DTW4TS: LEARNING TIME SERIES REPRESENTATIONS FROM RELIABLE DTW RELATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world time series inherently exhibit complex temporal misalignments, posing challenges for current representation learning methods that primarily rely on rigid, point-to-point temporal matching. While Dynamic Time Warping (DTW) naturally accommodates such misalignments through nonlinear alignment, existing DTW-based methods often use DTW relations directly as supervision and mainly rely on scalar similarities, leaving relation reliability and fine-grained warping path correspondences comparatively underexplored. To address these limitations, we propose DTW4TS, a reliability-aware framework for time series representation learning. Rather than utilizing all scalar scores uniformly, DTW4TS first mitigates the impact of unreliable DTW relations by assessing local neighborhood consistency and alignment path deviation. We further introduce a lightweight DTW Adapter that encodes reliable warping paths and injects alignment information into intermediate encoder layers. In addition, an elastic contrastive objective constructs cross-sequence positive pairs from reliable temporal correspondences, encouraging the learned representations to capture elastic relationships across time series. Experiments across classification, anomaly detection, and forecasting benchmarks show that DTW4TS achieves the highest average classification accuracy and anomaly-detection F1, and the lowest average forecasting MSE, among the evaluated baselines. The source code is available at `https://anonymous.4open.science/r/DTW4TS-E343`.

1 INTRODUCTION

Time series data are ubiquitous in real-world dynamic systems, with applications ranging from recognizing human movement patterns in sensor signals (Xu et al., 2025) to forecasting weather from historical meteorological observations (Wu et al., 2021) and detecting faults in equipment operating states (Yin et al., 2026). These tasks typically require models to capture underlying temporal dynamics from continuous observations. Consequently, learning discriminative time series representations has become a central research problem in time series analysis.

In recent years, a variety of general Time Series Representation Learning (TSRL) methods have been proposed to transform raw time series into informative representations that support downstream tasks such as classification, anomaly detection, and forecasting (Trirat et al., 2026). Unlike images and text, time series often exhibit complex temporal dependencies and periodic patterns, motivating model designs tailored to these characteristics (Wu et al., 2022). For instance, TimesNet (Wu et al., 2022) and PlanTS (Wang et al., 2025) exploit the multi-periodicity of time series by organizing them into 2D structures to capture complex temporal dynamics. Meanwhile, SoftCLT (Lee et al., 2024) and AdaTS (Kara et al., 2025) employ instance-wise and temporal contrastive objectives to improve the discriminative quality of the learned representations. Despite significant advances in encoder architectures and learning objectives, how to identify useful relations between temporally misaligned sequences and exploit their fine-grained temporal correspondences for representation learning remains comparatively underexplored.

Dynamic Time Warping (DTW) explicitly models temporal misalignment through nonlinear alignment (Salvador & Chan, 2007), providing a useful elastic prior for TSRL. Existing methods such as TS-SD and SoftCLT use DTW-derived distances to construct supervisory signals (Shi et al., 2021; Lee et al., 2024). However, a scalar similarity score alone provides limited information about the

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

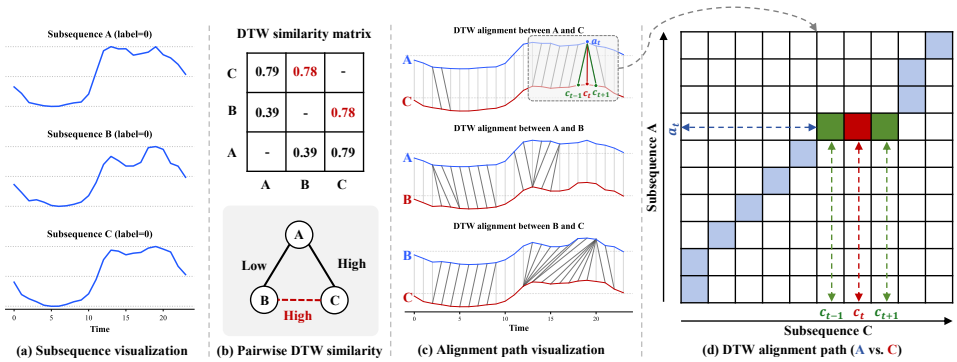


Figure 1: Illustration of DTW similarity inconsistency and elastic temporal alignment. (a) Three subsequences from the same class. (b) Pairwise DTW similarities, where A–C and B–C have high similarity, while A–B has low similarity. (c) Their pairwise DTW alignments. (d) A segment of warping path between A and C, illustrating nonlinear temporal correspondences.

suitability of a relation for learning. In practice, high DTW similarity may result from excessive temporal warping, whereas low similarity does not necessarily imply semantic dissimilarity. As illustrated in Figures 1(a) and 1(b), subsequences A, B, and C belong to the same class. A–C exhibit similar shapes and yield high DTW similarity, whereas A–B yield substantially lower similarity despite their shared semantic category. Meanwhile, B–C are also assigned high similarity, although the corresponding warping path exhibits pronounced local deviation, as shown in Figure 1(c). Consequently, C is considered similar to both A and B, whereas A–B remain weakly related, forming an inconsistent local similarity structure. Together, these examples illustrate that same-class membership can coexist with substantially different DTW similarities, while comparable similarity scores can conceal different alignment structures. This motivates complementing pairwise DTW scores with neighborhood and path information when selecting relations for supervision.

Moreover, existing DTW-based TSRL methods mainly exploit scalar similarities, leaving the fine-grained temporal information encoded in warping paths comparatively underutilized. Beyond the scalar distance, DTW also produces a warping path that specifies nonlinear temporal correspondences, including one-to-many and many-to-one matches between time points (Cuturi & Blondel, 2017). As shown in Figure 1(d), when aligning subsequences A and C, a time point a_t can be matched not only to c_t but also to its neighboring points c_{t-1} and c_{t+1}. Such correspondences reveal how similar local patterns are shifted or stretched along the temporal axis, providing richer structural information than a scalar similarity alone. Nevertheless, how to assess the reliability of DTW-based relations and selectively exploit their warping paths for TSRL remains underexplored.

To address these limitations, we propose DTW4TS, a reliability-aware framework for incorporating DTW relations and warping paths into time series representation learning. First, we design a strategy to assess the reliability of DTW relations from two complementary perspectives: local neighborhood consistency and alignment path deviation. We then introduce a lightweight DTW Adapter that encodes the geometry of reliable warping paths and injects the resulting alignment features into intermediate encoder layers through gated residuals. Finally, we develop an elastic contrastive objective that constructs cross-sequence positive pairs from temporal correspondences along reliable warping paths, encouraging the learned representations to capture elastic relationships across time series. These components enable DTW4TS to selectively exploit reliable DTW relations at both the architectural and objective levels, providing a reliability-aware mechanism for incorporating elastic alignment information into time series representation learning. Our main contributions are:

- **Reliability-Aware Framework:** We propose DTW4TS, a novel time series representation learning framework that goes beyond scalar DTW similarities to selectively exploit relations based on their estimated reliability. By jointly evaluating local neighborhood consistency and alignment path deviation, DTW4TS ranks and selects candidate relations to provide reliability-aware elastic supervision.

- **Dual-Level Structural Exploitation:** We design a lightweight DTW Adapter and an elastic contrastive objective to exploit selected warping paths at both the architectural and objective levels. The adapter encodes path geometry and injects the resulting alignment features into intermediate encoder layers through gated residuals, while the contrastive objective constructs cross-sequence positives from the selected temporal correspondences to capture elastic relationships across time series. The adapter is used only during training, requiring neither DTW computation nor adapter execution at inference.
- **Superior Performance Across Diverse Tasks:** Extensive experiments across classification, anomaly detection, and forecasting demonstrate relative gains of 1.19% in average classification accuracy and 12.47% in average anomaly-detection F1, together with an 11.82% reduction in average forecasting MSE, compared with the strongest competing baseline for each aggregate metric.

2 Related Work

Time Series Representation Learning Given its significant practical value, Time Series Representation Learning (TSRL) has received increasing attention in recent years. Existing methods mainly follow two directions: encoder architecture design and learning objective optimization (Trirat et al., 2026). Architecture-centric methods improve temporal modeling through specialized encoders, such as TimesNet (Wu et al., 2022) and MOMENT (Goswami et al., 2024), while objective-centric methods focus on representation supervision, as exemplified by TS2Vec (Yue et al., 2022), SoftCLT (Lee et al., 2024), and PlanTS (Wang et al., 2025). Due to space limitations, a more detailed review of representative TSRL methods is provided in Appendix A.

DTW-Based Time Series Learning Dynamic Time Warping (DTW) is a classic sequence alignment method that measures the distance between two time series under nonlinear temporal alignment (Berndt & Clifford, 1994; Chao et al., 2025). Owing to its ability to accommodate phase shifts and local temporal variations, DTW has been widely used in sequential data analysis (Jing et al., 2016; Wang et al., 2023). Recent studies have further incorporated DTW into representation learning. TimeCLR (Yang et al., 2022a) uses DTW to generate temporally warped augmentations, while TS-SD (Shi et al., 2021) and SoftCLT (Lee et al., 2024) use DTW-based distances to construct supervisory signals. Beyond time series, TempCLR (Yang et al., 2022b) employs DTW for sequence-level video-text alignment, while Hadji et al. (2021) exploit global temporal alignment and cycle consistency for temporal representation learning. Differentiable variants such as Soft-DTW and DecDTW further enable temporal alignment to be optimized jointly with neural models (Cuturi & Blondel, 2017; Xu et al., 2023). DTWformer (Yu et al., 2025) integrates Soft-DTW into the attention mechanism. However, within general Time Series Representation Learning, the reliability of cross-instance DTW relations and the joint use of selected warping paths at both the architectural and objective levels remain largely unaddressed. Motivated by this gap, we investigate the reliability-aware exploitation of DTW relations and warping paths for TSRL.

3 Method

Problem Definition. Given an unlabeled time series dataset $\mathcal{X} = \{x_i\}_{i=1}^{N}$ with $x_i \in \mathbb{R}^{T \times C}$, where N, T, and C denote the numbers of instances, timestamps, and channels, respectively, our goal is to learn an encoder f_θ that produces $r_i = f_\theta(x_i) \in \mathbb{R}^{T \times M}$. Here, M is the representation dimension and $r_{i,t} \in \mathbb{R}^M$ denotes the representation of timestamp t in x_i. The learned representations should capture discriminative temporal patterns and generalize across diverse downstream tasks.

Method Overview. As illustrated in Figure 2, DTW4TS applies crop-based augmentation to generate two views of the training series. In parallel, DTW reliability is assessed through local neighborhood consistency and alignment path deviation to identify reliable pairs and their corresponding warping paths. The selected pairs and paths are then mapped to the augmented views and fed into a shared DTW-aware encoder. Building on the TS2Vec-style encoder, the DTW-aware encoder introduces a lightweight DTW Adapter that injects encoded path information into intermediate layers through gated residuals. During training, reliable temporal correspondences are further leveraged by an elastic contrastive loss to provide elastic supervision across sequences.

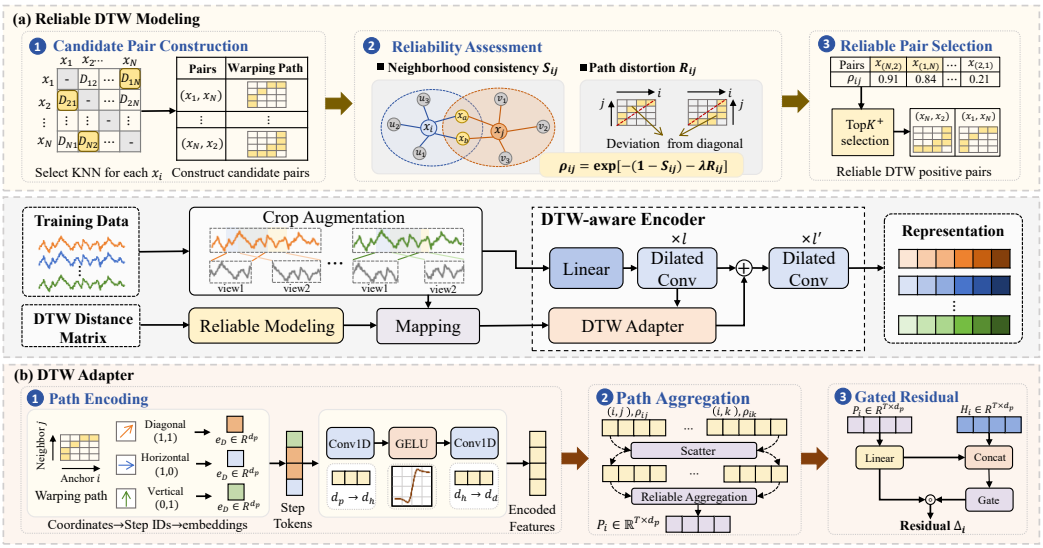


Figure 2: The overview of DTW4TS. Crop-based augmented views and reliable DTW pairs are jointly processed by a DTW-aware encoder to produce time series representations. (a) Reliable DTW modeling constructs candidate pairs, assesses their reliability from neighborhood consistency and path deviation, and selects reliable DTW pairs. (b) The DTW Adapter encodes reliable warping paths, aggregates path features, and injects gated residuals into intermediate encoder layers.

3.1 Reliable DTW Modeling

To reduce reliance on potentially misleading DTW relations as supervision, we propose a reliability modeling method before training. It identifies instance pairs that exhibit high structural reliability while retaining their warping paths for subsequent DTW-aware encoding and contrastive learning. As illustrated in Figure 2(a), this procedure consists of three stages: candidate pair construction, reliability assessment, and reliable pair selection.

Candidate Pair Construction. Given a time series dataset $\mathcal{X}$, we use a precomputed pairwise DTW distance matrix $\mathbf{D}$ as input to reliable DTW modeling, where D_{ij} denotes the DTW distance between x_i and x_j. For each anchor x_i, we retrieve its K-nearest neighbors according to DTW distance and construct candidate pairs between the anchor and these neighbors. The corresponding warping paths are retained for subsequent reliability assessment. Formally,

$$\mathcal{N}_i^K = \underset{j \neq i}{\arg\,\mathrm{TopK}}\left(-D_{ij}\right), \quad \mathcal{C}_i = \left\{(i,j) \mid j \in \mathcal{N}_i^K\right\}, \quad \mathcal{A}_{ij} = \{(a_n, b_n)\}_{n=1}^{|\mathcal{A}_{ij}|}, (i,j) \in \mathcal{C}_i \quad (1)$$

where $\mathcal{N}_i^K$, $\mathcal{C}_i$, and $\mathcal{A}_{ij}$ denote the K-nearest neighbor set of x_i, the resulting set of candidate pairs for x_i, and the warping path of pair (i,j), respectively. Each (a_n, b_n) denotes the n-th aligned index pair along the warping path.

Reliability Assessment. We assess the reliability of each candidate pair from local neighborhood consistency and alignment path deviation. First, if two instances share a substantial fraction of neighbors within their respective DTW-based candidate neighborhoods, their relation is more strongly supported by the surrounding samples. We therefore measure neighborhood consistency using the normalized overlap between their K-nearest DTW neighborhoods. Second, the DTW warping path characterizes nonlinear temporal correspondences through diagonal, horizontal, and vertical transitions. While moderate deviations from the main diagonal accommodate temporal shifts, excessive deviations may be associated with over-warping and less stable local alignments. We therefore quantify path deviation as the average normalized distance from the main diagonal. Based on these two complementary measures, we define the relation reliability score as:

$$S_{ij} = \frac{|\mathcal{N}_i^K \cap \mathcal{N}_j^K|}{K-1}, \quad R_{ij} = \frac{1}{|\mathcal{A}_{ij}|}\sum_{n=1}^{|\mathcal{A}_{ij}|}\frac{|a_n - b_n|}{\max(T-1,1)}, \quad \rho_{ij} = \exp\left[-\left(1 - S_{ij}\right) - \gamma R_{ij}\right] \quad (2)$$

where γ controls the contribution of path deviation relative to the neighborhood inconsistency term $1 - S_{ij}$. A smaller S_{ij} or a larger R_{ij} therefore reduces the reliability score ρ_{ij}.

Reliable Pair Selection. The DTW-based candidate neighborhood already restricts pair selection to locally similar instances. We therefore use the reliability score to further distinguish reliable relations within this local candidate set. For each anchor x_i, the K^+ candidates with the highest reliability scores are selected from its K candidate neighbors as reliable DTW neighbors. Formally,

$$\mathcal{N}_i^+ = \underset{j \in \mathcal{N}_i^K}{\arg \operatorname{TopK}^+} \left(\rho_{ij}\right), K^+ < K \tag{3}$$

where $\mathcal{N}_i^+$ contains the reliable neighbors selected for anchor x_i, and each $j \in \mathcal{N}_i^+$ forms a reliable DTW pair (i, j) with the anchor.

3.2 DTW Adapter for Representation Encoding

Reliable DTW modeling identifies reliable pairs and their corresponding warping paths for each anchor. However, these discrete paths cannot be directly integrated into a representation encoder. To bridge this gap, we introduce a lightweight DTW Adapter that encodes local path geometry into anchor-aligned representations and combines them with intermediate encoder features to generate gated residuals. As illustrated in Figure 2(b), the DTW Adapter consists of three stages: path encoding, path feature aggregation, and gated residual generation. The resulting residuals are injected into selected intermediate layers of the TS2Vec (Yue et al., 2022) encoder.

Path Encoding. For each reliable pair (x_i, x_j), let $\widetilde{\mathcal{A}}_{ij}$ denote the warping path mapped to the current augmented views, which retains only the valid temporal correspondences within the crop ranges. The diagonal, horizontal, and vertical transitions between consecutive path points characterize one-to-one and asymmetric one-to-many/many-to-one temporal correspondences. We derive each transition from adjacent coordinate increments and map its type to a learnable embedding. The resulting embedding sequence is then processed by a lightweight Path CNN, which comprises two 1D convolutional layers separated by a GELU activation, to capture local geometric patterns formed by successive transitions along the path. Formally,

$$\begin{aligned} \delta_{ij,n} &= (a_{n+1} - a_n,\, b_{n+1} - b_n) \in \{(1,1),(1,0),(0,1)\}, n = 1, \ldots, |\widetilde{\mathcal{A}}_{ij}| - 1, \\ \mathbf{E}_{ij} &= \left[\mathrm{Emb}(\delta_{ij,1}), \ldots, \mathrm{Emb}(\delta_{ij,|\widetilde{\mathcal{A}}_{ij}|-1})\right], \quad \mathbf{Z}_{ij} = \mathrm{Conv}_2\left(\mathrm{GELU}\left(\mathrm{Conv}_1(\mathbf{E}_{ij})\right)\right) \end{aligned} \tag{4}$$

where $\delta_{ij,n}$ denotes the n-th transition along the warping path, and $\mathrm{Emb}(\cdot)$ is a learnable embedding function. $\mathbf{E}_{ij} \in \mathbb{R}^{(|\widetilde{\mathcal{A}}_{ij}|-1)\times d_p}$ denotes the step-embedding sequence, which is transformed by the Path CNN into $\mathbf{Z}_{ij} \in \mathbb{R}^{(|\widetilde{\mathcal{A}}_{ij}|-1)\times d_p}$.

Path Feature Aggregation. The encoded features $\mathbf{Z}_{ij}$ are arranged along the warping path rather than the temporal axis of the anchor, and multiple path transitions may correspond to the same anchor timestamp. To align these features with the encoder representations, we scatter each step-level feature back to the temporal axis of x_i according to its anchor coordinate. Features assigned to the same timestamp are averaged, yielding an anchor-aligned path representation for each reliable pair. Since each anchor may have multiple reliable neighbors, we further aggregate these pairwise representations using the reliability scores as weights, so that more reliable paths contribute more to the aggregated representation. This process is formulated as follows:

$$\widetilde{\mathbf{P}}_{ij} = \mathrm{Scatter}\left(\mathbf{Z}_{ij}, \widetilde{\mathcal{A}}_{ij}\right), \quad \mathbf{P}_i = \frac{\sum_{j \in \mathcal{N}_i^+} \rho_{ij} \widetilde{\mathbf{P}}_{ij}}{\sum_{j \in \mathcal{N}_i^+} \rho_{ij}} \tag{5}$$

where $\mathrm{Scatter}(\cdot)$ maps path features to the anchor time axis and averages those assigned to the same timestamp. $\widetilde{\mathbf{P}}_{ij}$ and $\mathbf{P}_i$, both in $\mathbb{R}^{T \times d_p}$, denote the path representation of pair (i, j) and the corresponding aggregated representation over reliable neighbors, respectively.

Gated Residual Generation. The aggregated representation $\mathbf{P}_i$ provides elastic alignment information along the anchor temporal axis. Since the relevance of such information may vary across encoder layers, timestamps, and feature dimensions, we adaptively control its contribution to the encoder. Given the intermediate representation $\widetilde{\mathbf{H}}_i^l$, we first project $\mathbf{P}_i$ into the current hidden space.

The projected path representation is then concatenated with $\widetilde{\mathbf{H}}_i^l$ to generate gating weights through a linear transformation followed by a sigmoid function. These weights modulate the path information at each temporal position and feature dimension, producing the residual of the DTW Adapter:

$$\widetilde{\mathbf{P}}_i^l = \mathbf{P}_i \mathbf{W}_p^l + \mathbf{b}_p^l, \quad \mathbf{G}_i^l = \sigma\left(\left[\widetilde{\mathbf{H}}_i^l; \widetilde{\mathbf{P}}_i^l\right] \mathbf{W}_g^l + \mathbf{b}_g^l\right), \quad \mathbf{\Delta}_i^l = \mathbf{G}_i^l \odot \widetilde{\mathbf{P}}_i^l \tag{6}$$

where $\widetilde{\mathbf{H}}_i^l, \widetilde{\mathbf{P}}_i^l \in \mathbb{R}^{T \times M}$ denote the intermediate encoder representation and the projected path representation, respectively. $\mathbf{G}_i^l \in (0,1)^{T \times M}$ denotes the gating weights, $\odot$ denotes element-wise multiplication, and $\mathbf{\Delta}_i^l \in \mathbb{R}^{T \times M}$ is the resulting residual.

Integration Strategy. To retain the original TS2Vec encoding hierarchy while incorporating DTW path information, we insert the DTW Adapter only into selected intermediate layers. The early layers extract basic temporal patterns, the intermediate layers incorporate path information through gated residuals, and the later layers further refine the resulting representations. Formally,

$$\mathbf{H}_i^l = \begin{cases} \widetilde{\mathbf{H}}_i^l + \mathbf{\Delta}_i^l, & l \in \mathcal{I}_{\text{adp}}, \\ \widetilde{\mathbf{H}}_i^l, & l \notin \mathcal{I}_{\text{adp}} \end{cases} \tag{7}$$

where $\widetilde{\mathbf{H}}_i^l$ and $\mathbf{H}_i^l$ denote the representations before and after adapter integration, respectively, and $\mathcal{I}_{\text{adp}}$ denotes the set of encoder layers equipped with the DTW Adapter. The DTW Adapter is used only during training and is bypassed at inference, where representations are obtained directly from the backbone without DTW computation or neighbor retrieval.

3.3 Elastic Contrastive Learning

Following TS2Vec (Yue et al., 2022), we retain its instance-wise and temporal contrastive objectives, denoted by $\mathcal{L}_{\text{inst}}$ and $\mathcal{L}_{\text{temp}}$, respectively. For two augmented views of each training instance, let $\mathbf{r}_{i,t}$ and $\mathbf{r}'_{i,t}$ denote the corresponding view representations at timestamp t. However, both objectives construct positive pairs primarily from same-index temporal correspondences within augmented views and therefore do not explicitly model elastic alignments across different time series. To complement them, we introduce an elastic contrastive objective that exploits reliable temporal alignments derived from DTW warping paths. To ensure that both instances in a reliable cross-sequence relation co-occur in the same batch, we adopt a relation-aware sampling strategy that jointly samples each anchor with instances from its reliable neighbor set. The detailed sampling procedure is provided in Appendix B.

Elastic Contrastive Objective. We construct cross-sequence temporal positives from the mapped warping paths of reliable neighbors. For an anchor representation $\mathbf{r}_{i,t}$, a temporal representation $\mathbf{r}'_{j,s}$ is treated as an elastic positive if x_j is a reliable neighbor of x_i and the corresponding temporal positions (t, s) are aligned by their mapped DTW path. All remaining cross-sequence temporal representations are treated as negatives. Accordingly, the positive and negative sets are defined as:

$$\mathcal{P}_{i,t}^{\text{dtw}} = \left\{(j,s) \mid j \in \mathcal{N}_i^+,\ (t,s) \in \widetilde{\mathcal{A}}_{ij}\right\}, \quad \mathcal{Q}_{i,t}^{\text{dtw}} = \left\{(j,s) \mid j \neq i,\ s \in \Omega,\ (j,s) \notin \mathcal{P}_{i,t}^{\text{dtw}}\right\}, \tag{8}$$

where Ω denotes the set of valid temporal positions in the target augmented view.

We then introduce a multi-positive contrastive objective that treats all temporal alignments from reliable DTW paths as positives:

$$\mathcal{L}_{\text{elastic}}^{(i,t)} = -\log \frac{\displaystyle\sum_{(j,s) \in \mathcal{P}_{i,t}^{\text{dtw}}} \exp\left(\mathbf{r}_{i,t} \cdot \mathbf{r}'_{j,s}\right)}{\displaystyle\sum_{(j,s) \in \mathcal{P}_{i,t}^{\text{dtw}}} \exp\left(\mathbf{r}_{i,t} \cdot \mathbf{r}'_{j,s}\right) + \sum_{(j,s) \in \mathcal{Q}_{i,t}^{\text{dtw}}} \exp\left(\mathbf{r}_{i,t} \cdot \mathbf{r}'_{j,s}\right)}. \tag{9}$$

Overall Optimization Objective. Finally, we jointly optimize the original instance-wise and temporal contrastive objectives with the proposed elastic contrastive objective:

$$\mathcal{L} = (1 - \lambda)\left(\mathcal{L}_{\text{inst}} + \mathcal{L}_{\text{temp}}\right) + \lambda \mathcal{L}_{\text{elastic}} \tag{10}$$

where $\lambda \in [0, 1]$ denotes the weight assigned to the elastic contrastive objective.

Table 1: Classification results on UCR and UEA. Best in bold, second best underlined.

Dataset	Metric	DTW4TS	TSTCC	TS2Vec	TimesURL	SoftCLT	PlanTS	TimesNet	MOMENT	UniTS	FusAD	AimTS	FreRA
125 UCR	Avg. Acc	**0.835**	0.779	<u>0.829</u>	0.788	0.799	0.762	0.698	0.684	0.414	0.712	0.721	0.665
	Avg. Rank	**2.844**	5.644	<u>3.612</u>	5.120	4.580	6.208	8.064	7.516	11.584	7.244	6.988	8.596
	Num. Top1	**45**	22	<u>35</u>	17	23	16	8	14	0	7	16	9
27 UEA	Avg. Acc	**0.715**	0.669	0.689	0.680	0.702	0.654	0.688	0.619	0.398	<u>0.705</u>	0.469	0.672
	Avg. Rank	**4.500**	6.259	6.259	6.130	<u>5.370</u>	6.519	6.222	6.815	9.852	5.389	7.667	7.019
	Num. Top1	**4**	**4**	1	2	1	2	**4**	**4**	0	<u>3</u>	<u>3</u>	2

4 EXPERIMENTS

In this section, we evaluate DTW4TS on three downstream tasks: time series classification, anomaly detection, and forecasting. Detailed experimental settings are provided in Appendix D.

Baselines. To comprehensively evaluate the effectiveness of DTW4TS across diverse tasks, we compare it with a broad range of competitive baselines. These include objective-centric representation learning methods, such as TS-TCC, TS2Vec, TimesURL, SoftCLT, and PlanTS (Eldele et al., 2021; Yue et al., 2022; Liu & Chen, 2024; Lee et al., 2024; Wang et al., 2025), as well as architecture-centric general time series analysis models, including TimesNet, MOMENT, UniTS, and FusAD (Wu et al., 2022; Goswami et al., 2024; Gao et al., 2024; Zhang et al., 2026). In addition, we include recent task-specific methods for each downstream task: AimTS, FreRA (Liu et al., 2025; Tian et al., 2025) for classification, iTransformer, DADA (Liu et al., 2024; Shentu et al., 2025) for forecasting, and MLPMixer, CATCH (Tang et al., 2024; Wu et al., 2025) for anomaly detection.

4.1 CLASSIFICATION

Datasets and setups. For time series classification, we evaluate DTW4TS on 125 univariate datasets from the UCR Archive (Dau et al., 2019) and a commonly used subset of 27 multivariate datasets from the UEA Archive (Bagnall et al., 2018). The remaining three UEA datasets are excluded because baseline methods such as MOMENT encountered out-of-memory errors on them under our computational setup. Following TS2Vec (Yue et al., 2022), we train an SVM classifier with an RBF kernel on the learned representations. We report average accuracy (Avg. Acc.), average rank (Avg. Rank), and the number of datasets achieving the highest accuracy (Num. Top1).

Results. Table 1 compares DTW4TS with competing methods on the UCR and UEA archives. DTW4TS achieves the best aggregate classification performance on both benchmarks, with average accuracies of 0.835 on UCR and 0.715 on UEA, corresponding to a 1.19% relative gain in overall average accuracy over the strongest competing baseline across all 152 datasets. It also obtains the lowest average ranks of 2.844 and 4.500, respectively, suggesting that its gains are broadly distributed across datasets rather than concentrated on a small subset. In terms of dataset-wise wins, DTW4TS ranks first on 45 UCR datasets, more than any competing method, and achieves the highest accuracy on 4 UEA datasets. Overall, these results support the effectiveness of DTW4TS for both univariate and multivariate time series classification. Full results are provided in Appendix I.1.

4.2 ANOMALY DETECTION

Datasets and setups. To evaluate the learned representations for time series anomaly detection, we conduct experiments on five real-world datasets: CalIt2 (Asuncion & Newman, 2007), GECCO (Moritz et al., 2018), MSL (Hundman et al., 2018), ECG (Yoon et al., 2020), and KPI (Ren et al., 2019). Following TS2Vec (Yue et al., 2022), we adopt its anomaly scoring and threshold selection procedures, computing anomaly scores from the discrepancy between representations of masked and unmasked inputs. We report Precision (P), Recall (R), and F1 score (F1).

Results. Table 2 presents the anomaly detection results, where DTW4TS achieves the best overall performance, with average Precision, Recall, and F1 scores of 0.823, 0.325, and 0.433, respectively. It obtains the highest F1 scores on CalIt2, GECCO, and ECG, and improves the average F1 from 0.385, achieved by the strongest competing method SoftCLT, to 0.433. Moreover, DTW4TS achieves both the highest average Recall and the highest average Precision, indicating a favorable balance between detecting anomalous patterns and avoiding false alarms. Overall, these results support the effectiveness of DTW4TS representations for anomaly detection on the evaluated datasets.

Table 2: The overall anomaly detection results on five real-world datasets. Precision (P), Recall (R), and F1 score (F1) are reported. Best in bold, second best underlined.

Dataset	Metric	DTW4TS	TSTCC	TS2Vec	TimesURL	SoftCLT	PlanTS	TimesNet	MOMENT	UniTS	FusAD	MLPMixer	CATCH
CalIt2	P	0.525	0.516	0.418	0.580	0.369	0.402	0.419	0.333	**0.586**	0.411	0.384	0.324
	R	**0.221**	0.108	0.122	0.203	0.153	0.090	0.162	0.054	0.072	0.137	0.203	0.131
	F1	**0.311**	0.177	0.186	0.300	0.214	0.147	0.234	0.093	0.127	0.205	0.263	0.184
GECCO	P	0.946	0.954	0.956	0.963	0.950	0.919	0.947	0.634	0.928	0.980	**0.990**	0.949
	R	0.636	0.352	0.604	0.436	0.573	0.373	0.534	**0.699**	0.373	0.468	0.321	0.321
	F1	**0.758**	0.513	0.738	0.595	0.714	0.529	0.683	0.665	0.532	0.632	0.484	0.479
ECG	P	**0.968**	0.698	0.966	0.939	0.959	0.895	0.857	0.734	0.759	0.850	0.838	0.928
	R	**0.164**	0.049	0.106	0.075	0.097	0.057	0.022	0.039	0.032	0.037	0.040	0.108
	F1	**0.281**	0.090	0.192	0.139	0.175	0.106	0.043	0.074	0.062	0.071	0.075	0.194
MSL	P	0.744	0.641	0.647	0.663	0.647	0.686	0.719	0.631	0.629	**0.767**	0.684	0.515
	R	0.075	0.068	0.075	0.075	0.074	**0.079**	0.067	0.061	0.075	0.058	0.030	0.045
	F1	0.137	0.122	0.135	0.135	0.135	**0.141**	0.122	0.111	0.135	0.107	0.057	0.083
KPI	P	**0.934**	0.690	0.932	0.724	0.930	0.677	0.554	0.811	0.664	0.889	0.905	0.860
	R	0.531	0.235	0.523	0.235	0.542	0.183	0.147	0.534	0.177	0.291	0.290	0.439
	F1	0.677	0.350	0.670	0.354	**0.685**	0.288	0.232	0.644	0.279	0.433	0.440	0.581
Avg	P	**0.823**	0.700	0.784	0.774	0.771	0.716	0.699	0.629	0.713	0.779	0.760	0.715
	R	**0.325**	0.162	0.286	0.205	0.288	0.156	0.186	0.277	0.146	0.198	0.177	0.209
	F1	**0.433**	0.250	0.384	0.305	0.385	0.242	0.263	0.317	0.227	0.290	0.264	0.304

4.3 FORECASTING

Datasets and setups. To assess the effectiveness of DTW4TS on time series forecasting, we conduct experiments on six widely used datasets, including four ETT datasets (ETTh1, ETTh2, ETTm1, and ETTm2) (Zhou et al., 2021), Electricity (Trindade, 2015), and Exchange (Lai et al., 2018). Following the standard evaluation protocol, we train a residual-form L_2-regularized linear regression model (Ridge) on the learned representations, which predicts future residuals relative to the latest observed value and adds them back for forecasting. We adopt Mean Squared Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics.

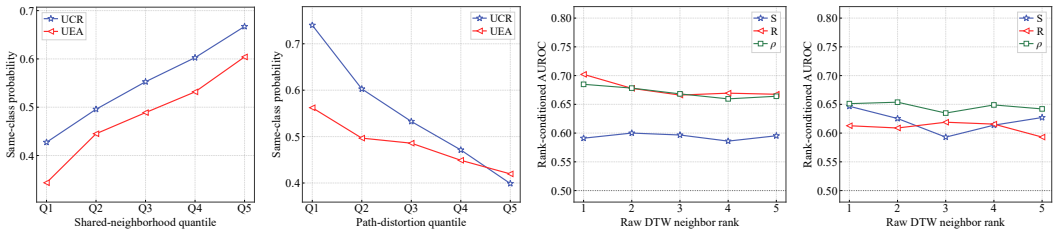
Results. Table 3 reports the forecasting results on six datasets. DTW4TS ranks among the top two in 7 of 12 dataset-metric comparisons and places first in 6 cases. It performs particularly well on ETTh2 and Exchange, delivering the best results on both MSE and MAE, while also recording the lowest MSE on ETTm2 and Electricity. Overall, DTW4TS yields the lowest average MSE of 0.195, outperforming the second-best result of 0.221, while attaining a competitive average MAE of 0.292. These results indicate that DTW4TS learns effective representations for forecasting across datasets with different temporal characteristics, with its main aggregate advantage reflected in MSE. Full results are provided in Appendix 10.

Table 3: Forecasting results averaged across five prediction lengths: $\{48, 96, 192, 336, 720\}$. Lower MSE and MAE values indicate better performance. The best results are in bold, while the second-best results are underlined.

Models		DTW4TS		TSTCC		TS2Vec		TimesURL		SoftCLT		PlanTS		TimesNet		MOMENT		UniTS		FusAD		iTransformer		DADA	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	Avg	0.088	0.227	0.248	0.419	0.124	0.274	0.145	0.300	0.129	0.278	0.132	0.283	0.070	0.202	0.068	0.200	0.073	0.207	0.076	0.218	**0.067**	**0.199**	0.068	0.201
ETTh2	Avg	**0.167**	**0.321**	0.197	0.354	0.180	0.338	0.184	0.340	0.181	0.338	0.204	0.349	0.179	0.330	0.174	0.323	0.182	0.333	0.168	0.322	0.173	0.325	0.172	0.322
ETTm1	Avg	0.075	0.198	0.152	0.313	0.077	0.208	0.070	0.198	0.075	0.203	0.126	0.268	0.053	0.168	0.046	0.156	0.047	0.158	**0.043**	**0.154**	0.048	0.161	0.048	0.159
ETTm2	Avg	**0.108**	0.248	0.182	0.340	0.121	0.269	0.124	0.269	0.125	0.268	0.137	0.270	0.113	0.244	0.110	**0.237**	0.109	0.240	0.112	0.252	0.109	0.239	0.110	0.238
Electricity	Avg	**0.449**	0.381	0.535	0.473	0.509	0.439	0.501	0.439	0.505	0.439	0.483	0.420	0.550	0.385	0.503	0.385	0.507	0.413	0.689	0.498	0.535	**0.377**	0.499	0.403
Exchange	Avg	**0.283**	**0.377**	0.578	0.561	0.427	0.450	0.384	0.424	0.440	0.458	0.373	0.419	0.445	0.447	0.429	0.432	0.461	0.447	0.613	0.597	0.397	0.423	0.498	0.455
Overall	Avg	**0.195**	0.292	0.315	0.410	0.240	0.329	0.235	0.328	0.242	0.331	0.242	0.335	0.235	0.296	0.222	0.289	0.230	0.300	0.284	0.340	0.221	**0.287**	0.233	0.296

4.4 MODEL ANALYSIS

Reliability Analysis. We empirically examine whether the proposed reliability measures provide useful indicators of DTW relation reliability. As shown in Figure 3(a), the same-class probability generally increases with shared neighborhood consistency on both UCR and UEA, while Figure 3(b) shows the opposite trend as path deviation increases. These observations suggest that local neighborhood consistency and path deviation provide complementary information for assessing candidate DTW relations. To examine whether these signals merely reflect raw DTW proximity, we further

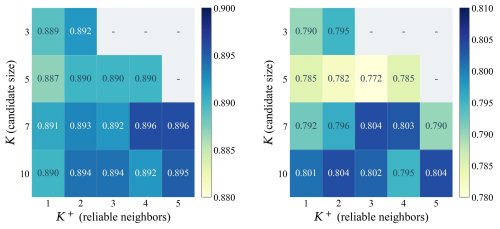


(a) Shared Neighborhood (b) Path Deviation (c) UCR Rank Analysis (d) UEA Rank Analysis

Figure 3: Reliability analysis on UCR and UEA datasets.

compute rank-conditioned AUROC for each raw DTW neighbor rank. As shown in Figures 3(c) and 3(d), S, $1 - R$, and the combined reliability score ρ remain above the random level of 0.5 on both archives, suggesting that these measures provide information beyond raw DTW neighbor rank. Additional analyses are provided in Appendix E.

Ablation Study. Table 4 evaluates the contributions of the main components of DTW4TS. We remove reliability modeling by directly using raw DTW relations, retain only S or R, remove the elastic contrastive loss or the DTW Adapter, and introduce a shuffled path control that preserves path length and model capacity while randomly permuting path transitions. Across the evaluated datasets, the full reliability model generally outperforms raw DTW relations and the variants using only S or R, while removing the elastic contrastive loss or the DTW Adapter leads to consistent performance degradation. The shuffled path control also performs worse than the original DTW paths, indicating that the ordered correspondences along warping paths provide useful structural information.



(a) Sensitivity on UCR (b) Sensitivity on UEA

Figure 4: Sensitivity analysis of K and K^+.

Variations		Cls. (Acc.↑)		Ano. (F1↑)		Fore. (MSE↓)	
		UGLA	SRSCP1	GECCO	ECG	ETTm1	Exchange
Reliability Modeling	w/o Reliability	0.946 ± 0.007	0.801 ± 0.003	0.736 ± 0.013	0.198 ± 0.002	0.080 ± 0.005	0.317 ± 0.004
	w/o R (S-only)	0.948 ± 0.006	0.758 ± 0.032	0.689 ± 0.018	0.274 ± 0.004	0.076 ± 0.003	0.296 ± 0.002
	w/o S (R-only)	0.954 ± 0.003	0.771 ± 0.014	0.736 ± 0.023	0.194 ± 0.008	0.084 ± 0.005	0.289 ± 0.003
w/o DTW Elastic Loss		0.929 ± 0.016	0.778 ± 0.015	0.730 ± 0.025	0.185 ± 0.009	0.076 ± 0.002	0.286 ± 0.001
w/o DTW Adapter		0.951 ± 0.004	0.793 ± 0.013	0.713 ± 0.010	0.197 ± 0.006	0.085 ± 0.004	0.303 ± 0.007
Shuffled Path		0.950 ± 0.002	0.783 ± 0.013	0.737 ± 0.018	0.191 ± 0.003	0.079 ± 0.005	0.292 ± 0.001
TS2Vec		0.932 ± 0.005	0.771 ± 0.006	0.738 ± 0.010	0.192 ± 0.008	0.077 ± 0.004	0.427 ± 0.008
DTW4TS (Full)		**0.958 ± 0.006**	**0.802 ± 0.009**	**0.758 ± 0.018**	**0.281 ± 0.006**	**0.075 ± 0.007**	**0.283 ± 0.004**

Table 4: Ablation study of the main components of DTW4TS. Results are reported as mean ± standard deviation over three independent runs. UGLA and SRSCP1 denote UWaveGestureLibraryAll (UCR) and SelfRegulationSCP1 (UEA), respectively.

Hyperparameter Sensitivity. To examine the robustness of DTW4TS, we sample 20 datasets from UCR and 10 datasets from UEA for hyperparameter sensitivity analysis. Figures 4(a) and 4(b) analyze the candidate neighborhood size K and the number of reliable neighbors K^+ on UCR and UEA, respectively. DTW4TS exhibits stable classification performance across a broad range of (K, K^+) settings on both archives, with $K = 7$ and $K^+ = 4$ providing consistently strong results. Additional sensitivity analyses are provided in Appendix F.

5 Conclusion

We propose DTW4TS, a reliability-aware representation learning framework that selects DTW relations using neighborhood consistency and path deviation, injects warping path information through a lightweight adapter, and exploits aligned correspondences via elastic contrastive learning. Experiments on classification, anomaly detection, and forecasting consistently demonstrate the effectiveness of the proposed framework across diverse downstream tasks. Although DTW-related computations are confined to training, pairwise DTW preprocessing still incurs additional cost. Exploring how to construct reliable DTW supervision from a small but informative subset of pairwise relations, while maintaining representation quality, remains an important direction for future work.

REFERENCES

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

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

Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pp. 359–370, 1994.

Zemin Chao, Qiaoyi Zheng, Zhixin Qi, and Hongzhi Wang. Fsmdtw: A fast index-free subsequence matching algorithm for dynamic time warping. *Proceedings of the VLDB Endowment*, 18(10): 3628–3640, 2025.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 894–903. PMLR, 2017. URL `http://proceedings.mlr.press/v70/cuturi17a.html`.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*, 2021.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37:140589–140631, 2024.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Isma Hadji, Konstantinos G Derpanis, and Allan D Jepson. Representation learning via global temporal alignment and cycle-consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11068–11077, 2021.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018.

J Jing, J Dauwels, T Rakthanmanon, E Keogh, SS Cash, and MB Westover. Rapid annotation of interictal epileptiform discharges via template matching under dynamic time warping. *Journal of neuroscience methods*, 274:179–190, 2016.

Denizhan Kara, Tomoyoshi Kimura, Jinyang Li, Bowen He, Yizhuo Chen, Yigong Hu, Hongjue Zhao, Shengzhong Liu, and Tarek F. Abdelzaher. Adats: Learning adaptive time series representations via dynamic soft contrasts. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Seunghan Lee, Taeyoung Park, and Kibok Lee. Soft contrastive learning for time series. In *International Conference on Learning Representations*, volume 2024, pp. 46815–46839, 2024.

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

Jiexi Liu and Songcan Chen. Timesurl: Self-supervised contrastive learning for universal time series representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 13918–13926, 2024.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18915–18923, 2025.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

S Moritz, F Rehbach, S Chandrasekaran, M Rebolledo, and T Bartz-Beielstein. Gecco industrial challenge 2018 dataset. *Tech. Rep.*, 2018.

Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3009–3017, 2019.

Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, October 2007. ISSN 1088-467X.

Qichao Shentu, Beibu Li, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. In *International Conference on Learning Representations*, volume 2025, pp. 81358–81381, 2025.

Pengxiang Shi, Wenwen Ye, and Zheng Qin. Self-supervised pre-training for time series classification. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021.

Qideng Tang, Chaofan Dai, Yahui Wu, and Haohao Zhou. Mlp-mixer based masked autoencoders are effective, explainable and robust for time series anomaly detection. *Proceedings of the VLDB Endowment*, 18(3):798–811, 2024.

Tian Tian, Chunyan Miao, and Hangwei Qian. Frera: a frequency-refined augmentation for contrastive learning on time series classification. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2835–2846, 2025.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

Patara Trirat, Yooju Shin, Junhyeok Kang, Youngeun Nam, Jihye Na, Minyoung Bae, Joeun Kim, Byunghyun Kim, and Jae-Gil Lee. Universal time-series representation learning: A survey. *ACM Computing Surveys*, 58(12):1–40, 2026.

Jia Wang, Xiao Wang, and Chi Zhang. Plants: Periodicity-aware latent-state representation learning for multivariate time series. *arXiv preprint arXiv:2509.05478*, 2025.

Jingyuan Wang, Chen Yang, Xiaohan Jiang, and Junjie Wu. When: A wavelet-dtw hybrid attention network for heterogeneous time series analysis. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2361–2373, 2023.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

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

Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. Catch: Channel-aware multivariate time series anomaly detection via frequency patching. In *International conference on learning representations*, volume 2025, pp. 17017–17045, 2025.

Jiandong Xie, Yue Cui, Feiteng Huang, Chao Liu, and Kai Zheng. Marina: An mlp-attention model for multivariate time-series analysis. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 2230–2239, 2022.

Maxwell Xu, Jaya Narain, Gregory Darnell, Haraldur Hallgrimsson, Hyewon Jeong, Darren Forde, Richard Fineman, Karthik Raghuram, James Rehg, and Shirley Ren. Relcon: Relative contrastive learning for a motion foundation model for wearable data. In *International Conference on Learning Representations*, volume 2025, pp. 53188–53206, 2025.

Ming Xu, Sourav Garg, Michael Milford, and Stephen Gould. Deep declarative dynamic time warping for end-to-end learning of alignment paths. In *International Conference on Learning Representations*, 2023.

Xinyu Yang, Zhenguo Zhang, and Rongyi Cui. Timeclr: A self-supervised contrastive learning framework for univariate time series representation. *Knowledge-Based Systems*, 245:108606, 2022a.

Yuncong Yang, Jiawei Ma, Shiyuan Huang, Long Chen, Xudong Lin, Guangxing Han, and Shih-Fu Chang. Tempclr: Temporal alignment representation with contrastive learning. *arXiv preprint arXiv:2212.13738*, 2022b.

Tao Yin, Shaochen Fu, Zhibin Zhang, Li Huang, Xiaohong Zhang, Yiyuan Yang, Kaixiang Yang, and Meng Yan. Scatterad: Temporal-topological scattering mechanism for time series anomaly detection. *Advances in Neural Information Processing Systems*, 38:52266–52297, 2026.

Susik Yoon, Jae-Gil Lee, and Byung Suk Lee. Ultrafast local outlier detection from a data stream with stationary region skipping. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1181–1191, 2020.

Zhongyang Yu, Zhihai Wang, Jidong Yuan, Huiting Pei, Yuxuan Fan, and Shijiang Li. Dtwformer: A dtw-based transformer for multivariate time series forecasting. *IEEE Transactions on Big Data*, 2025.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8980–8987, 2022.

Da Zhang, Bingyu Li, Zhiyuan Zhao, Feiping Nie, Junyu Gao, and Xuelong Li. Fusad: Time-frequency fusion with adaptive denoising for general time series analysis. In *42nd IEEE International Conference on Data Engineering, ICDE 2026, Montreal, QC, Canada, May 4-8, 2026*, pp. 1334–1348. IEEE, 2026.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

A Additional Related Work

Existing Time Series Representation Learning (TSRL) methods mainly follow two design directions: encoder architecture design and learning objective optimization (Trirat et al., 2026). The former improves feature extraction through specialized model architectures, while the latter shapes the representation space through carefully designed supervision signals.

Architecture-centric Representation Learning. Architecture-centric methods model complex temporal dependencies by designing specialized encoders that map raw time series into representation spaces. Early studies explored neural network-based architectures. For example, MARINA (Xie et al., 2022) employs an MLP-based residual network to model temporal correlations. Recent studies further incorporate inherent characteristics of time series into architecture design. TimesNet (Wu et al., 2022) transforms 1D sequences into 2D tensors via FFT and captures complex temporal patterns with Inception modules. TVNET (Li et al., 2025) enhances CNNs with 3D embedding and dynamic convolution to model multidimensional temporal relationships. MOMENT (Goswami et al., 2024) combines patch partitioning and random masking with the Transformer architecture for TSRL. FusAD (Zhang et al., 2026) further integrates multiscale modeling and time-frequency analysis to capture complex dynamic changes.

Objective-centric Representation Learning. Objective-centric methods optimize the representation space through learning objectives, among which self-supervised contrastive learning has become an important direction. TS-TCC (Eldele et al., 2021) constructs different augmented views and applies temporal contrastive objectives to learn contextual consistency. TS2Vec (Yue et al., 2022) further introduces hierarchical contrastive learning to learn discriminative representations at different temporal granularities. SoftCLT (Lee et al., 2024) introduces a soft contrastive learning strategy that constructs soft supervision signals from sequence distances and temporal relationships in the data space to guide instance-wise and temporal contrastive learning. TimesURL (Liu & Chen, 2024) introduces double Universums as special negatives and combines them with a reconstruction objective to capture both segment-level and instance-level information. To better capture periodicity, PlanTS (Wang et al., 2025) designs a period-aware multigranularity patch mechanism and a generalized contrastive loss to preserve instance-level and state-level similarities across different time scales.

B Relation-aware Batch Construction

The elastic contrastive objective requires both instances in a reliable DTW relation to be present in the same batch. We therefore adopt a relation-aware batch construction strategy. We first sample a set of seed anchors. For each anchor x_i, we sample up to K^+ distinct instances from its reliable neighbor set $\mathcal{N}_i^+$ and add them to the same batch. Duplicate instances are removed, and any remaining batch positions are filled with randomly sampled instances. Elastic positive pairs are constructed only for reliable pairs whose two distinct instances occur in the current batch and whose mapped warping path contains at least one valid correspondence after crop augmentation. Instances without valid elastic correspondences still contribute to the original instance-wise and temporal contrastive objectives. If no valid elastic pair exists in a batch, $\mathcal{L}_{\text{elastic}}$ is set to 0, and the training objective reduces to the original contrastive losses for that batch.

C Detailed Algorithm

D Experimental Details

D.1 Datasets

Classification. To evaluate the representation quality of DTW4TS on time series classification, we conduct experiments on two widely used real-world benchmark archives, UCR and UEA. Their characteristics are summarized as follows:

- **UCR** is a widely used benchmark archive for univariate time series classification, covering diverse application domains and temporal characteristics.

Algorithm 1 Training Procedure of DTW4TS

Require: Time series dataset $\mathcal{X} = \{x_i\}_{i=1}^{N}$; cached DTW distance matrix $\mathbf{D}$; candidate size K; reliable neighbor size K^+; reliability weight γ; elastic loss weight λ; adapter layer set $\mathcal{I}_{\text{adp}}$.
Ensure: Trained encoder f_θ.

▷ **Reliable DTW Modeling**
for $i = 1, \dots, N$ **do**
  $\mathcal{N}_i^K \leftarrow \arg\operatorname{TopK}_{j \neq i}(-D_{ij})$ ▷ Construct DTW candidate neighbors.
  for $j \in \mathcal{N}_i^K$ **do**
    $\mathcal{A}_{ij} \leftarrow \operatorname{DTWPath}(x_i, x_j)$ ▷ Compute the path for candidate pairs.
    $S_{ij} \leftarrow \frac{|\mathcal{N}_i^K \cap \mathcal{N}_j^K|}{K-1}$ ▷ Neighborhood consistency.
    $R_{ij} \leftarrow \frac{1}{|\mathcal{A}_{ij}|} \sum_{(a_n, b_n) \in \mathcal{A}_{ij}} \frac{|a_n - b_n|}{\max(T-1, 1)}$ ▷ Alignment path deviation.
    $\rho_{ij} \leftarrow \exp[-(1 - S_{ij}) - \gamma R_{ij}]$ ▷ Relation reliability.
  $\mathcal{N}_i^+ \leftarrow \arg\operatorname{TopK}^+_{j \in \mathcal{N}_i^K}(\rho_{ij})$ ▷ Select reliable DTW neighbors.
▷ **DTW-aware Representation Learning**
for each training minibatch $\mathcal{B}$ **do**
  $\widetilde{\mathcal{X}}, \widetilde{\mathcal{X}}' \leftarrow \operatorname{CropAugment}(\mathcal{B})$ ▷ Generate two augmented views.
  $\widetilde{\mathcal{A}}_{ij} \leftarrow \operatorname{MapPath}(\mathcal{A}_{ij}, \widetilde{x}_i, \widetilde{x}'_j),\ \forall j \in \mathcal{N}_i^+$ ▷ Map reliable paths to valid crop ranges.
  $\mathbf{H}^0 \leftarrow \widetilde{\mathcal{X}}, \quad \mathbf{H}'^0 \leftarrow \widetilde{\mathcal{X}}'$
  for $l = 1, \dots, L$ **do**
    $\widetilde{\mathbf{H}}^l \leftarrow \operatorname{EncoderLayer}_l(\mathbf{H}^{l-1})$
    $\widetilde{\mathbf{H}}'^l \leftarrow \operatorname{EncoderLayer}_l(\mathbf{H}'^{l-1})$
    if $l \in \mathcal{I}_{\text{adp}}$ **then**
      $\boldsymbol{\Delta}^l \leftarrow \operatorname{DTWAdapter}(\widetilde{\mathbf{H}}^l, \widetilde{\mathcal{A}}, \mathcal{N}^+, \rho)$
      $\boldsymbol{\Delta}'^l \leftarrow \operatorname{DTWAdapter}(\widetilde{\mathbf{H}}'^l, \widetilde{\mathcal{A}}, \mathcal{N}^+, \rho)$
      $\mathbf{H}^l \leftarrow \widetilde{\mathbf{H}}^l + \boldsymbol{\Delta}^l, \quad \mathbf{H}'^l \leftarrow \widetilde{\mathbf{H}}'^l + \boldsymbol{\Delta}'^l$ ▷ Inject DTW information.
    else
      $\mathbf{H}^l \leftarrow \widetilde{\mathbf{H}}^l, \quad \mathbf{H}'^l \leftarrow \widetilde{\mathbf{H}}'^l$
  $\mathbf{r} \leftarrow \mathbf{H}^L, \quad \mathbf{r}' \leftarrow \mathbf{H}'^L$ ▷ Final timestamp-level representations.
  $\mathcal{L}_{\text{inst}}, \mathcal{L}_{\text{temp}} \leftarrow \operatorname{TS2VecLoss}(\mathbf{r}, \mathbf{r}')$ ▷ Original hierarchical objectives.
  Construct elastic positive pairs $\mathcal{P}^{\text{dtw}}$ and negative pairs $\mathcal{Q}^{\text{dtw}}$.
  $\mathcal{L}_{\text{elastic}} \leftarrow \operatorname{ElasticLoss}(\mathbf{r}, \mathbf{r}', \mathcal{P}^{\text{dtw}}, \mathcal{Q}^{\text{dtw}})$ ▷ Applied symmetrically in both views.
  $\mathcal{L} \leftarrow (1 - \lambda)(\mathcal{L}_{\text{inst}} + \mathcal{L}_{\text{temp}}) + \lambda \mathcal{L}_{\text{elastic}}$ ▷ Overall objective.
  $\theta \leftarrow \operatorname{OptimizerStep}(\theta, \nabla_\theta \mathcal{L})$
return f_θ

- **UEA** is a representative benchmark archive for multivariate time series classification, containing real-world datasets from domains such as human activity recognition, biomedical analysis, and sensor monitoring.

Both UCR and UEA are standard benchmarks for time series classification and have been extensively adopted in prior representation learning studies (Yue et al., 2022; Li et al., 2025). In our experiments, we use 125 datasets from UCR and 27 datasets from UEA. The complete list of datasets used in our evaluation is provided in Appendix I.1.

Anomaly Detection. To evaluate the transferability of DTW4TS to anomaly detection, we conduct experiments on five publicly available real-world benchmark datasets, covering diverse application domains:

Algorithm 2 DTW Adapter

Require: Intermediate representation $\widetilde{\mathbf{H}}_i^l$; mapped reliable paths $\{\widetilde{\mathcal{A}}_{ij}\}_{j\in\mathcal{N}_i^+}$; reliability scores $\{\rho_{ij}\}_{j\in\mathcal{N}_i^+}$.

Ensure: Path-aware residual $\mathbf{\Delta}_i^l$.

▷ **Encode and align reliable warping paths.**

for $j \in \mathcal{N}_i^+$ **do**

$\mathbf{E}_{ij} \leftarrow \mathrm{StepEmbed}(\widetilde{\mathcal{A}}_{ij})$ ▷ Embed DTW transition types.

$\mathbf{Z}_{ij} \leftarrow \mathrm{Conv}_2\big(\mathrm{GELU}(\mathrm{Conv}_1(\mathbf{E}_{ij}))\big)$ ▷ Encode local path geometry.

$\widetilde{\mathbf{P}}_{ij} \leftarrow \mathrm{Scatter}(\mathbf{Z}_{ij}, \widetilde{\mathcal{A}}_{ij})$ ▷ Align path features to the anchor timeline.

▷ **Aggregate path information across reliable neighbors.**

$\mathbf{P}_i \leftarrow \frac{\sum_{j\in\mathcal{N}_i^+}\rho_{ij}\widetilde{\mathbf{P}}_{ij}}{\sum_{j\in\mathcal{N}_i^+}\rho_{ij}}$ ▷ Reliability-weighted aggregation.

▷ **Generate the gated residual.**

$\widetilde{\mathbf{P}}_i^l \leftarrow \mathbf{P}_i\mathbf{W}_p^l + \mathbf{b}_p^l$ ▷ Project path features to layer l.

$\mathbf{G}_i^l \leftarrow \sigma\left([\widetilde{\mathbf{H}}_i^l; \widetilde{\mathbf{P}}_i^l]\mathbf{W}_g^l + \mathbf{b}_g^l\right)$ ▷ Compute adaptive gates.

$\mathbf{\Delta}_i^l \leftarrow \mathbf{G}_i^l \odot \widetilde{\mathbf{P}}_i^l$ ▷ Produce the path-aware residual.

return $\mathbf{\Delta}_i^l$

- **CalIt2** (Asuncion & Newman, 2007) records building entry and exit counts over 15 weeks and is commonly used for detecting event-related anomalies.
- **GECCO** (Moritz et al., 2018) is a water quality monitoring dataset released for the GECCO Industrial Challenge.
- **ECG** (Yoon et al., 2020) contains electrocardiogram-derived signals, where abnormal cardiac patterns are treated as anomalies.
- **MSL** (Hundman et al., 2018) consists of spacecraft telemetry collected from the Mars Science Laboratory rover and contains labeled anomalous events.
- **KPI** (Ren et al., 2019) contains real-world key performance indicator (KPI) time series collected from large-scale Internet services.

These datasets cover heterogeneous anomaly detection scenarios and vary substantially in dimensionality, sequence length, and anomaly prevalence, providing a representative benchmark for evaluating the generalization ability of learned time series representations. More detailed dataset statistics are provided in Table 5.

Table 5: Dataset descriptions for anomaly detection. (AR: anomaly ratio)

Dataset	Domain	Dimension	Avg Train Length	Avg Val Length	Avg Test Length	Avg AR (%)
CalIt2	Visitor flow	2	2016	504	2520	4.09
GECCO	Water treatment	9	55408	13852	69261	1.25
ECG	Health	32	44883	11221	56105	16.27
MSL	Spacecraft	1	46653	11663	73730	5.88
KPI	Internet	1	40846	10212	51059	2.5

Forecasting. To evaluate the transferability of DTW4TS to time series forecasting, we conduct experiments on six widely used real-world benchmark datasets, covering electricity consumption, exchange rates, and electricity transformer monitoring:

- **ETT** includes four Electricity Transformer Temperature datasets, namely ETTh1, ETTh2, ETTm1, and ETTm2. ETTh1 and ETTh2 are sampled hourly, while ETTm1 and ETTm2 are recorded at 15 minute intervals.

- **Electricity** contains hourly electricity consumption records from 321 clients collected between 2012 and 2014.
- **Exchange** records daily exchange rates of eight currencies from 1990 to 2016.

These datasets are widely used benchmarks for time series forecasting and cover diverse temporal characteristics and sampling frequencies. All datasets are normalized using the mean and standard deviation computed from the training set. Samples are then extracted from each continuous time series using a sliding-window strategy. Detailed dataset statistics are provided in Table 6.

Table 6: Dataset descriptions for forecasting.

Dataset	Exchange	Electricity	ETTh1	ETTh2	ETTm1	ETTm2
Dataset Size	7207	26304	17420	17420	69680	69680
Variable Number	8	321	7	7	7	7
Sampling Frequency	1 day	1 hour	1 hour	1 hour	15 min	15 min

D.2 Implementation Details

During representation learning, class labels and downstream task information are assumed to be unavailable, making hyperparameter selection for unsupervised models inherently challenging. In supervised learning, techniques such as early stopping can rely on validation performance. Without labels, however, it is difficult to determine which training stage yields representations that generalize best to downstream tasks. Therefore, we adopt a fixed set of hyperparameters for DTW4TS across all downstream tasks without task-specific tuning.

In our experiments, the backbone encoder follows the TS2Vec dilated convolutional architecture with a depth of 10. The hidden-state dimension and output representation dimension are set to 64 and 320, respectively. For each anchor instance, we first retrieve the $K = 7$ nearest candidates according to DTW distance and then select the top $K^+ = 4$ candidates based on the proposed reliability score. DTW distances are computed using `tslearn.metrics.dtw`, and warping paths are obtained using `tslearn.metrics.dtw_path`. The path deviation coefficient γ is set to 10. The step embedding dimension d_p is set to 16, and the hidden dimension of the Path CNN is set to 32. The adapter layer set is fixed to $\mathcal{I}_{\text{adp}} = \{5\}$, and the dropout rate is set to 0.1. The elastic loss weight λ is fixed to 0.2 across all datasets and downstream tasks.

DTW candidate neighborhoods, reliability scores, and warping paths are constructed exclusively from the training split. Since DTW-based relations are used only during representation learning, the corresponding distances and paths are precomputed and cached for repeated use throughout training. During downstream evaluation, the DTW Adapter is disabled, and each instance is independently encoded by the trained backbone without DTW computation or neighbor retrieval. The model is trained using the AdamW optimizer with an initial learning rate of 10^{-3} and a fixed batch size of 8. Following the TS2Vec training protocol, the number of optimization iterations is set to 200 for datasets with fewer than 100,000 samples and to 600 otherwise. Final results are averaged over three independent runs. All DTW pairwise distance precomputation is performed on an Intel Xeon Silver 4210R CPU @ 2.40 GHz using 20 CPU threads. All experiments are implemented in Python 3.8 with PyTorch and conducted on an NVIDIA RTX A6000 GPU with 48 GB of memory.

D.3 Task-Specific Implementation Details

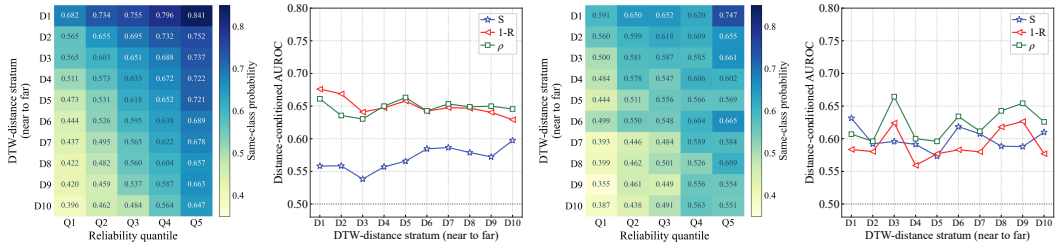
For forecasting and anomaly detection, each dataset is typically provided as a single long continuous time series rather than a collection of independent instances. To enable DTW-based relation modeling, we first segment the training sequence into causal sliding windows, thereby constructing a set of subsequence instances for pretraining. Following the downstream protocol of TS2Vec (Yue et al., 2022), the window length is set to 201, including 200 historical timestamps and the current timestamp. Since adjacent sliding windows are highly overlapping, using all windows would introduce substantial redundancy. We therefore uniformly sample 512 windows from the training split, which provides broad temporal coverage of the original sequence while keeping the training set compact.

The sampled windows form the complete pretraining instance pool and are used for all representation learning components, including the original instance-wise and temporal contrastive losses, the DTW elastic loss, and the DTW Adapter.

E Additional Reliability Analysis

We examine whether the proposed reliability modeling provides information beyond raw DTW distance by explicitly controlling for DTW proximity. Specifically, we first partition candidate DTW pairs into ten distance strata, from nearest ($D1$) to farthest ($D10$), according to their raw DTW distances. Within each distance stratum, the pairs are further divided into five quantiles ($Q1$–$Q5$) according to the combined reliability score ρ, and we measure the same-class probability in each group. As shown in Figures 5(a) and 5(c), the same-class probability generally increases from low- to high-reliability quantiles within most DTW-distance strata on both UCR and UEA. Since raw DTW distance is approximately controlled within each stratum, this trend suggests that ρ provides empirical information about semantic consistency beyond raw DTW proximity.

We further evaluate the discriminative ability of the individual reliability cues S and $1 - R$, together with the combined score ρ, within each DTW-distance stratum. Specifically, we define a post-hoc binary label indicating whether a candidate pair belongs to the same class and compute the AUROC of each reliability measure separately in every distance stratum. As shown in Figures 5(b) and 5(d), S, $1 - R$, and ρ remain above the random level of 0.5 across most distance strata, while ρ generally achieves stronger discrimination than either individual cue. These results provide further evidence that neighborhood consistency and path deviation contribute complementary information beyond raw DTW distance, and that their combination provides a stronger reliability score for ranking candidate DTW relations.



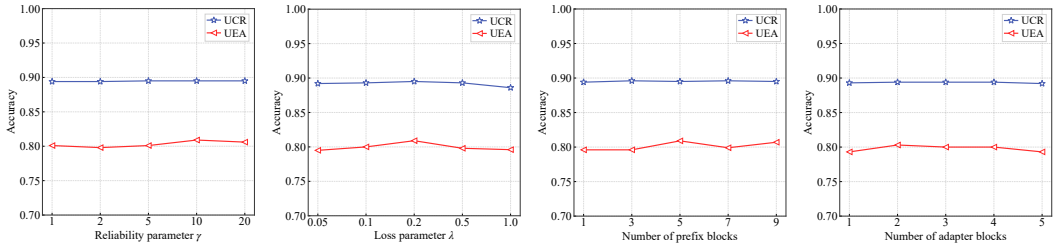
(a) UCR Reliability (b) UCR Distance Analysis (c) UEA Reliability (d) UEA Distance Analysis

Figure 5: Additional reliability analysis on UCR and UEA datasets.

F Additional Hyperparameter Sensitivity Analysis

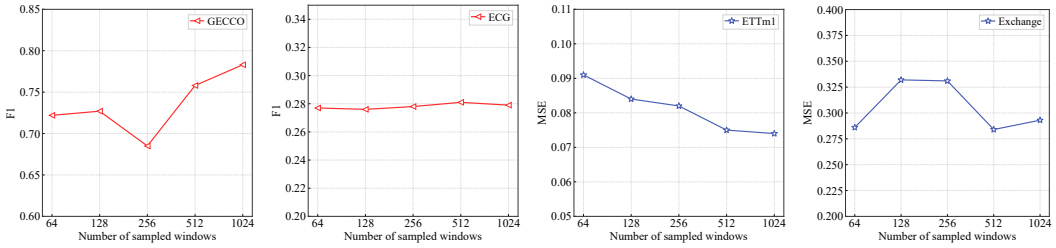
We analyze the sensitivity of DTW4TS to several key hyperparameters, including the reliability weight γ, the elastic loss weight λ, the number of prefix blocks, and the number of adapter blocks. As shown in Figure 6(a), DTW4TS exhibits stable classification performance across a broad range of γ, indicating that the reliability modeling is not overly sensitive to the relative contribution of path deviation. Similar robustness is observed for λ in Figure 6(b). Moderate values generally provide competitive performance on both UCR and UEA, while an excessively large λ slightly degrades performance, suggesting that the elastic objective is most effective when used as a complement to the original TS2Vec objectives.

We further investigate the architectural sensitivity of DTW4TS by varying the numbers of prefix blocks and adapter blocks. As shown in Figures 6(c) and 6(d), performance remains relatively stable across different configurations on both UCR and UEA. Increasing the number of blocks does not consistently lead to further improvements, indicating that DTW4TS does not rely on a deep or heavily parameterized auxiliary architecture to exploit DTW information. Overall, the consistently small performance variations across these settings demonstrate that DTW4TS is robust to the choice of its main hyperparameters.



(a) Reliability parameter γ (b) Loss parameter λ (c) Prefix blocks (d) Adapter blocks

Figure 6: Additional sensitivity studies of the main hyperparameters in DTW4TS.



(a) GECCO (b) ECG (c) ETTm1 (d) Exchange

Figure 7: Sensitivity to the number of sampled training windows on anomaly detection and forecasting datasets.

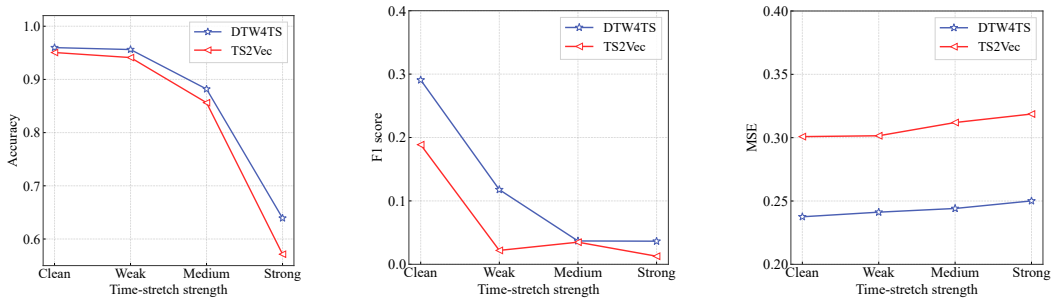
Sensitivity to the Number of Sampled Windows. For anomaly detection and forecasting, we further study the sensitivity to the number of uniformly sampled training windows, using $N_{\text{pool}} \in \{64, 128, 256, 512, 1024\}$. As shown in Figure 7, performance generally improves as the pool size increases to 512, while further increasing it to 1024 yields limited or inconsistent gains across datasets. Specifically, Figures 7(a) and 7(b) show that 512 windows provide strong performance for anomaly detection, achieving the best result on ECG and competitive performance on GECCO. For forecasting, Figures 7(c) and 7(d) show that using 512 windows achieves the best result on Exchange and remains close to the best result on ETTm1. These results support the use of 512 windows as a practical default for continuous sequence tasks, balancing representation quality and the computational cost of pairwise DTW construction.

G Robustness to Controlled Temporal Warping

We further evaluate the behavior of DTW4TS under semantics-preserving temporal misalignment by applying controlled local temporal warping to the test sequences of representative classification, anomaly detection, and forecasting datasets. Specifically, a contiguous segment is randomly selected and temporally stretched or compressed through linear resampling, after which the entire sequence is resampled back to its original length. The warping strength is controlled by both the affected segment length and the rescaling factor. As shown in Figure 8, DTW4TS generally preserves performance better than TS2Vec as the warping strength increases, with higher classification accuracy and anomaly detection F1, and lower forecasting MSE. Figure 9 illustrates representative transformations under different warping strengths. These results suggest that the elastic alignment design improves tolerance to controlled temporal distortions beyond the clean data setting.

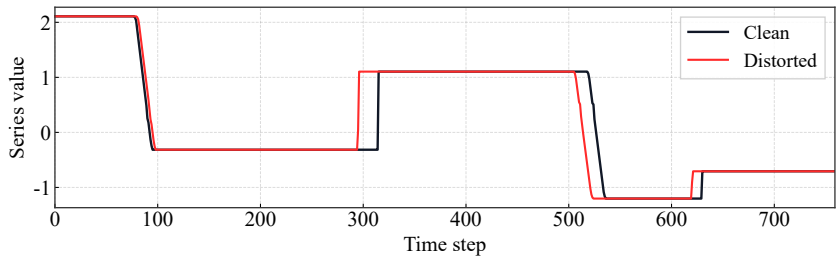
H Computational Cost Analysis

Table 7 reports the computational cost of DTW4TS on six representative datasets. For classification, DTW relations are constructed directly among training instances, while for anomaly detection and forecasting, at most 512 sliding windows are uniformly sampled from the training split for DTW relation construction. Here, N_{DTW} and T_{DTW} denote the number of instances and window length

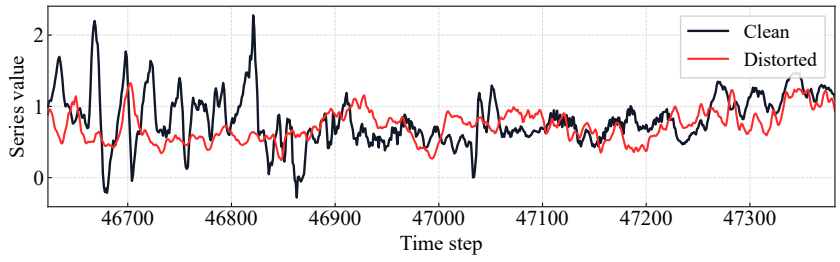


(a) Classification on UGLA (b) Anomaly detection on ECG (c) Forecasting on Exchange

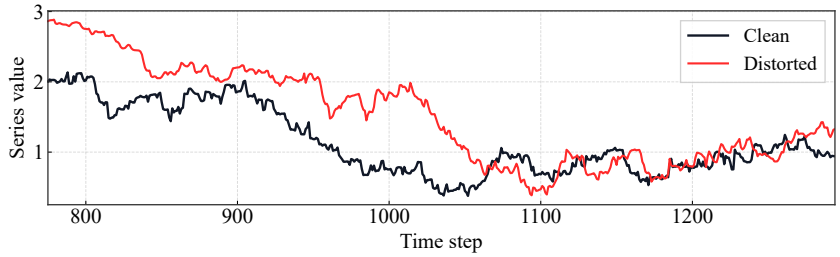
Figure 8: Robustness to controlled temporal warping across classification, anomaly detection, and forecasting.



(a) Weak temporal warping on UGLA



(b) Moderate temporal warping on ECG



(c) Strong temporal warping on Exchange

Figure 9: Examples of semantics-preserving temporal warping with increasing strength.

used for DTW computation, respectively. Pairwise DTW distances and candidate warping paths are constructed once before representation learning, while the remaining DTW-related operations are included in the reported training time. For datasets with $N_{\mathrm{DTW}} = 512$, DTW preprocessing requires only several to tens of seconds and the cached preprocessing results occupy about 1 MB, while the larger UGLA dataset incurs a higher preprocessing cost due to its larger number of pairwise comparisons. At inference, the DTW-related components are removed and representations are produced directly by the backbone without additional DTW computation.

Table 7: Computational cost of DTW4TS on six representative datasets. Runtime is reported as mean $\pm$ standard deviation. $\mathbf{N}_{\mathrm{DTW}}$ represents the number of training windows involved in the DTW preprocessing, and $\mathbf{T}_{\mathrm{DTW}}$ is the window length. DTW Precomp. denotes pairwise DTW distance computation, while Reliability denotes reliability score ρ and candidate warping path construction. Cache denotes the disk storage required for cached preprocessing results.

Task	**Dataset**	$\mathbf{N}_{\mathrm{DTW}}$	$\mathbf{T}_{\mathrm{DTW}}$	**DTW Precomp. (s)**	**Reliability (s)**	**Training (s)**	**Inference (s)**	**Cache (MB)**
Classification	UGLA	896	945	889.70 ± 21.84	95.77 ± 17.84	302.93 ± 9.98	261.67 ± 41.01	3.11
Classification	SRSCP1	268	896	47.47 ± 12.03	27.46 ± 1.42	95.43 ± 5.36	31.48 ± 4.42	0.29
Anomaly Detection	GECCO	512	201	8.52 ± 0.65	5.35 ± 0.12	73.93 ± 10.97	5.87 ± 0.98	1.03
Anomaly Detection	ECG	512	201	20.30 ± 1.08	11.27 ± 1.19	79.29 ± 1.40	12.72 ± 1.39	1.03
Forecasting	ETTm1	512	201	6.64 ± 1.50	6.90 ± 1.63	69.77 ± 3.21	4.97 ± 0.08	1.03
Forecasting	Exchange	512	201	5.67 ± 0.28	4.45 ± 0.30	88.72 ± 4.24	5.85 ± 1.20	1.03

I Full Experimental Results

I.1 Classification

Table 8: Classification accuracy on UCR datasets. The representation dimensions of these methods are all set to 320 for fair comparison.

Dataset	DTW4TS	TSTCC	TS2Vec	TimesURL	SoftCLT	PlanTS	TimesNet	MOMENT	UniTS	FusAD	AimTS	FreRA
ACSF1	0.850	0.670	**0.900**	0.820	0.690	0.660	0.400	0.450	0.200	0.730	0.430	0.680
Adiac	0.788	0.775	0.765	**0.790**	0.755	0.734	0.440	0.130	0.049	0.225	0.248	0.356
AllGestureWiimoteX	0.764	0.597	**0.776**	0.726	0.661	0.689	0.506	0.601	0.209	0.529	0.679	0.487
AllGestureWiimoteY	0.737	0.620	**0.791**	0.701	0.713	0.699	0.503	0.567	0.264	0.640	0.656	0.523
AllGestureWiimoteZ	**0.749**	0.557	0.746	0.683	0.644	0.656	0.456	0.523	0.213	0.594	0.571	0.544
ArrowHead	0.811	0.817	0.817	**0.863**	0.783	0.760	0.794	0.703	0.303	0.423	0.766	0.389
Beef	0.800	**0.900**	0.767	0.667	0.767	0.700	0.667	0.333	0.200	0.533	0.433	0.367
BirdChicken	0.800	**0.850**	0.800	0.750	0.700	0.650	0.800	0.700	0.550	0.800	0.800	**0.850**
BME	0.980	0.953	0.993	**1.000**	0.993	0.967	0.993	0.493	0.480	0.873	0.700	0.640
Car	0.850	**0.867**	0.700	0.817	0.717	0.700	0.733	0.533	0.217	0.583	0.500	0.500
CBF	**1.000**	0.932	**1.000**	0.991	0.990	0.984	0.944	0.962	0.331	0.978	0.973	0.944
Chinatown	**0.985**	0.983	0.965	0.959	0.983	0.977	0.980	0.726	0.271	0.936	0.977	0.904
ChlorineConcentration	0.811	**0.882**	0.836	0.769	0.796	0.713	0.607	0.550	0.533	0.590	0.576	0.561
CinCECGTorso	0.786	**0.933**	0.809	0.724	0.899	0.665	0.794	0.516	0.248	0.367	0.741	0.633
Coffee	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.964	0.536	0.536	0.679	0.643	**1.000**
Computers	0.660	0.628	0.660	0.624	0.640	0.692	0.568	**0.700**	0.648	0.664	**0.700**	0.688
CricketX	**0.810**	0.690	0.792	0.708	0.777	0.672	0.503	0.756	0.126	0.733	0.669	0.559
CricketY	**0.764**	0.669	0.754	0.713	0.741	0.600	0.482	0.705	0.149	0.677	0.708	0.487
CricketZ	**0.815**	0.697	0.800	0.710	0.772	0.662	0.474	0.726	0.118	0.731	0.708	0.608
Crop	0.759	0.757	0.756	0.745	**0.765**	0.737	0.753	0.737	0.593	0.751	0.593	0.709
DiatomSizeReduction	0.984	0.967	0.984	0.984	0.984	**0.990**	0.899	0.301	0.301	0.523	0.882	0.317
DistalPhalanxOutlineAgeGroup	0.734	0.734	0.727	**0.748**	0.734	0.734	0.741	0.727	0.468	**0.748**	0.662	0.712
DistalPhalanxOutlineCorrect	**0.790**	0.765	0.750	0.754	0.754	0.772	0.732	0.743	0.583	0.768	0.768	0.696
DistalPhalanxTW	0.691	0.669	**0.698**	0.662	0.684	0.655	0.684	**0.698**	0.547	0.684	0.597	0.676
DodgerLoopDay	0.538	**0.588**	0.525	0.513	0.500	0.500	0.238	0.363	0.150	0.450	0.363	0.450
DodgerLoopGame	0.833	**0.877**	0.841	0.862	0.804	**0.877**	0.514	0.478	0.544	0.551	0.652	0.696
DodgerLoopWeekend	0.957	**0.978**	0.957	0.964	0.935	0.935	0.754	0.957	0.797	0.710	**0.978**	0.913
Earthquakes	0.755	0.748	0.748	0.748	0.748	0.748	0.748	0.748	0.748	0.705	**0.763**	0.748
ECG200	**0.930**	0.890	0.920	0.870	0.850	0.910	0.880	0.810	0.640	0.880	0.880	0.790
ECG5000	0.940	0.941	0.935	0.940	0.940	**0.942**	0.928	0.927	0.899	0.915	0.935	0.928
ECGFiveDays	**1.000**	0.974	**1.000**	0.978	**1.000**	0.984	0.778	0.777	0.497	0.643	0.912	0.560
ElectricDevices	0.692	0.677	0.720	0.698	0.689	0.712	0.644	**0.766**	0.617	0.684	0.606	0.653
EOGHorizontalSignal	**0.564**	0.467	0.533	0.541	0.508	0.478	0.431	**0.564**	0.180	0.472	0.525	0.423
EthanolLevel	0.504	**0.676**	0.480	0.570	0.602	0.374	0.552	0.248	0.248	0.252	0.252	0.240
FaceAll	0.794	0.779	0.769	0.811	0.769	0.851	0.688	**0.891**	0.337	0.694	0.773	0.832
FaceFour	0.921	0.852	**0.932**	0.671	0.727	0.671	0.818	0.455	0.159	0.852	0.807	0.489
FacesUCR	**0.935**	0.871	0.925	0.870	0.884	0.804	0.822	0.628	0.175	0.761	0.830	0.572
FiftyWords	0.820	0.745	0.782	0.739	**0.822**	0.706	0.578	0.730	0.145	0.681	0.242	0.389
Fish	**0.949**	0.840	0.914	0.909	0.851	0.846	0.691	0.611	0.126	0.623	0.840	0.537
FordA	0.933	0.936	0.939	0.923	0.921	0.916	0.926	**0.950**	0.914	0.931	0.923	0.931
FordB	0.810	0.779	0.799	0.762	0.773	0.773	0.769	**0.825**	0.764	0.788	0.815	0.774
FreezerRegularTrain	0.989	0.989	0.984	**0.993**	0.991	0.986	0.951	0.787	0.767	0.944	0.986	0.767
FreezerSmallTrain	0.945	0.701	0.873	0.972	0.978	**0.989**	0.689	0.669	0.496	0.628	0.713	0.749
Fungi	0.952	0.839	**0.957**	0.710	0.952	0.511	0.941	0.097	0.081	0.699	0.731	0.548
GestureMidAirD1	**0.677**	0.631	0.615	0.515	**0.677**	0.492	0.600	0.523	0.108	0.615	0.492	0.523
GestureMidAirD2	**0.577**	0.477	0.485	0.415	0.569	0.377	0.562	0.485	0.062	0.477	0.508	0.500
GestureMidAirD3	0.354	0.377	0.308	0.246	**0.415**	0.285	0.385	0.262	0.085	0.285	0.238	0.192
GesturePebbleZ1	0.814	0.797	0.907	0.500	0.651	0.419	0.837	**0.948**	0.157	0.785	0.855	0.611
GesturePebbleZ2	0.899	0.690	0.873	0.500	0.570	0.361	0.696	**0.924**	0.215	0.842	0.823	0.557
GunPoint	0.987	0.973	0.980	0.973	0.980	**1.000**	0.900	0.860	0.493	0.773	0.980	0.720
GunPointAgeSpan	**1.000**	0.946	0.987	0.962	0.997	0.994	0.934	0.921	0.655	0.858	0.972	0.801
GunPointMaleVersusFemale	**1.000**	0.997	**1.000**	**1.000**	**1.000**	0.997	0.997	0.981	0.769	0.991	0.991	0.959
GunPointOldVersusYoung	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.921	0.632	**1.000**	**1.000**	**1.000**
Ham	0.762	0.676	0.705	0.752	0.695	0.610	**0.781**	0.743	0.514	0.733	0.724	0.638
HandOutlines	**0.930**	0.911	0.916	0.922	0.919	**0.930**	0.873	0.905	0.889	0.889	0.900	0.678
Haptics	**0.546**	0.461	0.520	0.497	0.484	0.464	0.370	0.360	0.208	0.198	0.438	0.383
Herring	0.641	0.594	**0.672**	0.625	**0.672**	0.625	0.500	0.594	0.594	0.594	0.594	0.594
HouseTwenty	0.941	0.782	0.916	0.798	0.916	0.832	0.832	**0.958**	0.580	0.950	**0.958**	0.941
InlineSkate	0.380	0.364	0.409	**0.416**	0.384	0.385	0.262	0.191	0.155	0.187	0.275	0.253
InsectEPGRegularTrain	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.811	0.474	**1.000**	**1.000**	**1.000**
InsectEPGSmallTrain	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.619	0.474	**1.000**	**1.000**	**1.000**

Dataset	DTW4TS	TSTCC	TS2Vec	TimesURL	SoftCLT	PlanTS	TimesNet	MOMENT	UniTS	FusAD	AimTS	FreRA
InsectWingbeatSound	**0.631**	0.621	0.626	0.585	0.613	0.557	0.579	0.581	0.130	0.517	0.557	0.428
ItalyPowerDemand	0.962	**0.968**	0.924	0.958	0.959	0.941	0.950	0.963	0.502	0.951	0.942	0.948
LargeKitchenAppliances	0.853	0.619	0.861	0.835	0.776	0.784	0.456	**0.891**	0.493	0.853	0.872	0.747
Lightning2	0.853	0.689	0.902	0.803	**0.918**	0.771	0.672	0.672	0.541	0.787	0.836	0.607
Lightning7	**0.849**	0.671	0.836	0.753	0.767	0.767	0.630	0.507	0.260	0.589	0.740	0.452
Mallat	0.938	0.950	0.921	**0.958**	0.837	0.850	0.818	0.374	0.123	0.576	0.801	0.730
Meat	**0.950**	0.900	0.867	0.917	0.883	0.800	0.600	0.333	0.333	0.567	0.350	0.500
MedicalImages	0.796	0.772	0.790	0.783	**0.797**	0.771	0.701	0.720	0.515	0.676	0.647	0.604
MelbournePedestrian	**0.962**	0.950	0.959	0.940	0.927	0.945	0.949	0.877	0.668	0.927	0.910	0.896
MiddlePhalanxOutlineAgeGroup	**0.649**	0.630	0.636	**0.649**	0.630	**0.649**	0.597	0.617	0.188	0.610	0.468	0.578
MiddlePhalanxOutlineCorrect	**0.849**	0.787	0.825	0.835	**0.849**	0.818	0.790	0.570	0.570	0.732	0.718	0.595
MiddlePhalanxTW	0.597	0.584	0.584	0.597	0.578	0.578	0.558	0.552	0.273	0.591	0.545	**0.604**
MixedShapesRegularTrain	0.921	0.911	0.918	0.930	0.920	0.902	0.803	0.918	0.658	0.840	**0.933**	0.883
MixedShapesSmallTrain	0.875	0.858	0.856	**0.900**	0.868	0.846	0.721	0.760	0.418	0.693	0.792	0.807
MoteStrain	0.872	0.865	0.863	0.832	0.838	0.814	**0.874**	0.871	0.689	0.869	0.696	0.787
NonInvasiveFetalECGThorax1	**0.938**	0.934	0.927	0.935	**0.938**	0.907	0.786	0.830	0.196	0.818	0.845	0.726
NonInvasiveFetalECGThorax2	0.938	**0.945**	0.942	0.940	0.937	0.931	0.843	0.871	0.155	0.880	0.784	0.791
OliveOil	**0.900**	0.800	0.867	0.800	0.833	0.500	0.300	0.400	0.400	0.400	0.400	0.400
OSULeaf	0.835	0.587	**0.851**	0.760	0.731	0.744	0.554	0.773	0.182	0.752	0.818	0.806
PhalangesOutlinesCorrect	**0.821**	0.808	0.803	0.788	0.761	0.753	0.744	0.613	0.613	0.809	0.808	0.717
Phoneme	0.298	0.205	**0.309**	0.238	0.257	0.193	0.111	0.258	0.116	0.303	0.160	0.266
PickupGestureWiimoteZ	**0.860**	0.780	0.840	0.680	0.800	0.580	0.680	0.500	0.280	0.600	0.600	0.620
PigAirwayPressure	0.548	0.115	**0.630**	0.269	0.255	0.380	0.043	0.058	0.029	0.077	0.096	0.101
PigArtPressure	0.938	0.226	**0.962**	0.649	0.615	0.688	0.096	0.255	0.019	0.346	0.346	0.620
PLAID	0.546	0.499	0.549	0.499	0.540	0.512	0.417	**0.808**	0.253	0.315	0.332	0.335
Plane	**1.000**	0.990	**1.000**	**1.000**	**1.000**	**1.000**	0.952	0.981	0.095	0.990	0.981	**1.000**
PowerCons	0.972	**1.000**	0.961	0.922	0.972	0.906	**1.000**	0.983	0.617	0.983	0.867	0.911
ProximalPhalanxOutlineAgeGroup	0.849	0.849	0.834	0.854	0.849	0.844	0.854	0.849	0.488	**0.868**	0.859	**0.868**
ProximalPhalanxOutlineCorrect	0.897	**0.900**	0.894	0.869	0.880	0.876	0.866	0.684	0.684	0.770	0.897	0.794
ProximalPhalanxTW	0.805	0.776	0.805	0.781	**0.829**	0.824	0.785	0.810	0.351	0.742	0.785	**0.829**
RefrigerationDevices	**0.627**	0.507	0.592	0.549	0.571	0.563	0.421	0.547	0.384	0.571	0.464	0.443
Rock	0.800	**0.840**	0.740	0.720	0.580	0.700	0.460	0.720	0.420	0.280	0.520	0.360
ScreenType	0.397	0.411	0.408	0.419	0.419	**0.504**	0.371	0.419	0.419	0.477	0.488	0.432
SemgHandGenderCh2	0.953	**0.963**	0.950	0.880	0.902	0.832	0.868	0.837	0.675	0.908	0.903	0.803
SemgHandMovementCh2	0.851	**0.858**	**0.858**	0.704	0.742	0.576	0.464	0.511	0.278	0.636	0.631	0.336
SemgHandSubjectCh2	0.938	0.947	**0.949**	0.820	0.860	0.709	0.769	0.780	0.400	0.847	0.756	0.518
ShakeGestureWiimoteZ	0.900	0.880	**0.940**	0.860	0.880	0.860	0.780	0.780	0.220	0.920	0.780	0.820
ShapeletSim	0.983	0.494	**1.000**	0.689	0.978	0.617	0.522	0.989	0.511	0.861	**1.000**	0.817
ShapesAll	0.893	0.783	**0.902**	0.845	0.847	0.798	0.617	0.762	0.077	0.643	0.407	0.628
SmallKitchenAppliances	0.755	0.645	0.728	0.760	0.755	0.709	0.523	0.696	0.552	0.699	**0.800**	0.685
SmoothSubspace	0.980	0.907	0.980	0.920	0.980	0.973	**0.987**	0.960	0.633	0.887	0.973	0.953
SonyAIBORobotSurface1	0.879	0.726	**0.910**	0.884	0.885	0.862	0.777	0.429	0.429	0.787	0.662	0.804
SonyAIBORobotSurface2	**0.891**	0.845	0.869	0.888	0.881	0.814	0.838	0.800	0.617	0.784	0.820	0.833
StarLightCurves	**0.972**	0.945	0.968	0.966	0.963	0.967	0.888	0.955	0.856	0.946	**0.972**	0.968
Strawberry	0.957	0.954	**0.968**	0.957	0.965	0.965	0.924	0.841	0.643	0.946	0.903	0.832
SwedishLeaf	**0.950**	0.936	0.941	0.926	0.930	0.917	0.810	0.886	0.104	0.910	0.834	0.904
Symbols	0.976	0.886	**0.977**	0.939	0.942	0.930	0.784	0.370	0.174	0.862	0.938	0.609
SyntheticControl	**1.000**	**1.000**	**1.000**	0.993	0.997	**1.000**	0.993	0.983	0.350	0.977	**1.000**	0.937
ToeSegmentation1	0.925	0.636	0.921	0.890	0.934	0.851	0.645	0.947	0.526	0.952	**0.969**	0.904
ToeSegmentation2	**0.915**	0.769	0.900	0.831	0.815	0.815	0.662	0.900	0.815	0.854	0.854	0.769
Trace	**1.000**	0.970	**1.000**	**1.000**	**1.000**	**1.000**	0.920	0.720	0.190	**1.000**	**1.000**	0.720
TwoLeadECG	0.986	0.941	0.987	0.980	0.972	0.992	0.748	0.722	0.500	0.859	**0.998**	0.707
TwoPatterns	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.983	**1.000**	0.629	**1.000**	**1.000**	0.930
UMD	**1.000**	0.958	**1.000**	0.993	0.993	0.993	**1.000**	0.910	0.500	0.993	0.910	0.715
UWaveGestureLibraryAll	0.958	0.961	0.932	0.929	**0.965**	0.910	0.917	0.961	0.692	0.936	0.902	0.736
UWaveGestureLibraryX	0.805	0.804	0.795	0.796	**0.819**	0.780	0.652	0.817	0.614	0.788	0.758	0.632
UWaveGestureLibraryY	0.726	0.705	0.724	0.724	**0.733**	0.709	0.574	0.713	0.537	0.667	0.661	0.596
UWaveGestureLibraryZ	0.762	0.733	**0.770**	0.742	0.756	0.734	0.578	0.740	0.542	0.714	0.739	0.638
Wafer	**0.998**	0.996	**0.998**	0.995	**0.998**	0.994	0.997	0.997	0.892	0.997	0.995	0.967
Wine	0.870	0.778	**0.926**	0.500	0.500	0.611	0.500	0.500	0.500	0.500	0.500	0.500
WordSynonyms	**0.698**	0.644	0.685	0.643	0.668	0.616	0.502	0.516	0.219	0.561	0.546	0.354
Worms	**0.740**	0.571	0.701	0.701	0.623	0.675	0.455	0.714	0.429	0.688	0.610	0.688
WormsTwoClass	0.792	0.649	0.792	0.805	0.753	0.636	0.545	**0.844**	0.571	0.779	0.455	0.831
Yoga	0.858	0.819	**0.879**	0.857	0.855	0.830	0.743	0.605	0.536	0.538	0.748	0.578
On 125 datasets:												
AVG	**0.835**	0.779	0.829	0.788	0.799	0.762	0.698	0.684	0.414	0.712	0.721	0.665
Rank	**2.844**	5.644	3.612	5.120	4.580	6.208	8.064	7.516	11.584	7.244	6.988	8.596

Table 9: Classification accuracy on 27 UEA datasets.

Dataset	DTW4TS	TSTCC	TS2Vec	TimesURL	SoftCLT	PlanTS	TimesNet	MOMENT	UniTS	FusAD	AimTS	FreRA
ArticularyWordRecognition	0.977	0.833	0.970	0.980	**0.993**	0.957	0.977	0.311	0.113	0.930	0.257	0.930
AtrialFibrillation	0.400	0.267	0.133	0.267	0.200	0.133	0.400	0.333	0.333	0.267	**0.800**	0.267
BasicMotions	0.975	0.333	0.975	0.975	0.975	0.975	0.975	0.999	0.225	**1.000**	0.368	**1.000**
CharacterTrajectories	0.987	0.903	0.975	0.990	0.983	0.983	0.971	0.550	0.714	**0.992**	0.317	0.982
Cricket	**0.986**	0.541	0.958	0.972	0.972	**0.986**	0.847	0.857	0.125	0.972	0.859	0.944
EigenWorms	0.420	0.779	0.420	0.420	0.420	0.092	0.473	0.732	0.420	**0.832**	0.566	0.672
Epilepsy	0.971	0.384	0.964	0.978	0.957	0.964	0.862	**1.000**	0.362	0.949	0.848	0.928
ERing	0.904	0.904	0.867	0.874	0.911	0.915	**0.930**	0.726	0.170	0.763	0.343	0.759
EthanolConcentration	0.285	**0.987**	0.278	0.251	0.304	0.266	0.255	0.304	0.247	0.228	0.350	0.240
FaceDetection	0.518	0.552	0.510	0.525	0.504	0.532	**0.665**	0.652	0.663	0.627	0.573	0.539
FingerMovements	0.580	0.980	0.490	0.490	0.560	0.440	0.530	**0.986**	0.490	0.550	0.065	0.500
HandMovementDirection	0.392	0.491	0.324	0.230	0.365	0.216	**0.541**	0.251	0.311	0.311	0.386	0.365
Handwriting	0.507	**0.776**	0.527	0.347	0.440	0.296	0.294	0.100	0.041	0.551	0.240	0.248
Heartbeat	0.732	0.517	0.722	0.702	0.698	0.707	0.693	0.299	0.722	0.659	0.353	**0.751**
JapaneseVowels	0.970	0.935	**0.981**	**0.981**	0.978	0.973	0.970	0.828	0.376	0.957	0.427	0.957
Libras	0.867	0.822	0.856	**0.878**	0.872	0.856	0.806	0.728	0.061	0.856	0.200	0.817
LSST	0.520	0.430	0.521	0.590	0.554	0.577	0.545	0.524	0.445	0.601	**0.722**	0.461
MotorImagery	0.560	0.610	0.490	0.460	0.500	0.500	0.480	0.590	0.500	0.510	**0.733**	0.500
NATOPS	**0.944**	0.400	0.894	0.933	0.939	0.928	0.917	0.667	0.306	0.822	0.570	0.911
PenDigits	**0.990**	0.856	0.988	0.987	0.987	0.987	0.966	0.400	0.945	0.977	0.297	0.966
PhonemeSpectra	0.241	0.712	0.234	0.232	0.248	0.220	0.124	**0.878**	0.110	0.253	0.261	0.304
RacketSports	**0.862**	0.855	0.836	0.776	0.809	0.809	0.836	0.822	0.283	0.842	0.528	0.770
SelfRegulationSCP1	0.802	0.270	0.771	0.805	0.788	**0.877**	0.826	0.626	0.778	0.812	0.520	0.809
SelfRegulationSCP2	0.544	**0.826**	0.589	0.533	0.528	0.556	0.511	0.105	0.500	0.550	0.622	0.544
SpokenArabicDigits	0.928	0.197	0.971	0.945	0.949	0.955	**0.981**	0.490	0.918	0.977	0.041	0.934
StandWalkJump	0.533	**0.978**	0.467	0.400	0.600	0.267	0.400	0.977	0.333	0.400	0.502	0.267
UWaveGestureLibrary	0.903	0.938	0.900	0.844	0.922	0.688	0.800	**0.977**	0.259	0.841	0.913	0.784
On 27 datasets:												
AVG	**0.715**	0.669	0.689	0.680	0.702	0.654	0.688	0.619	0.398	0.705	0.469	0.672
Rank	**4.500**	6.259	6.259	6.130	5.370	6.519	6.222	6.815	9.852	5.389	7.667	7.019

Table 10: Full forecasting results on six benchmark datasets. Lower MSE and MAE indicate better performance. Best results are shown in bold and second-best results are underlined.

Models		DTW4TS		TSTCC		TS2Vec		TimesURL		SoftCLT		PlanTS		TimesNet		MOMENT		UniTS		FusAD		iTransformer		DADA	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	48	0.051	0.170	0.158	0.323	0.070	0.205	0.072	0.207	0.067	0.198	0.060	0.185	0.043	0.160	**0.041**	**0.155**	0.043	0.158	0.052	0.178	0.042	0.157	**0.041**	0.157
	96	0.080	0.216	0.222	0.391	0.104	0.250	0.120	0.274	0.105	0.251	0.095	0.238	0.059	0.185	0.057	**0.181**	0.057	0.182	0.066	0.202	0.057	0.183	**0.056**	0.182
	192	0.093	0.232	0.240	0.413	0.130	0.280	0.152	0.307	0.130	0.279	0.126	0.277	0.074	0.207	**0.073**	**0.205**	0.075	0.211	0.082	0.227	**0.073**	0.206	0.074	0.207
	336	0.100	0.244	0.251	0.422	0.147	0.302	0.173	0.332	0.173	0.332	0.148	0.306	0.083	0.225	0.085	0.228	0.089	0.233	0.090	0.239	0.083	**0.221**	**0.082**	0.224
	720	0.117	0.272	0.371	0.546	0.168	0.332	0.209	0.379	0.168	0.332	0.230	0.408	0.088	0.236	0.086	0.231	0.101	0.250	0.092	0.243	**0.082**	**0.227**	0.088	0.233
	Avg	0.088	0.227	0.248	0.419	0.124	0.274	0.145	0.300	0.129	0.278	0.132	0.283	0.069	0.203	0.068	0.200	0.073	0.207	0.076	0.218	**0.067**	**0.199**	0.068	0.201
ETTh2	48	0.109	0.255	0.146	0.303	0.126	0.275	0.132	0.279	0.124	0.274	0.106	0.249	0.105	0.249	**0.099**	**0.237**	0.103	0.245	0.103	0.249	0.103	0.247	0.102	0.242
	96	0.156	0.308	0.181	0.340	0.166	0.323	0.171	0.324	0.166	0.323	0.154	0.303	0.134	0.284	**0.133**	**0.281**	0.144	0.295	0.142	0.295	0.134	0.285	0.135	0.284
	192	**0.177**	**0.328**	0.214	0.369	0.200	0.358	0.208	0.363	0.202	0.360	0.205	0.350	0.190	0.346	0.184	0.336	0.195	0.350	0.180	0.332	0.181	0.335	0.180	0.333
	336	**0.196**	**0.352**	0.227	0.381	0.199	0.360	0.207	0.366	0.200	0.360	0.237	0.383	0.228	0.378	0.221	0.374	0.231	0.386	0.198	0.365	0.214	0.370	0.215	0.371
	720	**0.200**	**0.363**	0.220	0.378	0.209	0.372	0.205	0.366	0.212	0.374	0.317	0.459	0.237	0.395	0.234	0.388	0.235	0.391	0.219	0.371	0.230	0.385	0.227	0.381
	Avg	**0.168**	**0.321**	0.198	0.354	0.180	0.338	0.185	0.340	0.181	0.338	0.204	0.349	0.179	0.330	0.174	0.323	0.182	0.333	**0.168**	0.322	0.172	0.324	0.172	0.322
ETTm1	48	0.021	0.107	0.068	0.206	0.031	0.132	0.029	0.128	0.031	0.132	0.031	0.131	**0.018**	**0.099**	0.019	0.101	0.019	0.100	**0.018**	0.100	**0.018**	**0.099**	0.019	0.101
	96	0.037	0.144	0.100	0.257	0.047	0.168	0.045	0.161	0.043	0.160	0.058	0.183	0.029	0.127	0.029	**0.125**	0.029	0.127	**0.027**	**0.125**	0.029	0.127	0.029	0.126
	192	0.067	0.196	0.153	0.324	0.072	0.206	0.065	0.195	0.070	0.202	0.116	0.268	0.044	0.160	0.043	**0.157**	0.044	0.159	**0.042**	**0.157**	0.047	0.164	0.045	0.162
	336	0.100	0.244	0.197	0.371	0.102	0.247	0.091	0.233	0.100	0.242	0.175	0.340	0.092	0.231	0.057	0.182	0.059	0.186	**0.054**	**0.179**	0.065	0.194	0.059	0.185
	720	0.146	0.299	0.239	0.407	0.135	0.284	0.123	0.273	0.131	0.280	0.248	0.420	0.083	0.221	0.080	0.216	0.082	0.219	**0.074**	**0.210**	0.082	0.219	0.086	0.223
	Avg	0.074	0.198	0.151	0.313	0.077	0.207	0.071	0.198	0.075	0.203	0.126	0.268	0.053	0.168	0.046	0.156	0.047	0.158	**0.043**	**0.154**	0.048	0.161	0.048	0.159
ETTm2	48	0.054	0.171	0.148	0.303	0.066	0.190	0.065	0.189	0.065	0.188	0.058	0.171	0.044	0.144	0.047	0.148	0.051	0.153	0.054	0.169	**0.041**	**0.140**	0.053	0.145
	96	0.073	0.201	0.155	0.312	0.086	0.221	0.086	0.223	0.085	0.220	0.078	0.207	0.073	0.199	**0.068**	0.189	0.073	0.196	0.076	0.211	**0.068**	**0.187**	0.079	0.198
	192	0.104	0.251	0.180	0.340	0.113	0.281	0.119	0.268	0.120	0.268	0.118	0.261	0.112	0.248	**0.102**	**0.236**	0.103	0.239	0.109	0.256	0.108	0.246	0.105	0.241
	336	0.137	0.290	0.201	0.362	0.150	0.305	0.153	0.308	0.157	0.312	0.173	0.319	0.146	0.291	0.144	0.277	0.133	0.277	**0.132**	0.282	0.144	0.288	**0.132**	**0.274**
	720	**0.173**	**0.326**	0.225	0.382	0.190	0.347	0.199	0.356	0.198	0.354	0.256	0.393	0.188	0.338	0.187	0.335	0.185	0.335	0.187	0.339	0.182	0.333	0.182	0.331
	Avg	**0.108**	0.248	0.182	0.340	0.121	0.269	0.124	0.269	0.125	0.268	0.137	0.270	0.113	0.244	0.110	**0.237**	0.109	0.240	0.112	0.251	0.109	0.239	0.110	0.238
Electricity	48	**0.252**	**0.260**	0.341	0.354	0.309	0.317	0.300	0.311	0.299	0.315	0.286	0.283	0.303	0.280	0.294	0.274	0.293	0.279	0.343	0.348	0.293	0.279	0.291	0.274
	96	**0.315**	**0.298**	0.407	0.400	0.363	0.354	0.354	0.348	0.362	0.354	0.339	0.327	0.364	0.310	0.339	0.301	0.347	0.309	0.437	0.396	0.347	0.303	0.338	0.303
	192	**0.379**	0.344	0.472	0.443	0.448	0.399	0.431	0.398	0.439	0.402	0.412	0.381	0.439	0.336	0.408	0.351	0.427	0.386	0.580	0.464	0.423	**0.330**	0.403	0.350
	336	**0.501**	0.417	0.579	0.505	0.562	0.470	0.555	0.475	0.562	0.471	0.537	0.461	0.587	0.407	0.532	0.414	0.556	0.456	0.794	0.541	0.575	**0.394**	0.534	0.430
	720	**0.800**	0.584	0.877	0.662	0.862	0.652	0.864	0.665	0.863	0.653	0.843	0.650	1.058	0.593	0.941	0.582	0.912	0.635	1.294	0.741	1.038	**0.579**	0.929	0.658
	Avg	**0.449**	0.381	0.535	0.473	0.509	0.438	0.501	0.439	0.505	0.439	0.483	0.420	0.550	0.385	0.503	0.384	0.507	0.413	0.690	0.498	0.535	**0.377**	0.499	0.403
Exchange	48	**0.048**	**0.169**	0.147	0.317	0.058	0.189	0.050	0.173	0.061	0.193	0.051	0.176	0.060	0.189	0.057	0.181	0.060	0.186	0.315	0.440	0.058	0.181	0.056	0.181
	96	**0.091**	**0.230**	0.191	0.359	0.106	0.256	**0.091**	0.234	0.112	0.264	0.093	0.237	0.108	0.245	0.108	0.245	0.109	0.246	0.373	0.483	0.110	0.246	0.103	0.236
	192	**0.178**	**0.336**	0.356	0.488	0.213	0.373	0.185	0.348	0.220	0.379	0.190	0.353	0.232	0.365	0.234	0.361	0.228	0.357	0.534	0.567	0.220	0.351	0.224	0.354
	336	**0.320**	**0.460**	0.681	0.654	0.458	0.530	0.353	0.484	0.470	0.537	0.389	0.494	0.530	0.568	0.458	0.514	0.454	0.517	0.775	0.682	0.439	0.510	0.464	0.525
	720	**0.780**	**0.691**	1.516	0.988	1.301	0.902	1.242	0.882	1.336	0.916	1.143	0.835	1.297	0.868	1.289	0.858	1.456	0.928	1.068	0.815	1.157	0.825	1.644	0.977
	Avg	**0.283**	**0.377**	0.578	0.561	0.427	0.450	0.384	0.424	0.440	0.458	0.373	0.419	0.445	0.447	0.429	0.432	0.461	0.447	0.613	0.597	0.397	0.423	0.498	0.455
Count (1st/2nd)		31/2		0/0		0/1		1/9		0/1		0/6		2/10		15/20		0/12		13/7		16/10		5/20	