Knowledge Aggregation Transformer Network for Multivariate Time Series Classification

Zhiwen Xiao, *Member, IEEE*, Huanlai Xing, *Member, IEEE*, Rong Qu, *Senior Member, IEEE*, Hui Li, Huagang Tong, Shouxi Luo, *Member, IEEE*, Jing Song, Li Feng, and Qian Wan

***Abstract*—Over the years, various sophisticated deep learning algorithms have surfaced for multivariate time series classification (MTSC), notably the dual-network-based model. This model comprises two parallel networks tailored to time series data: one for local feature extraction and the other for global relation extraction. However, effectively integrating these dual networks poses a significant challenge. To address this, we propose a knowledge aggregation transformer network (KATN) for MTSC. KATN, composed of four aggregation transformer blocks, extracts abundant regularizations and connections hidden within the data. Each block incorporates a modified residual network (MResNet) for local feature extraction and a multi-head attention network for global relation extraction. Initially, the block merges MResNet's output feature with that of the multi-head attention network through an additive operation. Subsequently, it aligns features with a fully connected (i.e., dense) layer and activates neural units using the Gaussian error linear unit function. This strategic feature aggregation allows for capturing long-range dependencies among multiple variables in multivariate time series data. Experimental results demonstrate that KATN significantly outperforms 6 state-of-the-art transformer variants, achieving a 'win'/'tie'/'lose' record of 9/6/15 and securing the lowest AVG_rank score. Furthermore, when evaluated against 18 existing MTSC algorithms across 13 UEA datasets, KATN consistently delivers superior performance, attaining the lowest AVG_rank score among all compared methods.**

***Index Terms*—Data Mining, Deep Learning, Feature Aggregation, Multivariate Time Series Classification, Transformer**

I. Introduction

MULTIVARIATE time series data has been widely applied in a variety of domains, e.g., heart failure risk analysis [1], industrial activity recognition [2], rumor detection [3], and anomaly detection [4], [5]. Different from image, text, and video data, multivariate time series consist of sequential data points that are organized chronologically and correspond to multiple time-dependent variables, incorporating both local and global patterns. An algorithm designed for multivariate time series classification (MTSC) aims to extract distinct local and global features from each univariate time series (UTS) while simultaneously identifying interconnections among these UTS sequences [6].

In recent times, deep learning (DL) models have garnered significant interest within the MTSC community. These algorithms effectively model the internal data representation hierarchy, capturing the inherent connections among representations [6], [7], [8]. Single-network-based and dual-network-based models are two popular DL-based streams for MTSC. A single-network-based model generally employs a unified (frequently hybridized) network architecture to perform both feature extraction and the identification of relationships within the data. For instance, He *et al.* [9] presented a general neural network model based on convolutional filtering, called Rel-CNN, to extract both local and global features in time series. Ma *et al.* [10] introduced a robust generation time series method, called AJ-RNN, which employs an adversarial joint-learning structure combined with recurrent neural network (RNN). Chen *et al.* [11] devised a paralleling attention structure that gracefully integrates two class of attention variants for MTSC. Chen *et al.* [12] put forward TAR-GAN, a rule mining-based approach that employs the generative adversarial technique to search useful shapelets of time series. On the other hand, a dual-network-based model usually comprises two parallel networks: one dedicated to extracting local features and the other aimed at capturing global relationships. The local feature network typically employs convolutional neural networks (CNNs) to discern local patterns, while the global relation network, commonly based on RNNs or attention-based networks, is responsible for uncovering connections among the extracted representations. Notable dual-network-based models includes the robust temporal feature network (RTFN) containing a temporal feature network and a long short-term memory (LSTM)-based attention network [13], LSTM-fully convolutional network (LSTM-FCN) consisting of LSTM-based network and FCN [14], residual attention net (ResNet-Transformer) composed of a residual network and a transformer-based network [15], robust semisupervised model (SelfMatch) comprising a residual network and an LSTM-based attention network [16], and densely knowledge-aware network (DKN) consisting of a residual multi-head convolutional network and a transformer-based network [17]. Unlike

This work was partially supported by the Natural Science Foundation of Hebei Province (No. F2022105027), the Doctoral Innovation Fund Program of Southwest Jiaotong University (No. CX-2025ZD09), Southwest Jiaotong University, China, and the Fundamental Research Funds for the Central Universities, P. R. China (Corresponding Author: Huanlai Xing).

Z. Xiao, H. Xing, S. Luo, L. Feng, and Q. Wan are with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 610031, China, and also with the Tangshan Institute of Southwest Jiaotong University, Tangshan 063000, China (Emails: xiao1994zw@163.com; hxx@home.swjtu.edu.cn; sxluo@swjtu.edu.cn; fengli@swjtu.edu.cn; qianwan@my.swjtu.edu.cn).

R. Qu is with the School of Computer Science, University of Nottingham, Nottingham NG7 2RD 455356, UK (Email: rong.qu@nottingham.ac.uk).

H. Li is with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (Email: lihui10@mail.xjtu.edu.cn).

H. Tong is with the College of Economic and Management, Nanjing Tech University, Puzhu Road(S), Nanjing 211816, China (Email: huagangtong@gmail.com).

J. Song is with the Informatization Research Institute, Southwest Jiaotong University, Chengdu 610031, China (Email: jesen811206@126.com).

single-network-based models, dual-network-based algorithms account for the characteristics of multivariate time series data, employing a divide-and-conquer approach to achieve commendable performance across various time series problems. However, dual-network-based algorithms may face the following two challenges:

- *Most dual-network-based algorithms use the concatenation feature aggregation method to integrate the output feature of the local feature network with that of the global relation network before reaching the final prediction layer. Nevertheless, this method may lead to an inadequate alignment between diverse feature types extracted by these two distinct networks. In particular, the lack of interaction between different feature types extracted at lower levels impedes the relationship exploration among various variables within multivariate time series data.*
- *As known, a learning model's performance usually hinges on quality of the semantic information extracted from lower and higher levels within the representation hierarchy* [18], [19]. *Thus, for an arbitrary dual-network-based model, enhancing the aggregation between features derived from the local feature network and those from the global relation network at each level significantly enriches the extracted semantic information, consequently enhancing the model's performance.*

Recently, a number of researchers have focused their efforts on aggregating features across various dimensions, enhance model's performance in diverse computer vision (CV) tasks [20]. Some studies considered the attention method into CNNs to combine the features on both spatial and channel dimensions, such as dual-attention network [21] and SCA-CNN [22]. Dosovitskiy *et al.* [23] proposed a spatial self-attention module to enhance long-range dependencies between image pixels. Chen *et al.* [24] devised Mixformer based on channel attention to fuse spatial and channel information in vision tasks. Chen *et al.* [25] developed a dual aggregation transformer network based on spatial window and channel-wise self-attention for image super-resolution. However, unlike CV, the feature aggregation methods in time series exhibit notable limitations, listed below.

- *Unlike image data, multivariate time series entails a sequence of chronologically arranged data points linked to several time-dependent variables. Thus, outstanding feature aggregation methods used in CV may not be directly transferable or suitable for time series domain.*
- *Currently, a significant gap exists in tailored methodologies capable of deeply integrating distinct feature types, e.g., local and global patterns, to capture long-range dependencies among multiple variables within multivariate time series data. This limitation is prominent in most dual-network-based algorithms, particularly in efficiently hybridizing the local feature network with the global relation network.*

To overcome the limitations above, we propose a knowledge aggregation transformer network (KATN) for MTSC. Different from most of the dual-network-based algorithms combining disparate feature types at higher levels through concatenation, KATN deeply aggregates various feature types within each level. This aggregation can enrich the semantic information substantially, thereby effectively extracting long-range dependencies among multiple variables in multivariate time series data.

Our primary contributions are outlined below.

- This paper designs a deep feature aggregation network for MTSC, called KATN. KATN comprises four aggregation transformer blocks adept at amalgamating various feature types at lower and higher levels, exploring abundant regularizations and connections within the data, as depicted in Fig. 1.
- In each aggregation transformer block, a modified residual network (MResNet) and a multi-head attention network are responsible for local feature and global relation extraction, respectively. The block uses an additive operation to connect the output feature of MResNet with that of the multi-head attention network. A fully connected (i.e., dense) layer is employed for feature alignment. Meanwhile, this block activates neural units via the Gaussian error linear unit (GeLU) [26] function.
- The experiments demonstrate that in comparison to 6 state-of-the-art (SOTA) transformer algorithms, KATN achieves 9 wins, 6 ties, and 15 losses, along with the lowest AVG_rank score of 2.400. Moreover, KATN surpasses 18 existing MTSC algorithms based on both 'win'/'tie'/'lose' metrics and AVG_rank, as measured by top-1 accuracy. Specifically, among the 30 datasets, KATN secures victory in 13, achieving the lowest AVG_rank score of 4.167.

The paper's structure for the remaining sections is: in Section II, we review the SOTA methods in MTSC and transformer-based algorithms, highlighting their strengths and limitations. Section III introduces the proposed KATN framework, elaborating on its architecture and core components. Following this, Section IV discusses the experimental setup, results, and comparative analysis. Finally, Section V concludes the paper by summarizing key findings and outlining potential future directions.

II. Related Work

This section reviews a number of MTSC and transformer algorithms.

A. MTSC Algorithms

Traditional and DL-based algorithms are two main streams for MTSC.

1) Traditional Algorithms: Distance- and feature-based approaches are two typical representatives of traditional algorithms for MTSC [6], [11]. The classical distance-based method employs nearest neighbor (NN) and dynamic time warping (DTW) to measure the similarities between spatial features in the data, such as, dependent DTW, adaptive DTW, and independent DTW [27]. The ensemble learning algorithm with DTW and NN has been developed to address various time series problems, e.g., the elastic ensemble approach [28], collective of transformation-based ensemble (COTE) method [29], hierarchical voting COTE (HIVE-COTE) [30], improved

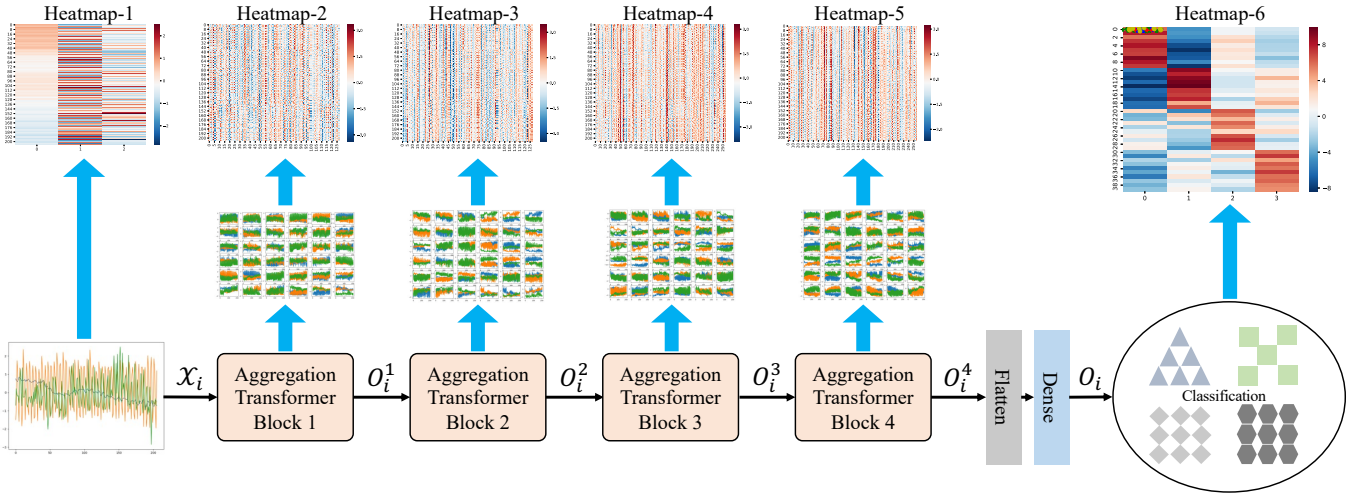


Fig. 1. Overview of KATN. The proposed KATN consists of four aggregation transformer blocks for local feature and global relation extraction in multivariate time series. We visualize the representations learned by each network block in the KATN, where the top picture visualizes the heatmap representation of each network layer. From these heatmaps, a noticeable pattern emerges: as the network's depth increases, samples sharing similar characteristics tend to cluster together more prominently. Note: $\mathcal{X}_i$, $i = 1, 2, ..., N$, is the i-th input sample, where N represents the number of input samples. O_i^j, $i = 1, 2, ..., N$, $j = 1, 2, 3, 4$, denotes the i-th output feature vector of j-th aggregation transformer block, while O_i stands for the i-th output feature vector of KATN.

meta HIVE-COTE (HIVE-COTE 2.0) [31], and explainable-by-design ensemble approach [32].

Feature-based algorithms concentrating on mining the representative features from the input. For instance, Li *et al.* [33] introduced an efficient shapelet-based discovery approach called bspcover to construct a collection of high-quality shapelets. Baldán and Benítez [34] proposed an interpretable representation approach to handle a variety of MTSC applications. Shifaz *et al.* [35] devised a scalable and accurate forest method, namely TS-Chief, to achieve excellent performance on various time series tasks. The pattern similarity method [36], hidden-unit logistic model [37], time series forest [38], online rule-based classifier learning [39], autoregressive tree-based ensemble [40], bag of symbolic Fourier approximation symbols [41], WEASEL+MUSE [42], and fuzzy cognitive map [43] are all feature-based.

2) DL-based Algorithms: DL-based algorithms aim to model the internal data representation hierarchy, mining the inherent connections among representations [6]. Single- and dual-network-based models present research streams [7], [8]. The flexible multi-head linear attention (FMLA) [44], InceptionTime [45], dynamic temporal pooling network [46], fully-convolutional network [47], DA-Net [11], multi-process collaborative architecture [48], TAR-GAN [12], Rel-CNN [9], shapelet-neural network [49], AJ-RNN [10], deep contrastive representation learning with self-distillation [50], reservoir computing [51] , ROCKET [52], dynamic graph attention autoencoder [53], dynamic component alignment [54], echo state network [55], and MiniROCKET [56] are well known single-network-based models. On the other hand, typical dual-network-based models include ResNet-Transformer [15], SelfMatch [16], DKN [17], LSTM-FCN [14], SelfMatch [16], robust neural temporal search (RNTS) [57], perceptive and lightweight capsule models [58], and attentional prototypical network (TapNet) [59].

B. Transformer Algorithms

Since 2017, the Transformer model has gained popularity across various domains, such as time series, CV, natural language processing, and information retrieval, due to its capability to capture relationships among features at diverse positions [60]. The self-attention based transformer [61] serves as a trailblazer in the realm of transformers, with numerous enhanced transformer architectures emerging in this domain. For instance, to reduce the self-attention complexity in both time and space, Wang *et al.* [62] introduced Linformer based on linear attention. Liu *et al.* [63] proposed a hierarchical transformer using shifted windows to model different image scales. Ding *et al.* [64] presented a simple and effective dual attention transformer to capture global context while maintaining computational efficiency. Li *et al.* [65] put forward a dilated convolutional transformer-based generative adversarial network for time series anomaly detection. The transformer with spatial self-attention [23], DKN [17], Mixformer [24], FMLA [44], dual aggregation transformer [25], intention-aware dynamic transformer [66], dynamic graph transformer [67], and temporal graph transformer [68] are typical transformer models.

III. The Proposed KATN

This section first introduces the structure of KATN and its key components. Then, it describes the loss function.

A. Overview

KATN comprises four aggregation transformer blocks responsible for amalgamating various feature types at lower and higher levels, mining abundant relationships and rules

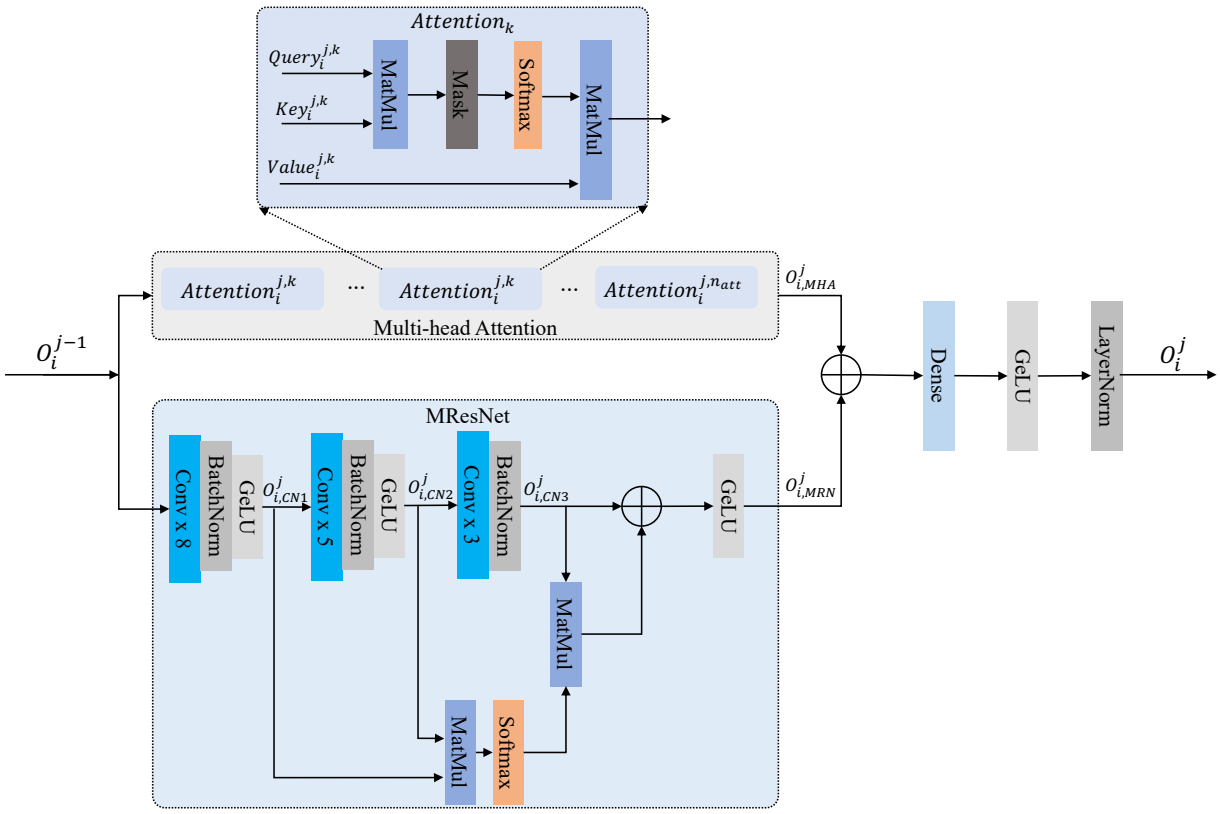


Fig. 2. Architecture of aggregation transformer block. This block integrates a modified residual network (MResNet) with a multi-head attention network in parallel. Similar to the vanilla ResNet [47], MResNet mainly consists of three 1-dimensional convolutional layers. However, MResNet adopts the attention-based method to combine the output features from these convolutional layers. Rather than the typically used rectified linear unit (ReLU) function, MResNet uses the Gaussian error linear unit (GeLU) [26] function for neural unit activation. The multi-head attention network consists of n_{att} attention modules mining relationships among features across diverse locations. Note: "MatMul" denotes the matrix multiplication operation. "Conv x 8" is a 1-dimensional convolutional layer with a kernel size of 8. "BatchNorm" and "LayerNorm" are the batch normalization and layer normalization operations, respectively. O_i^{j-1} and O_i^j are the input and output feature vectors of the j-th aggregation transformer block, $j = 1, 2, 3, 4$, associated with the input sample, $\mathcal{X}_i$, $i = 1, 2, ..., N$, where N represents the number of input samples. In particular, $O_i^{j-1} = \mathcal{X}_i$, when $j = 1$.

hidden in multivariate time series. The structure of KATN is shown in Fig. 1. Within each aggregation transformer block, MResNet and the multi-head attention network handle local feature and global relation extraction, respectively. This block connects the output feature of MResNet with that of the multi-head attention network using additive feature aggregation. It employs a fully connected (i.e., dense) layer for feature alignment. Neural units are activated by the GeLU function.

B. Aggregation Transformer

Each aggregation transformer block adeptly combines local patterns from MResNet and global patterns from the multi-head attention network, capturing extensive dependencies among multiple variables within multivariate time series data. The architecture of an aggregation transformer block is depicted in Fig. 2.

Let $\mathcal{X}_i$, $i = 1, 2, ..., N$, denote the i-th input sample, where N represents the number of input samples. Given input, $\mathcal{X}_i$, O_i^{j-1} and O_i^j are the input and output feature vectors of the j-th aggregation transformer block, $j = 1, 2, 3, 4$, respectively. Note that O_i^{j-1} is equal to $\mathcal{X}_i$, when $j = 1$.

1) MResNet: Similar to the vanilla ResNet [47], MResNet primarily comprises three 1-dimensional convolutional layers. But, MResNet employs an attention-based approach to integrate the output features from these convolutional layers. Different from ResNet that typically adopts the rectified linear unit (ReLU) function, MResNet utilizes GeLU activation for neural units.

Let $O_{i,CN1}^j$, $O_{i,CN2}^j$, and $O_{i,CN3}^j$ be the output feature vectors of three 1-dimensional convolutional layers in MResNet, respectively. These vectors are defined as:

$$\begin{aligned} O_{i,CN1}^j &= GeLU(BN(Conv(O_i^{j-1}))) \\ O_{i,CN2}^j &= GeLU(BN(Conv(O_{i,CN1}^j))) \\ O_{i,CN3}^j &= BN(Conv(O_{i,CN2}^j)) \end{aligned} \tag{1}$$

where, $Conv()$, $BN()$, and $GeLU()$ are the 1-dimensional convolution, batch normalization, and GeLU functions, respectively.

MResNet transforms the output feature vectors, namely $O_{i,CN1}^j$, $O_{i,CN2}^j$, and $O_{i,CN3}^j$, from three 1-dimensional convolutional layers to an output vector, $O_{i,MRN}^j$, by an attention-

TABLE I
OVERVIEW OF 30 UEA MULTIVARIABLE TIME SERIES BENCHMARK DATASETS. NOTES: AUDIO SPECTRA (AS), ELECTROCARDIOGRAM (ECG), ELECTROENCEPHALOGRAM (EEG), HUMAN ACTIVITY RECOGNITION (HAR), AND MAGNETOENCEPHALOGRAPHY (MEG).

Dataset Index	Dataset Name	Dimensions	SeriesLength	TrainSize	NumClasses	TestSize	Type
AWR	ArticularyWordRecognition	9	144	275	25	300	Motion
AF	AtrialFibrillation	2	640	15	3	15	ECG
BM	BasicMotions	6	100	40	4	40	HAR
CT	CharacterTrajectories	3	182	1422	20	1436	Motion
CK	Cricket	6	1197	108	12	72	HAR
DDG	DuckDuckGeese	1345	270	50	5	50	AS
EW	EigenWorms	6	17984	128	5	131	Motion
EP	Epilepsy	3	206	137	4	138	HAR
EC	EthanolConcentration	3	1751	261	4	263	HAR
ER	ERing	4	65	30	6	270	Other
FD	FaceDetection	144	62	5890	2	3524	EEG/MEG
FM	FingerMovements	28	50	316	2	100	EEG/MEG
HMD	HandMovementDirection	10	400	160	4	74	EEG/MEG
HW	Handwriting	3	152	150	26	850	HAR
HB	Heartbeat	61	405	204	2	205	AS
IW	InsectWingbeat	200	30	30000	10	20000	AS
JV	JapaneseVowels	12	29	270	9	370	AS
LIB	Libras	2	45	180	15	180	HAR
LSST	LSST	6	36	2459	14	2466	Other
MI	MotorImagery	64	3000	278	2	100	EEG/MEG
NATO	NATOPS	24	51	180	6	180	HAR
PD	PenDigits	2	8	7494	10	3498	EEG/MEG
PEMS	PEMS-SF	963	144	267	7	173	EEG/MEG
PS	PhonemeSpectra	11	217	3315	39	3353	AS
RS	RacketSports	6	30	151	4	152	HAR
SRS1	SelfRegulationSCP1	6	896	268	2	293	EEG/MEG
SRS2	SelfRegulationSCP2	7	1152	200	2	180	EEG/MEG
SAD	SpokenArabicDigits	13	93	6599	10	2199	AS
SWJ	StandWalkJump	4	2500	12	3	15	ECG
UW	UWaveGestureLibrary	3	315	120	8	320	HAR

based approach. $O^j_{i,MRN}$ is defined in Eq. (2).

$$O^j_{i,MRN} = GeLU(O^j_{i,CAF} + O^j_{i,CN3}) \tag{2}$$

where,

$$O^j_{i,CAF} = Softmax(\frac{O^j_{i,CN1} \cdot (O^j_{i,CN2})^T}{\sqrt{d^{j,k}_{i,MRN}}}) \cdot O^j_{i,CN3} \tag{3}$$

where, $(O^j_{i,CN2})^T$ and $d^{j,k}_{i,MRN}$ stand for the transpose and dimension of $O^j_{i,CN2}$, respectively. $Softmax()$ outputs the mathematical possibilities of a give vector.

2) Multi-head attention: The multi-head attention network is composed of n_{att} attention modules designed to mine relationships among the features across diverse locations. In the j-th aggregation transformer block, attention module, $Attention^{j,k}_i$, $k = 1, 2, ..., n_{att}$, maps a query, $Query^{j,k}_i$, and a connection of key-value pairs, $Key^{j,k}_i$-$Value^{j,k}_i$, to an output, $O^{j,k}_{i,att}$. $O^{j,k}_{i,att}$ is defined as:

$$O^{j,k}_{i,att} = Softmax(\frac{Query^{j,k}_i \cdot (Key^{j,k}_i)^T}{\sqrt{d^{j,k}_{i,MHA}}}) \cdot Value^{j,k}_i \tag{4}$$

where, $(Key^{j,k}_i)^T$ and $d^{j,k}_{i,MHA}$ represent the transpose and dimension of $Key^{j,k}_i$, respectively.

Let $O^j_{i,MHA}$ denote the output feature vector of the multi-head attention network in the j-th aggregation transformer block. $O^j_{i,MHA}$ is calculated in Eq. (5).

$$O^j_{i,MHA} = CONCAT([O^{j,1}_{i,att}, O^{j,2}_{i,att}, ..., O^{j,n_{att}}_{i,att}]) \tag{5}$$

where, $CONCAT()$ is the concatenation function.

The final feature aggregation within each aggregation transformer block is realized through a structured fusion of local and global feature representations. Specifically, the local patterns, denoted as $O^j_{i,MRN}$, capture short-range dependencies and contextual semantics through a hierarchical convolutional process enhanced by internal attention mechanisms. In parallel, the global relationships captured by the multi-head attention output $O^j_{i,MHA}$ encode long-term temporal dependencies and inter-variable interactions across the entire time window.

To merge these complementary representations, we adopt an additive feature fusion strategy:

$$O^j_i = LN(GeLU(Dense(O^j_{i,MHA} + O^j_{i,MRN}))) \tag{6}$$

where, O^j_i represent the output feature vector of the j-th aggregation transformer block. $LN()$ and $Dense()$ present the layer normalization and fully-connected (i.e., dense) functions, respectively.

This formulation ensures that both modalities contribute jointly to the final feature space, while the dense layer performs dimensional alignment and nonlinear transformation. The subsequent GeLU activation introduces smooth nonlinearity, and the layer normalization stabilizes the learning process by preventing internal covariate shift.

As the aggregation transformer blocks are stacked hierarchically, the output of each block O^j_i becomes the input to the next, progressively enriching the feature hierarchy. Ultimately, the last aggregation block outputs a comprehensive representation that integrates multi-scale dependencies—both local and global—across time and variables. This final representation

Algorithm 1 Training and Inference Procedure of KATN

Input: $\mathcal{D} = (\mathcal{D}_{train}, \mathcal{D}_{val}, \mathcal{D}_{test})$; ▷ Training, validation, and test sets.

Output: $\hat{Y}$; ▷ Predicted labels produced by KATN.

Randomly initialize model parameters θ_0;

for $m = 1$ to EPS **do** ▷ EPS: number of training epochs

Begin Forward Pass:

Feed each training sample $\mathcal{X}_i \in \mathcal{D}_{train}$ through the KATN architecture (Fig. 1), consisting of four sequential aggregation transformer blocks.

for $j = 1$ to 4 **do** ▷ Each block performs local-global feature fusion

Obtain $O^j_{i,CN1}$, $O^j_{i,CN2}$, and $O^j_{i,CN3}$ via Eq. (1);

Compute attention-weighted local feature: $O^j_{i,CAF}$ using Eq. (3);

Fuse local output: $O^j_{i,MRN} = GeLU(O^j_{i,CAF} + O^j_{i,CN3})$ via Eq. (2);

For each head k, compute global attention: $O^{j,k}_{i,att}$ using Eq. (4);

Concatenate heads: $O^j_{i,MHA}$ via Eq. (5);

Aggregate local-global features: $O^j_i = LN(GeLU(Dense(O^j_{i,MHA} + O^j_{i,MRN})))$ via Eq. (6);

end for

End Forward Pass

Begin Loss Computation:

Using Eq. (7), evaluate the objective:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(O_i) + \epsilon\|\theta\|_2^2$$

End Loss Computation

Begin Parameter Update:

Perform gradient descent:

$$\theta_m = \theta_{m-1} - \eta\nabla_{\theta_{m-1}}\mathcal{L}(\theta_{m-1})$$

where η is the learning rate.

End Parameter Update

if $m > 1$ **then**

Evaluate model on $\mathcal{D}_{val}$ to monitor generalization.

end if

end for

Begin Inference:

Deploy trained model on $\mathcal{D}_{test}$ to generate final predictions $\hat{Y}$.

End Inference

is then passed to the classification head for downstream prediction.

C. Loss Function

The loss function of KATN, denoted as $\mathcal{L}$, is formulated using the cross-entropy method. It quantifies the disparity between the ground-truth labels and the predicted vectors, as expressed in Eq. (7).

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i log(O_i) + \epsilon||\theta||_2^2 \tag{7}$$

where, O_i stands for the i-th output feature vector of KATN. y_i is the i-th ground truth label. θ denotes the KATN's parameters. ϵ represents the coefficient of $||\theta||_2^2$ (i.e., L_2 regularization). Following [13], [16], we set ϵ = 0.0005 in this paper. The implementation details of the proposed KATN framework are formally outlined in Algorithm 1, which provides a step-by-step description of its computational workflow.

IV. Performance and Evaluation

This section begins by detailing the experimental setup, including performance metrics and ablation study analysis. Subsequently, it evaluates the performance and computational efficiency of KATN through comprehensive experiments. The section concludes with an in-depth representation visualization analysis to further illustrate the practical applicability of the proposed method.

A. Experimental Setup

1) Dataset Description: As suggested in [11], [13], [14], [32], [34], [40], we use the University of East Anglia (UEA) multivariate time series archive in 2018 [69] for algorithmic performance evaluation. This archive covers a range of categories, varying from 2 to 39, with time series lengths ranging from 8 to 17,984. It includes data from 7 application scenarios, such as human activity recognition and motion analysis. Table I shows more details about the UEA archive.

TABLE II
Hyper-parameter settings of four aggregation transformer blocks.

Aggregation Transformer No.	n_{att}	MResNet's Channels	Dense Layer's Units
1	8	128	128
2	8	128	128
3	16	256	256
4	16	256	256

2) Implementation Details: The hyper-parameters for the four aggregation transformer blocks are summarized in Table II. For optimization, we employ the Adam Optimizer with the following configurations: a momentum term of 0.95, an initial learning rate of 0.001, and a decay value of 0.9. All experiments are performed on a computational platform running Ubuntu 18.04 OS and Python 3.6, powered by an Nvidia RTX 2080Ti GPU, Tensorflow 1.18, and an AMD R5 1400 CPU with 32GB RAM. To illustrate the training dynamics of KATN, Fig. 3 visualizes the loss curves across multiple datasets, including CharacterTrajectories, Cricket, FaceDetection, Heartbeat, LSST, PenDigits, PhonemeSpectra, SpokenArabicDigits, and StandWalkJump.

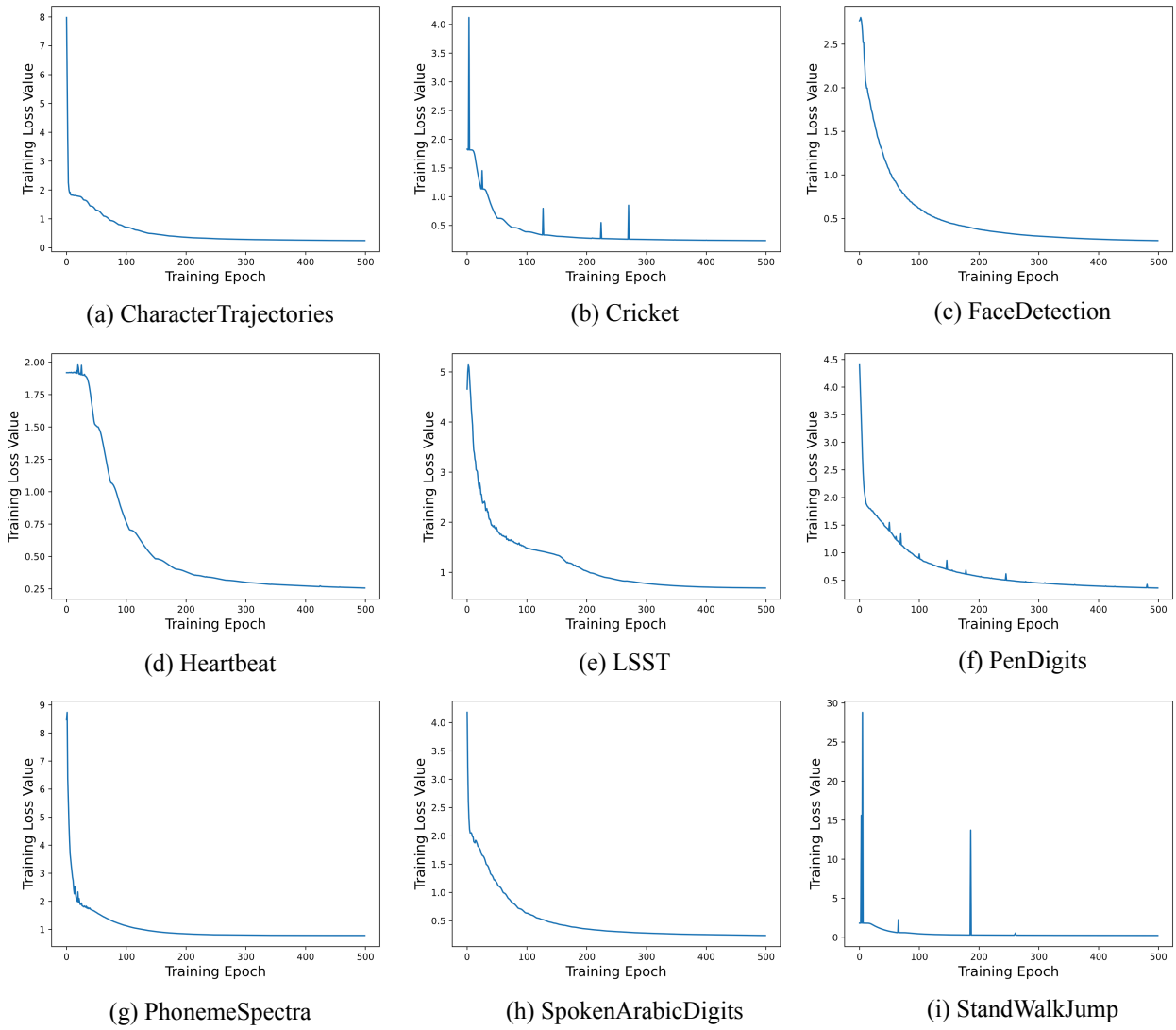


Fig. 3. Training loss values obtained during training on the CharacterTrajectories, Cricket, FaceDetection, Heartbeat, LSST, PenDigits, PhonemeSpectra, SpokenArabicDigits, and StandWalkJump datasets.

B. Performance Metrics

To validate the effectiveness of the proposed KATN, two commonly used metrics, 'win'/'tie'/'lose' and AVG_rank, both based on the top-1 accuracy, are considered. As suggested in [6], [7], [10], [11], [13], [14], [46], for any MTSC algorithm, the 'win', 'tie', and 'lose' scores indicate on how many datasets an algorithm performs better than, equivalent to, or worse than compared algorithms, respectively. The 'best' score represents the sum of the 'win' and 'tie' scores of the algorithm. In addition, similar to prior studies in [6], [7], [13], [14], [15], [16], [46], [57], [59], we use AVG_rank, which distinguishes between different algorithms based on the Wilcoxon signed-rank test with Holm's alpha correction at a significance level of 5%.

C. Ablation Study

To comprehensively validate the design choices of the proposed KATN model, we conduct an integrated ablation study encompassing both hyper-parameter sensitivity analysis and architectural component dissection. This dual-perspective analysis enables a rigorous examination of KATN's robustness and design effectiveness across 30 UEA benchmark MTSC datasets. Specifically, we partition the study into two subparts: (1) hyper-parameter sensitivity, focusing on structural configuration choices; and (2) architectural component analysis, isolating key modules to quantify their individual contributions.

1) Hyper-parameter Sensitivity Analysis: We begin by evaluating the impact of critical structural hyper-parameters that were systematically optimized during model development. This includes (i) the number of aggregation transformer blocks, and (ii) the activation function employed within each block. These parameters fundamentally affect the model's representation capacity, learning dynamics, and overall expressivity.

a) Effect of aggregation transformer depth: The number of aggregation transformer blocks determines the hierarchical

TABLE III
TOP-1 ACCURACY RESULTS WITH DIFFERENT AGGREGATION TRANSFORMER BLOCKS ACROSS 30 UEA BENCHMARK DATASETS.

Dataset Index	KATN-(1)	KATN-(2)	KATN-(3)	KATN	KATN-(5)
AWR	0.953	0.970	**0.993**	**0.993**	**0.993**
AF	0.333	0.400	0.467	**0.533**	**0.533**
BM	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
CT	0.741	0.935	0.951	**0.993**	0.986
CK	0.917	0.958	0.986	**1.000**	**1.000**
DDG	0.600	0.560	0.600	0.660	**0.680**
EW	0.527	0.727	0.684	**0.733**	**0.733**
EP	0.935	0.935	0.986	**0.913**	**0.913**
EC	0.321	0.349	0.372	0.399	**0.411**
ER	0.778	0.822	0.907	0.956	**0.968**
FD	0.518	0.528	0.614	**0.664**	**0.664**
FM	0.500	0.570	0.590	**0.620**	0.610
HMD	0.270	0.392	0.649	**0.688**	**0.688**
HW	0.192	0.294	0.287	**0.320**	**0.320**
HB	0.724	0.717	0.761	**0.762**	**0.762**
IW	0.228	0.302	0.491	0.515	**0.567**
JV	0.941	0.949	**0.978**	0.968	0.968
LIB	0.478	0.744	0.772	**0.839**	**0.839**
LSST	0.337	0.456	**0.652**	0.407	0.407
MI	0.530	0.600	0.580	0.620	**0.630**
NATO	0.850	0.889	**0.900**	0.889	0.889
PD	0.970	0.977	0.977	**0.982**	**0.982**
PEMS	0.832	0.888	0.939	**0.953**	**0.953**
PS	0.288	0.325	**0.466**	0.366	**0.466**
RS	0.796	0.829	**0.914**	0.875	0.875
SRS1	0.805	0.805	0.839	**0.908**	**0.908**
SRS2	0.511	0.550	0.561	**0.600**	**0.600**
SAD	0.872	0.953	0.963	**0.983**	**0.983**
SWJ	0.267	0.400	0.426	**0.600**	**0.600**
UW	0.759	0.868	0.868	0.881	**0.897**
Win	0	0	4	2	**7**
Tie	1	1	3	**17**	**17**
Lose	29	29	23	11	**6**
Best	1	1	7	19	**24**
Mean Accuracy	0.626	0.690	0.739	0.754	**0.761**

TABLE IV
TOP-1 ACCURACY RESULTS WITH DIFFERENT ACTIVATION FUNCTIONS ACROSS 30 UEA BENCHMARK DATASETS.

Dataset Index	ReLU	Leaky ReLU	EReLU	PReLU	GeLU
AWR	0.987	**0.993**	**0.993**	0.987	**0.993**
AF	0.400	0.467	0.467	0.467	**0.533**
BM	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
CT	0.979	0.983	**0.997**	0.983	0.993
CK	0.986	**1.000**	0.986	0.993	**1.000**
DDG	0.620	0.640	0.640	0.600	**0.660**
EW	0.712	0.712	**0.733**	0.727	**0.733**
EP	0.935	0.935	0.935	**0.952**	0.913
EC	0.349	0.352	0.352	0.372	**0.399**
ER	0.933	0.933	0.933	0.919	**0.956**
FD	0.631	0.614	0.634	**0.664**	**0.664**
FM	0.590	0.600	0.600	0.610	**0.620**
HMD	0.662	0.662	0.650	0.650	**0.688**
HW	0.308	0.310	**0.320**	**0.320**	**0.320**
HB	0.727	0.771	**0.765**	0.724	0.762
IW	0.491	**0.515**	**0.515**	**0.515**	**0.515**
JV	0.941	**0.978**	0.965	0.941	0.968
LIB	0.833	0.833	0.833	**0.850**	0.839
LSST	0.389	0.416	0.407	**0.421**	0.407
MI	0.590	0.610	**0.630**	0.610	0.620
NATO	**0.900**	0.872	0.872	0.872	0.889
PD	0.977	0.980	0.980	0.981	**0.982**
PEMS	0.930	0.930	**0.953**	**0.953**	**0.953**
PS	0.342	0.339	0.359	0.315	**0.366**
RS	0.856	0.856	0.868	0.803	**0.875**
SRS1	0.874	0.889	0.841	0.853	**0.908**
SRS2	0.533	0.533	0.533	0.500	**0.600**
SAD	0.972	0.963	0.978	0.980	**0.983**
SWJ	0.427	0.400	0.533	0.400	**0.600**
UW	0.869	0.859	0.903	**0.881**	**0.881**
Win	1	1	3	3	**13**
Tie	1	4	6	6	**9**
Lose	28	25	21	21	**8**
Best	2	5	9	9	**22**
Mean Accuracy	0.725	0.731	0.739	0.728	**0.754**

depth at which multi-scale temporal and cross-view dependencies are extracted. To investigate its effect, we construct five model variants with increasing aggregation depth:

- **KATN-(1)**: employs a single aggregation transformer block.
- **KATN-(2)**: integrates two aggregation transformer blocks.
- **KATN-(3)**: integrates three aggregation transformer blocks.
- **KATN**: the default version of our model, comprising four aggregation transformer blocks.
- **KATN-(5)**: extends the model to five aggregation transformer blocks.

The experimental results, summarized in Table III, reveal a consistent trend: increasing the number of transformer blocks yields progressively better classification accuracy, particularly for complex datasets such as InsectWingbeat. For instance, KATN-(5) achieves the highest accuracy across several challenging benchmarks, confirming the benefit of deeper aggregation in modeling intricate temporal patterns.

Nevertheless, the performance gain between KATN-(5) and the default KATN is marginal, with a mean accuracy improvement of only 0.007 across the 30 UEA datasets. Furthermore, the parameter count increases substantially from 3.368M (KATN) to 4.814M (KATN-(5)) on InsectWingbeat. This observation reveals a point of diminishing returns, wherein additional blocks yield negligible performance improvements while incurring higher computational costs. Thus, the default configuration with four aggregation transformer blocks is adopted as it offers a desirable balance between accuracy and efficiency.

b) Effect of activation function choice: Activation functions regulate non-linear transformations within each transformer block, thus playing a pivotal role in determining the expressive capacity of the network. To evaluate their effect, we test KATN under five widely used activation functions: ReLU, Leaky ReLU, Elastic ReLU (EReLU), Parametric ReLU (PReLU), and GeLU. The performance results across the 30 UEA datasets are presented in Table IV.

Among the candidates, GeLU exhibits the most stable and superior performance, with a 'win'/'tie'/'lose' count of 13/9/8 and a mean accuracy of 0.754. Its stochastic, smooth activation profile enhances gradient flow and enables finer feature selectivity—traits particularly valuable for modeling highly dynamic multivariate sequences. Consequently, GeLU is adopted as the default activation function in the final model architecture.

c) Hyper-parameter synthesis and interpretation: Taken together, the above analyses suggest that KATN exhibits high resilience to architectural variations, maintaining competitive performance across a spectrum of design choices. Although incremental increases in depth and activation sophistication do contribute to performance gains, the improvements quickly plateau as architectural complexity grows. This observation underscores KATN's structural efficiency: its default configuration—comprising four aggregation transformer blocks

TABLE V
TOP-1 ACCURACY RESULTS OBTAINED BY VARIOUS KATN VARIANTS ACROSS 30 UEA BENCHMARK DATASETS.

Dataset Index	KATN-w/o-MResNet	KATN-w/o-MHA	KATN-w/o-FC	KATN-MaMul	KATN-CONCAT	KATN -INTERN	KATN
AWR	0.973	0.970	0.980	0.953	**0.993**	0.992	**0.993**
AF	0.467	0.400	0.500	0.200	**0.533**	0.467	**0.533**
BM	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
CT	0.986	0.964	0.969	0.931	0.986	0.986	**0.993**
CK	0.943	0.944	0.986	0.958	0.951	0.986	**1.000**
DDG	0.520	0.540	0.600	0.600	0.560	0.620	**0.660**
EW	0.618	0.549	0.727	0.712	0.811	**0.847**	0.733
EP	0.807	0.666	0.935	**1.000**	0.979	**1.000**	0.913
EC	0.332	0.293	0.228	**0.487**	0.372	0.380	0.399
ER	0.941	0.930	0.930	0.859	0.933	**0.981**	0.956
FD	0.519	0.519	0.631	0.508	0.631	0.631	0.664
FM	0.560	0.550	0.550	0.570	**0.630**	0.450	0.620
HMD	0.338	0.278	0.378	0.312	0.662	0.392	**0.688**
HW	0.451	0.382	0.316	0.308	**0.511**	0.296	0.320
HB	0.619	0.619	0.727	0.518	0.765	**0.771**	0.762
IW	0.327	0.128	0.100	0.491	0.362	**0.595**	0.515
JV	0.918	0.924	0.949	0.941	0.930	**0.989**	0.968
LIB	0.889	0.833	0.817	0.806	**0.900**	0.878	0.839
LSST	**0.548**	0.456	0.337	0.265	0.347	0.456	0.407
MI	0.510	0.510	0.530	0.550	0.570	0.550	**0.620**
NATO	0.822	0.850	**0.900**	0.839	0.872	0.850	0.889
PD	0.939	0.973	0.969	0.831	0.948	0.965	**0.982**
PEMS	0.874	0.705	0.959	**0.994**	0.930	0.522	0.953
PS	0.215	0.104	0.247	0.269	0.325	0.292	**0.366**
RS	0.796	0.868	0.809	0.823	**0.879**	0.868	0.875
SRS1	0.843	0.771	0.874	0.724	**0.913**	0.874	0.908
SRS2	0.533	0.483	0.500	0.494	0.533	0.522	**0.600**
SAD	0.963	0.967	0.979	0.959	0.963	0.729	**0.983**
SWJ	0.426	0.200	0.400	0.267	0.533	0.333	**0.600**
UW	0.872	0.881	0.833	0.684	0.897	**0.916**	0.881
Win	1	0	1	2	5	6	**10**
Tie	1	1	1	2	**3**	2	**3**
Lose	28	29	28	26	22	22	**17**
Best	2	1	2	4	8	8	**13**
AVG_rank	4.833	5.383	4.283	5.183	2.867	3.333	**2.117**

and GeLU activation—not only delivers a robust accuracy-complexity trade-off but also demonstrates the model's scalability and generalization stability under diverse task conditions.

2) Architectural Component Analysis: Beyond hyperparameters, we assess the significance of core architectural modules through controlled ablations. This includes the multi-head attention mechanism, MResNet backbone, the fully connected projection layer, and the feature aggregation strategy.

a) Impact of multi-head attention and MResNet: We begin by evaluating the respective contributions of the multi-head attention module and the MResNet backbone. To this end, we examine two reduced variants of KATN:

- KATN-w/o-MHA: a variant in which the multi-head attention mechanism is removed.
- KATN-w/o-MResNet: a variant excluding the MResNet component.

The performance comparison, presented in Table V, reveals that the complete KATN architecture surpasses KATN-w/o-MHA on 27 datasets and outperforms KATN-w/o-MResNet on 26. These results highlight the complementary roles of global attention and local convolution in capturing temporal dynamics across varying scales. The attention mechanism enables the model to uncover long-range dependencies that are often elusive to convolutional filters, whereas MResNet preserves local continuity essential for fine-grained pattern recognition.

Despite the gains in accuracy, these modules introduce distinct computational costs. As summarized in Table VI, the average CPU inference time across the 30 UEA benchmark testing datasets increases from 90.773s (without MHA) and 75.423s (without MResNet) to 106.620s in the full model. This increase, however, is offset by substantial performance benefits, illustrating that the integration of attention and residual learning yields a synergistic effect critical to robust MTSC.

b) Effectiveness of fully connected layer: To assess the necessity of the fully connected (dense) layer employed within each aggregation transformer block, we introduce a variant termed KATN-w/o-FC, in which this layer is eliminated. The fully connected layer serves to harmonize the latent spaces of MResNet and the attention mechanism by projecting heterogeneous features into a unified representation space.

As indicated in Table V, the absence of the fully connected layer results in degraded performance on 28 out of 30 datasets, demonstrating that direct feature fusion without alignment significantly impairs the model's ability to integrate local and global information. In particular, the fully connected layer mitigates semantic incompatibility between disparate feature sources, thereby enabling coherent and discriminative joint representations.

The computational overhead of incorporating the fully connected layer is minimal. Based on evaluations on the 30 UEA benchmark testing datasets in Table VI, the average CPU inference time increases marginally from 101.716s to 106.620s. Given its role in facilitating effective cross-modal integration, this cost is well-justified by the substantial accuracy gains.

c) Effectiveness of additive feature aggregation: To investigate the impact of the feature fusion strategy within each aggregation block, we compare KATN with three alternative aggregation designs:

- KATN-MatMul: KATN with multiplicative feature aggregation, as shown in Fig. 4 (a).
- KATN-CONCAT: KATN with concatenation feature aggregation, as depicted in Fig. 4 (b).
- KATN-INTERN: KATN with interactive feature aggregation, as detailed in Fig. 4 (c).

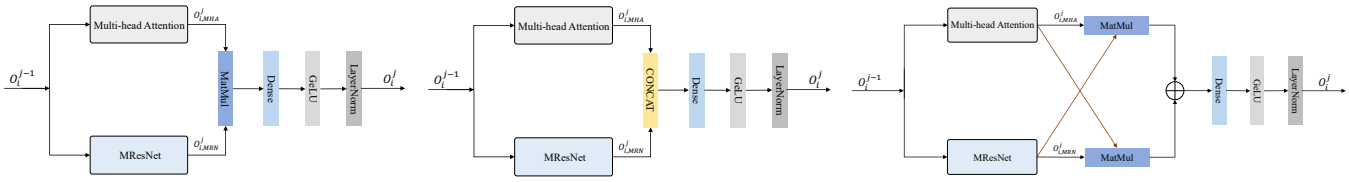
The quantitative results in Table V demonstrate that KATN, utilizing additive aggregation, achieves superior accuracy on 26, 18, and 20 datasets when compared to KATN-MatMul, KATN-CONCAT, and KATN-INTERN, respectively. These outcomes underline the advantage of additive fusion, which offers a lightweight yet semantically balanced integration of local and global descriptors without exacerbating feature dimensionality or causing representational entanglement.

The efficiency comparisons on the 30 UEA benchmark testing datasets further reinforce the merits of this design. According to Table VI, KATN records an average CPU inference time of 106.620s, whereas KATN-MatMul and KATN-CONCAT require 108.602s and 111.526s on CPU, respectively. Notably, KATN-INTERN incurs a prohibitive cost of 663.790s due to its intricate feature interaction operations. These findings suggest that although alternative schemes may enable more complex representation interactions, they do so at the expense of efficiency and, in many cases, accuracy.

Moreover, while concatenation increases feature dimensionality and often causes dominance of one representation stream over another, multiplicative and interaction-based approaches

TABLE VI
COMPARATIVE EVALUATION OF VARIOUS KATN VARIANTS: PARAMETERS, FLOATING POINT OPERATIONS (FLOPS), AND INFERENCE TIME ACROSS 30 UEA BENCHMARK TESTING DATASETS.

Dataset Index	Parameters (M)							FLOPs (M)							With GPU (s)							With CPU (s)						
	KATN-w/o-MResNet	KATN-w/o-MHA	KATN-w/o-FC	KATN-MaMul	KATN-CONCAT	KATN-INTERN	KATN	KATN-w/o-MResNet	KATN-w/o-MHA	KATN-w/o-FC	KATN-MaMul	KATN-CONCAT	KATN-INTERN	KATN	KATN-w/o-MResNet	KATN-w/o-MHA	KATN-w/o-FC	KATN-MaMul	KATN-CONCAT	KATN-INTERN	KATN	KATN-w/o-MResNet	KATN-w/o-MHA	KATN-w/o-FC	KATN-MaMul	KATN-CONCAT	KATN-INTERN	KATN
AWR	**1.556**	3.447	3.749	3.915	4.079	4.243	3.915	**3.098**	6.893	7.492	7.820	8.147	8.475	7.820	**0.837**	1.737	2.129	2.514	2.312	2.307	2.365	**4.148**	4.345	6.675	7.263	7.807	8.258	7.288
AF	**1.122**	3.009	3.307	3.473	3.637	3.801	3.473	**2.230**	6.017	6.608	6.936	7.264	7.591	6.936	**0.899**	1.617	2.096	2.410	2.240	2.217	2.303	**3.134**	1.465	4.012	4.259	4.412	4.487	4.294
BM	**0.735**	2.625	2.925	3.091	3.255	3.419	3.091	**1.456**	5.248	5.843	6.171	6.499	6.827	6.171	**1.012**	1.800	2.197	2.527	2.346	2.306	2.287	**0.783**	0.817	1.245	1.450	1.494	1.557	1.465
CT	**1.563**	3.451	3.749	3.915	4.079	4.243	3.915	**3.112**	6.899	7.492	7.820	8.148	8.475	7.820	**1.355**	2.082	2.842	3.242	3.144	3.141	3.059	24.919	**23.803**	38.671	41.984	45.669	47.082	42.040
CK	**4.310**	6.199	6.500	6.666	6.830	6.993	6.666	**8.606**	12.397	12.993	13.321	13.648	13.976	13.321	2.038	**1.860**	3.263	3.711	3.487	3.658	3.472	40.906	**10.411**	47.877	48.711	50.090	50.455	48.801
DDG	**1.664**	4.410	5.396	5.562	5.726	5.890	5.562	**3.314**	8.819	10.786	11.114	11.441	11.769	11.114	**0.869**	1.680	2.091	2.459	2.326	2.240	2.216	2.393	**2.131**	3.693	4.007	4.132	4.299	4.007
EW	**23.652**	25.542	25.842	26.008	26.172	26.336	26.008	**47.290**	51.082	51.678	52.005	52.333	52.661	52.005	**27.061**	28.061	30.237	30.248	31.299	32.342	30.242	**1652.679**	1683.679	1655.823	1765.859	1805.226	18335.343	1714.426
EP	**0.842**	2.730	3.028	3.195	3.358	3.522	3.195	**1.670**	5.458	6.051	6.378	6.706	7.034	6.378	**0.823**	1.693	2.067	2.408	2.234	2.256	2.228	3.423	**3.089**	5.215	5.653	5.996	6.265	5.731
EC	**2.424**	4.312	4.610	4.777	4.940	5.104	4.777	**4.834**	8.622	9.215	9.542	9.870	10.198	9.542	8.502	**3.435**	10.442	11.328	10.886	11.285	10.936	**63.831**	319.303	364.998	370.722	371.831	379.556	365.158
ER	**0.731**	2.620	2.919	3.085	3.249	3.413	3.085	**1.449**	5.238	5.832	6.159	6.487	6.815	6.159	**0.773**	1.646	1.968	2.343	2.219	2.164	2.143	**1.537**	1.960	2.692	3.018	3.263	3.405	3.058
FD	**0.735**	2.713	3.084	3.250	3.414	3.578	3.250	**1.456**	5.424	6.161	6.489	6.817	7.144	6.489	**1.162**	2.592	3.077	3.542	3.382	3.376	3.294	**12.177**	18.271	23.343	26.309	28.141	30.402	26.189
FM	**0.669**	2.573	2.885	3.051	3.215	3.378	3.051	**1.325**	5.145	5.763	6.091	6.418	6.746	6.091	**0.778**	1.928	2.220	2.633	2.416	2.402	2.399	**0.795**	0.872	1.281	1.518	1.527	1.605	1.492
HMD	**1.044**	2.936	3.239	3.405	3.569	3.732	3.405	**2.075**	5.871	6.471	6.799	7.127	7.454	6.799	**0.915**	1.694	2.193	2.586	2.305	2.337	2.389	5.362	**3.156**	7.085	7.614	7.860	8.511	7.488
HW	**1.643**	3.530	3.829	3.995	4.159	4.323	3.995	**3.272**	7.059	7.652	7.980	8.307	8.635	7.980	**1.029**	1.809	2.334	2.698	2.565	2.608	2.518	11.346	**11.713**	18.117	19.785	21.156	22.663	19.785
HB	**0.868**	2.793	3.121	3.287	3.451	3.615	3.287	**1.722**	5.584	6.236	6.564	6.892	7.219	6.564	**1.163**	1.813	2.400	2.901	2.611	2.760	2.626	14.146	**8.062**	18.446	19.470	20.468	21.259	19.443
IW	**0.788**	2.802	3.202	3.368	3.532	3.695	3.368	**1.563**	5.602	6.397	6.725	7.052	7.380	6.725	**1.823**	2.976	3.604	4.147	4.086	4.142	3.973	**22.839**	45.243	52.325	59.520	64.721	70.945	58.800
JV	**0.702**	2.596	2.899	3.065	3.229	3.393	3.065	**1.391**	5.190	5.792	6.120	6.448	6.775	6.120	**0.788**	1.893	2.247	2.597	2.415	2.403	2.418	**0.991**	1.300	1.781	2.044	2.139	2.251	2.038
LIB	**0.803**	2.690	2.989	3.155	3.319	3.482	3.155	**1.593**	5.379	5.971	6.299	6.626	6.954	6.299	**0.777**	2.085	2.397	2.834	2.647	2.551	2.594	**0.924**	1.131	1.595	1.785	1.925	1.972	1.793
LSST	**0.762**	2.651	2.951	3.118	3.281	3.445	3.118	**1.509**	5.301	5.897	6.224	6.552	6.880	6.224	**0.999**	2.417	2.777	3.155	3.003	3.043	3.021	**4.543**	7.126	8.731	9.755	10.636	11.665	9.739
MI	**2.198**	4.125	4.455	4.621	4.785	4.949	4.621	**4.383**	8.248	8.904	9.231	9.559	9.887	9.231	9.813	**3.105**	11.633	12.403	12.008	12.098	11.979	**55.249**	364.838	400.940	401.547	411.080	411.922	401.517
NATO	**0.720**	2.621	2.931	3.097	3.261	3.425	3.097	**1.426**	5.241	5.855	6.183	6.511	6.838	6.183	**0.824**	2.133	2.728	2.770	2.681	2.596	2.583	**0.981**	1.210	1.703	1.964	2.039	2.183	1.964
PD	**0.651**	2.538	2.836	3.002	3.166	3.330	3.002	**1.288**	5.074	5.666	5.994	6.322	6.649	5.994	**0.863**	1.760	2.120	2.526	2.296	2.326	2.289	**1.451**	2.561	3.074	3.503	3.734	4.107	3.542
PEMS	**1.381**	3.883	4.673	4.839	5.003	5.167	4.839	**2.747**	7.764	9.340	9.667	9.995	10.323	9.667	**0.959**	1.794	2.170	2.592	2.384	2.354	2.358	**2.722**	3.221	4.771	5.215	5.399	5.655	5.195
PS	**2.802**	4.694	4.997	5.163	5.327	5.491	5.163	**5.590**	9.387	9.988	10.316	10.644	10.971	10.316	**2.604**	2.856	4.417	5.121	5.113	5.113	4.833	76.478	114.812	**65.669**	122.520	131.358	138.352	122.553
RS	**0.663**	2.553	2.853	3.019	3.183	3.347	3.019	**1.313**	5.104	5.700	6.028	6.355	6.683	6.028	**0.771**	1.840	2.157	2.582	2.290	2.331	2.303	**0.715**	0.817	1.236	1.406	1.471	1.554	1.406
SRS1	**1.091**	2.981	3.281	3.447	3.611	3.775	3.447	**2.169**	5.960	6.556	6.884	7.212	7.539	6.884	3.069	**2.302**	4.584	5.116	4.876	4.946	4.816	93.561	**28.337**	113.075	115.190	118.958	121.737	115.397
SRS2	**1.223**	3.113	3.414	3.580	3.744	3.908	3.580	**2.432**	6.225	6.822	7.149	7.477	7.805	7.149	3.196	**2.165**	4.643	5.109	4.903	4.927	4.879	93.087	**24.760**	109.694	113.140	114.206	115.660	111.667
SAD	**0.874**	2.768	3.072	3.238	3.402	3.566	3.238	**1.735**	5.535	6.138	6.466	6.794	7.121	6.466	**1.114**	2.550	2.696	3.731	3.026	3.281	2.958	**13.828**	18.121	24.018	27.228	31.094	32.038	26.956
SWJ	**2.552**	4.440	4.739	4.905	5.069	5.233	4.905	**5.089**	8.878	9.472	9.800	10.127	10.455	9.800	2.315	**1.815**	3.593	4.046	3.935	3.769	3.755	39.236	**6.520**	43.599	43.812	45.386	44.418	44.005
UW	**1.276**	3.164	3.463	3.629	3.793	3.956	3.629	**2.539**	6.326	6.919	7.247	7.574	7.902	7.247	**1.138**	1.890	2.507	2.892	2.814	2.726	2.650	14.513	**10.104**	20.085	21.805	22.574	24.085	21.352
Average Value	N/A							N/A							**2.676**	2.968	4.194	4.639	4.475	4.533	4.396	**75.423**	90.773	101.716	108.602	111.526	663.790	106.620



(a) KATN-MatMul (b) KATN-CONCAT (c) KATN-INTERN

Fig. 4. Differences between three feature aggregation methods.

introduce nonlinear complexity that may impair stable gradient propagation and semantic interpretability. In contrast, the additive scheme promotes interpretable and structurally simple integration, preserving the intrinsic characteristics of each feature type while enabling effective temporal reasoning.

3) Summary of Ablation Insights: The ablation findings collectively demonstrate that KATN's performance is underpinned by both careful hyper-parameter selection and well-integrated architectural components. The depth and activation function directly affect representation richness and convergence dynamics, while attention, residual learning, and projection layers jointly support hierarchical temporal modeling. Feature fusion via additive integration further balances complexity and expressivity. Overall, the default KATN configuration reflects a principled balance between empirical performance and architectural parsimony.

D. Experimental Analysis

This section studies the performance of KATN on 30 UEA datasets.

1) Comparison with SOTA transformer algorithms: To explore the performance of KATN, we compare it with 6 SOTA transformer algorithms, including:

- VanTrans: the vanilla transformer network adapted to MTSC [61].
- SwinTrans: the hierarchical transformer network using shifted windows adapted to MTSC tasks [63].
- Linformer: the transformer network based on line attention adapted to MTSC tasks [41].
- FMLA: the transformer network using flexible multi-head linear attention for MTSC [44].
- DualAggTrans: the dual aggregation transformer network adapted to MTSC tasks [25].
- DualAttTrans: the dual attention transformer adapted to MTSC tasks [64].

Table VII shows the top-1 accuracy results obtained by various transformer algorithms on 30 UEA benchmark datasets. One can easily find that KATN outperforms 6 SOTA transformer algorithms since it achieves the highest 'win'/'tie'/'lose' score, namely 9/6/15, and the lowest AVG_rank value, namely 2.400. This is because KATN deeply aggregates various feature types within each level, mining abundant local features and global relations in the given multivariate time series.

TABLE VII
TOP-1 ACCURACY RESULTS OBTAINED BY VARIOUS TRANSFORMER VARIANTS ACROSS 30 UEA BENCHMARK DATASETS.

Dataset Index	VanTrans	SwinTrans	Linformer	FMLA	DualAggTrans	DualAttTrans	KATN
AWR	0.973	0.980	0.987	0.992	**0.993**	**0.993**	**0.993**
AF	0.467	0.467	0.467	**0.533**	0.467	0.467	**0.533**
BM	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
CT	0.966	0.970	**0.997**	0.979	0.989	0.986	0.993
CK	0.943	0.958	0.958	0.986	0.986	0.951	**1.000**
DDG	0.540	0.580	0.600	0.620	**0.660**	0.640	**0.660**
EW	0.618	0.847	0.489	**0.954**	0.733	0.628	0.733
EP	0.807	0.978	0.971	0.979	0.935	**1.000**	0.913
EC	0.332	0.228	0.323	0.380	0.352	0.372	**0.399**
ER	0.400	0.930	0.859	**0.981**	0.933	0.933	0.956
FD	0.519	0.583	0.556	0.631	0.634	0.631	**0.664**
FM	0.560	0.460	0.530	0.550	0.600	**0.630**	0.620
HMD	0.338	0.284	0.378	0.392	0.650	0.662	**0.688**
HW	0.451	0.187	0.357	**0.511**	0.320	0.294	0.320
HB	0.619	0.727	0.751	**0.771**	0.765	0.765	0.762
IW	0.327	0.100	0.208	0.362	0.515	**0.595**	0.515
JV	0.918	0.778	0.965	0.965	**0.989**	0.930	0.968
LIB	0.889	0.817	0.850	0.850	**0.878**	0.833	0.839
LSST	0.548	**0.652**	0.568	0.643	0.407	0.347	0.407
MI	0.510	0.500	0.590	0.550	**0.630**	0.620	0.620
NATO	0.822	0.800	0.850	0.850	0.850	0.872	**0.889**
PD	0.939	0.939	0.980	0.965	0.980	0.948	**0.982**
PEMS	0.874	**0.959**	0.751	0.522	0.953	0.930	0.953
PS	0.215	0.247	0.175	0.292	0.359	0.325	**0.366**
RS	0.796	0.809	0.868	0.868	0.823	0.829	**0.875**
SRS1	0.843	0.771	0.652	0.874	0.841	**0.913**	0.908
SRS2	0.533	0.450	0.550	0.522	0.533	**0.600**	**0.600**
SAD	0.963	0.979	**0.983**	0.959	0.978	0.963	**0.983**
SWJ	0.426	0.267	0.400	0.333	0.533	0.533	**0.600**
UW	0.872	0.728	0.894	**0.916**	0.903	0.897	0.881
Win	0	2	1	5	3	3	**9**
Tie	1	1	2	2	3	4	**6**
Lose	29	27	27	23	24	23	**15**
Best	1	3	3	7	6	7	**15**
AVG_rank	5.483	5.500	4.500	3.467	3.167	3.483	**2.400**

2) Comparison with existing MTSC algorithms: To examine the performance of KATN, we compare it with 18 existing MTSC algorithms, listed below.

- MLP: the multi-layer perceptron model applied to MTSC tasks [47].
- FCN: the fully convolutional network model applied to MTSC tasks [47].
- ResNet: the residual neural network model applied to MTSC tasks [47].
- InceptionTime: the neural network model based on Inception blocks applied to MTSC tasks [45].
- Three distance-based baseline algorithms: ED_I, DTW_I, and DTW_D [6].
- WM: the statistical feature selection approach based on bag-of-pattern (WEASEL+MUSE) [42].
- CBOSS: the contractable approximation symbols method using bag of symbolic Fourier [41].
- MLCN: the multivariate LSTM-FCN [14].
- TSF: the time series forest approach for MTSC [38].
- TapNet: the attentional prototype network model that gracefully embeds traditional classification and DL-based methods [59].
- XEM: the explainable-by-design ensemble method based on boosting-bagging and bias-variance [32].
- CMFM+SVM: the complexity measure-and-feature approach combined with SVM applied to MTSC tasks [34].
- MiniROCKET: the transform approach using fast deterministic technique, an improved version of ROCKET [56].
- DA-Net: the dual attention-based network model that deeply incorporates squeeze-excitation and sparse attention variants [11].
- Conv-GRU: the convolutional network model combined with gated linear unit structures [70].
- DKN: the densely knowledge-aware network [17].

Table VIII presents the top-1 accuracy results for the various MTSC algorithms evaluated. KATN outperforms all other algorithms, achieving a 'win'/'tie'/'loss' score of 6/7/17, and securing the lowest AVG_rank score of 4.167. DKN follows closely behind, ranking second in both the 'best' metric and AVG_rank. MiniROCKET ranks as the second-best algorithm according to the 'best' metric, while XEM achieves the third-best position in terms of AVG_rank. In contrast, MLP delivers the poorest performance among the algorithms tested.

The following provides an explanation for the results presented above. KATN effectively combines the local patterns extracted by MResNet and the global patterns extracted by the multi-head attention network, capturing long-range dependencies across multiple variables. DKN facilitates the transfer of knowledge between lower- and higher-level semantic information, effectively regularizing the model and contributing to its strong performance. MiniROCKET, on the other hand, employs simple linear classifiers alongside randomly initialized convolutional kernels to capture multi-scale subsequence features in time series, which enhances its efficiency in time series classification. XEM employs explicit boosting-bagging techniques and addresses the bias-variance trade-off, effectively uncovering potential correlations across different dimensions. In contrast, MLP, characterized by a shallow neural network structure, fail to mine the inherent relationships within the data, especially the interconnections among variables.

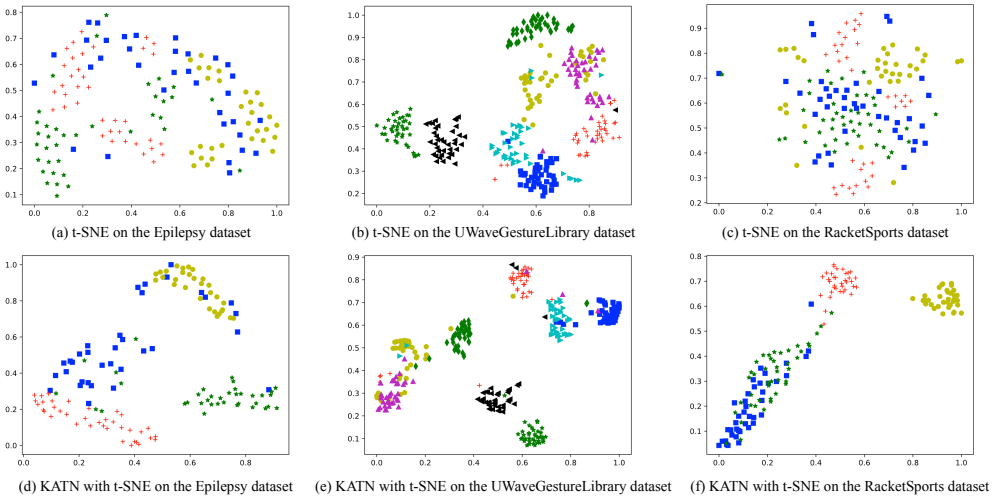
E. Computational Complexity Assessment

Based on the guidelines outlined in [71], we conduct a comprehensive comparison between the proposed KATN model and 13 representative DL algorithms in terms of parameter count, FLOPs, and inference time, using the 30 UEA benchmark testing datasets as the evaluation foundation. These comparative methods consist of six transformer-based models, namely VanTrans [61], SwinTrans [63], Linformer [41], FMLA [44], DualAggTrans [25], and DualAttTrans [64], as well as seven MTSC baseline models. The baseline group includes four single-network-based methods—InceptionTime [45], ResNet [47], MLP [47], and FCN [47]—and three dual-network-based architectures—TapNet [59], MLCN [14], and DKN [17]. The aggregated results of this comparison are summarized in Table IX.

A close examination reveals that transformer-based models generally impose substantially higher computational complexity. Across the 30 UEA benchmark testing datasets, most transformer variants exhibit larger parameter counts and higher FLOPs than the seven MTSC baseline models. The proposed KATN, while demonstrating robust representational capacity, presents a relatively high computational footprint on the majority of datasets, with only DualAggTrans and DualAttTrans

TABLE VIII
TOP-1 ACCURACY RESULTS OBTAINED BY VARIOUS MTSC ALGORITHMS ACROSS 30 UEA BENCHMARK DATASETS.

Dataset Index	MLP	FCN	ResNet	Inception Time	ED_I	DTW_D	DTW_I	CBOSS	WM	TSF	Mini ROCKET	XEM	TapNet	MLCN	CMFM + SVM	Conv-GRU	DA-Net	DKN	KATN
AWR	0.043	0.823	0.943	0.897	0.970	0.987	0.980	0.990	**0.993**	0.953	0.992	**0.993**	0.987	0.957	0.973	0.973	0.980	**0.993**	**0.993**
AF	0.400	0.200	0.200	0.267	0.267	0.220	0.267	0.267	0.267	0.200	0.133	0.467	0.333	0.333	0.267	0.467	0.467	0.467	**0.533**
BM	0.875	**1.000**	**1.000**	**1.000**	0.676	0.975	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.875	0.975	**1.000**	0.925	**1.000**	**1.000**
CT	0.056	0.741	0.977	0.935	0.964	0.989	0.969	0.986	0.990	0.931	0.065	0.979	0.997	0.917	0.970	0.966	**0.998**	0.986	0.993
CK	0.111	0.917	0.958	0.958	0.944	**1.000**	0.986	N/A	0.986	N/A	0.986	0.986	0.958	N/A	0.958	0.943	0.861	0.951	**1.000**
DDG	0.360	0.600	0.600	0.560	0.275	0.600	0.550	0.480	0.575	0.460	0.650	0.375	0.575	0.380	0.420	0.540	0.520	0.560	**0.660**
EW	0.233	0.684	0.833	0.727	0.549	0.618	N/A	0.511	0.890	0.712	**0.954**	0.527	0.489	0.330	0.847	0.811	0.489	0.628	0.733
EP	0.312	0.935	0.964	0.935	0.666	0.964	0.978	0.979	0.993	**1.000**	**1.000**	0.986	0.971	0.732	0.978	0.978	0.883	0.979	0.913
EC	0.300	0.349	0.317	0.321	0.293	0.323	0.304	0.304	0.316	**0.487**	0.380	0.372	0.323	0.373	0.228	0.332	0.338	0.372	0.399
ER	0.159	0.778	0.907	0.822	0.133	0.133	0.133	0.919	0.133	0.859	**0.981**	0.200	0.133	0.941	0.930	0.400	0.874	0.933	0.956
FD	0.565	0.518	0.534	0.528	0.519	0.529	N/A	0.513	0.545	0.508	0.631	0.614	0.556	0.555	0.583	0.640	0.648	0.631	**0.664**
FM	0.510	0.510	0.460	0.500	0.550	0.530	0.520	0.519	0.540	0.562	0.450	0.590	0.530	0.580	0.460	0.580	0.510	0.600	**0.620**
HMD	0.216	0.270	0.216	0.392	0.278	0.231	0.306	0.292	0.378	0.312	0.392	0.649	0.378	0.544	0.284	0.338	0.365	0.662	**0.688**
HW	0.038	0.192	0.382	0.294	0.200	0.286	0.316	0.504	**0.531**	0.191	0.511	0.287	0.357	0.305	0.187	0.451	0.159	0.231	0.320
HB	0.665	0.724	0.716	0.717	0.619	0.717	0.658	0.564	0.727	0.518	**0.771**	0.761	0.751	0.458	0.727	0.746	0.624	0.765	0.762
IW	0.104	0.491	0.231	0.302	0.128	N/A	N/A	N/A	N/A	N/A	**0.595**	0.228	0.208	N/A	0.100	0.208	0.567	0.362	0.515
JV	0.114	0.941	0.924	0.949	0.924	0.949	0.959	N/A	0.978	N/A	0.989	0.978	0.965	N/A	0.778	**0.991**	0.938	0.930	0.968
LIB	0.078	0.478	0.844	0.744	0.833	0.870	0.894	0.894	0.894	0.806	0.878	0.772	0.850	0.850	0.817	0.889	0.800	**0.900**	0.839
LSST	0.326	0.337	0.232	0.290	0.456	0.551	0.575	0.458	0.628	0.265	0.643	**0.652**	0.568	0.390	**0.652**	0.548	0.560	0.347	0.407
MI	0.530	0.590	0.560	0.530	0.510	N/A	N/A	0.390	0.500	0.550	0.550	0.600	0.590	0.510	0.500	0.512	0.500	**0.620**	**0.620**
NATO	0.167	0.900	0.878	0.889	0.850	0.883	0.850	0.850	0.883	0.839	0.928	0.916	**0.939**	0.900	0.800	0.916	0.878	0.872	0.889
PD	0.211	0.970	0.973	0.977	0.973	0.977	0.939	0.939	0.969	0.831	0.965	0.977	0.980	0.979	0.665	0.939	0.980	0.948	**0.982**
PEMS	0.340	0.832	0.828	0.888	0.705	0.711	0.734	0.730	N/A	**0.994**	0.522	0.942	0.751	0.745	0.959	0.874	0.867	0.930	0.953
PS	0.414	0.466	0.466	0.466	0.104	0.151	0.151	0.151	0.190	0.269	0.292	0.288	0.175	0.151	0.247	0.215	0.093	0.525	0.366
RS	0.276	0.796	0.836	0.829	0.868	0.803	0.842	0.854	0.914	0.823	0.868	**0.941**	0.868	0.856	0.809	0.888	0.803	0.879	0.875
SRS1	0.686	0.805	0.761	0.805	0.771	0.775	0.765	0.765	0.744	0.724	0.874	0.839	0.652	0.908	0.771	0.843	**0.924**	0.913	0.908
SRS2	0.456	0.511	0.511	0.561	0.483	0.539	0.533	0.533	0.522	0.494	0.522	0.550	0.550	0.506	0.450	0.566	0.561	**0.600**	**0.600**
SAD	0.108	0.729	0.932	0.872	0.967	0.963	0.959	N/A	0.982	N/A	0.100	0.973	**0.983**	N/A	0.979	0.963	0.980	0.963	**0.983**
SWJ	0.200	0.267	0.133	0.133	0.200	0.200	0.333	0.333	0.333	0.267	0.333	0.400	0.400	0.400	0.267	0.426	0.400	0.533	**0.600**
UW	0.131	0.497	0.759	0.544	0.881	0.903	0.868	0.869	0.903	0.684	0.916	0.897	0.894	0.859	0.728	**0.919**	0.833	0.897	0.881
Win	0	0	0	0	0	1	0	0	1	2	4	1	1	0	0	2	2	2	**6**
Tie	0	1	1	1	0	1	1	1	2	2	2	3	2	0	1	1	0	4	**7**
Lose	30	29	29	29	30	28	29	29	27	26	24	26	27	30	29	27	28	24	**17**
Best	0	1	1	1	0	2	1	1	3	4	6	4	3	0	1	3	2	6	**13**
AVG_rank	15.817	11.717	11.183	10.667	13.467	10.900	11.700	11.983	8.200	12.833	6.717	6.333	7.917	11.500	11.767	7.250	9.933	5.900	**4.167**



(a) t-SNE on the Epilepsy dataset (b) t-SNE on the UWaveGestureLibrary dataset (c) t-SNE on the RacketSports dataset

(d) KATN with t-SNE on the Epilepsy dataset (e) KATN with t-SNE on the UWaveGestureLibrary dataset (f) KATN with t-SNE on the RacketSports dataset

Fig. 5. Visualization of representations learned by t-SNE and KATN with t-SNE on the Epilepsy, UWaveGestureLibrary, and RacketSports datasets.

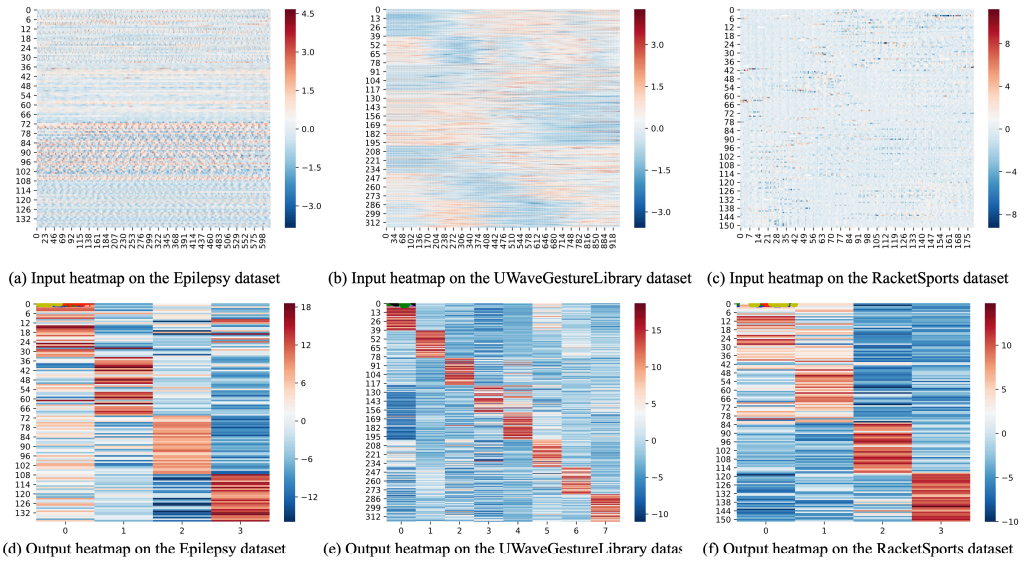
incurring greater complexity. This outcome reflects a deliberate architectural trade-off, where KATN prioritizes expressive temporal-spatial modeling while accepting a moderate increase in computational cost to enhance its modeling effectiveness.

The computational burden becomes more evident when assessing CPU-based inference efficiency. Across the 30 UEA benchmark testing datasets, all transformer variants, including KATN, exhibit substantially longer CPU inference times compared to the seven MTSC baseline models. For instance, the slowest MTSC baseline method, DKN, achieves an average inference time of 15.004s, whereas the fastest transformer model, Linformer, still requires 83.859s. This pronounced disparity highlights the considerably higher deployment cost associated with transformer architectures, particularly in CPU-limited environments.

Focusing specifically on GPU inference time across the same testing datasets, the differences among the seven transformer variants become marginal. For instance, Linformer achieves the shortest GPU average inference time (3.766s), while DualAttTrans incurs the highest (4.714s). This narrow variance suggests that GPU parallelism effectively mitigates the architectural overhead of transformers to a large extent.

TABLE IX
COMPARATIVE EVALUATION OF VARIOUS ALGORITHMS: PARAMETERS, FLOPS, AND INFERENCE TIME ACROSS 30 UEA BENCHMARK TESTING DATASETS.

Metric	Method	AWR	AF	BM	CT	CK	DDG	EW	EP	EC	ER	FD	FM	HMD	HW	HB	IW	JV	LIB	LSST	MI	NATO	PD	PEMS	PS	RS	SRS1	SRS2	SAD	SWJ	UW	Avergae Value
Parameter (M)	MLP	1.162	1.143	0.804	0.785	4.099	182.079	54.456	0.813	3.130	0.635	4.967	1.203	2.504	0.743	12.855	2.707	0.680	0.554	0.617	96.503	1.117	0.515	69.841	1.715	0.594	3.191	4.535	1.111	5.503	0.978	
	FCN	**0.292**	**0.282**	**0.286**	**0.285**	**0.287**	**1.658**	**0.286**	**0.283**	**0.283**	**0.285**	**0.427**	**0.309**	**0.290**	**0.286**	**0.342**	**0.486**	**0.293**	**0.284**	**0.288**	**0.345**	**0.305**	**0.283**	**1.267**	**0.296**	**0.286**	**0.286**	**0.287**	**0.294**	**0.284**	**0.284**	
	InceptionTime	1.596	1.589	1.591	1.593	1.592	2.062	1.591	1.590	1.590	1.639	1.598	1.592	1.594	1.610	1.660	1.594	1.591	1.593	1.611	1.597	1.590	1.928	1.599	1.591	1.590	1.591	1.594	1.590	1.590		
	ResNet	1.240	1.229	1.234	1.233	1.235	2.777	1.234	1.231	1.231	1.232	1.393	1.259	1.239	1.233	1.297	1.458	1.242	1.231	1.235	1.301	1.255	1.230	2.337	1.244	1.234	1.234	1.235	1.243	1.232	1.231	
	MLCN	0.386	0.367	0.331	0.326	0.754	25.934	7.199	0.327	0.623	0.306	1.111	0.421	0.556	0.322	1.972	0.923	0.327	0.294	0.309	12.684	0.404	0.288	10.883	0.463	0.304	0.636	0.810	0.384	0.929	0.349	
	TapNet	2.926	3.143	2.876	2.928	3.446	5.706	12.033	2.924	3.715	2.856	3.137	2.893	3.038	2.919	3.142	3.239	2.857	2.851	2.853	4.477	2.890	2.827	4.861	2.982	2.840	3.281	3.414	2.893	4.099	2.984	
	DKN	1.651	1.700	1.568	1.620	1.870	3.371	6.149	1.592	1.987	1.561	1.654	1.566	1.649	1.625	1.681	2.005	1.576	1.571	1.579	2.348	1.588	1.551	3.210	1.731	1.550	1.766	1.833	1.598	2.178	1.628	
	VanTrans	1.713	1.487	1.299	1.708	3.086	3.648	12.757	1.348	2.139	1.294	1.493	1.297	1.459	1.748	1.443	1.598	1.291	1.327	1.312	2.112	1.317	1.251	2.969	2.339	1.263	1.477	1.544	1.378	2.204	1.565	
	SwinTrans	1.420	1.196	1.006	1.417	2.794	2.799	12.465	1.057	1.848	1.003	1.143	0.995	1.165	1.457	1.127	1.225	0.996	1.037	1.020	1.796	1.017	0.960	2.279	2.045	0.971	1.185	1.251	1.083	1.913	1.274	
	Linformer	1.591	1.367	1.178	1.589	2.965	2.971	12.636	1.228	2.019	1.174	1.315	1.167	1.336	1.629	1.299	1.397	1.168	1.208	1.191	1.967	1.188	1.132	2.450	2.216	1.142	1.356	1.423	1.254	2.084	1.446	
	FMLA	1.903	1.677	1.489	1.898	3.276	3.838	12.947	1.538	2.329	1.484	1.683	1.487	1.649	1.938	1.633	1.788	1.481	1.517	1.502	2.302	1.507	1.441	3.159	2.529	1.453	1.667	1.734	1.568	2.394	1.755	
	DualAggTrans	4.158	3.717	3.334	4.159	6.909	5.805	26.251	3.438	5.020	3.328	3.493	3.294	3.648	4.239	3.531	3.611	3.309	3.398	3.361	4.864	3.340	3.246	5.082	5.407	3.263	3.691	3.823	3.482	5.148	3.872	
	DualAttTrans	4.353	3.914	3.530	4.356	7.105	5.445	26.447	3.635	5.217	3.525	3.632	3.481	3.843	4.436	3.704	3.726	3.502	3.596	3.557	5.036	3.529	3.443	4.880	5.601	3.459	3.887	4.019	3.675	5.345	4.069	
	KATN	3.915	3.473	3.091	3.915	6.666	5.562	26.008	3.195	4.777	3.085	3.250	3.051	3.405	3.995	3.287	3.368	3.065	3.155	3.118	4.621	3.097	3.002	4.839	5.163	3.019	3.447	3.580	3.238	4.905	3.629	
FLOPs (M)	MLP	2.321	2.283	1.604	8.194	1.566	364.155	108.909	1.622	6.257	1.266	9.930	2.402	5.004	1.482	25.707	5.410	1.357	1.105	1.230	193.002	2.230	1.026	139.679	3.426	1.184	6.378	9.066	2.219	11.003	1.953	N/A
	FCN	**0.588**	**0.568**	**0.576**	**0.578**	**0.574**	**3.319**	**0.576**	**0.570**	**0.570**	**0.572**	**0.858**	**0.621**	**0.584**	**0.575**	**0.688**	**0.975**	**0.590**	**0.571**	**0.579**	**0.694**	**0.613**	**0.569**	**2.537**	**0.595**	**0.576**	**0.575**	**0.578**	**0.592**	**0.572**	**0.571**	
	InceptionTime	6.097	7.544	5.723	10.799	5.995	102.146	88.135	6.046	11.979	5.529	8.051	5.820	7.524	5.895	12.953	6.919	5.467	5.443	5.454	63.759	5.779	5.297	43.417	6.641	5.401	9.386	10.863	5.889	15.495	6.475	
	ResNet	4.119	4.097	4.106	4.108	4.103	7.192	4.107	4.099	4.099	4.102	4.424	4.156	4.115	4.105	4.232	4.555	4.121	4.100	4.109	4.239	4.148	4.099	6.312	4.127	4.106	4.106	4.108	4.124	4.101	4.100	
	MLCN	0.772	0.734	0.662	1.508	0.652	51.868	14.390	0.653	1.246	0.612	2.221	0.842	1.111	0.644	3.943	1.846	0.653	0.588	0.618	25.368	0.807	0.576	21.766	0.926	0.608	1.272	1.620	0.768	1.857	0.697	
	TapNet	5.872	6.306	5.772	6.911	5.876	11.432	24.087	5.868	7.450	5.732	6.294	5.806	6.095	5.858	6.305	6.499	5.734	5.722	5.727	8.975	5.799	5.673	9.743	5.984	5.700	6.583	6.849	5.806	8.219	5.988	
	DKN	3.230	3.386	3.120	3.703	3.180	6.723	12.279	3.168	3.959	3.102	3.298	3.122	3.283	3.174	3.352	3.978	3.122	3.096	3.114	4.685	3.155	3.071	6.397	3.353	3.084	3.522	3.655	3.164	4.343	3.230	
	VanTrans	3.419	2.966	2.590	3.410	6.165	7.290	25.507	2.689	4.271	2.581	2.979	2.587	2.911	3.490	2.878	3.190	2.575	2.647	2.617	4.217	2.626	2.495	5.931	4.671	2.519	2.947	3.081	2.749	4.401	3.123	
	SwinTrans	2.833	2.386	2.006	2.828	5.581	5.592	24.924	2.107	3.690	1.999	2.280	1.985	2.324	2.908	2.249	2.445	1.986	2.067	2.033	3.585	2.027	1.915	4.551	4.083	1.935	2.363	2.496	2.160	3.819	2.542	
	Linformer	3.176	2.728	2.349	3.171	5.924	5.934	25.266	2.450	4.032	2.342	2.623	2.327	2.666	3.251	2.591	2.787	2.328	2.410	2.376	3.927	2.370	2.257	4.893	4.426	2.277	2.705	2.839	2.502	4.162	2.884	
	FMLA	3.799	3.346	2.970	3.789	6.545	7.669	25.887	3.068	4.650	2.961	3.358	2.966	3.290	3.869	3.258	3.569	2.954	3.027	2.996	4.596	3.006	2.875	6.310	5.051	2.898	3.326	3.461	3.129	4.781	3.503	
	DualAggTrans	8.306	7.422	6.657	8.306	13.807	11.600	52.491	6.864	10.028	6.645	6.975	6.577	7.285	8.466	7.050	7.211	6.606	6.785	6.710	9.717	6.669	6.480	10.153	10.802	6.514	7.370	7.635	6.952	10.286	7.733	
	DualAttTrans	8.695	7.818	7.050	8.701	14.199	10.878	52.884	7.259	10.423	7.039	7.253	6.951	7.674	8.861	7.397	7.442	6.993	7.180	7.103	10.061	7.046	6.876	9.749	11.190	6.906	7.762	8.027	7.338	10.680	8.127	
	KATN	7.820	6.936	6.171	7.820	13.321	11.114	52.005	6.378	9.542	6.159	6.489	6.091	6.799	7.980	6.564	6.725	6.120	6.299	6.224	9.231	6.183	5.994	9.667	10.316	6.028	6.884	7.149	6.466	9.800	7.247	
With GPU (s)	MLP	**0.038**	**0.026**	**0.027**	**0.068**	**0.047**	**0.477**	**0.521**	**0.030**	**0.052**	**0.034**	**0.600**	**0.029**	**0.033**	**0.055**	**0.149**	2.235	**0.038**	**0.031**	**0.125**	**0.463**	**0.032**	**0.097**	**0.645**	**0.325**	**0.032**	**0.057**	**0.054**	**0.129**	**0.028**	**0.045**	**0.217**
	FCN	0.973	0.976	0.759	0.937	0.859	0.837	2.033	0.834	1.406	0.837	1.099	0.806	0.814	0.844	0.904	**1.355**	0.815	0.851	0.974	1.032	0.875	0.966	0.871	1.152	0.808	1.267	1.022	1.165	0.878	0.920	0.996
	InceptionTime	1.099	1.004	1.165	1.334	1.121	1.083	4.047	2.375	1.917	1.189	1.884	1.171	1.053	2.533	1.165	2.085	1.107	1.093	1.484	1.384	1.121	1.140	1.675	2.573	1.143	1.478	1.275	1.984	1.078	1.206	1.532
	ResNet	0.977	0.935	0.961	1.221	0.965	0.934	4.041	0.926	2.099	0.887	1.483	1.054	0.934	1.034	1.000	1.861	1.027	1.127	1.350	1.310	1.117	0.954	1.000	1.890	1.004	1.490	1.176	1.856	0.960	1.142	1.290
	MLCN	3.383	13.993	2.583	4.306	27.072	6.138	5.569	4.606	4.360	1.863	2.496	1.669	2.946	3.700	2.686	2.208	1.508	1.535	1.769	7.708	1.604	1.022	3.986	1.098	1.240	1.992	2.567	2.729	1.640	0.908	4.028
	TapNet	1.019	0.871	0.874	1.444	1.047	10.093	6.298	0.946	2.512	0.923	2.408	1.073	0.969	1.105	1.328	4.037	1.020	1.005	1.257	1.786	1.128	0.966	8.306	2.113	0.958	1.883	1.358	1.599	1.000	1.278	2.087
	DKN	1.284	1.159	1.292	1.406	1.245	2.131	3.398	2.435	1.777	1.306	2.721	1.302	1.155	2.601	1.309	10.324	1.226	1.248	1.696	1.592	1.245	1.306	3.058	3.128	1.252	1.494	1.349	1.975	1.193	1.293	1.997
	VanTrans	2.095	2.071	2.488	2.699	3.358	2.059	21.048	2.046	11.255	2.159	3.113	2.142	2.124	2.262	2.410	3.380	2.147	2.250	2.580	12.681	2.270	2.101	2.348	4.275	2.062	4.744	4.799	3.530	3.668	2.408	3.952
	SwinTrans	2.085	2.216	2.319	2.855	3.466	2.058	19.745	2.032	11.626	2.437	3.251	2.079	2.123	2.282	2.415	3.164	2.109	2.189	2.496	12.363	2.209	2.061	2.521	4.561	2.029	4.890	4.985	3.463	3.742	2.539	3.944
	Linformer	2.037	2.012	2.283	2.689	3.205	2.017	20.880	2.099	9.876	2.339	3.029	2.081	2.076	2.158	2.350	3.054	2.070	2.170	2.493	11.118	2.182	2.117	2.472	4.125	2.039	4.329	4.371	3.111	3.878	2.325	3.766
	FMLA	2.164	2.139	2.570	2.788	3.469	2.127	21.742	2.113	11.626	2.230	3.215	2.213	2.194	2.337	2.490	3.492	2.218	2.325	2.665	13.100	2.344	2.171	2.425	4.416	2.130	4.900	4.957	3.647	3.789	2.488	4.083
	DualAggTrans	2.354	2.277	2.509	3.160	3.585	2.258	31.232	2.288	11.500	2.177	3.463	2.514	2.381	2.595	2.682	5.071	2.460	2.689	3.139	12.582	2.704	2.372	2.391	5.075	2.378	5.088	5.085	3.231	3.842	2.737	4.594
	DualAttTrans	2.339	2.296	2.631	3.165	3.588	2.265	32.655	2.271	12.025	2.179	4.131	2.545	2.345	2.592	3.025	4.475	2.469	2.650	3.101	12.642	2.669	3.292	2.385	5.130	2.373	5.070	5.084	3.101	4.157	2.765	4.714
	KATN	2.365	2.303	2.287	3.059	3.472	2.216	30.242	2.228	10.936	2.143	3.294	2.399	2.389	2.518	2.626	3.973	2.418	2.594	3.021	11.979	2.583	2.289	2.358	4.833	2.303	4.816	4.879	2.958	3.755	2.650	4.396
With CPU (s)	MLP	**0.180**	**0.170**	0.181	**0.180**	**0.181**	**0.228**	**0.254**	**0.181**	**0.188**	0.180	**0.248**	0.179	**0.184**	**0.180**	**0.201**	**0.475**	0.182	0.172	**0.204**	**0.242**	0.177	**0.183**	**0.287**	**0.270**	0.181	**0.178**	**0.183**	**0.201**	**0.173**	**0.175**	**0.207**
	FCN	0.344	0.113	**0.079**	1.802	0.655	0.614	16.242	0.240	3.095	**0.168**	2.243	**0.085**	0.257	0.951	0.731	5.913	**0.128**	**0.102**	0.716	2.484	**0.113**	0.265	0.880	4.831	**0.081**	1.829	1.462	1.695	0.311	0.737	1.639
	InceptionTime	2.401	0.664	0.423	13.598	4.454	0.992	121.256	1.630	23.553	1.091	11.129	0.456	1.710	6.569	4.083	26.891	0.748	0.610	4.535	15.697	0.665	1.570	1.572	33.786	0.451	13.475	10.735	11.059	2.097	5.357	10.775
	ResNet	2.081	0.567	0.318	11.820	3.948	1.243	116.298	1.405	20.842	0.948	10.462	0.361	1.470	6.058	3.915	26.804	0.623	0.553	4.133	13.973	0.549	1.444	1.941	31.775	0.363	11.873	9.400	10.024	1.829	4.736	10.059
	MLCN	2.804	12.259	1.872	5.730	24.642	5.954	123.254	3.790	36.196	1.186	4.601	0.849	7.733	3.445	5.227	12.372	0.941	0.723	1.800	14.536	0.918	0.640	9.118	18.253	0.678	20.370	12.150	3.166	2.356	8.250	11.527
	TapNet	2.778	0.682	0.377	15.934	5.345	10.929	143.189	1.837	27.686	1.187	16.999	0.560	1.974	7.839	5.617	54.096	0.801	0.591	5.464	19.352	0.778	1.808	10.974	42.985	0.407	15.889	12.593	13.381	2.409	6.164	14.354
	DKN	1.980	0.726	0.577	9.958	3.470	12.249	85.752	1.400	17.034	1.056	21.503	0.611	1.465	4.982	3.341	185.656	0.804	0.682	3.457	11.533	0.744	1.401	22.076	24.887	0.569	9.949	7.950	8.224	1.767	4.307	15.004
	VanTrans	5.380	4.123	1.168	38.482	53.356	3.625	1285.177	4.593	427.210	2.424	16.277	1.148	6.839	16.930	19.055	33.758	1.407	1.572	5.670	472.960	1.408	2.126	4.208	107.070	1.190	125.848	122.685	19.012	53.548	18.232	95.216
	SwinTrans	5.069	4.164	1.144	30.910	53.011	3.308	1266.689	4.395	423.553	1.951	14.849	1.094	7.098	14.023	18.399	27.767	1.336	1.272	5.145	482.600	1.345	1.836	3.769	95.385	1.034	129.769	120.393	17.453	50.656	18.287	93.590
	Linformer	4.401	3.496	1.134	26.107	42.417	3.039	1274.977	3.813	336.504	1.851	13.263	1.111	5.730	12.014	14.918	26.087	1.292	1.220	4.896	382.955	1.359	1.811	3.451	79.582	1.021	96.233	96.864	16.734	41.749	15.737	83.859
	FMLA	5.557	4.259	1.207	39.751	55.117	3.745	1327.587	4.745	441.307	2.504	16.814	1.185	7.064	17.488	19.684	34.872	1.454	1.624	5.857	488.568	1.454	2.196	4.347	110.604	1.230	130.001	126.733	19.639	55.315	18.833	98.358
	DualAggTrans	7.770	4.595	1.479	45.499	52.576	4.290	1800.423	6.046	408.678	3.192	27.959	1.537	8.167	21.365	20.908	63.946	2.107	1.880	10.468	434.299	2.042	3.701	5.640	132.678	1.455	125.220	120.761	29.381	47.144	23.063	113.942
	DualAttTrans	7.974	4.628	1.520	46.092	53.064	4.364	1896.345	6.167	401.186	3.257	28.957	1.561	8.201	21.689	21.166	68.017	2.144	1.914	10.689	435.991	2.099	3.786	5.844	135.441	1.465	126.157	121.964	29.466	47.536	23.437	117.404
	KATN	7.288	4.294	1.465	42.040	48.801	4.007	1714.426	5.731	365.158	3.058	26.189	1.492	7.488	19.785	19.443	58.800	2.038	1.793	9.739	401.517	1.964	3.542	5.195	122.553	1.406	115.397	111.667	26.956	44.005	21.352	106.620



(a) Input heatmap on the Epilepsy dataset (b) Input heatmap on the UWaveGestureLibrary dataset (c) Input heatmap on the RacketSports dataset

(d) Output heatmap on the Epilepsy dataset (e) Output heatmap on the UWaveGestureLibrary datas (f) Output heatmap on the RacketSports dataset

Fig. 6. Input and output heatmaps of KATN on the Epilepsy, UWaveGestureLibrary, and RacketSports datasets.

Nevertheless, the variation in CPU inference time among transformer models remains considerable. KATN, in particular, ranks third in CPU inference time, performing slightly faster than DualAggTrans and DualAttTrans, yet significantly slower than Linformer, FMLA, and VanTrans. This performance gap is primarily attributed to KATN's complex feature fusion strategy and temporal modeling architecture, which, while enhancing representational expressiveness, introduce greater computational overhead under serial processing conditions.

In conclusion, the evaluation across the 30 UEA benchmark testing datasets demonstrates that transformer-based models, including KATN, exhibit strong learning capacity but incur notable costs in inference efficiency, particularly on CPU. KATN navigates this trade-off by delivering competitive performance with a manageable computational footprint relative to other transformer-based models, although it remains more resource-intensive than simpler architectures.

F. Representation Visualization Analysis

To rigorously examine the expressive capacity of KATN, we

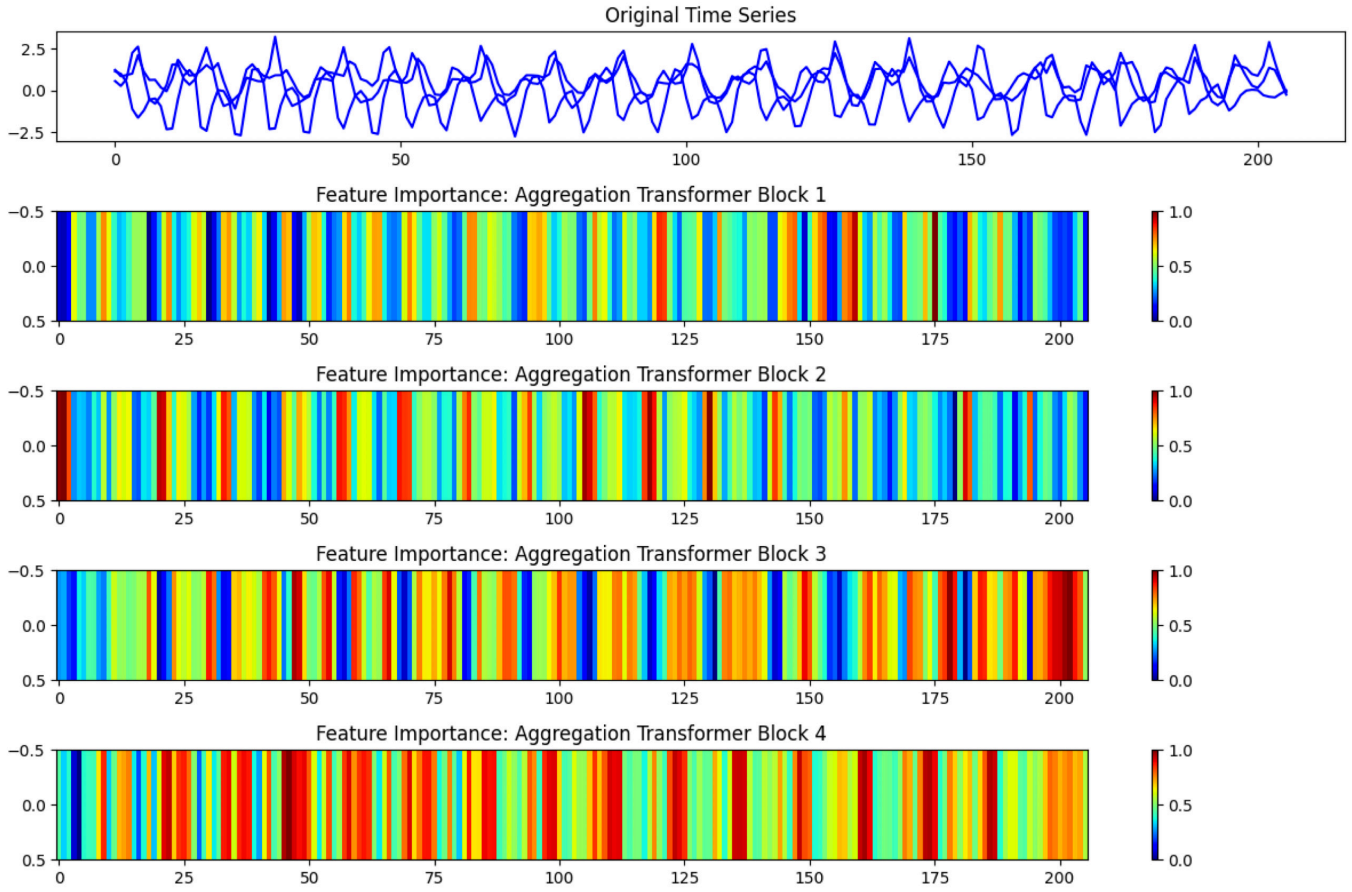


Fig. 7. Visualization of temporal receptive field expansion across aggregation transformer blocks in KATN on the Epilepsy dataset. This figure illustrates the hierarchical evolution of temporal activation patterns as input signals propagate through KATN's aggregation transformer blocks. The color intensity represents the normalized variance of feature responses at each time step, where warmer hues indicate time regions of greater importance for seizure characterization. The effective activation widths for blocks 1 to 4 are measured as 81, 95, 137, and 143 time steps, respectively, under a threshold of 0.5. This gradual increase reflects the model's ability to integrate information over progressively extended temporal ranges. By capturing both fine-grained local variations and long-range dependencies, KATN enables more comprehensive temporal reasoning, which is essential for accurately identifying complex seizure dynamics in multivariate time series.

perform a two-dimensional projection of its high-dimensional latent features using t-distributed stochastic neighbor embedding (t-SNE) [72], a nonlinear manifold learning method well-suited for visualizing complex feature spaces. As shown in Fig. 5, this projection enables a comparative analysis of embeddings across the Epilepsy, UWaveGestureLibrary, and RacketSports datasets. Notably, the embeddings produced by KATN exhibit more distinct and tightly grouped clusters than those derived from t-SNE applied directly to raw features. This improvement is particularly salient in the UWaveGestureLibrary dataset (Figs. 5(b) and 5(e)), where intra-class cohesion and inter-class boundaries are clearly delineated—reflecting the model's ability to extract semantically aligned, task-relevant representations.

To complement this analysis, we present the input and output heatmaps in Fig. 6, which illustrate how KATN hierarchically refines temporal features across successive layers. These visualizations reveal a selective enhancement of salient dynamics while diminishing the influence of spurious or redundant patterns, thereby improving representational compactness and class separability.

Beyond these spatial representations, we further investigate KATN's capacity for modeling long-range temporal dependencies—an essential trait for MTSC tasks involving delayed or distributed temporal cues, such as seizure recognition. Following the methodology in [73], we visualize the progressive expansion of temporal receptive fields across the model's four aggregation transformer blocks on the Epilepsy dataset (Fig. 7). The heatmaps, constructed via normalized feature variance at each time step, highlight regions with heightened predictive relevance, with warmer tones indicating greater salience.

A clear hierarchical pattern emerges: as the depth of the network increases, so too does the temporal span of effective activation. Specifically, the receptive fields expand from 81 time steps in the first transformer block to 95, 137, and 143 in subsequent layers, using a fixed salience threshold of 0.5. This progressive broadening indicates a gradual shift from localized temporal processing toward integration of extended temporal contexts. The integration of multi-head self-attention with

MResNet facilitates adaptive focusing on temporally distant yet semantically correlated events. Meanwhile, the additive feature aggregation design enables the seamless fusion of temporal information across multiple scales.

Through this multi-stage refinement, KATN constructs a temporally expressive representation space that unifies short-term cues and long-term dependencies. Such hierarchical modeling is essential for uncovering complex, non-local patterns in real-world time series. The observed temporal activation dynamics thus provide compelling evidence of KATN's ability to internalize and operationalize long-range temporal structures, enhancing its utility for challenging tasks such as epileptic event detection.

V. Conclusion

The proposed KATN incorporates four aggregation transformer blocks, each merging the local patterns extracted by MResNet with the global patterns derived from the multi-head attention network. These blocks are able to capture intricate long-range dependencies prevalent among multiple variables. Through extensive experiments, KATN demonstrates its superiority over 6 SOTA transformer algorithms, i.e., VanTrans, SwinTrans, Linformer, FMLA, DualAggTrans, and DualAttTrans, in terms of 'win'/'tie'/'lose' and AVG_rank. Furthermore, when compared to 18 MTSC algorithms on 30 UEA datasets, KATN is winner of 13 datasets, attaining an AVG_rank score of 4.167. These results signify KATN's promising potential in addressing various MTSC applications in real-world applications.

While KATN offers significant benefits, it also presents challenges, particularly due to its resource-intensive nature, which may hinder practical deployment. To mitigate this, we plan to incorporate network compression techniques, reducing computational complexity by eliminating redundant nodes. Additionally, we aim to improve the model's robustness through automatic hyper-parameter optimization, exploring methods such as grid search, random search, and Bayesian optimization to identify optimal configurations for diverse datasets and applications. Furthermore, we intend to leverage neural architecture search to optimize both hyper-parameters and the network architecture in an integrated manner, ensuring enhanced performance and adaptability.

References

[1] C. Chen, L. Li, M. Beetz, A. Banerjee, R. Gupta, and V. Grau, "Large language model-informed ecg dual attention network for heart failure risk prediction," *IEEE Trans. Big Data*, pp. 1–13, 2025.

[2] A. Hussain, T. Hussain, W. Ullah, S. U. Khan, M. J. Kim, K. Muhammad, J. D. Ser, and S. W. Baik, "Big data analysis for industrial activity recognition using attention-inspired sequential temporal convolution network," *IEEE Trans. Big Data*, pp. 1–12, 2024.

[3] Y. Shi, X. Zhang, Y. Shang, and N. Yu, "Don't be misled by emotion! disentangle emotions and semantics for cross-language and cross-domain rumor detection," *IEEE Trans. Big Data*, vol. 10, no. 3, pp. 249–259, 2024.

[4] F. Ding, B. Li, X. Ben, J. Zhao, and H. Zhou, "Alad: A new unsupervised time series anomaly detection paradigm based on activation learning," *IEEE Trans. Big Data*, pp. 1–13, 2024.

[5] A. Chen, X. Zhou, Y. Fan, and H. Chen, "Anomaly detection in multi-level model space," *IEEE Trans. Big Data*, pp. 1–12, 2025.

[6] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Min. Knowl. Disc.*, vol. 33, pp. 917–963, 2019.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, pp. 436–444, 2015.

[8] H. Xing, Z. Xiao, R. Qu, Z. Zhu, and B. Zhao, "An efficient federated distillation learning system for multitask time series classification," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[9] F. He, T.-Y. Fu, and W.-C. Lee, "Rel-cnn: Learning relationship features in time series for classification," *IEEE Trans. Knowl. Data En.*, vol. 35, no. 7, pp. 7412–7426, 2023.

[10] Q. Ma, S. Li, and G. W. Cottrell, "Adversarial joint-learning recurrent neural network for incomplete time series classification," *IEEE Trans. Pattern Anal.*, vol. 44, no. 4, pp. 1765–1776, 2022.

[11] R. Chen, X. Yan, S. Wang, and G. Xiao, "Da-net: Dual-attention network for multivariate time series classification," *Inf. Sci.*, vol. 610, pp. 472–487, 2022.

[12] G. He, L. Dai, Z. Yu, and C. L. P. Chen, "Gan-based temporal association rule mining on multivariate time series data," *IEEE Trans. Knowl. Data En.*, pp. 1–13, 2023.

[13] Z. Xiao, X. Xu, H. Xing, S. Luo, P. Dai, and D. Zhan, "Rtfn: A robust temporal feature network for time series classification," *Inf. Sci.*, vol. 571, pp. 65–86, 2021.

[14] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification," *Neural Networks*, vol. 116, pp. 237–245, 2019.

[15] S. Huang, L. Xu, and C. Jiang, "Artificial intelligence and advanced time series classification: Residual attention net for cross-domain modeling," *Fintech with Artificial Intelligence, Big Data, and Blockchain, Blockchain Technologies*, 2021.

[16] H. Xing, Z. Xiao, D. Zhan, S. Luo, P. Dai, and K. Li, "Selfmatch: Robust semisupervised time-series classification with self-distillation," *Int. J. Intell. Syst.*, pp. 1–28, 2022.

[17] Z. Xiao, H. Xing, R. Qu, L. Feng, S. Luo, P. Dai, B. Zhao, and Y. Dai, "Densely knowledge-aware network for multivariate time series classification," *IEEE Trans. Syst. Man Cy-S.*, pp. 1–13, 2024.

[18] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[19] P. Munro, "Backpropagation," *Sammut, C., Webb, G. I. (eds) Encyclopedia of Machine Learning*, 2011.

[20] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *in Proc. NeurIPS*, pp. 24 261–24 272, 2021.

[21] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. CVPR*, 2019, pp. 3146–3154.

[22] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. CVPR*, 2017, pp. 5659–5667.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[24] Q. Chen, Q. Wu, J. Wang, Q. Hu, T. Hu, E. Ding, J. Cheng, and J. Wang, "Mixformer: Mixing features across windows and dimensions," in *Proc. CVPR*, 2022, pp. 5249–5259.

[25] Z. Chen, Y. Zhang, J. Gu, L. Kong, X. Yang, and F. Yu, "Dual aggregation transformer for image super-resolution," in *Proc. CVPR*, 2023, pp. 12 312–12 321.

[26] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[27] A. Bagnall., J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Min. Knowl. Disc.*, vol. 31, pp. 1–55, 2017.

[28] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Min. Knowl. Disc.*, vol. 29, pp. 565–592, 2015.

[29] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time series classification with cote: the collective of transformation-based ensembles," *In Proc. ICDE 2016*, pp. 1548–1549, 2016.

[30] J. Lines, S. Taylor, and A. Bagnall, "Time series classification with hive-cote: the hierarchical of transformation-based ensembles," *ACM Trans. Knowl. Discov. D.*, vol. 21, no. 52, pp. 1–35, 2018.

[31] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, "Hive-cote 2.0: a new meta ensemble for time series classifcation," *Machine Learning*, vol. 110, pp. 3211–3243, 2021.

[32] K. Fauvel, É. Fromont, V. Masson, P. Faverdin, and A. Termier, "Xem: An explainable-by-design ensemble method for multivariate time series classification," *Data Min. Knowl. Disc.*, vol. 36, pp. 917–957, 2022.

[33] G. Li, B. Choi, J. Xu, S. S. Bhowmick, K.-P. Chun, and G. L.-H. Wong, "Efficient shapelet discovery for time series classification," *IEEE Trans. Knowl. Data En.*, vol. 34, no. 3, pp. 1149–1163, 2022.

[34] F. J. Baldán and J. M. Benítez, "Multivariate times series classification through an interpretable representation," *Inf. Sci.*, vol. 569, pp. 596–614, 2021.

[35] A. Shifaz, C. Pelletier, F. Petitjean, and G. I. Webb, "Ts-chief: A scalable and accurate forest algorithm for time series classification," *Data Min. Knowl. Disc.*, vol. 34, pp. 742–775, 2020.

[36] M. G. Baydogan and G. Runger, "Time series representation and similarity based on local auto patterns," *Data Min. Knowl. Disc.*, vol. 30, pp. 476–509, 2016.

[37] W. Pei, H. Dibeklioğlu, D. M. J. Tax, and L. van der Maaten, "Multivariate time-series classification using the hidden-unit logistic model," *IEEE Trans. Neur. Net. Lear.*, vol. 29, no. 4, pp. 920–931, 2018.

[38] H. Deng, G. Runger, E. Tuv, and M. Vladimir, "A time series forest for classification and feature extraction," *Inf. Sci.*, vol. 239, pp. 142–153, 2013.

[39] G. He, X. Xin, R. Peng, M. Han, J. Wang, and X. Wu, "Online rule-based classifier learning on dynamic unlabeled multivariate time series data," *IEEE Trans. Syst. Man Cy-S.*, vol. 52, no. 2, pp. 1121–1134, 2022.

[40] K. S. Tuncel and M. G. Baydogan, "Autoregressive forests for multivariate time series modeling," *Pattern Recogn.*, vol. 73, pp. 202–215, 2018.

[41] J. Large, A. Bagnall, S. Malinowski, and R. Tavenard, "From bop to boss and beyond: time series classification with dictionary based classifier," *arXiv preprint arXiv:1809.06751*, 2018.

[42] P. Schäfer and U. Leser, "Multivariate time series classification with weasel+muse," *arXiv preprint arXiv:1711.11343*, 2017.

[43] K. Wu, K. Yuan, Y. Teng, J. Liu, and L. Jiao, "Broad fuzzy cognitive map systems for time series classification," *App. Soft Comput.*, vol. 128, pp. 1–13, 2022.

[44] B. Zhao, H. Xing, X. Wang, F. Song, and Z. Xiao, "Rethinking attention mechanism in time series classification," *Inf. Sci.*, vol. 627, pp. 97–114, 2023.

[45] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: finding alexnet for time series classification," *Data Min. Knowl. Disc.*, vol. 34, pp. 1936–1962, 2020.

[46] D. Lee, S. Lee, and H. Yu, "Learnable dynamic temporal pooling for time series classification," *In Proc. AAAI*, vol. 35, no. 9, pp. 8288–8296, 2021.

[47] H. I. Fawaz, G. Forestie *et al.*, "Time series classification from scratch with deep neural networks: A strong baseline," *In Proc. IEEE IJCNN 2017*, pp. 1578–1585, 2017.

[48] Z. Xiao, X. Xu, H. Zhang, and E. Szczerbicki, "A new multi-process collaborative architecture for time series classification," *Knowl.-Based Syst.*, vol. 220, pp. 1–11, 2021.

[49] G. Li *et al.*, "Shapenet: A shapelet-neural network approach for multivariate time series classification," *In Proc. AAAI*, vol. 35, pp. 8375–8383, 2021.

[50] Z. Xiao, H. Xing, B. Zhao, R. Qu, S. Luo, P. Dai, K. Li, and Z. Zhu, "Deep contrastive representation learning with self-distillation," *IEEE Trans. Emerg. Top. Comput. Intell.*, pp. 1–13, 2023.

[51] F. M. Bianchi, S. Scardapane, S. Løkse, and R. Jenssen, "Reservoir computing approaches for representation and classification of multivariate time series," *IEEE Trans. Neur. Net. Learn.*, vol. 32, no. 5, pp. 2169–2179, 2021.

[52] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Min. Knowl. Disc.*, vol. 34, pp. 1454–1495, 2020.

[53] L. Sun, C. Li, Y. Ren, and Y. Zhang, "A multitask dynamic graph attention autoencoder for imbalanced multilabel time series classification," *IEEE Trans. Neur. Net. Lear.*, pp. 1–14, 2024.

[54] Y. Cheng, D. Lee, H. Oberhauser, and H. Li, "Generalized time series classification via component decomposition and alignment," *IEEE Trans. Big Data*, pp. 1–16, 2025.

[55] Z. Huang, C. Yang, X. Chen, X. Zhou, G. Chen, T. Huang, and W. Gui, "Functional deep echo state network improved by a bi-level optimization approach for multivariate time series classification," *Appl. Soft Comput.*, vol. 106, pp. 1–12, 2021.

[56] A. Dempster, D. F. Schmidt, and G. I. Webb, "Minirocket: A very fast (almost) deterministic transform for time series classification," *In Proc. KDD'21*, pp. 248–257, 2021.

[57] Z. Xiao, X. Xu, H. Xing, R. Qu, F. Song, and B. Zhao, "Rnts: Robust neural temporal search for time series classification," *In Proc. IJCNN 2021*, pp. 1–8, 2021.

[58] Z. Xiao, X. Xu, H. Xing, B. Zhao, X. Wang, F. Song, R. Qu, and L. Feng, "Dtcm: Deep transformer capsule mutual distillation for multivariate time series classification," *IEEE Trans. Cogn. Dev. Syst.*, pp. 1–17, 2024.

[59] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu, "Tapnet: Multivariate time series classification with attentional prototypical network," *In Proc. AAAI 2020*, pp. 6845–6852, 2020.

[60] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Comput. Surv.*, vol. 55, no. 6, 2022.

[61] A. Vaswani, N. Shazeer, J. U. N. Parmar, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *in Proc. NeurIPS 2017*, pp. 5998–6008, 2017.

[62] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[63] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. ICCV*, 2021, pp. 9992–10 002.

[64] M. Ding, B. Xiao, N. Codella, P. Luo, J. Wang, and L. Yuan, "Davit: Dual attention vision transformers," in *Proc. ECCV*. Springer, 2022, pp. 74–92.

[65] Y. Li, X. Peng, J. Zhang, Z. Li, and M. Wen, "Dct-gan: Dilated convolutional transformer-based gan for time series anomaly detection," *IEEE Trans. Knowl. Data En.*, vol. 35, no. 4, pp. 3632–3644, 2023.

[66] Z. Lin, S. Zang, R. Wang, Z. Sun, J. Senthilnath, C. Xu, and C. K. Kwoh, "Attention over self-attention: Intention-aware re-ranking with dynamic transformer encoders for recommendation," *IEEE Trans. Knowl. Data En.*, vol. 35, no. 8, pp. 7782–7795, 2023.

[67] Y. Liu, S. Pan, Y. G. Wang, F. Xiong, L. Wang, Q. Chen, and V. C. Lee, "Anomaly detection in dynamic graphs via transformer," *IEEE Trans. Knowl. Data En.*, vol. 35, no. 12, pp. 12 081–12 094, 2023.

[68] L. Xia, C. Huang, Y. Xu, and J. Pei, "Multi-behavior sequential recommendation with temporal graph transformer," *IEEE Trans. Knowl. Data En.*, vol. 35, no. 6, pp. 6099–6112, 2023.

[69] A. Bagnall, H. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The uea multivariate time series classification archive, 2018," *arXiv preprint arXiv: 1811.00075*, 2018.

[70] C. Liu, J. Zhen, and W. Shan, "Time series classification based on convolutional network with a gated linear units kernel," *Eng. Appl. Artif. Intel.*, vol. 123, pp. 1–11, 2023.

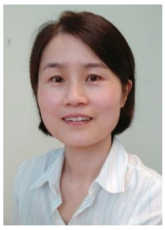
[71] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, "Model complexity of deep learning: A survey," *Knowl. Inf. Syst.*, vol. 63, pp. 2585–2619, 2021.

[72] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.

[73] A. A. Ismail, M. Gunady, H. C. Bravo, and S. Feizi, "Benchmarking deep learning interpretability in time series predictions," in *Proc. the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20, 2020.

Zhiwen Xiao (Member, IEEE) received the B.Eng. degree in network engineering from the Chengdu University of Information Technology, Chengdu, China, in 2019, and the M.Eng. degree in computer science from the Northwest A&F University, Yangling, China, in 2023. He is currently pursuing the Ph.D. degree in computer science with Southwest Jiaotong University, Chengdu. His research interests include data mining, time series analysis, federated learning, representation learning, semantic communication, and computer vision.

Huanlai Xing (Member, IEEE) received Ph.D. degree in computer science from University of Nottingham, Nottingham, U.K., in 2013. He was a Visiting Scholar in Computer Science, The University of Rhode Island, USA. Supervisor: Dr. Haibo He (Robert Haas Endowed Chair Professor, IEEE Fellow, https://www.ele.uri.edu/faculty/he/) in 2020-2021. Huanlai Xing is an Associate Professor and PhD Supervisor with the School of Computing and Artificial Intelligence, Southwest Jiaotong University. He is on Editorial Board (Young Scientists Committee) of SCIENCE CHINA INFORMATION SCIENCES. He was a member of several international conference program and senior program committees, such as IJCAI, ECML-PKDD, MobiMedia, ISCIT, ICCC, IJCNN, TrustCom, and ICSINC. His research interests include semantic communication, representation learning, data mining, reinforcement learning, machine learning, network function virtualization, and software defined networking.



Rong Qu (Senior Member, IEEE) is a full Professor at the School of Computer Science, University of Nottingham. She received her B.Sc. in Computer Science and Its Applications from Xidian University, China in 1996 and Ph.D. in Computer Science from The University of Nottingham, U.K. in 2003. Her research interests include the modeling and optimization for logistics transport scheduling, personnel scheduling, network routing, portfolio optimization and timetabling problems by using evolutionary algorithms, mathematical programming, constraint programming in operational research and artificial intelligence. These computational techniques are integrated with knowledge discovery, machine learning and data mining to provide intelligent decision support on logistic fleet operations at SMEs, workforce scheduling at hospitals, policy making in education, and cyber security for connected and autonomous vehicles.

Dr. Qu is an associated editor at Engineering Applications of Artificial Intelligence, IEEE Computational Intelligence Magazine, IEEE Transactions on Evolutionary Computation, Journal of Operational Research Society, and PeerJ Computer Science. She is a Senior IEEE Member since 2012 and the Vice-Chair of Evolutionary Computation Task Committee since 2019 and Technical Committee on Intelligent Systems Applications (2015-2018) at IEEE Computational Intelligence Society. She has guest edited special issues on the automated design of search algorithms and machine learning at the IEEE Transactions on Pattern Analysis and Machine Intelligence and IEEE Computational Intelligence Magazine.

Hui Li received the B.Sc. and M.Sc. degrees in applied mathematics from the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K., in 2008. He is currently a Professor with the School of Mathematics and Statistics, Xi'an Jiaotong University. His current research interests include evolutionary computation, multiobjective optimization, and machine learning. Dr. Li was a recipient of the 2010 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award as one of the inventors for MOEA/D.

Huagang Tong, received his Ph.D. degree in management sciences from Nanjing University of Aeronautics and Astronautics in 2022. He is an Associate Professor with College of Economic and Management, Nanjing Tech University, Puzhu Road(S), Nanjing 211816, China. His research interests are knowledge discovery, operation research, and data mining.

Shouxi Luo (Member, IEEE) received the bachelor's degree in communication engineering and the Ph.D. degree in communication and information systems from the University of Electronic Science and Technology of China, Chengdu, China, in 2011 and 2016, respectively. He is an Associate Professor with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu. His research interests include data center networks, software-defined networking, and networked systems.

Song Jing is a Senior Engineer with the Informatization Research Institute, Southwest Jiaotong University, and Deputy Director of the Intelligent Network Laboratory, National Engineering Research Center for High-Speed Railway and Urban Rail Transit. He was selected into Tianfu Ten-thousand Talents Program. He have involved in various projects and engineering tasks such as the National Railway Group's Railway 5G Network Security and Passenger Dedicated Line Passenger Service System Security Assurance Platform. He focuses on cybersecurity in cyberspace and information communication networks.

Li Feng received his PhD degree from Xi'an Jiaotong University, in 2005, under the supervision of Prof. Xiaohong Guan (Academian of CAS, IEEE Fellow). He is a Research Professor and PhD supervisor with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu. His research interests include artificial intelligence, cyber security and its applications.

Qian Wan received the B.Eng. degree in computer science from the Wuhan University, Wuhan, China, and the M.Des. degree in interaction design from the China University of Geosciences, Wuhan, China. He is pursuing the Ph.D. degree in computer science at Southwest Jiaotong University, Chengdu, China. His research interests include data model and mining, virtual reality, and augmented reality.