + (-1)^i b_n\frac{(2\pi n)^{2i+1}}{(2i+1)!P^{2i+1}}r^{2i+1}\right]. \tag{13}$$

For a vector input $\mathbf{h}^{(d)}$, scalar powers are replaced with Hadamard powers. The analytical coefficient can be separated into an order-dependent factor and a harmonic factor:

$$\frac{(2\pi n)^{2i}}{(2i)!P^{2i}} = \frac{(2\pi/P)^{2i}}{(2i)!}n^{2i}, \tag{14}$$

$$\frac{(2\pi n)^{2i+1}}{(2i+1)!P^{2i+1}} = \frac{(2\pi/P)^{2i+1}}{(2i+1)!}n^{2i+1}. \tag{15}$$

This factorization is the direct route from the strict expansion to Eqs. 7–9. The relaxed model removes the deterministic order factor from the feature dictionary and learns the effective coefficient at each harmonic-order pair.

C Temporal Backbone and Implementation Details

C.1 Channel-Specific and Channel-Shared Convolutions

For a multichannel sequence $\mathbf{z}$, a standard 1D convolution is

$$y_t^{(d')} = \sum_{d=1}^{D} \sum_{j=0}^{\kappa-1} W_{d',d,j} z_{t+j}^{(d)}. \tag{16}$$

The channel-specific depthwise convolution used by the backbone is

$$y_t^{(d)} = \sum_{j=0}^{\kappa-1} W_{d,j} z_{t+j-\lfloor \kappa/2 \rfloor}^{(d)}, \tag{17}$$

while the channel-shared depthwise convolution reuses the same temporal kernel for all variables:

$$y_t^{(d)} = \sum_{j=0}^{\kappa-1} W_j z_{t+j-\lfloor \kappa/2 \rfloor}^{(d)}. \tag{18}$$

For clarity, Algorithm 1 shows its PyTorch-style pseudocode. The two operators provide variable-specific and shared temporal features without directly defining the high-order lifting contribution.

C.2 VLNet

The backbone we used follows the existing convolutional system, containing a temporal augmentation stage, channel-specific and channel-shared value lifting branches, a residual feature branch, and multi-scale fusion before adaptive average pooling. A pointwise projection is followed by a channel-specific depthwise convolution and a channel-shared depthwise convolution. The former learns variable-specific temporal filters, while the latter applies a shared temporal kernel across variables. Their intermediate responses are passed through the high-order lifting pathway and fused with a residual temporal branch before adaptive pooling and classification. The schematic overview of the model is shown in Figure 4.

In this paper, this model architecture is called VLNet and is treated as one instantiation of FT-Lift: the temporal backbone supplies intermediate representations, the value-lifting layer performs the high-order transformation analyzed in the main text, and the remaining modules provide residual fusion and classification.

C.3 Complexity

For input dimension d, N factorization branches, and maximum Taylor index I, the relaxed dictionary contains $2N(I+1)$ odd/even term groups before aggregation. A direct implementation can compute powers incrementally to avoid repeated exponentiation. Proposition 2 shows that the harmonic branches can collapse in function space; the factorization width should therefore be interpreted as part of the parameterization rather than as additional Fourier frequencies.

Algorithm 1 PyTorch pseudocode for channel-shared Convolution

```
def ChannelSharingConv(features, d, k):
    """
    d:          int, channel size
    k:          int, kernel size
    features:   features shaped (B, d, l)
    """
    inter_channel_conv_weight = nn.Parameter(
        torch.nn.init.normal_(
            torch.empty(k, )
        )
    ).unsqueeze(0).unsqueeze(0).repeat(d, 1, 1)

    output = torch.nn.functional.conv1d(
        features,
        weight=inter_channel_conv_weight,
        stride=1,
        padding=k // 2,
        groups=d,
        dilation=1
    )
```

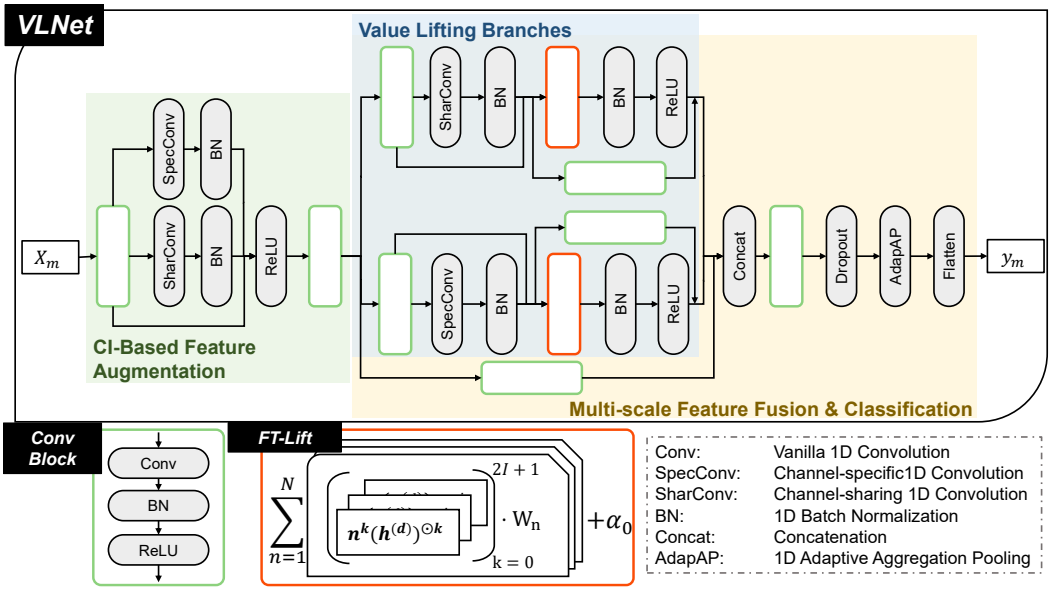


Figure 4: Overview of our VLNet framework.

D EXPERIMENTAL DETAILS

D.1 UEA DATASETS

For comparison with the general time-series frameworks, we selected 10 representative datasets as in Mu et al. (2025); Zhou et al. (2026) as follows: EthanolConcentration (EC), FaceDetection (FD), Handwriting (HW), Heartbeat (HB), JapaneseVowels (JV), PEMS-SF (PEMS), SelfRegulationSCP1 (SRS1), SelfRegulationSCP2 (SRS2), SpokenArabicDigits (SAD), and UWaveGestureLibrary (UW). From the whole UEA, we select 26 datasets from the UEA archive, with sample sizes ranging from 180 to 50,000, time-series lengths from 8 to 17,984, and dimensions ranging from 2 to 963, covering areas such as human activity recognition, motion classification, ECG/EEG/MEG classification, and audio spectrum classification. Following the 26-dataset setting used in prior MTSC studies Zhou et al. (2026), we exclude four datasets containing at most 100 samples.

D.2 HYPERPARAMETER SETTINGS & OPTIMIZATION

The existing implementation uses AdamW with a learning rate of 0.01, weight decay of 10^{-4}, a batch size of 16, and 200 epochs. The results in the broad baselines comparison are organized following previous studies Mu et al. (2025); Zhou et al. (2026). The controlled UEA study reports dataset-level mean accuracy over five random seeds (1–5), average rank, win/tie/loss counts, and paired Wilcoxon signed-rank tests. The default high-order configuration uses three harmonic branches and a low Taylor order. In addition, we fix all the random seeds according to the reproducibility settings in PyTorch[1].

In the matched strict/relaxed/direct-polynomial comparisons, optimizer settings, training budget, and temporal-backbone settings are kept identical across variants so that differences can be attributed to the value-lifting parameterization.

D.3 METRICS AND STATISTICS

Two main tables report dataset-level mean accuracy (Avg. ACC), top-k counts (#Top-1, #Top-2, and #Top-3), average rank (Avg. Rank), win/tie/loss counts (#Wins, #Draws, and #Loses), and paired Wilcoxon signed-rank tests against VLNet (p-value). We additionally use the Friedman test to compute the critical difference (CD) of the ranking results for both broad benchmark comparisons and structural ablations. For the matched lifting comparison in Table 3, we report mean accuracy (Mean ACC), the standard deviation across dataset-level mean accuracies (Std.), average rank (Avg. Rank), win/tie/loss counts (Relaxed-FT W/T/L), and paired Wilcoxon signed-rank tests (p vs. Relaxed-FT) against Relaxed-FT. The controlled UEA comparison and the repeated runs record ACC, balanced accuracy (BACC), macro recall (Macro REC), macro precision (Macro PRE), and Macro-F1 over repeated runs. For the matched lifting study, Holm correction is applied across the five prespecified pairwise comparisons against Relaxed-FT. Three tables tuning on I, K, and N report the ACC only.

D.4 BASELINES

We first selected 13 general time-series frameworks evaluated on 10 complex UEA datasets as done in (Mu et al., 2025), *i.e.*, LSTNet (Lai et al., 2018), LSSL (Gu et al., 2021), FEDformer (Zhou et al., 2022), Flowformer (Wu et al., 2022b), SCINet (Liu et al., 2022), Dlinear (Zeng et al., 2023), PatchTST (Nie et al., 2022), MICN (Wang et al., 2023), TimesNet (Wu et al., 2022a), Crossformer (Zhang & Yan, 2023), ModernTCN (Luo & Wang, 2024), TimeMixer++ (Wang et al., 2025c), and TVNet (Li et al., 2025). They are designed to address classification, imputation, forecasting, and anomaly detection. The results are derived from Mu et al. (2025); Wang et al. (2025c); Li et al. (2025).

We compare against 19 MTSC-dedicated advanced methods, comprising a feature-based algorithm: MiniRocket (Dempster et al., 2021); a GNN-based model: TodyNet (Liu et al., 2024); four CNN-based models: TSLANet (Eldele et al., 2024), EdgeMTSC (Zhou et al., 2026), InvConvNet (Germain et al., 2025), TimeSliver (Pandey et al., 2026); six Transformer-based models: TST (Zerveas et al., 2021), SVPT (Zuo et al., 2023), ConvTran (Foumani et al., 2024), TimeMIL (Chen et al.,

[1] https://pytorch.org/docs/stable/notes/randomness.html

2024), DKN (Xiao et al., 2024b), KATN (Xiao et al., 2025); a Fisher-based model: FIC-TSC (Chen et al., 2025), and two frequency-aware methods: MPTSNet (Mu et al., 2025), DTFPNet (Yu et al., 2025). The results are derived from Liu et al. (2024); Eldele et al. (2024); Zhou et al. (2026); Zerveas et al. (2021); Zuo et al. (2023); Foumani et al. (2024); Chen et al. (2024); Xiao et al. (2024b; 2025); Mu et al. (2025); Yu et al. (2025); Germain et al. (2025); Pandey et al. (2026); Chen et al. (2025).

E Existing Structural Ablation

The original system ablation reported 78.4% average accuracy after removing the channel-specific branch (VLNet-wo-specific), 78.8% after removing the channel-shared branch (VLNet-wo-generic), 78.6% after removing temporal augmentation (VLNet-wo-aug), 78.2% after removing the FT-Lift block (VLNet-wo-Ft-Lift), 77.7% after replacing the polynomial block with the original sine-cosine Fourier block (VLNet-w-fourier), 77.2% after removing the polynomial highway (VLNet-wo-phw), and 78.5% after removing the multi-scale highway (VLNet-wo-mhw), compared with approximately 79.0% for the full model. These experiments establish that the lifting path participates in the performance of the complete architecture, while Table 3 provides the more controlled comparison among alternative lifting parameterizations.

F NUMERICAL DIAGNOSTICS AT HIGH DEGREE

We construct a synthetic dataset to study the effect of lifting order and high-degree numerical behavior. For every target order $p \in \{1, 2, 4, 8\}$, inputs are sampled independently as $X \in \mathbb{R}^{4\times 128}$ with $x_{c,t} \sim \mathcal{N}(0, 1)$. We compute the probabilists' Hermite polynomial recursively by $H_0(x) = 1$, $H_1(x) = x$, and $H_{p+1}(x) = xH_p(x) - pH_{p-1}(x)$, and define

$$q_p(X) = \frac{1}{4 \cdot 128} \sum_{c=1}^{4} \sum_{t=1}^{128} \frac{H_p(x_{c,t})}{\sqrt{p!}} + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, 0.02^2). \tag{19}$$

The class is $y = \mathbb{I}[q_p(X) > 0]$. We sample until both classes contain the requested number of examples, then shuffle them. The training, validation, and test sets contain 1024, 256, and 512 examples, respectively, and are generated independently using seed offsets $1000s + p + 1$, $1000s + p + 2$, and $1000s + p + 3$ for run seed s. The $\sqrt{p!}$ factor normalizes the variance of H_p under a standard Gaussian measure. The value-order results in Table 4 report four independent runs for each configuration.

For the secondary numerical diagnostics, we fix the synthetic target at $p = 8$ and sweep the mean maximum polynomial degree over $K \in \{1, 2, 4, 6, 8, 10, 12, 16\}$. At the largest tested degree, the mean order-wise raw-gradient norm of Strict-FT varies from approximately 3.27×10^{-10} at low order to 1.24×10^{-23} at order 16, giving a max/min ratio of approximately 2.63×10^{13}. The corresponding ratios are approximately 1.05 for Relaxed-FT and 1.06 for both polynomial controls. This shows that the strict analytical coefficients produce a strongly order-dependent raw-gradient scale, but it does not imply a consistent accuracy advantage for the relaxed model.

The forward numerical scale also deteriorates with degree. At $K = 12$, the recorded maximum term magnitudes reach approximately 4.7×10^{27} for Direct Polynomial, 1.6×10^{19} for the parameter-matched polynomial, 3.4×10^{29} for Relaxed-FT, and 8.1×10^{30} for Strict-FT. Several degree-16 activation summaries are unavailable, even though the recorded non-finite gradient rate remains zero. Across the degree sweep, predictive accuracy remains in a narrow near-chance range, and the best method changes with K. We therefore treat these measurements as a caution against indiscriminate high-degree monomial expansion.

G Repeated-Run Results of the Existing Implementation

Table 5: Repeated-run results of the existing implementation over ten random seeds. *Max* and *Min* denote the averages of the best and worst runs, respectively.

	ACC	BACC	Macro F1	Macro PRE	Macro REC
AWR	**0.995**$_{\pm 0.002}$	**0.995**$_{\pm 0.002}$	**0.995**$_{\pm 0.002}$	**0.995**$_{\pm 0.002}$	**0.995**$_{\pm 0.002}$
CT	**0.998**$_{\pm 0.000}$	**0.998**$_{\pm 0.001}$	**0.998**$_{\pm 0.000}$	**0.998**$_{\pm 0.000}$	**0.998**$_{\pm 0.001}$
CR	**1.000**$_{\pm 0.000}$	**1.000**$_{\pm 0.000}$	**1.000**$_{\pm 0.000}$	**1.000**$_{\pm 0.000}$	**1.000**$_{\pm 0.000}$
ER	**0.928**$_{\pm 0.010}$	**0.928**$_{\pm 0.010}$	**0.928**$_{\pm 0.011}$	**0.933**$_{\pm 0.011}$	**0.928**$_{\pm 0.010}$
EP	**0.996**$_{\pm 0.004}$	**0.996**$_{\pm 0.003}$	**0.996**$_{\pm 0.003}$	**0.996**$_{\pm 0.003}$	**0.996**$_{\pm 0.003}$
FM	**0.626**$_{\pm 0.036}$	**0.625**$_{\pm 0.036}$	**0.617**$_{\pm 0.040}$	**0.637**$_{\pm 0.035}$	**0.625**$_{\pm 0.036}$
HMD	**0.431**$_{\pm 0.054}$	**0.362**$_{\pm 0.069}$	**0.336**$_{\pm 0.101}$	**0.351**$_{\pm 0.147}$	**0.362**$_{\pm 0.069}$
HW	**0.653**$_{\pm 0.026}$	**0.649**$_{\pm 0.025}$	**0.637**$_{\pm 0.032}$	**0.660**$_{\pm 0.031}$	**0.649**$_{\pm 0.025}$
HB	**0.795**$_{\pm 0.012}$	**0.684**$_{\pm 0.035}$	**0.701**$_{\pm 0.032}$	**0.775**$_{\pm 0.039}$	**0.684**$_{\pm 0.035}$
JV	**0.994**$_{\pm 0.003}$	**0.995**$_{\pm 0.003}$	**0.994**$_{\pm 0.003}$	**0.994**$_{\pm 0.003}$	**0.995**$_{\pm 0.003}$
MI	**0.638**$_{\pm 0.019}$	**0.638**$_{\pm 0.019}$	**0.633**$_{\pm 0.020}$	**0.647**$_{\pm 0.022}$	**0.638**$_{\pm 0.019}$
NATO	**0.981**$_{\pm 0.007}$	**0.981**$_{\pm 0.007}$	**0.980**$_{\pm 0.007}$	**0.982**$_{\pm 0.007}$	**0.981**$_{\pm 0.007}$
RS	**0.928**$_{\pm 0.013}$	**0.933**$_{\pm 0.012}$	**0.933**$_{\pm 0.012}$	**0.934**$_{\pm 0.012}$	**0.933**$_{\pm 0.012}$
SRS1	**0.867**$_{\pm 0.014}$	**0.867**$_{\pm 0.014}$	**0.867**$_{\pm 0.014}$	**0.871**$_{\pm 0.013}$	**0.867**$_{\pm 0.014}$
SRS2	**0.572**$_{\pm 0.018}$	**0.572**$_{\pm 0.018}$	**0.564**$_{\pm 0.021}$	**0.578**$_{\pm 0.019}$	**0.572**$_{\pm 0.018}$
SAD	**0.999**$_{\pm 0.000}$	**0.999**$_{\pm 0.000}$	**0.999**$_{\pm 0.000}$	**0.999**$_{\pm 0.000}$	**0.999**$_{\pm 0.000}$
UW	**0.927**$_{\pm 0.009}$	**0.927**$_{\pm 0.009}$	**0.926**$_{\pm 0.009}$	**0.928**$_{\pm 0.009}$	**0.927**$_{\pm 0.009}$
PS	**0.293**$_{\pm 0.012}$	**0.293**$_{\pm 0.012}$	**0.283**$_{\pm 0.013}$	**0.311**$_{\pm 0.012}$	**0.293**$_{\pm 0.012}$
EW	**0.710**$_{\pm 0.050}$	**0.594**$_{\pm 0.055}$	**0.625**$_{\pm 0.125}$	**0.675**$_{\pm 0.127}$	**0.594**$_{\pm 0.055}$
EC	**0.353**$_{\pm 0.012}$	**0.353**$_{\pm 0.012}$	**0.322**$_{\pm 0.027}$	**0.358**$_{\pm 0.021}$	**0.353**$_{\pm 0.012}$
FD	**0.694**$_{\pm 0.006}$	**0.694**$_{\pm 0.006}$	**0.693**$_{\pm 0.006}$	**0.695**$_{\pm 0.006}$	**0.694**$_{\pm 0.006}$
LSST	**0.597**$_{\pm 0.023}$	**0.375**$_{\pm 0.033}$	**0.478**$_{\pm 0.047}$	**0.530**$_{\pm 0.045}$	**0.375**$_{\pm 0.033}$
LIB	**0.919**$_{\pm 0.009}$	**0.919**$_{\pm 0.009}$	**0.918**$_{\pm 0.009}$	**0.925**$_{\pm 0.008}$	**0.919**$_{\pm 0.009}$
PEMS	**0.898**$_{\pm 0.021}$	**0.895**$_{\pm 0.021}$	**0.894**$_{\pm 0.022}$	**0.900**$_{\pm 0.020}$	**0.895**$_{\pm 0.021}$
PD	**0.990**$_{\pm 0.002}$	**0.990**$_{\pm 0.002}$	**0.990**$_{\pm 0.002}$	**0.990**$_{\pm 0.002}$	**0.990**$_{\pm 0.002}$
IW	**0.688**$_{\pm 0.003}$	**0.688**$_{\pm 0.003}$	**0.686**$_{\pm 0.004}$	**0.692**$_{\pm 0.006}$	**0.688**$_{\pm 0.003}$
Avg.	**0.787**	**0.767**	**0.769**	**0.783**	**0.767**
Max	**0.812**	**0.794**	**0.806**	**0.820**	**0.794**
Min	**0.763**	**0.741**	**0.732**	**0.742**	**0.741**

H TUNING N ON UEA DATASETS

Table 6: Results of the VLNet variants conditioned on $N \in \{1, 2, 3, 4, 5, 6\}$.

	VLNet-(N=1)	VLNet-(N=2)	VLNet-(N=3)	VLNet-(N=4)	VLNet-(N=5)	VLNet-(N=6)
AWR	0.993	0.99	0.993	0.997	0.997	0.993
CT	0.998	0.999	0.998	0.998	0.999	0.997
CR	1.0	1.0	1.0	1.0	1.0	1.0
ER	0.893	0.915	0.919	0.93	0.922	0.933
EP	1.0	1.0	0.993	0.993	0.993	0.993
FM	0.67	0.69	0.66	0.67	0.65	0.64
HMD	0.459	0.405	0.446	0.432	0.432	0.459
HW	0.647	0.634	0.629	0.679	0.689	0.639
HB	0.815	0.776	0.805	0.8	0.795	0.761
JV	0.995	0.995	0.997	0.997	0.997	0.992
MI	0.63	0.61	0.63	0.58	0.62	0.69
NATO	0.994	0.989	0.983	0.989	0.994	0.967
RS	0.928	0.928	0.934	0.914	0.941	0.901
SRS1	0.84	0.874	0.874	0.867	0.857	0.857
SRS2	0.544	0.572	0.572	0.594	0.611	0.556
SAD	1.0	0.999	0.999	0.999	1.0	0.999
UW	0.922	0.922	0.925	0.925	0.9	0.919
PS	0.293	0.285	0.287	0.29	0.302	0.298
EW	0.695	0.725	0.748	0.725	0.71	0.679
EC	0.35	0.357	0.357	0.327	0.361	0.335
FD	0.69	0.695	0.692	0.681	0.692	0.691
LSST	0.603	0.586	0.614	0.566	0.631	0.627
LIB	0.911	0.917	0.928	0.917	0.928	0.917
PEMS	0.89	0.902	0.896	0.884	0.896	0.89
PD	0.989	0.99	0.991	0.991	0.991	0.991
IW	0.695	0.683	0.681	0.694	0.693	0.687
Avg. Acc	0.786	0.786	0.79	0.786	0.792	0.785

I TUNING DEPTHWISE CONVOLUTION KERNEL SIZES

Table 7: Results of the VLNet variants conditioned on $\kappa \in \{5, 7, 9, 11, 13, 15, 17, 47, 67\}$.

	VLNet-(κ=5)	VLNet-(κ=7)	VLNet-(κ=9)	VLNet-(κ=11)	VLNet-(κ=13)	VLNet-(κ=15)	VLNet-(κ=17)	VLNet-(κ=47)	VLNet-(κ=67)
AWR	0.997	0.99	0.987	0.993	0.99	0.993	0.99	0.993	0.993
CT	0.995	0.997	0.997	0.998	0.997	0.999	0.998	0.998	0.998
CR	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ER	0.956	0.885	0.911	0.889	0.893	0.9	0.881	0.919	0.937
EP	1.0	1.0	0.993	1.0	1.0	1.0	1.0	0.993	0.993
FM	0.63	0.6	0.64	0.68	0.61	0.61	0.67	0.66	0.67
HMD	0.432	0.419	0.405	0.419	0.419	0.392	0.419	0.446	0.405
HW	0.338	0.319	0.507	0.532	0.485	0.526	0.525	0.629	0.669
HB	0.771	0.766	0.795	0.771	0.776	0.79	0.81	0.805	0.771
JV	0.992	1.0	0.995	0.997	0.995	0.997	0.997	0.997	0.995
MI	0.59	0.64	0.65	0.61	0.65	0.67	0.62	0.63	0.6
NATO	1.0	0.994	0.989	0.994	0.983	0.983	0.989	0.983	0.978
RS	0.868	0.934	0.934	0.941	0.934	0.921	0.921	0.934	0.934
SRS1	0.904	0.867	0.894	0.874	0.884	0.87	0.881	0.874	0.853
SRS2	0.539	0.567	0.567	0.611	0.567	0.561	0.578	0.572	0.578
SAD	0.999	0.998	1.0	0.998	0.999	1.0	1.0	0.999	1.0
UW	0.859	0.891	0.862	0.891	0.897	0.925	0.903	0.925	0.922
PS	0.263	0.279	0.295	0.274	0.273	0.318	0.295	0.287	0.293
EW	0.634	0.557	0.595	0.649	0.588	0.656	0.718	0.748	0.656
EC	0.361	0.369	0.346	0.35	0.346	0.361	0.354	0.357	0.35
FD	0.683	0.696	0.69	0.679	0.693	0.69	0.688	0.692	0.693
LSST	0.567	0.539	0.576	0.559	0.548	0.576	0.598	0.614	0.645
LIB	0.967	0.95	0.944	0.961	0.939	0.939	0.928	0.928	0.917
PEMS	0.838	0.844	0.821	0.844	0.879	0.884	0.896	0.896	0.873
PD	0.993	0.989	0.991	0.989	0.992	0.988	0.993	0.991	0.986
IW	0.684	0.695	0.691	0.685	0.683	0.692	0.685	0.681	0.687
Avg. Acc	0.764	0.761	0.772	0.776	0.77	0.778	0.782	0.79	0.784

J TUNING I ON UEA DATASETS

Table 8: Results of the VLNet variants conditioned on $I \in \{1, 2, 3, 4\}$.

	VLNet-($I = 1$)	VLNet-($I = 2$)	VLNet-($I = 3$)	VLNet-($I = 4$)
AWR	0.993	0.997	0.993	1.0
CT	0.998	0.999	0.999	0.999
CR	1.0	1.0	1.0	1.0
ER	0.919	0.922	0.911	0.93
EP	0.993	0.993	0.993	0.993
FM	0.66	0.64	0.65	0.61
HMD	0.446	0.432	0.459	0.459
HW	0.629	0.689	0.684	0.656
HB	0.805	0.785	0.8	0.776
JV	0.997	0.992	0.995	0.995
MI	0.63	0.61	0.62	0.62
NATO	0.983	0.978	0.989	0.989
RS	0.934	0.928	0.928	0.914
SRS1	0.874	0.881	0.887	0.884
SRS2	0.572	0.578	0.583	0.589
SAD	0.999	1.0	1.0	0.998
UW	0.925	0.931	0.912	0.919
PS	0.287	0.303	0.287	0.29
EW	0.748	0.702	0.725	0.42
EC	0.357	0.338	0.357	0.35
FD	0.692	0.676	0.688	0.683
LSST	0.614	0.569	0.601	0.61
LIB	0.928	0.922	0.889	0.917
PEMS	0.896	0.879	0.896	0.89
PD	0.991	0.991	0.991	0.986
IW	0.681	0.674	0.693	0.689
Avg. Acc	0.79	0.785	0.79	0.776