Generalized Time Warping Invariant Dictionary Learning for Time Series Classification and Clustering

Ruiyu Xu, Chao Wang, Yongxiang Li, and Jianguo Wu

***Abstract*—Dictionary learning is an effective tool for pattern recognition and classification of time series data. However, real-world time series data often exhibit temporal misalignment due to temporal delay, scaling or other temporal transformations, which poses significant challenges for effective dictionary learning. Dynamic time warping (DTW) is commonly used for dealing with such misalignment issues. Nevertheless, the DTW suffers overfitting or information loss due to its discrete nature in aligning time series data. To address this issue, we propose a generalized time warping invariant dictionary learning algorithm in this paper. Our approach features a generalized time warping operator, which consists of linear combinations of continuous basis functions for facilitating continuous temporal warping. The integration of the proposed operator and the dictionary learning is formulated as an optimization problem, where the block coordinate descent method is employed to jointly optimize warping paths, dictionaries, and sparse coefficients. The optimized results are then used as hyperspace distance measures to feed classification and clustering algorithms. The superiority of the proposed method in terms of dictionary learning, classification, and clustering is validated through ten sets of public datasets in comparison with various benchmark methods.**

***Index Terms*—Time warping, dictionary learning, time series classification, time series clustering.**

I. Introduction

DICTIONARY learning represents input time series data as the combination of a few basis functions, known as atoms. These atoms contain critical temporal features of the original time series data and thus are defined as the dictionary for the data. The dictionary learning has been successfully applied in various tasks such as feature extraction [1], reconstruction [2], denoising [3], compressed sensing [4], classification [5], [6], and clustering [3], [6]. However, traditional dictionary learning assumes that input data is well-aligned with atoms, which is often violated in practice. For example, biological processes exhibit variability in rate and progress across organisms, strains, individuals, and conditions, which pose difficulties in identifying disease progression among affected individuals [7]. Such misalignment also makes it challenging for traditional methods to reveal latent dictionaries and achieve accurate prediction in subsequent tasks.

In recent decades, various techniques have been proposed to address the misalignment in dictionary learning. They can be broadly classified into three categories: kernel approaches, shapelet learning approaches, and dynamic time warping (DTW, [8]) based approaches.

The Kernel approaches [9], [10], [11], [12], [13] map time series data into an implicit reproducing kernel Hilbert space, and dictionaries and atoms are learned in this implicit space. Although kernel approaches are flexible in dealing with various kinds of misalignments, their operations and learning in implicit space raise practical concerns in interpreting the results [14]. Moreover, the performance of kernel approaches highly depends on the selection of kernels and their parameters, which impairs their performance and robustness in subsequent tasks of classification and clustering [14].

The shapelet approaches [15], [16], [17], [18], [19], [20] represent the original time series data as various local atoms, known as shapelets, which can be used to extract local, phase-independent similarity in shape. Although these methods are capable of learning representative local patterns, there lack of an efficient and interpretable way for integrating local shapelets into a complete time series data. This limits the shapelet methods in providing a comprehensive analysis of global features. Moreover, shapelet approaches require a large training dataset to capture sufficient shapelet candidates, which can be computationally expensive in terms of time and resources.

To provide a more interpretive analysis of global patterns, the third category of approaches [14], [21], [22] introduces a kind of warping operator to align the input time series with a dictionary based on DTW alignment or its variants. In this framework, the sparse coding, dictionary, and warping operators are jointly optimized. However, the effectiveness of these methods is heavily influenced by the DTW alignment performance, which could be compromised due to multiple factors. First, DTW is sensitive to noise, as small fluctuations in the time series can result in significant changes in the alignment. It may produce

Received 29 June 2023; revised 18 March 2024; accepted 12 January 2025. Date of publication 27 January 2025; date of current version 3 April 2025. This work was supported by the NSFC under Grant NSFC-72171003, Grant NSFC-71932006, and Grant NSFC-72101147. Recommended for acceptance by K. M. Lee. *(Corresponding author: Jianguo Wu.)*

Ruiyu Xu and Jianguo Wu are with the Department of Industrial Engineering and Management, Peking University, Beijing 100871, China (e-mail: j.wu@pku.edu.cn).

Chao Wang is with the Department of Industrial and Systems Engineering, University of Iowa, Iowa City, IA 52242 USA.

Yongxiang Li is with the Department of Industrial Engineering and Management, Shanghai Jiao Tong University, Shanghai 200240, China.

Digital Object Identifier 10.1109/TPAMI.2025.3534202

inaccurate warping outcomes in the presence of noise. Second, DTW may yield counterintuitive and suboptimal alignments due to its discrete nature. This could result in a single point from one time series being mapped to a section of another time series [23]. Such alignments would lead to overfitting and information loss about the subtle details. Third, DTW assumes the start and end points of the two time series must be aligned, which is known as the boundary condition. This assumption hinders the application of DTW in aligning partial data among different time series, which is commonly encountered in signal extraction/enhancement.

To address these issues in DTW-based dictionary learning, this paper proposes a novel method called Generalized Time Warping Invariant Dictionary Learning (GTWIDL) that relaxes warping boundaries, expands applications, facilitates interpretability, and improves the performance of dictionary learning. More specifically, inspired by Zhou's works [24], [25] of continuous temporal alignment, the proposed method parameterizes the time warping operators as a linear combination of a few monotonic functions, which achieves more accurate alignments and relaxes the boundary conditions. Our model also features a l_1 Lasso penalty to ensure sparse parameters, which is especially time-efficient when dealing with multi-dimensional time series data. An iterative optimization algorithm is further developed to sequentially update the sparse coefficients, dictionary, and warping operators. Case studies show that this optimization algorithm consistently demonstrates high accuracy and convergence for various datasets. Based on the optimized results, time series classification and clustering algorithms are further developed to take advantage of the improved dictionary performance to achieve superior accuracy in corresponding tasks.

The contributions of the paper are as follows:

- We propose a novel framework for time warping invariant dictionary learning. By formulating the warping operators as a continuous and unconstrained warping path, our approach addresses the limitations of traditional DTW-based methods and facilitates joint alignment of multi-dimensional time series data.
- We develop an efficient optimization algorithm to jointly update the sparse coefficients, dictionary, and warping operators.
- Based on the proposed dictionary learning framework, we further develop classification and clustering algorithms by using the reconstruction error as the loss function. The classification or clustering accuracy is significantly improved.
- The proposed methods are evaluated on various public datasets consisting of both one-dimensional and multi-dimensional time series data. The experimental analysis indicates that the proposed algorithm surpasses the performance of traditional dictionary learning methods and existing time warping invariant techniques.

The remainder of this paper is organized as follows. Section II provides relevant background information and a review of related works. In Section III, we present our proposed generalized time warping invariant dictionary learning framework, optimization algorithm, and related analysis. Time series classification and clustering algorithms are further developed in Section IV. We evaluate the proposed methods on several representative datasets to demonstrate their effectiveness in Section V. Finally, Section VI provides a brief conclusion of this study.

II. Related Work

In this section, we present a brief overview of the related works, focusing on temporal alignment and time warping invariant approaches in dictionary learning, classification, and clustering methods.

A. Temporal Alignment

Misaligned data, typically unsynchronized along the time axis, often arises from multiple sources or disparate collection times and locations. Temporal alignment synchronizes these misaligned data streams or time series, enhancing feature and pattern extraction efficiency. A key focus in this field is Dynamic Time Warping (DTW) [8]. The DTW adjusts time series by local stretching and compression to minimize their euclidean difference post-alignment, thereby enabling an optimized pairwise match. Specifically, DTW is defined as follows.

Given two time series, $\mathrm{x} = [x_1, x_2, \ldots, x_{n_x}] \in \mathbb{R}^{n_x}$ and $\mathrm{y} = [y_1, y_2, \ldots, y_{n_y}] \in \mathbb{R}^{n_y}$, DTW aligns x and y such that the sum of the distances between the aligned samples is minimized:

$$\min_{\{\boldsymbol{p}^x, \boldsymbol{p}^y\} \in \boldsymbol{\Psi}} d_{DTW}(\mathrm{x}, \mathrm{y}) = \sum_{t=1}^{l} \left\| x_{p_t^x} - y_{p_t^y} \right\|_2^2, \tag{1}$$

where $\boldsymbol{p}^x \in \{1 : n_x\}^l$ and $\boldsymbol{p}^y \in \{1 : n_y\}^l$ are the warping paths of time series x and y respectively, $l \geq \max(n_x, n_y)$ is the number of indices used to align the samples, and l is automatically selected by the DTW algorithm. The ith frame in x, i.e., x_i, and the jth frame in y, i.e., y_j, are aligned if there exist $p_t^x = i$ and $p_t^y = j$ at some timestamp t. $\boldsymbol{\Psi}$ gives the constraints that warping paths $\boldsymbol{p}^x$ and $\boldsymbol{p}^y$ must satisfy, including the boundary, monotonicity and continuity constraints:

- Boundary: $p_1^x = 1, p_1^y = 1, p_l^x = n_x$ and $p_l^y = n_y$.
- Monotonicity: $t_1 > t_2 \Rightarrow p_{t_1}^x \geq p_{t_2}^x$ and $p_{t_1}^y \geq p_{t_2}^y$.
- Continuity: $[p_t^x, p_t^y] - [p_{t-1}^x, p_{t-1}^y] \in \{[1,1], [0,1], [1,0]\}$.

The DTW algorithm efficiently solves this optimization problem via a dynamic programming approach. The computational cost of the DTW algorithm is $O(n_x n_y)$ in space and time [26].

Extensive research has focused on enhancing the DTW algorithm, particularly in speeding up computations and refining warping path control. Techniques such as global constraints (Sakoe-Chiba band, Itakura Parallelogram band [27]) and multi-level searching approaches [26], [28] have significantly expedited DTW. Additionally, advanced feature extraction methods like derivatives [23], phase differences [29], and local structures [30], [31] have been explored to deepen the understanding of alignment. Despite the progress in improving the DTW, these methods still belong to the discrete warping framework, leading to less accurate warping paths. In particular, a single frame of one time series may be assigned to many consecutive frames in the other time series, causing overfitting or information loss. To address this issue, Zhou et al. [24], [25] proposed a Generalized

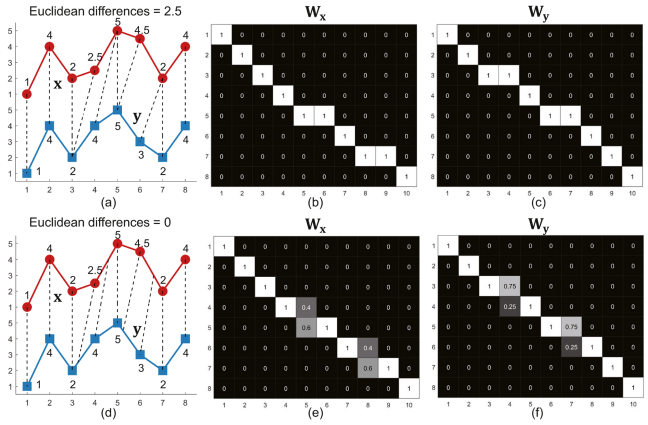


Fig. 1. An example of aligning time series in discrete and continuous ways. (a) Two 1-D time series ($n_x = n_y = 8$) and the discrete alignment between samples computed by DTW. (b)-(c) Discrete warping matrices for two time series. (d) Continuous alignment between samples. (e)-(f) Continuous warping matrices for two time series.

Canonical Time Warping (GCTW) method. The GCTW incorporates a linear combination of monotonic functions to represent the warping path and provides a more flexible and continuous temporal warping. GCTW also relaxes the boundary constraint in regular DTW.

Fig. 1 illustrates temporal alignment using both discrete and continuous methods. Discrete alignment, performed by DTW as shown in Fig. 1(a), involves pairwise alignment between two time series, leading to one-to-many alignments at specific frames. The corresponding discrete warping matrices (Fig. 1(b)–(c)) contain binary elements determined by warping paths. On the other hand, continuous alignment (Fig. 1(d)) allows the interpolation of unobserved frames, where the resulting warping matrices (Fig. 1(e)–(f)) facilitates decimal descriptions for more subtle alignment. For example, the aligned 4th frame (the 4th column in $\boldsymbol{W}_{\mathrm{x}}$ and $\boldsymbol{W}_{\mathrm{y}}$) in discrete and continuous alignment has 0 and 0.5 euclidean difference, respectively, which shows the superiority of the continuous strategy.

Despite its superior performance, the GCTW has not been investigated beyond time series alignment. In other words, there is a research gap/question about whether or how the superior alignment performance in GCTW can benefit broader time series applications such as modeling, classification, and clustering. Our work, i.e., the GTWIDL, aims to fill in this research gap by investigating dictionary learning with GCTW. This represents a significant contribution of our proposed method since it is the first work that provides a general framework for integrating GCTW with dictionary learning, which demonstrates great potential for leading future research in this area.

B. Time Warping Invariant Dictionary Learning

There is very limited research on dealing with time series misalignment via the DTW method in dictionary learning [14], [21], [22]. The general idea of these existing methods is to first align dictionary atoms and samples using the discrete DTW, then the reconstruction error is formulated as the objective of model parameters (dictionary atoms, sparse coefficients, and warping paths) and optimized by block coordinate descent algorithm.

Specifically, Yazdi et al. [21], [22] proposed a variant of the DTW method known as the Cosine Maximization Time Warping operator (COSTW), which aligns time series by minimizing the cosine angle between sequences. Integrated with the Orthogonal Matching Pursuit (OMP) method, it allows for suboptimal optimization of sparse coefficients and warping paths. Subsequently, with fixed sparse coefficients and alignment paths, the dictionary atoms are updated along the gradient direction until convergence. In contrast, Deng et al. [14] proposed to first fix the dictionary and then iteratively refine the warping paths and sparse coefficients through the application of traditional DTW methods and analytical solutions. Then, they utilize Principal Component Analysis (PCA) to scrutinize the aligned data, leading to the extraction of updated dictionary atoms.

Although these methods achieved success in time warping invariant dictionary learning, the use of discrete DTW and the computational load in optimization are two intrinsic issues in existing methods. The proposed GTWIDL method addresses these two issues by investigating the flexibility and feasibility when incorporating the GCTW [25] in dictionary learning. Specifically, by easing constraints and augmenting optimization processes, GTWIDL demonstrates superiority over traditional discrete DTW-based dictionary learning approaches. Moreover, by parameterizing the warping path, GTWIDL transforms the optimization problem related to the warping path into a quadratic programming problem, significantly reducing the algorithm's time complexity.

C. Time Warping Invariant Classification and Clustering

DTW method has been widely implemented in time series classification and clustering for handling misaligned data. It can not only provide distance measures, but also incorporate with techniques like dictionary learning for classification and clustering purposes. We briefly review these approaches below.

In classification tasks, distances between samples are often computed using DTW or its variants (Derivative Dynamic Time Warping [23], Weighted Dynamic Time Warping [29], etc.), with classification conducted via a 1-Nearest Neighbor (1NN) classifier. Alternatively, these sample distances can be input into other classifiers, such as SVMs, for classification [32]. Similarly, in clustering tasks, distance or similarity matrices are calculated using DTW or its variants, followed by spectral clustering for the final grouping [33], [34].

However, these DTW distance-based methods face challenges in handling large datasets. For example, classification tasks require computing DTW distances between all training samples and test samples, consequently leading to high computational complexity. Additionally, DTW distance-based methods only consider distance information, disregarding warping information, which limits their ability in extracting data patterns. To address these challenges, some works have focused on the combination of the DTW technique and centroid-based classification and clustering methods. Petitjean et al. [35] and Soheily et

al. [36] utilized DTW Barycenter Averaging (DBA) and generalized k-means in classification, reducing computational load and providing deeper insights. Similar centroid-based strategies are employed in clustering [37], [38] as well.

These centroid-based methods assume that there is only one centroid per class, failing to account for the presence of multiple patterns within a class. To fill this gap, dictionary learning methods have been utilized to incorporate multiple atoms for modeling the patterns. Based on the previously mentioned time warping invariant dictionary learning methods, Yazdi et al. [21] consolidated various class dictionaries into a global dictionary, aligning samples and representing them sparsely with this dictionary. They then calculated reconstruction errors using only subclass dictionaries, with the classification executed by the smallest error. Further extending this approach to clustering tasks, Yazdi et al. [22] iteratively refined dictionaries based on clustering results and updated cluster assignments based on the dictionaries until convergence. Besides, Deng et al. [14] employed an SVM classifier using the sparse coefficients on the global dictionary as features, and an Efficiently Learning Shapelets (ELS, [39]) feature selection method is used to improve classification accuracy.

Nevertheless, as aforementioned, the DTW suffers intrinsic deficiency due to its discrete nature in aligning time series data. This deficiency in data alignment will be propagated along the modeling and analysis methods and finally impairs the performance of classification and clustering.

As a result, we propose a novel time warping invariant classification and clustering method to acquire more accurate alignment and achieve improved performance in classification and clustering. Notably, global averaging methods under DTW [37] can be regarded as special cases of our proposed GTWIDL algorithm. When the number of dictionary atoms is limited to one, the proposed method essentially mirrors centroid-based methods with an additional scaling factor. The GTWIDL algorithm can adaptively learn the number of atoms, thereby offering enhanced robustness in handling multiple patterns within a class.

III. Time Warping Invariant Dictionary Learning

In this section, we present the formalization of the proposed time series dictionary learning that is invariant to time warping. Instead of using the DTW method, we adopt the concept of the GCTW method and describe warping paths in terms of linear combinations of predefined monotonic functions. Different from the GCTW that focuses on alignment between samples, the proposed GTWIDL learns data patterns by incorporating dictionaries and sparse coefficients, thereby facilitating subsequent tasks such as classification and clustering. Besides, we propose an efficient optimization algorithm, iteratively updating dictionaries, warping paths, and sparse coefficients to achieve optimal solutions. Finally, we provide a complexity analysis for the proposed dictionary learning method.

A. Problem Statement

Given a collection of m one-dimensional time series, $\{\mathrm{x}_i\}_{i=1}^m$, traditional dictionary learning methods seek a dictionary $\boldsymbol{D}$ with a set of K time series atoms $\{\boldsymbol{d}_k\}_{k=1}^K$, and the sparse representing coefficients $\boldsymbol{\alpha}_i = [\alpha_{i1}; \ldots; \alpha_{iK}] \in \mathbb{R}^K$ for each $\mathrm{x}_i = [x_1^i, x_2^i, \ldots, x_n^i] \in \mathbb{R}^{1\times n}$. Such an optimization problem could be solved by minimizing the mean squared error between the input and their reconstructions:

$$\min_{\{\boldsymbol{d}_k\}_{k=1}^K, \{\boldsymbol{\alpha}_i\}_{i=1}^m} \sum_{i=1}^m \left\{ \frac{1}{n} \left\| \mathrm{x}_i - \sum_{k=1}^K \alpha_{ik} \boldsymbol{d}_k \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha}_i \right\|_1 \right\}, \quad \text{s.t.} \quad \|\boldsymbol{d}_k\|_2 = 1, \tag{2}$$

where the l_1 penalty guarantees the sparsity of representing coefficients and λ represents the penalty parameter.

To address data misalignment, we introduce warping matrices $\{\boldsymbol{W}_i\}_{i=1}^m$ into the optimization problem. These matrices facilitate time warping of each sample, i.e., x_i, such that it aligns well with the dictionary. Furthermore, we impose a constraint on the sparse coefficients $\boldsymbol{\alpha}_i \geq \mathbf{0}$, which ensures the coherence of trends between samples and atoms. This constraint aims to reduce the feasible region of the dictionary. Otherwise, the variables $\{\tilde{\boldsymbol{\alpha}}_i\}_{i=1}^m = \{-\boldsymbol{\alpha}_i\}_{i=1}^m$ and $\{\tilde{\boldsymbol{d}}_k\}_{k=1}^K = \{-\boldsymbol{d}_k\}_{k=1}^K$ would produce the same results. As a result, the l_1 penalty is equivalent to a reduced linear function of $\boldsymbol{\alpha}_i$.

$$\min_{\{\boldsymbol{d}_k\}_{k=1}^K, \{\boldsymbol{\alpha}_i\}_{i=1}^m, \{\boldsymbol{W}_i\}_{i=1}^m} \sum_{i=1}^m \left\{ \frac{1}{n_i} \left\| \mathrm{x}_i - \sum_{k=1}^K \alpha_{ik} \boldsymbol{d}_k \boldsymbol{W}_i \right\|_2^2 + \lambda \mathbf{1}' \boldsymbol{\alpha}_i \right\}, \quad \text{s.t.} \ \boldsymbol{\alpha}_i \geq \mathbf{0}, \mathbf{1}_{n_D}' \boldsymbol{W}_i = \mathbf{1}_{n_i}', \|\boldsymbol{d}_k\|_2 = 1, \tag{3}$$

where the length of each sample and the length of atoms may differ, $\mathrm{X}_i \in \mathbb{R}^{1\times n_i}$, $\boldsymbol{d}_k \in \mathbb{R}^{1\times n_D}$, and the warping matrix $\boldsymbol{W}_i \in \mathbb{R}^{n_D \times n_i}$ is a presentation of the monotonous warping path. The predetermined length of atoms n_D is selected as the average length of training samples in our research. Such a framework can be extended to a p-dimensional case.

$$\min_{\{\boldsymbol{D}_j\}_{j=1}^p, \{\boldsymbol{\alpha}_i\}_{i=1}^m, \{\boldsymbol{W}_i\}_{i=1}^m} \sum_{i=1}^m \left\{ \frac{1}{n_i} \sum_{j=1}^p \left\| \mathrm{x}_{ij} - \boldsymbol{\alpha}_i^T \boldsymbol{D}_j \boldsymbol{W}_i \right\|_2^2 + \lambda \mathbf{1}' \boldsymbol{\alpha}_i \right\}, \quad \text{s.t.} \ \boldsymbol{\alpha}_i \geq \mathbf{0}, \mathbf{1}_{n_D}' \boldsymbol{W}_i = \mathbf{1}_{n_i}', \|\boldsymbol{d}_{jk}\|_2 = 1, \tag{4}$$

where X_i is a p-dimensional time series $\mathrm{X}_i \in \mathbb{R}^{p\times n_i}$ and the data in j-th dimension is denoted as x_{ij}. Accordingly, $\boldsymbol{d}_{jk} \in \mathbb{R}^{1\times n_D}$ is the k-th atom in j-th dimension and $\boldsymbol{D}_j = [\boldsymbol{d}_{j1}; \ldots; \boldsymbol{d}_{jK}] \in \mathbb{R}^{K\times n_D}$ denotes the dictionary in j-th dimension.

The proposed framework (4) facilitates multi-pattern recognition for multi-dimensional time series data with K atoms. In addition, compared with the existing dictionary learning approaches [14], [21], [22] mentioned in Section II-C, it significantly reduces the number of variables to be optimized and effectively reduces the computational complexity by sharing the sparse coefficients $\boldsymbol{\alpha}_i$ and the warping matrix $\boldsymbol{W}_i$ across all

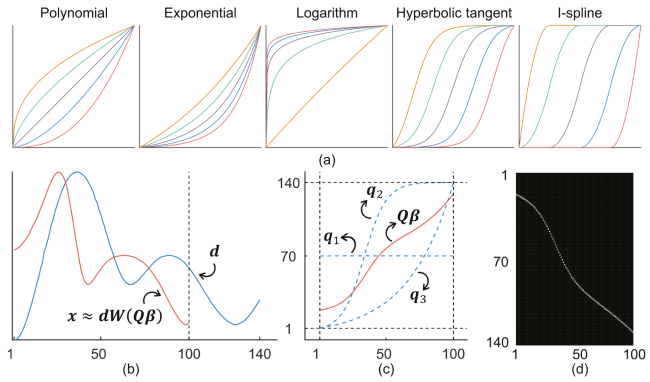


Fig. 2. Representing warping path as a combination of monotonic functions. (a) Five common choices for monotonic functions. (b) An example of time warping for $\mathrm{x} \in \mathbb{R}^{1\times 100}$ which is a sub-part of atom $\boldsymbol{d} \in \mathbb{R}^{1\times 140}$. (c) The warping function $\boldsymbol{p} = \boldsymbol{Q\beta}$ is a linear combination of three basis functions including a constant function (q1) and two monotonic functions (q2) and (q3). (d) The warping matrix $\boldsymbol{W}$ estimates x as $\boldsymbol{dW}$.

dimensions. Despite its advantages, the formulation in (4) is still based on DTW, which suffers various issues as mentioned in Section II-C and demonstrated in Fig. 1.

In order to address the aforementioned weaknesses of DTW, we parameterize the warping path $\boldsymbol{p}_i$ as a linear combination of monotonic functions

$$\boldsymbol{p}_i = \boldsymbol{Q}_i\boldsymbol{\beta}_i, \tag{5}$$

where $\boldsymbol{Q}_i = [\boldsymbol{q}_{i1}, \ldots, \boldsymbol{q}_{i\tilde{k}}] \in \mathbb{R}^{n_i\times\tilde{k}}$ is the basis set composed of $\tilde{k}$ pre-defined monotonically increasing functions and $\boldsymbol{\beta}_i \in \mathbb{R}^{\tilde{k}}, \boldsymbol{\beta}_i \geq 0$ is the non-negative coefficients. The monotonicity property of warping path $\boldsymbol{p}_i$ is preserved due to the fact that any non-negative combination of monotonically increasing trajectories retains monotonicity. Moreover, We provide a strategy to construct a continuous warping matrix $\boldsymbol{W}_i$ since the elements $\{p_{it}\}_{t=1}^{n_i}$ are decimals:

$$p_{it} = j \Rightarrow \boldsymbol{W}_i^{(t,\lfloor j\rfloor)} = \lceil j\rceil - j, \boldsymbol{W}_i^{(t,\lceil j\rceil)} = j - \lfloor j\rfloor, \tag{6}$$

where p_{it} denotes the position of the aligned frame in atoms corresponding to frame t in x_i, and the symbols $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ represent the floor function and the ceiling function, respectively. Therefore, the warping matrix $\boldsymbol{W}_i$ can be presented as a function of $\boldsymbol{\beta}_i$

$$\boldsymbol{W}_i = \boldsymbol{W}_i\left(\boldsymbol{p}_i\left(\boldsymbol{\beta}_i\right)\right) = \boldsymbol{W}_i\left(\boldsymbol{\beta}_i\right). \tag{7}$$

Fig. 2(a) illustrates five common choices for the basis monotonic functions, including (1) polynomial (ax^b), (2) exponential $(\exp(ax+b))$, (3) logarithm $(\log(ax+b))$, (4) hyperbolic tangent $(\tanh(ax+b))$ and (5) I-spline [40]. A constant basis can be added to the basis set to break the constraints that require warping paths to originate from the bottom-left corner and terminate at the upper-right corner.

Guided by the three types of DTW constraints on the warping path, we impose the following constraints on the coefficients $\boldsymbol{\beta}_i's$.

- Boundary: We enforce the position of the first frame, $p_{i1} = \boldsymbol{q}_i^{(1)}\boldsymbol{\beta}_i \in [1, n_D\gamma]$, and the last frame, $p_{in_i} = \boldsymbol{q}_i^{(n_i)}\boldsymbol{\beta}_i \in [n_D(1-\gamma), n_D]$, where $\boldsymbol{q}_i^{(1)} \in \mathbb{R}^{1\times\tilde{k}}$ and $\boldsymbol{q}_i^{(n_i)} \in \mathbb{R}^{1\times\tilde{k}}$ are the first and last rows of the basis matrix $\boldsymbol{Q_i}$ respectively and γ is a constraint parameter to avoid overcompression in the alignment.
- Monotonicity: We enforce $t_1 < t_2 \Rightarrow p_{it_1} \leq p_{it_2}$ by constraining the sign of the coefficients $\boldsymbol{\beta}_i \geq 0$.
- Continuity: We do not impose constraints in this aspect since the k pre-defined continuous basis functions ensure continuity.

In contrast to DTW, which imposes a tight boundary (i.e., $p_{i1} = 1, p_{in_i} = n_D$), our method allows sub-sequence alignments where the samples can be aligned with a sub-part of atoms. This relaxation is useful when the collected time series reflects part of a complete pattern. For instance, Fig. 2 illustrates an example of matching a shorter sample (red) to a sub-sequence of the longer atom (blue). In this sub-sequence alignment problem, our method models the warping path $\boldsymbol{p}$ as a linear combination of three basis functions including a constant function (q1) and two monotonic functions (q2) and (q3). The warping matrix $\boldsymbol{W}$ is then calculated by (6), which satisfies $\mathbf{1}'_{140}\boldsymbol{W} = \mathbf{1}'_{100}$.

In summary, we constrain the warping path by adding the following constraints on the coefficients $\boldsymbol{\beta}_i's$,

$$\boldsymbol{L}_i\boldsymbol{\beta}_i \leq \boldsymbol{b}_i, \tag{8}$$

where $\boldsymbol{L}_i = [-\boldsymbol{I}_{\tilde{k}}; -\boldsymbol{q}_i^{(1)}; \boldsymbol{q}_i^{(1)}; \boldsymbol{q}_i^{(n_i)}; -\boldsymbol{q}_i^{(n_i)}] \in \mathbb{R}^{(\tilde{k}+4)\times\tilde{k}}$ and $\boldsymbol{b}_i = [\mathbf{0}_{\tilde{k}}; -1; n_D\gamma; n_D; -n_D(1-\gamma)] \in \mathbb{R}^{\tilde{k}}$. The optimization problem (4) turns to the following problem

$$\min_{\{\boldsymbol{D}_j\}_{j=1}^{p},\{\boldsymbol{\alpha}_i\}_{i=1}^{m},\{\boldsymbol{\beta}_i\}_{i=1}^{m}} \sum_{i=1}^{m} \left\{\frac{1}{n_i}\sum_{j=1}^{p}\left\|\mathrm{x}_{ij} - \boldsymbol{\alpha}_i^T\boldsymbol{D}_j\boldsymbol{W}_i\left(\boldsymbol{\beta}_i\right)\right\|_2^2 + \lambda\mathbf{1}'\boldsymbol{\alpha}_i\right\},$$
$$\text{s.t. } \boldsymbol{\alpha}_i \geq \mathbf{0}, \boldsymbol{L}_i\boldsymbol{\beta}_i \leq \boldsymbol{b}_i, \left\|\boldsymbol{d}_{jk}\right\|_2 = 1, \tag{9}$$

We name this formulation along with the sequential optimization algorithm as generalized time warping invariant dictionary learning (GTWIDL).

B. Optimization

The optimization problem (9) is non-convex with respect to the dictionary atoms $\{\boldsymbol{d}_k\}_{k=1}^{K}$, the sparse coefficients $\{\boldsymbol{\alpha}_i\}_{i=1}^{m}$, and the combination coefficients of warping paths $\{\boldsymbol{\beta}_i\}_{i=1}^{m}$. In this subsection, we propose an efficient algorithm to solve this problem via the block coordinate descent (BCD) approach. The proposed algorithm consists of two steps, which alternately optimize the dictionary atoms with two coefficients fixed and optimize the two coefficients with dictionary atoms fixed.

In the first step, with dictionary atoms $\{\boldsymbol{d}_k\}_{k=1}^{K}$ fixed, the optimization problem (9) can be decomposed into m independent sub-problems

$$\min_{\boldsymbol{\alpha}_i,\boldsymbol{\beta}_i} \frac{1}{n_i}\sum_{j=1}^{p}\left\|\mathrm{x}_{ij} - \boldsymbol{\alpha}_i^T\boldsymbol{D}_j\boldsymbol{W}_i\left(\boldsymbol{Q}_i\boldsymbol{\beta}_i\right)\right\|_2^2 + \lambda\mathbf{1}'\boldsymbol{\alpha}_i,$$
$$\text{s.t. } \boldsymbol{\alpha}_i \geq \mathbf{0}, \boldsymbol{L}_i\boldsymbol{\beta}_i \leq \boldsymbol{b}_i. \tag{10}$$

It is worth noting that parallel computing can be easily implemented in these independent sub-problems. To shorten the notation, we denote the term $\boldsymbol{\alpha}_i^T \boldsymbol{D}_j \boldsymbol{W}_i(\boldsymbol{Q}_i\boldsymbol{\beta}_i)$ as $\boldsymbol{Z}(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) = [z_1(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i), \ldots, z_{n_i}(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)] \in \mathbb{R}^{1\times n_i}$. For the t-th frame, we have

$$z_t(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) = \left[\boldsymbol{\alpha}_i^T \boldsymbol{D}_j \boldsymbol{W}_i(\boldsymbol{Q}_i\boldsymbol{\beta}_i)\right]_t = \boldsymbol{\alpha}_i^T \boldsymbol{D}_j \left[\boldsymbol{W}_i(\boldsymbol{Q}_i\boldsymbol{\beta}_i)\right]_t, \tag{11}$$

where $[\cdot]_t$ denotes the t-th column of a matrix. First consider a simple case where $\boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i$ is an integer, where $\boldsymbol{q}_i^{(t)} \in \mathbb{R}^{1\times k}$ are the t-th row of the basis matrix $\boldsymbol{Q_i}$. $[\boldsymbol{W}_i(\boldsymbol{Q}_i\boldsymbol{\beta}_i)]_t$ is a zero vector with only the $\boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i$-th element equal 1. Then, $\boldsymbol{\alpha}_i^T \boldsymbol{D}_j[\boldsymbol{W}_i(\boldsymbol{Q}_i\boldsymbol{\beta}_i)]_t$ equals the $\boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i$-th frame of $\boldsymbol{\alpha}_i^T \boldsymbol{D}_j$, i.e., $z_t(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) = [\boldsymbol{\alpha}_i^T \boldsymbol{D}_j]_{\boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i}$. The first-order Taylor expansion of z_t around $\boldsymbol{\beta}_i$ can be represented as

$$\begin{aligned} z_t(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i + \Delta\boldsymbol{\beta}_i) &\approx z_t(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) + \nabla\left(\boldsymbol{\alpha}_i^T \boldsymbol{D}_j\right)\big|_{\boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i} \frac{\partial \boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i}{\partial \boldsymbol{\beta}_i} \Delta\boldsymbol{\beta}_i, \\ &= z_t(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) + \nabla\left(\boldsymbol{\alpha}_i^T \boldsymbol{D}_j\right)\big|_{\boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i} \boldsymbol{q}_i^{(t)} \Delta\boldsymbol{\beta}_i, \end{aligned} \tag{12}$$

where $\Delta\boldsymbol{\beta}_i$ denotes a small displacement around $\boldsymbol{\beta}_i$ and $\nabla(\boldsymbol{\alpha}_i^T \boldsymbol{D}_j)|_{\boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i} \in \mathbb{R}$ denotes the gradient of the combination of atoms $\boldsymbol{\alpha}_i^T \boldsymbol{D}_j$ around the $\boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i$-th frame. The term, $\frac{\partial \boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i}{\partial \boldsymbol{\beta}_i} = \boldsymbol{q}_i^{(t)} \in \mathbb{R}^{1\times \tilde{k}}$, is the Jacobian of the warping path. Equation (12) can be extended to the generalized cases where $\boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i$ is a decimal. The gradient term, $\nabla(\boldsymbol{\alpha}_i^T \boldsymbol{D}_j)|_{\boldsymbol{q}_i^{(t)}\boldsymbol{\beta}_i}$, can be calculated by linear interpolation. Put together the approximations of $z_t(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i + \Delta\boldsymbol{\beta}_i)$ and we have

$$\begin{aligned} \boldsymbol{Z}^T(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i + \Delta\boldsymbol{\beta}_i) &\approx \boldsymbol{Z}^T(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) + \boldsymbol{G}_{ij}^{\beta} \Delta\boldsymbol{\beta}_i, \\ \text{where } \boldsymbol{G}_{ij}^{\beta} &= \begin{bmatrix} \nabla\left(\boldsymbol{\alpha}_i^T \boldsymbol{D}_j\right)\big|_{\boldsymbol{q}_i^{(1)}\boldsymbol{\beta}_i} \boldsymbol{q}_i^{(1)} \\ \vdots \\ \nabla\left(\boldsymbol{\alpha}_i^T \boldsymbol{D}_j\right)\big|_{\boldsymbol{q}_i^{(n_i)}\boldsymbol{\beta}_i} \boldsymbol{q}_i^{(n_i)} \end{bmatrix} \in \mathbb{R}^{n_i\times \tilde{k}}. \end{aligned} \tag{13}$$

The first-order Taylor expansion of z_t around $\boldsymbol{\alpha}_i$ can be represented as

$$\boldsymbol{Z}^T(\boldsymbol{\alpha}_i + \Delta\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) \approx \boldsymbol{Z}^T(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) + \boldsymbol{G}_{ij}^{\alpha} \Delta\boldsymbol{\alpha}_i, \tag{14}$$

where $\boldsymbol{G}_{ij}^{\alpha} = \boldsymbol{W}_i^T(\boldsymbol{Q}_i\boldsymbol{\beta}_i)\boldsymbol{D}_j^T \in \mathbb{R}^{n_i\times K}$. Ignoring the second-order interaction term of $\Delta\boldsymbol{\alpha}_i$ and $\Delta\boldsymbol{\beta}_i$, we have the following approximation

$$\boldsymbol{Z}^T(\boldsymbol{\alpha}_i + \Delta\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i + \Delta\boldsymbol{\beta}_i) \approx \boldsymbol{Z}^T(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) + \boldsymbol{G}_{ij}^{\alpha} \Delta\boldsymbol{\alpha}_i + \boldsymbol{G}_{ij}^{\beta} \Delta\boldsymbol{\beta}_i. \tag{15}$$

Now we use the Gauss-Newton method to iteratively update $\hat{\boldsymbol{\alpha}}_i = \boldsymbol{\alpha}_i + \Delta\boldsymbol{\alpha}_i$ and $\hat{\boldsymbol{\beta}}_i = \boldsymbol{\beta}_i + \Delta\boldsymbol{\beta}_i$ by plugging (15) into optimization problem (10)

$$\begin{aligned} \min_{\Delta\boldsymbol{\alpha}_i, \Delta\boldsymbol{\beta}_i} \frac{1}{n_i} \sum_{j=1}^{p} \left\| \boldsymbol{v}_{ij} - \boldsymbol{G}_{ij}^{\alpha} \Delta\boldsymbol{\alpha}_i - \boldsymbol{G}_{ij}^{\beta} \Delta\boldsymbol{\beta}_i \right\|_2^2 + \lambda \mathbf{1}'(\boldsymbol{\alpha}_i + \Delta\boldsymbol{\alpha}_i), \\ \text{s.t. } \boldsymbol{\alpha}_i + \Delta\boldsymbol{\alpha}_i \geq 0, \boldsymbol{L}_i(\boldsymbol{\beta}_i + \Delta\boldsymbol{\beta}_i) \leq \boldsymbol{b}_i, \end{aligned} \tag{16}$$

where $\boldsymbol{v}_{ij} = \mathrm{x}_{ij}^T - \boldsymbol{\alpha}_i^T \boldsymbol{D}_j \boldsymbol{W}_i(\boldsymbol{Q}_i\boldsymbol{\beta}_i)$. Such an optimization problem can be transformed into a quadratic programming problem

$$\min_{\boldsymbol{\delta}_i} \frac{1}{2} \boldsymbol{\delta}_i^T \boldsymbol{H}_i \boldsymbol{\delta}_i + \boldsymbol{f}_i^T \boldsymbol{\delta}_i, \text{ s.t. } \tilde{\boldsymbol{L}}_i \boldsymbol{\delta}_i \leq \tilde{\boldsymbol{b}}_i - \tilde{\boldsymbol{L}}_i \boldsymbol{\eta}_i, \tag{17}$$

where $\boldsymbol{\delta}_i = \begin{bmatrix} \Delta\boldsymbol{\alpha}_i \\ \Delta\boldsymbol{\beta}_i \end{bmatrix}$, $\boldsymbol{\eta}_i = \begin{bmatrix} \boldsymbol{\alpha}_i \\ \boldsymbol{\beta}_i \end{bmatrix}$, $\boldsymbol{H}_i = \frac{1}{n_i} \sum_{j=1}^{p} \boldsymbol{G}_{ij}^T \boldsymbol{G}_{ij}$, $\boldsymbol{f}_i = -\frac{1}{n_i} \sum_{j=1}^{p} \boldsymbol{G}_{ij}^T \boldsymbol{v}_{ij} + \frac{1}{2}\lambda[\mathbf{1}_K; \mathbf{0}_{\tilde{k}}]$, $\tilde{\boldsymbol{L}}_i = \begin{bmatrix} -\boldsymbol{I}_K, \mathbf{0} \\ \mathbf{0}, \boldsymbol{L}_i \end{bmatrix}$, $\tilde{\boldsymbol{b}}_i = [\mathbf{0}_K; \boldsymbol{b}_i]$, $\boldsymbol{G}_{ij} = [\boldsymbol{G}_{ij}^{\alpha}, \boldsymbol{G}_{ij}^{\beta}]$.

Such quadratic programming problems can be easily solved and the optimization process turns to solve multiple quadratic programming problems, which is equivalent to the sequential quadratic programming (SQP) method. In this method, the initialization of $\boldsymbol{\alpha}_i$ is randomly sampled from the interval $[0, 1]$ and satisfies the condition $\sum \boldsymbol{\alpha}_i = 1$. The initialization of $\boldsymbol{\beta}_i$ is selected to ensure that the generated warping path $\boldsymbol{q}_i$ is a straight line from bottom left to top right without any time warping. The iterative updates of $\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i$ are performed until convergence. The stop conditions are set as

$$\Delta\boldsymbol{\alpha}_i \leq \epsilon_{\alpha}, \Delta\boldsymbol{\beta}_i \leq \epsilon_{\beta}, \tag{18}$$

where ϵ_{α} and ϵ_{β} are pre-specified small positive numbers (e.g., 10^{-3}).

In the second step, with the coefficients $\boldsymbol{\alpha}_i's, \boldsymbol{\beta}_i's$ fixed, the optimization problem of updating dictionary atoms $\boldsymbol{D}_j$ can be represented as

$$\begin{aligned} \min_{\{\boldsymbol{d}_{jk}\}_{k=1}^K} \frac{1}{n_i} \sum_{i=1}^{m} \left\| \mathrm{x}_{ij} - \sum_{k=1}^{K} \boldsymbol{\alpha}_{ik} \boldsymbol{d}_{jk} \boldsymbol{W}_i(\boldsymbol{\beta}_i) \right\|_2^2, \\ \text{s.t. } \|\boldsymbol{d}_{jk}\|_2 = 1. \end{aligned} \tag{19}$$

We iteratively update the k-th atom $\boldsymbol{d}_{jk}$ with the other atoms fixed. The update formula can be easily deduced by derivation

$$\hat{\boldsymbol{d}}_{jk} = \boldsymbol{d}_{jk} + \left(\sum_{i=1}^{m} \alpha_{ik} \boldsymbol{u}_{ik} \boldsymbol{W}_i^T \right) \left(\sum_{i=1}^{m} \alpha_{ik}^2 \boldsymbol{W}_i \boldsymbol{W}_i^T \right)^{-1}, \tag{20}$$

where $\boldsymbol{u}_{ik} = \mathrm{x}_{ij} - \sum_{k'\neq k} \boldsymbol{\alpha}_{ik'} \boldsymbol{d}_{jk'} \boldsymbol{W}_i(\boldsymbol{\beta}_i)$ denotes the residual without the k-th atom. However, the existence of the inverse term is not guaranteed due to the potential occurrence of a warping matrix $\boldsymbol{W}_i$ with zeros at the start and end frames. Therefore, we approximate this update formula with

$$\hat{\boldsymbol{d}}_{jk} = \boldsymbol{d}_{jk} + \frac{1}{\sum_{i=1}^{m} \alpha_{ik}^2} \left(\sum_{i=1}^{m} \alpha_{ik} \tilde{\boldsymbol{u}}_{ik} \right), \tag{21}$$

where $\tilde{\boldsymbol{u}}_{ik} = \mathcal{F}(\mathrm{x}_{ij} \tilde{\boldsymbol{W}}_i(\boldsymbol{\beta}_i)) - \sum_{k'\neq k} \boldsymbol{\alpha}_{ik'} \boldsymbol{d}_{jk'}$ and the missing value imputation function $\mathcal{F}(\cdot)$ is used to fill the leading and trailing missing values of $\mathrm{x}_{ij} \tilde{\boldsymbol{W}}_i(\boldsymbol{\beta}_i)$ with their nearest non-missing value. Here, $\tilde{\boldsymbol{W}}_i$ denotes the inverse warping matrix which warps time series x_{ij} to the space of the dictionary.

In this method, the initialization of $\boldsymbol{d}_{jk}'s$ is set as the first $\tilde{k}$ largest eigenvectors based on the samples after stretching. The

Algorithm 1: GTWIDL Algorithm.

Input : $\{\boldsymbol{X}_i\}_{i=1}^m, \lambda, K$
Output: $\{\boldsymbol{D}_j\}_{j=1}^p, \{\boldsymbol{\alpha}_i\}_{i=1}^m, \{\boldsymbol{\beta}_i\}_{i=1}^m$
Initialization: Initial $\{\boldsymbol{D}_j\}_{j=1}^p, \{\boldsymbol{\alpha}_i\}_{i=1}^m, \{\boldsymbol{\beta}_i\}_{i=1}^m$ and $\{\boldsymbol{Q}_i\}_{i=1}^m$ **repeat**
  for $i = 1, \ldots, m$ **do in parallel**
    repeat
      calculate $\boldsymbol{H}_i, \boldsymbol{f}_i, \tilde{\boldsymbol{L}}_i, \tilde{\boldsymbol{b}}_i$;
      update $\hat{\boldsymbol{\alpha}}_i \leftarrow \boldsymbol{\alpha}_i + \Delta\boldsymbol{\alpha}_i$ and $\hat{\boldsymbol{\beta}}_i \leftarrow \boldsymbol{\beta}_i + \Delta\boldsymbol{\beta}_i$ by solving (17);
    until *convergence*;
  end
  for $j = 1, \ldots, p$ **do in parallel**
    repeat
      for k=*1,. . .,K* **do**
        calculate $\tilde{\boldsymbol{u}}_{ik}$;
        update $\hat{\boldsymbol{d}}_{jk} \leftarrow \boldsymbol{d}_{jk} + \frac{1}{\sum_{i=1}^m \alpha_{ik}^2}\left(\sum_{i=1}^m \alpha_{ik}\tilde{\boldsymbol{u}}_{ik}\right)$;
      end
    until *convergence*;
  end
until *convergence*;

stop conditions are set as

$$\left\|\hat{\boldsymbol{d}}_{jk} - \boldsymbol{d}_{jk}\right\|_2^2 \leq \epsilon_{\boldsymbol{d}},\ k = 1, \ldots, K, \tag{22}$$

where $\epsilon_{\boldsymbol{d}}$ is pre-specified small positive numbers (e.g., 10^{-2}). Similar conditions are set for the relative difference of successive estimates as stopping conditions for the block coordinate descent processes. The optimization algorithm is summarized in Algorithm 1.

C. Complexity Analysis

The time complexity of the GTWIDL algorithm consists of two parts: coefficients update and dictionary update. In the coefficient update part, the time cost of one loop is $O((\tilde{k}+K)^2 mn_D + (\tilde{k}+K)^3 m)$. The first term relates to the calculation of $\boldsymbol{H}_i, \boldsymbol{f}_i, \tilde{\boldsymbol{L}}_i, \tilde{\boldsymbol{b}}_i$ on m samples, and the second term is associated with solving the quadratic problem. In the dictionary update part, the time cost of one loop is $O(\tilde{k}mn_D + Kmn_D)$. Here, the first term accounts for the calculation of $\tilde{\boldsymbol{W}}_i$ on m samples, and the second term relates to the dictionary update based on (21). Since $(\tilde{k}+K) \ll n_D$, the computational complexity for one iteration is $O((\tilde{k}+K)^2 mn_D)$. Denoting T_1, T_2 as the maximum iteration of BCD update and coefficients update separately, the total time complexity of the GTWIDL algorithm is $O(T_1T_2(\tilde{k}+K)^2 mn_D)$. It is demonstrated in our case studies that the GTWIDL algorithm converges fast, where T_1 can be set from 5 to 50 and T_2 can be set from 5 to 20.

The comparison of degrees of freedom and complexity between existing dictionary learning algorithms [14], [21] and the GTWIDL is provided in Table I. By utilizing a linear combination of basis functions to approximate the warping path, our proposed method requires fewer optimization parameters compared to other approaches. In addition, the GTWIDL method offers superior computational efficiency due to its linear complexity with respect to the sequence length. In contrast, other methods require quadratic time complexity to calculate the DTW warping path and cubic time complexity to implement PCA or SVD for dictionary updating. The running time comparison in a simulation study is illustrated in Fig. 3. As the increase of sequence length, TWI-kSVD [21] and RISLDTW [14] require much more running time compared with the proposed GTWIDL method.

TABLE I
COMPARISON OF DICTIONARY LEARNING ALGORITHMS IN DEGREES OF FREEDOM AND COMPLEXITY

Method	Degrees of Freedom			Complexity $O(\cdot)$
	Warping	Coding	Learning	
TWI-kSVD	mpn_D	mK	pKn_D	$T_1\left(T_2mn_D^2 + Kn_D^3\right)$
RISLDTW	mn_D	mpK	pKn_D	$T_1\left(T_2mn_D^2 + pn_D^3\right)$
GTWIDL	$m\tilde{k}$	mK	pKn_D	$T_1T_2\left(\tilde{k}+K\right)^2 mn_D$

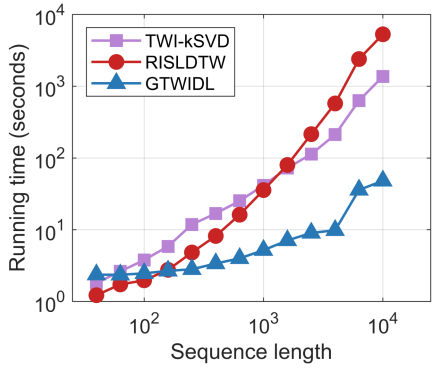


Fig. 3. Simulation of the running time (in seconds) for dictionary learning on a dataset containing 10 samples.

IV. CLASSIFICATION AND CLUSTERING BASED ON GTWIDL

In this section, we propose classification and clustering algorithms tailored for the proposed GTWIDL, where the basic idea is to assign test samples to the class whose dictionary yields the minimum reconstruction error.

A. Classification Based on GTWIDL

Various approaches have been proposed for using dictionary learning algorithms in classification tasks. One such method [14] involves training a separate dictionary for each class and then combining them into a global dictionary. This global dictionary is then used to sparsely code both training and test samples. The global sparse coefficients are then used for subsequent classification tasks through common approaches, e.g., KNN and SVM. Another approach [21], [41] employs the same global dictionary to learn the alignment path and global sparse coefficients for each test sample. Then, the reconstruction error is calculated for each class based on the sub-dictionary and sub-corresponding sparse coefficients. The test sample is assigned to the class with the minimum reconstruction error. However, the local dictionaries

may share similar atoms across different classes. Therefore, the global dictionary may suffer homogeneous atoms and produce non-unique optimal sparse coding results. In such cases, the sparse coefficients and reconstruction error under the global dictionary may not be sufficiently accurate and discriminative for the classification task.

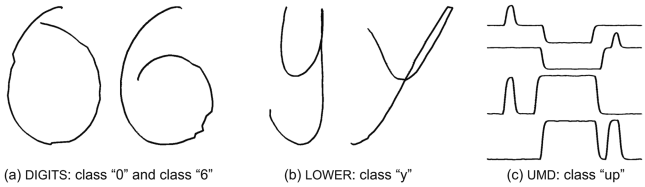
In this subsection, we adopt a classification strategy that focuses on the local dictionary per class, which is also used in [6], [42]. First, all dictionaries are learned using training samples based on the proposed GTWIDL method. Then, given the dictionary, we optimize Problem (10) for each test sample under each class with respect to the warping path and sparse coding. Subsequently, we calculate the reconstruction error by $\sum_{j=1}^{p} \|\mathrm{x}_{ij} - \boldsymbol{\alpha}_i^T \boldsymbol{D}_j \boldsymbol{W}_i (\boldsymbol{Q}_i \boldsymbol{\beta}_i)\|_2^2$. Finally, the test sample is assigned to the class exhibiting the minimum reconstruction error.

The complexity of such a classification algorithm is linear with respect to the sequence length, making it superior to other time warping invariant classification methods. A notable advantage of our approach is that it does not need to implement DTW for every pair of training and test samples, as required by DTW distance-based methods. Instead, our algorithm only requires the assignment of each test sample to a few atoms, rendering it more efficient for large datasets.

One potential concern in the proposed classification method is that the reconstruction error may be sensitive to the number of atoms K in each dictionary. It is observed that as the number of dictionary atoms increases, the reconstruction error decreases rapidly at first, and then keeps stable around a certain value. Therefore, we propose an adaptive strategy to select a suitable K for different classes. This approach is superior to setting a fixed K for all classes as it not only ensures that the reconstruction error approximately converges but also avoids the computation with an excessive number of dictionary atoms. We first learn a dictionary with enough atoms to fully fit the training samples. Subsequently, we apply Singular Value Decomposition (SVD) on the warped time series to obtain the eigenvalues, denoted as $\lambda_1 \geq \cdots \geq \lambda_m$. The value of K is then chosen such that $\frac{\sum_{k=1}^{K} \lambda_i}{\sum_{k=1} \lambda_i} \geq \zeta$ and $\frac{\sum_{k=1}^{K-1} \lambda_i}{\sum_{k=1} \lambda_i} < \zeta$, where ζ reflects how well the dictionary fits the data. The model parameter ζ and λ are further selected through cross-validation. The details of this tuning process can be found in Section V-B. Note that due to the inherent differences in data patterns among classes, the atoms of one class may be meaningless to the data samples from other classes. As a result, it is highly likely that the reconstruction error for samples using their own class's dictionary will be smaller than using dictionaries of other classes, even if the dictionary sizes of other classes are much larger.

B. Clustering Based on GTWIDL

In this subsection, we present a clustering method developed based on the GTWIDL method and the above classification framework. First, we propose a pre-clustering approach handling temporal alignment. This approach initially learns a global dictionary, along with the corresponding warping paths and sparse coefficients for all samples. The global sparse coefficients are then utilized in spectral clustering to achieve an initialization of clustering. Subsequently, similar to the classification method previously introduced, the initial clustering results are utilized as temporary labels to learn dictionaries for all classes. Samples are then reassigned to the class whose dictionary yields the minimal reconstruction error. We repeat the step of dictionary learning and label reassignment until convergence. The parameters of the GTWIDL method are set to their default values, i.e., $\lambda = 0.0001$ and $\zeta = 0.7$, in this task.



(a) DIGITS: class "0" and class "6" (b) LOWER: class "y" (c) UMD: class "up"

Fig. 4. Time series characteristics within and between classes. (a) Examples of class "0" and class "6" in dataset DIGITS. (b) Examples of class "y" in dataset LOWER. (c) Examples of class "up" in dataset UMD.

V. Experiments

In this section, the performance of the proposed GTWIDL method is evaluated from three perspectives: (1) reconstruction to demonstrate its superior representation ability; (2) classification; and (3) clustering to investigate its discriminative ability. All experiments were conducted using Matlab on an Intel Xeon Processor E5-4627 v4 with 128 GB RAM, ensuring robust and reliable results.

A. Datasets

The research experiments were performed on a total of ten datasets, which comprise four public handwritten character datasets, two synthetic datasets, and four time series datasets from the UCR archive [43]. The four public handwritten character datasets used included DIGITS, LOWER, UPPER, and CHAR-TRAJ. These datasets involve naturally multivariate time series of varying lengths [44], and represent the 2-dimensional handwritten character trajectory of air-handwritten motion gestures of digits, upper and lower case letters, and other characters. Furthermore, two synthetic datasets, namely BME and UMD, were employed, which contained time series exhibiting local temporal features within classes while demonstrating distinctive global patterns. In addition, four UCR time series datasets with varying delays were chosen, including ArrowHead, DSR (short for DiatomSizeReduction), FiftyWords, and Trace. All these datasets were subject to misalignment to some extent.

Fig. 4 illustrates the characteristics of the considered time series through several examples. For instance, time series may share similar global patterns between different classes, such as "0" and "6". Time series within the same class, such as "y", may exhibit similar global patterns but different shapes representing different writing styles. Time series may even have different global patterns, as in the case of the "up" class, while sharing only local events (such as the "small bell") that may occur at different time stamps.

TABLE II
DATA DESCRIPTION

Dataset	Nb. class	Dimensions	Train size	Test size	Length range
DIGITS	10	2	100	500	29∼218
LOWER	26	2	260	1210	27∼263
UPPER	26	2	260	6241	27∼412
CHAR-TRAJ	20	3	200	2658	109∼205
UMD	3	1	360	1440	150
BME	3	1	300	1500	128
ArrowHead	3	1	36	175	251
DSR	4	1	16	306	345
FiftyWords	50	1	450	455	270
Trace	4	1	100	100	275

TABLE III
THE CANDIDATE SETS OF PARAMETERS IN THE CLASSIFICATION FRAMEWORK

	Para.	Ranges	Description
WDTW	g	[0.01,0.02,0.03,0.05,0.08,0.1]	Weighted coefficient
TWI-kSVD	K	10	Dictionary size
	τ	2	Sparsity constraint
RISLDTW	c	[0,0.1,0.2,0.3,0.4,0.5,0.6]	Variance proportion
RISLDTW-ELS	α_1	[0,0.01,0.02,0.04,0.06,0.08,0.1]	Regularisation
	α_2	[0,0.01,0.02,0.03,0.04,0.05]	Regularisation
GTWIDL	λ	[0.001,0.0005,0.0001,0.00005,0]	Sparsity constraint
	ζ	[0.5,0.7,0.8,0.9,0.95,0.99]	Proportion threshold

In the experiments of dictionary learning and classification, for a fair comparison, the training and testing data are split based on the definition in the dataset when available. Otherwise, we randomly selected ten samples from each class to constitute the training dataset, with the remaining samples serving as the testing dataset. This random selection process was repeated ten times to ensure the robustness of our results. In the experiments on clustering, we utilized the aforementioned training datasets as our target datasets. The data characteristics are given in Table II. In addition, we use a zero-padding strategy [21] to circumvent the problem of variable lengths for some methods demanding a uniform length and splice multidimensional time series into one-dimensional time series for univariate methods.

B. Benchmarks

In the task of dictionary learning, we use a traditional dictionary learning method (DL) without temporal warping and two time warping invariant dictionary learning methods, i.e., TWI-kSVD [21] and RISLDTW [14], as the benchmarks.

In the task of time series classification, a variant of RISLDTW namely RISLDTW-ELS, is added which further uses a feature selection model called Efficiently Learning Shapelets (ELS, [39]) to improve accuracy. Additionally, three distance-based methods are also set as benchmarks. They respectively calculate DTW distance [8], DDTW distance [23], and WDTW distance [29] between training and testing samples. A 1NN classifier is then applied to provide final classification results. For each method, parameters are learned via grid search using three-fold cross-validation on the training dataset. Details of parameter candidate sets are provided in Table III.

In the task of time series clustering, traditional sparse subspace clustering (SSC, [45]), the variant of TWI-kSVD called TWI-DLCLUST, the variant of RISLDTW, and the three DTW distance-based methods are used as benchmarks. For the SSC method, the self-representative coefficients are fed to the spectral clustering method without temporal alignment. For the RISLDTW method, the k-means clustering method is applied for clustering based on the learned sparse coding. For the DTW distance-based methods, clustering results are achieved via spectral clustering method based on the learned distances, where the similarity matrix is calculated by $S_{ij} = \exp(-\frac{d_{ij}}{\sigma^2})$ and σ is set to 5 in the following experiments.

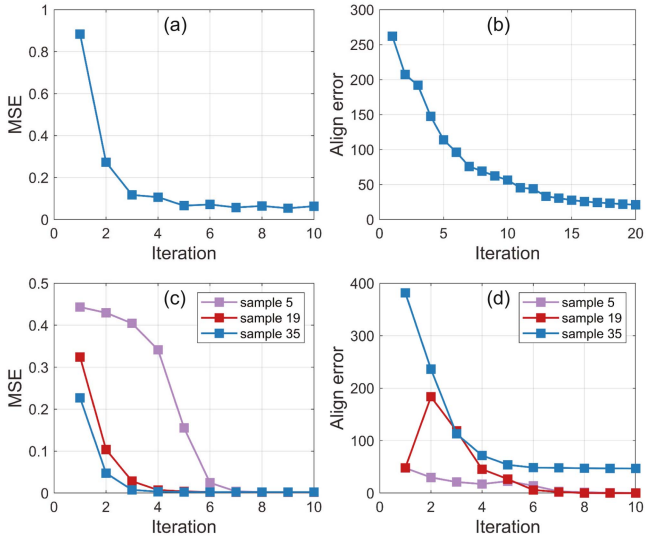


Fig. 5. The reconstruction error and the alignment error against the number of iterations on the first class of the Trace dataset. The top row displays the average iteration curves when updating the dictionary, and the bottom row shows the iteration curves when updating coefficients with a fixed dictionary.

C. Evaluation on Dictionary Learning

In this subsection, we first investigate the convergence properties of the proposed GTWIDL algorithm. Here we set two metrics to evaluate the degree of convergence, i.e., reconstruction error and alignment error. Specifically, the reconstruction error is measured by Mean Squared Error (MSE), which was defined as

$$\mathrm{MSE} = \frac{1}{n_i}\sum_{j=1}^{p}\left\|\mathrm{x}_{ij} - \boldsymbol{\alpha}_i^T \boldsymbol{D}_j \boldsymbol{W}_i\left(\boldsymbol{Q}_i\boldsymbol{\beta}_i\right)\right\|_2^2. \quad (23)$$

The alignment error is measured by the sum-of-pair euclidean distance between the estimated alignment path and the ground truth, which was defined as

$$\mathrm{Error}_{\mathrm{align}} = \frac{1}{n_i}\left\|\boldsymbol{W}_i\left(\boldsymbol{Q}_i\boldsymbol{\beta}_i\right) - \boldsymbol{W}_i^t\right\|_2^2, \quad (24)$$

where $\boldsymbol{W}_i^t$ denotes the ground truth of the alignment path. When the ground truth is unavailable, we use the final alignment path obtained after 100 iterations as a reference.

Fig. 5 illustrates the reconstruction error and the alignment error against the number of iterations on the first class of the Trace dataset. Fig. 5(a)–(b) show the average iteration curves when updating the dictionary, which corresponds to the outer iteration loop in the GTWIDL algorithm. The reconstruction error converges rapidly around the 3 rd iteration, which implies that the proposed algorithm could efficiently handle the misaligned data and fit the samples well. The alignment error

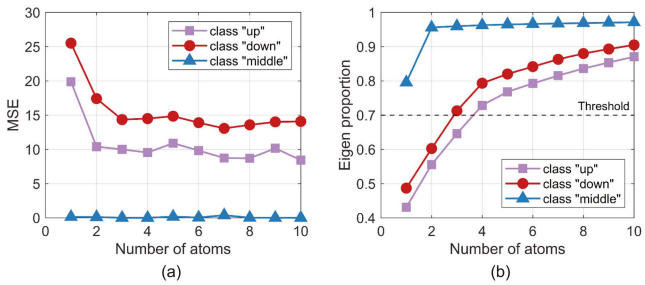


Fig. 6. (a) The reconstruction error against the number of atoms for the UMD dataset. (b) The cumulative proportion of eigenvalues based on the samples after alignments.

converges more slowly, around the 15th iteration. This is because the optimal alignment error is often not unique and different paths may have similar alignment performance. For example, as we can see from Fig. 5(c)–(d), for sample 35 of the Trace dataset, our algorithm converges around the 4th iteration with the reconstruction error close to zero. However, the alignment error converges around value 50, which is much larger than zero. It indicates that the alignment path at the 4th iteration differs from the ground truth but still has excellent alignment performance. Fig. 5(c)–(d) shows the iteration curves when updating coefficients with a fixed dictionary, which corresponds to the inner iteration loop in the GTWIDL algorithm. We test samples 5,19 and 35 from the first class of the Trace dataset. Both the reconstruction error and alignment error converge rapidly around the 5th iteration. As a result, we demonstrate that the proposed GTWIDL algorithm has great convergence properties and is capable of efficiently finding the dictionary and aligning samples. We set T_1 from 5 to 50 and set T_2 from 5 to 20 to achieve such performance, which is discussed in Section III-C.

To investigate the influence of the number of atoms on the proposed algorithm, we test the GTWIDL algorithm on the UMD dataset, which contains several different patterns in each class. For example, the "up" and "down" classes both have four different patterns, while the "middle" class has only one pattern. Fig. 6(a) illustrates the reconstruction error against the number of atoms. The MSE value for the "up" and "down" classes converge when the number of atoms equals four, while the MSE value for the "middle" class converges at the first point. To avoid overfitting and provide refined atoms, the number of atoms should not be too large and should be set around the number of underlying patterns. As introduced in Section IV-A, by setting the threshold ζ at approximately 0.7 (shown in Fig. 6(b)), the optimal value for the dictionary's dimension can be obtained. However, it is important to note that the optimal thresholds may vary depending on the dataset being analyzed. To accommodate such variations, we provide a candidate list for 3-fold cross-validation to select the threshold in the classification framework, which is displayed in Table III.

Fig. 7 illustrates the progress of applying the proposed GTWIDL algorithm to the Trace dataset. The initial misaligned samples belonging to four different patterns are shown in Fig. 7(a). The atoms learned directly from the misaligned data (Fig. 7(e)) fail to capture the local patterns accurately. This is particularly clear for class 1 and class 2, where the learned atoms tend to average the misaligned samples, resulting in abrupt rises and falls. After the first iteration, the data are roughly aligned (Fig. 7(b)), and the simple patterns of class 2 and class 4 can be learned accurately (Fig. 7(f)). With an increasing number of iterations, the alignments of samples become more precise. After the third iteration (Fig. 7(d)), all samples from the four classes are well-aligned. Furthermore, the learned atoms increasingly capture the local patterns and accurately represent the characteristics of different samples. Among the four classes, samples from class 3 are the most challenging to align because they have two key positions for alignment. Specifically, the first key position lies around frame 80 with a rise and the second lies around frame 200 with a Z-shape wave. This indicates that when samples share more local patterns that require more complex warping paths, it becomes difficult for the GTWIDL algorithm to learn dictionaries and align the samples. This is reasonable since such warping paths have more local features and it takes more iterations to provide accurate estimation, especially when we only use a few basis paths in our GTWIDL algorithm.

To further assess the efficacy of dictionary learning, we implement the GTWIDL algorithm on the 'y' class within the two-dimensional LOWER handwriting dataset and engage in an in-depth analysis. In this dataset, each of the six subjects writes 10 samples of the character 'y'. First, we utilize the adaptive strategy for selecting the atom number outlined in Section IV-A to determine the optimal number of dictionary atoms. With the threshold ζ set at 0.7, the results reveal that the ideal number of atoms is 2. This finding suggests the presence of two predominant patterns in the 'y' class, corroborated by observations that misaligned samples from this class typically exhibit two distinct writing styles, as depicted in Fig. 8(a).

Subsequently, with the atom number fixed at 2, we proceed with the execution of the GTWIDL algorithm. The outcomes of the learned dictionary atoms are then compared with various benchmarks, as illustrated in Fig. 8(b)–(e). The standard kSVD method cannot align the samples and the corresponding atoms only show one pattern. The TWI-kSVD method manages to capture both patterns but the learned atoms yield unsmooth results. For the RISLDTW method, the learned mean series of aligned samples, as depicted in the left subfigure in Fig. 8(d), is affected by two patterns and represents a compromise between them. The right two subfigures show the eigenvectors corresponding to the first two largest eigenvalues, which are found to be uninterpretable. In contrast, our proposed GTWIDL method captures both patterns by two smooth atoms, as shown in Fig. 8(e). This result suggests the superiority of our method in terms of temporal alignment and dictionary learning. Furthermore, we enhance the visualization by representing the reconstruction error of each pattern on the associated atom through a shadow band. This shadow band comprises multiple axis-aligned ellipses, where the radius of each axis corresponds to the standard error of the reconstruction for that specific dimension. It is evident that the two writing styles exhibit larger variability in the areas encircled in orange, typically located at the start, end, or turning points of the writing. This reconstruction error analysis enhances our understanding and interpretation capabilities regarding the

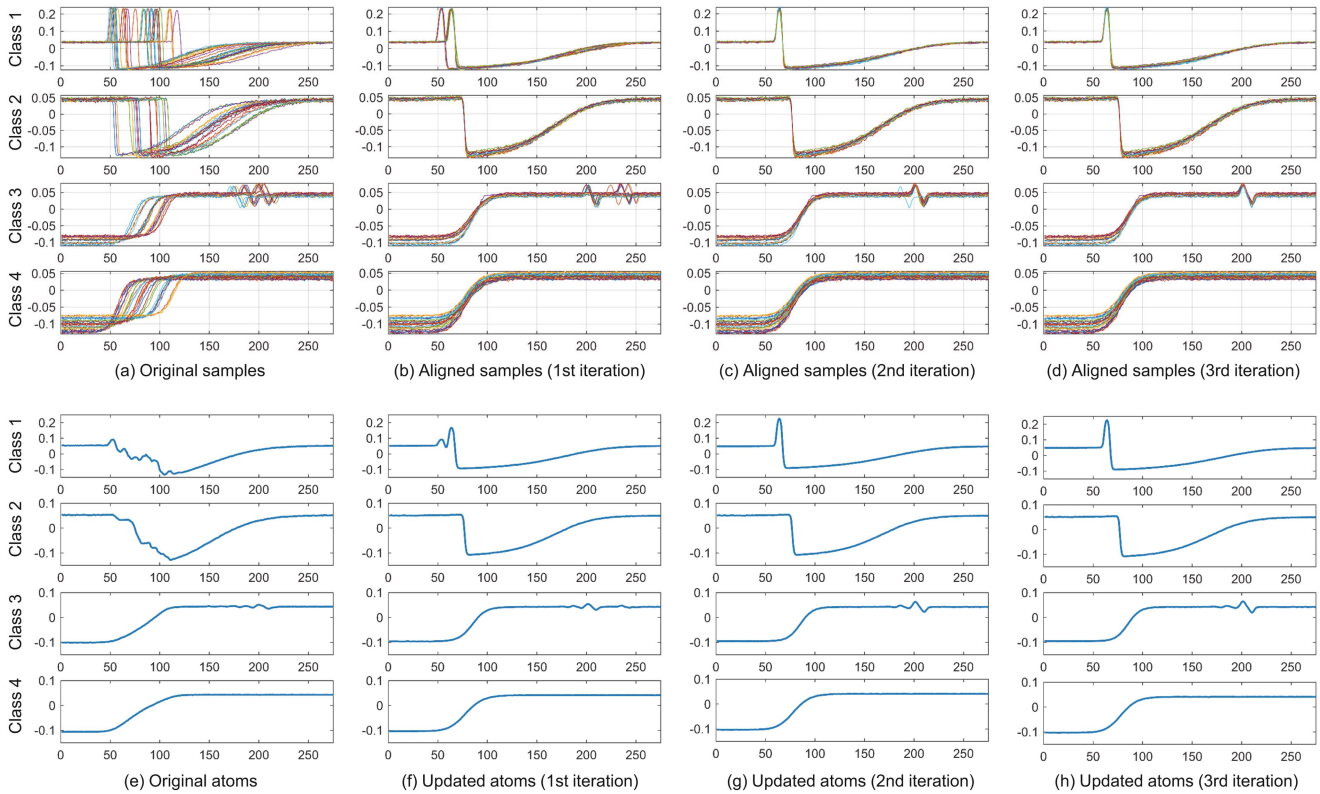


Fig. 7. The top and the bottom rows visualize the aligned samples and the updated atoms for each class of the Trace dataset at the initialization, after the 1st iteration, the 2nd iteration, and the 3 rd iteration, respectively.

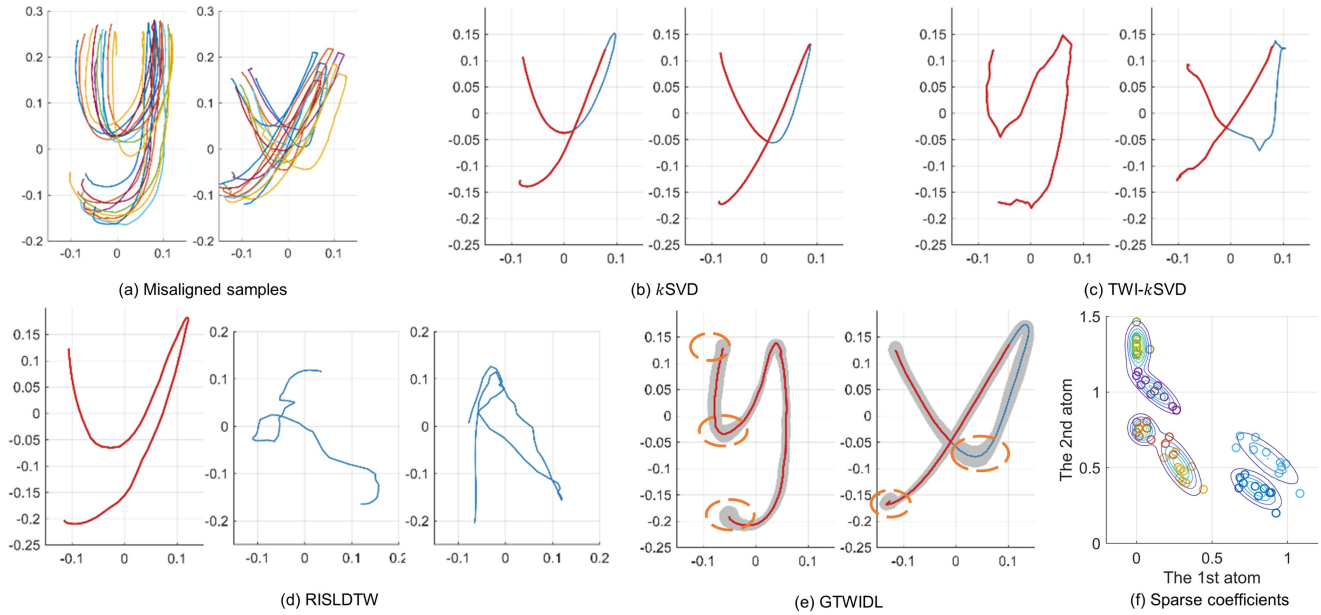


Fig. 8. (a) Misaligned samples from 'y' class in the LOWER dataset. (b-e) Learned atoms via four dictionary learning methods. The red lines of atoms represent the writing rendered on the paper and the blue lines represent the writing rendered in the air or the writing lacking interpretable patterns. (f) Visualization and clustering of the sparse coefficients of GTWIDL.

TABLE IV
CLASSIFICATION ACCURACY COMPARISON

	DTW	DDTW	WDTW	TWI-kSVD	RISLDTW	RISLDTW-ELS	GTWIDL
DIGITS	0.9852	0.9812	0.9860	0.9778	0.9802	0.9788	**0.9976**
LOWER	0.9459	0.9099	0.9473	0.9398	0.9345	0.9411	**0.9888**
UPPER	0.8589	0.8381	0.8630	0.8665	0.9139	0.9101	**0.9885**
char-traj	0.9398	0.8430	**0.9605**	0.9564	0.9416	0.9420	0.9595
UMD	0.9889	0.9979	**1.0000**	0.9653	0.9951	**1.0000**	**1.0000**
BME	0.9560	0.9893	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
ArrowHead	0.7029	0.7771	0.7486	0.7029	0.8286	0.8286	**0.8514**
DSR	**0.9673**	0.9346	**0.9673**	0.9379	0.9248	0.8399	0.9608
FiftyWords	0.6967	0.6989	**0.7714**	0.7297	0.6484	0.6659	0.7429
Trace	**1.0000**	**1.0000**	**1.0000**	0.9900	**1.0000**	**1.0000**	**1.0000**
Wins	2	1	6	1	2	3	**7**
Avg. rank	4.4	4.9	2.0	4.7	4.1	3.5	**1.4**

Best in bold.

TABLE V
CLUSTERING ACCURACY COMPARISON

	SSC	DTW	DDTW	WDTW	TWI-kSVD	RISLDTW	GTWIDL(Init.)	GTWIDL
DIGITS	0.8250	0.9390	0.9450	0.9290	0.9430	0.6580	0.9820	**0.9850**
LOWER	0.6750	0.6769	0.6625	0.7202	0.6702	0.6327	0.7877	**0.8181**
UPPER	0.6300	0.6796	0.7269	0.7385	0.6662	0.5638	0.7723	**0.8292**
char-traj	0.6565	0.8010	0.6810	0.8110	0.7720	0.6240	0.8070	**0.8175**
UMD	0.4416	0.6061	**0.6797**	0.6580	0.5628	0.5758	0.5714	0.6537
BME	0.6474	0.6579	0.7947	0.6579	0.6579	0.6368	0.7000	**0.8211**
ArrowHead	0.5833	0.5556	0.6944	0.5556	0.6111	0.5556	0.6944	**0.7222**
DSR	0.7500	0.9375	0.7500	0.8750	0.8750	0.9375	0.8125	**1.0000**
FiftyWords	0.3156	0.3822	0.4044	0.3689	0.4156	0.3378	0.4533	**0.5089**
Trace	0.5900	0.7500	**0.9700**	0.7500	0.8500	0.7800	0.7800	**0.9700**
Wins	0	0	2	0	0	0	0	**9**
Avg. rank	6.9	4.5	3.7	4.2	4.6	6.4	3.3	**1.2**

Best in bold.

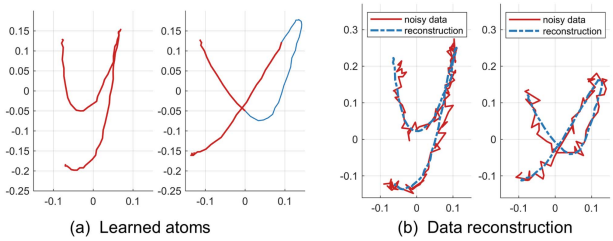


Fig. 9. (a) Atoms learned from the noisy data. (b) Two noisy samples along with the reconstructed data.

data patterns. The sparse coefficients provide a comprehensive insight into the distribution of samples within this class, as shown in Fig. 8(f). By using the Gaussian Mixture Model (GMM) clustering approach, the samples are categorized into six distinct clusters, which effectively mirrors the variation in writing styles among the six subjects, thereby enhancing the depth of information extraction for each sample.

Finally, the proposed method demonstrates robustness in the presence of noise. We introduce Gaussian noise into the dataset. The reconstructed data and the learned dictionary atoms are shown in Fig. 9. Clearly, the proposed dictionary learning approach can effectively alleviate the noise impacts, leading to a smoother data representation and enhanced pattern recognition.

D. Evaluation on Classification

In this subsection, we compare the classification performance of our proposed GTWIDL algorithm with other benchmark methods mentioned in Section V-B. The results are presented in Table IV with the best performance highlighted in bold. The two last rows give, over all the datasets, the number of times a method reaches the best value as well as its average performance rank. It can be observed that our proposed classification method achieves the best overall performance, with a total number of 7 wins and an average ranking of 1.4.

From the results presented in Table IV, we draw the following conclusions. First, our proposed GTWIDL algorithm outperforms other state-of-the-art dictionary learning methods, such as TWI-kSVD, RISLDTW, and RISLDTW-ELS, on all datasets. This is consistent with the analysis presented in Section V-C, showing that our algorithm achieves better performance in both temporal alignment and dictionary learning. Second, the proposed GTWIDL method is also effective in dealing with multi-dimensional time series. Among the four handwritten character datasets, the GTWIDL algorithm achieved three wins and one runner-up. This demonstrates the superior design of the optimization problem given in (4), where the sparse coefficients and warping matrix are shared across all dimensions. Third, the WDTW method also achieved good performance with wins on 6 out of 10 datasets. However, distance-based methods like WDTW cannot provide interpretable patterns like the proposed dictionary-based methods. Moreover, calculating each pair of samples using WDTW is time-consuming, which limits its applicability to large datasets. In contrast, our proposed GTWIDL algorithm offers interpretable and visible atoms to characterize patterns by learning dictionaries, which is beneficial for providing further analysis of each class's patterns. Moreover,

our proposed optimization algorithm is time-efficient for large datasets since the number of atoms is much smaller than the number of samples.

E. Evaluation on Clustering

In this subsection, we conducted a comparison between the proposed clustering algorithm and other benchmarks. The results are presented in Table V with the best performance highlighted in bold. Similar to the classification results, the proposed clustering method achieved the best overall performance with a total number of best values (Wins) of 9 and an average ranking (Avg. rank) of 1.2.

Table V also shows the clustering initialization results of the proposed GTWIDL. Surprisingly, using the proposed pre-clustering step alone without subsequent iterations achieves better clustering accuracy than the other six benchmarks. Further iterative refinements of the clustering results lead to additional enhancements, indicating its superiority and effectiveness. It is worth noting that the DDTW distance-based method also achieves good performance in some cases. This is because the DDTW distance utilizes derivatives as features, focusing on local patterns and thereby making it more robust for the clustering task.

VI. Conclusion

This paper presents a generalized time warping invariant dictionary learning approach for capturing the underlying patterns of time series data subject to temporal warping. Our method can effectively handle multi-dimensional datasets with different sequence lengths by estimating the warping matrix as a linear combination of basis warping functions. Additionally, we introduce a block coordinate descent-based optimization algorithm that jointly solves warping paths, dictionaries, and sparse coding coefficients, thereby improving time efficiency for large datasets. We then employ our proposed dictionary learning method to develop both classification and clustering algorithms, whereby samples are assigned to the class whose dictionary yields the minimum reconstruction error. Our experiments demonstrate that our proposed GTWIDL method provides interpretable atoms, which outperforms other dictionary learning methods in terms of representative and discriminative ability. Furthermore, our method outperforms other state-of-the-art classification and clustering benchmarks, which highlights its superior performance in various applications.

One advantage of the proposed GTWIDL is that it can enhance the understanding of data patterns and distributions through further analysis of the learned dictionary, sparse coefficients, and warping paths. For example, by using the information criterion methods or SVD, we are able to identify the number of data patterns, which offers insights into the dataset complexity. Besides, we can gain an intuitive understanding of the specific forms of data patterns by visualizing the learned dictionary atoms. Another advantage of the proposed dictionary learning approach is that it can improve the accuracy of subsequent tasks involving misaligned data, such as denoising, classification, and clustering. For instance, if the original data contains noise, GTWIDL enables the learning of noise-robust data patterns, reducing the impact of noise in data reconstruction.

There are several interesting topics for further investigation based on this work. First, the current classification strategy only considers the reconstruction error on the local dictionary. Improved results may be achieved by proposing a classification strategy that takes both reconstruction error and sparse coefficients into account. Second, since data can be aligned with sub-parts of atoms, the global data patterns could be recognized with few local observations at the beginning of a process. These patterns can be subsequently used for data prediction or system monitoring in the following data-collecting process. Finally, dictionary learning is widely used in image and video processing. The proposed one-directional time warping invariant dictionary learning has the potential to be extended to a generalized approach focusing on two-directional or three-directional spatio-temporal warping invariance.

Acknowledgments

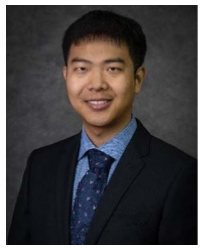
The authors would like to thank the editors and reviewers for thoughtful feedback, and thank the contributors for creating, cleaning and curating the open datasets used in this study.

References

[1] T. Liu, Y. Si, D. Wen, M. Zang, and L. Lang, "Dictionary learning for VQ feature extraction in ECG beats classification," *Expert Syst. Appl.*, vol. 53, pp. 129–137, 2016.
[2] J. Caballero, A. N. Price, D. Rueckert, and J. V. Hajnal, "Dictionary learning and time sparsity for dynamic MR data reconstruction," *IEEE Trans. Med. Imag.*, vol. 33, no. 4, pp. 979–994, Apr. 2014.
[3] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 457–464.
[4] X. Chen, Z. Du, J. Li, X. Li, and H. Zhang, "Compressed sensing based on dictionary learning for extracting impulse components," *Signal Process.*, vol. 96, pp. 94–109, 2014.
[5] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *Int. J. Comput. Vis.*, vol. 109, pp. 209–232, 2014.
[6] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. 2010 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3501–3508.
[7] J. Aach and G. M. Church, "Aligning gene expression time series with time warping algorithms," *Bioinformatics*, vol. 17, no. 6, pp. 495–508, 2001.
[8] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. 3rd Int. Conf. Knowl. Discov. Data Mining*, Seattle, WA, USA, 1994, pp. 359–370.
[9] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. 7th Int. Conf. Artif. Neural Netw.*, Springer, 2005, pp. 583–588.
[10] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Kernel dictionary learning," in *Proc. 2012 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2012, pp. 2021–2024.
[11] J. Yin, Z. Liu, Z. Jin, and W. Yang, "Kernel sparse representation based classification," *Neurocomputing*, vol. 77, no. 1, pp. 120–128, 2012.
[12] Z. Chen, W. Zuo, Q. Hu, and L. Lin, "Kernel sparse representation for time series classification," *Inf. Sci.*, vol. 292, pp. 15–26, 2015.
[13] L. Zhang et al., "Kernel sparse representation-based classifier," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1684–1695, Apr. 2012.
[14] H. Deng, W. Chen, Q. Shen, A. J. Ma, P. C. Yuen, and G. Feng, "Invariant subspace learning for time series data based on dynamic time warping distance," *Pattern Recognit.*, vol. 102, 2020, Art. no. 107210.

[15] M. Lewicki and T. J. Sejnowski, "Coding time-varying signals using sparse, shift-invariant representations," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1998, pp. 730–736.
[16] B. Mailhé, S. Lesage, R. Gribonval, F. Bimbot, and P. Vandergheynst, "Shift-invariant dictionary learning for sparse representations: Extending K-SVD," in *Proc. 16th Eur. Signal Process. Conf.*, 2008, pp. 1–5.
[17] B. Yang, R. Liu, and X. Chen, "Fault diagnosis for a wind turbine generator bearing via sparse representation and shift-invariant K-SVD," *IEEE Trans. Ind. Inform.*, vol. 13, no. 3, pp. 1321–1331, Jun. 2017.
[18] G. Zheng, Y. Yang, and J. Carbonell, "Efficient shift-invariant dictionary learning," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 2095–2104.
[19] G. Li, Z. He, J. Tang, J. Deng, X. Liu, and H. Zhu, "Dictionary learning and shift-invariant sparse coding denoising for controlled-source electromagnetic data combined with complementary ensemble empirical mode decompositioncsem data dictionary learning denoising," *Geophysics*, vol. 86, no. 3, pp. E185–E198, 2021.
[20] H. Deng, W. Chen, A. J. Ma, Q. Shen, P. C. Yuen, and G. Feng, "Robust shapelets learning: Transform-invariant prototypes," in *Proc. 1st Chin. Conf. Pattern Recognit. Comput. Vis.*, Springer, 2018, pp. 491–502.
[21] S. V. Yazdi and A. Douzal-Chouakria, "Time warp invariant kSVD: Sparse coding and dictionary learning for time series under time warp," *Pattern Recognit. Lett.*, vol. 112, pp. 1–8, 2018.
[22] S. V. Yazdi, A. Douzal-Chouakria, P. Gallinari, and M. Moussallam, "Time warp invariant dictionary learning for time series clustering: Application to music data stream analysis," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Springer, 2019, pp. 356–372.
[23] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *Proc. 2001 SIAM Int. Conf. Data Mining*, 2011, pp. 1–11.
[24] F. Zhou and F. De la Torre, "Generalized time warping for multi-modal alignment of human motion," in *Proc. 2012 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1282–1289.
[25] F. Zhou and F. De la Torre, "Generalized canonical time warping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 279–294, Feb. 2016.
[26] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.
[27] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
[28] S. Chu, E. Keogh, D. Hart, and M. Pazzani, "Iterative deepening dynamic time warping for time series," in *Proc. 2002 SIAM Int. Conf. Data Mining*, SIAM, 2002, pp. 195–212.
[29] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognit.*, vol. 44, no. 9, pp. 2231–2240, 2011.
[30] Y. Xie and B. Wiltgen, "Adaptive feature based dynamic time warping," *Int. J. Comput. Sci. Netw. Secur.*, vol. 10, no. 1, pp. 264–273, 2010.
[31] J. Zhao and L. Itti, "shapeDTW: Shape dynamic time warping," *Pattern Recognit.*, vol. 74, pp. 171–184, 2018.
[32] R. J. Kate, "Using dynamic time warping distances as features for improved time series classification," *Data Mining Knowl. Discov.*, vol. 30, pp. 283–312, 2016.
[33] H. Du et al., "Dynamic time warping and spectral clustering based fault detection and diagnosis of railway point machines," in *Proc. 2019 IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 595–600.
[34] J. Lohrer and M. Lienkamp, "Building representative velocity profiles using FastDTW and spectral clustering," in *Proc. 14th Int. Conf. ITS Telecommun.*, 2015, pp. 45–49.
[35] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh, "Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm," *Knowl. Inf. Syst.*, vol. 47, pp. 1–26, 2016.
[36] S. Soheily-Khah, A. Douzal-Chouakria, and E. Gaussier, "Generalized K-means-based clustering for temporal data under weighted and kernel time warp," *Pattern Recognit. Lett.*, vol. 75, pp. 63–69, 2016.
[37] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognit.*, vol. 44, no. 3, pp. 678–693, 2011.
[38] H. Izakian, W. Pedrycz, and I. Jamal, "Fuzzy clustering of time series data using dynamic time warping distance," *Eng. Appl. Artif. Intell.*, vol. 39, pp. 235–244, 2015.
[39] L. Hou, J. Kwok, and J. Zurada, "Efficient learning of timeseries shapelets," in *Proc. AAAI Conf. Artif. Intell.*, 2016, Art. no. 1.
[40] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*. Berlin, Germany: Springer, 2005.
[41] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[42] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proc. 2008 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[43] H. A. Dau et al., "The UCR time series classification archive," Oct. 2018. [Online]. Available: https://www.cs.ucr.edu/eamonn/time_series_data_2018/
[44] M. Chen, G. AlRegib, and B.-H. Juang, "6DMG: A new 6D motion gesture database," in *Proc. 3rd Multimedia Syst. Conf.*, 2012, pp. 83–88.
[45] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

Ruiyu Xu received the BE degree from the Department of Energy and Resources Engineering, Peking University, Beijing, China, in 2019. He is currently working toward the PhD degree with the Department of Industrial Engineering and Management, Peking University. His research interests include time series analysis, deep learning, quality and reliability engineering, and advanced manufacturing.



Chao Wang received the BS degree in mechanical engineering from the Hefei University of Technology, in 2012, the MS degree in mechanical engineering from the University of Science and Technology of China, in 2015, and the MS degree in statistics and the PhD degree in industrial and systems engineering from the University of Wisconsin-Madison, in 2018 and 2019, respectively. He is an assistant professor with the Department of Industrial and Systems Engineering, University of Iowa. His research interests include statistical modeling, analysis, monitoring and control for complex systems. He is member of INFORMS, IISE, and SME.

Yongxiang Li received the PhD degree in data science from the City University of Hong Kong, in 2019. Currently, he is an associate professor with the Department of Industrial Engineering and Management, Shanghai Jiao Tong University, Shanghai, China. His research focuses on both the theoretical and applied aspects of data science integrated with domain knowledge for quality and reliability engineering using methodologies from statistics, machine learning, and signal processing. He has been working on the research directions, such as computer experiments, quality control, anomaly detection, and fault diagnostics.

Jianguo Wu received the BS degree in mechanical engineering from Tsinghua University, China, in 2009, the MS degree in mechanical engineering from Purdue University, in 2011, and the MS degree in statistics and the PhD degree in industrial and systems engineering from the University of Wisconsin-Madison, in 2014 and 2015. Currently, he is an associate professor with the Department of Industrial Engineering and Management, Peking University, Beijing, China. He was an assistant professor with the Department of IMSE, UTEP, TX, USA from 2015 to 2017. His research interests are mainly in quality control and reliability engineering of intelligent manufacturing and complex systems through engineering-informed machine learning and advanced data analytics. He is a recipient of the STARS Award from the University of Texas Systems, Overseas Distinguished Young Scholars from China, P & G Faculty Fellowship, BOSS Award from MSEC, and several Best Paper Award/Finalists from INFORMS/IISE Annual Meeting. He is an associate editor of the *Journal of Intelligent Manufacturing* and *IISE Transactions*, and a member of the INFORMS, IISE, and SME.