Value and Shape-Aware Transformer for Multivariate Time Series Classification

Wenjie Xi* Rundong Zuo† Alejandro Alvarez* Jie Zhang* Jessica Lin*

Abstract

Multivariate time series classification is a crucial task in data mining, attracting growing research interest due to its broad applications. While many existing methods focus on discovering discriminative patterns in time series, real-world data does not always present such patterns, and sometimes raw numerical values can also serve as discriminative features. Additionally, the recent success of Transformer models has inspired many studies. However, when applying to time series classification, the self-attention mechanisms in Transformer models could introduce classification-irrelevant features, thereby compromising accuracy. To address these challenges, we propose a novel method, VSFormer, that incorporates both discriminative patterns (**shape**) and numerical information (**value**). In addition, we extract class-specific prior information derived from supervised information to enrich the positional encoding and provide classification-oriented self-attention learning, thereby enhancing its effectiveness. Extensive experiments on all 30 UEA archived datasets demonstrate the superior performance of our method compared to SOTA models. Through ablation studies, we demonstrate the effectiveness of the improved encoding layer and the proposed self-attention mechanism. Finally, We provide a case study on a real-world time series dataset without discriminative patterns to interpret our model.

1 Introduction

A multivariate time series (MTS) consists of multiple time-ordered measurements or observations across different variables. Multivariate Time Series Classification (MTSC) is one of the key tasks for MTS, and it is widely applied in various fields such as motion recognition [21], solar flare prediction [1], and human activity recognition [26]. As a result, it has garnered increasing research interest recently [22].

While numerous methodologies have emerged showcasing promising results [22], one popular kind of approach revolves around identifying and exploiting repeated patterns (subsequences) in time series, treating them as discriminatory features for classification. Such subsequences typically undergo normalization to underscore their shape characteristics. In [27], the authors introduced the notion of Shapelets—the subsequences that can best represent a class—and use them with traditional classifiers such as decision trees to classify time series. Building upon this idea, recent deep learning methods [15, 19, 31] have targeted the search for such patterns through the use of neural networks resulting in improved performances.

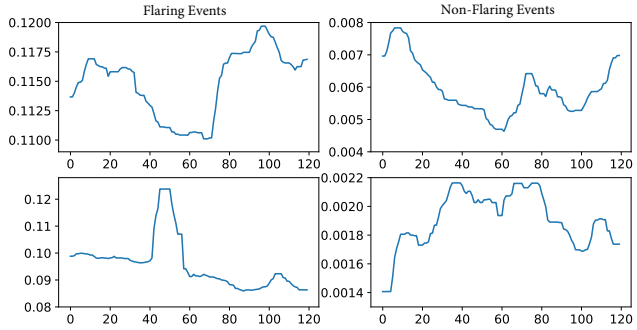


Figure 1: Two time series samples of flaring events (left) and two samples of non-flaring events (right).

However, a crucial limitation of these methods is their foundational assumption for the presence of discriminative patterns for MTSC. Many real-world time series data lack such specific patterns that can effectively distinguish one sequence from another, rendering these methods sub-optimal. A related example as presented in Figure 1 is found in solar flare prediction. The two time series on the left correspond to conditions leading to flaring events and those on the right to non-flaring events. While it may be challenging to identify discriminative patterns (also known as **shape**), the raw numerical information (the **value**), differing by orders of magnitude, provide a significant distinction between the two classes (The values from the first class (0.08-0.12) are at least one order of magnitude larger than those from the second class (0.0014-0.008)). Shape-based methods require the normalization of subsequences for meaningful comparison of shapes; however, such a step also strips away valuable information presented in the raw values of time series.

*Department of Computer Science, George Mason University, United States. {wxi, aalvar10, jzhang7, jessica}@gmu.edu.

†Department of Computer Science, Hong Kong Baptist University, Hong Kong. {csrdzuo}@comp.hkbu.edu.hk.

Second, the recent success of Transformers [12, 25] has motivated researchers to explore their potential in time series classification [8,13,28,31]. However, the self-attention mechanisms of existing Transformer models for MTSC are not classification-oriented. Given they directly make use of time points or subsequences as input, this can result in the introduction of a substantial number of classification irrelevant features [7,31] affecting the training of the Transformer due to noise and thus compromising their efficacy for classification.

As a way to address these challenges, we propose the Value and Shape-aware Transformer (VSFormer) with Prior-Enhanced Self-Attention. VSFormer incorporates both discriminative patterns (termed as **shape**) and numerical information (termed as **value**), which enhances the performance in cases where discriminative patterns are lacking, or where both shape and value information are important. Specifically, our model has two branches. The first focuses on learning from the **shapes** in the time series, while the second extracts insights from the raw **values**. The shape branch locates shape tokens from each time series via motif discovery to find repeated patterns by class, allowing for more targeted and accurate extraction of significant shapes. The value branch then partitions the time series into segments across different granularity levels and calculates interval-based statistics to form value tokens. Moreover, class-specific prior information is extracted for each token and used to enrich the encoding process. Such prior information is also used to provide classification-oriented self-attention learning, enhancing classification-relevant features and thus attenuating irrelevant noise. Finally, a decision layer is designed to fuse shape and value representations for the final classification.

To summarize, our work has the following main contributions:
1). We propose VSFormer, a novel approach incorporating both discriminative shapes and numerical information for time series classification.
2). We improve the existing positional encoding and introduce class-specific prior information derived from the training data for both shape and value to enrich it.
3). We propose Prior-Enhanced Self-Attention which is classification-oriented to enhance classification-relevant features and reduce the impact of noise.
4). Extensive experiments were conducted on all 30 UEA archive datasets showing that our model outperforms SOTA models. We also experimented with a real-world application showing the superiority of our model in a case where no discriminative shapes are present.

2 Related Work

2.1 Multivariate Time Series Classification Existing works for MTSC can be roughly categorized into distance-based, pattern-based, and deep-learning-based methods. Distance-based methods rely on measuring the dissimilarity between time series using distance measures such as Euclidean Distance (ED) and Dynamic Time Warping (DTW) [23] and use 1-nearest-neighbor for classification. Pattern-based methods extract bag-of-patterns or discriminative patterns from raw time series. A prominent example of using bag-of-patterns is WEASEL+MUSE [24], which constructs a multivariate feature vector from each variable of the MTS using various sliding windows, extracts discrete features, and undergoes feature selection to remove non-discriminative features. Meanwhile, several approaches have been devoted to using shapelets [27], such as Generalized Random Shapelet Forests (gRSF) [18] that generate shapelet-based decision trees by randomly choosing a subset of shapelets.

More recently, deep learning models have shown notable success in MTSC [22]. MLSTM-FCNs [17] employ a combination of LSTM and stacked CNN layers for feature extraction from time series. TapNet [29] further introduces the attentional prototype learning for fully and semi-supervised MTSC. ROCKET [9] and its optimized counterpart, MiniRocket [10], use random convolutional kernels for time series transformation and subsequently train classifiers on these transformed features.

Several deep-learning models also focus on finding discriminative patterns. For instance, ShapeNet [19] utilizes embedding learning to map subsequences into a unified space and employs clustering to discern these patterns. RLPAM [15] uses reinforcement learning to detect patterns beneficial for classification tasks. Meanwhile, SVP-T [31] uses k-means to capture vast amounts of time series subsequences and feed them into a Transformer encoder. Despite their remarkable performances, these models have the assumption that some discriminative patterns exist in the time series, and overlook scenarios where such patterns are absent in the datasets.

2.2 Transformers for Time Series Classification The success of the Transformer model has inspired numerous researchers to adapt it for time series classification. Beyond SVP-T, which is designed specifically for MTSC, several representation learning methods have taken classification as their downstream task. For example, TS-TCC [13] facilitates unsupervised representation learning from time series data by utilizing temporal and contextual contrasting modules. TST [28] in-

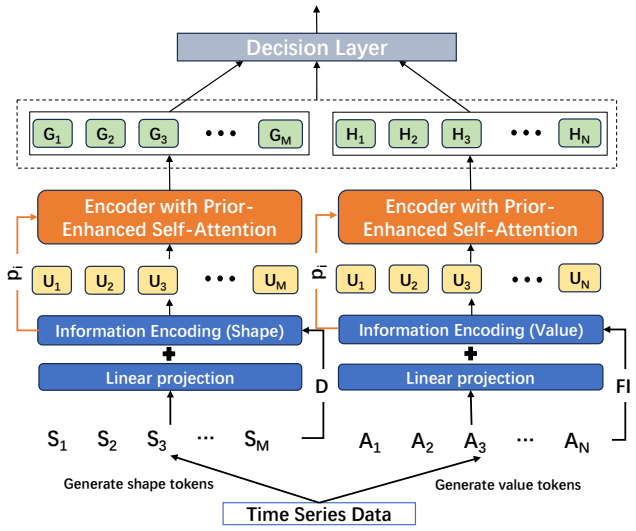


Figure 2: The overall architecture of VSFormer

troduces a novel transformer encoder-based framework for representation learning, while TARNet [8] integrates a task-aware reconstruction strategy to bolster downstream task performance. However, these models introduce many classification-irrelevant features into self-attention learning, potentially compromising the efficacy of time series classification.

3 Problem Formulation

A multivariate time series, $\mathbf{X} = \{X^1, ..., X^V\} \in \mathbb{R}^{V \times T}$, is a collection of several univariate time series, $X^i \in \{x_1, ..., x_T\}$. $T \in \mathbb{Z}^+$ is the total number of observations, and $V \in \mathbb{Z}^+$ represents the number of variables.

An MTS dataset consists of pairs of MTS and associated labels, which can be represented as $\{(\mathbf{X}_1, y_1), ..., (\mathbf{X}_N, y_N)\}$. Here, $y = \{y_1, ..., y_C\} \in \mathbb{R}^C$ denotes the corresponding class, and N is the total count of MTS instances in the dataset.

A multivariate time series classification problem aims to train a classifier that can predict the class label for an unlabeled, previously unseen MTS.

4 Methodology

4.1 Overall Architecture Figure 2 shows the overall structure of VSFormer. The time series dataset is initially subjected to two distinct preprocessing steps for **shape** and **value**, yielding input tokens and class-specific prior information. These tokens are then encoded using our improved Time Series Information (TSI) Encoding and fed into the Transformer encoders with our proposed Prior-Enhanced Self-Attention (PESA). Finally, the representations from the two encoding branches are fused by the decision layer

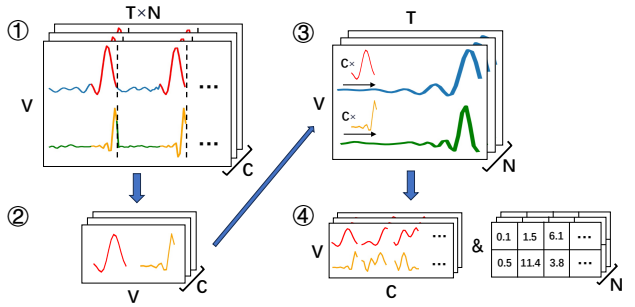


Figure 3: The shape token generation process (with $k = 1$ for clarity). Steps: ① Concatenate sequences per class by variable and identify repeated patterns (highlighted in red and orange). ② Extract prototype shapes. ③ Conduct a similarity search for each instance. ④ Generate a set of shapes alongside their associated distances.

obtaining the final results.

4.2 Input Token Generation In this section, we outline the data preprocessing steps for generating both shape and value tokens.

Shape Tokens. Figure 3 shows the shape token generation process. We employ STOMP [30], a well-known motif (repeated patterns) discovery algorithm to capture significant shapes within the data. All time series from the same class are concatenated by their variable, which results in a lengthy multivariate sequence represented as $\mathcal{X}^{v,c} = \{\mathbf{X}_1^{v,c}, ..., \mathbf{X}_N^{v,c}\}$, where v denotes the variable, and c denotes the class. Subsequently, we search for the top-k motif pairs within this extended sequence (for illustrative clarity, in Figure 3, Step 2, we set $k = 1$). This results in a total of $\mathcal{M} = k \times V \times C$ motif pairs. From each motif pair, we choose one motif instance, denoted by $\hat{S}^{v,c}$, as the prototype shape specific to its variable and class. The intuition is to identify a small set of patterns (shapes) representing each variable per class. Next, we search within each time series instance in the corresponding variable to locate subsequences similar to these prototype shapes (Figure 3, Step 3). Thus, for the i-th instance, we acquire shape token set $\mathbf{S} = \{S^{1,1}, S^{1,2}, ..., S^{V,C}\}$ and their associated Z-normalized Euclidean Distances, represented as $\mathbf{D} = \{D^{1,1}, D^{1,2}, ..., D^{V,C}\}$ (Figure 3, Step 4). To simplify notation, these are rewritten as $\mathbf{S} = \{S_1, S_2, ..., S_{\mathcal{M}}\}$ and $\mathbf{D} = \{D_1, D_2, ..., D_{\mathcal{M}}\}$.

Value Tokens. Individual time points within a time series are insufficient for revealing meaningful information associated with class labels [7]. To address this problem, instead of utilizing all time points as inputs,

we propose to derive value tokens by calculating statistical features from the various time series intervals. The simplest approach is to divide the time series into w equal-length intervals. However, since the optimal choice of w (or equivalently, the length of the interval) is unknown, to ensure capturing a sufficient amount of information from the time series, we propose a multi-granularity approach, by iterating through the different values of w ranging from 1 to $M \in \mathbb{Z}^+$, and segmenting each time series variable, X^v, into M sets of equal-length intervals. Each set of intervals represents a different granularity level $\mathcal{G}^w$. Note that when $w = 1$, the interval is the whole time series.

For each interval, we calculate three statistical features: mean (μ), standard deviation (σ), and slope (ψ), which is derived from fitting a linear regression to the subsequence. Notably, [11,22] have corroborated the efficacy of these features in time series classification.

Through this method, we generate a set of value tokens for each instance, symbolized as $\mathbf{A} = \{A^1_{1,\mu}, A^1_{1,\sigma}, A^1_{1,\psi}, ..., A^w_{1,\mu}, A^w_{1,\sigma}, A^w_{1,\psi}, ..., A^M_{M,\mu}, A^M_{M,\sigma}, A^M_{M,\psi}\}$ that encapsulates crucial statistical information about the different intervals within the time series. For a multivariate time series, we have a total of $\mathcal{N} = V \times 3 \times [(1 + M) \times M/2]$ statistical features. The value token set $\mathbf{A}$ could then be rewritten as $\mathbf{A} = \{A_1, A_2, ..., A_\mathcal{N}\}$ for simplicity.

4.3 Class-Specific Prior Information Class-specific prior information corresponds to each input token and can be directly calculated from supervised information. We introduce it to enrich the encoding process and guide self-attention learning to enhance classification-relevant features and reduce noise.

Prior information for shape. For shape tokens, we use the corresponding distances of the shapes to calculate the weight as their prior information.

First, we compute the weights for prototype shapes. For each prototype shape $\hat{S}^{v,c}$, we compute two distinct types of distances, D_1 and D_2. The former, D_1, represents the mean intra-class distance for each prototype shape and is computed by averaging all distances within the same class c. In contrast, D_2 describes the mean inter-class distance and is determined by averaging all distances corresponding to different classes. Using both, we calculate a ratio $\hat{D} = \frac{D_2}{D_1+D_2}$ that quantifies the distinctiveness of the prototype shapes across classes. A value of $\hat{D} > 0.5$ indicates that the shape is discriminative and significantly contributes to classification. Conversely, $\hat{D} = 0.5$ implies that the shape is common across all classes and does not contribute to discrimination. Note that $\hat{D} < 0.5$ indicates that the prototype shape is a repeated pattern that does not belong to its class, which is obviously contrary to our method and is therefore unlikely to happen.

We then assign a weight $w_{\hat{S}}$ to each prototype shape $\hat{S}^{v,c}$ through the following computations:

$$x = \max(\hat{D} - 0.5, 0), \tag{4.1}$$

$$w_{\hat{S}} = e^{\alpha x},\ \alpha \geq 0, \tag{4.2}$$

which ensures that prototype shapes with higher discriminative power (larger $\hat{D}$) receive greater weights.

Subsequently, for each shape token $S^{v,c}_i$, we determine its distance d to the prototype shape $\hat{S}^{v,c}$ and assign it a weight w_{S_i} using the formula:

$$w_{S_i} = \beta e^{-d} + 1,\ \beta \geq 0, \tag{4.3}$$

which ensures the less similar the shape, the smaller the weight.

Finally, we compute the final weight $w^{v,c,i}_{\text{final}}$, which serves as the prior information for each shape token:

$$p_i = w^{v,c,i}_{\text{final}} = w_{\hat{S}^{v,c}} \times w_{S^{v,c}_i}. \tag{4.4}$$

The procedure ensures that a shape token more similar to a discriminative prototype shape receives a higher weight. Note that α and β are weighting hyperparameters. Setting either one to zero will remove the influence of its corresponding weight on the final weights.

Prior information for value. Generating prior information for value tokens involves an entropy-based feature importance calculation. Given the set of value tokens $\mathbf{A}$ and corresponding class labels from the training set, we first compute the entropy $H(A)$, quantifying the uncertainty associated with these tokens. The entropy is defined as:

$$H(A) = -\sum_{i=1}^{C} Pr(i|A) \log_2 Pr(i|A), \tag{4.5}$$

where $Pr(i|A)$ represents the probability of class i given a value token A, and C denotes the total number of classes.

Next, the conditional entropy $H(A|Y)$ measures the average entropy of A given the class label Y:

$$H(A|Y) = -\sum_{j=1}^{C} Pr(Y = j) \sum_{i=1}^{C} Pr(i|A, Y = j) \times \log_2 Pr(i|A, Y = j), \tag{4.6}$$

where $Pr(Y = j)$ is the probability of class j, and $Pr(i|A, Y = j)$ is the probability of class i given a value token A and class label $Y = j$.

The feature importance, denoted by $FI(A)$, is computed using the gain in entropy, which indicates the reduction in uncertainty about class labels provided by the value token A:

$$FI(A) = H(A) - H(A|Y). \tag{4.7}$$

The feature importance $FI(A)$ is then used as the prior information for the value token A, where higher feature importance indicates a greater relevance of the value token for the classification task.

4.4 Time Series Information Encoding While the relative position of shapes within a time series instance is often more crucial than sequence order, the authors of SVP-T [31] proposed a positional encoding scheme to encode the variable ID, start timestamp, and end timestamp of a shape, with all information normalized to the range of [0, 1] (by dividing them with the number of variables or time series length). This approach, however, exhibits a limitation: While variable information is inherently discrete, normalization transforms it into a continuous value, leading to artificial ordering. For instance, with three variables, the positional encoding scheme normalizes them to 1/3, 2/3, and 1, respectively. As a result, the third variable appears numerically three times greater than the first, which is a misleading representation. To preserve the discreteness of variable information, we implement binary encoding to transform the variables into binary digits B_i. We further extend this approach by incorporating class-specific prior information into the encoding process. As a result, we introduce Time Series Information (TSI) Encoding. The structure of TSI Encoding for the i-th token is outlined in Table 1.

Table 1: Time Series Information Encoding

Variable	Start timestamp	End timestamp	Prior
B_i	$t_{i,start}/T$	$t_{i,end}/T$	p_i

4.5 Encoding Layer In the encoding layer, the output from TSI Encoding, I_i, is passed through a linear transformation. Simultaneously, the input termed $Token$ is processed by a distinct linear projection layer. The outcomes from both processes are then added to form the input U_i for the Transformer Encoder. The process can be represented as:

$$U_i = I_i W_I + W_S Token, \tag{4.8}$$

where $W_I \in \mathbb{R}^{d_I \times d_{model}}$ and $W_S \in \mathbb{R}^{d_{model} \times d_{Token}}$ are trainable weights, d_{model} denotes the input dimension of the Transformer Encoder, while d_I and d_{Token} represent the dimensions of I_i and $Token$, respectively.

4.6 Prior-Enhanced Self-Attention Here, we introduce a time series classification-oriented self-attention, Prior-Enhanced Self-Attention (PESA) mechanism, which incorporates class-specific prior information into the attention computation to enhance classification-relevant features and attenuate noise.

As presented by [25], the conventional self-attention mechanism employs the query matrix Q, key matrix K, and value matrix V. The attention score matrix A is given by:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right), \tag{4.9}$$

where d is the dimensionality of the queries and keys.

We bring in the prior score matrix P, constructed using class-specific prior information p_i. Specifically, every element $P_{i,j}$ of matrix P is characterized as:

$$P_{i,j} = \begin{cases} p_i \cdot p_j & \text{if } i \neq j \\ 1 & \text{if } i = j. \end{cases} \tag{4.10}$$

This matrix P effectively captures the interactions among different input tokens, emphasizing those that are highly informative for classification within the attention mechanism.

The final PESA is deduced by taking an element-wise product (represented by $\otimes$) of the attention score matrix A and the prior score matrix P, followed by a softmax operation, and finally multiplying the resulting matrix with the value matrix V:

$$\text{PESA}(Q, K, V, P) = \text{softmax}(A \otimes P)V. \tag{4.11}$$

By integrating class-specific prior information, we infuse the supervised knowledge externally, which optimizes the self-attention learning process, enhancing its performance for time series classification.

4.7 Decision Layer The decision layer plays an instrumental role in the fusion of shape and value representations learned from previous network layers. These representations, denoted as R_{shape} and R_{value}, are transformed into class probability spaces, G and H, respectively, through two distinct linear layers:

$$G = W_{\text{shape}} R_{\text{shape}}, \tag{4.12}$$

$$H = W_{\text{value}} R_{\text{value}}, \tag{4.13}$$

where W_{shape} and W_{value} are the learnable weights for the shape and value representations, correspondingly.

We introduce a balancing factor λ to regulate the contribution of shape and value information toward the final decision. This factor is computed by passing the concatenated R_{shape} and R_{value} through a linear layer, followed by a sigmoid activation function:

$$\lambda = \sigma(W_\lambda[R_{\text{shape}}, R_{\text{value}}]), \tag{4.14}$$

where W_λ represents the learnable weight and σ is the sigmoid activation function.

The layer then computes the final class probabilities O as a λ-weighted combination of G and H, followed by a softmax operation:

$$O = \text{softmax}(\lambda G + (1 - \lambda)H). \tag{4.15}$$

By learning λ from both shape and value representations, the model can adaptively adjust the emphasis on these aspects depending on their relevance to the classification task. That is, if the shape information is crucial, λ will be closer to 1, giving higher weight to G. Conversely, if the value information is more important, λ will approach 0, and the influence of H will be stronger.

5 Experiments

5.1 Datasets We evaluate our method on all 30 datasets from the well-known UEA MTSC archive [5]. We also conduct a case study using the extensive *Space Weather Analytics for Solar Flares (SWAN-SF)* dataset [3] in Section 5.7.

5.2 Comparison Methods We compare our proposed approach with three benchmark methods [5] in MTSC: (1) **EDI**, the 1-nearest neighbor with Euclidean Distance, (2) **DTWI**, Dimension-Independent Dynamic Time Warping, and (3) **DTWD**, Dimension-Dependent Dynamic Time Warping. We also compare with eleven SOTA methods: (1) **WEASEL+MUSE** [24], the state-of-the-art bag-of-patterns model which extracts features into word representations, (2) **SRL** [14], a representation learning approach using negative sampling with an encoder network structure, (3) **MLSTM-FCNs** [17], a deep learning framework combining an LSTM layer and the FCN layer with Squeeze-and-Excitation mechanism, (4) **TapNet** [29], an attentional prototypical network for semi- and fully-supervised learning in MTSC, (5) **ShapeNet** [19], a network which projects subsequences into a unified space through embedding learning and uses clustering to find discriminative patterns, (6) **TARNet** [8], a network which integrates a task-aware reconstruction strategy to bolster downstream task performance, (7) **ROCKET** [9], a method using random convolutional kernels to transform time series and using the transformed features to train classifier, (8) **MiniRocket** [10], a streamlined version of ROCKET, offers faster computation by optimizing the feature transformation process without compromising accuracy. (9) **RLPAM** [15], the network uses reinforcement learning to find patterns that can provide useful information for classifiers. (10) **TST** [28], a timestamp-level Transformer model which uses per-time-point input mechanism, (11) **SVP-T** [31], the SOTA Transformer-based model in MTSC which uses shapes as the input.

5.3 Experiment Settings We conduct our experiments on a machine equipped with an AMD EPYC Rome 7742 CPU @ 2.60 GHz and an NVIDIA Tesla A100 GPU. Our implementation utilizes Python 3.8.6 and Pytorch 2.0.0. For the UEA datasets, we adhere to the original splits for the training set and test set. In the case of the solar flare dataset, we allocate 80% for training and the remaining 20% for testing. Across all experiments, we reserve 20% of the training set for validation, aiding in hyperparameter tuning. The details of hyperparameters are shown in Appendix A.2. We use classification accuracy as the evaluation metric, and report the average rank and the standard deviation (STD) of ranks to show the stability of the model on different datasets. We also show how many wins/ties our model has compared to others to show performance superiority. To underscore the statistical significance, we employ the Wilcoxon signed-rank test.

5.4 Experimental Results All the results for the baseline methods are sourced from the original papers or the survey paper [22], except for the results of TST, which are sourced from [31], and those of both ROCKET and MiniRocket, which are obtained from [15]. All the best are highlighted in bold. Any result denoted by "N/A" means that it is not reported in the original paper or cannot be produced.

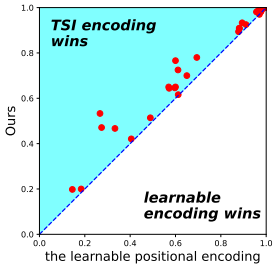
From Table 2, we can observe that our method outperforms all SOTA methods with the best average rank (4.233). Moreover, our method has the lowest STD of ranks (2.294), indicating stable performance across all datasets with different characteristics. This stability can be attributed to the integration of both value and shape information in our method. It is worth noting that while EDI also exhibits a low STD of ranks, its high average rank suggests consistently poor performance across all tested datasets. In addition, methods such as RLPAM and TST demonstrate excellent results on specific datasets but have high STD of ranks (4.640 and 5.049, respectively), which indicates instability in their performance across different datasets, likely due to their specific focuses—RLPAM on pattern finding and

Table 2: Results of our method and baseline methods on all 30 UEA archive datasets

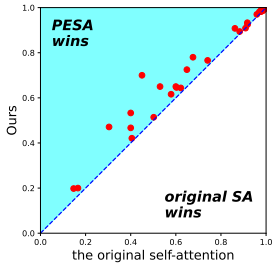
	EDI	DTWI	DTWD	MLSTM -FCNs	WEASEL +MUSE	SRL	TapNet	ShapeNet	TARNet	ROCKET	MiniR	RLPAM	TST	SVP-T	Ours
ArticularyWordRecognition	0.970	0.980	0.987	0.973	0.990	0.987	0.987	0.987	0.977	**0.993**	0.992	0.923	0.983	**0.993**	**0.993**
AtrialFibrillation	0.267	0.267	0.220	0.267	0.333	0.133	0.333	0.400	**1.000**	0.067	0.133	0.733	0.200	0.400	0.467
BasicMotions	0.676	**1.000**	0.975	0.950	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.975	**1.000**	**1.000**
CharacterTrajectories	0.964	0.969	0.989	0.985	0.990	0.994	**0.997**	0.980	0.994	N/A	0.993	0.978	N/A	0.990	0.991
Cricket	0.944	0.986	**1.000**	0.917	**1.000**	0.986	0.958	0.986	**1.000**	**1.000**	0.986	**1.000**	0.958	**1.000**	**1.000**
DuckDuckGeese	0.275	0.550	0.600	0.675	0.575	0.675	0.575	0.725	**0.750**	0.520	0.650	0.700	0.480	0.700	0.700
EigenWorms	0.549	N/A	0.618	0.504	0.890	0.878	0.489	0.878	0.420	0.901	**0.962**	0.908	N/A	0.923	0.725
Epilepsy	0.666	0.978	0.964	0.761	**1.000**	0.957	0.971	0.987	**1.000**	0.993	**1.000**	0.978	0.920	0.986	0.986
ERing	0.133	0.133	0.133	0.133	0.133	0.133	0.133	0.133	0.919	**0.981**	**0.981**	0.819	0.933	0.937	0.970
EthanolConcentration	0.293	0.304	0.323	0.373	0.430	0.236	0.323	0.312	0.323	0.380	0.468	0.369	0.337	0.331	**0.471**
FaceDetection	0.519	N/A	0.529	0.545	0.545	0.528	0.556	0.602	0.641	0.630	0.620	0.621	**0.681**	0.512	0.646
FingerMovements	0.550	0.520	0.530	0.580	0.490	0.540	0.530	0.580	0.620	0.530	0.550	0.640	**0.776**	0.600	0.650
HandMovementDirection	0.278	0.306	0.231	0.365	0.365	0.270	0.378	0.338	0.392	0.446	0.392	**0.635**	0.608	0.392	0.514
Handwriting	0.200	0.316	0.286	0.286	**0.605**	0.533	0.357	0.452	0.281	0.585	0.507	0.522	0.305	0.433	0.421
Heartbeat	0.619	0.658	0.717	0.663	0.727	0.737	0.751	0.756	0.780	0.726	0.771	0.779	0.712	**0.790**	0.766
InsectWingbeat	0.128	N/A	N/A	0.167	N/A	0.160	0.208	0.250	0.137	N/A	0.595	0.352	**0.684**	0.184	0.200
JapaneseVowels	0.924	0.959	0.949	0.976	0.973	0.989	0.965	0.984	0.992	0.965	0.989	0.935	**0.994**	0.978	0.981
Libras	0.833	0.894	0.870	0.856	0.878	0.867	0.850	0.856	**1.000**	0.906	0.922	0.794	0.844	0.883	0.894
LSST	0.456	0.575	0.551	0.373	0.590	0.558	0.568	0.590	**0.976**	0.639	0.643	0.643	0.381	0.666	0.616
MotorImagery	0.510	N/A	0.500	0.510	0.500	0.540	0.590	0.610	0.630	0.560	0.550	0.610	N/A	**0.650**	**0.650**
NATOPS	0.850	0.850	0.883	0.889	0.870	0.944	0.939	0.883	0.911	0.894	0.928	**0.950**	0.900	0.906	0.933
PenDigits	0.973	0.939	0.977	0.978	0.948	**0.983**	0.980	0.977	0.976	0.982	N/A	0.982	0.974	**0.983**	**0.983**
PEMS-SF	0.705	0.734	0.711	0.699	N/A	0.688	0.751	0.751	**0.936**	0.832	0.522	0.632	0.919	0.867	0.78
Phoneme	0.104	0.151	0.151	0.110	0.190	0.246	0.175	**0.298**	0.165	0.280	0.292	0.175	0.088	0.176	0.198
RacketSports	0.868	0.842	0.803	0.803	0.934	0.862	0.868	0.882	**0.987**	0.921	0.868	0.868	0.829	0.842	0.908
SelfRegulationSCP1	0.771	0.765	0.775	0.874	0.710	0.846	0.652	0.782	0.816	0.846	**0.925**	0.802	**0.925**	0.884	**0.925**
SelfRegulationSCP2	0.483	0.533	0.539	0.472	0.460	0.556	0.550	0.578	0.622	0.540	0.522	0.632	0.589	0.600	**0.644**
SpokenArabicDigits	0.967	0.959	0.963	0.990	0.982	0.956	0.983	0.975	0.985	**0.998**	0.993	0.621	0.993	0.986	0.982
StandWalkJump	0.200	0.333	0.200	0.067	0.333	0.400	0.400	0.533	0.533	0.530	0.333	**0.667**	0.267	0.467	0.533
UWaveGestureLibrary	0.881	0.868	0.903	0.891	0.916	0.884	0.894	0.906	0.878	0.938	0.938	**0.944**	0.903	0.941	0.909
Average rank	12.633	11.567	10.800	10.517	8.500	8.533	8.467	7.033	5.517	7.000	5.783	6.167	8.817	5.183	**4.233**
STD of ranks	2.442	3.194	2.645	3.253	4.365	4.017	3.091	2.918	4.362	4.129	4.024	4.640	5.049	3.068	**2.294**
Ours 1-to-1 wins/ties	30	30	30	29	25	24	26	23	19	19	17	20	23	23	-
Wilcoxon Test p-value	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.006	0.174	0.056	0.175	0.286	0.005	0.046	-

Table 3: Ablation study of each branch in VSFormer

	Value only	Shape only	VSFormer
Average accuracy	0.643	0.693	**0.748**
Average rank	2.533	2.317	**1.150**



(a) Our method versus the learnable positional encoding.



(b) Our method versus the original self-attention.

Figure 4: Ablation studies showing the comparative performance of our method with different configurations.

TST on using single points as input. Meanwhile, our method wins/ties the two SOTA transformer methods, TST and SVP-T, in 23 out of 30 datasets, achieving wins or ties on the majority of datasets. The Wilcoxon signed-rank test indicates that our method is significantly better than three benchmark methods (EDI, DTWI, and DTWD) and seven state-of-the-art methods (MLSTM-FCNs, WEASEL+MUSE, SRL, TapNet, ShapeNet, TST and SVP-T) with a significance level of $p < 0.05$.

5.5 Ablation Study We explore the impact of distinct components within our proposed method, concentrating on comparing our TSI encoding with learnable positional encoding, and our PESA against the original self-attention. Additionally, we study the effectiveness of combining value and shape information in our model.

Initially, we experiment with the learnable positional encoding, in place of our TSI encoding, across all 30 UEA archive datasets. As visualized in Figure 4a, our method wins/draws (as indicated by red dots in the cyan region) in terms of accuracy on all datasets, showing the superiority of our proposed TSI encoding in Section 4.4. Subsequently, we replace our PESA with the vanilla self-attention mechanism [25] and conduct the experiments. As depicted in Figure 4b, all red dots reside in the cyan region, further illustrating the effectiveness of our novel self-attention in Section 4.6 over the original one.

We justify the effectiveness of incorporating both shape and value through ablation study in Table 3. For Value only and Shape only, we remove the decision layer and only use the value and the shape branch, re-

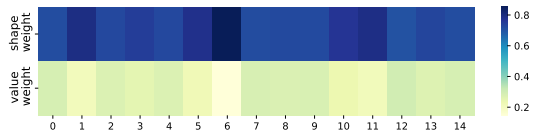


Figure 5: The heat map illustrates the distribution of the shape weight (λ) and value weight ($1-\lambda$) for each instance in the AtrialFibrillation dataset.

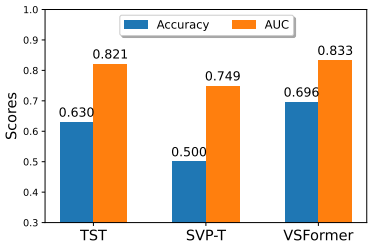


Figure 6: Performance comparison of TST, SVP-T, and VSFormer regarding the accuracy and AUC on the dataset.

spectively. Table 3 shows that VSFormer has the highest average accuracy and average rank, indicating the effectiveness of combining value and shape information for time series classification.

5.6 Effectiveness Analysis We investigate the effectiveness of λ in the decision layer. As described earlier, λ in Formula 4.14 and 4.15 regulates the contribution of shape and value information in the final decision, allowing the model to emphasize the information that is more informative.

We test on the AtrialFibrillation dataset, which comprises two-channel ECG recordings, aiming to predict spontaneous termination of atrial fibrillation [5]. The dataset has three classes: non-termination, self-terminating at least one minute after recording, and immediate termination within one second of recording end. [19] has highlighted the presence of discriminative patterns that can effectively distinguish between classes. This observation is corroborated by the learning of λ in our model. As illustrated in the heat map in Figure 5, the colors corresponding to the shape weights (λ) are notably darker than those for value weights ($1-\lambda$) across all 15 test set samples of AtrialFibrillation. This indicates that in every sample, the shape information carries more weight in determining the final classification results compared to the value information.

5.7 Case Study: Solar Flare Detection To further illustrate the superiority of VSFormer in dealing with data that do not have clear discriminative patterns, we use the *SWAN-SF* dataset, as discussed earlier. For data preprocessing, we employ linear interpolation to manage missing data and median smoothing to mitigate artificial spikes. Samples are extracted and categorized into four classes according to flare level using Sample Types sampling [2]. Each feature undergoes global Min-Max normalization. Since the dataset is extremely imbalanced, we implement undersampling to equalize sample numbers across classes. More details can be found in Appendix A.1.

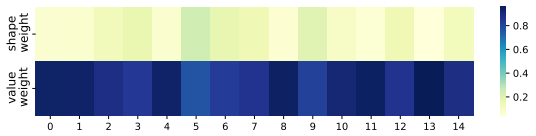


Figure 7: The heat map for 15 randomly selected test samples in the SWAN-SF dataset.

Notably, most existing literature uses the timestamp-level method [1, 6, 16, 20], owing to the dataset's lack of discriminative patterns. We evaluate the performance of TST, SVP-T, and VSFormer on the dataset, and assess both accuracy and Area Under the Curve (AUC) metrics.

As depicted in Figure 6, VSFormer outperforms the other two in both metrics, and TST outperforms SVP-T. The superior performance of TST over SVP-T can be attributed to its timestamp-level basis, while SVP-T relies on shape for time series representation—a challenge given the elusive nature of discriminative shapes in SWAN-SF data. VSFormer integrates both shape and value information, emphasizing value information as evidenced in Figure 7 (the color of the value weight is markedly darker than the shape weight), which results in better performance than SVP-T. Moreover, the benefits of TSI encoding and PESA in VSFormer over the learnable positional encoding and traditional self-attention employed by TST also explain VSFormer's superior performance.

6 Conclusion

In this work, we introduce VSFormer, a value and shape-aware Transformer tailored for MTSC. The model incorporates both discriminative patterns and numerical information, enhancing the performance in cases where discriminative patterns are lacking. In addition to using class-specific prior information to extend the encoding layer, we introduce it into self-attention learning to enhance classification-relevant features and reduce the impact of noise. Extensive experiments on all 30 UEA archives demonstrate the performance superiority of our model over SOTA models. The case study also shows that our model has superior performance in the case of the absence of discriminative patterns.

References

[1] A. Ahmadzadeh, B. Aydin, M. K. Georgoulis, D. J. Kempton, S. S. Mahajan, and R. A. Angryk. How to train your flare prediction model: Revisiting robust sampling of rare events. *The Astrophysical Journal Supplement Series*, 254(2):23, 2021.

[2] A. Alvarez. *Improving Solar Flare Forecasting Based On Time Series Magnetic Features Using Advanced Machine Learning*. PhD thesis, George Mason University, Fairfax, Virginia, USA, Nov. 2022.

[3] R. Angryk, P. Martens, B. Aydin, D. Kempton, S. Mahajan, S. Basodi, A. Ahmadzadeh, X. Cai, S. Filali Boubrahimi, S. M. Hamdi, M. Schuh, and M. Georgoulis. SWAN-SF, 2020.

[4] R. A. Angryk, P. C. Martens, B. Aydin, D. Kempton, S. S. Mahajan, S. Basodi, A. Ahmadzadeh, X. Cai, S. Filali Boubrahimi, S. M. Hamdi, et al. Multivariate time series dataset for space weather data analytics. *Scientific data*, 7(1):227, 2020.

[5] A. Bagnall, H. A. 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.

[6] M. G. Bobra and S. Couvidat. Solar flare prediction using sdo/hmi vector magnetic field data with a machine-learning algorithm. *The Astrophysical Journal*, 798(2):135, 2015.

[7] M. Cheng, Q. Liu, Z. Liu, Z. Li, Y. Luo, and E. Chen. Formertime: Hierarchical multi-scale representations for multivariate time series classification. In *Proceedings of the ACM Web Conference 2023*, pages 1437–1445, 2023.

[8] R. R. Chowdhury, X. Zhang, J. Shang, R. K. Gupta, and D. Hong. Tarnet: Task-aware reconstruction for time-series transformer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 212–220, 2022.

[9] A. Dempster, F. Petitjean, and G. I. Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.

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

[11] H. Deng, G. Runger, E. Tuv, and M. Vladimir. A time series forest for classification and feature extraction. *Information Sciences*, 239:142–153, 2013.

[12] 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. In *International Conference on Learning Representations*, 2020.

[13] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan. Time-series representation learning via temporal and contextual contrasting. In *In the Proceeding of 31th International Joint Conference on Artificial Intelligence*, pages 2352–2359, 2021.

[14] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

[15] G. Gao, Q. Gao, X. Yang, M. Pajic, and M. Chi. A reinforcement learning-informed pattern mining framework for multivariate time series classification. In *In the Proceeding of 31th International Joint Conference on Artificial Intelligence*, 2022.

[16] A. Ji, B. Aydin, M. K. Georgoulis, and R. Angryk. All-clear flare prediction using interval-based time series classifiers. In *2020 IEEE International Conference on Big Data*, pages 4218–4225. IEEE, 2020.

[17] F. Karim, S. Majumdar, H. Darabi, and S. Harford. Multivariate lstm-fcns for time series classification. *Neural networks*, 116:237–245, 2019.

[18] I. Karlsson, P. Papapetrou, and H. Boström. Generalized random shapelet forests. *Data mining and knowledge discovery*, 30:1053–1085, 2016.

[19] G. Li, B. Choi, J. Xu, S. S. Bhowmick, K.-P. Chun, and G. L.-H. Wong. Shapenet: A shapelet-neural network approach for multivariate time series classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8375–8383, 2021.

[20] H. Liu, C. Liu, J. T. Wang, and H. Wang. Predicting solar flares using a long short-term memory network. *The Astrophysical Journal*, 877(2):121, 2019.

[21] T. Rakthanmanon and E. Keogh. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *proceedings of the 2013 SIAM International Conference on Data Mining*, pages 668–676. SIAM, 2013.

[22] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 35(2):401–449, 2021.

[23] H. Sakoe. Dynamic-programming approach to continuous speech recognition. In *1971 Proc. the International Congress of Acoustics, Budapest*, 1971.

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

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[26] N. Wang, J. Zhou, J. Li, B. Han, F. Li, and S. Chen. Hardenvr: Harassment detection in social virtual reality. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 94–104. IEEE, 2024.

[27] L. Ye and E. Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956, 2009.

[28] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty,

and C. Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.

[29] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6845–6852, 2020.

[30] Y. Zhu, Z. Zimmerman, N. S. Senobari, C.-C. M. Yeh, G. Funning, A. Mueen, P. Brisk, and E. Keogh. Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins. In *2016 IEEE 16th international conference on data mining*, pages 739–748. IEEE, 2016.

[31] R. Zuo, G. Li, B. Choi, S. S. Bhowmick, D. N.-y. Mah, and G. L. Wong. Svp-t: a shape-level variable-position transformer for multivariate time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11497–11505, 2023.