A Unified Shape-Aware Foundation Model for Time Series Classification

Zhen Liu[1,2], Yucheng Wang[2], Boyuan Li[1], Junhao Zheng[1], Emadeldeen Eldele[3], Min Wu[2]*, Qianli Ma[1]*

[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
[3]Department of Computer Science, Khalifa University, UAE
cszhenliu@foxmail.com, wumin@a-star.edu.sg, qianlima@scut.edu.cn

Abstract

Foundation models pre-trained on large-scale source datasets are reshaping the traditional training paradigm for time series classification. However, existing time series foundation models primarily focus on forecasting tasks and often overlook classification-specific challenges, such as modeling interpretable shapelets that capture class-discriminative temporal features. To bridge this gap, we propose UniShape, a unified shape-aware foundation model designed for time series classification. UniShape incorporates a shape-aware adapter that adaptively aggregates multiscale discriminative subsequences (shapes) into class tokens, effectively selecting the most relevant subsequence scales to enhance model interpretability. Meanwhile, a prototype-based pretraining module is introduced to jointly learn instance- and shape-level representations, enabling the capture of transferable shape patterns. Pre-trained on a large-scale multi-domain time series dataset comprising 1.89 million samples, UniShape exhibits superior generalization across diverse target domains. Experiments on 128 UCR datasets and 30 additional time series datasets demonstrate that UniShape achieves state-of-the-art classification performance, with interpretability and ablation analyses further validating its effectiveness.

Code — https://github.com/qianlima-lab/UniShape

Introduction

Deep learning models have achieved notable success in time series classification (TSC) across various domains (Ismail Fawaz et al. 2019; Luo et al. 2024). However, most existing methods (Liu et al. 2023a; Mohammadi Foumani et al. 2024) are trained on small-scale datasets, thereby limiting their generalization capability in cross-domain settings. In contrast, foundation models (FMs) have exhibited strong transferability in vision and language tasks (Zhang et al. 2024), prompting a growing interest in their application to time series data. Yet, existing efforts mainly focused on time series forecasting tasks (Ansari et al. 2024; Li et al. 2025), while FM development specifically for classification tasks is still in its early stages. Designing a unified FM for TSC thus presents an open and impactful research problem.

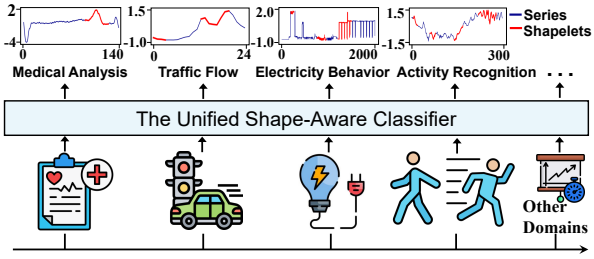

Figure 1: Illustration of the unified shape-aware classifier trained across multiple domains. Red lines represent shapelets extracted from the *ECG5000*, *Chinatown*, *HouseTwenty*, and *CricketX* datasets, corresponding to four distinct domains in the UCR time series archive (Dau et al. 2019).

The fundamental differences between time series forecasting and classification present key challenges for developing effective FMs for classification. Forecasting focuses on learning temporal dynamics such as trends and seasonality from historical data (Woo et al. 2022), aiming to predict future values based on sufficiently long contextual input sequences. Conversely, classification emphasizes identifying discriminative local patterns within fixed-length samples from different subjects (Ismail Fawaz et al. 2020; Liu et al. 2023b), thus assigning discrete class labels to unseen samples. While forecasting outputs continuous, multi-step numerical sequences, classification involves extracting informative and interpretable patterns from individual samples. As a result, forecasting-oriented FMs often fail to capture class-discriminative features essential for accurate TSC.

Although a few recent studies (Goswami et al. 2024; Feofanov et al. 2025) have made initial attempts to apply FMs for TSC, they largely neglect the interpretability that is vital to domains such as healthcare. Shapelets are discriminative subsequences widely used to enhance the interpretability of TSC results, as they can represent key class patterns (e.g., the red segment in Figure 1) (Ye and Keogh 2011). However, existing shapelet-based methods typically rely on label supervision with domain-specific assumptions (Li et al. 2021; Liu et al. 2025), limiting their applicability in FM pretraining scenarios where annotations are scarce. Also, shapelets inherently exhibit multiscale properties (Grabocka et al. 2014; Yamaguchi, Ueno, and Kashima 2023), as discrimi-

*Corresponding authors.

native subsequence patterns may appear at varying lengths and temporal locations. Therefore, effectively modeling and integrating multiscale shapelet representations within a unified FM during pretraining remains a challenge.

To this end, this paper proposes a **Uni**fied **Shape**-aware foundation model named **UniShape** for TSC that aims to enhance both downstream classification performance and interpretability. UniShape comprises two key components. First, the shape-aware adapter processes multiscale subsequences (or shapes) as input tokens and adaptively aggregates discriminative features of varying lengths via an attention-based pooling mechanism to generate instance-level class tokens. Building on these class and shape tokens, the prototype-based pretraining module then jointly optimizes instance-level and shape-level representations for class prototype learning, enabling the model to capture generalizable shapelet patterns. Through this design, UniShape achieves effective transferability and improved interpretability when fine-tuned on diverse TSC target domains following pretraining on a large-scale multi-domain dataset.

The main contributions are summarized as follows:

- We propose UniShape, a unified foundation model for TSC that effectively captures multiscale shapelet features and automatically selects optimal subsequence scales via a shape-aware adapter.
- We introduce a prototype-based pretraining module that jointly learns discriminative representations at both instance-level and shape-level tokens, thus enhancing the model's generalization capability in target domains.
- Extensive experiments on 158 univariate time series datasets demonstrate that UniShape significantly outperforms state-of-the-art methods in classification performance, while also exhibiting good interpretability.

Related Work

Time Series Classification. Early TSC approaches primarily relied on dynamic time warping and nearest neighbor classifiers (Keogh and Kasetty 2002). Recent efforts have shifted towards non-deep learning methods (Guillaume, Vrain, and Elloumi 2022; Middlehurst, Schäfer, and Bagnall 2024), such as the Rocket family of algorithms (Dempster, Petitjean, and Webb 2020; Dempster, Schmidt, and Webb 2021, 2023), alongside deep learning models based on neural networks (Mohammadi Foumani et al. 2024). In particular, both self-supervised representation learning (Yue et al. 2022) and end-to-end supervised classification frameworks (Ismail Fawaz et al. 2020; Eldele et al. 2024) have shown strong performance. However, most existing TSC models are trained on single-domain datasets, limiting their transferability across multiple domains.

Time Series Foundation Models. Time series FMs have seen rapid development (Liang et al. 2024). For example, recent studies exhibit strong forecasting generalization by pretraining on large-scale datasets merged from multiple domains (Woo et al. 2024; Li et al. 2025). Zhou et al. (2023) further demonstrate the adaptability of large language models, while Goswami et al. (2024) and Gao et al. (2024) design task-agnostic FMs pretrained on time series datasets for multiple downstream tasks (including classification). For classification-specific FMs, Lin et al. (2024) and Feofanov et al. (2025) introduce multi-scale normalization techniques, highlighting the potential of FMs for TSC. However, most efforts focus on enhancing classification accuracy, with limited attention to the interpretability of the results.

Time Series Shapelets. Shapelets have attracted considerable interest for enhancing the interpretability of TSC models (Liu et al. 2024; Wen et al. 2024). Traditional approaches (Ye and Keogh 2011; Rakthanmanon and Keogh 2013) rely on exhaustive searches to identify discriminative subsequences, incurring high computational costs. Recent deep learning methods (Li et al. 2021; Liu et al. 2025) adopt gradient-based frameworks to improve shapelet discovery efficiency. However, most existing methods are restricted to scenario-specific shapelet discovery, limiting their applicability as universal representations across diverse domains.

Background

Problem Statement. This paper focuses on building a foundation model for the univariate TSC problem, using a two-stage paradigm: pretraining and fine-tuning. The model is first pre-trained on a large-scale source dataset and then fine-tuned on domain-specific target datasets. Compared to training deep learning models from scratch, FMs provide better parameter initialization and enhanced generalization (Ma et al. 2024), making them particularly suitable for transfer learning across multiple target domains.

Formally, let the pretraining source dataset be denoted as $\mathcal{D}_s = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, where each time series sample $\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_T^{(i)}] \in \mathbb{R}^T$ has length T, and $y^{(i)}$ denotes its class label. A foundation model $f(\cdot)$ is first pre-trained on $\mathcal{D}_s$, and its parameters are then frozen and used to initialize downstream training. During fine-tuning, the model $f(\cdot)$ is further trained on the target dataset $\mathcal{D}_t = \{(\mathbf{x}_t^{(j)}, y_t^{(j)})\}_{j=1}^{M}$, where $\mathbf{x}_t^{(j)} \in \mathbb{R}^T$ and $y_t^{(j)}$ represent the input data and label in the target dataset. The goal is to use pre-trained knowledge learned from $\mathcal{D}_s$ to improve classification performance on the target dataset $\mathcal{D}_t$.

The Pretraining Source Dataset. A large-scale time series source dataset is essential for effective foundation model pretraining. However, most existing TSC datasets are small and domain-specific, limiting their utility for this purpose. To address this limitation, we construct a comprehensive pretraining dataset following (Lin et al. 2024), integrating three primary sources: (1) the UCR time series archive (Dau et al. 2019), (2) the UEA time series archive (Bagnall et al. 2018), and (3) eight additional datasets commonly used in prior studies (Eldele et al. 2021; Zhang et al. 2022).

To ensure consistency in input channels across domains, multivariate sequences are decomposed into distinct univariate series using a channel-independent transformation. To address inconsistencies in sequence length, all inputs are resized to a fixed length of 512 using PyTorch's interpolation function (Feofanov et al. 2025). To prevent test data leakage, only the training sets of each sub-dataset from the above three data sources are employed. The resulting source

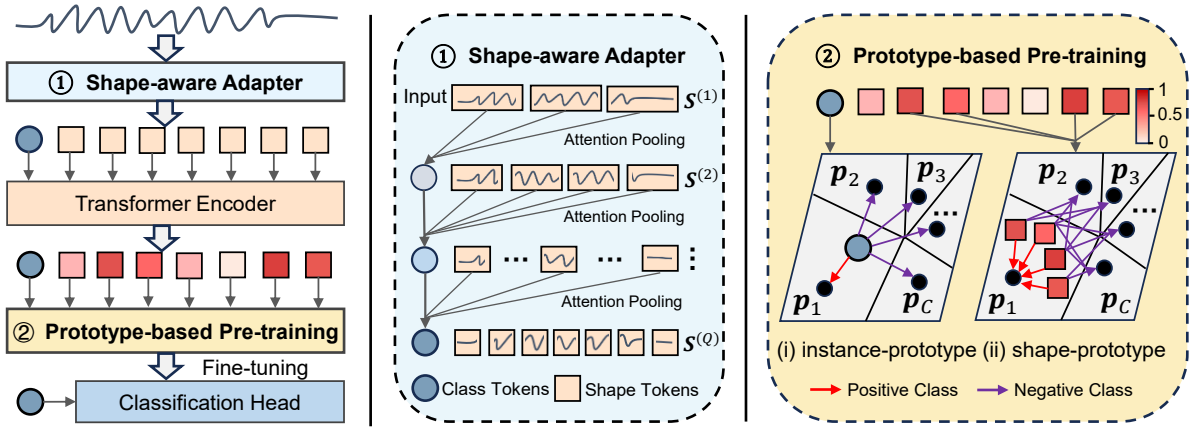

Figure 2: The overall architecture of the UniShape framework. UniShape comprises two core modules: ① a shape-aware adapter that takes variable-length subsequences $[\mathcal{S}^{(q)}]_{q=1}^{Q}$ as input and applies attention pooling to fuse discriminative patterns into class tokens; and ② a prototype-based pretraining module that jointly uses instance-prototype and shape-prototype contrastive learning to optimize prototype representations $\{\mathbf{p}_c\}_{c=1}^{C}$ based on instance-level class tokens and subsequence-level shape tokens.

dataset contains approximately 1.89 million univariate time series samples, providing a large-scale and diverse corpus suitable for time series foundation model pretraining.

Method

Architecture Overview

As shown in Figure 2, the overall architecture of UniShape comprises two key components: (i) a shape-aware adapter and (ii) a prototype-based pretraining module.

The shape-aware adapter begins by segmenting each input time series into multi-scale subsequences (or shapes) using sliding windows of varying lengths. Each group of fixed-length subsequences is processed by a shared-parameter adapter that aggregates shape tokens into a class token via attention pooling. These class tokens are then hierarchically fused in a top-down manner to capture discriminative representations of shapes across scales.

In the prototype-based pretraining module, the class and shape tokens generated by the adapter are fed into a transformer encoder. Class prototype learning is performed at two levels: instance-level, utilizing class tokens to capture global class-discriminative features across samples; and shape-level, selecting high-confidence shape tokens via class attention scores to model local discriminative patterns within individual samples. After pretraining, UniShape fine-tunes on target datasets by passing the final class token into a classification head, yielding the classification output.

Shape-Aware Adapter

Deep learning models typically require fixed-length inputs, posing challenges for handling variable-length time series with different sampling scales. A naive solution, which involves training separate models for each input length, is inefficient and hinders transferability in FM design. Inspired by the success of adapters in NLP for their parameter efficiency and scalability (Houlsby et al. 2019), we propose a *shape-aware adapter* to enable unified and efficient modeling of multi-scale subsequences within time series FMs.

Given a univariate time series $\mathbf{x} = [x_1, x_2, \ldots, x_T] \in \mathbb{R}^T$, we define Q temporal scales using sliding window configurations (W_q, K_q) for $q \in [1, Q]$, where W_q and K_q denote the window length and stride, respectively. Each scale q produces a subsequence set:

$$\mathcal{S}^{(q)} = \left\{ \mathbf{s}_i^{(q)} = [x_i, \ldots, x_{i+W_q-1}] \in \mathbb{R}^{W_q} \right\}, \quad (1)$$

where $i = 1, 1 + K_q, \ldots, T - K_q$. This yields $N_q = \left\lfloor \frac{T-W_q}{K_q} \right\rfloor + 1$ subsequences at scale q.

Each subsequence $\mathbf{s}_i^{(q)}$ is transformed into a d-dimensional shape token $\mathbf{z}_i^{(q)}$ via a lightweight normalization unit (Feofanov et al. 2025). To obtain $\mathbf{z}_i^{(q)}$, we first compute the mean μ and standard deviation σ of $\mathbf{x}$, as well as its first-order differential $\Delta\mathbf{x} = [x_2 - x_1, \ldots, x_T - x_{T-1}, 0]$. Each subsequence and its differential form are normalized: $\hat{\mathbf{s}}_i^{(q)} = \frac{\mathbf{s}_i^{(q)} - \mu}{\sigma}$, $\widehat{\Delta\mathbf{s}}_i^{(q)} = \frac{\Delta\mathbf{s}_i^{(q)} - \Delta\mu}{\Delta\sigma}$. Then, two 1D CNNs encode these normalized inputs: $\mathbf{h}_i^{(q)} = \mathrm{CNN}_1(\hat{\mathbf{s}}_i^{(q)})$, $\mathbf{g}_i^{(q)} = \mathrm{CNN}_2(\widehat{\Delta\mathbf{s}}_i^{(q)})$. In parallel, local mean $\mu_i^{(q)}$ and standard deviation $\sigma_i^{(q)}$ of each subsequence $\mathbf{s}_i^{(q)}$ are embedded using a numerically multi-scaled embedding module (Lin et al. 2024), yielding: $\mathbf{e}(\mu_i^{(q)}), \mathbf{e}(\sigma_i^{(q)}) \in \mathbb{R}^d$. Finally, all embeddings are concatenated and linearly projected into a shape token:

$$\mathbf{z}_i^{(q)} = \mathrm{Linear}\left(\left[\mathbf{h}_i^{(q)}, \mathbf{g}_i^{(q)}, \mathbf{e}(\mu_i^{(q)}), \mathbf{e}(\sigma_i^{(q)}) \right] \right) \in \mathbb{R}^d. \quad (2)$$

To extract class-discriminative representations from shape tokens, the adapter combines multi-resolution convolutional

encoding with attention-based aggregation in a lightweight design. Its core consists of three parallel 1D CNNs with varying kernel sizes (Ismail Fawaz et al. 2020), capturing discriminative temporal patterns across multiple resolutions. To aggregate these features into a class token, we employ an attention pooling mechanism with a linear complexity (Ilse, Tomczak, and Welling 2018). For each scale q, an attention head ψ_{ATTN} assigns weights $\alpha_i^{(q)} \in [0,1]$ to shape tokens $\mathbf{z}_i^{(q)} \in \mathbb{R}^d$ via two linear layers with $\tanh$ and sigmoid activations. The aggregated class token $\mathbf{c}^{(q)}$ is computed as:

$$\alpha_i^{(q)} = \psi_{\text{ATTN}}(\mathbf{z}_i^{(q)}), \quad \mathbf{c}^{(q)} = \sum_{i=1}^{N_q} \alpha_i^{(q)} \mathbf{z}_i^{(q)}, \tag{3}$$

where $\alpha_i^{(q)}$ reflects the discriminative importance of each shape, enhancing the interpretability of classification results.

To integrate information from multi-scale shape tokens, we adopt a coarse-to-fine (from larger to smaller W_q) hierarchical fusion strategy. At each scale $q > 1$, the previous class token $\mathbf{c}^{(q-1)}$ is prepended to the shape tokens:

$$\mathcal{Z}^{(q)} = \left[\mathbf{c}^{(q-1)}\right] \oplus \left[\mathbf{z}_i^{(q)}\right]_{i=1}^{N_q}, \tag{4}$$

which enables the hierarchical fusion of class tokens and aggregates discriminative patterns across shape scales.

After fusing the Q input scales, we obtain a final class token $\mathbf{c}^{(Q)}$ and corresponding shape tokens $[\mathbf{z}_i^{(Q)}]_{i=1}^{N_Q}$. Each fixed-scale subsequence set $\mathcal{S}^{(q)}$ is independently normalized to produce shape tokens $[\mathbf{z}_i^{(q)}]_{i=1}^{N_q}$. The shape-aware adapter then applies attention pooling to map each token set to a d-dimensional class token $\mathbf{c}^{(q)}$. This adapter, shared across all scales $q \in [1, Q]$, serves as a unified module that adaptively highlights discriminative temporal features. The design supports efficient multi-scale integration and facilitates transferability of shapelet patterns during pretraining.

Prototype-based Pretraining

Prototype learning captures class-level features by learning embeddings for each class, enabling strong generalization and domain adaptation with limited labeled data (Snell, Swersky, and Zemel 2017). Class labels play a vital role in guiding the learning of shapelet representations. This supervision helps the adapter to select the optimal temporal scales, enhancing hierarchical class token fusion. Notably, attention scores computed during shape token aggregation (Eq. (3)) can be ambiguous without class-level signals, making it difficult to assess the discriminative relevance of each shape token. To address this, we introduce a prototype-based pretraining module that aligns class and shape tokens with their respective class prototypes, thereby reducing reliance on large amounts of source labeled data. This is achieved through instance-level and shape-level contrastive objectives that encourage learning transferable shapelet patterns.

Instance-Prototype Contrastive Learning. To improve token representations, UniShape adopts a Transformer encoder as its backbone, given its demonstrated effectiveness in FM architectures (Liang et al. 2024). The class token $\mathbf{c}^{(Q)}$ and associated shape tokens $[\mathbf{z}_i^{(Q)}]_{i=1}^{N_Q}$, produced by the shape-aware adapter, are fed into the transformer encoder to obtain refined outputs $\mathbf{c}^{(Q)\prime}$ and $[\mathbf{z_i}^{(Q)\prime}]_{i=1}^{N_Q}$.

Contrastive learning has proven effective in pretraining by encouraging the model to learn discriminative features through the comparison of positive and negative pairs (Chen et al. 2020). Motivated by this, we introduce an *instance-level contrastive learning* strategy to align class tokens with their corresponding class prototypes. This approach is composed of two key components: (i) the initialization and optimization of a set of learnable class prototype vectors, and (ii) the formulation of a contrastive loss using the class tokens.

Specifically, we define a learnable prototype set as:

$$\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_c, \ldots, \mathbf{p}_C\}, \quad \mathbf{p}_c \in \mathbb{R}^d, \tag{5}$$

where C is the number of classes of the training dataset and $\mathbf{p}_c$ denotes the prototype embedding for class c. These prototypes are dynamically updated during training. For each labeled instance, the class token $\mathbf{c}^{(Q)\prime}$ is used to update its corresponding prototype via exponential moving average:

$$\mathbf{p}_y \leftarrow \beta \mathbf{p}_y + (1-\beta)\mathbf{c}^{(Q)\prime}, \tag{6}$$

where y is the ground-truth class label and $\beta \in (0,1)$ is the momentum coefficient.

For unlabeled samples, the class token $\mathbf{c}^{(Q)\prime}$ is assigned a pseudo-label by identifying the nearest class prototype $\mathbf{p}_+$ according to cosine similarity. The instance-prototype contrastive loss is defined as:

$$\mathcal{L}_{\text{ins}} = -\log \frac{\exp(\text{sim}(\mathbf{c}^{(Q)\prime}, \mathbf{p}_+)/\tau)}{\sum_{j=1}^{C} \exp(\text{sim}(\mathbf{c}^{(Q)\prime}, \mathbf{p}_j)/\tau)}, \tag{7}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and τ is a temperature scaling factor. Using class tokens for prototype learning allows the model to capture global class-discriminative features across samples.

Shape-Prototype Contrastive Learning. Due to potential intra-class distributional variance and the limited ability of class tokens to capture local discriminative features within the time series, relying solely on instance-level class tokens for prototype learning may hinder the acquisition of optimal shapelet representations. Therefore, we introduce *shape-level contrastive learning* based on the shape tokens $[\mathbf{z}_i^{(Q)\prime}]_{i=1}^{N_Q}$. The attention head from the adapter (Eq. (3)) are reused to select high-confidence shape tokens. Let $\mathcal{Z}_{\text{top}} \subset [\mathbf{z}_i^{(Q)\prime}]_{i=1}^{N_Q}$ denote the top-ϵ tokens with the highest scores. The shape-prototype contrastive loss is defined as:

$$\mathcal{L}_{\text{shape}} = \frac{1}{|\mathcal{Z}_{\text{top}}|} \sum_{\mathbf{z_i}^{(Q)\prime} \in \mathcal{Z}_{\text{top}}} -\log \frac{\exp(\text{sim}(\mathbf{z_i}^{(Q)\prime}, \mathbf{p}_+)/\tau)}{\sum_{j=1}^{C} \exp(\text{sim}(\mathbf{z_i}^{(Q)\prime}, \mathbf{p}_j)/\tau)}, \tag{8}$$

where the prototype $\mathbf{p}_+$ for each shape token is determined either by the ground-truth label or the pseudo-label of the corresponding class token $\mathbf{c}^{(Q)\prime}$.

Hence, the prototype-based pretraining loss combines both levels of contrastive learning:

$$\mathcal{L}_{\text{proto}} = (1-\lambda)\mathcal{L}_{\text{ins}} + \lambda \mathcal{L}_{\text{shape}}, \tag{9}$$

where $\lambda \in (0,1)$ is a hyperparameter that balances the contributions of the instance-level and shape-level losses.

The Overall Training Loss Function

Pretraining Loss. To improve training stability and increase sample diversity, we adopt the momentum contrastive learning framework of MoCo v3 (Chen, Xie, and He 2021) for pretraining. For each input time series $\mathbf{x}$, two augmented views $\mathbf{x}_1$ and $\mathbf{x}_2$ are generated via random cropping (Lin et al. 2024) and independently encoded to produce class and shape tokens. To support pretraining under weak supervision, we employ the MoCo v3 self-supervised contrastive loss, which enforces consistency between different views of the same instance. This objective promotes representation learning from large-scale unlabeled data and is defined as:

$$\mathcal{L}_{\text{self}} = -\log \frac{\exp\left(\text{sim}(\mathbf{q}, \mathbf{k}^+)/\tau\right)}{\sum_j \exp\left(\text{sim}(\mathbf{q}, \mathbf{k}_j)/\tau\right)}, \tag{10}$$

where $\mathbf{q}$ is the query embedding, $\mathbf{k}^+$ is the positive key from the momentum encoder.

Hence, the overall pretraining objective is formulated as:

$$\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{\text{proto}} + \mathcal{L}_{\text{self}}. \tag{11}$$

Fine-tuning Loss. The UniShape model is fine-tuned on the target dataset using the pre-trained adapter, encoder and a randomly initialized classification head. Supervised training is conducted using the cross-entropy loss:

$$\mathcal{L}_{\text{ce}} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{12}$$

where $\hat{y}_i$ denotes the predicted probability for class i of sample $\mathbf{x}_t$, and y_i is the one-hot encoded ground-truth label.

To enhance interpretability, we incorporate the $\mathcal{L}_{\text{shape}}$ as an auxiliary objective. This loss aligns shape tokens with their corresponding class labels, encouraging the model to focus on discriminative shapelet patterns. Therefore, the overall fine-tuning loss is defined as:

$$\mathcal{L}_{\text{finetune}} = \mathcal{L}_{\text{ce}} + \mu\mathcal{L}_{\text{shape}}, \tag{13}$$

where μ controls the training weight of $\mathcal{L}_{\text{shape}}$.

Experiments

Datasets. UniShape is pre-trained on a source dataset of 1.89 million samples using five subsequence scales ($Q = 5$), with window lengths and strides $W_q = K_q \in \{64, 32, 16, 8, 4\}$. The reason for these window length settings is detailed in Appendix A. For downstream TSC tasks, UniShape is evaluated on 158 univariate datasets, including 128 UCR time series datasets widely used for classification tasks (Dau et al. 2019). To evaluate zero-shot generalization, we further assess performance on 30 datasets from diverse domains (Middlehurst, Schäfer, and Bagnall 2024), which are not included in the pretraining source dataset. All datasets use their official train/test splits for evaluation.

Baselines. We compare UniShape against 16 methods, grouped as follows: (i) **Non-deep learning (NDL)**: Rocket (Dempster, Petitjean, and Webb 2020), MiniRocket (Dempster, Schmidt, and Webb 2021), RDST (Guillaume, Vrain, and Elloumi 2022), MultiRocket-Hydra (MR-H) (Dempster, Schmidt, and Webb 2023); (ii) **Domain-specific deep learning (DS)**: InceptionTime (Ismail Fawaz et al. 2020), TS2Vec (Yue et al. 2022), PatchTST (Nie et al. 2023), TimesNet (Wu et al. 2023), SoftShape (Liu et al. 2025); (iii) **Foundation models (FMs)**: GPT4TS (Zhou et al. 2023), MOMENT (Goswami et al. 2024), UniTS (Gao et al. 2024), NuTime (Lin et al. 2024), Mantis (Feofanov et al. 2025). We also include two forecasting-based FMs for zero-shot classification: Chronos (Ansari et al. 2024) and Moirai (Woo et al. 2024). All baselines use author-recommended hyperparameters in a unified Python environment for fair comparison.

	Method	# Params	Avg. Acc	Avg. Rank	P-value
NDL	Rocket	-	0.8487	4.87	7.80E-06
	MiniRocket	-	0.8545	4.84	3.25E-03
	RDST	-	0.8571	4.80	8.58E-03
	MR-H	-	0.8621	3.97	2.96E-02
DS	InceptionTime	386.9 K	0.8315	6.10	2.55E-11
	TS2Vec	637.2 K	0.8016	8.32	3.12E-11
	PatchTST	431.2 K	0.6500	12.69	2.12E-26
	TimesNet	7.4 M	0.6897	11.83	3.08E-24
	SoftShape	472.5 K	0.8388	5.89	3.68E-32
FMs	GPT4TS*	84.1 M	0.7100	11.69	7.62E-23
	MOMENT*	341.2 M	0.7020	12.10	7.04E-25
	UniTS	1.1 M	0.7357	11.43	3.20E-23
	NuTime	2.4 M	0.8353	6.68	2.08E-10
	Mantis	8.7 M	0.8441	5.21	1.69E-06
	UniShape	3.1 M	**0.8708**	**2.71**	-

Table 1: The statistical test results comparison on 128 UCR datasets under the fully supervised setting. *# Params* denotes the number of parameters in baselines. * indicates that GPT4TS and MOMENT utilize only 2.9 million (M) and 3.1 thousand (K) parameters for fine-tuning, respectively. Best results are in **bold**.

Implementation Settings. UniShape training comprises two stages: pretraining and fine-tuning. Pretraining is conducted for up to 30 epochs with a batch size of 2048. By default, prototype-based pretraining uses 10% labeled data, a momentum coefficient $\beta = 0.9$, shape token selection ratio $\epsilon = 60\%$, and a weighting factor $\lambda = 0.01$. Fine-tuning runs for up to 300 epochs with an auxiliary loss weight $\mu = 0.01$. For deep learning-based methods, we follow the settings in (Early et al. 2024) and report test classification accuracy using the model checkpoint with the lowest training loss. For all baselines and UniShape, results are averaged over five independent runs with different random seeds. All experiments are conducted using Python 3.9.21, PyTorch 1.12.1, and four NVIDIA RTX A6000 GPUs.

Further details on the datasets, baselines, and implementation settings are provided in Appendix A and available at https://github.com/qianlima-lab/UniShape.

Overall Evaluation Results

Table 1 presents the test classification performance of UniShape compared to baselines on 128 UCR datasets un-

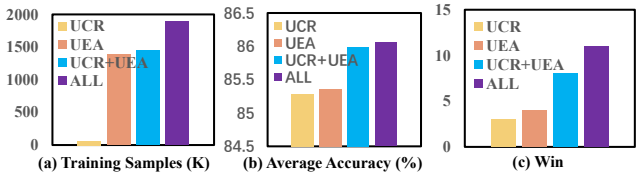

Figure 3: Results comparison on 18 UCR datasets with different numbers of pretraining samples. *Win* denotes the number of datasets where the method performs best.

Labeling Ratio	0%	1%	10%	50%	100%
Avg. Acc	0.8395	0.8410	0.8529	0.8574	**0.8588**
P-value	1.87E-02	3.28E-02	2.00E-01	4.10E-01	-

Table 2: The statistical test classification results comparison on 18 UCR datasets with different training sample labeling ratios for pretraining. The best result is in **bold**.

der a fully supervised setting. Detailed per-dataset results are provided in Appendix B. Each method is evaluated using three metrics: *Avg. Acc* (average accuracy across 128 datasets), *Avg. Rank* (test accuracy-based rank), and P-value (Wilcoxon signed-rank test (Demšar 2006) assessing whether UniShape significantly outperforms each baseline; significance at $p < 0.05$). Parameter counts exclude NDL-based methods due to their non-trainable architectures.

UniShape achieves the highest *Avg. Acc* and the lowest *Avg. Rank*, indicating strong generalization across diverse target datasets. GPT4TS uses a pre-trained language model for fine-tuning, while MOMENT and UniTS lack design considerations for TSC tasks, resulting in substantially lower accuracy than most NDL and DS-based methods. This suggests that task-agnostic FMs not specifically designed for TSC may struggle to adapt effectively to the classification task. In contrast, UniShape significantly outperforms other FMs such as NuTime and Mantis, despite having only 3.1 million parameters. These findings collectively demonstrate that UniShape provides a more effective and parameter-efficient foundation for TSC.

Pretraining Analysis

We analyze the impact of pretraining on two dimensions: the size of the pretraining dataset and the labeling ratio of its samples. Following (Liu et al. 2025), and considering the computational cost of using all 128 UCR datasets as target domains, we select 18 UCR datasets that vary in domain, class count, and sample size for analysis. Figure 3 illustrates UniShape's performance across four pretraining scales: UCR (∼60K), UEA (∼1.39M), UCR+UEA (∼1.45M), and ALL (1.89M samples). Classification accuracy consistently improves with larger pretraining datasets, indicating that data scale enhances target-domain performance, while pretraining solely on the smaller UCR archive still yields competitive fine-tuning results.

Pretraining on the full 1.89 million samples is time-consuming per run, making it computationally intensive across all labeling ratios. For data scale analysis results in

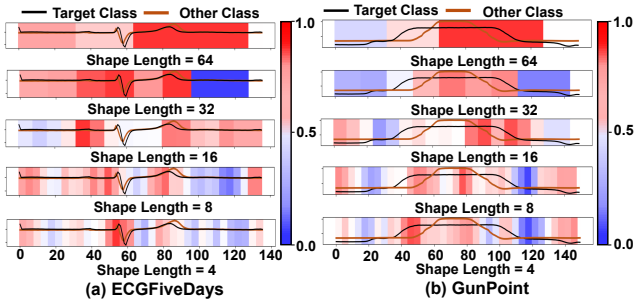

Figure 4: Visualization of attention scores learned by the shape-aware adapter across different shape lengths. Darker red denotes higher attention, highlighting discriminative regions for the target class, while darker blue indicates lower attention and reduced relevance to target features.

Figure 3, we employ only the training samples from the UCR archive for pretraining to reduce computational cost. In Table 2, the 0% labeling ratio initializes prototypes randomly and updates them via pseudo-labels during pretraining. The 100% labeled setting achieves the highest accuracy, demonstrating the effectiveness of incorporating supervised class information for fine-tuning. However, performance gains beyond 10% labeling are marginal and statistically insignificant (P-value > 0.05). Hence, we adopt 10% labeling as the default setting for pretraining to balance performance and annotation cost. Detailed results of Figure 3 and Table 2, please refer to Appendix C.

Interpretability Analysis

To evaluate the interpretability of UniShape, we examine the attention scores produced by the shape-aware adapter, specifically the attention head ψ_{ATTN}, across multiple shape lengths. As shown in Figure 4, we conduct this analysis on two representative UCR datasets: *ECGFiveDays* and *GunPoint*, drawn from the healthcare and motion domains, respectively. In *ECGFiveDays*, Rakthanmanon and Keogh (2013) identified the delayed *T-wave* in the [75, 95] interval as the key discriminative segment. UniShape consistently allocates high attention to this region, as well as to some regions with minimal temporal differences. For *GunPoint* (Ye and Keogh 2011), which differentiates *Gun* and *NoGun* gestures, critical segments lie in [30, 60] and [90, 110] due to motion overshoot. UniShape effectively assigns high attention to these intervals, selecting them as shapelets to enhance classification interpretability. These results highlight UniShape's advantages over existing time series FMs, which often neglect the interpretability of classification results.

Results on Zero-Shot Feature Extraction

Zero-shot learning evaluates a model's ability to generalize to unseen data distributions without fine-tuning on the target domain (Pourpanah et al. 2022), providing a critical measure of the generalization ability of FMs. Table 3 presents the zero-shot feature extraction performance of UniShape and seven time series FMs on 30 additional time series datasets from diverse domains. For detailed results of Table 3, please

Method	Avg. Acc	Avg. Rank	P-value
RandomForest	0.6930	3.77	2.57E-02
GPT4TS	0.5600	6.37	1.79E-06
MOMENT	0.6972	4.17	3.98E-02
Chronos	0.6793	4.10	4.91E-03
Moirai	0.5691	6.37	5.69E-06
UniTS	0.3431	8.90	6.24E-10
NuTime	0.6917	3.53	3.36E-03
Mantis	0.7052	3.67	3.15E-02
UniShape	**0.7262**	**3.07**	-

Table 3: The statistical test results comparison on 30 additional datasets with a zero-shot feature extraction. Best results are in **bold**.

refer to Appendix D. Following the setup in (Feofanov et al. 2025), each dataset is processed using a frozen FM to extract representations for both training and test sets. A Random Forest classifier (Breiman 2001) is then trained on these representations to assess classification performance. A baseline using Random Forest trained directly on raw time series is also included for comparison.

As shown in Table 3, UniShape outperforms all baselines. Mantis and NuTime also perform competitively, with better *Avg. Rank* than the RandomForest baseline. In contrast, GPT4TS, Moirai, and UniTS perform poorly for TSC in the zero-shot setting, suggesting a limited ability to capture classification-specific temporal patterns. These results demonstrate that UniShape exhibits strong generalization and transferability in zero-shot scenarios, underscoring its potential as a universal time series FM to extract meaningful class-discriminative shapelet features without fine-tuning.

Ablation Study

This section conducts a comprehensive ablation study examining the training paradigm and model architectures. Following the setup in Table 2, we only use the UCR archive for pretraining and the same 18 UCR datasets as target domains to reduce computational cost. Two training paradigms are evaluated: (a) **w/o Pretraining**: models trained from scratch on target datasets. (b) **Pretraining and Fine-tuning**: models are first pre-trained and then fine-tuned for classification. The model architecture ablation includes three components: (a) **Adapter Module**: i) *w/o Adapter*: removes the shape-aware adapter, using a fixed-length subsequence set as input. ii) *re Trans*: replaces the CNN layer in the adapter with a Transformer. iii) *re MLP*: replaces the CNN layer with an MLP. (b) **Prototype-based pretraining**: i) *w/o Ins*: removes the instance-prototype contrastive loss. ii) *w/o Shape*: removes the shape-prototype contrastive loss. iii) *w/o Proto*: removes both instance and shape prototype losses. (c) **Transformer Encoder**: i) *re CNN*: replaces the Transformer encoder with an Inception-based CNN. ii) *re MLP*: replaces the Transformer encoder with an MLP.

Table 4 reports the statistical results across all ablation settings. Detailed results for Table 4 are provided in Appendix E. Excluding the pretraining phase leads to a substantial performance decline, underscoring the importance of pretraining. Substituting the CNN layer in the adapter with a Transformer or MLP further reduces performance, indicating that CNNs are more effective for multi-scale shapelet learning. In the prototype-based pretraining module, *w/o Ins* yields lower performance than *w/o Shape*, suggesting that instance-prototype pretraining is more crucial for shapelet representation learning. Removing both components (*w/o Proto*) further degrades performance, confirming their complementary roles. Previous work (Eldele et al. 2024) has shown that CNNs outperform Transformers in a domain-specific training setting for TSC tasks. However, under the FM setting with pretraining and fine-tuning, the Transformer encoder achieves a better *Avg. Rank* than the CNN-based encoder. In contrast, replacing the Transformer with an MLP as the encoder leads to poor classification performance, even worse than training from scratch. This indicates that MLPs struggle to learn generalizable shapelet representations for TSC through pretraining, compared to Transformers and CNNs. Further analysis of hyperparameters ϵ, λ, μ, and runtime is presented in Appendix F.

Method		Avg. Acc	Avg. Rank	P-value
UniShape		**0.8529**	**3.00**	–
w/o Pretraining				
Encoder	using Trans	0.8365 (-1.64%)	6.33	2.96E-02
	re CNN	0.8264 (-2.65%)	6.61	2.33E-02
	re MLP	0.6832 (-17.0%)	8.11	3.35E-03
Pretraining and Fine-tuning				
Adapter	w/o Adapter	0.8428 (-1.01%)	5.50	1.65E-02
	re Trans	0.8446 (-0.83%)	5.72	1.68E-02
	re MLP	0.8431 (-0.98%)	6.78	2.70E-02
Prototype	w/o Ins	0.8444 (-0.85%)	5.56	5.92E-03
	w/o Shape	0.8470 (-0.59%)	4.39	2.11E-02
	w/o Proto	0.8411 (-1.18%)	5.89	8.15E-03
Encoder	re CNN	0.8512 (-0.16%)	3.94	3.40E-01
	re MLP	0.5651 (-28.8%)	9.61	4.11E-04

Table 4: Ablation results on 18 UCR time series datasets. Among them, *w/o* means without, and *re* means replace.

Conclusion

We present a unified shape-aware foundation model named UniShape for time series classification. UniShape integrates a shape-aware adapter and a prototype-based pretraining module, enabling effective learning of multi-scale shapelet representations. By pretraining on a large-scale source dataset, UniShape captures transferable shapelet patterns applicable to diverse target domains. Experiments show that UniShape outperforms baselines in both fully supervised and zero-shot learning settings. Yet, this work focuses solely on univariate TSC. In the future, we aim to extend UniShape toward modeling multivariate dependencies for more generalizable foundation models.

Acknowledgments

We thank the anonymous reviewers for their helpful feedbacks. We thank Professor Eamonn Keogh and all the people who have contributed to the UCR & UEA archives and other time series datasets. The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant Nos. 62272173, 62273109), 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), the MTI WS Fund - AI for Manufacturing COE - Common Model Projects (Grant No. W24MCMF012), and the China Scholarship Council program (Grant No. 202406150081).

References

Ansari, A. F.; Stella, L.; Turkmen, A. C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Arango, S. P.; Kapoor, S.; et al. 2024. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*.

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

Breiman, L. 2001. Random forests. *Machine learning*, 45: 5–32.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 1597–1607.

Chen, X.; Xie, S.; and He, K. 2021. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9640–9649.

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

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

Dempster, A.; Schmidt, D. F.; and Webb, G. I. 2021. Minirocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 248–257.

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

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

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

Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; Kwoh, C. K.; Li, X.; and Guan, C. 2021. Time-series representation learning via temporal and contextual contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2352–2359.

Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; and Li, X. 2024. TSLANet: rethinking transformers for time series representation learning. In *International Conference on Machine Learning*, 12409–12428.

Feofanov, V.; Wen, S.; Alonso, M.; Ilbert, R.; Guo, H.; Tiomoko, M.; Pan, L.; Zhang, J.; and Redko, I. 2025. Mantis: Lightweight calibrated foundation model for user-friendly time series classification. *arXiv preprint arXiv:2502.15637*.

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

Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. MOMENT: A family of open time-series foundation models. In *International Conference on Machine Learning*, 16115–16152.

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

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

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2790–2799.

Ilse, M.; Tomczak, J.; and Welling, M. 2018. Attention-based deep multiple instance learning. In *International Conference on Machine Learning*, 2127–2136.

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

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. 2020. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6): 1936–1962.

Keogh, E.; and Kasetty, S. 2002. On the need for time series data mining benchmarks: a survey and empirical demonstration. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 102–111.

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

Li, Z.; Qiu, X.; Chen, P.; Wang, Y.; Cheng, H.; Shu, Y.; Hu, J.; Guo, C.; Zhou, A.; Jensen, C. S.; et al. 2025. Tsfm-bench: A comprehensive and unified benchmark of foundation models for time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5595–5606.

Liang, Y.; Wen, H.; Nie, Y.; Jiang, Y.; Jin, M.; Song, D.; Pan, S.; and Wen, Q. 2024. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 6555–6565.

Lin, C.; Wen, X.; Cao, W.; Huang, C.; Bian, J.; Lin, S.; and Wu, Z. 2024. NuTime: Numerically multi-scaled embedding for large-scale time-series pretraining. *Transactions on Machine Learning Research*.

Liu, Z.; Luo, Y.; Li, B.; Eldele, E.; Wu, M.; and Ma, Q. 2025. Learning soft sparse shapes for efficient time-series classification. In *International Conference on Machine Learning*.

Liu, Z.; Ma, P.; Chen, D.; Pei, W.; and Ma, Q. 2023a. Scale-teaching: robust multi-scale training for time series classification with noisy labels. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 33726–33757.

Liu, Z.; Ma, Q.; Ma, P.; and Wang, L. 2023b. Temporal-frequency co-training for time series semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 8923–8931.

Liu, Z.; Pei, W.; Lan, D.; and Ma, Q. 2024. Diffusion language-shapelets for semi-supervised time-series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 14079–14087.

Luo, Y.; Liu, Z.; Wang, L.; Wu, B.; Zheng, J.; and Ma, Q. 2024. Knowledge-empowered dynamic graph network for irregularly sampled medical time series. *Advances in Neural Information Processing Systems*, 37: 67172–67199.

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

Middlehurst, M.; Schäfer, P.; and Bagnall, A. 2024. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, 38(4): 1958–2031.

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

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

Pourpanah, F.; Abdar, M.; Luo, Y.; Zhou, X.; Wang, R.; Lim, C. P.; Wang, X.-Z.; and Wu, Q. J. 2022. A review of generalized zero-shot learning methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4051–4070.

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

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30.

Wen, Y.; Ma, T.; Weng, L.; Nguyen, L.; and Julius, A. A. 2024. Abstracted shapes as tokens-a generalizable and interpretable model for time-series classification. *Advances in Neural Information Processing Systems*, 37: 92246–92272.

Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; and Sahoo, D. 2024. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning*, 53140–53164.

Woo, G.; Liu, C.; Sahoo, D.; Kumar, A.; and Hoi, S. 2022. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International Conference on Learning Representations*.

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

Yamaguchi, A.; Ueno, K.; and Kashima, H. 2023. Time-series shapelets with learnable lengths. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2866–2876.

Ye, L.; and Keogh, E. 2011. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data Mining and Knowledge Discovery*, 22(1): 149–182.

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

Zhang, J.; Huang, J.; Jin, S.; and Lu, S. 2024. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8): 5625–5644.

Zhang, X.; Zhao, Z.; Tsiligkaridis, T.; and Zitnik, M. 2022. Self-supervised contrastive pre-training for time series via time-frequency consistency. In *Advances in Neural Information Processing Systems*, 3988–4003.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in Neural Information Processing Systems*, 36: 43322–43355.