Deep Multiview Learning From Sequentially Unaligned Data

DOAN PHONG TUNG and ATSUHIRO TAKASU, (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: $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d_x \times n}$ and $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d_y \times n}$, where $(x_i, y_i)$ 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...
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 nonlinear [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 ℓ₂ 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. 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 A is denoted by a_{i,j} or [A]_{i,j}. We denote the Frobenius inner product between A and B as \langle A, B \rangle := \sum_{i,j} a_{i,j}b_{i,j}. 0_d is a vector of dimension d whose all elements are zeros. The expression \( x \in \mathbb{R}^d \) indicates that vector x has d elements, each of which is greater than or equal to zero. The norm \( \ell_p \) of a vector x, where \( p \in \{1, 2\} \) in this article, is \( ||x||_p = (|x_1|^p + \ldots + |x_d|^p)^{\frac{1}{p}} \).

### 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

\(^1\)More discussions are given in III-C
sequences $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$ and $Y = [y_1, \ldots, y_m] \in \mathbb{R}^{d \times m}$, a distance matrix $D(X, Y) \in \mathbb{R}^{n \times m}$ is defined such that the element at position $(i, j)$, denoted by $d_{ij}$, is the squared distance, i.e., $d_{ij} = ||x_i - y_j||_2^2$. The DTW algorithm constructs a cumulative sum matrix $S(X, Y)$ using the following recursive formulas:

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

(2)

The DTW distance between the two sequences is then defined as follows:

$$S(X, Y) = \min_{\pi} \sum_{i,j} s_{i,j}$$

where $\pi = \{(i_1^*, j_1^*), \ldots, (i_p^*, j_p^*)\}$ 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 $s_{n,m} = d_{1,1} + \cdots + d_{n,m}$ 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 as follows:

$$\min_{s} |\gamma| = \arg\min_{\gamma} \langle \gamma, \eta \rangle + \frac{1}{\beta} \Omega(\gamma).$$

(5)

The equation shows that the smooth min operator also depends on the selection of the regularization function $\Omega(\gamma)$. Shannon entropy ($\sum_{i=1}^{k} \gamma_i \ln \gamma_i$) or squared $\ell_2$ 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:

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

(6)

where the generalized smooth approximation of DTW is defined by $D \text{TW}_\Omega(X, Y) := s_{n,m}$. Note that we can have different versions of DTW$_\Omega$, e.g., DTW$_\Omega$-entropy or DTW$_\Omega$-squared $\ell_2$, depending on selection of the regularization $\Omega(\gamma)$. Figure 1 (d) and (e) show the cumulative sum matrices of DTW$_\Omega$-entropy and DTW$_\Omega$-squared $\ell_2$, respectively. The generalized smooth DTW distance is different from the original DTW because it is differentiable. Furthermore, by minimizing DTW$_\Omega$, 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 $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d_x \times n}$ and $Y = [y_1, \ldots, y_m] \in \mathbb{R}^{d_y \times m}$ from different representation spaces ($d_x \neq d_y$), our method maps them into a shared subspace: $Z_x = [z_1, \ldots, z_n] = f_x(X, \theta_x) \in \mathbb{R}^{d_x \times n}$ and $Z_y = [z_2, \ldots, z_m] = f_y(Y, \theta_y) \in \mathbb{R}^{d_y \times m}$, where $f_x(\cdot, \cdot)$ and $f_y(\cdot, \cdot)$ are projection functions and $\theta_x$ and $\theta_y$ denotes their parameters.
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 $z_{x_i}$. 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 the parameters $\theta_x$ and $\theta_y$ are now equivalent to collections of all the weights matrices of the corresponding DNNs.

**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:

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

(7)

where the two regularization terms are of the following form:

$$\mathcal{L}_v(Z^v) = \sum_{i=1}^{d} \sum_{j \neq i} |c^v_{ij}|,$$

(8)

where $v \in \{x, y\}$ and $c^v_{ij}$ 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 $\theta_x$ and $\theta_y$ can be trained using the gradient-based method. To compute the gradient of $\mathcal{L}_{GSCA}$ w.r.t. all
the parameters $\theta_x$ and $\theta_y$, we compute its gradients w.r.t. the outputs $Z'$ and $Z'$ 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
\[
\frac{\partial L_{GSCA}}{\partial Z^x} = \frac{\partial DTW_{\Omega}(Z^x, Z^y)}{\partial Z^x} + \lambda_1 \frac{\partial L_\lambda(Z^x)}{\partial Z^x}.
\]
The gradient of the generalized smooth DTW w.r.t. $Z^x$ can be computed as
\[
\frac{\partial DTW_{\Omega}(Z^x, Z^y)}{\partial Z^x} = [\frac{\partial s'_{n,m}}{\partial z_i^x}, \ldots, \frac{\partial s'_{n,m}}{\partial z_n^x}],
\]
where
\[
\frac{\partial s'_{n,m}}{\partial z_i^x} = \sum_{j=1}^m \frac{\partial s'_{n,m}}{\partial d_{i,j}} \frac{\partial d_{i,j}}{\partial z_i^x} = 2 \sum_{j=1}^m e_{i,j} (z_{i,j}^x - z_{i,j}^y)
\]
for $i = 1, \ldots, n$. (12)

In equation (12), we abused the notations defined in Section II, where $s'_{n,m} := DTW_{\Omega}(Z^x, Z^y)$ and $d_{i,j} := ||z_{i,j}^x - z_{i,j}^y||_2^2$. 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 $L_\lambda$ w.r.t. $Z^x$ can be computed as
\[
\frac{\partial L_\lambda(Z^x)}{\partial Z^x} = H^x Z^x,
\]
where $H^x \in \mathbb{R}^{d \times d}$, whose elements are defined as
\[
h^x_{i,j} = \begin{cases} 1 & \text{if } c^x_{i,j} > 0 \\ 0 & \text{if } i = j \text{ or } c^x_{i,j} = 0 \\ -1 & \text{if } c^x_{i,j} < 0. \\ \end{cases}
\]

The gradient $\frac{\partial L_{GSCA}}{\partial Z^x}$ 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 $\Omega(\gamma)$ in $DTW_{\Omega}$, 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 $\ell_2$ norm is selected as GSCA-s. In fact, the objective of GSCA-e is equivalent to that of DSCA because $DTW_{\Omega=\text{entropy}}$ is equal to $DTW_\beta$, 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 $DTW_{\Omega}$ 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 $DTW_\beta$ w.r.t. $Z^x$ and $Z^y$ associate with a summation over all feasible alignment matrices which requires $O(n^2m^2)$ in both computational time and memory storage. Note that by selecting squared $\ell_2$ 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 $DTW_{\Omega=\text{squared}}$ is a nonexact approximation of the original DTW [10]. $DTW_{\Omega=\text{entropy}}$, instead, can exactly approximate DTW. It converges to the original warping distance as $\beta \to \infty$.

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 $g_\Omega(Z^x, \Phi_x)$ and $g_\Omega(Z^y, \Phi_y)$ denote the functions that map the representation sequences $Z^x$ and $Z^y$ back to the original spaces, where $\Phi_x$ and $\Phi_y$ are their corresponding parameters. Then, the objective of GSCAE is as follows
\[
L_{GSCAE} = L_{GSCA} + \lambda \left( \frac{1}{n} ||X - g_\Omega(Z^x, \Phi_x)||_2^2 + \frac{1}{m} ||Y - g_\Omega(Z^y, \Phi_y)||_2^2 \right),
\]
where $\lambda > 0$ is a trade-off parameter. Similar to projection functions in GSCA, $g_\Omega(\cdot, \cdot)$ and $g_\Omega(\cdot, \cdot)$ can also be parameterized by deep feed-forward neural networks or RNNs. Diagrams of GSCAE are illustrated in Figure 3.

Algorithm I 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 $\epsilon$.

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

1: for $t = 1, \ldots, T$ do
2: random sample subsequence $Z_i^y(t)$ of length $na$;
3: random sample subsequence $Z_i^y(t)$ of length $ma$;
4: $C_i^x(t) = \rho C_i^x(t-1) + (1 - \rho) \frac{1}{a} \sum_{j=0}^{a-1} Z_i^y(t)_{ij}^y$;
5: $C_i^y(t) = \rho C_i^y(t-1) + (1 - \rho) \frac{1}{a} \sum_{j=0}^{a-1} Z_i^x(t)_{ij}^x$;
6: compute $\frac{\partial L_{GSCA}}{\partial Z_i^x(t)}$ and $\frac{\partial L_{GSCA}}{\partial Z_i^y(t)}$;
7: compute gradient $\nabla^x \theta$ using backpropagation;
8: $\Delta \theta(t) = \mu \Delta \theta(t-1) + \epsilon \nabla^x \theta$;
9: $\theta(t) = \theta(t-1) + \Delta \theta(t)$;
10: end for

2Equation (13) in [7]
By minimizing $L_{GSCA}$, 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 $\lambda$ 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. $\theta_x$ and $\theta_y$, which are associated with both $L_{GSCA}$ and autoencoder parts. Furthermore, we also need to compute the gradients w.r.t. $\Phi_x$ and $\Phi_y$, 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 $\Omega(y)$ in DTW$_\Omega$. We denote them as GSCAE-e if $\Omega(y)$ is Shannon entropy and GSCAE-s when squared $\ell_2$ 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 $X^{(k)} \in \mathbb{R}^{d_x(k) \times n}$ for $k = 1, \ldots, v$, we assume that each sample of these sequences belongs to one of $c$ disjoint classes. The cluster assignment is often denoted by a matrix $F(k) = [f_1^{(k)}, \ldots, f_{n(k)}^{(k)}] \in \mathbb{R}^{c \times n(k)}$, where $f_i^{(k)}$ is the cluster indicator vector for sample $x_i^{(k)}$ in the sequence $X^{(k)}$. As in [18]–[20], for each view, we define a scaled cluster indicator matrix

$$\tilde{F}^{(k)} = [\tilde{f}_1^{(k)}, \ldots, \tilde{f}_{n(k)}^{(k)}] = (F(k)^\top F(k))^{-1/2} F(k)^\top.$$  

It turns out that

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

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

\begin{figure*}[h]
\centering
\includegraphics[width=\textwidth]{fig3.png}
\caption{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. $X = [x_1^{(k)}, \ldots, x_{n(k)}^{(k)}]$ and $Y = [y_1^{(k)}, \ldots, y_{n(k)}^{(k)}]$ denote the reconstructed inputs.}
\end{figure*}
the representation sequences in the new space are possibly unequal in length and sample-wise mismatched, we introduce a consensus label sequence $Z \in \mathbb{R}^{v \times n}$ of a prespecified length $n$ and minimize its DTW distances with all sequences $Z^{(k)}$. The optimization problem of GMSA is as follows:

$$\min_{Z, \theta^{(1)}, \ldots, \theta^{(v)}} \sum_{k=1}^{v} \text{DTW}_\Omega(Z, Z^{(k)}) + \sum_{k=1}^{v} \lambda_k \mathcal{L}_k(Z^{(k)}),$$
subject to $Z \geq 0$ and $ZZ^\top = I$. (18)

where $\lambda_k$ 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 $Z$ 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 $Z$ 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. $Z^{(k)}$ for $k = 1, \ldots, v$ as follows:

$$\frac{\partial \mathcal{L}_{GMSA}(Z^{(k)})}{\partial Z^{(k)}} = \frac{\partial \text{DTW}_\Omega(Z, Z^{(k)})}{\partial Z^{(k)}} + \lambda_k \frac{\partial \mathcal{L}_k(Z^{(k)})}{\partial Z^{(k)}}.$$ (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 $k = 1, \ldots, v$.

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

$$\min_{Z} \sum_{k=1}^{v} \text{DTW}_\Omega(Z, Z^{(k)})$$
subject to $Z \geq 0$ and $ZZ^\top = I$. (20)

By adding an extra penalty term $\xi \|ZZ^\top - I\|^2_2$ to the objective of problem (20), we can remove the orthogonal constraint. Denote $G = \sum_{k=1}^{v} \frac{\partial \text{DTW}_\Omega(Z, Z^{(k)})}{\partial Z^{(k)}}$, then the update rule for $Z$ is as follows:

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

Algorithm 2 Alternating Optimization Algorithm for GMSA

**Input:** Input data sequences $X^{(k)}$ for $k = 1, \ldots, v$.

**Output:** The optimal DNNs’ parameters $\theta^{*}$ and consensus sequence $Z^{*}$.

1. for $k = 1, \ldots, v$
2. Run k-means on $X^{(k)}$ to generate cluster indicator matrix $F^{(k)}$;
3. Initialize $Z^{(k)} = (F^{(k)}_\top F^{(k)} - \zeta F^{(k)}_\top)_{\theta}$;
4. Initialize $\theta^{(k)} = \arg\min_{\theta} \|Z^{(k)} - f_k(X^{(k)}, \Theta)\|^2_2$;
5. end for
6. repeat
7. Update the consensus $Z$ using equation (21);
8. Update the DNNs’ parameters $\theta$ using a stochastic algorithm;
9. Recompute $Z^{(k)} = f_k(X^{(k)}, \theta^{(k)})$ for $k = 1, \ldots, v$;
10. until convergence

To guarantee the orthogonality of $Z$, we set $\xi$ 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 $Z$, 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 $X^{(k)}$ to generate their cluster indicator matrices $F^{(k)}$ for $k = 1, \ldots, v$. The initial value of $Z^{(k)}$’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; and compare GMSA with the following multiview method:  
- 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.

We note that DTW_Ω in the objective of our methods has different versions depending on the selection of the regularization functions Ω(γ). Therefore, we add suffixes -e and -s to our methods, indicating that Ω(γ) is Shannon entropy and squared ℓ2 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 parameter between the autoencoder regularization term and alignment measurement results on the tuning set for each method. We select the optimal parameters that return the best evaluation results at \(\{\lambda_1 = \lambda_2 = \ldots = \lambda_5\}\), and the value is chosen using grid search. Let \(n_{a}, n_{s}, n_{l}\) 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 \(\{n_{a}, \max(n_{a}, 0.5n_{a}), \max(n_{a}, 0.75n_{a}), n_{a}, \min(n_{l}, 1.25n_{a}), \min(n_{l}, 1.5n_{a}), n_{l}\}\). 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 \(\text{iter\_max} = 50\).

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 \([-\pi/3, \pi/3]\). 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 \([x_i, y_i, l_i]|1 \leq i \leq 5 \times 10^4, x_i \in \mathbb{R}^{784}, y_i \in \mathbb{R}^{576}, l_i \in \{0, \ldots, 9\}\) of the new training

\[\begin{array}{cccccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
\end{array}\]

\[\begin{array}{cccccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
\end{array}\]
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 \((5 \times 10^4)\)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 \( x_{i=l} \) 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 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 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.

### 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,

### 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 variances on projections of the test set are reported.

| Method     | ACC (%) | NMI (%) | Error (%) |
|------------|---------|---------|-----------|
| Inputs     | 45.14 (5.27) | 48.06 (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) | 54.26 (2.90) | 28.03 (3.14) |
| MATW       | 69.18 (4.16) | 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) | \textbf{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) |
Algorithm 1). Figure 7 shows that the computing gradients of the units are $1500$ layers whose activation functions are ReLU, and the numbers connected networks. These networks consist of three hidden two long sequences and feed them separately to two fully we concatenate two views of the training utterances into in the data for training the RNNs. For DCTW and DSCA, thereby we can take better advantage of the sequential nature 1415 training utterances to compute the stochastic gradient. and articulatory input sequences that correspond to one of the Algorithm 1, at each iteration, we randomly sample acoustic LSTM networks. Note that while training these models using construction in GSCAE is performed by the symmetric deep view recon-

---

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).

### Table 2. Phone class separation on the projections of the acoustic view learned by different methods.

| 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 (3.22) | 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) |

---

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

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 $\Omega(q)$ in DTW$_2$ to be squared $\ell_2$ norm. However, as shown in Table 2, the class separation results were slightly decreased. Because DTW$_2$-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
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 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 \( z_i^{(k)} \) 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...
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 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 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 4) |
| GMSA-e  | 90.63 (View 4) | 90.91 (View 4) | 2.88 (View 5) |
| GMSA-s  | 85.34 (View 4) | 87.17 (View 4) | 3.79 (View 3) |

---

![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.](image-url)
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 (w/o Binary view)     | 88.03   | 90.02   | 5.04      |
|         | GMSA-e (w/o Euclidean distance transform view) | 85.41 (↓ 2.62) | 87.03 (↓ 2.99) | 7.17 (↑ 2.13) |
|         | GMSA-e (w/o Solution of the Poisson equation view) | 85.26 (↓ 2.77) | 85.82 (↓ 4.20) | 6.49 (↑ 1.45) |
|         | GMSA-e (w/o view 1)          | 87.42 (↓ 0.61) | 87.67 (↓ 3.24) | 3.91 (↑ 0.63) |
|         | GMSA-e (w/o view 2)          | 89.05 (↓ 2.05) | 89.93 (↓ 1.98) | 3.35 (↑ 0.47) |
|         | GMSA-e (w/o view 3)          | 89.17 (↓ 2.06) | 88.62 (↓ 2.29) | 3.28 (↑ 0.40) |
|         | GMSA-e (w/o view 4)          | 85.56 (↓ 3.87) | 85.15 (↓ 4.87) | 5.13 (↑ 2.23) |
|         | GMSA-e (w/o view 5)          | 88.01 (↓ 2.62) | 87.84 (↓ 3.07) | 4.03 (↑ 1.15) |

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 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,
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:

$$\min_{\Omega(\gamma)}(\gamma) := \min_{\gamma \in \Delta^k} \gamma \ln \gamma + \frac{1}{\beta} \Omega(\gamma),$$

where the regularization term $\Omega(\gamma)$ must be a strictly convex function [9]. Two widely used functions are Shannon entropy and squared $\ell_2$ norm.

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

$$\min_{\gamma \in \Delta^k} \sum_{i=1}^{k} \gamma_i \ln \gamma_i + \frac{1}{\beta} \sum_{i=1}^{k} \gamma_i \ln \gamma_i.$$  \hfill (23)

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

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

$$+ \lambda_2 \sum_{i=1}^{k} \gamma_i.$$ \hfill (24)

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

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

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

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

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

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

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

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

$$\nabla \min_{\Omega(\gamma)}(\gamma) = \frac{e^{-\beta \eta_i}}{\sum_{j=1}^{k} e^{-\beta \eta_j}}.$$ \hfill (29)

**Squared $\ell_2$ Norm**: When $\Omega(\gamma) = \frac{1}{2} \sum_{i=1}^{k} \gamma_i^2$, the smooth min

$$\min_{\gamma \in \Delta^k} \sum_{i=1}^{k} \gamma_i \ln \gamma_i + \frac{1}{2 \beta} \sum_{i=1}^{k} \gamma_i^2.$$ \hfill (30)

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

$$\gamma^* = \arg \min_{\gamma \in \Delta^k} \frac{1}{2 \beta} \sum_{i=1}^{k} \gamma_i^2.$$ \hfill (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 $\Pi$ denote the set of all warping paths

$$\pi = (i_1, j_1), \ldots, (i_p, j_p),$$ \hfill (32)

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

$$\text{DTW}_{\Omega}(X, Y) = \min_{\pi \in \Pi} \sum_{i=1}^{p} e^{-\beta s(\pi)}.$$ \hfill (33)

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

$$r_{i,j} = -\frac{1}{\beta} \ln \sum_{\pi_0 \in \Pi_{i,j}} e^{-\beta s(\pi_0)}.$$ \hfill (34)

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

$$r_{i,j} = \min_{\Omega} \left(s(\pi_0) \mid \pi_0 \in \Pi_{i,j}\right).$$ \hfill (35)

We can rewrite Equation (34) as follows:

$$r_{i,j} = \frac{1}{\beta} \ln \left( \sum_{\pi_0 \in \Pi_{i,j}} e^{-\beta s(\pi_0)} \right)$$

$$+ \sum_{\pi_1 \in \Pi_{i,j-1}} e^{-\beta s(\pi_1)+d_{i,j}}$$

$$+ \sum_{\pi_2 \in \Pi_{i-1,j}} e^{-\beta s(\pi_2)+d_{i,j}}.$$ \hfill (36)
The similar expressions for the sums over $\pi$

In this section, we provide the derivation of the update rule of the algorithm in terms of both time and space. Induces the sparsity in $q$ when the squared $nm$ computes the gradient matrix $\nabla F$, where $e_i$ is the element at position $(i, j)$, of the generalized smooth DTW w.r.t. the distance matrix $D$. 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 $\ell_2$ norm is used as regularization in DTW, $q_{i,j}$ become sparse because of equation (31). This then induces the sparsity in $E$, 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 $\xi \|ZZ^T - I\|_F^2$ and introducing a Lagrange multiplier matrix $\Psi \in \mathbb{R}^{c \times n}$, we have the following Lagrange function

$$L(Z, \Psi) = \sum_{k=1}^{v} \text{DTW}_\Omega(Z, Z^{(k)}) + \xi \|ZZ^T - I\|_F^2 + \text{Tr}(\Psi^T Z).$$

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

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

Then

$$\Psi = 4\xi Z - 4\xi ZZ^T Z - G. \quad (45)$$

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

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

Then, we obtain the update rule for $Z$:

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

**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. 2659–2664, Dec. 2004.
[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. Kovoor, “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 $\ell_1$-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.

* * *