Linear and Deep Order-Preserving Wasserstein Discriminant Analysis

Bing Su, Jiahuan Zhou, *Member, IEEE*, Ji-Rong Wen, *Senior Member, IEEE*, and Ying Wu, *Fellow, IEEE*

Abstract—Supervised dimensionality reduction for sequence data learns a transformation that maps the observations in sequences onto a low-dimensional subspace by maximizing the separability of sequences in different classes. It is typically more challenging than conventional dimensionality reduction for static data, because measuring the separability of sequences involves non-linear procedures to manipulate the temporal structures. In this paper, we propose a linear method, called order-preserving Wasserstein discriminant analysis (OWDA), and its deep extension, namely DeepOWDA, to learn linear and non-linear discriminative subspace for sequence data, respectively. We construct novel separability measures between sequence classes based on the order-preserving Wasserstein (OPW) distance to capture the essential differences among their temporal structures. Specifically, for each class, we extract the OPW barycenter and construct the intra-class scatter as the dispersion of the training sequences around the barycenter. The inter-class distance is measured as the OPW distance between the corresponding barycenters. We learn the linear and non-linear transformations by maximizing the inter-class distance and minimizing the intra-class scatter. In this way, the proposed OWDA and DeepOWDA are able to concentrate on the distinctive differences among classes by lifting the geometric relations with temporal constraints. Experiments on four 3D action recognition datasets show the effectiveness of OWDA and DeepOWDA.

Index Terms—Optimal transport, order-preserving Wasserstein distance, barycenter, dimensionality reduction, sequence classification

1 Introduction

The sequence classification problem arises in a wide range of real-world applications. A sequence is comprised of a series of ordered observations, where each individual observation is generally of no special interest, but the sequence as a whole represents the target object. The observations in the same sequence are not independent and their relationship reveals the temporal structure of the sequence. For instance, all ordered frames in an action video as a whole represent the action and these frames are temporally related. Low-dimensional and discriminative representations of frame-wide observations in sequences are crucial to reduce the complexity of the subsequent modeling and improve the classification performance. Supervised dimensionality reduction for sequence data (DRS) attempts to learn such low-dimensional discriminative representations by transforming the observations in the noisy high-dimensional space to a subspace.

In this paper, we propose a linear supervised DRS method by using the Fisher criterion to maximize the ratio of the inter-sequence-class separability to the intra-sequence-class dispersion. For each class, we extract the order-preserving Wasserstein barycenter and measure the dispersion of training sequences around the barycenter w.r.t. the order-preserving Wasserstein (OPW) distance [1], [2]. We measure the inter-class separability between two classes as the OPW distance between the corresponding barycenters. In this way, the intra-class and inter-class separabilities are uniformly measured with the OPW distance. We employ OPW to perform temporal alignment between sequences and barycenters with different local durations, lengths, and temporal distortions. Through alignment, temporal information is encoded into the separabilities.

Most existing DRS methods [4], [5], [6] depend on dynamic time warping (DTW) [3] to measure the separability. Due to the boundary condition and the strict order-preserving constraint, DTW cannot tackle local reorder distortions and may not fully capture the essential differences of different patterns. As shown in Fig. 1, the two action sequences "jump" and "run" start from different poses, resulting in reordered poses in the run-up phase, and "jump" vacates after a run-up. For DTW, some different running poses are wrongly aligned (shown in blue bold) and the vacated poses of "jump" are forced to align to a single pose of "run" (shown in green). Many pairwise differences among the vacant poses and the running poses in the same cycle are not included.

Differently, OPW casts the temporal alignment as a transport problem. It encourages transport between temporally adjacent observations, but allows local reorders or distortions. In Fig. 1, the reordered running poses are correctly aligned by OPW. For the boxed parts, the vacated poses of "jump" are dispersedly aligned to different poses in a periodic cycle of "run" (shown in red). OPW is able to determine the true distinctive observation pairs that reflect the essential

- *Bing Su and Ji-Rong Wen are with the Beijing Key Laboratory of Big Data Management and Analysis Methods, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China E-mail: subingats@gmail.com, jrwen@ruc.edu.cn.*
- *Jiahuan Zhou and Ying Wu are with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208 USA. E-mail: zhoujh09@gmail.com, yingwu@ece.northwestern.edu.*

Manuscript received 4 Mar. 2020; revised 5 Jan. 2021; accepted 7 Jan. 2021.
Date of publication 12 Jan. 2021; date of current version 5 May 2022.
(Corresponding author: J. Zhou.)
Recommended for acceptance by Z. Tu.
Digital Object Identifier no. 10.1109/TPAMI.2021.3050750

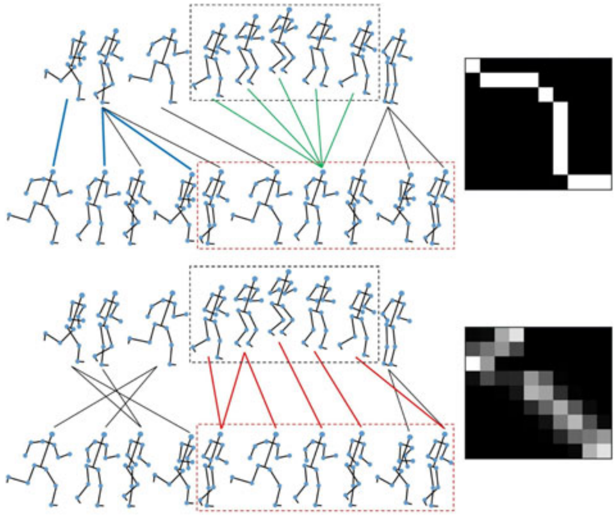


Fig. 1. The two action sequences "jump" and "run" differ in the boxed parts and local orders at the beginning stage. Top: the DTW [3] alignment. The alignment matrix is shown on the right. The white grid in row i and column j indicates that the i-th and j-th observations in the two sequences are aligned. Bottom: the OPW [1] alignment. For each pose, only the alignment with the largest transport probability is shown. Such aligned pairs reflect the essential difference between "jump" and "run" because the take-off-landing cycle is dispersedly aligned to a running cycle. The transport matrix is shown on the right. The grey value of a grid indicates the probability of aligning the corresponding observations. The probabilities among the boxed part are scattered and more pairwise local differences among poses are employed.

differences of two sequences or barycenters, so that the DRS method can focus on discriminating these distinctions. Since OPW is more robust to local distortions, our OPW-based intra-class scatter also better encodes the intra-class variations. In addition, different from the binary DTW alignment, the transport measures the probabilities of how different observation pairs contribute to the total difference. The probabilities among the boxed parts are scattered and more local relations among all observations are considered by the proposed DRS method.

The main contributions of this paper are three-fold. 1. We propose novel OPW-based separability measures among sequence classes to reflect their essential differences. Especially, we construct unified intra-class and inter-class scatters based on the learned optimal transports to encode the temporal relationships and employ more local pairwise differences. 2. We provide mathematical derivations to compute the barycenter for sequence data w.r.t. the OPW distance, based on which we further derive a discrete and explicit formulation of the covariance matrix for sequence data. The OPW barycenter and the derived covariance can be considered as the first and second order statistics for sequences, respectively. 3. We learn a discriminative subspace in which the sequences from different classes are maximally separated w.r.t. the OPW distance under the Fisher's criterion, which can be extended to other criteria.

This paper is an extension of the conference paper [7], where the new contributions include 1. the deep extension of the proposed OWDA, namely DeepOWDA, which learns nonlinear transformations using deep neural networks; 2. the evaluation of iterative variants of OWDA and DeepOWDA that jointly learn the subspace and the associated optimal transports in an alternative manner; 3. the evaluation of the variants of OWDA and DeepOWDA with fixed uniform weights for barycenters; 4. the experimental evaluation on the large scale NTU RGB-D dataset; 5. comparisons with state-of-the-art results on four 3D action recognition datasets; 6. comparisons with other sequence distances such as DTW, Soft-DTW and CTW; 7. the experimental evaluation on the effects of different types of frame-wide features; 8. more in-depth analyses and discussions of the proposed methods and the related works.

2 Related Work

Discriminant Analysis. Supervised linear dimensionality reduction for static data has been extensively studied in the literature. The well-known linear discriminant analysis (LDA) learns the projection by maximizing the ratio of inter-class distance to the intra-class distance. Various methods are proposed to improve or extend LDA in specific situations. The null space LDA [8], generalized ULDA [9] and orthogonal LDA [10] deal with the small sample size problem resulting in singular scatter matrices. To handle heteroscedastic data, heteroscedastic LDA [11] incorporates the second-order information into the between-class scatter, and subclass discriminant analysis [12] divides each class into several homoscedastic subclasses and then applies LDA to the subclasses. Max-min distance analysis approaches [13], [14], [15] maximize the minimum pairwise between-class distance in the subspace. Marginal Fisher analysis [16] only uses the neighboring samples and the samples distributed around the class boundaries to construct the intra-class and inter-class scatters. Wasserstein discriminant analysis [17] employs the regularized Wasserstein distance to measure the distance between the empirical probabilities of class populations. Kernal-LDA [18] and DeepLDA [19] extend LDA to learn non-linear transformations by kernel trick and employing deep neural networks, respectively. These advances cannot be applied to observations in sequences directly because the observations do not satisfy the basic i.i.d. assumption.

Dimensionality Reduction for Sequence Data. Far less attention has been paid to DRS. In [20], a kernel-based sufficient dimensionality reduction approach is proposed to improve the performance of sequence labeling, where each observation in sequences has a label. In this paper, we learn the projection to improve the performance of sequence classification that each entire sequence is associated with a single label. Canonical Time Warping (CTW) [21], generalized CTW (GCTW) [22], [23], and Deep CTW (DCTW) [24], [25] are unsupervised distances between sequences from different modals where vectors may have different dimensions. They use two separate transformations to map two sequences into a common subspace in which the sequences are maximally correlated. The transformation for the same sequence is different when aligned to different sequences. In [26], SoftDTW determines a soft alignment between two sequences that results in the soft-minimum of all feasible alignments, but each feasible alignment is strict order-preserving. In [27], [28], kernelized rank pooling (KRP) and generalized rank pooling (GRP) are pooling methods that encode different sequences into different subspace representations in an unsupervised manner. In contrast, the proposed OWDA and DeepOWDA are supervised DRS methods that learn a common subspace so that sequences from different classes are better separated in this subspace. The transformation remains the same for all sequences. CTW, GCTW, DCTW, SoftDTW, KPR, and GRP can be applied to the low-dimensional sequences transformed by OWDA and DeepOWDA.

In [29], a Mahalanobis distance for observations in sequences is learned to improve the performance of multivariate sequence alignment, where the ground-truth alignments between sequences are given. In this paper, we learn the projection without any alignment annotations. In [30], supervised word mover's distance learns a transformation to better separate different documents with the optimal transport distance. The order of words in the documents is ignored. The objective is minimizing the stochastic leave-one-out nearest neighbor classification error on a per-document level. The gradient-based iterative solution is developed to optimize it. In this paper, we learn a transformation to better separate sequences from different classes. The objective is maximizing the separability among sequence classes. We build novel separability measures to encode the temporal information and obtain a closed-form solution. In [31], the embedding vectors of tree nodes are learned by minimizing a surrogate of the classification error using the nearest prototype classifier w.r.t. the tree edit distance, where the prototypes are selected from the training trees. In this paper, we minimize the distances between training sequences to the corresponding barycenters w.r.t. the OPW distance.

In [4], [5], [6], linear sequence discriminant analysis (LSDA) and max-min inter-sequence distance analysis (MMSDA) are proposed for DRS, respectively. LSDA and MMSDA extract a representative sequence and a intra-class variance matrix for each class based on the statistics of a trained HMM. The DTW distance between the representative sequences is used as the inter-class distance. The similarities for measuring the inter-class distance and intra-class scatter are inconsistent, because the HMM-based intra-class variance does not measure the dispersion of the DTW distances among the sequences. In [32], [33], latent temporal LDA (LT-LDA) divides all observations of sequences from each class into several vector subclasses by dynamically aligning the sequences in this class to the DTW barycenter. All the subclasses are treated as independent to construct the inter-class separability. Therefore, the temporal information among the subclasses of the same sequence class is not fully explored.

Different from these methods, in this paper, we employ the OPW distance instead of the DTW distance as the similarity measure between sequences, and construct the intra-class scatter and the inter-class distance consistently w.r.t. the OPW distance. We extract the OPW barycenter as the representative sequence, which is non-parametric and has better scalability without the need of training HMMs with massive parameters. We use the OPW distance between the OPW barycenters as the inter-class separability between two sequence classes, which explicitly encodes the temporal information among the elements of the OPW barycenters. MMSDA optimizes the max-min distance criterion, which is more suited to tackle the class separation problem. Note that the proposed method can also be extended by applying the max-min distance criterion to the constructed inter-class and intra-class scatters. In this paper, we only compare with the DRS methods optimizing the same Fisher criterion.

Skeleton-Based 3D Action Recognition. Most 3D action recognition methods either learn a representation of the entire sequence or employ sequence models such as LSTM and HMM. For the first category, many methods obtain the entire representation from the sequence of frame-wide features. In [34], the pairwise relative positions or angles of joints are used as the feature of each frame. Each action sequence is encoded into a vector by Fourier temporal pyramid. In [35], [36], the joint positions are used as the feature of each frame and the covariance-based features are extracted from the sequence of frame-wide features. In [37], the histogram of relative joint positions is used as the frame-wide feature and all frame-wide features are encoded by rank pooling. In [38], translations and rotations of parts are extracted as features and each sequence cast as a curve in the Lie group is encoded by Fourier temporal pyramid. Such methods can imply a loss of temporal information.

For the second category, each action is represented as a sequence of frame-wide features and sequences are directly input to sequence models for classification. In [39], the histogram of relative joint positions is used as the feature of each frame and the sequences are modeled by HMM. In [40], [41], [42], [43], [44], joint positions, relative motions between successive frames, or the combined or normalized versions are used as frame-wide features and the sequences are modeled by recurrent neural networks such as temporal sliding LSTM (TD-LSTM), Bi-LSTM, spatio-temporal LSTM with trust gates, global context-aware attention LSTM, and independently recurrent neural network (IndRNN). Most recent works such as [45], [46], [47], [48], [49] model the body skeletons as spatio-temporal graphs by viewing joints as nodes and bones as edges and employ graph neural networks for classification.

In this paper, our purpose is not to develop a state-of-the-art 3D action recognition method. We apply the proposed DRS method, OWDA, to 3D action sequences to evaluate its performance. The proposed OWDA benefits both categories of 3D action recognition methods. After projection by OWDA, the sequences are more discriminative and the temporal information is enhanced. As a result, more useful information is encoded into the entire representation and sequence models need to learn few parameters. However, OWDA cannot be combined directly with graph-based methods because the reduced features cannot be modeled into a graph according to the structure of the human body after dimensionality reduction is performed to the frame-wide features of concatenated joint positions.

3 Linear Order-Preserving Wasserstein Discriminant Analysis

The proposed linear OWDA consists of three stages: extracting the OPW barycenter per class, constructing the separability scatters based on the barycenters, and learning the projection by maximizing the separability. In this section, we first briefly review the OPW distance, and then present the details of the three stages, respectively.

3.1 Background on OPW

We first briefly review the *order-preserving Wasserstein (OPW) distance* [1], [2]. For two sequences $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_x}]$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N_y}]$ with lengths N_x and N_y, respectively, where the dimension of features is q, i.e., $\boldsymbol{x}_i, \boldsymbol{y}_j \in \mathbb{R}^q$, the OPW distance is defined as:

$$\begin{aligned} &d_{OPW}(\boldsymbol{X},\boldsymbol{Y}) := \langle \boldsymbol{T}^*, \boldsymbol{D}\rangle \\ &s.t.\ \boldsymbol{T}^* = \underset{\boldsymbol{T}\in\Phi(\boldsymbol{\gamma},\boldsymbol{\beta})}{arg\min} \langle \boldsymbol{T},\boldsymbol{D}\rangle - \lambda_1 I(\boldsymbol{T}) + \lambda_2 KL(\boldsymbol{T}||\boldsymbol{P}), \end{aligned} \tag{1}$$

where $\boldsymbol{D} := [d(\boldsymbol{x}_i,\boldsymbol{y}_j)]_{ij} \in \mathbb{R}^{N_x\times N_y}$ is the matrix of all the pairwise distances between supporting points, $d(\cdot,\cdot)$ is set to the squared euclidean distance in this paper. $\boldsymbol{T} := [t_{ij}]_{ij} \in \mathbb{R}^{N_x\times N_y}$ is the transport matrix, $\langle\cdot,\cdot\rangle$ is the Frobenius dot product, and $\Phi(\boldsymbol{\gamma},\boldsymbol{\beta}) := \{\boldsymbol{T}\in\mathbb{R}_+^{N_x\times N_y} | \boldsymbol{T}\mathbf{1}_{N_y} = \boldsymbol{\gamma}, \boldsymbol{T}^T\mathbf{1}_{N_x} = \boldsymbol{\beta}\}$ is the feasible set of the transport $\boldsymbol{T}$. $I(\boldsymbol{T}) = \sum_{i,j}\frac{t_{ij}}{(\frac{i}{N_x}-\frac{j}{N_y})^2+1}$ is the inverse difference moment of the transport matrix $\boldsymbol{T}$ to encourage the local homogeneity that large values appear near the diagonal, and $KL(\boldsymbol{T}||\boldsymbol{P})$ is the Kullback-Leibler divergence between $\boldsymbol{T}$ and a prior distribution $\boldsymbol{P}$:

$$p_{ij} := \boldsymbol{P}(i,j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\ell^2(i,j)}{2\sigma^2}}, \tag{2}$$

where $\ell(i,j)$ is the vertical distance from the position (i,j) to the diagonal line. $\lambda_1 > 0$, $\lambda_2 > 0$, and σ are hyperparameters. It is assumed that the weights of instances in the same sequence are the same, *i.e.*, $\boldsymbol{\gamma} = (\frac{1}{N_x},\ldots,\frac{1}{N_x})$ and $\boldsymbol{\beta} = (\frac{1}{N_y},\ldots,\frac{1}{N_y})$, respectively. In [1], OPW is solved by the Sinkhorn's algorithm with a complexity of N_xN_yq.

3.2 Order-Preserving Wasserstein Barycenter

For a sequence class with a set of training sequences, we want to extract a single representative sequence that reveals the average temporal structures and general evolution trends, which can serve as the mean sequence of a set of sequences similar to the mean vector of a set of vectors. Extending the averaging operation to sequences is challenging. As the lengths of different sequences are different, it is not plausible to perform directly averaging to the observations at the same time step.

Recall that the mean of a set of vectors can also be viewed as the barycenter of the vectors with regard to the euclidean distance. Similarly, for sequence data, the barycenter of a set of sequences with regard to a sort of sequence distance can also act as the mean sequence in some sense. We extract the barycenter with regard to the OPW distance, which we call the *order-preserving Wasserstein barycenter*.

The barycenter $\boldsymbol{U} = (\boldsymbol{\mu},\boldsymbol{\gamma})$ consists of a sequence of ordered supporting points and a weight sequence associating each supporting point with a probability value. $\boldsymbol{\mu} = [\boldsymbol{\mu}_i, i = 1,\ldots,L] \in \mathbb{R}^{q\times L}$ is the sequence of supporting points and $\boldsymbol{\gamma} = [\gamma_i, i = 1,\ldots,L]$ is the sequence of associated weights. $\boldsymbol{\gamma}\in\mathbb{R}^{1\times L}$ lies in the simplex Θ_L. L is a pre-set value, which indicates the maximum allowed number of supporting points of the barycenter.

Given a set of sequences $\boldsymbol{X}_k, k = 1,\ldots,N$, N_k denotes the length of $\boldsymbol{X}_k$, $\boldsymbol{D}_k$ denotes the matrix of all pairwise ground distances between any $\boldsymbol{\mu}_i$ and observations in $\boldsymbol{X}_k$, which depends on $\boldsymbol{\mu}$:

$$\boldsymbol{D}_k(\boldsymbol{\mu}) := [d(\boldsymbol{\mu}_i, \boldsymbol{x}_j^k)]_{ij} \in \mathbb{R}^{L\times N_k}. \tag{3}$$

$\boldsymbol{T}_k$ denotes the transport between $\boldsymbol{U}$ and $\boldsymbol{X}_k$. The optimal transport determined by OPW is given by $\arg\min_{\boldsymbol{T}_k\in\Phi(\boldsymbol{\gamma},\boldsymbol{\beta}_k)} W(\boldsymbol{U},\boldsymbol{X}_k,\boldsymbol{T}_k)$, where

$$W(\boldsymbol{U},\boldsymbol{X}_k,\boldsymbol{T}_k) = \langle\boldsymbol{T}_k, \boldsymbol{D}_k(\boldsymbol{\mu})\rangle - \lambda_1 I(\boldsymbol{T}_k) + \lambda_2 KL(\boldsymbol{T}_k||\boldsymbol{P}). \tag{4}$$

By assuming that these sequences are equally weighted, the order-preserving Wasserstein barycenter is such that

$$\boldsymbol{U} = \arg\min_{\boldsymbol{U}} \sum_{k=1}^{N} \min_{\boldsymbol{T}_k\in\Phi(\boldsymbol{\gamma},\boldsymbol{\beta}_k)} \frac{1}{N} W(\boldsymbol{U},\boldsymbol{X}_k,\boldsymbol{T}_k). \tag{5}$$

Both the supporting point sequence $\boldsymbol{\mu}$ and the weight sequence $\boldsymbol{\gamma}$ need to be learned. However, the objective function (5) is not convex w.r.t. them simultaneously. We employ the alternating updating strategy to minimize (5), where $\boldsymbol{\gamma}, \boldsymbol{T}_k$ and $\boldsymbol{\mu}$ are updated alternatively by temporarily fixing the other. To initialize $\boldsymbol{\mu}$, we divide $\boldsymbol{X}_k, k = 1,\ldots,N$ uniformly into L segments, respectively, and use the mean of vectors in the ith segments in all $\boldsymbol{X}_k$ as the initial $\boldsymbol{\mu}_i$.

In procedure 1, we first update the weight sequence $\boldsymbol{\gamma}$ and the optimal transports $\boldsymbol{T}_k, k = 1,\ldots,N$ by fixing $\boldsymbol{\mu}$. Eq. (4) can be reformulated as follows.

$$\langle\boldsymbol{T}_k, \boldsymbol{D}_k(\boldsymbol{\mu})\rangle - \lambda_1 I(\boldsymbol{T}_k) + \lambda_2 KL(\boldsymbol{T}_k||\boldsymbol{P}) = \lambda_2 KL(\boldsymbol{T}_k||\boldsymbol{K}_k), \tag{6}$$

where $d_{ij}^k = d(\boldsymbol{\mu}_i, \boldsymbol{x}_j^k)$, $s_{ij}^{\lambda_1} = \frac{\lambda_1}{(\frac{i}{N}-\frac{j}{M})^2+1}$, and $\boldsymbol{K}_k = [p_{ij}e^{\frac{1}{\lambda_2}(s_{ij}^{\lambda_1} - d_{ij}^k)}]_{ij}$.

$\boldsymbol{D}_k(\boldsymbol{\mu}), k = 1,\ldots,N$ are fixed when $\boldsymbol{\mu}$ is fixed, hence $\boldsymbol{K}_k$ are also fixed. Problem (5) is thereby reformulated as

$$\begin{aligned} &\min_{\boldsymbol{\gamma},\boldsymbol{T}_k,k=1,\ldots,N} \sum_{k=1}^{N}\frac{1}{N} KL(\boldsymbol{T}_k||\boldsymbol{K}_k) \\ &s.t.\ \exists\boldsymbol{\gamma}\in\Theta_L, \boldsymbol{T}_k\mathbf{1}_{N_k} = \boldsymbol{\gamma}, \forall k = 1,\ldots,N \\ &\boldsymbol{T}_k{}^T\mathbf{1}_L = [\tfrac{1}{N_k},\ldots,\tfrac{1}{N_k}]^T, k = 1,\ldots,N. \end{aligned} \tag{7}$$

where $\Theta_L := \{\boldsymbol{\gamma}\in\mathbb{R}^L | \gamma_i \geq 0, \forall i = 1,\ldots,L, \sum_{i=1}^L \gamma_i = 1\}$.

By defining $\boldsymbol{T} = (\boldsymbol{T}_k)_{k=1}^N \in (\mathbb{R}_+^{L\times N_k})^N$ and $\boldsymbol{K} = (\boldsymbol{K}_k)_{k=1}^N \in (\mathbb{R}_+^{L\times N_k})^N$, Problem (7) is rewritten as

$$\begin{aligned} &\min_{\boldsymbol{\gamma},\boldsymbol{T}} KL_N(\boldsymbol{T}||\boldsymbol{K}), \boldsymbol{\gamma}\in\Theta_L \\ &s.t.\ \boldsymbol{T}\in\boldsymbol{\Phi}_1\cap\boldsymbol{\Phi}_2 \end{aligned}, \tag{8}$$

where $KL_N(\boldsymbol{T}||\boldsymbol{K}) := \sum_{k=1}^N \frac{1}{N} KL(\boldsymbol{T}_k||\boldsymbol{K}_k)$,

$$\begin{aligned} \boldsymbol{\Phi}_1 &:= \Big\{\boldsymbol{T}\in(\mathbb{R}_+^{L\times N_k})^N : \boldsymbol{T}_k{}^T\mathbf{1}_L = [\tfrac{1}{N_k},\ldots,\tfrac{1}{N_k}]^T, \forall k\Big\}, \\ \boldsymbol{\Phi}_2 &:= \Big\{\boldsymbol{T}\in(\mathbb{R}_+^{L\times N_k})^N : \exists\boldsymbol{\gamma}\in\Theta_L, \boldsymbol{T}_k\mathbf{1}_{N_k} = \boldsymbol{\gamma}, \forall k\Big\}. \end{aligned}$$

In [50], it is shown that the iterative Bregman projection (IBP) [51], [52] can solve Problem (8) efficiently. Specifically, as proved in [1], each $\boldsymbol{T}_k$ is a rescaled version of $\boldsymbol{K}_k$ with the form of $diag(\boldsymbol{\kappa}_{k1})\boldsymbol{K}_k diag(\boldsymbol{\kappa}_{k2})$, and the scaling vectors can be updated using the Sinkhorn's iterations:

$$\boldsymbol{\kappa}_{k1}^{(n)} \leftarrow \boldsymbol{\gamma}^{(n)}./\boldsymbol{K}_k\boldsymbol{\kappa}_{k2}^{(n)}, \tag{9}$$

$$\boldsymbol{\kappa}_{k2}^{(n+1)} \leftarrow \left[\tfrac{1}{N_k}, \ldots, \tfrac{1}{N_k}\right]^T ./(\boldsymbol{K}_k)^T \boldsymbol{\kappa}_{k1}^{(n)}. \tag{10}$$

As given in [50], $\boldsymbol{\gamma}^{(n)}$ is the update of the weights:

$$\boldsymbol{\gamma}^{(n)} \leftarrow \prod_{k=1}^{N} \Big(\boldsymbol{\kappa}_{k1}^{(n)} \odot ((\boldsymbol{K}_k)^T \boldsymbol{\kappa}_{k2}^{(n)})\Big)^{\frac{1}{N}}. \tag{11}$$

where $\odot$ is the element-wise product. The iterations continue until convergence. Given the learned weights and the fixed supporting points, we perform OPW to obtain the updates of the optimal transports $\boldsymbol{T}_k$, for $k = 1, \ldots, N$.

In procedure 2, we update the supporting point sequence $\boldsymbol{\mu}$ by fixing the weight sequence $\boldsymbol{\gamma}$ and optimal transports $\boldsymbol{T}_k^*, k = 1, \ldots, N$ updated in procedure 1. In Eq. (4), only the first term evolves $\boldsymbol{\mu}$. By viewing the sequences $\boldsymbol{\mu}$ and $\boldsymbol{X}_k$ as matrices, we have

$$\begin{aligned}\big\langle \boldsymbol{T}_k^*, \boldsymbol{D}_k(\boldsymbol{\mu})\big\rangle = {}& diag(\boldsymbol{\mu}^T\boldsymbol{\mu})^T\boldsymbol{\gamma} - 2\big\langle \boldsymbol{T}_k^*, \boldsymbol{\mu}^T\boldsymbol{X}_k\big\rangle \\ & + diag(\boldsymbol{X}_k^T\boldsymbol{X}_k)^T[\tfrac{1}{N_k}, \ldots, \tfrac{1}{N_k}]^T.\end{aligned}$$

We follow [53] to optimize the local quadratic approximation of the following function: $diag(\boldsymbol{\mu}^T\boldsymbol{\mu})^T\boldsymbol{\gamma} - 2\big\langle \boldsymbol{T}_k^*, \boldsymbol{\mu}^T\boldsymbol{X}_k\big\rangle = \|\boldsymbol{\mu} diag(\boldsymbol{\gamma}^{\frac{1}{2}}) - \boldsymbol{X}_k{\boldsymbol{T}_k^*}^T diag(\boldsymbol{\gamma}^{-\frac{1}{2}})\|^2 - \|\boldsymbol{X}_k{\boldsymbol{T}_k^*}^T diag(\boldsymbol{\gamma}^{-\frac{1}{2}})\|^2$. Given a single sequence $\boldsymbol{X}_k$, the Newton update is $\boldsymbol{\mu} \leftarrow \boldsymbol{X}_k{\boldsymbol{T}_k^*}^T$ $diag(\boldsymbol{\gamma}^{-1})$.

For all N training sequences, $\boldsymbol{\mu}$ is finally updated by

$$\boldsymbol{\mu} \leftarrow (1-\xi)\boldsymbol{\mu} + \xi\left(\sum_{k=1}^{N} \boldsymbol{X}_k{\boldsymbol{T}_k^*}^T\right) diag(\boldsymbol{\gamma}^{-1}), \tag{12}$$

where $\xi \in [0, 1]$ is a pre-set value.

We cycle the two alternative procedures until the change in the objective function value Eq. (5) is less than a threshold or a maximum number of steps is reached. It was shown in [50], [52], [54] that the iterative Bregman projection for updating $\boldsymbol{\gamma}$ converges linearly. The convergence rate of the Newton's method for updating $\boldsymbol{\mu}$ is quadratic. It can be difficult to obtain the global convergence rate of the overall alternating optimization. In our experiments, it converges in about 10 iterations. The complexity per iteration is $O(NTLq)$, where T is the average length of sequences.

3.3 Covariance

For a set of sequences, the barycenter serves as the "mean" sequence and reflects the average evolution. The dispersion of the sequences around the barycenter can be straightforwardly measured by accumulating the OPW distances:

$$d_w = \sum_{k=1}^{N} d_{OPW}(\boldsymbol{U}, \boldsymbol{X}_k) = \sum_{k=1}^{N} \big\langle \boldsymbol{T}_k^*, \boldsymbol{D}_k\big\rangle, \tag{13}$$

where the optimal transports $\boldsymbol{T}_k^*$ between $\boldsymbol{U}$ and $\boldsymbol{X}_k$, for $k = 1, \ldots, N$, are the by-products when determining the barycenter, so no extra calculations are needed.

To measure the covariance over different dimensions, we define a covariance matrix $\boldsymbol{\Gamma}$ so that $tr(\boldsymbol{\Gamma}) = d_w$. $\boldsymbol{\Gamma}$ can be constructed by accumulating the weighted outer products between any $\boldsymbol{\mu}_i$ and observations in $\boldsymbol{X}_k$ as follows:

$$\boldsymbol{\Gamma} = \sum_{k=1}^{N}\sum_{i=1}^{L}\sum_{j=1}^{N_k} {t_{ij}^k}^* (\boldsymbol{\mu}_i - \boldsymbol{x}_j^k)(\boldsymbol{\mu}_i - \boldsymbol{x}_j^k)^T. \tag{14}$$

We can find that $\boldsymbol{\Gamma}$ captures all local relations between elements of the barycenter and the observations in all sequences. All element-observation pairs contribute to the total covariance with different weights. The weight of a pair $(\boldsymbol{\mu}_i, \boldsymbol{x}_j^k)$ is actually the corresponding element ${t_{ij}^k}^*$ of the learned transport $\boldsymbol{T}_k^*$, so it reflects the probability of matching the pair. In this way, the local pairwise relations or joint probabilities are encoded. The weights are larger for the pairs that have high joint probabilities, since the matched pairs probably correspond to the same temporal structure. The differences between pairs with low joint probabilities are also incorporated, but with smaller weights, to consider soft alignments and compensate possible missing matches. As a result, the constructed $\boldsymbol{\Gamma}$ better reflects the spatial-temporal variances in different dimensions.

In [55], the optimal transport based variance for continuous one-dimensional densities is defined as $E(d_W^2(\boldsymbol{U}, \boldsymbol{X}_k))$, which is a scalar, where d_W is the Wasserstein metric. In this section, we derive an explicit and discrete formulation of the intra-class scatter $\boldsymbol{\Gamma}$ in Eq. (14) for multi-dimensional sequences. It satisfies $tr(\boldsymbol{\Gamma}) = \sum_k d_{OPW}(\boldsymbol{U}, \boldsymbol{X}_k)$, where OPW can be viewed as the squared 2nd order Wasserstein distance with temporal constraints. Therefore, the constructed $\boldsymbol{\Gamma}$ in Eq. (14) is consistent with the definition in [55].

3.4 Learning the Projection

Our goal is to learn a transformation that projects the observations in sequences onto a low-dimensional subspace, in which the sequences from different classes get better separated. We employ the Fisher criterion to maximize the separability, *i.e.*, we maximize the ratio of the inter-sequence-class distance to the intra-sequence-class dispersion.

For each sequence class $\omega_c, c = 1, \ldots, C$, we extract the order-preserving Wasserstein barycenter $\boldsymbol{U}^c$ and the covariance matrix $\boldsymbol{\Gamma}^c$ from the training sequences of the class. C is the total number of classes. We define the intra-sequence-class scatter as the weighted sum of covariances:

$$\boldsymbol{\Gamma}_w = \sum_{c=1}^{C} p^c \boldsymbol{\Gamma}^c, \tag{15}$$

where p^c is the prior probability of class ω_c and can be estimated as the ratio of the number of sequences of ω_c to the total number of sequences of all classes.

We measure the distance between two classes ω_c and $\omega_{c'}$ by the OPW distance between the corresponding order-preserving Wasserstein barycenters.

$$d_b(\omega_c, \omega_{c'}) = d_{OPW}(\boldsymbol{U}^c, \boldsymbol{U}^{c'}) = \big\langle \boldsymbol{T}_{cc'}^*, \boldsymbol{D}_{cc'}\big\rangle, \tag{16}$$

where $\boldsymbol{D}_{cc'}$ is the matrix of all pairwise distance between $\boldsymbol{\mu}_i^c$ and $\boldsymbol{\mu}_j^{c'}$, and $\boldsymbol{T}_{cc'}^*$ is the optimal OPW transport between the two barycenters. The corresponding between-class scatter $\boldsymbol{\Gamma}_{b(cc')}$ is the weighted sum of outer products between elements of the two barycenters, so that $d_b(\omega_c, \omega_{c'}) = tr(\boldsymbol{\Gamma}_{b(cc')})$:

$$\mathbf{\Gamma}_{b(cc')} = \sum_{i=1}^{L}\sum_{j=1}^{L} t_{ij}^{cc'^*}(\boldsymbol{\mu}_i^c - \boldsymbol{\mu}_j^{c'})(\boldsymbol{\mu}_i^c - \boldsymbol{\mu}_j^{c'})^T. \tag{17}$$

We define the overall inter-sequence-class scatter as the weighted sum of all pairwise between-class scatters:

$$\mathbf{\Gamma}_b = \sum_{c=1}^{C-1}\sum_{c'=c+1}^{C} p^c p^{c'} \mathbf{\Gamma}_{b(cc')}. \tag{18}$$

We observe again that all the differences between elements in all barycenters contribute to the overall inter-class scatter according to different weights. The weight $t_{ij}^{cc'^*}$ of a pair $(\boldsymbol{\mu}_i^c, \boldsymbol{\mu}_j^{c'})$ encodes the local relations of the two elements and indicates their joint probability. $\mathbf{\Gamma}_b$ concentrates more on the differences between the pairs with large joint probabilities. Such differences reflect the essential distinctions of two classes, because the matched pairs represent the homologous temporal structures and thus are distinctive for discriminating the two classes. Different from the alignments by DTW, where the weights are 1 for a small portion of aligned pairs and 0 for other pairs, the weights by OPW are soft probabilities and hence $\mathbf{\Gamma}_b$ also incorporates the differences between the pairs with smaller weights. This compromises more information and is more robust to incorrect or ambiguous alignments caused by noises.

When both the features in sequences and their dimensions are not linearly related, the ranks of $\mathbf{\Gamma}_w$ and $\mathbf{\Gamma}_b$ are $min(N^t, q)$ and $min(CL, q)$, respectively, where N^t is the number of all features in all training sequences. When $N^t \geq q$ $(CL \geq q)$, $\mathbf{\Gamma}_w$ $(\mathbf{\Gamma}_b)$ is full-rank. In extreme cases when there are too few training sequences so that $N^t < q$, we can use PCA to remove the null space of $\mathbf{\Gamma}_w$ or add a identity matrix multiplied by a small scalar to $\mathbf{\Gamma}_w$.

The objective of learning the projection $\boldsymbol{W}$ using the Fisher criterion is formulated as the ratio-trace problem:

$$\max_{\boldsymbol{W}} tr((\boldsymbol{W}^T\mathbf{\Gamma}_w\boldsymbol{W})^{-1}\boldsymbol{W}^T\mathbf{\Gamma}_b\boldsymbol{W}). \tag{19}$$

The optimal $\mathbf{W}^*$ of Problem (19) is the matrix whose columns are the eigenvectors of $\Gamma_w^{-1}\Gamma_b$ w.r.t. the q' largest eigenvalues, where q' is the reduced dimensionality. The proposed DRS method is called *Order-preserving Wasserstein Discriminant Analysis (OWDA).*

3.5 Discussion

Complexity. Let N_a and T denote the average number of sequences per class and the average length of sequences, respectively. The complexities for calculating the barycenters for all C classes, calculating the inter-class and intra-class scatters, and solving (19) are $O(CN_aTLq)$, $O(C^2L^2q^2)$, $O(CN_aLTq^2)$, and $O(q^3)$, respectively. The overall complexity of linear OWDA is $O(C^2L^2q^2 + CN_aLTq^2 + q^3)$. It scales linearly with the number of samples, but cubically with the dimension of features q due to the eigen-decomposition (19). We simultaneously diagonalize the intra-class and inter-class scatters [10] to solve (19). Any advanced methods for large-scale eigen-decomposition can be applied to accelerate our method.

Subclass Extension. Our model can be extended to fit multiple barycenters for each class. By implementing off-the-shelf clustering methods on the training sequences for each class given a sequence distance such as OPW, each class can be clustered into several subclasses. Therefore, our method can extract a barycenter for each subclass. However, whether we need to use one or multiple barycenters per class depends heavily on the data. If the data exhibit unimodal distributions, using only one barycenter per class is enough. On the other hand, if we use multiple barycenters, although we may gain performance improvement, the computation cost increases.

4 Deep Order-Preserving Wasserstein Discriminant Analysis

The temporal evolution and distortion of sequences may be highly non-linear, and sometimes a linear transformation may not be able to fully distinguish the temporal structures among sequences of different classes. In this section, we extend the proposed OWDA to learn non-linear transformations using a deep neural network. We refer to this deep extension as DeepOWDA.

Specifically, instead of using a linear projection matrix $\boldsymbol{W}$, we employ a deep neural network to perform nonlinear transformations on the frame-wide features of sequences. The neural network $f(\cdot, \theta)$ is parameterized by θ and the output for an input feature vector $\boldsymbol{x}$ is denoted by $f(\boldsymbol{x}, \theta)$. As a result, the output of a sequence $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_x}]$ is transformed into $f(\boldsymbol{X}, \theta) = [f(\boldsymbol{x}_1, \theta), \ldots, f(\boldsymbol{x}_{N_x}, \theta)]$.

DeepOWDA trains the network by maximizing the ratio of the OPW-based inter-sequence-class scatter $\mathbf{\Gamma}_b$ and the intra-sequence-class scatter $\mathbf{\Gamma}_w$ in the subspace so that the transformed sequences from different classes get better separated w.r.t. the OPW distance. However, to construct the inter-class and intra-class scatters in the latent subspace, the barycenters of the transformed sequences for all classes and the OPW distances among barycenters need first to be calculated, which require solving minimization problems over transports and depends on the network to be learned.

Taking a closer look at Eqs.(5), (14), and (17), we can find that for given training sequences, the barycenter sequence $\boldsymbol{\mu}$ and covariance $\mathbf{\Gamma}$ are functions of the optimal transports $T_k^{c*}, k = 1, \ldots, N^c$ between the training sequences and the corresponding barycenter for each class $c = 1, \ldots, C$, and the between-class scatter $\mathbf{\Gamma}_b$ is a function of the optimal transports $\boldsymbol{T}_{cc'}^*, c, c' = 1, \ldots, C$ between barycenters of different classes. To make the objective tackle, we first calculate the barycenters as well as the related intra-class optimal transports $\boldsymbol{T}_k^{c*}, k = 1, \ldots, N^c, c = 1, \ldots, C$ and inter-class optimal transports $\boldsymbol{T}_{cc'}^*, c, c' = 1, \ldots, C$ from the original sequences. We fix these inter-class and intra-class optimal transports to construct the barycenters and scatters in the latent subspace. For the c-th sequence class, the barycenter sequence $\boldsymbol{\mu}^c = [\boldsymbol{\mu}_i^c, i = 1, \ldots, L]$ is constructed as

$$\boldsymbol{\mu}_i^c = \sum_{k=1}^{N^c}\sum_{j=1}^{N_k} \boldsymbol{T}_k^{c*}(i, j) f(\boldsymbol{x}_j^{ck}, \theta), i = 1, \ldots, L. \tag{20}$$

where $\boldsymbol{x}_j^{ck}$ is the jth observation of the k-th sequence sample of the c-th class.

The covariance is calculated as:

$$\mathbf{\Gamma}^c = \sum_{k=1}^{N^c}\sum_{i=1}^{L}\sum_{j=1}^{N_k} {t_{ij}^{k}}^{*}(\boldsymbol{\mu}_i^c - f(\boldsymbol{x}_j^{ck},\theta))(\boldsymbol{\mu}_i^c - f(\boldsymbol{x}_j^{ck},\theta))^T. \tag{21}$$

The between-class scatter is calculated as:

$$\mathbf{\Gamma}_{b(cc')} = \sum_{i=1}^{L}\sum_{j=1}^{L} {t_{ij}^{cc'}}^{*}(\boldsymbol{\mu}_i^c - \boldsymbol{\mu}_j^{c'})(\boldsymbol{\mu}_i^c - \boldsymbol{\mu}_j^{c'})^T. \tag{22}$$

The overall intra-class scatter and inter-class scatter are computed by Eqs. (15) and (18). Here, we use the intra-class and inter-class optimal transports in the original space to approximate those in the subspace. The optimal transports in the original space reflect the essential correspondences between original sequences. To minimize the intra-class scatter, the correspondences between feature vectors with large transport probabilities from sequences of the same class are further enhanced. Therefore, the optimal transports in the subspace may not change too much from those in the original space. In addition, since the transport matrices build the optimal correspondences between features in the original sequences under the OPW distance, discriminating sequences according to such correspondences is also likely to lead to better separability between different sequence classes, even if the optimal transports change in the subspace.

The original Fisher criterion is known to cause the so-called class separation problem, i.e., it overemphasizes distant class pairs with large inter-class distances. This achieves higher rewards because the Fisher criterion maximizes the sum of pairwise inter-class distance. As a result, in the learned subspace, the distances between classes that are already separated become larger, but the neighboring classes are more difficult to distinguish. This problem is exacerbated when using deep neural networks with strong fitting capabilities.

To alleviate this problem, following [19], we also apply a reformulated objective to DeepOWDA. Let $\alpha_i, i = 1, \ldots, C-1$ denote the eigenvalues of $\mathbf{\Gamma}_w^{-1}\mathbf{\Gamma}_b$ in descending order, and $\boldsymbol{v}_i, i = \ldots, C-1$ denotes the corresponding eigenvectors. Each α_i can be viewed as measuring the discriminative capacity of the direction of $\mathbf{v}_i$. Different from maximizing the sum of all eigenvalues as in the original Fisher criterion, which may overemphasis the largest few α_i, we only maximize the sum of the top k eigenvalues that are smaller than a pre-set threshold ϵ. The loss function of DeepOWDA is formulated as follows.

$$\begin{aligned} &\max{}_\theta \frac{1}{k}\sum_{i=1}^{k}\alpha_{j+i} \\ &s.t. \alpha_{j+1} < \epsilon, \alpha_j \geq \epsilon, \\ &\mathbf{\Gamma}_w^{-1}\mathbf{\Gamma}_b\mathbf{v}_i = \alpha_i\mathbf{v}_i, i = 1, \ldots, C-1. \end{aligned} \tag{23}$$

This formulation forces the network to discriminate confusing sequence classes and gain more discriminative power. Each eigenvalue can take the derivative w.r.t. the parameters and the loss function Eq.(23) is differentiable. The deep network is trained by back-propagation. During training, in each mini-batch, sufficient features from all classes are needed to estimate the scatters. Thus the batch size should be sufficiently large.

Given training sequences, the barycenter of each class is calculated in the original space to learn the projection. In the learned subspace, the barycenters and the corresponding optimal transports between the training sequences to them may change. Therefore, determining the barycenters and learning the projection are interlaced, as solving one depends on the other. Our solution implicitly assumes that salient temporal correspondences are often preserved after transformation; thus, the optimal transports in the original space can be used to approximate those in the reduced low-dimensional subspace (the barycenters and scatters are actually based on the optimal transports). After projection, the sequences of the same class are more concentrated to the barycenter and the barycenters of different classes are further away w.r.t. the OPW distance. Therefore, sequences from different classes are better separated.

In some cases, such optimal transports may be quite different in the original space and the subspace, additional confusions may be introduced due to the change of optimal transports in the subspace learned with the optimal transports in the original space. To address this problem, we can employ an alternating optimization scheme. Specifically, we learn an initial network using the optimal transports in the original space, use the network to transform the training sequences, and then re-infer the barycenters and associated optimal transports from the transformed sequences. The updated optimal transports in the subspace are used in turn to re-train the network. The two procedures repeat iteratively until the subspace cannot be improved anymore. We denote such an iterative solution by DeepOWDA-ite. The iterative process can also be applied to linear OWDA, which we denote by OWDA-ite. In this case, OWDA and DeepOWDA can be viewed as the 1-iterations of OWDA-ite and DeepOWDA-ite. However, the iterative process not only increases the computational complexity greatly but also does not necessarily guarantee convergence in theory.

Complexity. Let N_b denote the number of training sequences per class in each batch. The complexity of DeepOWDA per iteration is $O(C^2L^2q^2 + CN_bLTq^2 + q^3)$. We fix the number of iterations to 500 in our experiments.

5 Experiments

In this section, we evaluate the performances of the proposed linear and deep OWDA methods on four 3D-action datasets.

5.1 Datasets

The *MSR Sports Action3D dataset [34], [56]* contains 557 depth sequences captured by Kinect camera from 20 sports actions. Ten persons performed each action for two or three times. The skeleton joint positions of humans are also available in this dataset. In [34], [57], the authors split the dataset into a training set and a test set, where the training set includes the sequences performed by about half of the persons and the test set includes the rest. We follow this experimental setup and report our results on the test set. The *MSR Daily Activity3D dataset [34]* contains 320 daily activity sequences from 16 activity classes. The sequences were captured by a Kinect device. Ten subjects performed each activity in two poses. We follow the split of the dataset as in [34], [57] again and report our results on the test set.

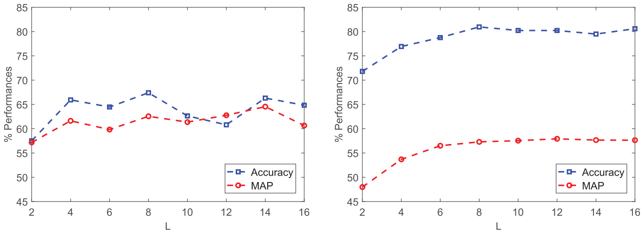
The *ChaLearn Gesture Recognition dataset* [58], [59] contains 955 Italian gesture sequences captured by Kinect camera from 20 different Italian gestures. Because we focus on individual sequence classification rather than sequence detection or segmentation, we follow [37], [60], [61] to perform experiments on the segmented sequences given by the ground-truth segments. Each segmented sequence contains only one gesture instance. 27 persons performed these gestures. Other annotations of this dataset include the foreground segmentation and joint skeletons. This dataset includes training set, validation set, and test set. Following [37], [60], [61], we learn the projections and train the classifiers on the training set, and report the results on the validation set.

The *NTU RGB+D dataset* [62] contains 56,880 action samples from 60 action classes. The action videos are performed by different subjects and recorded from different views. For each sample, the 3D coordinates of 25 major body joints per subject at all frames are available. The dataset provides two standard evaluation protocols. In the Cross-Subject (CS) evaluation, the videos of different subjects are split into training and testing groups. The training and testing sets contain 40,320 and 16,560 action sequences, respectively. In the Cross-View (CV) evaluation, the videos from different viewpoints are split into training and testing groups. The training and testing sets contain 37,920 and 18,960 action sequences, respectively.

5.2 Experimental Setup

Implementation Details. We perform zero-centralization on original frame-wide features for OWDA and divide the frame-wide features by 10 for DeepOWDA unless otherwise specified. For DeepLDA and DeepOWDA, the neural network for transformation is a three-layer perceptron, each fully connected layer is followed by a ReLU nonlinear function, and L_2 regularization is applied to the outputs. The number of nodes in all hidden layers is fixed at 1024.

Classification. We extract a feature vector from each frame as the observation of the frame. In this way, we represent each video by a sequence of observations. For evaluation, we employ the proposed linear and deep OWDA methods to project the observations in sequences onto subspaces with different dimensions. In the learned subspaces, we employ two sequence classifiers to classify the transformed sequences: the SVM classifier and the nearest neighbor (NN) classifier. For the SVM classifier, we first encode each sequence of observations into a fixed-dimensional vector by the unsupervised rank pooling [37]. Rank pooling learns two linear functions to rank the forward and reverse timing orders of the observations by the support vector regression, respectively. The parameters of the two linear functions are concatenated to form the pooling vector. Then, we train linear SVMs by taking these resulting vectors as input. We determine the hyper-parameter C of the linear SVMs by cross-validation. At the testing phase, we encode the test sequence of observations into a vector by rank pooling, and then employ the leaned SVMs to classify the encoded vector.



(a) (b)

Fig. 2. Performances as functions of L by (a) the SVM classifier and (b) the NN classifier on the MSR Action3D dataset.

For the NN classifier, we employ the OPW distance as the dissimilarity measure between two sequences. Specifically, for a test sequence, we calculate its OPW distance to all training sequences. We predict its class label as the label of the training sequence which has the smallest OPW distance with it among all training sequences.

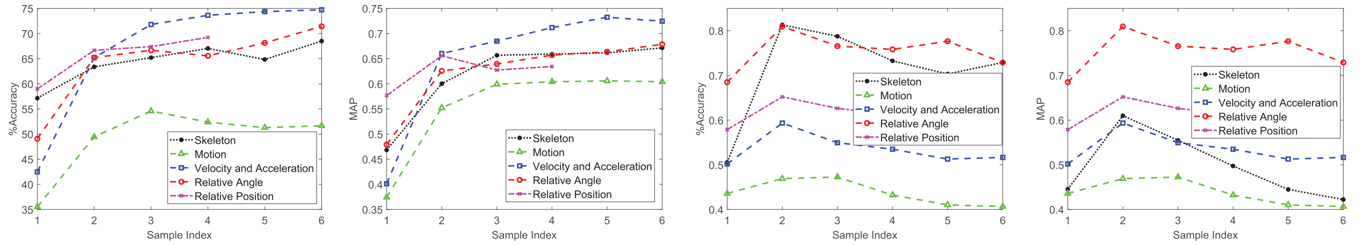
Performance Measures. We adopt the accuracy and MAP (mean average precision) as performance measures. For the SVM classifier, we train a multi-class SVM to evaluate the classification accuracy. We train a binary SVM for each class and use the scores to rank all training encoded vectors to evaluate the MAP. Additional evaluations by using the multi-class precision and recall as performance measures with this classifier are presented in the supplementary file , which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3050750. For the NN classifier, to evaluate the MAP, we view each test sequence as a query to rank all training sequences with the OPW distance.

5.3 Ablation Study

Influence of Hyper-Parameters. We set the values of the hyper-parameters λ_1, λ_2, and δ of OPW as suggested in [2] on the first three datasets when the same frame-wide features are used as in [2]. When using the raw-skeleton-based features, we set λ_1, λ_2, and δ of OPW on the MSR Action3D dataset to 10, 0.1, and 12, respectively, following the suggested setting on the MSR Activity3D dataset, since the two datasets are relatively similar. In [2], λ_2 was fixed to 0.1 for all datasets, and OPW is not sensitive to λ_1. Since λ_1, λ_2, and δ influence our method through OPW distance, our method should share similar sensitivities to them. The NTU dataset is not evaluated in [2]. We fix λ_1, λ_2, and δ of OPW to 10, 0.1, and 1, respectively.

In addition to the reduced dimension q', both linear OWDA and DeepOWDA only introduce one additional hyper-parameter, i.e., the length L of the barycenter per class. Fig. 2 shows the influence of L on linear OWDA on the MSR Action3D dataset with the 192-dimensional relative-angles-based features. When L is too small, the barycenter cannot capture enough temporal structures, and hence some temporal information is lost. When L is too large, the barycenter may contain some noisy elements, resulting in overfitting. In most cases, $L = 8$ achieves satisfactory results. We simply fix L to 8 in all the following experiments.

Training Time. For linear OWDA, in most cases, the calculation of the barycenter converges in about 10 iterations. The procedures after learning the barycenters are closed-form



(a) (b) (c) (d)

Fig. 3. Comparison of different types of frame-wide features. (a) Accuracies with the SVM classifier, (b) MAPs with the SVM classifier, (c) accuracies with the NN classifier, and (d) MAPs with the NN classifier as functions of the dimensionality of the subspace on the MSR Action3D dataset.

calculations. Therefore, the practical training time is not too long. On the MSR Action3D dataset with the 192-dimensional relative-angles-based features, the MSR Activity3D dataset with the 390-dimensional pairwise-joint-position-based features, and the Chalearn dataset with the 100-dimensional histogram-based features, the training times of linear OWDA are 43.1753, 265.7691, 385.8162 (sec), respectively.

Effects of Different Frame-Wide Features. The proposed OWDA and DeepOWDA can take sequences with any frame-wide features as input. We compare five different types of frame-wide features on the MSR Action3D dataset, including the 60-dimensional raw skeleton-based features where all joint locations in a frame are concatenated to form the feature; the 60-dimensional preprocessed motion-based features used in [40], the 120-dimensional frame-wide features based on velocity and acceleration of the joint positions used in [36], [63], the 192-dimensional pairwise-joint-angle-based features provided by the authors of [34], which are the relative angles of all the 3D joints w.r.t. other joints; and the 390-dimensional pairwise-joint-position-based features provided by the authors of [34], [57].

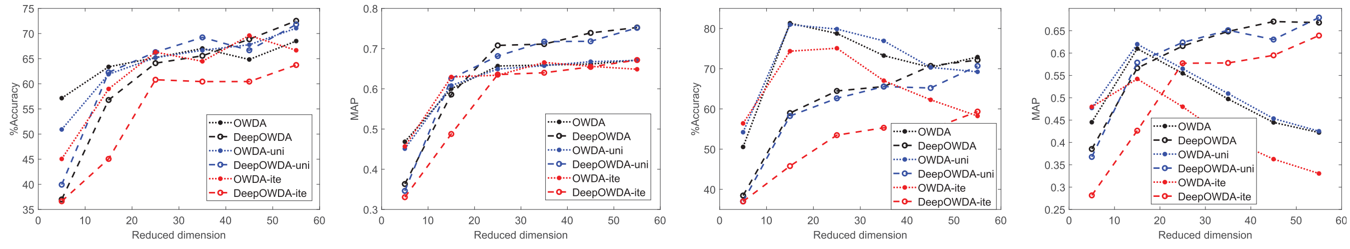
Results by OWDA are shown in Fig. 3 and results by DeepOWDA are shown in the supplementary file available online. The reduced dimension is uniformly sampled according to the total linearly independent dimensions and the x-axis in Fig. 3 represents the sampling index. The indexes of 1 to 6 correspond to dimensions of 5 to 55 with an interval of 10 for Skeleton and Motion-based features, 5 to 105 with an interval of 20 for velocity and acceleration features, 5 to 30 with an interval of 5 for the relative angle-based features, and 5 to 305 with an interval of 60 for relative position-based features. The dimensions of the relative position, velocity and acceleration-based frame-wide features are larger, so DeepOWDA can retain more dimensions, encode more information, and achieve better performances. On all datasets, the number of layers of the neural network and the number of hidden nodes in the middle layers are fixed to a large number. Due to the small size of the MSR Action3D dataset, DeepOWDA may overfit to the original joint positions. On the other hand, the relation position and motion-based features reduce the dependence on absolute positions. For linear OWDA, the raw skeleton-based features with a small dimension achieve performances comparable to other high-dimensional features. For different classifiers, the performances of different features are also different.

In order to simplify and clarify the process of using OWDA and DeepOWDA, in the following experiments, we use the raw skeleton-based frame-wide features on all datasets except the ChaLearn dataset, unless otherwise specified. On the ChaLearn data set, we directly employ the histogram-of-joint-positions-based frame-wide features provided by the authors of [37]. Specifically, for each frame, the relative locations of body joints are quantized w.r.t. a pre-clustered codebook, and the histogram of the quantized codewords serves as the feature with a dimensionality of 100. On the NTU dataset, some actions involve interactions between two subjects and all joints of both subjects are recorded. When only one subject appears in a frame, the corresponding joint positions of the second subject are set to 0. On this dataset, the dimensionality of the raw skeleton-based frame-wide feature is 150.

Influence of the Weight Sequence $\boldsymbol{\gamma}$. The barycenter learning algorithm jointly learns the supporting points and their weights. Approximately, each supporting point in the barycenter can be regarded as a stage or state of the sequence class and its weight can be viewed as the proportion of the duration of the stage. If a stage lasts for a long time in most samples of a class, then this stage may indeed be relatively important and its weight should be larger than other stages.

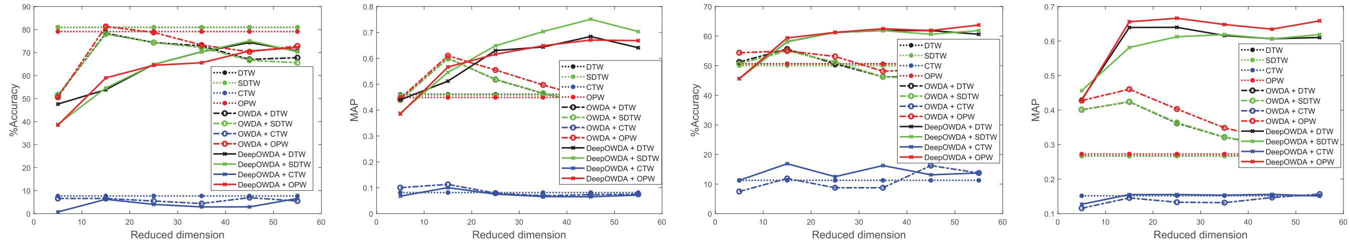
We can also fix the weights to uniform weights and only learn the supporting points when learning the barycenter. In this case, the proposed methods are denoted by OWDA-uni and DeepOWDA-uni, where the optimal transports are updated in procedure 1 and the supporting points are updated in procedure 2. Fig. 4 compares the performances of OWDA/DeepOWDA with learned weights and fixed uniform weights on the MSR Action3D dataset. We observe that OWDA-uni/DeepOWDA-uni achieves comparable results with OWDA/DeepOWDA. Since the length of barycenters is set much smaller than the average length of the training sequences, only stages with long enough durations can be captured. Therefore, the weights of all stages may not be too small and have less impact on performances. However, learning the weights jointly does not increase the learning complexity much, while can increase flexibility and may be useful when the length of barycenters is large.

Effect of the Iterative Solutions. OWDA and DeepOWDA approximate the optimal transports in the subspace by those in the original space. We compare them with OWDA-ite and DeepOWDA-ite which iteratively update the transformation and optimal transports. For OWDA-ite and DeepOWDA-ite, we iterate 5 times. The comparisons on the MSR Action3D dataset are shown in Fig. 4. The performance degradation of DeepOWDA-ite may be caused by the change in the amplitude of the transformed frame-wide features



(a) (b) (c) (d)

Fig. 4. Comparisons with OWDA-uni, DeepOWDA-uni, OWDA-ite, DeepOWDA-ite. (a) Accuracies with the SVM classifier, (b) MAPs with the SVM classifier, (c) accuracies with the NN classifier, and (d) MAPs with the NN classifier as functions of the dimensionality of the subspace on the MSR Action3D dataset.



(a) (b) (c) (d)

Fig. 5. Comparisons with using different sequence distances in the NN classifier. (a) Accuracies on the MSR Action3D dataset, (b) MAPs on the MSR Action3D dataset, (c) accuracies on the MSR Activity3D dataset, and (d) MAPs on the MSR Activity3D dataset as functions of the dimensionality of the subspace.

during iterations, which affects the inference of the optimal transports in the subspace by OPW. We observe that the differences in performances of OWDA-ite and OWDA are very small for the SVM classifier. Since the iterative process cannot guarantee the decrease of the objective function, OWDA-ite cannot improve the performances.

Classification With Different Sequence Distances. Although OWDA and DeepOWDA learn the transformation based on the OPW distance, other sequence distances can also be applied in the learned subspace. On the MSR Action3D dataset and the MSR Activity3D dataset, we use the nearest neighbor classifier with the DTW, SoftDTW (denoted by SDTW), and CTW distances to classify the original sequences and the transformed sequences, where CTW preserves 95% of energy. Fig. 5 compares the performances of using these distances and the OPW distance. All sequences are from the same modal, the maximally correlated subspace learned by CTW may not be discriminative and loses useful information, therefore, CTW performs inferior to other distances. Although OWDA and DeepOWDA construct separability among sequence classes based on the OPW distance, they can also greatly improve the DTW and SoftDTW distances. In some cases, the DTW and SoftDTW distances even outperform the OPW distance in the learned subspaces. This may be because the transformed sequences are more discriminative by enhancing temporal information and the DTW and SoftDTW distances with stricter temporal constraints can better separate them.

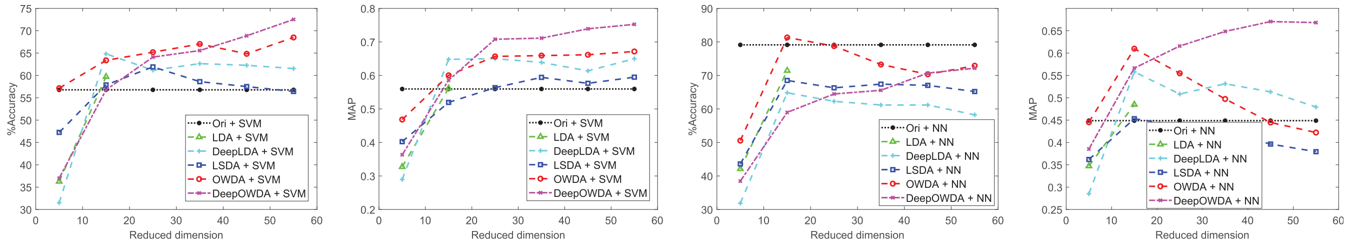
5.4 Comparison With Other DRS Methods

We compare the proposed OWDA and DeepOWDA with other dimensionality reduction methods for sequences. OWDA employs the Fisher criterion. As discussed in Section 2, different criteria are generally suited for different cases. In addition, OWDA can also be extended by employing other criteria. Therefore, to obtain a fair comparison, we only compare with those methods based on Fisher criterion, including LDA, DeepLDA, and LSDA. For LSDA, we use the same hyper-parameters as in [4], [5]. For both linear OWDA and DeepOWDA, the hyper-parameter L is fixed to 8 in all our experiments. We adapt the implementations of IBP in [54] and Newton's update in [53] to perform the computation of the OPW barycenter. We implement DeepOWDA using Keras with the Theano backend.

LDA and DeepLDA are based on the *i.i.d.* assumption. To apply them to sequence data, we view the observations in sequences as independent samples with the same class label. We employ the drtoolbox [64] to implement LDA. We employ Vahidoo's Keras code[1] to implement DeepLDA. In addition, we also evaluate the performances using both classifiers in the original space. Our implementation of OWDA and DeepOWDA is available[2].

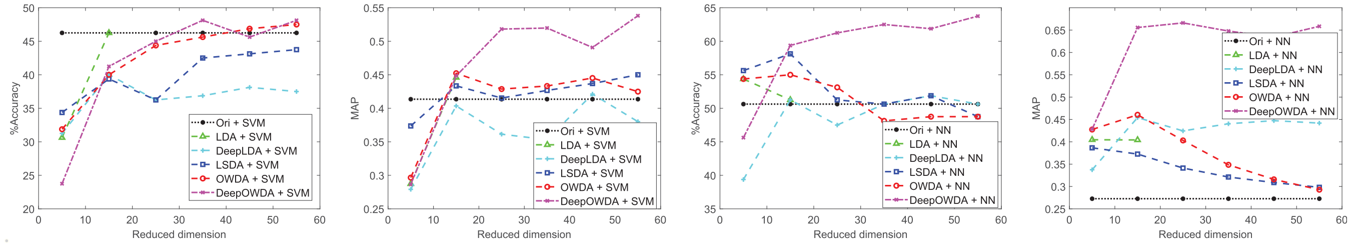
Results on the Action3D Dataset. On this dataset, the magnitude of the real-world raw skeleton data is not normalized. To avoid numerical problems when calculating OPW distances, when using the NN classifier in the subspace, we divide the absolute joint location coordinates by $\sqrt{2}$ and divide the transformed features by $\sqrt{2q'}$ and 2 for OWDA and DeepOWDA, respectively. The results of different DRS methods with different reduced dimensions are shown in Fig. 6. We can observe that the proposed linear OWDA outperforms other linear DRS methods by a significant margin with both classifiers. Compared with the original sequences with 60-dimensional observations, OWDA achieves better accuracy and MAP with a margin of more than 5% when

[1] https://github.com/VahidooX/DeepLDA
[2] https://github.com/BingSu12/OWDA



(a) (b) (c) (d)

Fig. 6. (a) Accuracies with the SVM classifier, (b) MAPs with the SVM classifier, (c) accuracies with the NN classifier, and (d) MAPs with the NN classifier as functions of the dimensionality of the subspace on the MSR Action3D dataset.



(a) (b) (c) (d)

Fig. 7. (a) Accuracies with the SVM classifier, (b) MAPs with the SVM classifier, (c) accuracies with the NN classifier, and (d) MAPs with the NN classifier as functions of the dimensionality of the subspace on the MSR Daily Activity3D dataset.

more than 25 dimensions are preserved for the SVM classifier, and achieves comparable accuracy and MAP using only 15 dimensions for the NN classifier.

DeepOWDA outperforms the non-linear DeepLDA by a large margin for both classifiers when more than 25 dimensions are preserved. It also achieves much higher MAPs than linear OWDA and other linear DRS methods. Compared with the original 60-dimensional observations, DeepOWDA using only more than 15 dimensions achieves better performances for the SVM classifier and improves the MAP by a margin of 10% for the NN classifier.

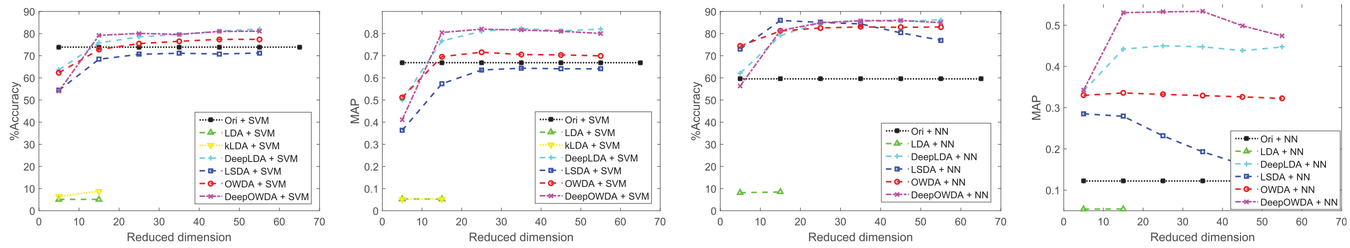
Results on the Activity3D Dataset. To avoid numerical problems, when using the NN classifier, we divide the absolute joint location coordinates by 2 and divide the transformed features by $\sqrt{q'}$ and 2 for OWDA and DeepOWDA, respectively. Fig. 7 depicts the performances of different DRS methods as functions of the reduced dimension by both classifiers on the Activity3D dataset. For the SVM classifier, OWDA and DeepOWDA with more than 35 dimensions achieve slightly better accuracies than classifying the original sequences directly without any DRS methods. Generally, OWDA and DeepOWDA also outperform other DRS methods.

For the NN classifier, LSDA generally obtains better accuracies than linear OWDA, but OWDA achieves much better MAPs than other linear DRS methods. DeepOWDA achieves the best accuracy and MAP. Especially, DeepOWDA outperforms other methods by a margin of about 20% on MAP. For a test sequence, the NN classifier only employs its nearest training sequence when calculating the accuracy, but ranks all training sequences according to the OPW distances w.r.t. it when calculating the MAP. The objective of OWDA and DeepOWDA is to minimize the overall dispersion for sequence classes and maximize the overall separability among classes. This makes most sequences from different classes more different, but does not pay special attention to the margins among classes. For a test sequence, the nearest training sequence may not belong to the same class due to noises or variances, but generally, most training sequences from the same class will be ranked in front of those from different classes.

Results on the ChaLearn Dataset. Fig. 8 presents the results of different DRS methods as functions of the reduced dimension by both classifiers on the ChaLearn dataset. For the SVM classifier, DeepOWDA performs comparable with DeepLDA, and both outperform linear methods. Linear OWDA outperforms other linear methods by a margin of about 5% on most reduced dimensions. OWDA is the only linear DRS method that is able to improve the original features. Moreover, OWDA achieves this by preserving only 25 dimensions. This indicates that OWDA enhances the temporal separability and discards noises successfully.

The performances of LDA and kLDA are far below those of other methods. The reason is that the observations in sequences are not independent. Performing LDA and kLDA forcibly by viewing them as independent samples not only aggravates the within-class ambiguity, but also may break their temporal relations. Moreover, LDA and kLDA can preserve $C-1=19$ dimensions at most. It is difficult to separate sequences from different classes with such few dimensions. In contrast, since the barycenter of each class has $L=8$ supporting points, OWDA is able to preserve $LC-1=159$ dimensions, if $d > 159$.

For the NN classifier, OWDA, LSDA, DeepOWDA, and DeepLDA improve the original features greatly. Compared with LSDA, OWDA achieves comparable accuracy and much higher MAP. Specifically, OWDA outperforms the original features by a margin of 20%. The MAPs of OWDA are 5% higher than those of LSDA on almost all dimensions. Compared with DeepLDA, DeepOWDA achieves comparable



(a) (b) (c) (d)

Fig. 8. (a) Accuracies with the SVM classifier, (b) MAPs with the SVM classifier (c) accuracies with the NN classifier, and (d) MAPs with the NN classifier as functions of the dimensionality of the subspace on the Chalearn Gesture dataset.

accuracy and much higher MAP. Specifically, DeepOWDA outperforms the original features by a margin of about 40% on MAP. The MAPs of DeepOWDA are 5% higher than those of DeepLDA on most dimensions.

Results on the NTU Dataset. Due to the large number of sequence samples, calculating the intra-class and inter-class scatters from all frames in all training sequences to obtain the linear transformations is prohibitively time-consuming, and simple SVM and NN classifiers may be less effective. Therefore, we only evaluate DeepLDA and DeepOWDA with mini-batch based optimization, and use the deep independent recurrent neural network (IndRNN) [44], [65] for classification in the learned subspaces. In [44], [65], a preprocessing alignment is applied to the original skeleton data so that the joint locations of the same subject identity lie in the same data array over time. Since the processed data are not provided and there is no explanation of how such alignment is performed, we only use the unaligned skeleton-based frame-wise features, this leads to degraded performances. We re-implement IndRNN on the same unaligned data for a fair comparison. All the hyperparameters of IndRNN and experimental settings on this dataset remain the same as in [44], [65]. The comparisons are shown in Fig. 9. DeepOWDA outperforms DeepLDA in both CS and CV settings.

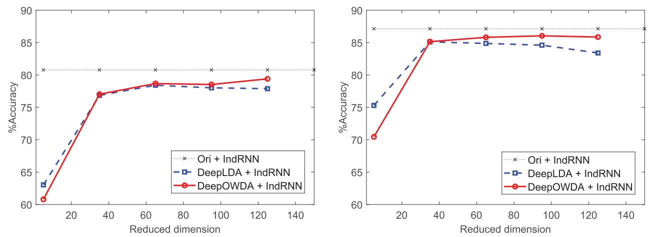
5.5 Comparison With State-of-the-Art Methods

Our goal is not to design an end-to-end sequence classification method, but to develop a DRS method that produces low-dimensional discriminative temporal representations. Our method can serve as a ubiquitous component in different classification pipelines to improve the original representations and benefit the subsequent classifiers. For example, recurrent neural networks (RNNs) are seldom used for feature learning, but often as classifiers by taking handcrafted or CNN-learned frame-wide features as input. Our method can be applied to these features before they are fed into RNNs. In this way, RNNs can estimate fewer parameters and better capture the temporal dependencies.

On the ChaLearn dataset, we have shown that our method outperforms other DRS methods and improves different sequence classification methods. We compare our results by using the frame-wide features in [37] and the SVM-based classifier with some other methods. Multi-class precision, recall, and F-score are used as performance measures as in [5], [28], [61], [66], [67]. Comparisons are shown in Table 1. DeepOWDA followed by a relatively simple SVM classifier with rank pooling outperforms other methods significantly using only 55 percent of the original dimension.

On the MSR Activity3D dataset, covariance representations and kernel-SVM based methods such as Ker-RP-POL [36] and Kernelized-COV [68] achieve superior results. Kernelized-COV employs the Kernelized covariance of all frame-wide features of a sequence as the representation of the sequence. Our proposed OWDA can be applied before Kernelized-COV to enhance the temporal representations. Specifically, we employ the 120-dimensional velocity-and-acceleration-based frame-wide features provided in [36]. We perform the proposed OWDA to reduce the dimension to 80 and then employ Kernelized-COV for classification. As shown in Tab. 2, the result obtained by the linear OWDA in this way has already outperformed the state-of-the-art results, so we did not evaluate DeepOWDA on this dataset.

On the MSR Action3D dataset, we extract the 120-dimensional velocity-and-acceleration-based frame-wide features ourselves, reduce the dimension to 80 by OWDA, and use Kernelized-COV for classification. As in Fig. 5 of the supplementary file available online, we also apply different dimensionality reduction methods to such features and use



(a) (b)

Fig. 9. Accuracies with the IndRNN classifier as functions of the dimensionality of the subspace in (a) the CS setting and (b) the CV setting on the NTU RGB+D dataset.

TABLE 1
Comparison With Other Methods on the ChaLearn Dataset

Method	Precision	Recall	F-score
Wu *et al.* [66]	0.599	0.593	0.596
Pfister *et al.* [61]	0.612	0.623	0.617
Fernando *et al.* [67]	0.753	0.751	0.752
Cherian *et al.* [28]	0.753	0.752	0.751
LSDA+SVM [5]	0.768	0.767	0.767
LT-LDA+SVM [33]	0.784	0.783	0.783
OWDA+SVM	0.773	0.773	0.772
DeepOWDA+SVM	**0.827**	**0.826**	**0.826**

TABLE 2
Comparison With State-of-the-Art Methods on the MSR Activity3D Dataset

Method	Accuracy
Actionlet Ensemble [34]	85.8%
Moving Pose [63]	73.8%
COV-$J_{\mathcal{H}}$-SVM [35]	75.5%
Ker-RP-RBF [36]	96.3%
Kernelized-COV [68]	96.3%
LRTS [69]	80.6%
Qiao *et al.* [70]	75.0%
Baradel *et al.* [71]	90.0%
Luo *et al.* [72]	86.9%
Ji *et al.* [73]	81.3%
DSSCA SSLM [74]	97.5%
MDMTL [75]	93.8%
OWDA+Kernelized-COV	**98.1%**

SVM or NN for classification. Comparisons are shown in Table 3. We observe that DeepOWDA using a relatively simple classifier obtains comparable results with LSTM-based models.

On the NTU RGB+D dataset, we use the proposed DeepOWDA to reduce the 150-dimensional original skeleton based frame-wise features to 95 and employ the densely connected IndRNN for classification. Different from the experiments in Fig. 9, since the dimension is reduced, we reduce the number of filters in the first dense layer by half and keep the growth rate unchanged. As a result, the number of parameters is reduced from 2,314,428 to 1,804,740. The comparisons with RNN-based methods without data augmentation are shown in Table 4, where “IndRNN” indicates the results reported in [44], where the original skeleton based frame-wise features are preprocessed by aligning the subject identities, and “Ori+IndRNN” indicates the results of IndRNN by directly taking the original skeleton based frame-wise features as input.

TABLE 3
Comparison With State-of-the-Art Methods on the MSR Action3D Dataset

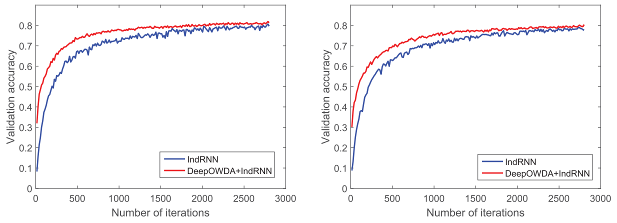
Method	Accuracy
Actionlet Ensemble [34]	88.2%
Moving Pose [63]	91.7%
COV-$J_{\mathcal{H}}$-SVM [35]	80.4%
Ker-RP-RBF [36]	**96.9%**
Kernelized-COV [68]	96.2%
GRP [28]	81.7%
LT-LDA+Kernelized-COV [33]	91.9%
TS-LSTM-GM [40]	91.2%
LT-LDA+LSTM-GM [33]	92.7%
FTP-SVM [41]	90.0%
Bi-LSTM [41]	86.2%
OWDA+Kernelized-COV	87.6%
LDA+SVM	38.1%
LSDA+SVM	67.8%
DeepLDA+SVM	84.6%
OWDA+SVM	74.7%
DeepOWDA+SVM	92.3%
DeepLDA+NN	78.8%
DeepOWDA+NN	93.8%

TABLE 4
Comparison With State-of-the-Art Methods on the NTU RGB+D Dataset

Method	CS	CV
PLSTM [62]	62.9%	70.3%
SkeletonNet [76]	75.9%	81.2%
Clips+CNN+MTLN [77]	79.6%	84.8%
Enhanced Visualization+CNN [78]	80.0%	87.2%
HCN [79]	86.5%	91.1%
TCN+TTN [80]	77.6%	84.3%
JL_d+RNN [81]	70.3%	82.4%
STA-LSTM [82]	73.4%	81.2%
Pose conditioned STA-LSTM [71]	77.1%	84.5%
ST-LSTM [43]	69.2%	77.7%
EleAtt-GRU [83]	79.8%	87.1%
TS-SAN [84]	**87.2%**	**92.7%**
SkeleMotion + Yang *et al.* [85]	76.5%	84.7%
ARRN-LSTM [86]	80.7%	88.8%
IndRNN [44]	84.9%	90.4%
Ori + IndRNN	80.8%	87.1%
DeepOWDA + IndRNN	79.0%	86.6%

DeepOWDA performs slightly inferior to the original features. The possible reasons are as follows. 1. All joint positions contain useful information for distinguishing the large number of actions. Reducing the dimension by DeepOWDA loses some discriminative information. 2. Sequences in this large-scale dataset show large within-class variations and may not be sufficiently represented by a single barycenter per class. Adding the number of barycenters per class may further increase the performances of DeepOWDA. 3. IndRNN and DeepOWDA have different objective functions and distinguish sequences in different ways. Maximizing the separability constructed by DeepOWDA will not necessarily preserve or enhance the discriminative information required by IndRNN. 4. We directly use IndRNN with hyper-parameters tuned for the original sequences to classify the transformed sequences by DeepOWDA. Using a validation set to select appropriate hyper-parameters may further improve the final performance.

The performance of DeepOWDA is gapped w.r.t state-of-the-art results. The overall performance may be related to many factors, such as preprocessing, hyper-parameters, computing resources, and classifiers. E.g., applying advanced skeleton-based action classification methods on the subspace learned by DeepOWDA may further improve the final performance. However, since our goal is not to achieve state-of-the-art results on this specific dataset, we did not perform any pre-processing or tune the hyper-parameters. We aim at evaluating the effectiveness of the proposed dimensionality reduction method. As shown in Table 4, DeepOWDA using only 63.3% of the original dimensions obtains results comparable to the original features by the IndRNN classifier. After transforming the sequences by DeepOWDA, IndRNN can not only adopt a lighter weight model with much fewer learnable parameters, but also converge faster during training, as shown in Fig. 10. This is especially suitable in resource-constrained situations.



(a) (b)

Fig. 10. The frame-level accuracy of IndRNN on the validation set as a function of the number of iterations by taking the original skeleton features and the transformed features with DeepOWDA as input on the NTU dataset for the (a) CS and (b) CV setting.

6 Conclusion

In this paper, we have presented a linear DRS method, *i.e.*, OWDA, and its deep extension, *i.e.*, DeepOWDA, to map the non-independent observations in sequences onto a low-dimensional subspace, so that the entire sequences from different classes are better discriminated with the OPW distance. To manipulate the structured sequences with various lengths, we learn the OPW barycenter of the sequence samples from a class to represent the average temporal structures and evolutions. We construct the covariance of the class in such a way that the trace of the covariance measures the variability of the OPW distances between the sequence samples and the barycenter. Similarly, we construct the pair-wise inter-class scatter so that the trance of the scatter measures the OPW distance between the corresponding barycenters of the two classes. We show that the intra-class and inter-class scatters are actually the weighted sums of all the pairwise outer-products between observations in sequences or elements of barycenters. Therefore, all local relationships are learned and incorporated. Experimental results on four 3D action datasets demonstrate the effectiveness of the proposed OWDA and DeepOWDA.

Acknowledgments

The authors would like to thank the associate editor and anonymous reviewers for their valuable comments. This work was supported in part by the National Natural Science Foundation of China under Grants 61976206 and 61832017, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019 100020098, CCF-Tencent Open Fund RAGR20200110, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China, National Science Foundation grant IIS-1619078, IIS-1815561, and the Army Research Office ARO W911NF-16-1-0138. This work was also supported in part by Public Computing Cloud, Renmin University of China.

References

[1] B. Su and G. Hua, “Order-preserving wasserstein distance for sequence matching,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1049–1057.

[2] B. Su and G. Hua, “Order-preserving optimal transport for distances between sequences,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 2961–2974, Dec. 2019.

[3] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Trans. Acoust., Speech Signal Process.*, vol. TASSP-26, no. 1, pp. 43–49, Feb. 1978.

[4] B. Su and X. Ding, “Linear sequence discriminant analysis: A model-based dimensionality reduction method for vector sequences,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 889–896.

[5] B. Su, X. Ding, H. Wang, and Y. Wu, “Discriminative dimensionality reduction for multi-dimensional sequences,” *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 40, no. 1, pp. 77–91, Jan. 2018.

[6] B. Su, X. Ding, C. Liu, H. Wang, and Y. Wu, “Discriminative transformation for multi-dimensional temporal sequences,” *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3579–3593, Jul. 2017.

[7] B. Su, J. Zhou, and Y. Wu, “Order-preserving wasserstein discriminant analysis,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9885–9894.

[8] L. F. Chen, H. Y. M. Liao, M. T. Ko, J. C. Lin, and G. J. Yu, “A new LDA-based face recognition system which can solve the small sample size problem,” *Pattern Recognit.*, vol. 33, no. 10, pp. 1713–1726, 2000.

[9] J. Ye, R. Janardan, Q. Li, and H. Park, “Feature extraction via generalized uncorrelated linear discriminant analysis,” in *Proc. 21st ACM Int. Conf. Mach. Learn.*, 2004, p. 113.

[10] J. Ye, “Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems,” *J. Mach. Learn. Res.*, vol. 6, no. Apr, pp. 483–502, 2005.

[11] M. Loog and R. Duin, “Linear dimensionality reduction via a heteroscedastic extension of LDA: The chernoff criterion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 732–739, Jun. 2004.

[12] M. Zhu and A. Martinez, “Subclass discriminant analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1274–1286, Aug. 2006.

[13] W. Bian and D. Tao, “Max-min distance analysis by using sequential SDP relaxation for dimension reduction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 1037–1050, May 2011.

[14] Y. Zhang and D.-Y. Yeung, “Worst-case linear discriminant analysis,” in *Adv. Neural Inform. Process. Syst.*, pp. 2568–2576, 2010.

[15] B. Su, X. Ding, C. Liu, and Y. Wu, “Heteroscedastic max-min distance analysis,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4539–4547.

[16] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, “Graph embedding and extensions: A general framework for dimensionality reduction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jun. 2007.

[17] R. Flamary, M. Cuturi, N. Courty, and A. Rakotomamonjy, “Wasserstein discriminant analysis,” *Mach. Learn.*, vol. 107, no. 12, pp. 1923–1945, 2018.

[18] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, “Fisher discriminant analysis with kernels,” in *Proc. IEEE 9th Signal Process. Soc. Workshop Neural Netw, Signal Process.*, 1999, pp. 41–48.

[19] M. Dorfer, R. Kelz, and G. Widmer, “Deep linear discriminant analysis,” in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–13.

[20] A. Shyr, R. Urtasun, and M. I. Jordan, “Sufficient dimension reduction for visual sequence classification,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3610–3617.

[21] F. Zhou and F. Torre, “Canonical time warping for alignment of human behavior,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 2286–2294.

[22] F. Zhou and F. De la Torre, “Generalized time warping for multi-modal alignment of human motion,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1282–1289.

[23] 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.

[24] G. Trigeorgis, M. A. Nicolaou, S. Zafeiriou, and B. W. Schuller, “Deep canonical time warping,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5110–5118.

[25] 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.

[26] M. Cuturi and M. Blondel, “Soft-DTW: A differentiable loss function for time-series,” in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 894–903.

[27] A. Cherian, S. Sra, S. Gould, and R. Hartley, “Non-linear temporal subspace representations for activity recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2197–2206.

[28] A. Cherian, B. Fernando, M. Harandi, and S. Gould, “Generalized rank pooling for activity recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3222–3231.

[29] R. Lajugie, D. Garreau, F. Bach, and S. Arlot, “Metric learning for temporal sequence alignment,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 1817–1825.

[30] G. Huang, C. Guo, M. J. Kusner, Y. Sun, F. Sha, and K. Q. Weinberger, "Supervised word mover's distance," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4862–4870.

[31] B. Paaßen, C. Gallicchio, A. Micheli, and B. Hammer, "Tree edit distance learning via adaptive symbol embeddings," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3976–3985.

[32] B. Su and Y. Wu, "Learning low-dimensional temporal representations," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4761–4770.

[33] B. Su and Y. Wu, "Learning low-dimensional temporal representations with latent alignments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 11, pp. 2842–2857, Nov. 2020.

[34] J. Wang, Z. Liu, and Y. Wu, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1290–1297.

[35] M. Harandi, M. Salzmann, and F. Porikli, "Bregman divergences for infinite dimensional covariance matrices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1003–1010.

[36] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li, "Beyond covariance: Feature representation with nonlinear kernel matrices," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4570–4578.

[37] B. Fernando, E. Gavves, J. O. M., A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5378–5387.

[38] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4041–4049.

[39] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 20–27.

[40] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1012–1020.

[41] A. Ben Tanfous, H. Drira, and B. Ben Amor, "Coding Kendall's shape trajectories for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2840–2849.

[42] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1647–1656.

[43] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3007–3021, Dec. 2018.

[44] S. Li, W. Li, C. Cook, Y. Gao, and C. Zhu, "Deep independently recurrent neural network (indrnn)," 2019, *arXiv: 1910.06251*.

[45] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12 026–12 035.

[46] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3595–3603.

[47] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1227–1236.

[48] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7912–7921.

[49] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Symbiotic graph neural networks for 3D skeleton-based human action recognition and motion prediction," 2019, *arXiv : 1910.02212*.

[50] J. D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, "Iterative bregman projections for regularized transportation problems," *SIAM J. Sci. Comput.*, vol. 37, no. 2, pp. A1111–A1138, 2014.

[51] L. M. Brègman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *Ussr Comput. Math. Math. Phys.*, vol. 7, no. 3, pp. 200–217, 1967.

[52] H. H. Bauschke and A. S. Lewis, "Dykstras algorithm with bregman projections: A convergence proof," *Optimization*, vol. 48, no. 4, pp. 409–427, 2000.

[53] M. Cuturi and A. Doucet, "Fast computation of wasserstein barycenters," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 685–693.

[54] J. Ye, P. Wu, J. Z. Wang, and J. Li, "Fast discrete distribution clustering using wasserstein barycenter with sparse support," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2317–2332, May 2017.

[55] A. Petersen and H.-G. Müller, "Wasserstein covariance for multiple random densities," *Biometrika*, vol. 106, no. 2, pp. 339–351, 2019.

[56] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *Proc. IEEE Int. Workshop CVPR Hum. Communicative Behavior Anal.*, 2010, pp. 9–14.

[57] J. Wang and Y. Wu, "Learning maximum margin temporal warping for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2688–2695.

[58] S. Escalera *et al.*, "Multi-modal gesture recognition challenge 2013: Dataset and results," in *Proc. 15th ACM Int. Conf. Multimodal Interact.*, 2013, pp. 445–452.

[59] S. Escalera *et al.*, "Chalearn multi-modal gesture recognition 2013: Grand challenge and workshop summary," in *Proc. 15th ACM Int. Conf. Multimodal Interact.*, 2013, pp. 365–368.

[60] A. Yao, L. Van Gool, and P. Kohli, "Gesture recognition portfolios for personalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1915–1922.

[61] T. Pfister, J. Charles, and A. Zisserman, "Domain-adaptive discriminative one-shot learning of gestures," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 814–829.

[62] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1010–1019.

[63] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2752–2759.

[64] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[65] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (indRNN): Building a longer and deeper RNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5457–5466.

[66] J. Wu, J. Cheng, C. Zhao, and H. Lu, "Fusing multi-modal features for gesture recognition," in *Proc. 15th ACM Int. Conf. Multimodal Interact.*, 2013, pp. 453–460.

[67] B. Fernando, E. Gavves, J. O. M., A. Ghodrati, and T. Tuytelaars, "Rank pooling for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 773–787, Apr. 2017.

[68] J. Cavazza, A. Zunino, M. San Biagio, and V. Murino, "Kernelized covariance for action recognition," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 408–413.

[69] C. Jia and Y. Fu, "Low-rank tensor subspace learning for RGB-D action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4641–4652, Oct. 2016.

[70] R. Qiao, L. Liu, C. Shen, and A. van den Hengel, "Learning discriminative trajectorylet detector sets for accurate skeleton-based action recognition," *Pattern Recognit.*, vol. 66, pp. 202–212, 2017.

[71] F. Baradel, C. Wolf, and J. Mille, "Pose-conditioned spatio-temporal attention for human action recognition," 2017, *arXiv : 1703.10106*.

[72] Z. Luo, B. Peng, D.-A. Huang, A. Alahi, and L. Fei-Fei, "Unsupervised learning of long-term motion dynamics for videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2203–2212.

[73] X. Ji, J. Cheng, W. Feng, and D. Tao, "Skeleton embedded motion body partition for human action recognition using depth sequences," *Signal Process.*, vol. 143, pp. 56–68, 2018.

[74] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang, "Deep multimodal feature analysis for action recognition in RGB+ D videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1045–1058, May 2018.

[75] A.-A. Liu, N. Xu, W.-Z. Nie, Y.-T. Su, and Y.-D. Zhang, "Multi-domain and multi-task learning for human action recognition," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 853–867, Feb. 2019.

[76] Q. Ke, S. An, M. Bennamoun, F. Sohel, and F. Boussaid, "SkeletonNet: Mining deep part features for 3-D action recognition," *IEEE Signal Process. Lett.*, vol. 24, no. 6, pp. 731–735, Jun. 2017.

[77] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3288–3297.

[78] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognit.*, vol. 68, pp. 346–362, 2017.

[79] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 786–792.

[80] S. Lohit, Q. Wang, and P. Turaga, "Temporal transformer networks: Joint learning of invariant and discriminative time warping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12 426–12 435.

[81] S. Zhang, X. Liu, and J. Xiao, "On geometric features for skeleton-based action recognition using multilayer LSTM networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2017, pp. 148–157.

[82] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4263–4270.

[83] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng, "EleAtt-RNN: Adding attentiveness to neurons in recurrent neural networks," *IEEE Trans. Image Process.*, vol. 29, pp. 1061–1073, Sep. 2019.

[84] S. Cho, M. H. Maqbool, F. Liu, and H. Foroosh, "Self-attention network for skeleton-based human action recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, 635–644,

[85] C. Caetano, J. Sena, F. Brémond, J. A. Dos Santos, and W. R. Schwartz, "Skelemotion: A new representation of skeleton joint sequences based on motion information for 3D action recognition," in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveillance*, 2019, pp. 1–8.

[86] W. Zheng, L. Li, Z. Zhang, Y. Huang, and L. Wang, "Relational network for skeleton-based action recognition," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2019, pp. 826–831.

Bing Su received the BS degree in information engineering from the Beijing Institute of Technology, Beijing, China, in 2010, and the PhD degree in electronic engineering from Tsinghua University, Beijing, China, in 2016. From 2016 to 2020, he worked with the Institute of Software, Chinese Academy of Sciences, Beijing. Currently, he is an associate professor at the Gaoling School of Artificial Intelligence, Renmin University of China. His research interests include pattern recognition, computer vision, and machine learning.

Jiahuan Zhou (Member, IEEE) received the BE degree from Tsinghua University, in 2013, and the PhD degree from the Department of Electrical Engineering and Computer Science, Northwestern University, 2018. From 2019 to 2020, he was a postdoctoral fellow in Northwestern University. Currently, he is a research assistant professor at Northwestern University. His current research interests include computer vision, pattern recognition, visual identification, and machine learning.

Ji-Rong Wen (Senior Member, IEEE) is currently a full professor at the Gaoling School of Artificial Intelligence, Renmin University of China. He worked at Microsoft Research Asia for 14 years and many of his research results have been integrated into important Microsoft products (e.g., Bing). He serves as an associate editor of *ACM Transactions on Information Systems*. He is a program chair of SIGIR 2020. His main research interests include web data management, information retrieval, data mining, and machine learning.

Ying Wu (Fellow, IEEE) received the BS degree from the Huazhong University of Science and Technology, Wuhan, China, in 1994, the MS from Tsinghua University, Beijing, China, in 1997, and the PhD degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, Illinois, in 2001. From 1997 to 2001, he was a research assistant at the Beckman Institute for Advanced Science and Technology at UIUC. During summer 1999 and 2000, he was a research intern with Microsoft Research, Redmond, Washington. In 2001, he joined the Department of Electrical and Computer Engineering at Northwestern University, Evanston, Illinois, as an assistant professor. He was promoted to an associate professor, in 2007 and a full professor, in 2012. He is currently a full professor of electrical engineering and computer science at Northwestern University. His current research interests include computer vision, robotics, image and video analysis, pattern recognition, machine learning, multimedia data mining, and human-computer interaction. He serves as the associate editor-in-chief for *APR Journal of Machine Vision and Applications*, and associate editors for the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Circuits and Systems for Video Technology*, *SPIE Journal of Electronic Imaging*. He served as program chair and area chairs for CVPR, ICCV and ECCV. He received the Robert T. Chien Award at UIUC, in 2001, and the NSF CAREER award, in 2003. He is a fellow of the IAPR.