

Received November 9, 2020, accepted November 20, 2020, date of publication December 3, 2020, date of current version December 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3042217

Deep Multiview Learning From Sequentially Unaligned Data

DOAN PHONG TUNG¹ AND ATSUHIRO TAKASU^{1,2}, (Member, IEEE)

¹Graduate University for Advanced Studies, SOKENDAI, Kanagawa 240-0193, Japan

²National Institute of Informatics, Tokyo 101-8430, Japan

Corresponding author: Doan Phong Tung (tungdp@nii.ac.jp)

This work was supported by the “Cross-ministerial Strategic Innovation Promotion Program (SIP) Second Phase, Big-data and AI-enabled Cyberspace Technologies” through the New Energy and Industrial Technology Development Organization (NEDO).

ABSTRACT Multiview learning is concerned with machine learning problems, where data are represented by distinct feature sets or views. Recently, this definition has been extended to accommodate sequential data, i.e., each view of the data is in the form of a sequence. Multiview sequential data pose major challenges for representation learning, including i) *absence of sample correspondence information between the views*, ii) *complex relations among samples within each view*, and iii) *high complexity for handling multiple sequences*. In this article, we first introduce a generalized deep learning model that can simultaneously discover sample correspondence and capture the cross-view relations among the data sequences. The model parameters can be optimized using a gradient descent-based algorithm. The complexity for computing the gradient is at most quadratic with respect to sequence lengths in terms of both computational time and space. Based on this model, we propose a second model by integrating the objective with reconstruction losses of autoencoders. This allows the second model to provide a better trade-off between view-specific and cross-view relations in the data. Finally, to handle multiple (more than two) data sequences, we develop a third model along with a convergence-guaranteed optimization algorithm. Extensive experiments on public datasets demonstrate the superior performances of our models over competing methods.


INDEX TERMS Multiview learning, dynamic time warping, smooth approximation, deep learning, sequential data.

I. INTRODUCTION

In many real-world applications, data are often collected from various perspectives, each of which presents a view of the same data and has its own representation space and relation characteristics. Multiview learning methods aim to exploit consistency and complementary information between these views to learn new representations for the data. Therefore, these methods often have better generalization ability. Recently, the definition of multiview learning has been extended to accommodate sequential data, i.e., each view of the data is in the form of a sequence. For instance, human actions can be presented by several video sequences with different features, such as binary, Euclidean distance transform, and Poisson equation solutions [55] (see Figure 8).

Multiview sequential learning has posed major challenges that are difficult for conventional methods to accommodate. First, most multiview learning methods essentially rely on an assumption that all views of the data are equal in size and

sample-wise matching. Here, we take canonical correlation analysis (CCA) [1] and its variants [2]–[4] as representatives. These methods project data samples from two different views into a shared subspace and then minimize the squared difference between the projections subject to whitening constraints. Thus, the two-view training data must have the following form: $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d_x \times n}$ and $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \mathbb{R}^{d_y \times n}$, where $(\mathbf{x}_i, \mathbf{y}_i)$ is a matching pair ($1 \leq i \leq n$). However, these requirements are likely to be violated in sequential settings. For example, sample deletion and/or insertion often occurs when collecting data sequences because of the temporal failures of devices and other man-made reasons. In addition, the asynchronization of data collection devices, e.g., sensors have dissimilar sampling frequencies, also induces misalignment among the collected sequences. A widely used alignment algorithm, dynamic time warping (DTW) [5], can be used to match samples in correspondence as a preprocessing step before performing conventional multiview learning methods. Unfortunately, DTW fails when the dimensions of the two sequences vary ($d_x \neq d_y$). Second, in addition to the ambiguous cross-view relations mentioned, multiview

The associate editor coordinating the review of this manuscript and approving it for publication was Jonghoon Kim .

sequential data also involve complex view-specific relations that span through the length of the sequences. CCA-based methods capture these relations through linear [1] and non-linear [2]–[4] projection functions. However, they ignore the sequential order that naturally exists among samples within each view. Finally, in practice, the input data often comprise more than two sequences. Handling multiple sequential views is a difficult task that certainly involves high resource requirements. In addition, the discriminative properties of the learned representation might be degenerated because of the absence of label information and the presence of irrelevant information from multiple views.

In this article, we first propose generalized sequential correlation analysis (GSCA)—a novel deep neural network (DNN)-based method—to tackle the aforementioned challenges. Our method parameterizes the projection functions that map data sequences into the shared subspace by DNNs. Various types of DNNs can be selected regarding the relations among samples within each view. In this work, we use feed-forward neural networks and recurrent neural networks (RNNs) for implementation. In the shared subspace, our method minimizes the generalized smooth DTW distance between projections of the two views subject to soft whitening constraints. This allows GSCA to discover the sample correspondences and capture the relations between the views simultaneously. Because the generalized smooth DTW is a differentiable approximation of the original DTW, parameters of our model can be optimized in a unified manner using gradient descent-based algorithms. Computing the gradient generally takes a quadratic time and requires a quadratic memory space with respect to (w.r.t.) the sequence lengths. We can further increase the computation speed and reduce the space requirement by selecting squared ℓ_2 norm for regularization, which induces sparsity in the gradient of the generalized smooth DTW. Second, to provide a better balance between view-specific and cross-view relations, we combine our objective function with the reconstruction losses of autoencoders [6]. This forms generalized sequentially correlated autoencoders (GSCAE), which are a new variant of the proposed model. Finally, we further develop the third model called generalized multiple sequences analysis (GMSA) to handle multiple data sequences. Slightly differing from the two first proposed methods, GMSA uses DNNs to map input data sequences directly into the label space. Thereby, we expect that the learned representation can have cluster interpretability and better discriminability. Because no supervised information is given, we introduce a consensus label sequence that is then aligned with projections of all the input sequences. An efficient algorithm with a convergence guarantee is also provided to optimize both the consensus and the DNNs' parameters.

This article includes materials from [7] with significant expansion. First, GSCA and GSCAE are expanded versions of deep sequential correlation analysis (DSCA) and deep sequentially correlated autoencoders (DSCAE), which are proposed in the preliminary work, respectively.

Efficiencies of GSCA and GSCAE are much better than those of DSCA and DSCAE. By taking advantage of the generalized smooth DTW distance, complexities in both terms of time and space for computing the gradients when training GSCA and GSCAE are reduced significantly.¹ Second, while [7] introduced the two deep models based only on feed-forward neural networks, in this article, we further extend the model concept to RNNs, i.e., long short-term memory (LSTM) [8], to better capture the sequential relation within each view. Finally, we propose the GMSA model to handle multiple data sequences simultaneously, which was not considered in [7], and learn a more interpretable representation. Experiments using both two- and multiple-view datasets were designed carefully to provide a fair comparison with existing competitors. In summary, the contributions of this article are as follows:

- Introduces a novel deep multiview model that can discover sample correspondence implicitly while learning the representation from sequential data.
- Extends the proposed model based on the reconstruction loss regularization of autoencoders, which allows a trade-off between information within each view and information in the correlation across the views.
- Derives a third model that can handle multiple (more than two) data sequences and learn more interpretable and discriminative representation.
- Extensive experiments on various public datasets demonstrate the superior performances of the proposed models over existing methods.

The remainder of this article is organized as follows: Section II briefly presents some background for the methods proposed in this article. The GSCA model and its autoencoder-based variant are introduced in Sections III and IV, respectively. The third model for handling multiple data sequences along with a convergence-guaranteed optimization algorithm is described in Section V. After reviewing the related works in Section VI, we present the experimental results in Section VII. Section VIII concludes this article.

Notations. Throughout this article, scalars, vectors, and matrices are denoted by lower-case, bold lower-case, and bold uppercase letters, respectively. An element at position (i, j) of a matrix \mathbf{A} is denoted by $a_{i,j}$ or $[\mathbf{A}]_{i,j}$. We denote the Frobenius inner product between \mathbf{A} and \mathbf{B} as $\langle \mathbf{A}, \mathbf{B} \rangle := \sum_{i,j} a_{i,j} b_{i,j}$. $\mathbf{0}_d$ is a vector of dimension d whose all elements are zeros. The expression $\mathbf{x} \in \mathbb{R}_+^d$ indicates that vector \mathbf{x} has d elements, each of which is greater than or equal to zero. The norm ℓ_p of a vector \mathbf{x} , where $p \in \{1, 2\}$ in this article, is $\|\mathbf{x}\|_p = (|x_1|^p + \dots + |x_d|^p)^{\frac{1}{p}}$.

II. BACKGROUND

A. DYNAMIC TIME WARPING

The DTW [5] algorithm measures the similarity between two sequences whose lengths are possibly different and whose sample correspondences are probably unknown. Given two

¹More discussions are given in III-C

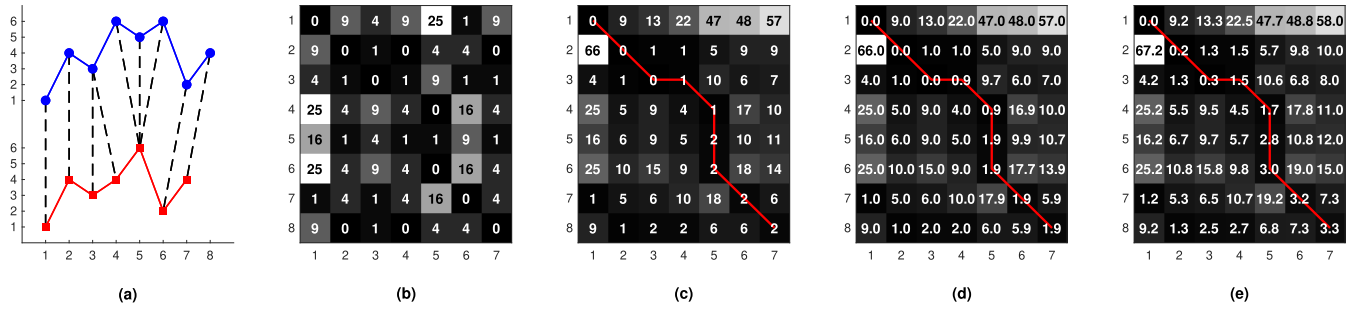


FIGURE 1. A toy example of DTW and its smooth approximation: (a) two example sequences, (b) the distance matrix, (c) the cumulative sum matrix, (d) the cumulative sum matrix of $\text{DTW}_{\Omega=\text{entropy}}$ and (e) the cumulative sum matrix of $\text{DTW}_{\Omega=\text{squared } \ell_2}$. The red line depicts the optimal warping path, which encodes the sample correspondences between the two sequences.

sequences $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ and $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m] \in \mathbb{R}^{d \times m}$, a distance matrix $\mathbf{D}(\mathbf{X}, \mathbf{Y}) \in \mathbb{R}^{n \times m}$ is defined such that the element at position (i, j) , denoted by $d_{i,j}$, is the squared distance, i.e., $d_{i,j} = \|\mathbf{x}_i - \mathbf{y}_j\|_2^2$. The DTW algorithm constructs a cumulative sum matrix $\mathbf{S}(\mathbf{X}, \mathbf{Y})$ using the following recursive formulas:

$$s_{1,1} = d_{1,1} \tag{1}$$

$$s_{i,j} = d_{i,j} + \min(s_{i-1,j}, s_{i,j-1}, s_{i-1,j-1}), \tag{2}$$

The DTW distance between the two sequences is then defined as $\text{DTW}(\mathbf{X}, \mathbf{Y}) := s_{n,m}$. By backtracking from the last element $s_{n,m}$ to the start element $s_{1,1}$, an optimal warping path

$$\boldsymbol{\pi}^* = \langle (i_1^*, j_1^*), \dots, (i_p^*, j_p^*) \rangle, \tag{3}$$

that satisfies: i) *boundary condition*: $(i_1, j_1) = (1, 1)$ and $(i_p, j_p) = (n, m)$; ii) *continuous condition*: $(i_{r+1} - i_r, j_{r+1} - j_r) \in \{(0, 1), (1, 0), (1, 1)\}$, where $1 \leq r \leq p - 1$; and iii) *monotonic condition*: if $1 \leq r \leq t \leq p$, then $i_r \leq i_t$ and $j_r \leq j_t$ is formed. This path has the smallest cumulative sum $s_{n,m} = d_{i_1^*, j_1^*} + \dots + d_{i_p^*, j_p^*}$ and encodes the sample correspondences between the two sequences. A toy example of DTW is shown in Figure 1.

B. GENERALIZED SMOOTH DTW

The optimal warping path can be discovered by minimizing DTW; however, original DTW is not differentiable because of the nonsmoothness of min operator in equation (2), which makes it difficult to minimize using gradient-based methods. To alleviate this issue, [9], [10] studied a smooth min operator that serves as an essential basis to develop the differentiable approximations of DTW.

Let $\boldsymbol{\eta} = [\eta_1, \dots, \eta_k]^T \in \mathbb{R}^k$, the smooth min operator is defined as follows:

$$\min_{\Omega}(\boldsymbol{\eta}) := \min_{\boldsymbol{\gamma} \in \Delta^k} \langle \boldsymbol{\gamma}, \boldsymbol{\eta} \rangle + \frac{1}{\beta} \Omega(\boldsymbol{\gamma}), \tag{4}$$

where $\Delta^k := \{\boldsymbol{\gamma} \in \mathbb{R}_+^k : \|\boldsymbol{\gamma}\|_1 = 1\}$ is a $(k - 1)$ unit simplex, $\langle \cdot, \cdot \rangle$ denotes an inner product, Ω is a strictly convex function on Δ^k , and β is a nonnegative regularization parameter. Because (4) is strictly convex, its minimum is unique and

equal to the gradient (based on Danskin’s theorem [11]):

$$\nabla \min_{\Omega}(\boldsymbol{\eta}) = \underset{\boldsymbol{\gamma} \in \Delta^k}{\text{argmin}} \langle \boldsymbol{\gamma}, \boldsymbol{\eta} \rangle + \frac{1}{\beta} \Omega(\boldsymbol{\gamma}). \tag{5}$$

The equation shows that the smooth min operator also depends on the selection of the regularization function $\Omega(\boldsymbol{\gamma})$. Shannon entropy ($\sum_{i=1}^k \gamma_i \ln \gamma_i$) or squared ℓ_2 norm ($\frac{1}{2} \sum_{i=1}^k \gamma_i^2$) are often chosen. While the former induces closed-form solutions for both smooth min and its gradient, the latter forces the gradient to be sparse. More details are given in Appendix A.

As the definition of the smooth min operator is already given, we can arrive at the following recursive formulation:

$$\begin{aligned} s'_{1,1} &= d_{1,1} \\ s'_{i,j} &= d_{i,j} + \min_{\Omega}(s'_{i-1,j}, s'_{i,j-1}, s'_{i-1,j-1}), \end{aligned} \tag{6}$$

where the generalized smooth approximation of DTW is defined by $\text{DTW}_{\Omega}(\mathbf{X}, \mathbf{Y}) := s'_{n,m}$. Note that we can have different versions of DTW_{Ω} , e.g., $\text{DTW}_{\Omega=\text{entropy}}$ or $\text{DTW}_{\Omega=\text{squared } \ell_2}$, depending on selection of the regularization $\Omega(\boldsymbol{\gamma})$. Figure 1 (d) and (e) show the cumulative sum matrices of $\text{DTW}_{\Omega=\text{entropy}}$ and $\text{DTW}_{\Omega=\text{squared } \ell_2}$, respectively. The generalized smooth DTW distance is different from the original DTW because it is differentiable. Furthermore, by minimizing DTW_{Ω} , the optimal warping path is discovered implicitly instead of specified directly, as in the original DTW.

III. GENERALIZED SEQUENTIAL CORRELATION ANALYSIS

In this section, we propose *Generalized sequential correlation analysis* (GSCA) for learning representation from two-view sequential data. We first present the model and its objective function. We then describe the optimization algorithm and compare the proposed method with DSCA, which was proposed in our preliminary work [7].

Given two data sequences $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d_x \times n}$ and $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m] \in \mathbb{R}^{d_y \times m}$ from different representation spaces ($d_x \neq d_y$), our method maps them into a shared subspace: $\mathbf{Z}^X = [\mathbf{z}_1^x, \dots, \mathbf{z}_n^x] = f_x(\mathbf{X}, \boldsymbol{\theta}_x) \in \mathbb{R}^{d \times n}$ and $\mathbf{Z}^Y = [\mathbf{z}_1^y, \dots, \mathbf{z}_m^y] = f_y(\mathbf{Y}, \boldsymbol{\theta}_y) \in \mathbb{R}^{d \times m}$, where $f_x(\cdot, \cdot)$ and $f_y(\cdot, \cdot)$ are projection functions and $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_y$ denotes their parameters.

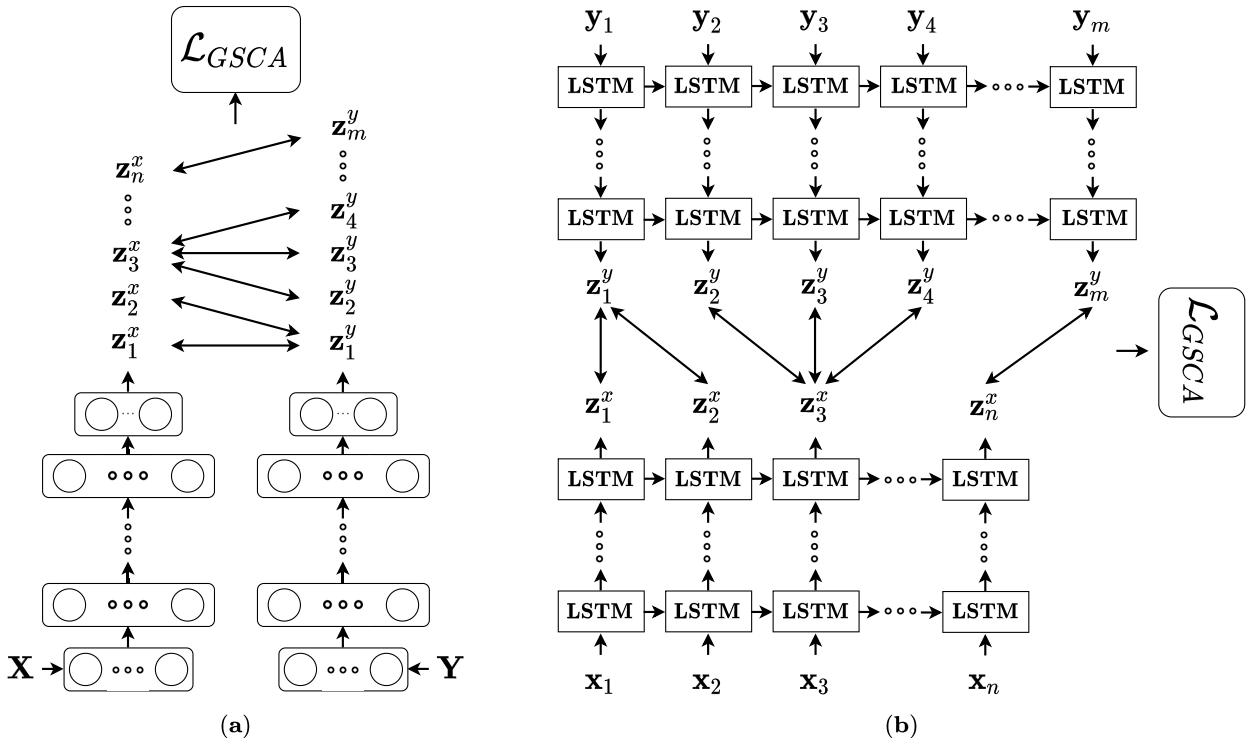


FIGURE 2. Diagrams of GSCA, where the projection functions are parameterized by (a) deep feed-forward neural networks or (b) deep RNNs (unfolded deep LSTM networks are shown). The symbol \leftrightarrow denotes the sample correspondences that are discovered implicitly by minimizing the objective \mathcal{L}_{GSCA} . Note that each deep network includes a batch normalization (BN) layer at the output, which is not shown in the diagrams

The projection functions are parameterized by deep feed-forward neural networks or RNNs. If the former is selected, each sample x_i of the sequence X is passed through several fully connected feed-forward layers to compute the output z_i^x . These outputs are then assembled into columns of the matrix Z^x following the increasing order of the index i . The second view is processed in the same manner. Note that we use batch normalization (BN) [12] as the final layer. Thus, the output features empirically have zero mean and unit variance. For the latter, we stack several LSTM units to form two deep LSTM networks. Each network is also equipped with a BN layer to perform the normalization. The representation sequences Z^x and Z^y are then computed by feeding the input sequence X and Y , respectively, to the networks. Note that the parameters θ_x and θ_y are now equivalent to collections of all the weights matrices of the corresponding DNNs. Figure 2 shows the diagrams of the proposed method.

A. OBJECTIVE

Because the input sequences are unaligned, the sample-wise correspondence information between their representations Z^x and Z^y is also absent. Our method aims at minimizing the generalized smooth DTW distance between Z^x and Z^y . This allows the method to implicitly discover the optimal warping path, which encodes the sample correspondences as mentioned in Section II. In addition, the squared distances between the corresponding representation samples from the

two views are also reduced simultaneously, pulling them closer in the shared subspace. The objective function of our method is as follows:

$$\mathcal{L}_{GSCA} = DTW_{\Omega}(Z^x, Z^y) + \lambda_1 \mathcal{L}_x(Z^x) + \lambda_2 \mathcal{L}_y(Z^y), \quad (7)$$

where the two regularization terms are of the following form:

$$\mathcal{L}_v(Z^v) = \sum_{i=1}^d \sum_{j \neq i}^d |c_{i,j}^v|, \quad (8)$$

where $v \in \{x, y\}$ and $c_{i,j}^v$ is the element at the (i, j) position of the matrix $C^v = Z^v Z^{v\top}$. These regularization functions are smooth approximations of the whitening constraints in CCA-based methods. More specifically, the whitening constraints enforce the features of the representations to be pairwise uncorrelated ($C^v = I$). They are used to prevent trivial solutions, e.g., all the data samples are mapped into a single point in the shared subspace. In our method, because the representation sequences are normalized by BN layers, we further use the l_1 -norm to encourage sparsity in the off-diagonal elements of C^v . $\lambda_1 > 0$ and $\lambda_2 > 0$ are regularization parameters that control the trade-off between whitening and warping the two representation sequences.

B. OPTIMIZATION

The parameters θ_x and θ_y can be trained using the gradient-based method. To compute the gradient of \mathcal{L}_{GSCA} w.r.t. all

the parameters θ_x and θ_y , we compute its gradients w.r.t. the outputs \mathbf{Z}^x and \mathbf{Z}^y and then use backpropagation [13] in the case of feed-forward neural networks or backpropagation through time (BTT) [14] if the RNNs are used. We have

$$\frac{\partial \mathcal{L}_{GSCA}}{\partial \mathbf{Z}^x} = \frac{\partial \text{DTW}_{\Omega}(\mathbf{Z}^x, \mathbf{Z}^y)}{\partial \mathbf{Z}^x} + \lambda_1 \frac{\partial \mathcal{L}_x(\mathbf{Z}^x)}{\partial \mathbf{Z}^x}. \quad (9)$$

The gradient of the generalized smooth DTW w.r.t. \mathbf{Z}^x can be computed as

$$\frac{\partial \text{DTW}_{\Omega}(\mathbf{Z}^x, \mathbf{Z}^y)}{\partial \mathbf{Z}^x} = \left[\frac{\partial s'_{n,m}}{\partial z_1^x}, \dots, \frac{\partial s'_{n,m}}{\partial z_n^x} \right], \quad (10)$$

where

$$\begin{aligned} \frac{\partial s'_{n,m}}{\partial z_i^x} &= \sum_{j=1}^m \frac{\partial s'_{n,m}}{\partial d_{i,j}} \frac{\partial d_{i,j}}{\partial z_i^x} \\ &= 2 \sum_{j=1}^m e_{i,j} (z_i^x - z_j^y) \quad \text{for } i = 1, \dots, n. \end{aligned} \quad (11)$$

In equation (12), we abused the notations defined in Section II, where $s'_{n,m} := \text{DTW}_{\Omega}(\mathbf{Z}^x, \mathbf{Z}^y)$ and $d_{i,j} := \|z_i^x - z_j^y\|_2^2$. The derivative $e_{i,j} = \frac{\partial s'_{n,m}}{\partial d_{i,j}}$ can be computed efficiently using a forward-backward algorithm. The details of the algorithm and its complexity are given in Appendix C. The gradient of \mathcal{L}_x w.r.t. \mathbf{Z}^x can be computed as

$$\frac{\partial \mathcal{L}_x(\mathbf{Z}^x)}{\partial \mathbf{Z}^x} = \mathbf{H}^x \mathbf{Z}^x, \quad (13)$$

where $\mathbf{H}^x \in \mathbb{R}^{d \times d}$, whose elements are defined as

$$h_{i,j}^x = \begin{cases} 1 & \text{if } c_{i,j}^x > 0 \\ 0 & \text{if } i = j \text{ or } c_{i,j}^x = 0 \\ -1 & \text{if } c_{i,j}^x < 0. \end{cases} \quad (14)$$

The gradient $\frac{\partial \mathcal{L}_{GSCA}}{\partial \mathbf{Z}^x}$ can be computed in a similar manner. Our model can be trained using a full-batch algorithm (L-BFGS) [15], as in [2]. For large datasets, however, this algorithm is both time and memory inefficient. An alternative is based on stochastic gradient descent (SGD) [16], [17] where the gradient is estimated based on a much smaller number of training samples (a minibatch). The details are shown in Algorithm 1. Note that we use a stochastic estimate of the covariance matrix for each view because at each iteration, t , the algorithm can access only a small number of samples instead of the whole training set.

C. COMPARISON WITH DSCA

GSCA is expanded on DSCA, which was proposed in our preliminary work [7]. Because of the exploitation of the generalized smooth DTW distance, GSCA is more generalized and efficient than DSCA. More specifically, depending on the selection of the regularization function $\Omega(\mathbf{y})$ in DTW_{Ω} , we have different versions of the proposed method. We denote the version with the Shannon entropy as GSCA-e and the other where the squared ℓ_2 norm is selected as GSCA-s. In fact, the objective of GSCA-e is equivalent

Algorithm 1 Stochastic Algorithm for GSCA

Input: Batch size ratio $\alpha \in [0, 1]$, time constant $\rho \in [0, 1]$, momentum $\mu \in [0, 1]$, and learning rate ϵ .

Output: Optimal DNNs parameters $\theta^* = [\theta_x^*, \theta_y^*]$.

- 1: **for** $t = 1, \dots, T$ **do**
- 2: random sample subsequence $\mathbf{Z}_{(t)}^x$ of length $n\alpha$;
- 3: random sample subsequence $\mathbf{Z}_{(t)}^y$ of length $m\alpha$;
- 4: $\mathbf{C}_{(t)}^x = \rho \mathbf{C}_{(t-1)}^x + (1 - \rho) \frac{1}{\alpha} \mathbf{Z}_{(t)}^x \mathbf{Z}_{(t)}^{x \top}$;
- 5: $\mathbf{C}_{(t)}^y = \rho \mathbf{C}_{(t-1)}^y + (1 - \rho) \frac{1}{\alpha} \mathbf{Z}_{(t)}^y \mathbf{Z}_{(t)}^{y \top}$;
- 6: compute $\frac{\partial \mathcal{L}_{GSCA}}{\partial \mathbf{Z}_{(t)}^x}$ and $\frac{\partial \mathcal{L}_{GSCA}}{\partial \mathbf{Z}_{(t)}^y}$;
- 7: compute gradient ∇_{θ} using backpropagation;
- 8: $\Delta \theta_{(t)} = \mu \Delta \theta_{(t-1)} - \epsilon \nabla_{\theta}$;
- 9: $\theta_{(t)} = \theta_{(t-1)} + \Delta \theta_{(t)}$;
- 10: **end for**

to that of DSCA because $\text{DTW}_{\Omega=\text{entropy}}$ is equal to DTW_{β} , i.e., a smooth approximation of DTW employed in the preliminary work. The proof is given in Appendix B. Despite that, computing the gradient of DTW_{Ω} w.r.t. the representation sequences is more efficient because its complexity is only $O(nm)$ in terms of both time and space. In contrast, the gradients of DTW_{β} w.r.t. \mathbf{Z}^x and \mathbf{Z}^y associate with a summation over all feasible alignment matrices² requires $O(n^2 m^2)$ in both computational time and memory storage. Note that by selecting squared ℓ_2 norm as the regularization, the complexity of GSCA can be reduced further because of the sparsity of the gradient. However, this advantage possibly comes at a cost of lower alignment accuracy because $\text{DTW}_{\Omega=\text{squared } \ell_2}$ is a nonexact approximation of the original DTW [10]. $\text{DTW}_{\Omega=\text{entropy}}$, instead, can exactly approximate DTW. It converges to the original warping distance as $\beta \rightarrow \infty$.

IV. GENERALIZED SEQUENTIALLY CORRELATED AUTOENCODERS

In this section, we develop GSCAE as a variant of the proposed method. The objective of GSCAE is formed by integrating reconstruction losses of autoencoders with the objective of GSCA. Let $g_x(\mathbf{Z}^x, \Phi_x)$ and $g_y(\mathbf{Z}^y, \Phi_y)$ denote the functions that map the representation sequences \mathbf{Z}^x and \mathbf{Z}^y back to the original spaces, where Φ_x and Φ_y are their corresponding parameters. Then, the objective of GSCAE is as follows

$$\begin{aligned} \mathcal{L}_{GSCAE} &= \mathcal{L}_{GSCA} + \lambda \left(\frac{1}{n} \|\mathbf{X} - g_x(\mathbf{Z}^x, \Phi_x)\|_2^2 \right. \\ &\quad \left. + \frac{1}{m} \|\mathbf{Y} - g_y(\mathbf{Z}^y, \Phi_y)\|_2^2 \right), \end{aligned} \quad (15)$$

where $\lambda > 0$ is a trade-off parameter. Similar to projection functions in GSCA, $g_x(\cdot, \cdot)$ and $g_y(\cdot, \cdot)$ can also be parameterized by deep feed-forward neural networks or RNNs. Diagrams of GSCAE are illustrated in Figure 3.

²Equation (13) in [7]

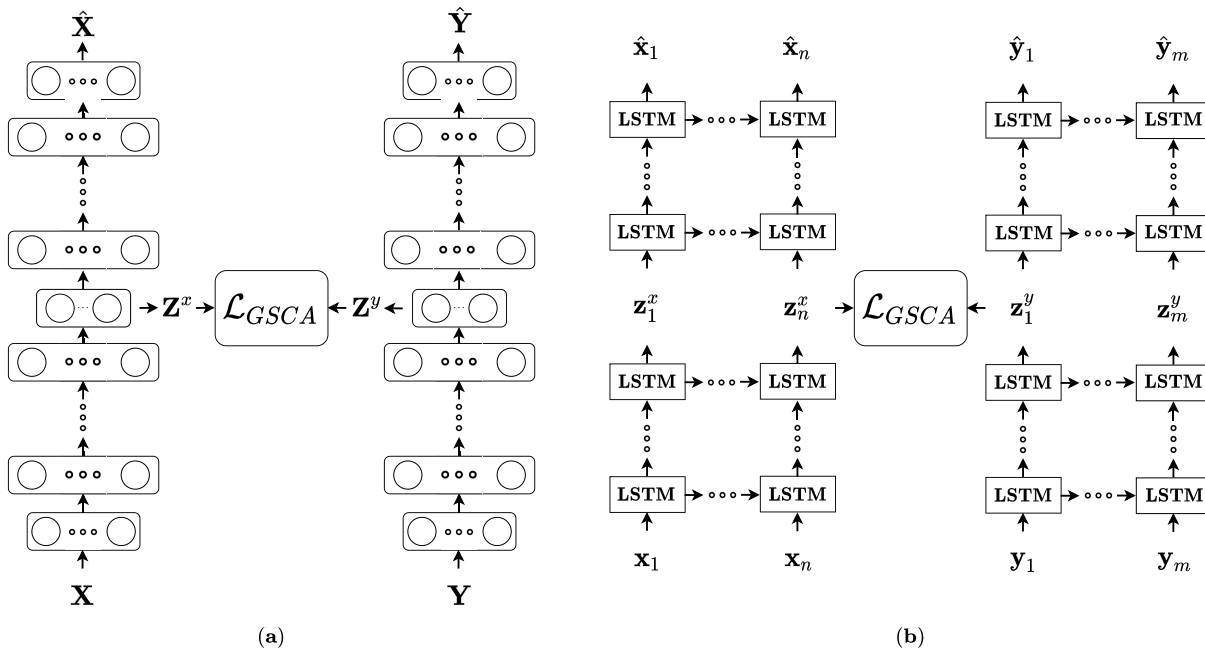


FIGURE 3. Diagrams of *generalized sequentially correlated autoencoders (GSCAE)*, where the projections functions and reconstruction functions are parameterized by (a) deep feed-forward neural networks or (b) deep RNNs. $\hat{X} = [\hat{x}_1, \dots, \hat{x}_N]$ and $\hat{Y} = [\hat{y}_1, \dots, \hat{y}_M]$ denote the reconstructed inputs.

By minimizing \mathcal{L}_{GSCA} , the correspondences of samples between the views are discovered implicitly and their corresponding squared distances are also reduced. This amounts to maximizing mutual information, which presents the relation between the views. The view-specific relation, on the other hand, is expressed via minimizing the reconstruction errors. This is equivalent to maximizing a bound on the mutual information between the input and output of each view. Thus, GSCAE provide us a better trade-off between information within each view and cross-view information.

The advantages of GSCAE over GSCA come at some costs. For a specific application with a particular dataset, we need to carefully tune the trade-off parameter λ for GSCAE to achieve optimal performance. In addition, training GSCAE certainly requires more computational resources than those for GSCA. Specifically, when training GSCAE using a stochastic-based algorithm, we need to compute the gradients w.r.t. θ_x and θ_y , which are associated with both \mathcal{L}_{GSCA} and autoencoder parts. Furthermore, we also need to compute the gradients w.r.t. Φ_x and Φ_y , which are only dependent on the reconstruction losses. We note that similar to GSCA, GSCAE have different versions depending on the selection of the regularization function $\Omega(\gamma)$ in DTW_{Ω} . We denote them as GSCAE-e if $\Omega(\gamma)$ is Shannon entropy and GSCAE-s when squared ℓ_2 norm is selected.

V. GENERALIZED MULTIPLE SEQUENCES ANALYSIS

In this section, we propose generalized multiple sequences analysis (GMSA), which is an extended variant of GSCA for learning representation from multiple data sequences. Slightly differing from the previously proposed method, GMSA directly projects all the data sequences into the label

subspace to learn more interpretable and discriminative representations. The projection functions are parameterized by DNNs, as in GSCA. To accommodate sequential mismatching, we introduce a consensus label sequence that is then aligned to all the output sequences of the DNNs. An alternating optimization algorithm is finally derived to solve the objective function w.r.t. parameters of the DNNs and the consensus label sequence.

A. OBJECTIVE

Given v data sequences $X^{(k)} \in \mathbb{R}^{d_x^{(k)} \times n^{(k)}}$ for $k = 1, \dots, v$, we assume that each sample of these sequences belongs to one of c disjoint classes. The cluster assignment is often denoted by a matrix $F^{(k)} = [f_1^{(k)}, \dots, f_n^{(k)}] \in \mathbb{R}^{c \times n^{(k)}}$, where $f_i^{(k)}$ is the cluster indicator vector³ of sample $x_i^{(k)}$ in the sequence $X^{(k)}$. As in [18]–[20], for each view, we define a scaled cluster indicator matrix

$$\tilde{F}^{(k)} = [\tilde{f}_1^{(k)}, \dots, \tilde{f}_{n^{(k)}}^{(k)}] = (F^{(k)\top} F^{(k)})^{-\frac{1}{2}} F^{(k)\top}. \tag{16}$$

It turns out that

$$\tilde{F}^{(k)} \geq \mathbf{0} \text{ and } \tilde{F}^{(k)} \tilde{F}^{(k)\top} = \mathbf{I}. \tag{17}$$

In GMSA, each input data sequence is passed through a DNN to compute its output sequence $Z^{(k)} = f_k(X^{(k)}, \theta^{(k)}) \in \mathbb{R}^{c \times n^{(k)}}$, where $\theta^{(k)}$ is a collection of all parameters of the k^{th} network. The dimension of the new subspace is exactly the number of the classes, indicating that the input sequences are mapped into the same space with the labels. Because

³ $f_i^{(k)} \in \{0, 1\}^{c \times 1}$ such that $f_{j,i}^{(k)} = 1$ if $x_i^{(k)}$ belongs to the j^{th} class and zero otherwise.

the representation sequences in the new space are possibly unequal in length and sample-wise mismatched, we introduce a consensus label sequence $\mathbf{Z} \in \mathbb{R}^{c \times n}$ of a prespecified length n and minimize its DTW distances with all sequences $\mathbf{Z}^{(k)}$. The optimization problem of GMSA is as follows:

$$\begin{aligned} \min_{\mathbf{Z}, \theta^{(1)}, \dots, \theta^{(v)}} \quad & \sum_{k=1}^v \text{DTW}_{\Omega}(\mathbf{Z}, \mathbf{Z}^{(k)}) + \sum_{k=1}^v \lambda_k \mathcal{L}_k(\mathbf{Z}^{(k)}), \\ \text{subject to } \quad & \mathbf{Z} \geq \mathbf{0} \text{ and } \mathbf{Z}\mathbf{Z}^T = \mathbf{I}, \end{aligned} \quad (18)$$

where λ_k is the weighted parameter for soft whitening regularization of the k^{th} view, which is similar to GSCA. The nonnegative and orthogonal constraints are derived from (17), enforcing \mathbf{Z} to satisfy the cluster indicator conditions. By introducing the consensus label sequence, we can avoid minimizing the sum of all pairwise DTW distances between output sequences, which is computationally demanding and prone to errors. We note that deep discriminant analysis with time warping in [21] also utilizes the idea of mapping the input data sequences into the label space. However, the authors assumed that the supervised information was already available. In our model, the sequential labels are not given in advance. Therefore, GMSA is completely unsupervised and its optimization problem is more difficult to solve.

B. OPTIMIZATION

In this section, we propose an alternating algorithm to solve the optimization problem of GMSA. More specifically, we update all the parameters of the DNNs iteratively when \mathbf{Z} is fixed and then optimize the consensus label sequence after recomputing all output sequences of the DNNs.

Let \mathcal{L}_{GMSA} denote the objective function in (18). When fixing the consensus label sequence, we can compute gradients of \mathcal{L}_{GMSA} w.r.t. $\mathbf{Z}^{(k)}$ for $k = 1, \dots, v$ as follows:

$$\frac{\partial \mathcal{L}_{GMSA}(\mathbf{Z}^{(k)})}{\partial \mathbf{Z}^{(k)}} = \frac{\partial \text{DTW}_{\Omega}(\mathbf{Z}, \mathbf{Z}^{(k)})}{\partial \mathbf{Z}^{(k)}} + \lambda_k \frac{\partial \mathcal{L}_k(\mathbf{Z}^{(k)})}{\partial \mathbf{Z}^{(k)}}. \quad (19)$$

Similar to optimization for GSCA, these gradients are then backpropagated to compute the gradients of \mathcal{L}_{GMSA} w.r.t. the parameters $\theta^{(k)}$ for $k = 1, \dots, v$.

When all parameters $\theta^{(k)}$ are fixed, the output sequences $\mathbf{Z}^{(k)}$ are recomputed and the optimization problem in (18) reduces to

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \sum_{k=1}^v \text{DTW}_{\Omega}(\mathbf{Z}, \mathbf{Z}^{(k)}) \\ \text{subject to } \quad & \mathbf{Z} \geq \mathbf{0} \text{ and } \mathbf{Z}\mathbf{Z}^T = \mathbf{I}. \end{aligned} \quad (20)$$

By adding an extra penalty term $\xi \|\mathbf{Z}\mathbf{Z}^T - \mathbf{I}\|_F^2$ to the objective of problem (20), we can remove the orthogonal constraint. Denote $\mathbf{G} = \sum_{k=1}^v \frac{\partial \text{DTW}_{\Omega}(\mathbf{Z}, \mathbf{Z}^{(k)})}{\partial \mathbf{Z}^{(k)}}$; then, the update rule for \mathbf{Z} is as follows:

$$z_{i,j} \leftarrow z_{i,j} \frac{[4\xi \mathbf{Z}]_{i,j}}{[\mathbf{G} + 4\xi \mathbf{Z}\mathbf{Z}^T \mathbf{Z}]_{i,j}}. \quad (21)$$

Algorithm 2 Alternating Optimization Algorithm for GMSA

- Input:** Input data sequences $\mathbf{X}^{(k)}$ for $k = 1, \dots, v$.
Output: The optimal DNNs' parameters θ^* and consensus sequence \mathbf{Z}^* .
- 1: **for** $k = 1, \dots, v$ **do**
 - 2: Run k-means on $\mathbf{X}^{(k)}$ to generate cluster indicator matrix $\mathbf{F}^{(k)}$;
 - 3: Initialize $\mathbf{Z}^{(k)} = (\mathbf{F}^{(k)\top} \mathbf{F}^{(k)})^{-\frac{1}{2}} \mathbf{F}^{(k)\top}$;
 - 4: Initialize $\theta^{(k)} = \underset{\Theta}{\operatorname{argmin}} \|\mathbf{Z}^{(k)} - f_k(\mathbf{X}^{(k)}, \Theta)\|_F^2$.
 - 5: **end for**
 - 6: **repeat**
 - 7: Update the consensus \mathbf{Z} using equation (21);
 - 8: Update the DNNs' parameters θ using a stochastic algorithm;
 - 9: Recompute $\mathbf{Z}^{(k)} = f_k(\mathbf{X}^{(k)}, \theta^{(k)})$ for $k = 1, \dots, v$;
 - 10: **until** convergence

To guarantee the orthogonality of \mathbf{Z} , we set ξ to a relatively large value, $\xi = 10^6$, in our experiments. The derivation of equation (21) is given in Appendix D. Because we use the Karush–Kuhn–Tucker (KKT) condition to update \mathbf{Z} , the objective value of GMSA can be ensured to decrease monotonically. However, note that the objective function is not jointly convex w.r.t. all the variables and that the alternating optimization algorithm is not guaranteed to converge to the global optimum. Therefore, a good initial guess can help the algorithm to achieve a better optimal solution and converge faster. In this work, we separately run k-means algorithm on the input sequences $\mathbf{X}^{(k)}$ to generate their cluster indicator matrices $\mathbf{F}^{(k)}$ for $k = 1, \dots, v$. The initial value of $\mathbf{Z}^{(k)}$'s output of the DNNs are then computed using equation (16). Afterward, Algorithm 2 summarizes the alternating optimization procedure of GMSA.

VI. RELATED WORKS

To capture the relations between data samples among the views, conventional methods such as CCA and its variants implicitly assume that data of the views are equal in size and sample-wise matching. However, these assumptions are likely to be violated in sequential settings because data sequences often have different lengths and the sample correspondence information is also absent. One major approach to solve the aforementioned problem is directly combining CCA with DTW [22], [24]–[27]. These methods find a subspace such that projections of the two sequences are aligned and the learned representations of the two views are maximally correlated. Recently, [21] has proposed to combine deep CCA [2] with DTW. More complex and nonlinear embeddings can be obtained based on DNNs. For more details about the advantages of deep methods over the shallow ones, we refer the readers to a recent overview [28]. However, this direct combination has several serious drawbacks. Because the DTW problem is discrete and its objective is not differentiable, the alignment and representations are not optimized in

a unified manner. Specifically, the new alignment is updated while fixing the representations and vice versa. Without a good initialization, this update scheme is prone to suboptimal solutions. In addition, when DNNs are used to map the two views into the new subspace, their expensive training procedures need to be performed multiple times. This makes the approach inefficient and unsuitable for extending to larger deep models.

Another approach is based on manifold alignment. These methods project data from two different but correlated manifolds to a subspace, simultaneously preserving the local structures and ensuring their closeness. A subgroup of this technique includes semisupervised methods [31]–[34] that utilize several sample-wise correspondences known in advance between the manifolds while learning a new subspace. In contrast, the second subgroup, which we focus on in this article, contains unsupervised manifold alignment methods that do not require correspondences to be predetermined. Because there is no prior information on the sample-wise pairing, [35] created a connection between the two views by comparing their local geometry, which is characterized by the k -nearest neighbors (k -NNs). [36] has recently proposed a variant of this method where the local geometry information is measured in the fuzzy granule space instead of the original space. [37] built a k -NN graph for each view and extracted a series of graph-based descriptors for each data sample. The cross-view similarity and dissimilarity matrices are then computed in the descriptors space. [38] constructed the cross-view similarity matrix based on probability prediction results, which are obtained by performing classification tasks on both views. [39] took a different approach to the correspondence problem. Specifically, they encoded the cross-view sample-wise matching into a binary matrix, which is jointly optimized with the projection matrices of the two views. [40] further extended the correspondence matrix with an extra row and column. They aimed to better handle outliers that had no corresponding sample from the other set. Nevertheless, these methods cannot take advantage of sequential order in the data to discover more accurate correspondence. In addition, they are also limited to shallow models and sensitive to noise that corrupts the adjacency and geometric information of the data.

[41], [42] proposed hybrid methods that combine unsupervised manifold alignment with DTW. Thus, they also inherit drawbacks from the two presented approaches. Our methods (GSCA and GSCAE) differ from the first approach because they discover the sample correspondence implicitly while learning the representations instead of alternately updating the projections of the views and aligning them. In addition, the proposed objective functions allow us to design an efficient stochastic algorithm for training the models, making them applicable to other deep models. Our methods also differ from the second approach because the smooth DTW can better utilize the sequential order of the data. Furthermore, we parameterize the projection functions by deep feed-forward neural networks and/or RNNs. Therefore, our models can better capture view-specific relations

as they expand through the sequences and learn much more robust and richer nonlinear embedding functions. We note that [43] recently approached the problem of misalignment in multiview sequential learning using memory-based neural networks. Instead of recovering the sample correspondence between the views, this approach stores view-specific information in memory and makes it accessible to the neural network of the other view. Although having promising results in practice, we exclude this approach from our experiments because it is a supervised method, which is not the interest of this article.

In this work, we also consider a more challenging case where the input data comprise more than two sequences. To handle this problem, existing methods such as [44], [45] combine multiset CCA [46] and an approximation of DTW where the warping path is approximated by a linear combination of monotonic basic functions. In comparison with GMSA, these methods are less favorable to data with a complex latent structure because they can only learn a simple linear projection for each view. In addition, how to select a proper set of monotonic basics for a particular dataset remains unclear. Another closely related work of GMSA is deep discriminant analysis with time warping [21]. This method simultaneously projects and aligns the input data sequences to a given label sequence. In contrast, in our case, supervised information is unavailable. Therefore, GMSA solves a more complex problem.

VII. EXPERIMENTS

A. COMPARED METHODS

We compare GSCA and its variant GSCAE with DSCA, which was proposed in our primary work [7], and the following two-view baselines:

- Canonical time warping (CTW) [22]—a direct combination of CCA and DTW;
- Canonical soft time warping (CSTW) [23]—a probabilistic extension of CTW, where the alignment is considered a variable that follows Gibbs distribution. The alignment and projection matrices are alternatively optimized using the Expectation–Maximization (EM) algorithm.
- Autoencoder regularized CTW (AECTW) [29]—a variant of CTW with autoencoder-based regularizations;
- Deep CTW (DCTW) [21]—a direct combination of Deep CCA and DTW;
- Locally unsupervised manifold alignment (LUMA) [35]—an unsupervised manifold alignment-based method that establishes a connection between any two samples from the two views by comparing their local geometries;
- Fuzzy granule manifold alignment (FGMA) [36]—a variant of LUMA, where the local geometry information is collected in the fuzzy granule space instead of the original space.
- Generalized unsupervised manifold alignment (GUMA) [39]—another unsupervised manifold

alignment-based method that encodes cross-view sample-wise correspondence into a binary matrix that is jointly optimized with the projections;

- Manifold alignment time warping (MATW) [42]—a hybrid method where sample alignment is performed by DTW and where projection matrices are optimized to preserve the underlying structures of the two views;
- Generalized canonical time warping (GCTW) [44]—Multiset CCA is used to project data sequence into a shared subspace with an approximation of the DTW algorithm to align the projected sequences.

and compare GMSA with the following multiview method:

We note that DTW_{Ω} in the objective of our methods has different versions depending on the selection of the regularization functions $\Omega(\gamma)$. Therefore, we add suffixes -e and -s to our methods, indicating that $\Omega(\gamma)$ is Shannon entropy and squared ℓ_2 norm, respectively.

B. EVALUATION MEASUREMENTS

All datasets in our experiments are divided into training, tuning, and test sets. We evaluate these methods by measuring class separation in the learned embedding spaces on the test set. First, we perform clustering tasks on the projections of the first view and evaluate how well the clusters agree with the ground-truth labels.⁴ We follow the same procedure in [47], where spectral clustering [48] is used to handle possibly nonconvex cluster shapes. We set the number of clusters to the number of ground-truth classes available in each dataset. Clustering accuracy (ACC) and normalized mutual information (NMI) [49] are used as measurements for assessing the clustering performance. Second, we test the accuracy of a simple linear classifier on the learned embeddings. We train one-versus-one linear support vector machines (SVMs) [50] on the projected training set of the first view (label information is used). The trained model is then used to classify projections of the test set, and the percentage of errors is reported as a measurement of classification performance.

C. PARAMETER TUNING

We select the optimal parameters that return the best evaluation measurement results on the tuning set for each method.

Two-View Methods: Dimension d of the new subspace is selected from $\{5, 10, 20, 30, 50, 70\}$. For manifold alignment-based methods, we select the parameter that balances between sample matching and geometry preserving from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. The number of neighbors for building the k-NN graph that encodes the local geometry is selected from $\{1, 3, 5, 10, 15, 30\}$. We found that these methods return the best average results at $k = 5$. The trade-off parameter between the autoencoder regularization term and the alignment objective in AECTW and GSCAE is selected using a grid search. For the soft whitening constraints in DSCA, GSCA, and GSCAE, we set $\lambda_x = \lambda_y$, and their values are also selected using grid search. Another important

⁴For GMSA, we use the projections of the views as their cluster indicators.

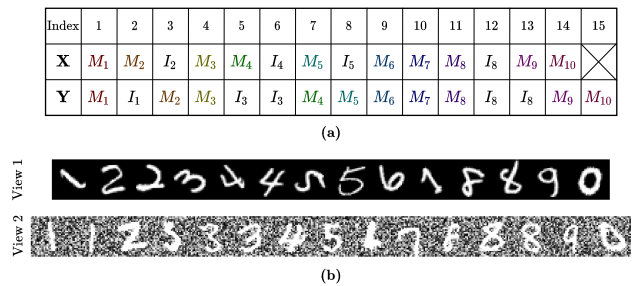


FIGURE 4. A toy example of how to generate misaligned sequences. (a) Hidden states generated by pHMM and (b) the corresponding two data sequences generated from the noisy MNIST digits dataset.

parameter is β , which controls the regularization in the smooth min operator. We set $\beta = 1$, as suggested in [7], [51]. For the DNNs used in the compared methods, their topology and configurations are data-dependent and are specified in the following subsections.

Multiview Methods: For GCTW, similar to the two-view methods, we select a dimension of the new subspace from $\{5, 10, 20, 30, 50, 70\}$. To approximate the optimal warping paths, we use five hyperbolic tangent and five polynomial functions as the monotonic basics. The other parameters are set according to the original article. In contrast to GCTW, our method GMSA projects the input sequences into the label space. Therefore, we choose the dimension d of the new space for GMSA such that it is identical to the number of classes available in the datasets. We use the same soft whitening regularization parameters for all the views: $\lambda_1 = \lambda_2 = \dots = \lambda_v$, and the value is chosen using grid search. Let n_a, n_s , and n_l denote the average, shortest, and longest lengths, respectively, of the input data sequences. Then, the length of the consensus sequence Z is selected from a rounded set $\{n_s, \max(n_s, 0.5n_a), \max(n_s, 0.75n_a), n_a, \min(n_l, 1.25n_a), \min(n_l, 1.5n_a), n_l\}$. The alternating optimization algorithm of GMSA is determined to be converged if the relative reduction of the objective is smaller than a tolerance $\epsilon = 10^{-5}$. In practice, we also terminate the algorithm if the number of iterations exceeds a prespecified value $iter_max = 50$.

D. TWO-VIEW DATA I: NOISY MNIST DIGITS

In this experiment, we utilize the MNIST dataset [17], which consists of 28×28 grayscale digit $[0, 9]$ images divided into 60K/10K for training/testing. Following the procedure in [47], we generate two-view data as follows. For the first view, we rescale the pixel to $[0, 1]$ and randomly rotate the images at angles uniformly sampled from $[-\frac{\pi}{4}, \frac{\pi}{4}]$. For each image of the first view, we randomly select an image of the same identity from the original dataset, add noise uniformly sampled from $[0, 1]$, and truncate the pixel value to $[0, 1]$. This image is further resized to 24×24 and used for the second view. 10K from 60K image pairs of the original training set are set aside for tuning.

From 50K image pairs $\{(x_i, y_i, l_i) | 1 \leq i \leq 5 \times 10^4, x_i \in \mathbb{R}^{784}, y_i \in \mathbb{R}^{576}, l_i \in \{0, \dots, 9\}\}$ of the new training

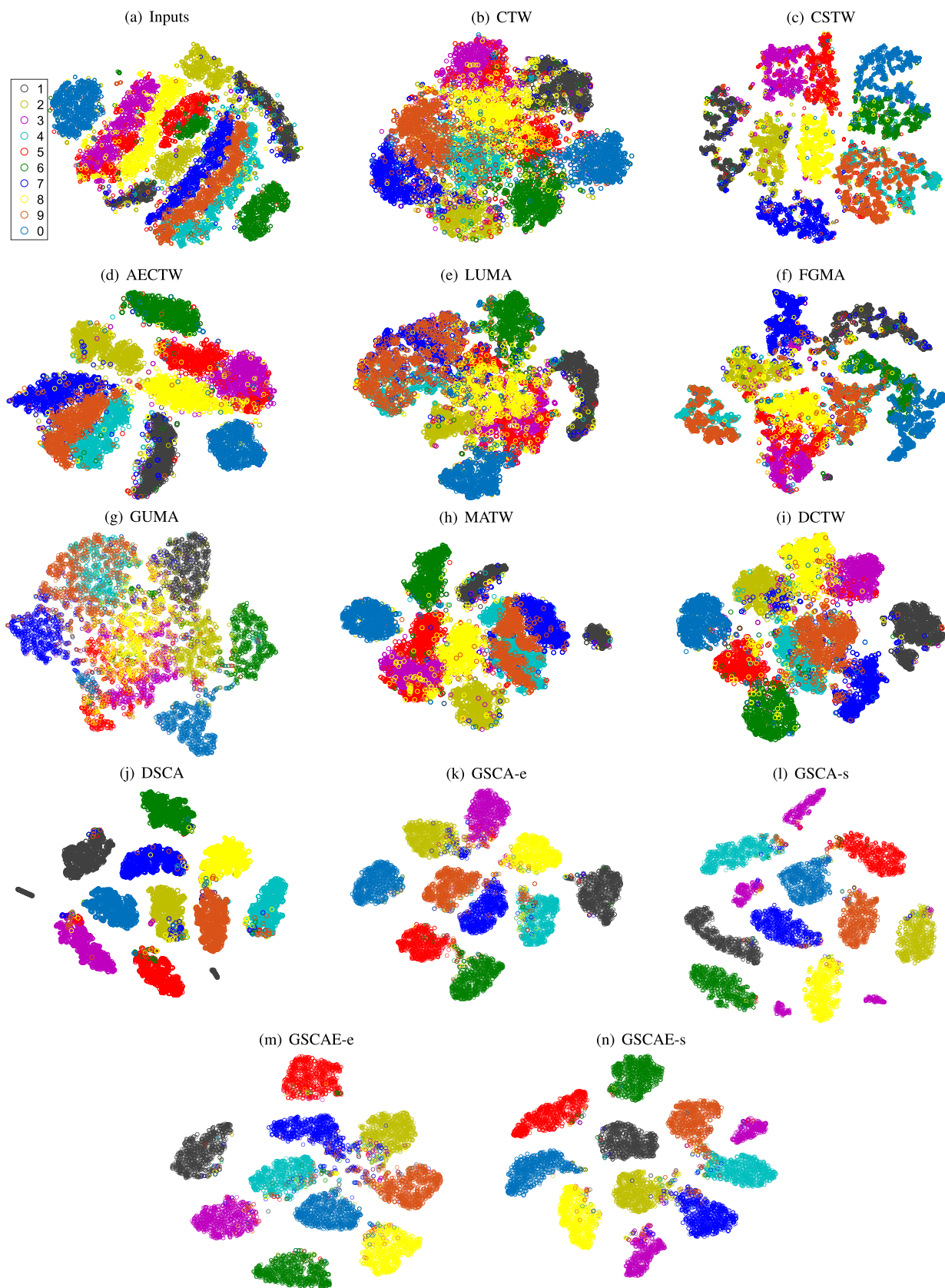


FIGURE 5. t-SNE [53] visualization of the projected test set of noisy MNIST digits on shared subspaces with dimension $d = 10$ returned by different methods.

set, we generate two misaligned sequences using a profile hidden Markov model (pHMM) [52]. Specifically, we generate two state sequences, consisting of MATCHING state M_i that emits the i^{th} matching sample and INSERT state I_i intended for emitting sample replication. The transition probability is chosen such that from any state, the next state is MATCHING with probability 0.6 and INSERT with probability 0.4. We terminate the state sequences after reaching the $(5 \times 10^4)^{\text{th}}$ MATCHING. The state M_i of the first sequence corresponds to sample x_i . For state I_i , we replicate x_i by randomly selecting a sample $x_{l=i}$ from its class (having the same identity). (Similarly for the second sequence). Figure 4 shows a toy example of how to generate misaligned sequences using pHMM for a given small set of 10 training image pairs.

In this experiment, we used feed-forward neural networks for all DNN-based methods, including DCTW, DSCA, GSCA, and GSCAE. To parameterize the projection functions that map data from original spaces to the new subspace, we use two fully connected networks whose numbers of sigmoid units at hidden layers are 1200 – 1200 – 1200 (for the first view) and 1000 – 1000 – 1000 (for the second view). Note that each network includes a BN layer of d units on the top as an output layer. The view reconstruction in GSCAE is performed by the symmetric DNNs. Figure 5 visualizes the first view in the original space and its projections in the subspaces learned by different methods. The class separation results are shown in Table 1.

The results show that among manifold alignment-based methods, FGMA had better scores than LUMA and GUMA. FGMA evaluates the local geometry of the data after converting them into a fuzzy granule space. Thus, FGMA can discover more complex local structure information. MATW is a hybrid method. Differing from LUMA, FGMA, and GUMA, MATW discovers sample correspondences by DTW. Thus, it can take advantage of sequential order in the data to find the cross-view correspondence. Nevertheless, these methods returned poor results on noisy MNIST digits dataset because the noise corrupted the geometric information. In contrast, the deep learning-based methods returned much higher class separation results, even in noisy conditions. These methods mapped samples of the same class to similar locations while suppressing noise and rotational variation in the data. We also observe that methods with a smooth approximation of DTW, including DSCA, GSCA, and GSCAE, worked much better than CTW, AECTW, and DCTW, which directly combine the original DTW with variants of CCA. By minimizing the differentiable version of DTW, alignment and projection can be optimized in a unified manner. CSTW also implicitly optimizes DTW because it considers the warping path as a probabilistic variable. However, its EM algorithm still updates the alignment and projection matrices alternatively, which is prone to suboptimal solutions.

The experimental results also show that by carefully tuning the trade-off parameters and combining CTW or GSCA with autoencoders (forming the variants AECTW or GSCAE) can improve their performances. We note that each method

TABLE 1. Clustering (ACC, NMI) and classifying (Error) results on the noisy MNIST digits dataset. The data sequences are generated randomly five times using the pHMM-based procedure. Each method is performed on these data to learn the new embeddings and the average results along with variance on projections of the test set are reported.

Method	ACC (%)	NMI (%)	Error (%)
Inputs	45.14 (5.27)	48.05 (6.77)	15.61 (2.86)
CTW	66.33 (4.47)	52.17 (4.81)	20.28 (3.21)
CSTW	75.16 (1.62)	73.91 (2.01)	7.92 (1.50)
AECTW	82.62 (2.22)	79.66 (2.12)	8.24 (1.72)
LUMA	62.37 (3.21)	61.25 (3.71)	26.23 (3.64)
FGMA	66.51 (2.19)	65.93 (2.87)	24.36 (2.92)
GUMA	60.38 (3.47)	54.25 (2.90)	28.03 (3.14)
MATW	69.18 (4.86)	67.44 (5.13)	21.39 (3.03)
DCTW	86.46 (1.13)	84.22 (2.09)	6.23 (1.46)
DSCA	95.12 (0.98)	93.21 (1.34)	2.81 (0.58)
GSCA-e	95.14 (1.03)	93.14 (1.12)	2.80 (0.61)
GSCA-s	90.57 (0.95)	89.93 (1.17)	5.06 (0.92)
GSCAE-e	96.67 (0.81)	95.66 (0.68)	3.12 (1.02)
GSCAE-s	91.48 (1.15)	91.35 (0.87)	4.47 (1.33)

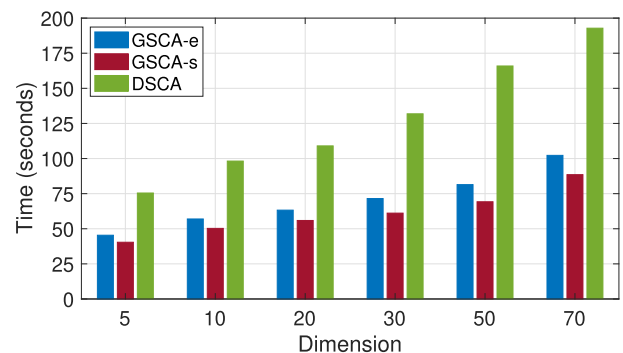


FIGURE 6. Average times for computing stochastic gradients of GSCA-e, GSCA-s, and DSCA over a batch size ratio $\alpha = 0.1$ (equivalent to a batch size of about 1.5K) on noisy MNIST digits dataset with different dimensions d of the learned subspace.

proposed in this article has two versions depending on the selection of the regularization $\Omega(\eta)$. Although GSCA-e and DSCA have similar scores as their objectives are equivalent, computing the gradient of GSCA-e is much more efficient. Figure 6 shows the average times for computing the stochastic gradients of GSCA-e, GSCA-s, and DSCA over different dimensions d . Because of the sparsity of the gradient induced by squared ℓ_2 norm, training GSCA-s and GSCAE-s are generally faster than GSCA-e and GSCAE-e, respectively. However, this advantage comes at a cost of slightly lower class separation scores.

E. TWO-VIEW DATA II: ACOUSTIC AND ARTICULATORY RECORDINGS

We next evaluate the performances of the two-view methods on the Wisconsin X-ray microbeam (XRMB) corpus [54], which consists of 2537 utterances recorded from 47 American English speakers. Lengths of the utterances vary from 63 to 2941 frames. Each frame is basically described by 39D acoustic features (13-dimensional mel-frequency cepstral coefficients [MFCCs] along with their first and second derivatives) and 16D articulatory features (horizontal/vertical displacement of 8 pellets attached to the tongue, lips, and

jaw). To incorporate context information and generate two sequential views of different frame rates, we slide windows of 7 and 9 frame sizes over each utterance with one frame step size. The frames within the windows are concatenated, resulting in 273D acoustic and 144D articulatory input samples. Because each original frame belongs to one of 41 phone classes, we consider the labels of the central frames as those of the newly generated inputs.

The utterances are presently characterized by two sequences whose lengths are different and the sample correspondences are also missing. We randomly divide them into 1415/471/471 for training/tuning/testing. We use RNNs for GSCA and GSCAE to better capture the sequential nature of the data in this experiment. Specifically, for each view, we stack three LSTM units with the same numbers of memory cells (1500 for acoustic view and 1200 for articulatory view) along with a fully connected BN layer of d units at the output to parameterize the projection function. The view reconstruction in GSCAE is performed by the symmetric deep LSTM networks. Note that while training these models using Algorithm 1, at each iteration, we randomly sample acoustic and articulatory input sequences that correspond to one of the 1415 training utterances to compute the stochastic gradient. Thereby, we can take better advantage of the sequential nature in the data for training the RNNs. For DCTW and DSCA, we concatenate two views of the training utterances into two long sequences and feed them separately to two fully connected networks. These networks consist of three hidden layers whose activation functions are ReLU, and the numbers of the units are 1500 – 1500 – 1500 for the acoustic view and 1200 – 1200 – 1200 for the articulatory view.

Table 2 shows the phone class separation results on representations obtained by different methods. Similar to the results on the noisy MNIST digits dataset, the DNN-based methods outperformed CTW, CSTW, AECTW, and the manifold alignment-based methods. The DNNs enable those methods to approximate projection functions nonlinearly, improving the quality of the learned embeddings. In this experiment, the designed RNNs have shown positive effects on our methods. They allow the models to better capture the sequential relations among data samples. As a consequence, GSCA and GSCAE achieved the highest scores among the compared methods. We also see that the results of AECTW and GSCAE surpass those of CTW and GSCA, respectively. This again validates the benefits of coupling autoencoder-based regularizations with the objective functions for providing a better trade-off between view-specific and cross-view information.

We then investigate average times for computing stochastic gradients of GSCA-e, GSCA-s, and DSCA. Because of the differences between their network architectures (RNN in GSCA and feed-forward network in DSCA), we exclude the computational time of the backpropagation⁵ (Line 7 in Algorithm 1). Figure 7 shows that the computing gradients

TABLE 2. Phone class separation on the projections of the acoustic view learned by different methods. The testing set is randomly divided into six folds. Clustering and classification tasks are performed on each fold and the average results along with their variances are reported.

Method	ACC (%)	NMI (%)	Error (%)
Inputs	48.54 (3.51)	50.26 (3.65)	36.15 (4.51)
CTW	67.74 (3.11)	67.18 (3.32)	28.05 (3.88)
CSTW	68.82 (2.59)	67.49 (2.63)	27.64 (2.97)
AECTW	68.16 (2.32)	69.55 (2.69)	27.92 (3.63)
LUMA	56.23 (3.16)	56.83 (2.88)	30.33 (3.47)
FGMA	58.12 (2.92)	58.41 (2.95)	29.62 (3.03)
GUMA	55.73 (3.25)	54.61 (3.12)	30.97 (3.55)
MATW	64.51 (3.04)	63.38 (2.92)	29.35 (3.85)
DCTW	74.51 (3.07)	75.19 (2.90)	27.50 (3.28)
DSCA	79.03 (2.91)	79.22 (3.27)	27.01 (2.99)
GSCA-e	82.45 (3.23)	82.83 (2.85)	26.48 (3.05)
GSCA-s	81.68 (2.84)	81.02 (3.19)	26.87 (2.94)
GSCAE-e	83.90 (2.93)	82.91 (3.15)	25.35 (3.13)
GSCAE-s	82.02 (3.02)	80.93 (2.89)	26.07 (2.86)

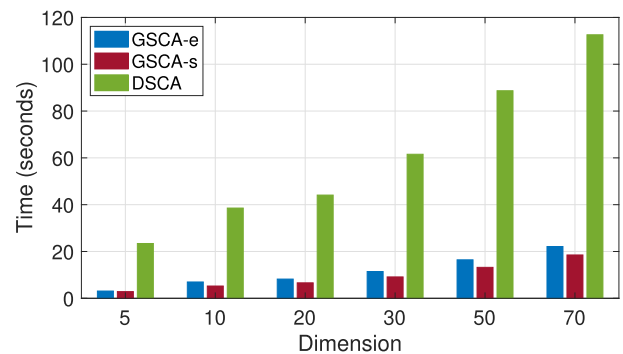


FIGURE 7. Average times for computing stochastic gradients of GSCA-e, GSCA-s, and DSCA on the XRMB dataset. For a fair comparison, computation times for backpropagation (or BTT) are excluded. The computation is taken on minibatches with the sizes are equal to average length of the training utterances (about 1K samples).

of GSCA are much faster than that of our primary model DSCA. This efficiency originates from the use of the new generalized smooth DTW. We can further reduce the training time by setting $\Omega(\eta)$ in DTW_{Ω} to be squared ℓ_2 norm. However, as shown in Table 2, the class separation results were slightly decreased. Because $DTW_{\Omega=\text{squared } \ell_2}$ is a nonexact approximation of DTW, GSCA-s and GSCAE-s included some certain bias in comparison with the Shannon entropy-based versions.

F. MULTIVIEW DATA I: HUMAN ACTIONS WITH MULTIPLE FEATURE SETS

In this experiment, we evaluate the performances of multiview methods, including GMSA and GCTW, on the Weizmann dataset [55], which consists of 90 videos of nine subjects, each performing ten actions: wave-one-hand (wave 1), wave-two-hand (wave 2), side, jump-in-place (pjump), jump-forward (jump), jack, skip, bend, walk, and run. Similar to [56], we concatenate videos of the same subject into a long video sequence following the presented order of the actions. Background is subtracted from each frame of these video sequences and the frames are then rescaled to the

⁵Note that in GSCA, we use BTT for computing gradients for its RNNs

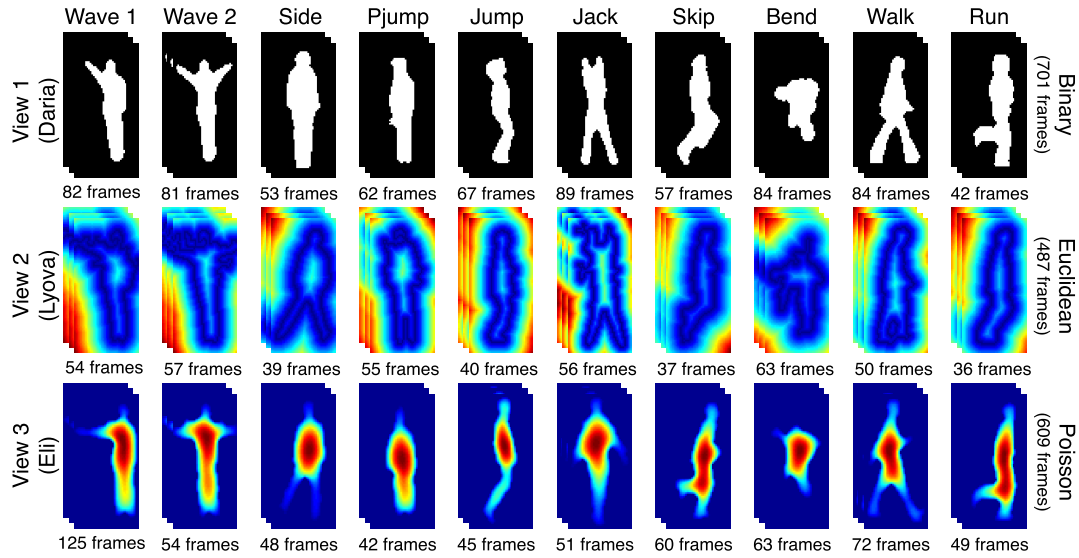


FIGURE 8. Three-view sequential data generated from the Weizmann dataset. The views are constructed by concatenating ten action videos of three subjects named Daria, Lyova, and Eli, respectively. Note that each view has different features: Binary (view 1), Euclidean distance transform (view 2), and solution of Poisson equation (view 3).

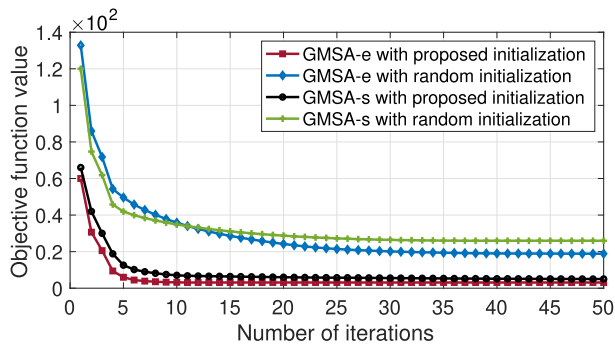


FIGURE 9. Convergence curves (objective function value averaged over five runs against the number of iterations) of GMSA-e and GMSA-s on the Weizmann dataset.

size 80×40 . There are three types of features that can be computed to characterize the frames, including *type 1*: binary, *type 2*: Euclidean distance transform [57], and *type 3*: solution of the Poisson equation [58]. We generate three-view sequential data for training by selecting video sequences of the first three subjects, each of which is represented by one of the three feature types without repetition. As a result, each view of the data has different features, and each frame of the views belongs to one of the ten classes (see Figure 8 for more details). To reduce the dimensions of the feature space (3200), the top 123 principal components that preserve 99% of the total energy are selected. Videos of the next three subjects are used for tuning, and the remaining subjects’ videos are utilized for testing.

For GMSA, we use RNNs to parameterize the projection functions. We use a similar network configuration for all the views because three views of the dataset have the same input dimensions. Specifically, the data of each view are passed through a deep network with three stacked LSTM units, each

TABLE 3. Performance measures of clustering (ACC, NMI) and classifying (Error) on the projections of the Weizmann dataset, using GCTW and GMSA. Each method is run randomly five times, and their average results along with the variances on the test set are reported.

Method		ACC (%)	NMI (%)	Error (%)
Input	view 1	53.75 (2.35)	62.48 (1.17)	14.21 (2.11)
	view 2	57.92 (1.72)	63.73 (2.02)	12.37 (1.27)
	view 3	49.15 (1.87)	55.22 (1.21)	13.53 (1.61)
GCTW	view 1	67.32 (2.06)	72.37 (1.27)	10.49 (1.14)
	view 2	68.49 (1.94)	74.01 (1.83)	8.72 (0.98)
	view 3	65.87 (1.55)	70.38 (2.13)	10.15 (1.36)
GMSA-e	view 1	86.65 (1.82)	88.50 (1.80)	5.04 (1.85)
	view 2	88.03 (1.90)	90.02 (1.89)	5.94 (1.85)
	view 3	84.62 (1.51)	86.12 (1.19)	6.56 (1.66)
GMSA-s	view 1	84.62 (2.08)	87.11 (1.96)	5.75 (1.91)
	view 2	87.59 (1.73)	88.54 (1.87)	7.04 (1.75)
	view 3	83.42 (1.57)	85.15 (1.49)	6.12 (1.55)

of which has 256 memory cells. The output layer of the network is a BN layer with $d = 10$ units. Because the new subspace in GMSA corresponds to the label space, we expect the learned representations can have cluster interpretability and better discriminability. Let $z_i^{(k)} \in \mathbb{R}^d$ be the projection of the testing sample $x_i^{(k)}$, we then assign $x_i^{(k)}$ to class j such that $z_{j,i}^{(k)}$ is the largest element of $z_i^{(k)}$. Table 3 shows the class separation results on the representations learned by GCTW and GMSA.

The results show that the performances of clustering and classification on the new embeddings returned by GCTW and GMSA are much better than those on the original space. These results indicate that integrating complementary information from different views helps the two methods to improve the quality of the new representations. In addition, the results also show that the improvement of GMSA is more considerable because it can learn more richer nonlinear embeddings. In contrast, GCTW limits itself to

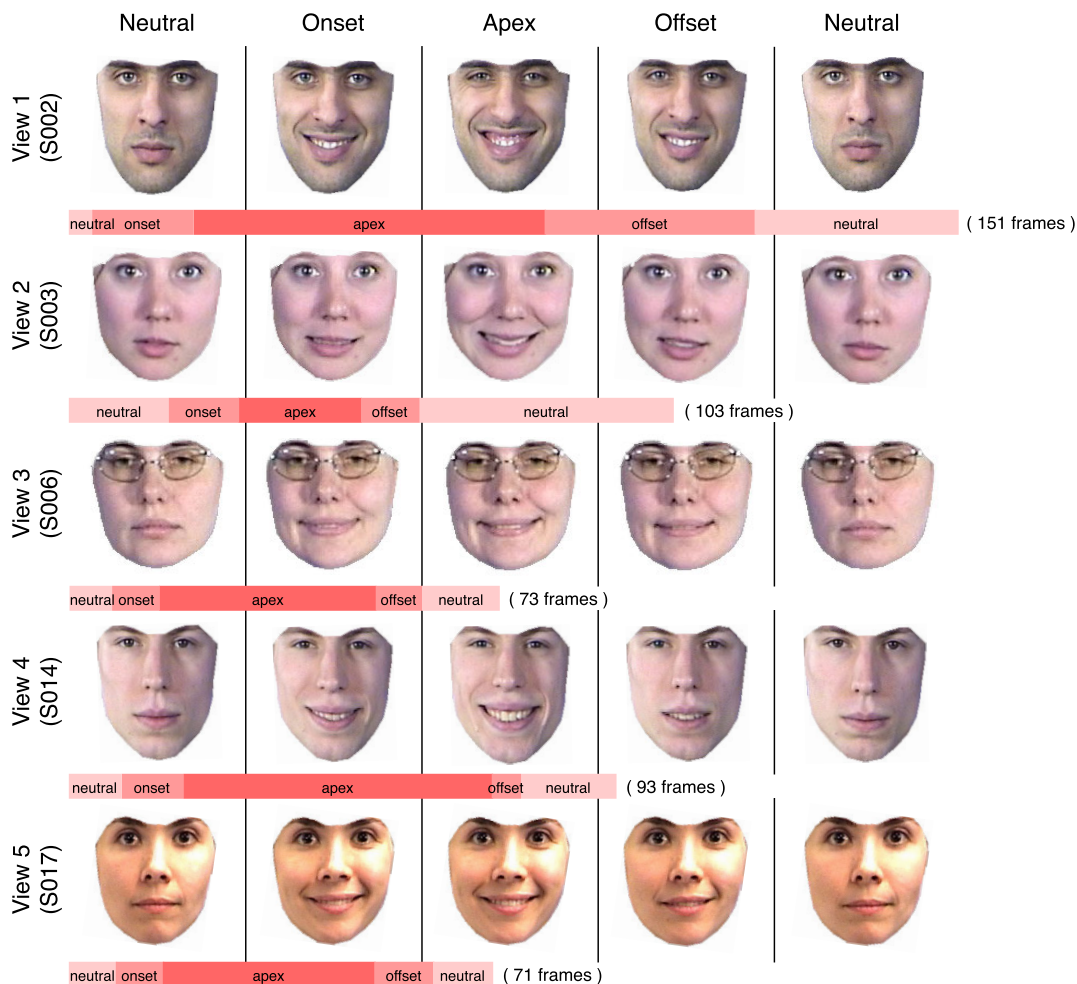


FIGURE 10. Five-view sequential data generated from the MMI facial expression dataset. The representative facial images of the classes are depicted. The bottom of each view shows the duration of the corresponding ground-truth temporal labels along with the total number of frames.

a shallow model where only linear projection matrices can be obtained. Another unfavorable property of GCTW is that the accuracy of its alignment procedure heavily depends on the selection of the monotonic basic functions. However, how to choose a suitable collection of basics for a particular dataset remains unclear. Therefore, the inappropriate settings of monotonic bases might degenerate the results of GCTW.

We then empirically explore the convergence of the optimization algorithm for GMSA. Each outer iteration of the algorithm includes two steps: finding the optimal consensus label sequence and updating the parameters for all DNN branches of the models. Figure 9 shows the convergence curves (objective function value against the number of iterations) with and without the proposed initialization (see Algorithm 2) on the Weizmann dataset. The results show that the algorithm still converges even with a random start. However, the proposed initialization improved the performance of the optimization procedure significantly. Not only does this help the algorithm to converge faster, but a good initial guess also

TABLE 4. Class separation results on the representations learned by GCTW and GMSA on the MMI facial expression dataset. Each method is run randomly five times, and their best average scores along with the corresponding views are reported.

Method	ACC (%)	NMI (%)	Error (%)
Input	64.40 (View 1)	61.51 (View 4)	10.29 (View 1)
GCTW	77.32 (View 1)	76.03 (View 4)	8.55 (View 5)
GMSA-e	90.63 (View 4)	90.91 (View 4)	2.88 (View 5)
GMSA-s	86.34 (View 4)	87.17 (View 4)	3.79 (View 3)

allows better solutions to be obtained. These results again elucidate the efficiency of the GMSA model.

G. MULTIVIEW DATA II: MMI FACIAL ACTION UNITS

We next exploit the MMI facial expression dataset [59], which contains more than 2900 videos of 75 different subjects, each performing a particular combination of an action unit (AU). In our work, we focus on videos of AU12, which corresponds to a smile. These videos consist of different number of frames, and each belongs to one of three classes: *neutral* (when facial muscle is inactive), *apex* (when facial muscle intensity is strongest), and *onset* (when facial muscle

TABLE 5. Ablation analysis of GMSA-e. The views are removed one by one, ablating one corresponding branch of DNN from the model. The best class separation scores of the ablated GMSA-e along with their differences to the results of the original one (full views) are reported.

Dataset	Model	ACC (%)	NMI (%)	Error (%)
Weizmann	GMSA-e	88.03	90.02	5.04
	GMSA-e (w/o Binary view)	85.41 (↓ 2.62)	87.03 (↓ 2.99)	7.17 (↑ 2.13)
	GMSA-e (w/o Euclidean distance transform view)	83.26 (↓ 4.77)	85.82 (↓ 4.20)	6.49 (↑ 1.45)
	GMSA-e (w/o Solution of the Poisson equation view)	86.93 (↓ 1.10)	88.14 (↓ 1.88)	6.08 (↑ 1.04)
MMI	GMSA-e	90.63	90.91	2.88
	GMSA-e (w/o view 1)	87.42 (↓ 3.21)	87.67 (↓ 3.24)	3.91 (↑ 1.03)
	GMSA-e (w/o view 2)	89.05 (↓ 1.58)	88.93 (↓ 1.98)	3.35 (↑ 0.47)
	GMSA-e (w/o view 3)	89.17 (↓ 1.46)	88.62 (↓ 2.29)	3.28 (↑ 0.40)
	GMSA-e (w/o view 4)	85.56 (↓ 5.07)	85.15 (↓ 4.87)	5.11 (↑ 2.23)
GMSA-e (w/o view 5)	88.01 (↓ 2.62)	87.84 (↓ 3.07)	4.03 (↑ 1.15)	

starts to activate) or *offset* (when facial muscle begins to relax). We first preprocess each frame by performing face cropping and face alignment using *dlib-ml* [60]. The results are depicted in Figure 10. We then convert them to grayscale and reduce their dimension. Specifically, we utilize whitening PCA to pick the top 400 components, preserving 99% of the total energy. Finally, we generate sequential data with five views using videos S002–005, S003–023, S006–026, S014–009, and S017–004. Tuning and testing are performed on videos of the same subjects but in different trails (S002–006, S003–024, S006–025, S014–010, and S017–006).

In this experiment, RNNs are used to parameterize the projection functions for GMSA. We stack two LSTM units, which each has 800 memory cells, and a BN layer with $d = 3$ units as the output layer to form a deep network for each view. The projections of the views are used to predict the cluster labels and perform the classification task. Table 4 shows the results of GCTW and GMSA.

The results show the same pattern as on the Weizmann dataset: i.e., the representations learned by GCTW and GMSA significantly improve the performances of clustering and classification tasks in comparison with those on the original input features. Because GMSA-s and GMSA-e are nonlinear methods, their results are much better than those of GCTW. We also note that the multiple alignments in GMSA are simpler than that in GCTW because of the introduction of the consensus sequences. GCTW discovers the sample correspondences by instead performing pairwise alignment between every two views, while there are up to five views in this dataset. Therefore, more errors potentially occurred and propagated through update iterations in GCTW.

Finally, we investigated the convergence of Algorithm 2 on the MMI facial expression dataset. Its convergence curve, which shows the objective value against the number of iterations, is depicted in Figure 11. The figure shows that the algorithm always converges, regardless of the initial conditions. Because the updated equation for the consensus label sequence satisfies the KKT conditions and the optimization for DNNs' parameters is based on the gradient descent method, the objective value is guaranteed to not increase after each iteration. In addition, as on the Weizmann dataset, we also observe that the proposed initialization significantly

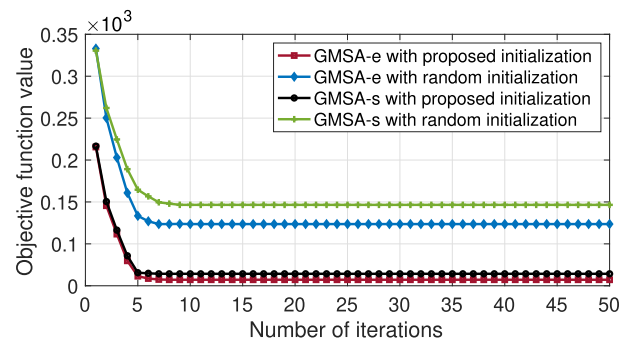


FIGURE 11. Convergence curves (objective function value averaged over five runs against the number of iterations) of GMSA-e and GMSA-s on the MMI facial expression dataset.

improves the performance of the algorithm. With a better initial value, the algorithm could converge with a much lower objective value, hence obtaining a superior optimal solution.

H. ABLATION ANALYSIS OF GMSA

In this section, we conducted ablation experiments to investigate the multiview effect in the GMSA-e model. For a v -views dataset,⁶ the GMSA-e model consists of v branches of DNNs, and each of which maps an input data sequence from one view into the shared label space. Following the same procedure in [61] and [62], we remove the branches of DNNs one by one and report the results in Table 5. The results show that some views are more important than others. For example, the absence of the Euclidean distance transform view in the Weizmann dataset or view 4 in the MMI dataset produces the most significant reduction in the results of the model. However, all of the views contribute more or less to the improvement of the model's performance. GMSA-e with full views has better separation scores than it does with view absence. This result again verifies the advantages of GMSA, which can handle multiple sequential views instantly.

VIII. CONCLUSION

Multiview sequential data pose many challenges to representation learning. In particular, the data sequences of different views are often unequal in size and sample-wise mismatching. Therefore, in this article, we introduced GSCA,

⁶ $v = 3$ for the Weizmann human action dataset and $v = 5$ in case of the MMI facial expression dataset.

a DNN-based method that can discover sample correspondence implicitly while learning representations. By using a generalized smooth DTW distance, which is a differentiable approximation of the original DTW, our model can be trained using a gradient descent-based algorithm, where the gradient can be computed efficiently in terms of both time and space. Our model can be easily extended to improve its learning performance. For instance, we added two DNNs to the GSCA model for view reconstructions, forming a new variant GSCAE. The second model allows a trade-off between view-specific and cross-view relations when learning the representations. Given more than two data sequences, it is obvious that both GSCA and GSCAE are inapplicable. Hence, we further develop the third model called GMSA that can simultaneously handle multiple data sequences and learn more interpretable representations. Through extensive experimentation on different publicly available datasets, our methods were compared with various baselines. The results show that the performances of our methods surpass those of the competitors of all the datasets.

APPENDIX A SMOOTH MIN OPERATOR

The smooth min operator is defined as:

$$\min_{\Omega}(\boldsymbol{\eta}) := \min_{\boldsymbol{\gamma} \in \Delta^k} \langle \boldsymbol{\gamma}, \boldsymbol{\eta} \rangle + \frac{1}{\beta} \Omega(\boldsymbol{\gamma}), \quad (22)$$

where the regularization term $\Omega(\boldsymbol{\gamma})$ must be a strictly convex function [9]. Two widely used functions are Shannon entropy and squared ℓ_2 norm.

Shannon Entropy: If $\Omega(\boldsymbol{\gamma}) = \sum_{i=1}^k \gamma_i \ln \gamma_i$, we obtain

$$\min_{\Omega}(\boldsymbol{\eta}) = \min_{\boldsymbol{\gamma} \in \Delta^k} \sum_{i=1}^k \gamma_i \eta_i + \frac{1}{\beta} \sum_{i=1}^k \gamma_i \ln \gamma_i. \quad (23)$$

Because the objective is strictly convex, we can take its Lagrangian:

$$L = \sum_{i=1}^k \gamma_i \eta_i + \frac{1}{\beta} \sum_{i=1}^k \gamma_i \ln \gamma_i + \lambda_1 \left(1 - \sum_{i=1}^k \gamma_i \right) + \lambda_2 \sum_{i=1}^k \gamma_i. \quad (24)$$

With KKT conditions $\frac{\partial L}{\partial \gamma_i} = 0$ and slackness $\lambda_2 \gamma_i = 0$, we have

$$\gamma_i = e^{\beta \lambda_1 - \beta \eta_i - 1} \quad \forall i = 1, \dots, k. \quad (25)$$

Combining with the simplex constraint: $\sum_{i=1}^k \gamma_i = 1$, we obtain

$$e^{\beta \lambda_1} = \frac{e}{\sum_{i=1}^k e^{-\beta \eta_i}}. \quad (26)$$

Plugging this back into equation (25), we arrive at the minimum of (23)

$$\gamma_i = \frac{e^{-\beta \eta_i}}{\sum_{j=1}^k e^{-\beta \eta_j}}. \quad (27)$$

In summary, when using Shannon entropy as regularization, we have closed-form solutions of the smooth min operator and its gradient

$$\min_{\Omega}(\boldsymbol{\eta}) = -\frac{1}{\beta} \ln \sum_{i=1}^k e^{-\beta \eta_i}, \quad (28)$$

$$\nabla \min_{\Omega}(\boldsymbol{\eta}) = \frac{e^{-\beta \boldsymbol{\eta}}}{\sum_{j=1}^k e^{-\beta \eta_j}}. \quad (29)$$

Squared ℓ_2 Norm: When $\Omega(\boldsymbol{\gamma}) = \frac{1}{2} \sum_{i=1}^k \gamma_i^2$, the smooth min becomes

$$\min_{\Omega}(\boldsymbol{\eta}) = \min_{\boldsymbol{\gamma} \in \Delta^k} \sum_{i=1}^k \gamma_i \eta_i + \frac{1}{2\beta} \sum_{i=1}^k \gamma_i^2. \quad (30)$$

It can be easily shown that the minimum $\boldsymbol{\gamma}^*$ (i.e. $\nabla \min_{\Omega}(\boldsymbol{\eta})$) of (30) is the projection of $-\beta \boldsymbol{\eta}$ onto the simplex Δ^k

$$\boldsymbol{\gamma}^* = \operatorname{argmin}_{\boldsymbol{\gamma} \in \Delta^k} \|-\beta \boldsymbol{\eta} - \boldsymbol{\gamma}\|_2^2, \quad (31)$$

which is likely to be sparse. The solution of (31) can be efficiently obtained using the algorithm proposed in [63]–[65] with a complexity of $O(k \ln k)$.

APPENDIX B GENERALIZED SMOOTH DTW WITH ENTROPY REGULARIZATION

Theorem 1: Let $\boldsymbol{\Pi}$ denote the set of all warping paths

$$\boldsymbol{\pi} = \langle (i_1, j_1), \dots, (i_p, j_p) \rangle, \quad (32)$$

where the set satisfies three conditions: *Boundary, Continuity, and Monotonicity*, as described in Section II, and $\{s(\boldsymbol{\pi}) = d_{i_1, j_1} + \dots + d_{i_p, j_p} | \boldsymbol{\pi} \in \boldsymbol{\Pi}\}$ be a set of cumulative sums corresponding to all the warping paths. If the regularization Ω is the Shannon entropy, then

$$\begin{aligned} DTW_{\Omega}(\mathbf{X}, \mathbf{Y}) &= DTW_{\beta}(\mathbf{X}, \mathbf{Y}) \\ &= -\frac{1}{\beta} \ln \sum_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} e^{-\beta s(\boldsymbol{\pi})}. \end{aligned} \quad (33)$$

Proof: Let $\boldsymbol{\Pi}_{i,j} \subset \boldsymbol{\Pi}$ be the set of all warping paths from $(1, 1)$ to (i, j) and denote

$$r_{i,j} = -\frac{1}{\beta} \ln \sum_{\boldsymbol{\pi}_0 \in \boldsymbol{\Pi}_{i,j}} e^{-\beta s(\boldsymbol{\pi}_0)}. \quad (34)$$

Note that when the regularization Ω is the Shannon entropy smooth min has a closed-form expression, as shown in equation (28), we also have

$$r_{i,j} = \min_{\Omega}(\{s(\boldsymbol{\pi}_0) | \boldsymbol{\pi}_0 \in \boldsymbol{\Pi}_{i,j}\}). \quad (35)$$

We can rewrite Equation (34) as follows:

$$\begin{aligned} r_{i,j} &= -\frac{1}{\beta} \ln \left(\sum_{\boldsymbol{\pi}_1 \in \boldsymbol{\Pi}_{i-1,j}} e^{-\beta (s(\boldsymbol{\pi}_1) + d_{i,j})} \right. \\ &\quad + \sum_{\boldsymbol{\pi}_2 \in \boldsymbol{\Pi}_{i,j-1}} e^{-\beta (s(\boldsymbol{\pi}_2) + d_{i,j})} \\ &\quad \left. + \sum_{\boldsymbol{\pi}_3 \in \boldsymbol{\Pi}_{i-1,j-1}} e^{-\beta (s(\boldsymbol{\pi}_3) + d_{i,j})} \right), \end{aligned} \quad (36)$$

$$\begin{aligned}
 &= -\frac{1}{\beta} \ln e^{-\beta d_{i,j}} \left(\sum_{\pi_1 \in \Pi_{i-1,j}} e^{-\beta s(\pi_1)} \right. \\
 &\quad + \sum_{\pi_2 \in \Pi_{i,j-1}} e^{-\beta s(\pi_2)} \\
 &\quad \left. + \sum_{\pi_3 \in \Pi_{i-1,j-1}} e^{-\beta s(\pi_3)} \right), \quad (37) \\
 &= d_{i,j} + -\frac{1}{\beta} \ln \left(\sum_{\pi_1 \in \Pi_{i-1,j}} e^{-\beta s(\pi_1)} \right. \\
 &\quad + \sum_{\pi_2 \in \Pi_{i,j-1}} e^{-\beta s(\pi_2)} \\
 &\quad \left. + \sum_{\pi_3 \in \Pi_{i-1,j-1}} e^{-\beta s(\pi_3)} \right). \quad (38)
 \end{aligned}$$

Using the expression in equation (28) again, we obtain

$$\sum_{\pi_1 \in \Pi_{i-1,j}} e^{-\beta s(\pi_1)} = e^{\ln \sum_{\pi_1 \in \Pi_{i-1,j}} e^{-\beta s(\pi_1)}}, \quad (39)$$

$$\begin{aligned}
 &= e^{-\beta \min_{\Omega} (s(\pi_1) | \pi_1 \in \Pi_{i-1,j})}, \quad (40) \\
 &= e^{-\beta r_{i-1,j}}. \quad (41)
 \end{aligned}$$

The similar expressions for the sums over $\pi_2 \in \Pi_{i,j-1}$ and $\pi_3 \in \Pi_{i-1,j-1}$ can be derived in the same manner. Substituting (41) into (38), we have

$$r_{i,j} = d_{i,j} + \min_{\Omega} (r_{i-1,j}, r_{i,j-1}, r_{i-1,j-1}). \quad (42)$$

By recursively applying equation (42) for $i = 1 \dots, n$ and $j = 1, \dots, m$, we can arrive at equation (33), completing the proof. \square

APPENDIX C FORWARD-BACKWARD ALGORITHM

To compute $e_{i,j}$ in equation (12), we use the forward-backward algorithm, which is originally introduced in [10]. The details are shown in Algorithm 3. The algorithm indeed computes the gradient matrix \mathbf{E} , where $e_{i,j}$ is the element at position (i, j) , of the generalized smooth DTW w.r.t. the distance matrix \mathbf{D} . It includes a forward step and a backward step. Both of them perform constant-time operations in nm times. Therefore, the computational complexity of the algorithm is $O(nm)$. In addition, during the computation, the algorithm stores several matrices whose largest size is $3nm$. Thus, its space complexity is also $O(nm)$. Note that when the squared ℓ_2 norm is used as regularization in DTW_{Ω} , $q_{i,j}$ become sparse because of equation (31). This then induces the sparsity in \mathbf{E} , further reducing the complexity of the algorithm in terms of both time and space.

APPENDIX D UPDATE RULE FOR CONSENSUS LABEL SEQUENCE

In this section, we provide the derivation of the update rule for the consensus label sequence in equation (21). By adding an extra term $\xi \|\mathbf{Z}\mathbf{Z}^T - \mathbf{I}\|_F^2$ and introducing a Lagrange

Algorithm 3 Forward-Backward Algorithm

Input: Distance matrix $\mathbf{D} \in \mathbb{R}^{n \times m}$

Output: Gradient matrix $\mathbf{E} = \frac{\partial \text{DTW}_{\Omega}(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{D}} \in \mathbb{R}^{n \times m}$.

▷ Forward pass:

1: $s'_{0,0} = 0, s'_{i,0} = s'_{0,j} = \infty \forall i, j$.

2: **for** $i = 1, \dots, n$ and $j = 1, \dots, m$ **do**

3: $s'_{i,j} = d_{i,j} + \min_{\Omega} (s'_{i-1,j}, s'_{i,j-1}, s'_{i-1,j-1})$

4: $q_{i,j} = \nabla \min_{\Omega} (s'_{i-1,j}, s'_{i,j-1}, s'_{i-1,j-1}) \in \mathbb{R}^3$

5: **end for**

▷ Backward pass:

6: $q_{i,m+1} = q_{n+1,j} = \mathbf{0}_3, e_{i,m+1} = e_{n+1,j} = 0 \forall i, j$.

7: $q_{n+1,m+1} = [0, 1, 0], e_{n+1,m+1} = 1$.

8: **for** $i = 1, \dots, n$ and $j = 1, \dots, m$ **do**

9: $e_{i,j} = q_{i,j+1,1}e_{i,j+1} + q_{i+1,j+1,2}e_{i+1,j+1} + q_{i+1,j,3}e_{i+1,j}$

10: **end for**

multiplier matrix $\Psi \in \mathbb{R}^{c \times n}$, we have the following Lagrange function

$$\begin{aligned}
 \mathcal{L}(\mathbf{Z}, \Psi) &= \sum_{k=1}^v \text{DTW}_{\Omega}(\mathbf{Z}, \mathbf{Z}^{(k)}) + \xi \|\mathbf{Z}\mathbf{Z}^T - \mathbf{I}\|_F^2 \\
 &\quad + \text{Tr}(\Psi^T \mathbf{Z}). \quad (43)
 \end{aligned}$$

Taking the derivative of $\mathcal{L}(\mathbf{Z}, \Psi)$ w.r.t. \mathbf{Z} and setting it to zero, we obtain

$$\begin{aligned}
 \frac{\partial \mathcal{L}(\mathbf{Z}, \Psi)}{\partial \mathbf{Z}} &= \sum_{k=1}^v \frac{\partial \text{DTW}_{\Omega}(\mathbf{Z}, \mathbf{Z}^{(k)})}{\partial \mathbf{Z}} \\
 &\quad + 4\xi(\mathbf{Z}\mathbf{Z}^T - \mathbf{I})\mathbf{Z} + \Psi = 0. \quad (44)
 \end{aligned}$$

Then

$$\Psi = 4\xi \mathbf{Z} - 4\xi \mathbf{Z}\mathbf{Z}^T \mathbf{Z} - \mathbf{G}, \quad (45)$$

where $\mathbf{G} = \sum_{k=1}^v \frac{\partial \text{DTW}_{\Omega}(\mathbf{Z}, \mathbf{Z}^{(k)})}{\partial \mathbf{Z}}$. According to the Karush–Kuhn–Tucker condition [66], i.e. $\psi_{i,j} z_{i,j} = 0$, we can arrive at the following equation:

$$[4\xi \mathbf{Z} - 4\xi \mathbf{Z}\mathbf{Z}^T \mathbf{Z} - \mathbf{G}]_{i,j} z_{i,j} = 0. \quad (46)$$

Then, we obtain the update rule for \mathbf{Z} :

$$z_{i,j} \leftarrow z_{i,j} \frac{[4\xi \mathbf{Z}]_{i,j}}{[\mathbf{G} + 4\xi \mathbf{Z}\mathbf{Z}^T \mathbf{Z}]_{i,j}}. \quad (47)$$

REFERENCES

- [1] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, p. 321, Dec. 1936.
- [2] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1247–1255.
- [3] W. Wang, R. Arora, K. Livescu, and N. Srebro, "Stochastic optimization for deep CCA via nonlinear orthogonal iterations," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2015, pp. 688–695.
- [4] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004.

- [5] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, vol. 14. Englewood Cliffs, NJ, USA: PTR, 1993.
- [6] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 689–696.
- [7] T. Doan and T. Atsuhiro, "Deep multi-view learning from sequential data without correspondence," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
- [8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 8, no. 9, pp. 1735–1780, 1997.
- [9] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.
- [10] A. Mensch and M. Blondel, "Differentiable dynamic programming for structured prediction and attention," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 1–65.
- [11] J. M. Danskin, "The theory of max-min, with applications," *SIAM J. Appl. Math.*, vol. 14, no. 4, pp. 641–664, 1966.
- [12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
- [14] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
- [15] J. Nocedal and S. J. Wright, "Nonlinear equations," in *Numerical Optimization*. New York, NY, USA: Springer, 2006, pp. 270–302.
- [16] L. Bottou, "Stochastic gradient learning in neural networks," *Proc. NeuroNmes*, vol. 91, no. 8, p. 12, Nov. 1991.
- [17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [18] J. Ye, Z. Zhao, and M. Wu, "Discriminative k-means for clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1649–1656.
- [19] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and Out-of-Sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, Nov. 2011.
- [20] F. De la Torre, "A least-squares framework for component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1041–1055, Jun. 2012.
- [21] G. Trigeorgis, M. A. Nicolaou, B. W. Schuller, and S. Zafeiriou, "Deep canonical time warping for simultaneous alignment and representation learning of sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1128–1138, May 2018.
- [22] F. Zhou and F. Torre, "Canonical time warping for alignment of human behavior," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2286–2294.
- [23] K. Kawano, S. Koide, and T. Kutsuna, "Canonical soft time warping," in *Proc. Asian Conf. Mach. Learn.*, 2019, pp. 551–566.
- [24] Y. Panagakis, M. A. Nicolaou, S. Zafeiriou, and M. Pantic, "Robust canonical time warping for the alignment of grossly corrupted sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 540–547.
- [25] Y. Panagakis, M. A. Nicolaou, S. Zafeiriou, and M. Pantic, "Robust correlated and individual component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1665–1678, Aug. 2016.
- [26] L. Zafeiriou, E. Antonakos, S. Zafeiriou, and M. Pantic, "Joint unsupervised deformable spatio-temporal alignment of sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3382–3390.
- [27] C. Jia, M. Shao, and Y. Fu, "Sparse canonical temporal alignment with deep tensor decomposition for action recognition," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 738–750, Feb. 2017.
- [28] B. Rastii, D. Hong, R. Hang, P. Ghamisi, X. Kang, J. Chanussot, and J. A. Benediktsson, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep (overview and toolbox)," *IEEE Geosci. Remote Sens. Mag.*, early access, Apr. 29, 2020, doi: [10.1109/MGRS.2020.2979764](https://doi.org/10.1109/MGRS.2020.2979764).
- [29] L. Nie, Y. Wang, X. Zhang, X. Huang, and Z. Luo, "Enhancing temporal alignment with autoencoder regularization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 4873–4879.
- [30] X. Zhang, L. Nie, L. Lan, X. Huang, and Z. Luo, "Stacked marginal time warping for temporal alignment," *Neural Process. Lett.*, vol. 49, no. 2, pp. 711–735, Apr. 2019.
- [31] C. Wang and S. Mahadevan, "Manifold alignment using procrustes analysis," in *Proc. 25th Int. Conf. Mach. Learn. ICML*, 2008, pp. 1120–1127.
- [32] C. Wang and S. Mahadevan, "Manifold alignment preserving global geometry," in *Proc. IJCAI*, 2013, pp. 1743–1749.
- [33] D. Hong, N. Yokoya, N. Ge, J. Chanussot, and X. X. Zhu, "Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 193–205, Jan. 2019.
- [34] T. A. Abeo, X.-J. Shen, E. D. Ganaa, Q. Zhu, B.-K. Bao, and Z.-J. Zha, "Manifold alignment via global and local structures preserving PCA framework," *IEEE Access*, vol. 7, pp. 38123–38134, 2019.
- [35] C. Wang and S. Mahadevan, "Manifold alignment without correspondence," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1273–1278.
- [36] W. Li, J. Xue, Y. Chen, X. Zhang, C. Tang, Q. Zhang, and Y. Gao, "Fuzzy granule manifold alignment preserving local topology," *IEEE Access*, vol. 8, pp. 178695–178705, 2020.
- [37] D. Tuia, M. Volpi, and G. Camps-Valls, "Unsupervised alignment of image manifolds with centrality measures," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 912–917.
- [38] L. Ma, C. Luo, J. Peng, and Q. Du, "Unsupervised manifold alignment for cross-domain classification of remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 10, pp. 1650–1654, Oct. 2019.
- [39] Z. Cui, H. Chang, S. Shan, and X. Chen, "Generalized unsupervised manifold alignment," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2429–2437.
- [40] K. Fan, A. Mian, W. Liu, and L. Li, "Unsupervised manifold alignment using soft-assign technique," *Mach. Vis. Appl.*, vol. 27, no. 6, pp. 929–942, Aug. 2016.
- [41] D. Gong and G. Medioni, "Dynamic manifold warping for view invariant action recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 571–578.
- [42] H. T. Vu, C. Carey, and S. Mahadevan, "Manifold warping: Manifold alignment over time," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, p. 8.
- [43] H. Le, T. Tran, and S. Venkatesh, "Dual memory neural computer for asynchronous two-view sequential learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1637–1645.
- [44] 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.
- [45] H. Wang, W. Yang, C. Yuan, H. Ling, and W. Hu, "Human activity prediction using temporally-weighted generalized time warping," *Neurocomputing*, vol. 225, pp. 139–147, Feb. 2017.
- [46] M. A. Hasan, "On multi-set canonical correlation analysis," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2009, pp. 1128–1133.
- [47] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1083–1092.
- [48] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.
- [49] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.
- [50] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.
- [51] Cuturi, Marco, and Mathieu Blondel, "Soft-DTW: A differentiable loss function for time-series," in *Proc. Int. Conf. Mach. Learn.*, pp. 894–903, 2017.
- [52] S. R. Eddy, "Profile hidden Markov models," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, Oct. 1998.
- [53] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
- [54] J. R. Westbury and J. G. T. Dembowski, "X-ray microbeam speech production database user's handbook," in *Waisman Center on Mental Retardation Human Development*. Madison, WI, USA: Univ. Wisconsin, 1994.
- [55] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.
- [56] M. Hoai and F. De la Torre, "Max-margin early event detectors," *Int. J. Comput. Vis.*, vol. 107, no. 2, pp. 191–202, Apr. 2014.
- [57] C. R. Maurer, R. Qi, and V. Raghavan, "A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 265–270, Feb. 2003.
- [58] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape representation and classification using the Poisson equation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1991–2005, Dec. 2006.

- [59] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2005, p. 5.
- [60] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, Jul. 2009.
- [61] L. Sha, X. Zhang, F. Qian, B. Chang, and Z. Sui, "A multi-view fusion neural network for answer selection," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5422–5429.
- [62] Y. Deng, Y. Xie, Y. Li, M. Yang, N. Du, W. Fan, K. Lei, and Y. Shen, "Multi-task learning with multi-view attention for answer selection and knowledge base question answering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6318–6325.
- [63] P. Brucker, "An $O(n)$ algorithm for quadratic knapsack problems," *Oper. Res. Lett.*, vol. 3, no. 3, pp. 163–166, 1984.
- [64] P. M. Pardalos and N. Kover, "An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds," *Math. Program.*, vol. 46, nos. 1–3, pp. 321–328, Jan. 1990.
- [65] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1 -ball for learning in high dimensions," in *Proc. 25th Int. Conf. Mach. Learn. - ICML*, 2008, pp. 272–279.
- [66] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.



DOAN PHONG TUNG received the B.S. degree in computer engineering from the Hanoi University of Science and Technology in 2014. He is currently pursuing the Ph.D. degree with the Graduate University for Advanced Studies, SOKENDAI, Japan. His current research interests include representation learning, sequential data, multiview learning, and combinatorial optimization.



ATSUHIRO TAKASU (Member, IEEE) received the B.E., M.E., and Dr.Eng. degrees from The University of Tokyo, Japan, in 1984, 1986, and 1989, respectively. He is currently a Professor with the National Institute of Informatics, Japan. His research interests are data engineering and data mining. He is a member of the ACM, IEICE, IPSJ, and JSAI.

• • •