Learning Soft Sparse Shapes for Efficient Time-Series Classification

Zhen Liu [1 2] **Yicheng Luo** [1] **Boyuan Li** [1] **Emadeldeen Eldele** [2] **Min Wu**[† 2] **Qianli Ma**[† 1]

Abstract

Shapelets are discriminative subsequences (or shapes) with high interpretability in time series classification. Due to the time-intensive nature of shapelet discovery, existing shapelet-based methods mainly focus on selecting discriminative shapes while discarding others to achieve candidate subsequence sparsification. However, this approach may exclude beneficial shapes and overlook the varying contributions of shapelets to classification performance. To this end, we propose a **Soft** sparse **Shape**s (**SoftShape**) model for efficient time series classification. Our approach mainly introduces soft shape sparsification and soft shape learning blocks. The former transforms shapes into soft representations based on classification contribution scores, merging lower-scored ones into a single shape to retain and differentiate all subsequence information. The latter facilitates intra- and inter-shape temporal pattern learning, improving model efficiency by using sparsified soft shapes as inputs. Specifically, we employ a learnable router to activate a subset of class-specific expert networks for intra-shape pattern learning. Meanwhile, a shared expert network learns inter-shape patterns by converting sparsified shapes into sequences. Extensive experiments show that SoftShape outperforms state-of-the-art methods and produces interpretable results. Our source code is available at https://github.com/qianlima-lab/SoftShape.

1. Introduction

Time series classification (TSC) is a critical task with various real-world applications, such as human activity recognition (Lara & Labrador, 2012) and medical diagnostics (Wang et al., 2024). Unlike traditional data types, time series data consists of ordered numerical observations with temporal dependencies, which is often difficult for human intuition to understand. Recent studies (Mohammadi Foumani et al., 2024; Middlehurst et al., 2024; Jin et al., 2024; Ma et al., 2024) indicate that deep neural networks showed remarkable success in TSC, even without specialized knowledge. However, their black-box nature is a barrier to adopting effective models in critical applications such as healthcare, where insights about model decisions are important. Therefore, enhancing the interpretability of these models for TSC is a crucial ongoing issue.

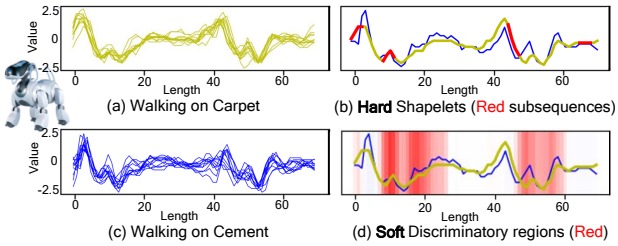


Figure 1: (a) Yellow and (c) Blue lines represent two time series class samples from the *SonyAIBORobotSurface1* UCR dataset. (b) Red subsequences denote shapelets for the "Walking on Carpet" class. (d) Red regions show discriminative parts of the "Walking on Carpet" class, with darker red indicating greater discriminative power.

Shapelets, which are discriminative subsequences that characterize target classes, offer a promising approach towards interpretable TSC models (Ye & Keogh, 2009). For instance, Figures 1(a) and 1(c) show time series classes of a robotic dog walking on carpet and cement (Vail & Veloso, 2004). The red subsequences shown in Figure 1(b) illustrate shapelets, as selected by Hou et al. (2016). However, identifying these shapelets is usually computationally intensive (Rakthanmanon & Keogh, 2013), as it requires evaluating subsequences across varying positions and lengths.

Existing methods (Grabocka et al., 2014; Li et al., 2020) address this issue via a shapelet transformation strategy for candidate subsequence sparsification. Specifically, they convert each subsequence into a feature that quantifies the distance between a time series sample and the subsequence (Ma et al., 2020). Subsequently, this strategy was followed by many methods (Li et al., 2021; Le et al., 2024) to learn

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China [2]Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. Correspondence to: Qianli Ma <qianlima@scut.edu.cn>, Min Wu <wumin@i2r.a-star.edu.sg>.

shapelet representations, thereby enhancing the model's interpretability. While efficient, this approach discards many subsequences from the entire time series in a *hard* manner, those that could be beneficial for the classification tasks. In addition, this approach fails to account for the varying importance of shapelets to classification. As shown in Figure 1(b), this leads to poor capture of class patterns in the third and fourth subsequences due to minimal class differences, while omitting many informative regions.

Recent advances in time series patch tokenization (Nie et al., 2023; Bian et al., 2024), i.e., converting time series into subsequence-based tokens, and in the mixture of experts (MoE) architectures (Riquelme et al., 2021; Fedus et al., 2022) inspired a new direction. In computer vision, MoE routers dynamically assign input patches to specialized experts, which can improve both efficiency and class-specific feature learning (Chen et al., 2022; Chowdhury et al., 2023). By analogy, time series shapelets, which can exhibit intra-class similarity and inter-class distribution, can act as patch tokens, with MoE enabling adaptive focus on the most discriminative subsequences. Yet, existing shapelet-based methods overlooked such an approach.

In this paper, we propose the **Soft** sparse **Shape**s (**SoftShape**) model for TSC, which replaces hard shapelet sparsification with soft shapelets. Specifically, we introduce a soft shape sparsification mechanism to reduce the number of learning shapes, thereby enhancing the efficiency of model training. To clarify our motivation, Figure 1(d) shows the classification results of InceptionTime (Ismail Fawaz et al., 2020) using multiple-instance learning (Early et al., 2024). Unlike the *hard* way depicted in Figure 1(b), our *soft* shapelet sparsification assigns weights to shapes based on their classification contribution scores, effectively preserving and differentiating all subsequence information from the time series. Furthermore, we enhance the discriminability of the learned soft shapes by combining: (1) Intra-shape patterns via MoE, where experts specialize in class-specific shapelet types; and (2) Inter-shape dependencies via a shared expert that models temporal relationships among sparsified shapes.

In summary, the major contributions are as follows:

- We propose SoftShape, a soft shapelet sparsification approach for TSC. This weighted aggregation of subsequences is based on classification contribution scores, avoiding information loss from hard filtering.
- We introduce a dual-pattern shapelet learning approach based on an MoE-driven intra-shape specialization and sequence-aware inter-shape modeling, improving the discriminative power of learned shape embeddings.
- We conduct extensive experiments on 128 UCR time series datasets, demonstrating the superior performance of our proposed method against state-of-the-art approaches while showcasing interpretable results.

2. Related Work

2.1. Deep Learning for Time-Series Classification

Recently, deep learning has received significant attention from scholars in time series classification tasks due to its powerful feature extraction capabilities (Ismail Fawaz et al., 2020; Zhang et al., 2024). For example, Ismail Fawaz et al. (2019) and Mohammadi Foumani et al. (2024) have systematically reviewed deep learning approaches for time series classification, focusing on different types of deep neural networks and deep learning paradigms, respectively. In addition, Middlehurst et al. (2024) and Ma et al. (2024) have conducted extensive experiments to evaluate recent time series classification methods. Their findings demonstrate that deep learning models can be effectively applied to time series classification tasks. Despite their strong performance, these deep learning-based time series classification methods often struggle to improve the interpretability of classification results.

2.2. Time-Series Classification with Shapelets

Ye & Keogh (2009) first introduced shapelets, providing good interpretability for time series classification results. Early methods (Mueen et al., 2011; Rakthanmanon & Keogh, 2013) selected the most discriminative subsequences as shapelets using an evaluation function, but the sparsification process was computationally intensive due to the numerous candidate subsequences. To tackle this problem, Hills et al. (2014) introduced the shapelet transform, converting all candidate subsequences into distance-based features for shapelet discovery. Recent studies (Qu et al., 2024; Le et al., 2024; Wen et al., 2025) have applied the shapelet transform strategy in deep learning models for time series classification, thereby improving model interoperability. However, these methods often handle candidate subsequences in a hard way, which easily leads to the loss of helpful temporal patterns from the time series sample.

More recently, Nie et al. (2023); Wu et al. (2025) showed that using patches of time series as input data enhances forecasting performance and reduces training time. Building on this, Luo & Wang (2024), Wang et al. (2024), and Wen et al. (2024) validated the effectiveness of patch techniques for time series classification. Eldele et al. (2024) further demonstrated that using time series patches as CNN inputs outperformed Transformer-based methods. However, these methods typically use all patches as inputs, leading to computational inefficiency in TSC for long sequences.

2.3. Mixture-of-Experts in Time Series

Mixture of experts (MoE) typically consist of multiple subnetworks (experts) and a learnable router (Shazeer et al., 2017; Zhou et al., 2022). MoE has been utilized in time series forecasting for decades (Zeevi et al., 1996; Ni et al., 2024) and recently applied to improve the efficiency of time series forecasting foundation models (Liu et al., 2024; Shi et al., 2025). In addition, Huang et al. (2025) extended MoE to graph-based models for time series anomaly detection. Wen et al. (2025) introduced a gated router within MoE to integrate DNNs for learning time series representations, combining them with shapelet transform features to address the performance limitations of using shapelet transform alone in TSC. Unlike the above methods, we utilize the MoE router to activate class-specific experts for learning intra-shape local temporal patterns. Furthermore, we employ a shared expert to capture global temporal patterns across sparsified shapes from the same time series, thereby enhancing the discriminative power of shape embeddings.

3. Preliminaries

3.1. Problem Definition

In this study, we focus on the use of deep learning models for univariate time series classification. Let the time series dataset be represented as $\mathcal{D} = \{(\mathcal{X}_n, \mathcal{Y}_n)\}_{n=1}^{N}$, where N denotes the total number of time series samples. Each univariate time series $\mathcal{X}_n = \{x_1, x_2, \ldots, x_T\}$ is an ordered sequence of T real-valued observations. The corresponding label $\mathcal{Y}_n \in \mathbb{R}^C$ is a one-hot encoded vector, where each value $y_c \in \{0, 1\}$ indicates whether $\mathcal{X}_n$ belongs to class c, with C representing the total number of classes. Given a deep learning model parameterized by θ, the objective of the TSC task is to optimize the function $f_\theta : \mathbb{R}^T \to \mathbb{R}^C$ such that it can accurately predict the label $\mathcal{Y}_n$ corresponding to any input time series $\mathcal{X}_n$.

3.2. Time Series Shapelets

Given a time series $\mathcal{X}_n$, a subsequence of length m, where $m < T$, is denoted as $\mathcal{S}_{n,p}^m = \{x_p, x_{p+1}, \ldots, x_{p+m-1}\}$. Here, p denotes the starting index of $\mathcal{S}_{n,p}^m$, where $0 \le p < T - m$. All subsequences of length m from the time series $\mathcal{X}_n$ can be extracted using a sliding window of size m with a fixed step size q (commonly $q = 1$), traversing from $t = 1$ to T. Thus, the set of all subsequences of length m from $\mathcal{X}_n$ can be defined as $\mathcal{S}_n^m = \{\mathcal{S}_{n,p}^m \mid 0 \le p < T - m\}$.

Due to the length of the subsequences varying within the range $2 \le m \le T - 1$, the candidate set $\mathcal{A}$ of all possible subsequences for the time series dataset $\mathcal{D}$ is expressed as:

$$\mathcal{A} = \bigcup_{n=1}^{N} \bigcup_{m=2}^{T-1} \mathcal{S}_n^m = \{\mathcal{S}_{n,p}^m \mid 0 \le p < T - m\}. \quad (1)$$

Shapelets are defined as a subset of discriminative subsequences within $\mathcal{A}$ that maximally represent the class of each time series $\mathcal{X}_n$ (Ye & Keogh, 2009). Given that every shape in $\mathcal{A}$ could potentially serve as a shapelet, a brute-force search to evaluate all subsequences of each $\mathcal{X}_n$ using an evaluation function becomes computationally prohibitive as the number of samples N and the sequence length T increase. Thus, the design of appropriate sparsification techniques for the set $\mathcal{A}$ is crucial for enhancing the computational efficiency of shapelet-based TSC methods.

4. Methodology

4.1. Model Overview

The overall architecture of the SoftShape model is illustrated in Figure 2. SoftShape begins with a shape embedding layer that transforms the input time series into shape embeddings corresponding to multiple equal-length subsequences. These embeddings are normalized with a norm layer and then processed using soft shape sparsification (see Section 4.2). Specifically, an attention head assigns weight scores to each shape embedding based on its classification contribution. High-weight embeddings are scaled by their scores to form soft shapes, while low-weight embeddings are merged into a single shape for sparsification. The normalized sparsified shape embeddings are then input into the soft shape learning block to learn temporal patterns (see Section 4.3). Within this block, a MoE router activates a few class-specific experts to capture intra-shape temporal patterns (see Section 4.3.1). Meanwhile, these sparsified soft shapes are transformed into sequences, with a shared expert learning inter-shape temporal patterns (see Section 4.3.2). Finally, the sparsified soft shape embeddings, along with the intra-shape and inter-shape embeddings learned by the soft shape learning block, are combined in a residual way. After stacking L layers, the output is passed to a linear layer for classification training (see Section 4.4).

4.2. Soft Shape Sparsification

For each input time series $\mathcal{X}_n$, we use a 1D convolutional neural network (CNN) as the shape embedding layer to obtain $M = \frac{T-m}{q} + 1$ $(q < m)$ overlapping subsequence embeddings, denoted $\hat{\mathcal{S}}_n^m = \{\hat{\mathcal{S}}_{n,p}^m \mid 0 \le p < T-m\}$. Each shape embedding $\hat{\mathcal{S}}_{n,p}^m$ is computed using the following convolution operation:

$$\hat{\mathcal{S}}_{n,p}^m = \left\{ \sum_{i=0}^{\mathrm{m}-1} \mathbf{W}_i \mathcal{S}_{n,p}^m \mid p = 0, q, 2q, \ldots, T - m \right\}, \quad (2)$$

where $\mathbf{W}_i$ denotes the weights of the i-th filter within the kernel of the CNN, and $\hat{\mathcal{S}}_{n,p}^m \in \mathbb{R}^d$, with d representing the dimension of the shape embedding. Like patch-based methods (Nie et al., 2023; Eldele et al., 2024), we incorporate

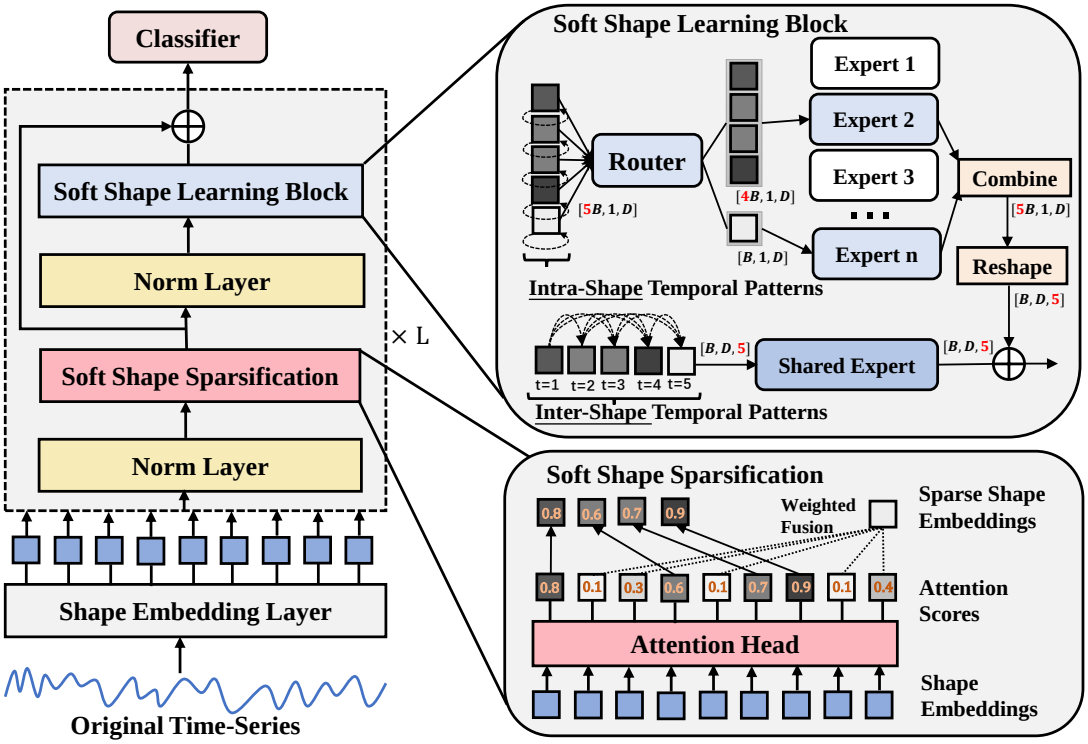


Figure 2: The general architecture of SoftShape model. SoftShape mainly involves: a) Soft Shape Sparsification converts shapes into soft forms and fuses those with lower scores into a single shape. b) Soft Shape Learning Block employs a MoE router to activate class-specific experts to learn intra-shape patterns and utilizes a shared expert to learn inter-shape patterns.

learnable positional embeddings into shape embeddings to capture temporal dependencies within the time series.

To identify the most discriminative shape embeddings in $\hat{\mathcal{S}}_n^m$, we employ a parameter-shared attention head across soft shape learning blocks at different depths. Unlike the self-attention mechanism in Transformers, we employ a gated attention mechanism (Ilse et al., 2018) that assumes independence among shape embeddings. This design facilitates label-guided identification of key shapes while maintaining linear computational complexity with respect to the number of shapes (Early et al., 2024). The attention head evaluates the contribution score of each $\hat{\mathcal{S}}_{n,p}^m$ for classification training, defined as follows:

$$\alpha(\hat{\mathcal{S}}_{n,p}^m) = \sigma\left(\mathbf{W}_2 \tanh\left(\mathbf{W}_1 \hat{\mathcal{S}}_{n,p}^m + \mathbf{b}_1\right) + \mathbf{b}_2\right), \quad (3)$$

where $\sigma(\cdot)$ is the sigmoid function outputting $\alpha \in (0, 1)$, $\tanh(\cdot)$ is the hyperbolic tangent function, $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{attn}} \times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{1 \times d_{\text{attn}}}$ are the weights of a linear layer, and $\mathbf{b}_1 \in \mathbb{R}^{d_{\text{attn}}}$ and $\mathbf{b}_2 \in \mathbb{R}$ are the bias terms.

The value of $\alpha(\hat{\mathcal{S}}_{n,p}^m)$ closer to 1 indicates a greater contribution to enhancing classification performance. Hence, we can perform soft shape sparsification using scores from the attention head based on their ranking. For each attention score of $\hat{\mathcal{S}}_{n,p}^m$ ranked within the top η proportion among the total $J = \frac{T-m}{q} + 1$ shape embeddings in $\hat{\mathcal{S}}_n^m$, we convert them into soft shape embeddings as follows:

$$\widetilde{\mathcal{S}}_{n,p}^m = \alpha(\hat{\mathcal{S}}_{n,p}^m)\hat{\mathcal{S}}_{n,p}^m. \quad (4)$$

For those with scores below the top η proportion, we fuse them into a single shape embedding as follows:

$$\widetilde{\mathcal{S}}_{\text{n, fused}}^m = \sum_{p \in \mathcal{E}} \alpha(\hat{\mathcal{S}}_{n,p}^m)\hat{\mathcal{S}}_{n,p}^m, \quad (5)$$

where $\mathcal{E}$ denotes the index of scores that are below the top η proportion. Finally, the soft shape embeddings are sorted according to their original order in $\hat{\mathcal{S}}_n^m$, with the fused embedding $\widetilde{\mathcal{S}}_{\text{n, fused}}^m$ appended at the end to form the sparsified soft shape embeddings $\widetilde{\mathcal{S}}_n^m$.

Through this sparsification process, we transform the time series $\mathcal{X}_n$ into sparsified soft shape embeddings $\widetilde{\mathcal{S}}_n^m$. Compared to directly using all shape embeddings in $\hat{\mathcal{S}}_n^m$ as input, the number of shape embeddings in $\widetilde{\mathcal{S}}_n^m$ is reduced, thereby decreasing the cost of the model's computation.

4.3. Soft Shape Learning Block

4.3.1. Intra-Shape Learning

Intuitively, shapelets from different classes show distinct temporal patterns, whereas those from the same class are more similar. Recently, Chen et al. (2022); Chowdhury et al. (2023) theoretically indicated that the MoE router can direct input embeddings with similar class patterns to the same expert while filtering out class-irrelevant features. This allows the router to selectively activate a subset of experts (sub-networks) for each shape embedding, enabling class-specific feature learning while reducing computational overhead compared to activating all experts simultaneously. Therefore, we introduce a module that employs the MoE router to activate class-specific experts for learning intra-shape temporal patterns, thereby enhancing the discriminative capability of each soft shape embedding. The function of the MoE router is defined as:

$$G(\widetilde{\mathcal{S}}^m_{n,p}) = \text{TOP}_k(\text{softmax}(\mathbf{W}_t\widetilde{\mathcal{S}}^m_{n,p})), \tag{6}$$

$$\text{TOP}_k(v) = \begin{cases} v_i, & \text{if } v_i \text{ is in the top } k \text{ values of } v, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where $\mathbf{W}_t \in \mathbb{R}^{\hat{C}\times d}$ are the learning weights of the router that convert the sparsified shape embedding $\widetilde{\mathcal{S}}^m_{n,p}$ into a vector of corresponding to the total number of experts $\hat{C}$. The $\text{TOP}_k(v)$ is applied after the softmax function to prevent the values of $G(\widetilde{\mathcal{S}}^m_{n,p})$ being zero almost everywhere during training (Riquelme et al., 2021), particularly when $k = 1$.

In the paper, each MoE expert employs a lightweight MLP network to learn class-specific patterns using each soft shape embedding as input. The e-th expert function is defined as:

$$h_e(\theta, \widetilde{\mathcal{S}}^m_{n,p}) = \hat{G}_e(\widetilde{\mathcal{S}}^m_{n,p})\text{GeLU}(\mathbf{W}_e\widetilde{\mathcal{S}}^m_{n,p} + \mathbf{b}_e), \tag{8}$$

$$\hat{G}_e(\widetilde{\mathcal{S}}^m_{n,p}) = \frac{e^{G_e(\widetilde{\mathcal{S}}^m_{n,p})}}{\sum_{i\in\text{TOP}_k(v)} e^{G_i(\widetilde{\mathcal{S}}^m_{n,p})}}, \tag{9}$$

where $\mathbf{W}_e$ and $\mathbf{b}_e$ represent the weight and bias terms of the MLP network, and $\text{GeLU}(\cdot)$ denotes the gaussian error linear unit activation function. The MoE experts' parameters are shared across soft shape learning blocks at different depths, ensuring the efficiency of parameter optimization.

However, the MoE router often tends to converge to a state where it assigns higher weights to a few favored experts. This imbalance can cause the favored experts to train faster and dominate the expert selection process, resulting in underutilization and potential underfitting of less frequently chosen experts. To address this issue, following (Shazeer et al., 2017), we employ two loss functions as follows:

$$\mathcal{L}_{\text{imp}}(\widetilde{\mathcal{S}}^m_{n,p}) = \mathbf{W}_{\text{imp}}\text{CV}\left(\sum_{\widetilde{\mathcal{S}}^m_{n,p}\in\widetilde{\mathcal{S}}^m_n} G(\widetilde{\mathcal{S}}^m_{n,p})\right)^2, \tag{10}$$

where $\mathbf{W}_{\text{imp}}$ is the learning weight parameters and $\text{CV}(\cdot)$ denotes the coefficient of variation, thus ensuring all experts contribute equally.

$$\mathcal{L}_{\text{load}}(\widetilde{\mathcal{S}}^m_{n,p}) = \mathbf{W}_{\text{load}}\text{CV}(\text{Load}(\widetilde{\mathcal{S}}^m_{n,p}))^2, \tag{11}$$

where $\mathbf{W}_{\text{load}}$ is the learning weight parameters for load balancing and $\text{Load}(\widetilde{\mathcal{S}}^m_{n,p})$ is the number of soft shape embeddings assigned to each expert, thereby reducing the risk of underutilization. By jointly optimizing $\mathcal{L}_{\text{imp}}$ and $\mathcal{L}_{\text{load}}$ in the overall loss function, we can alleviate the likelihood of underfitting in some experts. In addition, following Zoph et al. (2022), we apply a residual connection between the input shape embedding $\widetilde{\mathcal{S}}^m_{n,p}$ and the output $h_e(\theta, \widetilde{\mathcal{S}}^m_{n,p})$ to enhance training stability.

4.3.2. Inter-Shape Learning

While the MoE router activates a few experts to learn class-specific temporal patterns within each soft shape, it treats the multiple shapes extracted from each time series sample independently, thereby overlooking the temporal dependencies among shapes. In general, learning intra-shape temporal patterns is beneficial for capturing local discriminative features, but it can be difficult to learn the global temporal patterns from the time series that are vital for classification. To this end, we introduce a shared expert to learn the temporal patterns among sparsified shapes within the time series.

The sparsified soft shape embeddings are first transformed into a sequence where each shape is treated as an individual unit. This transformation is defined as:

$$\mathcal{Q}^m_n = (\widetilde{\mathcal{S}}^m_n)^T, \quad \text{where} \quad \widetilde{\mathcal{S}}^m_n \in \mathbb{R}^{B\times Num\times d}, \tag{12}$$

where $\mathcal{Q}^m_n \in \mathbb{R}^{B\times d\times Num}$, and B denotes the batch size, $Num = J \times \eta + 1$ is the number of sparsified soft shapes of each time series, and d is the dimension of one soft shape embedding. Recent studies (Middlehurst et al., 2024) have shown that CNN-based models perform remarkably in TSC tasks. In this work, we employ a CNN-based Inception module (Ismail Fawaz et al., 2020) as the shared expert. The shared expert comprises three 1D convolutional layers with different kernel sizes, sliding along the third dimension of $\mathcal{Q}^m_n$ from 0 to Num. This allows it to learn multi-resolution temporal patterns across soft shapes effectively. For raw input time series samples, the input dimension containing B univariate time series can be represented as $(B, 1, T)$, where T is the length of each time series. Traditional CNN-based methods (Ismail Fawaz et al., 2020) typically transform each point in the time series into d-dimensional embeddings, resulting in an input dimension of (B, d, T). Unlike the above, we treat each shape as a sequence point and convert it into d-dimensional embeddings for training.

As shown in Equations (1) and (5), with a higher sparsity rate $(1 - \eta)$, the actual sequence length Num of $\mathcal{Q}^m_n$ fed

into the shared expert is significantly smaller than T, thus reducing the model's computational load. The shared expert ensures efficient learning of temporal patterns across sparsified soft shapes, better capturing global features of the time series for classification training.

4.4. The Training of SoftShape

The SoftShape model stacks L layers of the soft shape learning block for classification training. Each layer's output is processed using a GeLU$(\cdot)$ activation function. The final layer outputs three types of embeddings: the sparsified soft shape embeddings, the intra-shape embeddings, which are combined through a residual way denoted as O_n^m. To accurately represent the learned attention scores for each shape embedding associated with the target class, we employ conjunctive pooling (Early et al., 2024) for training:

$$\hat{\mathcal{Y}}_n = \frac{1}{Num} \sum_{i=0}^{Num} \alpha(O_n^m)\phi(O_{n,i}^m), \tag{13}$$

where ϕ denotes a linear layer as the classifier. For all time series samples, we use the cross-entropy loss for classification training, defined as:

$$\mathcal{L}_{\text{ce}}(\mathcal{X}, \mathcal{Y}) = -\frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{C} 1\{y_i = j\} \log(\hat{\mathcal{Y}}_i^{\,j}), \tag{14}$$

where $\hat{\mathcal{Y}}_i^{\,j}$ is the predicted class probability at the j-th class for the sample $\mathcal{X}_i$. Therefore, the overall training objective for SoftShape is given by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \lambda(\mathcal{L}_{\text{imp}} + \mathcal{L}_{\text{load}}), \tag{15}$$

where λ is a hyperparameter to adjust the training loss ratio. Additionally, the pseudo-code for SoftShape can be found in Algorithm 1 located in the Appendix.

5. Experiments

To evaluate the performance of various methods for TSC, we conduct experiments using the UCR time series archive (Dau et al., 2019), a widely recognized benchmark in TSC (Ismail Fawaz et al., 2019). Many datasets in the UCR archive contain a significantly higher number of test samples compared to the training samples. Also, the original UCR time series datasets lack a specific validation set, increasing the risk of overfitting in deep learning methods. Following (Dau et al., 2019; Ma et al., 2024), we merge the original training and test sets of each UCR dataset. These merged datasets are then divided into train-validation-test sets at a ratio of 60%-20%-20%. This paper compares SoftShape against 19 baseline methods, categorized as follows:

- Deep Learning Methods (DL):
  - *CNN-based (DL-CNN):* FCN (Wang et al., 2017), T-Loss (Franceschi et al., 2019), SelfTime (Fan et al., 2020), TS-TCC (Eldele et al., 2021), TS2Vec (Yue et al., 2022), TimesNet (Wu et al., 2023), InceptionTime (Ismail Fawaz et al., 2020), LightTS (Campos et al., 2023), ShapeConv (Qu et al., 2024), ModernTCN (Luo & Wang, 2024), and TSLANet (Eldele et al., 2024).
  - *Transformer-based (DL-Trans):* TST (Zerveas et al., 2021), PatchTST (Nie et al., 2023), Shapeformer (Le et al., 2024), and Medformer (Wang et al., 2024).
  - *Foundation models (DL-FM):* GPT4TS (Zhou et al., 2023) and UniTS (Gao et al., 2024).
- Non-Deep Learning Methods (Non-DL): RDST (Guillaume et al., 2022) and MultiRocket-Hydra (MR-H) (Dempster et al., 2023).

For information on datasets, baselines, and implementation details, please refer to Appendix A.

5.1. Main Results

Table 1: Test classification accuracy comparisons on 128 UCR time series datasets. The best results are in **bold**, and the second best results are underlined.

	Methods	Avg. Acc	Avg. Rank	Win	P-value
DL-CNN	FCN	0.8296	9.53	13	1.43E-12
	T-Loss	0.8325	11.12	9	2.95E-14
	SelfTime	0.8017	13.80	0	4.53E-25
	TS-TCC	0.7807	13.96	0	1.60E-15
	TS2Vec	0.8691	8.43	9	1.69E-15
	TimesNet	0.8367	10.13	7	4.22E-15
	InceptionTime	0.9181	4.05	29	7.39E-06
	ShapeConv	0.7688	13.91	5	3.58E-24
	ModernTCN	0.7938	11.37	9	1.78E-18
	TSLANet	0.9205	3.68	31	1.06E-03
DL Trans	TST	0.7755	13.54	1	2.00E-19
	PatchTST	0.8265	9.56	12	1.27E-15
	Medformer	0.8541	9.26	7	7.15E-16
DL FM	GPT4TS	0.8593	9.34	6	7.89E-16
	UniTS	0.8502	9.66	5	1.41E-13
Non DL	RDST	0.8897	6.41	23	7.54E-10
	MR-H	0.8972	5.51	29	3.80E-07
SoftShape (Ours)		**0.9334**	**2.72**	**53**	-

The classification accuracies on the test sets of the 128 UCR datasets are presented in Table 1, with detailed results provided in Appendix B.1. The average ranking (*Avg. Rank*) reflects the method's relative position based on test accuracy, where lower values signify higher accuracy. *Win* indicates the number of datasets where the baseline achieves the highest test accuracy. The *P-value* from the Wilcoxon signed-rank test (Demšar, 2006) assesses the significance of

differences in test accuracy between pairwise methods. A *P-value* < 0.05 suggests SoftShape outperforms the baseline.

Due to the high computational cost of LightTS (Campos et al., 2023) and Shapeformer (Le et al., 2024), their results are reported only on 18 selected UCR time series datasets in Table 10 of Appendix B.1. Detailed reasons for 18 UCR dataset selection are in Appendix B.3. The experimental results in Table 1 highlight SoftShape's superior classification performance, demonstrating its effectiveness in TSC. Also, the P-value results confirm the statistical significance of SoftShape's performance compared to baselines, highlighting its capacity to capture the discriminative patterns of time series data. The critical diagram (Figure 6) in the Appendix further supports the statistical significance of SoftShape's performance improvement.

5.2. Ablation Study

Table 2: Test accuracy of ablation study on 128 UCR datasets. The best results are in **bold**, and the second best results are underlined.

Methods	Avg. Acc	Avg. Rank	Win	P-value
w/o Soft Sparse	0.9123	3.04	29	4.69E-06
w/o Intra	0.9245	2.75	31	5.39E-04
w/o Inter	0.9022	3.74	19	1.81E-09
w/o Intra & Inter	0.8696	5.02	11	4.35E-16
with Linear Shape	0.9164	3.23	22	1.80E-09
SoftShape (Ours)	**0.9334**	**2.04**	**60**	-

The ablation studies include: 1) **w/o Soft Sparse** removes the soft sparse process, using only the selected top η shape embeddings for training; 2) **w/o Intra** eliminates the intra-shape learning process via the class-specific experts; 3) **w/o Inter** removes the inter-shape learning process via the shared expert; 4) **w/o Intra & Inter** excludes both the intra-shape learning and inter-shape learning processes; 5) **with Linear Shape** employs a linear layer commonly used in (Nie et al., 2023; Wang et al., 2024) to replace the 1D CNN in Equation (2) as the shape embedding layer.

Table 2 presents the statistical ablation results, with detailed results provided in Appendix B.2. **w/o Soft Sparse** reduces accuracy and lowers the *Avg. Rank*, indicating the importance of this process in capturing the most helpful shape embeddings for classification. Similarly, **w/o Intra** highlights the significant role of the intra-shape module in SoftShape. Moreover, **w/o Inter** leads to a large drop in performance, underscoring the critical contribution of inter-shape relationships in learning global patterns of time series. When both the intra- and inter-shape learning processes are excluded (**w/o Intra & Inter**), performance further degrades, demonstrating that these two processes are foundational to SoftShape's effectiveness. Finally, **with Linear Shape** confirms the superiority of the CNN architecture for learning shape embeddings. These results confirm the importance of each component in the SoftShape model.

5.3. Shape Sparsification and Learning Analysis

Table 3: Test accuracy of different sparse ratios on 18 UCR datasets. The best results are in **bold**, and the second best results are underlined.

Sparse Ratio	Avg. Acc	Avg. Rank	Win	P-value
0%	0.9461	2.44	4	-
10%	**0.9469**	**2.39**	**7**	2.97E-01
30%	0.9448	2.78	5	2.66E-01
50%	0.9453	2.61	6	3.46E-01
70%	0.9323	3.89	5	9.37E-03
90%	0.9261	4.50	2	4.02E-04

Evaluating all 128 UCR time series datasets takes considerable time, so we selected 18 datasets from the UCR time series archive for analysis and subsequent experiments. Table 3 presents the test results when combining shape embeddings with weighted fusion at varying sparse ratio $(1-\eta)$ proportions during sparsification, with detailed results provided in Appendix B.3. When $(1-\eta) \leq 50\%$, test accuracy showed no significant decline compared to the non-sparsified case (0%), with all p-values > 0.05. Conversely, when $(1-\eta) > 50\%$, particularly at 90%, performance declined significantly compared to 0%, indicating that excessive sparsification can remove some shape embeddings beneficial for capturing inter-shape global temporal patterns. Furthermore, the performance degradation under high sparsity ratios suggests that hard shapelet-based methods discard numerous informative subsequences, which may result in the loss of critical patterns useful for classification. As performance at 50% is statistically comparable to 0%, we set η to 50% in the main experiments to optimize the computational efficiency of the soft shape learning block.

Table 4 illustrates the effect of the number of activated experts (k) selected by the MoE router during intra-shape learning, with detailed results provided in Appendix B.3. The results show that when $k = 3$ or $k = 4$, the average rank is worse than when $k = 1$ or $k = 2$. This suggests that activating a greater number of class-specific experts for each soft shape increases computational cost while diminishing the class discriminability of the intra-shape embeddings.

Also, we select one non-deep learning method (MR-H) and three deep learning baselines for runtime analysis. MR-H uses randomly initialized convolutional kernels for feature extraction, with training times measured on a CPU. The deep learning methods are evaluated on a GPU. InceptionTime is an advanced deep learning method described in

Table 4: Test accuracy comparison of the number of activated experts on 18 UCR datasets. The best result is in **bold**.

Activated Experts	k=1	k=2	k=3	k=4
Avg. Rank	**1.89**	1.94	2.78	2.39

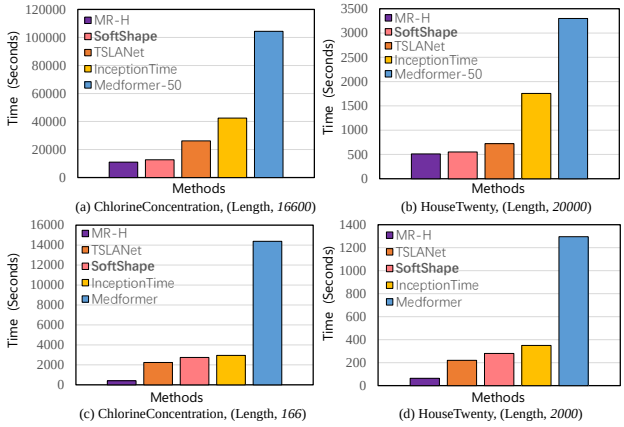


Figure 3: Running time analysis on the *ChlorineConcentration* and *HouseTwenty* datasets.

(Middlehurst et al., 2024). TSLANet and Medformer are recent patch-based models utilizing CNNs and Transformers, respectively. To evaluate SoftShape's sparsification efficiency, we select two datasets: *ChlorineConcentration* (4,307 samples, length 166) and *HouseTwenty* (159 samples, length 2,000). In Figures 3(a) and 3(b), sequence lengths are extended to 16,600 and 20,000 by sample concatenation to simulate long-sequence scenarios. Given Medformer's high computational cost for long sequences, we report only its runtime at 50 epochs (out of a maximum of 500), denoted as Medformer-50. As shown in Figures 3(a) and 3(b), SoftShape outperforms all deep learning baselines and is slightly slower than MR-H. Specifically, Medformer's self-attention leads to higher training time with long sequences, while SoftShape's linear-complexity gated attention ensures greater efficiency. In short-sequence [Figure 3(c)] and small-sample [Figure 3(d)] scenarios, SoftShape exhibits slightly higher runtime than TSLANet and substantially higher than MR-H. In these settings, sparsification offers limited efficiency gains, as the number of sparsified shapes does not decrease significantly compared to a greater number of training samples with long sequences. Additionally, Table 14 in Appendix B.3 presents the inference times and parameter counts for the settings in Figures 3(c) and 3(d).

5.4. Visualization Analysis

To evaluate SoftShape's effectiveness and interpretability, we use Multiple Instance Learning (MIL) (Early et al., 2024) for visualization. MIL identifies key time points in time series that improve classification performance. This study applies MIL to highlight significant shapes. Figure 4 shows the discriminative regions and attention scores learned by SoftShape on *Class 1* of the *Trace* dataset. Deeper red indicates beneficial regions for classification, while deeper blue shows less relevant ones. Figures 4(a) and 4(b) treat each shape (length $m = 64$) as one sequence point, while Figures 4(c) and 4(d) map these results to the entire time series. SoftShape assigns higher attention scores to subsequences with significant differences between *Class 1* and *Class 2*, while sparsifying less discriminative regions. These findings indicate that SoftShape learns highly discriminative shapes as shapelets, enhancing interpretability through attention scores. Further MIL visualisations on the *Lightning2* dataset in the Appendix of Figure 7 support this finding.

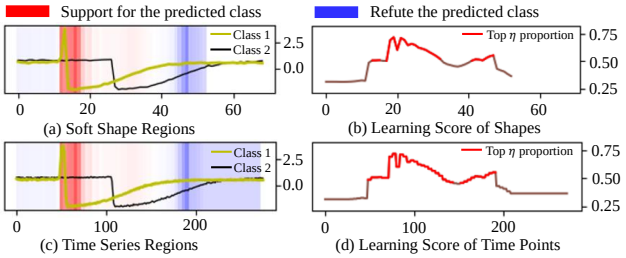
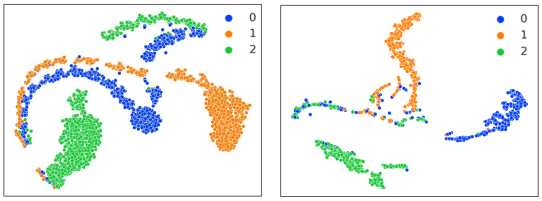


Figure 4: The MIL visualization on the *Trace* dataset.

The t-distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten & Hinton, 2008) is used to visualize shape embeddings learned by SoftShape. Figure 5(a) shows input shape embeddings from 1D CNN (Eq. (2)) on the *CBF* dataset. Figures 5(b) and 5(c) present sparsified intra-shape and inter-shape embeddings from the soft shape learning block, while Figure 5(d) displays final output sparsified shape embeddings for classification. Compared to Figure 5(a), the intra-shape embeddings [Figure 5(b)] learned by activating class-specific experts form clusters for samples of the same class but still mix some samples from different classes. The inter-shape embeddings [Figure 5(c)] learned by a shared expert effectively separate different classes while dispersing same-class samples into multiple clusters. Combining intra- and inter-shape embeddings, Figure 5(d) balances class distinction and clusters samples of the same class closely together. These results indicate that SoftShape leverages intra- and inter-shape patterns to improve shape embedding discriminability. Further t-SNE visualization on the *TwoPatterns*, *Fiftywords*, and *ECG200* datasets with different classification difficulty can be seen in the Appendix of Figures 8, 9, and 10, respectively.

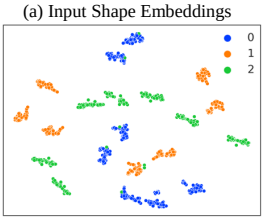
5.5. Hyperparameter Analysis

Table 5 presents the classification results of SoftShape for various sliding fixed step sizes q. We observed worse *Avg.*

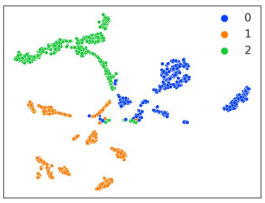


(a) Input Shape Embeddings

(b) Intra-Shape Embeddings



(c) Inter-Shape Embeddings



(d) Output Shape Embeddings

Figure 5: The t-SNE visualization on the *CBF* dataset.

Table 5: Test accuracy comparison of different sliding window sizes on 18 UCR datasets. The best result is in **bold**.

Sliding Step Size	q=1	q=2	q=3	q=4	q=m
Avg. Rank	3.83	2.78	2.44	**1.78**	2.94

Table 6: Test accuracy comparison of model depth on 18 UCR datasets. The best result is in **bold**.

Depth	L=1	L=2	L=3	L=4	L=5	L=6
Avg. Rank	3.72	**2.61**	3.06	3.33	3.44	3.06

Rank values when $q = 1$ or $q = m$ (non-overlapping). With $q = 1$, the maximum subsequences result in redundancy and irrelevant information. Conversely, $q = m$ generates the fewest subsequences, possibly missing important discriminative subsequences. Table 6 examines the influence of model depth L, revealing that SoftShape achieves the lowest *Avg. Rank* at $L = 2$, demonstrating shape embeddings capture discriminative features effectively, even at low depths. Detailed results of Tables 5 and 6 provided in Appendix B.4. Also, Tables 18, 19, 20, and 21 in Appendix B.4 present statistical test results on the impact of the maximum number of class-specific experts, class-specific expert networks, shared expert networks, and the hyperparameter λ.

To evaluate the impact of the hyperparameter shape length m on classification performance, we conducted three experiments: (1) Val-Select: Choose a fixed m for one SoftShape model using the validation set. (2) Fixed-8: Set m = 8 for one SoftShape model. (3) Multi-Seq: Use fixed lengths (8, 16, 32) in parallel three SoftShape models, with cross-scale residual fusion (Liu et al., 2023) for classification. As shown in Table 7 (detailed results in Appendix B.4, there was no significant difference in performance between Val-Select and Multi-Seq (p-value < 0.05). The Val-Select model also has fewer parameters and a lower runtime. In addition, Tables 22 and 23 in Appendix B.5 present analyses of SoftShape's time series forecasting performance.

Table 7: Test accuracy comparison of shape length m on 18 UCR datasets. The best result is in **bold**.

Methods	Val-Select	Fixed-8	Multi-Seq
Avg. Rank	1.72	2.28	**1.44**
P-value	2.88E-01	4.68E-02	-

6. Conclusion

In this paper, we propose a soft sparse shapes model for efficient time series classification. Specifically, we introduce a soft shape sparsification mechanism that merges low-discriminative subsequences into a single shape based on learned attention scores, reducing the number of shapes for shapelet learning. Moreover, we propose a soft shape learning block for learning intra-shape and inter-shape temporal patterns, enhancing shape embedding discriminability. Extensive experiments demonstrate that SoftShape achieves state-of-the-art classification performance while providing interpretable results. However, this study does not consider the modelling of relationships among multiple variables of the time series. In the future, we will investigate deep learning methods for multivariate TSC.

Acknowledgements

We thank the anonymous reviewers for their helpful feedbacks. We thank Professor Eamonn Keogh and all the people who have contributed to the UCR time series classification archive. Zhen Liu and Yicheng Luo equally contributed to this work. The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant No. 62272173), the Natural Science Foundation of Guangdong Province (Grant Nos. 2024A1515010089, 2022A1515010179), the Science and Technology Planning Project of Guangdong Province (Grant No. 2023A0505050106), the National Key R&D Program of China (Grant No. 2023YFA1011601), and the China Scholarship Council program (Grant No. 202406150081).

Impact Statement

Our proposed work `SoftShape` aims to advance the field of Machine Learning by providing a more efficient and interpretable model for time series classification across various applications. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bian, Y., Ju, X., Li, J., Xu, Z., Cheng, D., and Xu, Q. Multi-patch prediction: Adapting language models for time series representation learning. In *Forty-first International Conference on Machine Learning*, 2024.

Campos, D., Zhang, M., Yang, B., Kieu, T., Guo, C., and Jensen, C. S. Lightts: Lightweight time series classification with adaptive ensemble distillation. *Proceedings of the ACM on Management of Data*, 1(2):1–27, 2023.

Chen, Z., Deng, Y., Wu, Y., Gu, Q., and Li, Y. Towards understanding the mixture-of-experts layer in deep learning. *Advances in neural information processing systems*, 35: 23049–23062, 2022.

Chowdhury, M. N. R., Zhang, S., Wang, M., Liu, S., and Chen, P.-Y. Patch-level routing in mixture-of-experts is provably sample-efficient for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6074–6114. PMLR, 2023.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., and Keogh, E. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Dempster, A., Schmidt, D. F., and Webb, G. I. Hydra: Competing convolutional kernels for fast and accurate time series classification. *Data Mining and Knowledge Discovery*, 37(5):1779–1805, 2023.

Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.

Early, J., Cheung, G., Cutajar, K., Xie, H., Kandola, J., and Twomey, N. Inherently interpretable time series classification via multiple instance learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., Li, X., and Guan, C. Time-series representation learning via temporal and contextual contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

Eldele, E., Ragab, M., Chen, Z., Wu, M., and Li, X. Tslanet: Rethinking transformers for time series representation learning. In *Forty-first International Conference on Machine Learning*, 2024.

Fan, H., Zhang, F., and Gao, Y. Self-supervised time series representation learning by inter-intra relational reasoning. *arXiv preprint arXiv:2011.13548*, 2020.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

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

Gao, S., Koker, T., Queen, O., Hartvigsen, T., Tsiligkaridis, T., and Zitnik, M. Units: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37:140589–140631, 2024.

Grabocka, J., Schilling, N., Wistuba, M., and Schmidt-Thieme, L. Learning time-series shapelets. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 392–401, 2014.

Guillaume, A., Vrain, C., and Elloumi, W. Random dilated shapelet transform: A new approach for time series shapelets. In *International Conference on Pattern Recognition and Artificial Intelligence*, pp. 653–664. Springer, 2022.

Hills, J., Lines, J., Baranauskas, E., Mapp, J., and Bagnall, A. Classification of time series by shapelet transformation. *Data mining and knowledge discovery*, 28:851–881, 2014.

Hou, L., Kwok, J., and Zurada, J. Efficient learning of time-series shapelets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Huang, X., Chen, W., Hu, B., and Mao, Z. Graph mixture of experts and memory-augmented routers for multivariate time series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17476–17484, 2025.

Ilse, M., Tomczak, J., and Welling, M. Attention-based deep multiple instance learning. In *International conference on machine learning*, pp. 2127–2136. PMLR, 2018.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33 (4):917–963, 2019.

Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

Jin, M., Koh, H. Y., Wen, Q., Zambon, D., Alippi, C., Webb, G. I., King, I., and Pan, S. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lara, O. D. and Labrador, M. A. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209, 2012.

Le, X.-M., Luo, L., Aickelin, U., and Tran, M.-T. Shapeformer: Shapelet transformer for multivariate time series classification. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1484–1494, 2024.

Li, G., Choi, B., Xu, J., Bhowmick, S. S., Chun, K.-P., and Wong, G. L.-H. Efficient shapelet discovery for time series classification. *IEEE transactions on knowledge and data engineering*, 34(3):1149–1163, 2020.

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

Liu, X., Liu, J., Woo, G., Aksu, T., Liang, Y., Zimmermann, R., Liu, C., Savarese, S., Xiong, C., and Sahoo, D. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. *arXiv preprint arXiv:2410.10469*, 2024.

Liu, Z., Chen, D., Pei, W., Ma, Q., et al. Scale-teaching: Robust multi-scale training for time series classification with noisy labels. *Advances in Neural Information Processing Systems*, 36:33726–33757, 2023.

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

Ma, Q., Zhuang, W., Li, S., Huang, D., and Cottrell, G. Adversarial dynamic shapelet networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5069–5076, 2020.

Ma, Q., Liu, Z., Zheng, Z., Huang, Z., Zhu, S., Yu, Z., and Kwok, J. T. A survey on time-series pre-trained models. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):7536–7555, 2024.

Middlehurst, M., Schäfer, P., and Bagnall, A. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, pp. 1–74, 2024.

Mohammadi Foumani, N., Miller, L., Tan, C. W., Webb, G. I., Forestier, G., and Salehi, M. Deep learning for time series classification and extrinsic regression: A current survey. *ACM Computing Surveys*, 56(9):1–45, 2024.

Mueen, A., Keogh, E., and Young, N. Logical-shapelets: an expressive primitive for time series classification. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1154–1162, 2011.

Ni, R., Lin, Z., Wang, S., and Fanti, G. Mixture-of-linear-experts for long-term time series forecasting. In *International Conference on Artificial Intelligence and Statistics*, pp. 4672–4680. PMLR, 2024.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Qu, E., Wang, Y., Luo, X., He, W., Ren, K., and Li, D. Cnn kernels can be the best shapelets. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34: 8583–8595, 2021.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Shi, X., Wang, S., Nie, Y., Li, D., Ye, Z., Wen, Q., and Jin, M. Time-moe: Billion-scale time series foundation models with mixture of experts. In *The Twelfth International Conference on Learning Representations*, 2025.

Vail, D. and Veloso, M. Learning from accelerometer data on a legged robot. *IFAC Proceedings Volumes*, 37(8): 822–827, 2004.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Wang, Y., Huang, N., Li, T., Yan, Y., and Zhang, X. Medformer: A multi-granularity patching transformer for medical time-series classification. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

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

Wen, Y., Ma, T., Weng, T.-W., Nguyen, L. M., and Julius, A. A. Abstracted shapes as tokens-a generalizable and interpretable model for time-series classification. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Wen, Y., Ma, T., Luss, R., Bhattacharjya, D., Fokoue, A., and Julius, A. A. Shedding light on time series classification using interpretability gated networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.

Wu, Y., Meng, X., Hu, H., Zhang, J., Dong, Y., and Lu, D. Affirm: Interactive mamba with adaptive fourier filters for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21599–21607, 2025.

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

Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., and Xu, B. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8980–8987, 2022.

Zeevi, A., Meir, R., and Adler, R. Time series prediction using mixtures of experts. *Advances in neural information processing systems*, 9, 1996.

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

Zhang, K., Wen, Q., Zhang, C., Cai, R., Jin, M., Liu, Y., Zhang, J. Y., Liang, Y., Pang, G., Song, D., et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Zhou, T., Niu, P., Sun, L., Jin, R., et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V., Dai, A. M., Le, Q. V., Laudon, J., et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

Zoph, B., Bello, I., Kumar, S., Du, N., Huang, Y., Dean, J., Shazeer, N., and Fedus, W. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.