



Partial ordered Wasserstein distance for sequential data

Tung Doan, Tuan Phan, Phu Nguyen, Khoat Than, Muriel Visani, Atsuhiko Takasu

► To cite this version:

Tung Doan, Tuan Phan, Phu Nguyen, Khoat Than, Muriel Visani, et al.. Partial ordered Wasserstein distance for sequential data. *Neurocomputing*, 2024, 595, pp.127908. 10.1016/j.neucom.2024.127908 . hal-04956335

HAL Id: hal-04956335

<https://hal.science/hal-04956335v1>

Submitted on 26 Mar 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Highlights

Partial Ordered Wasserstein Distance for Sequential Data

Tung Doan, Tuan Phan, Phu Nguyen, Khoat Than, Muriel Visani, Atsuhiko Takasu

- A novel method for calculating the distance between sequences is proposed. Building upon the optimal transport framework with limited transportation, the resulted distance is flexible when aligning the two sequences and robust on sequential data that is contaminated by outliers.
- The paper further analyzes the properties of the proposed distance. Based on that, an effective procedure is introduced to automatically and adaptively select the amount of transported mass that is crucial for the outlier robustness of the proposed distance.
- The application of the proposed method are studied in two tasks, including time-series classification and multi-step localization.
- Extensive experiments are conducted on widely available public datasets to evaluate the performance of the proposed distance.

Partial Ordered Wasserstein Distance for Sequential Data

Tung Doan^{a,*}, Tuan Phan^a, Phu Nguyen^a, Khoat Than^a, Muriel Visani^b and Atsuhiko Takasu^c

^a*School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, 11615, Vietnam*

^b*La Rochelle Université, Laboratoire Informatique, Image et Interaction (L3i), La Rochelle, 17042, France*

^c*National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda, Tokyo, 101-8430, Japan*

ARTICLE INFO

Keywords:

Optimal transport
Outlier robustness
Sequence alignment
Time-series classification
Multi-step localization

ABSTRACT

Measuring the distance between data sequences is a challenging problem, especially in the presence of outliers and local distortions. Existing measures typically align the two sequences before calculating their distance based on the difference between the corresponding elements. However, those alignments are not flexible enough to accommodate local distortions and severe effects of outliers. In this article, we propose a novel distance, termed as *Partial Ordered Wasserstein* (POW), which is flexible to align two sequences and robust w.r.t outliers. We further analyze some properties of the proposed distance, and show that POW enables a simple way to automatically and adaptively select the amount of transported mass, so as to accommodate outliers. Two different applications of POW are then studied: time-series classification and multi-step localization. Finally, we conduct extensive experiments on widely available public datasets to evaluate the performance of the proposed distances. Experimental results, obtained via a thorough experimental protocol, show the performance superiority of POW over several existing distance measures. Our Python source code is available on <https://github.com/TungDP/Partial-Ordered-Wasserstein-Distance>

1. Introduction

Sequential data is now ubiquitous in various fields, including computer vision [44, 32, 17], finance [6, 68], bioinformatics [40], and traffic monitoring [49, 48], to name a few. Machine learning and data mining tasks involving sequential data typically require an effective distance metric to compare sequences. However, finding such a distance is a challenging problem. Firstly, sequences can have different lengths, making conventional distance measures for vectors such as Euclidean distance or its variants mostly inapplicable. Secondly, local distortions often occur in sequential data. For instance, in a sequence of steps to perform the same action, such as making a Taco Salad, one person may add tortilla before adding meat, while another person may do the opposite. Thus, a proper distance metric should be able to recognize similarity despite such local swaps. Last but not least, sequential data often contains anomalous elements due to errors in the data collection process. Therefore, when measuring the distance between two sequences, there needs to be a mechanism to remove the influence of such outliers.


Many attempts have been made to define a meaningful distance between sequences. Among those efforts, Dynamic Time Warping (DTW) [62] is perhaps the most widely adopted distance. The calculation begins with aligning the two sequences using a dynamic programming algorithm to find correspondences between their elements. The final distance is then defined as the cumulative sum of differences between corresponding elements, measured by a ground

distance function. Although DTW is able to compare sequences with different lengths, it still suffers from two severe limitations. Firstly, it strictly adheres to the monotonicity constraint, that mandates an ascending order among the corresponding elements. As sequential data often exhibit local swaps and distortions, DTW may induce inaccurate alignments in such cases. Secondly, and more critically, DTW is sensitive to outliers. By enforcing correspondences between all elements of both sequences, DTW may align abnormal elements with normal ones in the presence of outliers, leading to suboptimal alignments and inaccurate distance measurements. The alignment example in Figure 1(a) demonstrates how DTW aligns sequences.

Recently, [64] opened up a new direction for solving the first limitation of DTW. More specifically, the authors view elements of the sequence as empirical samples from an unknown distribution. Alignment between the two sequences is then carried out using the Optimal Transport (OT) framework [70, 58, 71]. Slightly different from the original version of OT, the authors incorporate two order-preserving regularizations to take into account the differences in positions among the elements. As a result, this approach effectively relaxes the strict constraints in DTW and shows competitive performance on various sequential datasets [65, 8, 66]. Despite these advancements, OT-based distances are still hindered by the outlier issue. Like DTW, OT mandates transportation between all elements of both sequences. Consequently, in the presence of outliers in sequential data, OT-based methods incorporate anomalous elements in the alignment, resulting in deteriorated distance measures. Figure 1(b) shows an example of alignment obtained using OPW [64] – a representative of OT-based methods.

In this paper, to fully address the aforementioned issues of DTW, we introduce a new distance between sequences,

*Corresponding author

 tungdp@soict.hust.edu.vn (T. Doan);

tuan.pm194461@sis.hust.edu.vn (T. Phan); phu.nd194447@sis.hust.edu.vn (P. Nguyen); khoattq@soict.hust.edu.vn (K. Than);

muriel.visani@univ-lr.fr (M. Visani); takasu@nii.ac.jp (A. Takasu)

ORCID(s): 0000-0003-1798-9671 (T. Doan); 0000-0001-7513-4749 (M. Visani); 0000-0002-9061-7949 (A. Takasu)

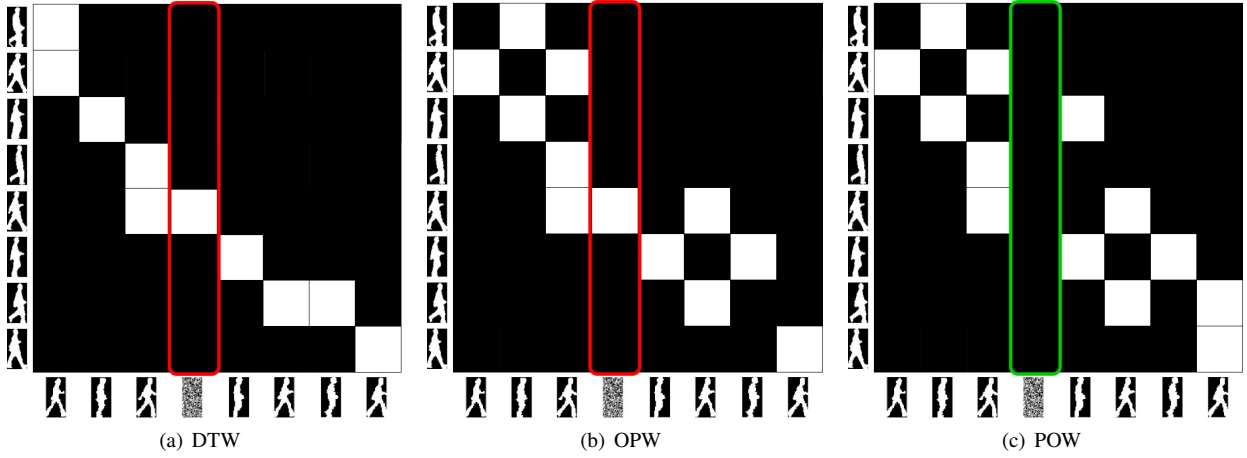


Figure 1: The alignment matrices returned by (a) DTW, (b) OPW, and (c) POW, respectively, for two video sequences featuring different subjects performing a walking action taken from the Weizmann dataset. In these matrices, if the entry in position (i, j) is represented in white, it indicates that the i^{th} frame of the first sequence is matched with the j^{th} frame of the second sequence, and vice versa. In the alignment produced by DTW, if two white cells are at positions (i_1, j_1) and (i_2, j_2) , then $i_1 < i_2$ implies $j_1 < j_2$. This is the monotonicity constraint that is relaxed in OPW and POW to produce more flexible alignments. It is important to note that the fourth frame of the second sequence is an outlier and should not be matched to any frame of the first sequence. POW achieves this condition, where the column is marked in green, while the corresponding columns obtained by DTW and OPW contain a white cell and are marked in red.

named *Partial Order Wasserstein* (POW). Its definition is also based on the optimal transport framework. However, unlike the existing OT-based distances that transport all the mass from one sequence to the other, POW limits the amount of transported mass. In this way, the mass from the abnormal elements of one sequence is prevented from being transported to elements of the other sequence, and vice-versa. As a result, it minimizes the impact of outliers on the distance calculation. In addition, we also incorporate a simple yet effective regularization term into the framework. This encourages the transportation between elements with relatively similar positions and prevents transportation between distant elements. Figure 1(c) illustrates the robust and flexible alignment returned by POW.

Compared to the regularization terms proposed in [64], our regularization term is much simpler and more effective. It has a reduced number of hyperparameters, decreasing dependence on the tuning process, which can be very tedious. In addition, by avoiding introducing a negative term into the objective function as in [64], our regularization makes the calculation procedure of the proposed distance more stable. We note that the robustness of POW is heavily dependent on the percentage of transported mass. Thus, we provide an analysis of the proposed distance's properties. Based on that, we introduce an effective procedure for automatically selecting the amount of mass to be transported.

Finally, we study applications of the proposed distance in different tasks: time series classification and multi-step localization. Extensive experiments were conducted on widely used public datasets to evaluate the performance of the proposed distance. The results verify that our approaches

are more robust and effective in comparison with existing distance measures for sequential data.

In summary, the contributions of this paper are as follows:

- Introduction of *Partial Order Wasserstein* (POW) - a novel distance measure for sequential data. Leveraging the optimal transport framework with limited transportation, POW offers flexibility in aligning two sequences and robustness against outliers affecting sequential data.
- Analysis of the POW distance's properties. This analysis leads to the development of an effective procedure for automatically and adaptively selecting the amount of transported mass, a crucial aspect for outlier robustness.
- Applications of POW in two tasks, namely time series classification and multi-step localization.
- Extensive experiments on widely used benchmark datasets to evaluate the performance of the proposed distance measure, and an in-depth discussion about these results.

The rest of this paper is organized as follows: In section 2, we briefly review existing distance measures for sequential data. We then present some background in section 3 to derive the definition of a new distance. Section 4 introduces an outlier-robust method for calculating a novel distance measure, termed as Partial Ordered Wasserstein (POW) distance, between two sequences. In this section, we also present an effective procedure for automatically and adaptively selecting the amount of transported mass. In section

5, we study applications of the proposed distance on two tasks: time series classification and multi-step localization. Extensive experiments for evaluating the performance of the proposed distance, as well as an in-depth discussion about these results, are provided in Section 6. Finally, section 7 concludes the paper.

Notations. We use the following conventions and notations throughout this paper. Lower-case letters in bold denote vectors, while capital letters in bold denote matrices. For a matrix \mathbf{A} , its entry at position (i, j) is denoted by $a_{i,j}$. $\delta_{\mathbf{x}_i}$ represents the Dirac unit mass concentrated at the element \mathbf{x}_i . The simplex in \mathbb{R}^N is denoted by $\Delta_N = \{\boldsymbol{\alpha} \in \mathbb{R}^N | \alpha_i \geq 0 \ \forall i \text{ and } \sum_{i=1}^N \alpha_i = 1\}$. $\langle \mathbf{A}, \mathbf{B} \rangle_F$ represents the Frobenius inner product of two matrices \mathbf{A} and \mathbf{B} , calculated as $\text{Tr}(\mathbf{A}^\top \mathbf{B})$. For $\boldsymbol{\kappa} \in \mathbb{R}^N$, $\text{diag}(\boldsymbol{\kappa})$ is the $N \times N$ matrix with diagonal $\boldsymbol{\kappa}$ and zero otherwise. We use the notation $\mathbf{1}_N$ to represent an N-dimensional column vector with all entries equal to one, and $\mathbf{0}_N$ represents an N-dimensional column vector with all entries equal to zero. The notation $\boldsymbol{\alpha}/\boldsymbol{\beta}$ denotes element-wise division between vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. We denote the probability vector of a uniform distribution over N elements as $\mathbf{u}_N = \left[\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}\right]^\top$. Note that, in this paper, the uniform distribution over elements within the interval (a, b) is also abbreviated by $\mathbf{u}_{(a,b)}$.

2. Related work

To the best of our knowledge, Dynamic Time Warping (DTW) [62] is the most widely used distance for sequential data. However, as mentioned above, it has two severe limitations that narrow its application range. More specifically, DTW is sensitive to outliers and excessively restrictive when aligning the two sequences. In this section, we briefly review existing DTW variants that attempt to improve its robustness. We then present a recent OT-based approach that aims to promote more flexible alignments.

Robust variants of DTW. [43] argued that abnormal instances often distort the local shape of sequences. This can lead DTW to produce skewed alignments and pathological distance results. To address this issue, the authors proposed Derivative Dynamic Time Warping (DDTW), which uses the first derivatives containing more robust local shape information of the sequences to obtain more meaningful alignments. In a similar vein, [78] introduced Shape Dynamic Time Warping (ShapeDTW), utilizing various local shape descriptors like Piecewise Aggregate Approximation (PAA) [42], Discrete Wavelet Transform (DWT), and slope to achieve more robust alignment. Slope is also exploited in [75]. However, the authors in [75] further integrated filtering and discretization techniques to reduce the impact of outliers. Another shape-based variant of DTW was Shape Segment Dynamic Time Warping (SSDTW) [34]. It divides the sequences into segments to characterize their local structures before carrying out alignment.

Locality is also considered by weighted-based variants of DTW for alleviating the influences of abnormal elements.

Specifically, [39] proposed Weighted Dynamic Time Warping (WDTW) that assigns weights to alignments, taking the local differences between elements of the sequences into account. It is widely known that the original Dynamic Time Warping (DTW) is a special case of WDTW when the weight for each alignment in DTW is set to 1. [49] further introduced Gaussian Weighted Dynamic Time Warping (GW-DTW). It is an extension of WDTW, where a Gaussian probability function was introduced for calculating the weights of alignments. [5] proposed to weight features of the elements instead of the alignments. Local stability, estimated through a sequential averaging method, is utilized in [56] to determine alignment weights.

Both shape-based and weighted-based approaches attempt to attenuate the effects of outliers. However, because they are based on the original DTW, their alignments still include the abnormal instances. In other words, these variants have no mechanism to remove outliers. Thus, they are forced to establish correspondences between all elements (including outliers) of the two sequences. As a consequence, both shape-based and weighted-based variants of DTW probably produce unreliable final distance results under the presence of outliers.

Recently, [24] has proposed Drop-DTW with additional costs for dropping outliers. However, as Drop-DTW remains the monotonicity constraint from the original DTW, it cannot produce flexible alignments as our measure does.

Our proposed distance measure, POW, is completely different from the above approaches. Based on Optimal Transport (OT) framework with limited transportation, POW is computed after eliminating all the outliers with proper selection of transported mass. In Section 6, we will empirically show the importance of its flexibility, which enables superior performance of our distance measure in various disciplines.

OT-based approach. In order to alleviate the inflexibility issue of DTW, [64] proposed the Order-Preserving Wasserstein (OPW) distance for sequential data. This measure considers elements of the sequences as empirical samples from an unknown distribution. The OPW distance is then computed after matching the elements between the sequences using the Optimal Transport (OT) framework [70, 58, 71]. To preserve the order among elements of the sequences, the authors employed a regularization term called Inverse Difference Moment (IDM) from [3] and assigned a Gaussian prior distribution on the transportation matrix. In comparison with our distance measure, OPW has two drawbacks. First, similar to DTW, OPW has no mechanism to remove outliers. The reason is that the OT framework enforces transporting all mass from one sequence to another. As a result, the mass from outliers is also transported. Note that, in the OT-based distances, the mass transported between any two elements represents the degree of their correspondence. Our measure, in contrast, enables eliminating abnormal elements from distance calculation. By limiting the amount of transported mass, our approach can prevent the transportation of outliers, thus reducing their effects on the distance measurement. Second, the IDM regularization

of OPW introduces a negative regularization term into the objective function of the OT problem. With an inappropriate selection of the corresponding regularization parameter, the matrix scaling algorithm for computing OPW becomes unstable and may never converge. Our distance is completely different. The regularization term introduced in this paper preserves the positivity of the objective function. As a result, the proposed distance measure is more stable and can avoid the non-convergence issue.

Following the same direction, [35] introduced Grouped OPW (G-OPW), which extends OPW by considering group-wise matching instead of element-wise matching. [41] proposed replacing the Gaussian prior distribution in OPW with a prior distribution constructed based on neighbor relationships among the elements to account for data transitions. [8] developed the Wasserstein subsequence kernel (WSK), which divides sequences into segments of even length and then computes the distance based on the OT framework. Because these distances are extensions of OPW or are based on the original OT, they are still susceptible to outliers and offer unreliable distance measures. Recently, [46] applied unbalanced OT [50] to two normalized sequences. The measure only considers the difference in terms of position between elements of the two sequences. Thus, a closed-form distance called Optimal Transport Warping (OTW) is obtained with linear computational time. However, as the actual geometrical difference between elements is ignored, OTW is likely prone to error.

3. Background

Wasserstein distance. Given two sets of elements $X = \{x_1, \dots, x_N\}$ and $Y = \{y_1, \dots, y_M\}$ in the same space, their corresponding empirical probabilities are denoted by $\mu_x = \sum_{i=1}^N \alpha_i \delta_{x_i}$ and $\nu_y = \sum_{j=1}^M \beta_j \delta_{y_j}$, respectively. Here, N and M are the numbers of elements of the two sets X and Y , respectively. The vector $\alpha = [\alpha_1, \dots, \alpha_N]^\top$ formed by gathering all the weights α_i belongs to the simplex Δ_N . Similarly, $\beta = [\beta_1, \dots, \beta_M]^\top \in \Delta_M$. Without any further assumptions on the prior knowledge of the elements, one can set $\alpha = u_N$ and $\beta = u_M$ to uniform distributions. Given the pairwise ground distance matrix $D \in \mathbb{R}_+^{N \times M}$, where each entry $d_{i,j}$ represents a geometric distance between x_i and y_j , the problem of optimally transport μ_x toward ν_y is defined as follows:

$$W(\mu_x, \nu_y) = \min_{T \in \Pi(\alpha, \beta)} \langle D, T \rangle_F = \min_{T \in \Pi(\alpha, \beta)} \sum_{i=1}^N \sum_{j=1}^M d_{i,j} t_{i,j}. \quad (1)$$

Here, the minimum $W(\mu_x, \nu_y)$ is known as the Wasserstein distance [71] between the two empirical probability measures μ_x and ν_y . Since μ_x and ν_y are fixed to uniform distribution, hereafter, we also denote $W(X, Y)$ the Wasserstein distance between the two sets X and Y . $T \in \mathbb{R}^{N \times M}$ is a transport matrix, which belongs to a transportation polytope

$\Pi(\alpha, \beta)$:

$$\Pi(\alpha, \beta) = \{T \in \mathbb{R}_+^{N \times M} | T \mathbf{1}_M = \alpha, T^\top \mathbf{1}_N = \beta\} \quad (2)$$

The entry $t_{i,j}$ of the transport matrix describes the amount of mass from α_i at x_i transported toward the mass β_j at y_j . In the other view, we can consider the transport matrix T as soft alignment indicators. Each element $t_{i,j}$ represents the degree of correspondence between x_i and y_j .

Order-preserving Wasserstein distance. Wasserstein distance is ineffective on sequential data because it ignores the order relationship among elements of the sequences. Recently, [64] introduced two regularization terms to the original OT problem to preserve the order information within the sequences. The first regularization favors the transport matrix T with large *inverse difference moment* (IDM), which is computed as:

$$I(T) = \sum_{i=1}^N \sum_{j=1}^M \frac{t_{i,j}}{\left(\frac{i}{N} - \frac{j}{M}\right)^2 + 1} \quad (3)$$

The second regularization encourages T to be closed to the prior matrix P , whose elements is calculated as:

$$p_{i,j} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{l_{i,j}^2}{2\sigma^2}}, \quad (4)$$

where $l_{i,j} = \frac{|\frac{i}{N} - \frac{j}{M}|}{\sqrt{\frac{1}{N^2} + \frac{1}{M^2}}}$ and σ is a small variance. As consequences, definition of the Order-preserving Wasserstein (OPW) distance is given as follows:

$$\begin{aligned} OPW(X, Y) &= \langle D, T_{OPW}^* \rangle_F \\ \text{s.t. } T_{OPW}^* &= \underset{T \in \Pi(u_N, u_M)}{\operatorname{argmin}} \langle D, T \rangle_F - \lambda_1 I(T) + \lambda_2 KL(T || P) \end{aligned} \quad (5)$$

where X and Y are two sequences of elements, λ_1 and λ_2 are hyperparameters that correspond to the two regularization terms, and $KL(T || P)$ is the Kullback-Leibler divergence between transport matrix T and the prior matrix P . Solution of the problem (6) is obtained through a matrix scaling algorithm, which alternates between two steps

$$\kappa_2 \leftarrow \beta / K \kappa_1, \quad (7)$$

$$\kappa_1 \leftarrow \alpha / K \kappa_2, \quad (8)$$

where $K = e^{-\frac{1}{\lambda_2} \hat{D}}$ is the element-wise exponential of the matrix $-\frac{1}{\lambda_2} \hat{D}$ and each entry of \hat{D} is

$$\hat{d}_{i,j} = d_{i,j} - \frac{\lambda_1}{\left(\frac{i}{N} - \frac{j}{M}\right)^2 + 1} + \lambda_2 \left(\frac{l_{i,j}^2}{2\sigma^2} + \log(\sigma \sqrt{2\pi}) \right). \quad (9)$$

After convergence, the solution is computed as:

$$T_{OPW}^* = \operatorname{diag}(\kappa_1) K \operatorname{diag}(\kappa_2) \quad (10)$$

We can observe in equation (9) that IDM regularization implicitly adds a negative term into each entry of \hat{D} . When

λ_1 is often set to high value as recommended in [64], entries of \mathbf{K} are unbounded due to exponential operators on positive operands, leading to the well-known instability of the matrix scaling algorithm [58]. In addition, the two regularization terms of OPW introduces three additional hyperparameters, namely λ_1 , λ_2 and σ . In the original paper [64], the authors had to select values for these parameters through a large number of combinations, which is very laborious.

4. Partial ordered Wasserstein distance

In this section, we propose the *Partial Ordered Wasserstein* (POW) distance for sequential data. We first derive its definition as the solution to an optimization problem. We then demonstrate the connection of the proposed formulation with the original OT and propose an efficient algorithm to solve it. An analysis of POW is also provided. Based on this analysis, we finally introduce an effective procedure for automatically selecting the transported mass, which is vital for the outlier robustness of the proposed distance.

4.1. Formulation of POW

Given two data sequences $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ and $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_M] \in \mathbb{R}^{d \times M}$ of possibly different lengths ($N \neq M$), we view their elements as samples independently drawn from uniform distributions. Let $\mathbf{D} \in \mathbb{R}_+^{N \times M}$ denotes the pairwise ground distance matrix with each entry $d_{i,j}$ represents a geometric distance between \mathbf{x}_i and \mathbf{y}_j . We first consider the following regularized OT problem to find the optimal alignment between the two sequences:

$$T^* = \underset{T \in \Pi(\mathbf{u}_N, \mathbf{u}_M)}{\operatorname{argmin}} \langle \mathbf{D}, T \rangle_F + \lambda \sum_{i=1}^N \sum_{j=1}^M \left(\frac{i}{N} - \frac{j}{M} \right)^2 t_{i,j}. \quad (11)$$

Here, $\left(\frac{i}{N} - \frac{j}{M} \right)^2$ can be considered as the weight assigned to entry $t_{i,j}$ of the transport matrix, and λ is a positive parameter. We can easily observe that the weight is small if the relative positions $\frac{i}{N}$ and $\frac{j}{M}$ are similar, and large otherwise. Therefore, by minimizing the additional regularization term along with the objective of the original OT problem, we implicitly encourage $t_{i,j}$ to increase if \mathbf{x}_i and \mathbf{y}_j are relatively close to each other and decrease if they are far apart. We found that our regularization is simpler and more efficient than those proposed in [64]. First, it involves fewer parameters, thus alleviating the parameter-tuning burden. Second, without including any negative term in the objective function, solving (11) using matrix scaling algorithms is more stable than that in OPW.

With the additional regularization term, the OT-based problem in (11) can encode the sequential information among elements into the optimal transport matrix. However, it is still susceptible to outliers. More specifically, all the mass is enforced to be transported from one sequence to the other in formulation (11). When the sequences contain outliers, the transportation of their corresponding mass severely deteriorates the quality of the solution. Thus, motivated by the idea of partial optimal transport [13, 29], we propose

to limit the amount of transported mass in (11). We first introduce s such that $0 < s \leq 1$ as a fraction of mass to be transported. Then the feasible set of the transport matrix changes into

$$\Pi_s(\mathbf{u}_N, \mathbf{u}_M) = \{T \in \mathbb{R}_+^{N \times M} | T\mathbf{1}_M \leq \mathbf{u}_N, T^\top \mathbf{1}_N \leq \mathbf{u}_M, \mathbf{1}_N^\top T \mathbf{1}_M = s\}. \quad (12)$$

Now, we can derive the definition of the *Partial Ordered Wasserstein* (POW) distance as follows

$$\begin{aligned} POW(\mathbf{X}, \mathbf{Y}) &= \langle \mathbf{D}, T_{POW}^* \rangle_F \\ \text{s.t. } T_{POW}^* &= \underset{T \in \Pi_s(\mathbf{u}_N, \mathbf{u}_M)}{\operatorname{argmin}} \langle \mathbf{D}, T \rangle_F \\ &\quad + \lambda \sum_{i=1}^N \sum_{j=1}^M \left(\frac{i}{N} - \frac{j}{M} \right)^2 t_{i,j}. \end{aligned} \quad (13)$$

It is evident that the robustness of the POW distance heavily depends on the selection of the fraction s . The optimal value should not be too high, as it may include outliers in the transportation. Conversely, the fraction s should not be too small, as it may prohibit the masses corresponding to normal elements from being transported. Towards the end of this section, we will provide an analysis of the properties of POW. Based on this analysis, an effective procedure is introduced for automatically and adaptively selecting the value of s .

4.2. Optimization

Objective of the problem (14) can be rewritten as follows:

$$\begin{aligned} &\sum_{i=1}^N \sum_{j=1}^M d_{i,j} t_{i,j} + \lambda \sum_{i=1}^N \sum_{j=1}^M \left(\frac{i}{N} - \frac{j}{M} \right)^2 t_{i,j} \\ &= \sum_{i=1}^N \sum_{j=1}^M \left[d_{i,j} + \lambda \left(\frac{i}{N} - \frac{j}{M} \right)^2 \right] t_{i,j}. \end{aligned} \quad (15)$$

Thus, the problem (14) is equivalent to

$$T_{POW}^* = \underset{T \in \Pi_s(\mathbf{u}_N, \mathbf{u}_M)}{\operatorname{argmin}} \langle \hat{\mathbf{D}}, T \rangle_F, \quad (16)$$

where $\hat{\mathbf{D}}$ can be considered as modified ground distance matrix of size N by M with each entry

$$\hat{d}_{i,j} = d_{i,j} + \lambda \left(\frac{i}{N} - \frac{j}{M} \right)^2. \quad (17)$$

Problem (16) can be solved by adding dummy points (according to [18]) to expand the cost matrix

$$\tilde{\mathbf{D}} = \begin{bmatrix} \hat{\mathbf{D}} & \mathbf{0}_N \\ \mathbf{0}_M^\top & A \end{bmatrix}, \quad (18)$$

where $A > 0^1$. Solving the problem (16) now turns into solving the following problem:

$$\tilde{T}_{POW}^* = \underset{T \in \Pi(\tilde{\alpha}, \tilde{\beta})}{\operatorname{argmin}} \langle \tilde{\mathbf{D}}, T \rangle_F, \quad (19)$$

¹In this paper, we set $A = 1$ for all the experiments.

Algorithm 1 Calculate POW

Require: Two sequences $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M] \in \mathbb{R}^{d \times M}$, transported mass $s \in (0, 1]$, penalty parameter $\lambda \in \mathbb{R}_+$.

Ensure: $POW(\mathbf{X}, \mathbf{Y})$.

- 1: Compute the ground distance matrix \mathbf{D} ;
- 2: Compute matrix $\tilde{\mathbf{D}}$ using Equation (17);
- 3: Compute extended matrix $\tilde{\mathbf{D}}$ using Equation (18);
- 4: Initialize:

$$\tilde{\alpha} = \left[\frac{1}{N}, \dots, \frac{1}{N}, 1-s \right]^\top, \tilde{\beta} = \left[\frac{1}{M}, \dots, \frac{1}{M}, 1-s \right]^\top;$$

- 5: Use entropic regularization method [19] to approximate:

$$\tilde{\mathbf{T}}_{POW}^* = \underset{T \in \Pi(\tilde{\alpha}, \tilde{\beta})}{\operatorname{argmin}} \langle \tilde{\mathbf{D}}, T \rangle_F.$$

- 6: Remove the last row and column of $\tilde{\mathbf{T}}_{POW}^*$ to obtain \mathbf{T}_{POW}^* ;
- 7: Compute $POW(\mathbf{X}, \mathbf{Y}) = \langle \mathbf{D}, \mathbf{T}_{POW}^* \rangle_F$.

where $\tilde{\alpha} = [\mathbf{u}_N, 1-s]$, $\tilde{\beta} = [\mathbf{u}_M, 1-s]$, and $\tilde{\mathbf{T}}_{POW}^*$ denotes the optimal transport matrix with the size of $(N+1)$ -by- $(M+1)$. This is an original OT problem and we can approximate its optimal solution using entropic regularization method [19]. Then, the optimal solution of the problem (16) can be obtained by removing the last row and column of the optimal solution of the problem (19).

4.3. Analysis

Complexity. We first analyze the complexity of our method. In Algorithm 1, it is clear that steps 1, 2, and 3 involve calculating and extending the ground distance matrix of size $N \times M$. Without loss of generality, we assume that $N \geq M$. Thus, these steps have a computational complexity of $O(N^2)$. Step 7 has a similar complexity because it relates to the calculation of the Frobenius product between two matrices of the same size, N -by- M . While steps 4 and 6 can be computed in linear time, the most computationally intensive part of our method is induced by step 5, which requires solving an OT problem. Fortunately, the entropic regularization strategy in [19] allows us to approximate the optimal solution of this problem via the matrix scaling algorithm in quadratic time [4, 51]. Therefore, the total computational complexity of the proposed method is $O(N^2)$.

Regarding memory usage, our algorithm stores several matrices, including the ground distance matrix, the transport matrix, and their extended versions. The sizes of these matrices are at most $(N+1)$ -by- $(M+1)$. Thus, the total space requirement of our algorithm is also $O(N^2)$.

Numerical stability. We then investigate the numerical stability of the algorithm for calculating the POW distance. Similar to OPW, our method also employs matrix scaling algorithm (in step 5 of Algorithm 1). The following two steps: $\tilde{\kappa}_2 \leftarrow \tilde{\beta} / \tilde{\mathbf{K}} \tilde{\kappa}_1$ and $\tilde{\kappa}_1 \leftarrow \tilde{\alpha} / \tilde{\mathbf{K}} \tilde{\kappa}_2$, where $\tilde{\mathbf{K}} = e^{-\tau \tilde{\mathbf{D}}}$ and

τ is a positive parameter for the entropic regularization, are accordingly alternated until convergence. However, different from OPW, where the matrix scaling algorithm is unstable due to exponential operators on positive operands as discussed in section 3, we found that such a problem does not occur in our method. More specifically, our regularization guarantees that entries of the matrix $\tilde{\mathbf{D}}$ computed in (17) and (18) are always positive. Therefore, computing $\tilde{\mathbf{K}} = e^{-\tau \tilde{\mathbf{D}}}$ via the entry-wise exponential operator with a positive parameter τ limits the values of its entries within the range $[0, 1]$. In addition, we observe that entries of both $\tilde{\beta}$ and $\tilde{\alpha}$ are also in the range $[0, 1]$. Thus, $\tilde{\kappa}_1$ and $\tilde{\kappa}_2$ are always bounded as the algorithm alternates between the two steps mentioned above. This is in contrast with the algorithm for calculating OPW, where the matrix \mathbf{K} is initialized without an upper bound. As a result, when alternating between equations (7) and (8), values of entries of κ_1 and κ_2 can easily exceed the machine precision limit, making the algorithm prone to instability. This phenomenon has also been acknowledged in the original paper [64].

Outlier robustness. We finally study the robustness of our distance against outliers. The parameter s —the fraction of transported mass—plays a vital role in reducing the impact of outliers on POW. Thus, for a given pair of sequences \mathbf{X} and \mathbf{Y} of lengths N and M , respectively, we denote $POW_{\mathbf{X}, \mathbf{Y}}(s)$ as a function representing the dependence of the distance on the transported mass s . Let \mathbf{Y} be the normal sequence, while \mathbf{X} contains both normal elements and outliers. We denote \mathcal{I}^+ and \mathcal{I}^- as the sets of indices for normal elements and outliers, respectively. If $N^+ = |\mathcal{I}^+|$ and $N^- = |\mathcal{I}^-|$ denote the cardinalities of these two sets, then we have $N = N^+ + N^-$, where N is the length of the sequence \mathbf{X} . We assume that outliers in \mathbf{X} are separated from the elements in \mathbf{Y} as follows:

Assumption 1. For a constant $C > 1$ and d_{\max} being the largest ground distance between normal elements of the two sequences, then every outliers in \mathbf{X} is at a distance of at least $C(d_{\max} + \lambda)$ from any element in \mathbf{Y} .

Under above assumption, we can establish the following:

Lemma 1. Let $s^* = \frac{N^+}{N}$ and

$$\mathbf{T}_{s^*}^* = \underset{T \in \Pi_{s^*}(\mathbf{u}_N, \mathbf{u}_M)}{\operatorname{argmin}} \langle \mathbf{D}, T \rangle_F + \lambda \sum_{i=1}^N \sum_{j=1}^M \left(\frac{i}{N} - \frac{j}{M} \right)^2 t_{i,j}. \quad (20)$$

Then, given Assumption 1, $\mathbf{T}_{s^*}^*$ has no entry $t_{i,j}^* > 0$ such that $i \in \mathcal{I}^-$.

The proof of Lemma 1 is provided in Appendix A. This lemma implies that, with a proper selection of the transported mass $s = s^*$, all the masses from outliers will be excluded from transportation. As a consequence,

$$POW_{\mathbf{X}, \mathbf{Y}}(s^*) = \langle \mathbf{D}, \mathbf{T}_{s^*}^* \rangle_F \quad (21)$$

is robust against outliers. We further show the following:

Algorithm 2 Approximating the best value for the transported mass

Require: Two sequences X and Y , a positive penalty parameter λ , and a width ω .

Ensure: Approximation of the best s^* .

```

1:  $n \leftarrow \frac{1}{\omega}$ 
2:  $s_0 \leftarrow 0$ 
   /* Generate a series of values  $\{s_0, s_1, s_2, \dots, s_n\}$  */
3: for  $i \leftarrow 1$  to  $n$  do
4:    $s_i \leftarrow s_{i-1} + \omega$ 
5: end for
6: Compute  $POW_{X,Y}(s_i) \forall i$  in parallel using Algorithm 1
   /* Compute the second derivative  $\{\delta_0, \delta_1, \delta_2, \dots, \delta_n\}$  */
7: for  $i \leftarrow 0$  to  $n - 2$  do
8:    $\delta_i = POW_{X,Y}(s_{i+2}) - 2 \times POW_{X,Y}(s_{i+1}) + POW_{X,Y}(s_i)$ 
9: end for
10: Smooth the second derivative using the moving average method.
11: Detect the highest peak  $\delta_{i^*}$  in the second derivative.
12: Approximate  $s^* \approx s_{i^*}$ 
    
```

Lemma 2. For a small constant $\epsilon > 0$, given Assumption 1, then the first derivative values

$$\frac{\partial POW_{X,Y}(s^* - \epsilon)}{\partial s} \leq d_{\max} \quad \text{and} \quad \frac{\partial POW_{X,Y}(s^* + \epsilon)}{\partial s} \geq C(d_{\max} + \lambda). \quad (22)$$

The proof of this lemma is presented in Appendix B. Lemma 2 implies that the first derivative of the function $POW_{X,Y}(s)$ will exhibit a noticeable increase in the vicinity of s^* . By determining this abrupt change, we can obtain an approximation of the best value for s . To do so, we propose a procedure, the details of which are presented in Algorithm 2. It begins by generating a series of values $s_0, s_1, s_2, \dots, s_n$, where $s_0 = 0$, $s_n = 1$, and $s_{i+1} - s_i = \omega; \forall i$ with $\omega > 0$ being a small width. From these values, we compute a series of the corresponding $POW_{X,Y}(s_i); \forall i$. Note that $POW_{X,Y}(s_i)$ is independent of $POW_{X,Y}(s_j)$ for all $j \neq i$. Thus, the computation of the series can be carried out in parallel. Then, the second derivative at s_i , denoted by δ_i , can be calculated as $POW_{X,Y}(s_{i+2}) - 2POW_{X,Y}(s_{i+1}) + POW_{X,Y}(s_i)$. Repeating the calculation for all i , we obtain a series of discrete values of the second derivative $\delta_0, \delta_1, \delta_2, \dots, \delta_n$. After smoothing using a moving average over a sliding window, we detect the highest peak from the smoothed series of discrete values of the second derivative. This peak indicates an abrupt change in the first derivative, hence providing an approximation of the best value for s .

It is worth noting that the approximation accuracy of Algorithm 2 mostly depends on the parameter ω . Smaller the width is, better accuracy can be achieved. However, decreasing the width will increase the running time. Fortunately, the computational time of Algorithm 2 is linear in $n = \frac{1}{\omega}$. Thus, in this paper, we set $\omega = 0.001$ for all the experiments. We note that, in line 6 of the Algorithm 2, the Algorithm

1 is used n times to compute $POW_{X,Y}(s_i), \forall i$. However, as mentioned above, the calculation of $POW_{X,Y}(s_i)$ for each s_i is independent. Thus, we can carry out the computation in parallel, and the time complexity of Algorithm 2 remains quadratic w.r.t the length of the input sequences.

5. Applications

In this section, we study applications of the proposed distance in two tasks, including time-series classification and multi-step localization. For each task, we will state the problem, briefly review several existing baselines, and describe how to apply POW to solve the problem.

5.1. Time-series classification

Time-series classification (TSC) is one of the most important tasks in time series analysis due to its widespread applications, such as human action recognition [45], fault detection [59], and electrocardiogram diagnosis [27], among others. The goal of TSC is to predict the class label for a given unlabeled time series input as accurately as possible. To achieve this goal, many TSC methods have been proposed over the past few years [1, 31, 69]. These methods can be divided into three groups: *i) model-based methods*, which assume that time series from the same class are generated by a model and classify any unlabeled time series to the class of the closest-fitting model, *ii) featured-based methods*, which extract features from time series and classify them based on the extracted features, and *iii) distance-based methods*, which classify time series into the nearest class in term of a predefined distance measure. Methods from the first group would provide very accurate classification results if the assumptions on the underlying generative models were appropriate for the datasets. However, in practice, deriving the relevant assumptions is often a challenging task. The second group can provide classification results with interpretability through feature extraction. However, interpretability comes at the cost of high complexity due to the time-consuming feature extraction process.

In this paper, with the introduction of the POW distance, we adopt an approach from the third group: distance-based TSC methods. There are three main distance-based TSC approaches. The first approach is to directly exploit the predefined distance measures within the k nearest neighbor (k-NN) classifier. The second approach is to employ the distances to obtain a time series kernel and enable the use of kernel methods. Recently, a third approach emerged, that exploits distances for sequential data within deep neural network (DNN) models. For instance, [14] introduces a neural network layer employing DTW to align the inputs and the outputs. Slightly different from that, [38] utilizes DTW to align inputs and the layer weights. [11] substitutes the dot product in the first layer of convolutional neural network (CNN) with DTW, forming a deep classifier termed as dynamic CNN. [47] approximates DTW by CNN and uses the approximation to classify EEG data. In OTW [46], OT is integrated into DNNs following a similar approach to [14].

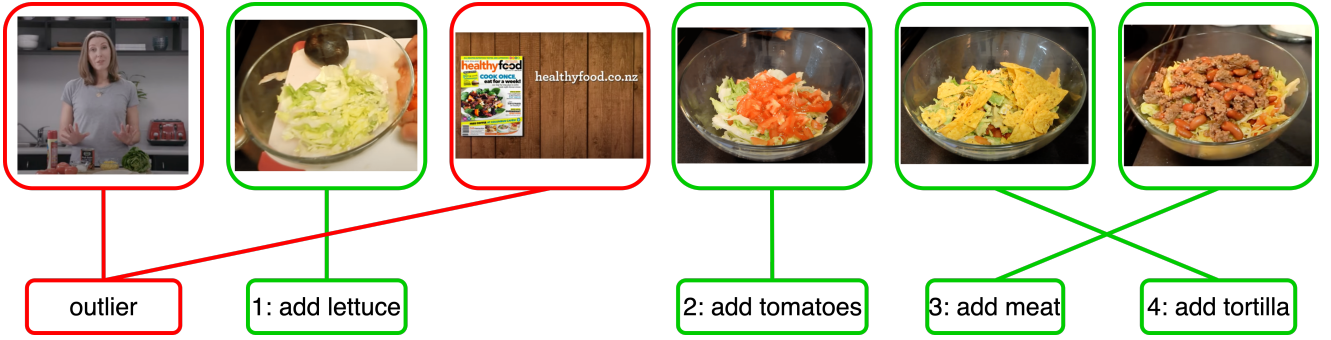


Figure 2: Multi-step localization illustration. Given an instructional video (e.g., making taco salad) and a list of steps in text (e.g., “add onion”, “add lettuce”, “add tortilla”, “add meat”), the task of multi-step localization is to align each video frame to the correct step.

We recommend [10, 1, 37, 61] for more detailed surveys of such approaches.

To evaluate the proposed distance and compare it with the existing ones, we choose the first approach. Because POW is directly used by the k-NN classifier, the classification results reflect more comprehensively the quality of the distance. As a result, this first approach is still widely used in recent works for comparing distance performances [10, 61, 28]. On the other hand, in the second approach, the distance implicitly contributes to the classification results via the kernel. In addition, k-NN is simple and cannot perform complex feature extraction as DNNs do. Thus, the performance of k-NN in time series classification is much more dependent on the distance used than those of the DNN-based approaches.

It is worth noting that, despite the simplicity of k-NN, it is still a popular choice for time series classification. Experimental results from previous studies [73, 52, 7] have shown that the combination of k-NN with DTW reaches substantial performance and seems to be particularly difficult to significantly outperform. To verify the performance of the proposed distance combined with k-NN classifiers, in section 6, we utilize a portion of the UCR time series classification archive [21], which contains more than 120 datasets originally collected from various domains, such as healthcare, robotics, and transportation. Please note that the UCR archive provides only univariate time series data. Therefore, we also apply our method to the Weizmann dataset [30], which is used for human action recognition. Given labeled videos of different activities, the task is to predict which activity is performed in an unlabeled video. All the videos are pre-processed to get the form of multivariate sequences. Further details are provided in section 6.

5.2. Multi-step localization

In this work, we harness the alignment capability of POW to address the multi-step localization task, which involves inferring the temporal location of procedure steps within instructional videos. Specifically, during the inference phase, we aim to match each frame, represented by embeddings, with its corresponding step. The multi-step

localization task presents several challenges. Firstly, instructional videos often contain many frames unrelated to the primary activity, such as scenes featuring people talking, occasional dishwashing, or advertisements. These frames are considered outliers, and attempting to match them to the steps would result in meaningless alignments. Secondly, the activities performed in instructional videos may not strictly follow the order of the given steps. For example, when making taco salad, one person may add tortilla before adding meat, while another may follow a reverse order. Traditional alignment methods, such as DTW and its variants, tend to induce false correspondences because they strictly preserve the order when aligning video frames and steps. Finally, instructional videos often exhibit a segment structure where many successive frames correspond to the same steps. Failing to account for this characteristic in the alignment process can lead to suboptimal results. A visual representation of the task is given in Figure 2.

Existing methods for multi-step localization can be divided into three groups, including *fully supervised methods*, where the start and end times of each step in the video are required [12, 53, 79, 67], *weakly supervised methods*, which rely solely on knowledge of the steps present in the video [36, 60, 23, 16, 80, 17, 24, 25], and *unsupervised methods*, which processes text narrations obtained from automated speech recognition (ASR) in conjunction with the videos [63, 44, 26]. While the first group seems less ideal because it requires laborious efforts for labeling the endpoints in the videos, methods in the third group are often overly complex as they have to process the text that comes from ASR on the audio, which is noisy and error-prone. Our method falls into the second group as it only requires information about the steps in the form of a set. However, existing methods in this group typically assume that the activity in the video follows the same order as the given corresponding set of steps. This assumption contradicts the fact mentioned earlier that local reverses often occur. In contrast to existing methods, our approach can handle videos in which people perform the same activity in different orders, thanks to the flexible alignment capability of POW. It is worth noting that recent work in [25] relaxes the order requirement by creating flow graphs

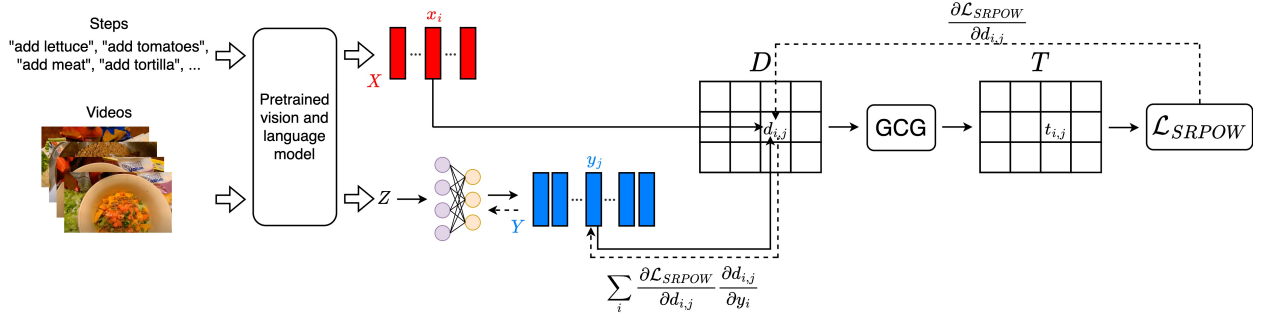


Figure 3: Training phase of the proposed method for multi-step localization. The solid arrows show the forward computation and the dashed arrows indicate gradient backpropagation.

that capture all possible orders among steps. However, these flow graphs are constructed assuming that each step occurs only once in each video, which is often violated in reality. For example, several videos demonstrating the preparation of Kerala Fish Curry repeat steps such as "stir mixture" multiple times, once after adding chili powder and again after adding fish.

While POW can provide outlier-robust and flexible alignment, it lacks a mechanism to effectively handle segment structures in instructional videos. As a solution, we introduce of a segment regularization and integrate it with POW, creating a new variant termed *segment-regularized POW* (SRPOW). Specifically, let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{d \times M}$ and $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_M] \in \mathbb{R}^{d \times M}$ represent the step and frame-wise video embeddings, respectively. The transport matrix $\mathbf{T} \in \mathbb{R}^{N \times M}$, where each element $t_{i,j}$ indicates the degree of correspondence between the i^{th} step and the j^{th} frame, encodes the alignment between the video and the steps. Given that many successive frames are allocated to the same steps, we can expect \mathbf{T} to exhibit smoothness with respect to its columns. The smoothness of the transport matrix can be represented by the following term.

$$\|\mathbf{S}\mathbf{T}^\top\|_F^2, \quad (23)$$

where $\mathbf{S} \in \mathbb{R}^{(M-2) \times M}$ is the second-order difference matrix

$$\mathbf{S} = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 & 1 \end{bmatrix}. \quad (24)$$

Then, SRPOW is defined as

$$SRPOW(\mathbf{X}, \mathbf{Y}) = \langle \mathbf{D}, \mathbf{T}_{SRPOW}^* \rangle_F \quad (25)$$

$$\begin{aligned} \text{s.t. } \mathbf{T}_{SRPOW}^* &= \underset{\mathbf{T} \in \Pi_s(\mathbf{u}_N, \mathbf{u}_M)}{\operatorname{argmin}} \langle \mathbf{D}, \mathbf{T} \rangle_F \\ &+ \lambda \sum_{i=1}^N \sum_{j=1}^M \left(\frac{i}{N} - \frac{j}{M} \right)^2 t_{i,j} \\ &+ \gamma \|\mathbf{S}\mathbf{T}^\top\|_F^2, \end{aligned} \quad (26)$$

where γ is a positive penalty parameter. This penalization explicitly promotes consistency within local regions and discourages abrupt transitions between columns of the optimal transport matrix. However, with the introduction of this new regularization, solving problem (26) becomes challenging because the matrix scaling algorithm is not directly applicable. Fortunately, we observe that the regularizer specified in equation (23) is continuous, which ensures the continuity of the objective function as well. Furthermore, given that the constraint set $\Pi_s(\mathbf{u}_N, \mathbf{u}_M)$ is a convex, closed, and bounded subset of $\mathbb{R}^{N \times M}$, the objective function reaches its minimum on $\Pi_s(\mathbf{u}_N, \mathbf{u}_M)$. If the regularizer is strictly convex, this minimum is unique, which applies to our segment regularization as mentioned in equation (23). Therefore, we employ the generalized conditional gradient (GCG) algorithm [9] to solve problem (26). This algorithm operates through an iterative process that ensures every iteration falls within $\Pi_s(\mathbf{u}_N, \mathbf{u}_M)$. Consequently, the convergence of GCG when solving our problem is guaranteed. Details of the algorithm are given in Appendix C.

We utilize SRPOW in both training and inference phases. At the training phase, given pairs of set of steps in text and a video, we feed them to a pretrained vision and language model that was obtained from training on a large instructional video dataset [55]. The outputs of the model for each pair are step embeddings $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ and visual embeddings $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_N] \in \mathbb{R}^{d \times N}$. Subsequently, we feed the obtained visual embeddings through a two-layer fully connected neural network, resulting in the framewise video embeddings $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_M] \in \mathbb{R}^{d \times M}$. The network is trained by minimizing the loss $\mathcal{L}_{SRPOW} = SRPOW(\mathbf{X}, \mathbf{Y})$. Fig. 3 illustrates the training phase of the proposed method. At the inference phase, we feed the unlabeled video and its corresponding steps into the trained model to obtain the framewise video embeddings and step embeddings. We then solve the SRPOW problem (26) between these two sets of embeddings to compute the optimal transport matrix \mathbf{T}^* . For the j^{th} column of \mathbf{T}^* , we find the index i such that $i = \operatorname{argmax}_{1 \leq k \leq N} t_{k,j}^*$. Subsequently, we align the j^{th} frame with the i^{th} step.

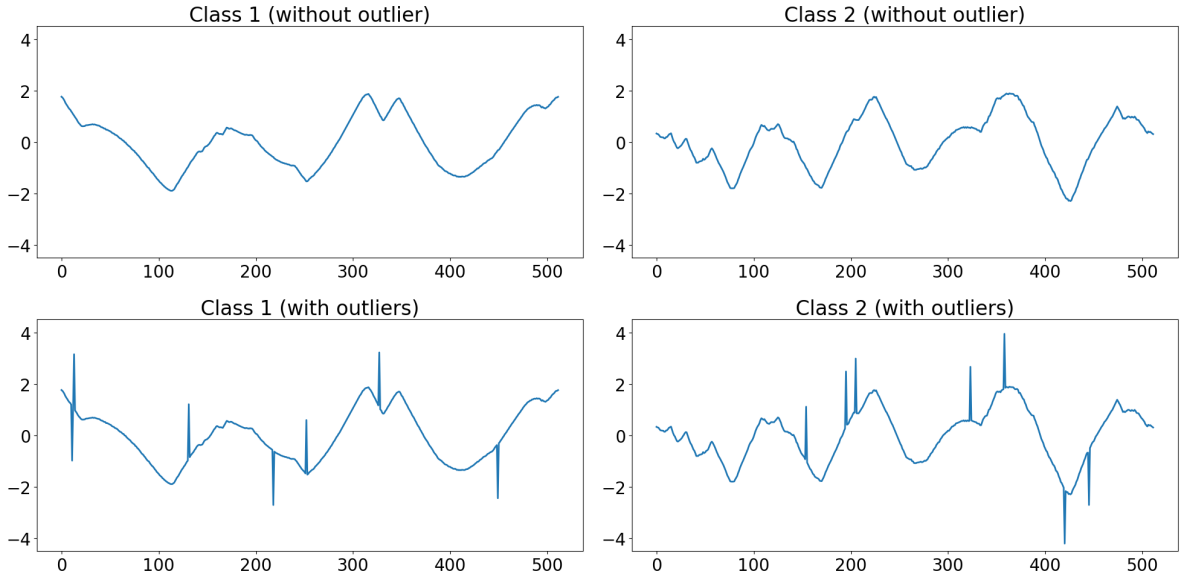


Figure 4: Samples of Time series in BirdChicken dataset from The UCR archive. The top subfigures show original samples, one for each class, from BirdChicken dataset. The bottom subfigures show these samples after adding outliers

We have provided detailed discussions on time-series classification and multi-step localization tasks as applications of our proposed distance. However, it is important to note that POW has broader potential applications due to its ability to align and calculate differences between two sequences. For example, POW can be applied in time series clustering tasks [57, 2, 33], where time series are grouped based on their similarity without requiring supervised information. Additionally, POW can also be utilized in video and text retrieval tasks [24, 44], where the goal is to find corresponding clips or captions given a query caption or clip. These examples illustrate the versatility and potential of POW beyond its initial applications, opening avenues for future exploration.

6. Experiments

In this section, we conduct experiments to evaluate the performance of the proposed distances in two tasks: time-series classification and multi-step localization. For each task, we will provide a brief description of the preprocessing steps applied to the datasets used in the experiments. We will then mention the compared distance measures, followed by a description of parameter tuning and the definitions of the evaluation metrics. Finally, we will present the results and provide a discussion on the performances of all the distance measures.

6.1. Time-series classification

Datasets. We selected 20 datasets from the UCR time series archive [21]. Each of these datasets includes univariate time series of equal length from different classes and was already divided into training and test sets. To make the experiments more challenging, we added outliers to the time series. Specifically, for each dataset, we randomly sampled

a proportion p from a uniform distribution $\mathbf{u}_{(0.1,0.5)}$. Let N denote the length of the time series in the considered dataset. We then randomly added or subtracted $p \times N$ elements at randomly selected positions in each time-series sequence, up to its maximum value.

Figure 4 (bottom) shows the time series of the BirdChicken dataset from The UCR archive after adding outliers. It can be observed that the outliers distort the shape of the time series and pose significant challenges to the distance measurement process. We repeated this procedure five times. Consequently, from each dataset selected from the UCR time series archive, we generated five datasets contaminated by outliers with proportions within the range $[0.1, 0.5]$. Each generated dataset was further divided into training and test sets with a consistent ratio of 60/40.

We also aimed to investigate the performance of the proposed methods on multivariate time series. For this purpose, we utilized the Weizmann dataset [30], which comprises 90 videos featuring nine subjects, each performing ten natural actions: bend, jack, jump-forward (jump), jump-in-place (pjump), run, side, skip, walk, wave-one-hand (wave 1), and wave-two-hand (wave 2). Figure 5(a) depicts the ten action classes in the Weizmann dataset.

Initially, we subtracted the background from each frame of these video sequences and rescaled them to a size of 70×35 . For each 70×35 rescaled frame, we computed binary features, as illustrated in Fig. 5(b). To reduce the dimensionality of the feature space (from 2450), we retained the top 123 principal components, preserving 99% of the total energy, for our experiments. Similar to the univariate time-series datasets, to make the experiments more challenging, we randomly sampled a proportion p from a uniform distribution $\mathbf{u}_{(0.1,0.5)}$. Subsequently, $p \times N$ random 123-dimensional binary vectors were generated, where N denotes the length of the video. These vectors replaced the



Figure 5: Weizmann action dataset. The dataset consists of (a) Ten action classes performed by different subjects and (b) Binary features, which are extracted from videos, forming multivariate sequences.

original vectors at randomly selected positions within the video sequences. This procedure was repeated five times, and each resulting dataset was further divided into training and test sets with a consistent ratio of 60/40.

Compared distance measures. For classification task, we utilize k -nearest neighbor (k-NN) classifier equipped different distances, including POW and the followings:

- Dynamic time warping (DTW) [62] – a widely used distance measure for time-series data,
- Soft DTW (Soft-DTW) [20] – a smooth variant of DTW,
- Shape segment DTW (SSDTW) [34], which represents local structure of the time series via segments before calculating the distance,
- Gaussian weighted DTW (GW-DTW) [49] – a weighted variant of DTW, where the weights are normally distributed.
- Drop DTW (Drop-DTW) [24], which introduces additional costs to DTW for dropping outliers,
- Order-preserving Wasserstein (OPW) [64] – an OT-based approach for calculating distance between sequences,
- Wasserstein subsequence kernel (WSK), which applies OT framework on subsequences instead of elements of the sequences as in OPW, and

Dataset	λ	Dataset	λ
BME	50	GunPoint	10
BeetleFly	10	Herring	50
BirdChicken	1	ItalyPowerDemand	10
Chinatown	10	MoteStrain	50
Coffee	5	OliveOil	10
DistalPhalanxOutlineCorrect	1	Plane	1
DistalPhalanxTW	5	SmoothSubspace	5
ECG200	50	SonyAIBORobotSurface1	5
FaceFour	5	SonyAIBORobotSurface2	10
Fungi	10	ToeSegmentation2	100

Table 1

The optimal λ of POW distance on different datasets that are contaminated by outliers in UCR Archive for the 1-NN classifier.

- Optimal transport warping (OTW) [46] – an unbalanced OT-based approach, which considers only element-wise difference in terms of position for fast computation.

Evaluation methods. We use classification accuracy, which is the ratio of the number of correct predictions to the total number of predictions to evaluate performance of the k-NN classifier equipped with different distance measures. In addition, to make our evaluation more reliable, we conducted the McNemar's test [54] between the proposed distance and each its competitors at the significance level of 5%.

Parameter tuning. We note that the performance of the k-NN classifier also depends on the number of neighbors, denoted as k , in addition to the equipped distances. Thus, in

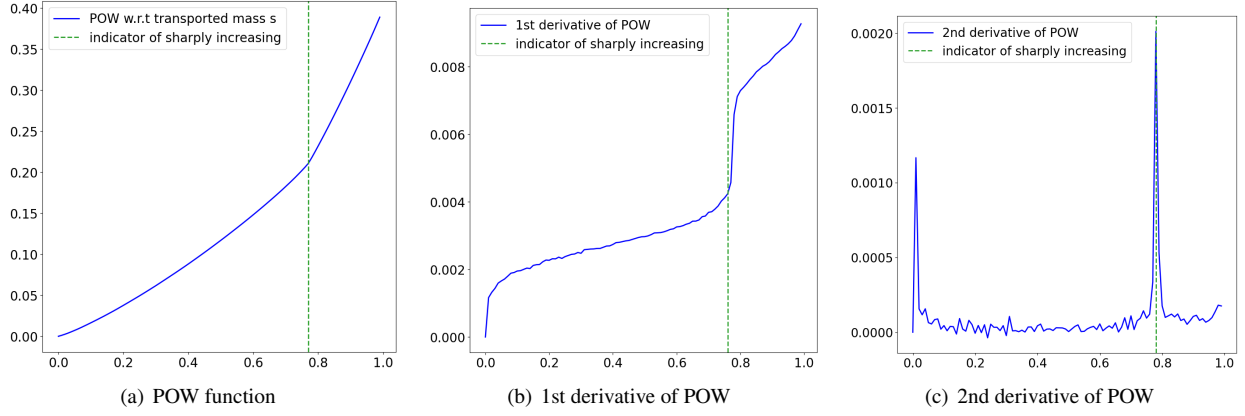


Figure 6: An example of automatically selecting transported mass s for POW. (a) Given two time series from the UCR archive with outlier fraction of 0.23, the POW distance function with respect to s is calculated. (b) The first derivative of the POW function shows a sharp increase, which is the indicator for selecting s . (c) Position of the indicator can be located by detecting the highest peak in the second derivative of the POW function.

the experiments, we ran all the above classifiers with k selected from the set $\{1, 3, 5, 7, 15, 30\}$. We used the Euclidean distance measure as the ground distance between elements of the two sequences for all the distance measures. For the POW distance, we applied algorithm 2 to automatically approximate the best transported mass s for all the datasets. The remaining parameter λ was selected using grid search. The best values of λ , which returned the highest classification accuracy on the univariate time-series datasets, are summarized in Table 1. For Soft-DTW, we selected its smoothness parameter α from $\{10, 1, 0.1, 0.01, 0.001\}$ and found that it works reasonably well at $\alpha = 1$, which empirically agrees with [20]. OPW has three parameters, including λ_1 and λ_2 , which correspond to its regularization terms, and σ , which is required for computing the prior distribution of the transport matrix, respectively. We again selected them using grid search. During the process of selecting parameters for OPW, we found that it is unstable with relatively large $\lambda_1 (> 50)$, which is coherent with the numerical stability analysis in section 4.3. Therefore, we set $\lambda_1 = 50$. For most of the datasets, OPW showed optimal performance at $\lambda_2 = 0.1$ and $\sigma = 1$. For the remaining distances, their parameters were set according to the original papers.

Results and discussion. The outlier robustness of the proposed distance is proven to be heavily dependent on the selection of the parameter s , which represents the transported mass, as demonstrated in Lemma 1. Manually selecting proper values of s for all the datasets can be a tedious task. Fortunately, Algorithm 2 helps us automatically and adaptively approximate the best values for s . Figure 6(a) depicts the POW distance function with respect to s between two time series in the BirdChicken dataset, which has an outlier fraction of $p = 0.23$. Since the masses for all elements of the time series are equal and sum up to 1, the best value for the transported mass is $s^* = 0.77$. According to Lemma 2, we expect the first derivative of POW to increase sharply at $s = 0.77$. This is validated in Figure 6(b). To detect

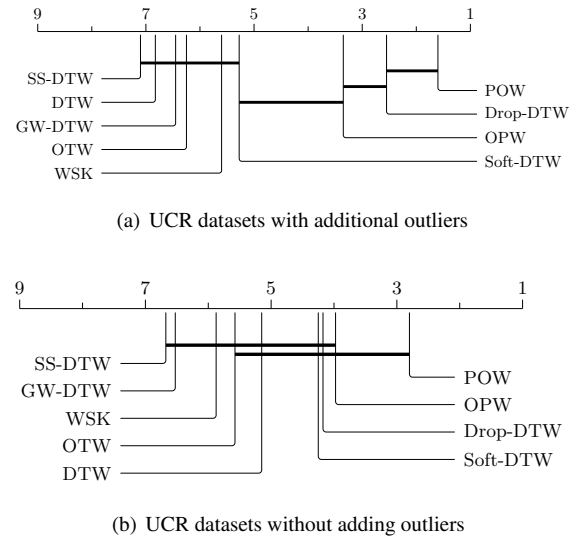


Figure 7: Critical difference diagrams showing the mean ranks, which are obtained from based on the Wilcoxon signed-rank test, of k -NN with different distance measures on 20 UCR datasets (a) with and (b) without adding outliers. The lower the rank (further to the right) the better distance measure compared to the others on average. The lines indicates that there is no significant difference in performance among the distances crossed by that particular line in terms of the Friedman test that compares ranks of multiple distances [22].

this value, Algorithm 2 examines the second derivative of POW, searching for the highest peak, which indicates the best approximation values for s . An illustration is provided in Figure 6(c).

With a reliable approximation of s , we then investigate the contribution of POW to the accuracy of the k -NN classifier. Table 2 shows the classification results of a k -NN classifier using various distance measures on 20 datasets taken from the UCR time series archive [21] and then

Partial Ordered Wasserstein Distance for Sequential Data

Dataset	Variants of DTW					OT-based distances			
	DTW	Soft-DTW	SS-DTW	GW-DTW	Drop-DTW	OPW	WSK	OTW	POW
BME	79.33 (1.18)*	85.73 (1.12)*	83.47 (1.32)*	82.99 (1.24)*	82.53 (1.59)*	84.27 (2.89)*	72.18 (1.75)*	80.14 (1.09)*	89.47 (0.73)
BeetleFly	67.00 (2.58)*	60.00 (3.54)*	63.23 (2.16)*	64.11 (2.83)*	68.46 (2.74)*	68.00 (2.55)*	65.12 (3.09)*	62.87 (3.15)*	80.40 (5.48)
BirdChicken	84.00 (2.11)†	76.00 (4.18)	65.60 (3.14)*	72.13 (2.51)*	79.00 (4.48)†	58.00 (4.57)*	62.03 (3.47)*	60.06 (4.25)*	75.28 (2.65)
Chinatown	93.44 (0.23)*	94.33 (0.70)*	92.38 (1.80)*	93.17 (2.11)*	95.98 (0.63)*	95.25 (0.88)*	94.17 (0.74)*	91.04 (1.26)*	97.98 (0.50)
Coffee	75.71 (2.82)*	75.00 (2.53)*	76.01 (1.92)*	77.24 (2.12)*	88.57 (2.99)	89.29 (3.57)	84.61 (2.68)*	83.19 (3.02)*	88.59 (3.42)
DistalPhalanxOutlineCorrect	71.01 (2.56)	70.51 (1.04)*	71.32 (1.10)	70.41 (1.45)*	73.62 (2.18)	69.78 (2.50)*	64.03 (2.87)*	65.56 (2.92)*	72.10 (1.43)
DistalPhalanxTW	57.70 (1.55)	58.99 (1.83)†	55.16 (1.38)*	54.25 (1.53)*	56.42 (2.41)	59.01 (3.81)†	52.48 (1.74)*	53.55 (2.26)*	56.69 (2.18)
ECG200	83.40 (1.26)*	85.80 (1.10)*	84.09 (1.04)*	84.73 (1.25)*	88.60 (0.89)*	88.30 (1.52)*	86.05 (1.19)*	85.11 (1.77)*	91.40 (1.52)
FaceFour	88.18 (0.62)*	91.36 (1.22)*	87.42 (0.98)*	86.15 (1.08)*	93.18 (0.80)	93.41 (0.51)	90.16 (0.95)*	91.41 (1.20)*	93.86 (0.62)
Fungi	61.29 (1.29)*	72.80 (1.24)*	70.20 (1.39)*	71.11 (1.54)*	83.33 (0.76)	82.05 (1.49)	74.52 (0.90)*	75.41 (0.87)*	83.02 (0.95)
GunPoint	89.73 (1.30)*	92.40 (0.76)	90.81 (1.31)*	91.13 (1.06)	93.73 (1.01)	92.00 (0.67)	91.02 (0.88)*	90.09 (1.03)*	92.87 (0.87)
Herring	48.13 (1.23)*	42.50 (1.31)*	46.05 (1.48)*	45.27 (1.28)*	51.88 (4.34)*	50.89 (6.50)*	48.98 (4.62)*	47.49 (4.13)*	53.13 (3.31)
ItalyPowerDemand	89.83 (0.91)*	92.06 (0.76)*	91.16 (3.87)*	91.42 (4.14)*	93.37 (1.86)	93.00 (1.39)	91.52 (1.82)*	90.86 (2.57)*	94.61 (1.30)
MoteStrain	75.89 (0.74)*	76.20 (1.23)*	74.08 (2.14)*	75.15 (2.51)*	85.83 (0.97)	84.20 (1.41)*	83.53 (1.25)*	82.25 (2.44)*	85.95 (0.63)
OliveOil	73.33 (2.22)*	76.67 (1.82)*	78.68 (2.00)*	79.12 (2.55)*	86.00 (1.49)	84.27 (2.39)	81.23 (1.83)*	80.11 (2.63)*	85.44 (1.61)
Plane	96.76 (0.52)*	97.80 (0.80)*	96.86 (1.12)*	97.16 (1.27)*	98.29 (1.43)	99.00 (0.84)	97.03 (1.22)*	96.88 (2.02)*	99.24 (0.93)
SmoothSubspace	82.00 (2.31)*	84.66 (2.54)*	83.23 (2.35)*	84.37 (2.62)*	85.73 (2.93)*	88.00 (2.26)*	87.53 (1.91)*	86.15 (2.01)*	91.60 (2.52)
SonyAIBORobotSurface1	67.55 (0.64)*	67.55 (0.86)*	66.20 (1.03)*	67.12 (1.33)*	79.60 (1.49)*	85.69 (0.96)*	82.27 (1.46)*	84.13 (1.55)*	90.03 (1.28)
SonyAIBORobotSurface2	81.68 (1.03)*	81.99 (0.98)*	80.02 (1.02)*	82.15 (0.88)*	85.86 (0.84)	80.82 (1.64)*	79.05 (1.53)*	81.18 (1.32)*	84.68 (1.10)
ToeSegmentation2	85.38 (1.15)*	87.33 (1.17)*	86.22 (2.10)*	86.00 (1.67)*	89.47 (1.85)	88.15 (2.01)*	87.64 (1.04)*	86.81 (1.88)*	90.46 (1.42)
Average	77.57 (12.59)	78.48 (13.88)	77.11 (13.21)	77.76 (13.18)	82.97 (12.28)	81.67 (13.43)	78.76 (13.80)	78.71 (13.65)	84.84 (12.31)

Table 2

The classification results of k-NN with different distance measures on UCR archive datasets with additional outliers. The classifiers were executed on each dataset five times with varying proportions of outliers, and the average accuracy in percentage, along with the standard deviations (in brackets), are reported. The highest accuracy for each dataset is highlighted in **bold**. The symbol * indicates that the corresponding distance measure performed worse than POW, while the symbol † indicates that the corresponding distance measure performed better than POW with significance at the 5% level by McNemar's test [22].

Dataset	Variants of DTW					OT-based distances			
	DTW	Soft-DTW	SS-DTW	GW-DTW	Drop-DTW	OPW	WSK	OTW	POW
BME	89.50 (1.20)†	97.20 (1.15)†	85.90 (1.30)†	85.10 (1.25)†	86.67 (1.55)†	83.33 (2.80)	76.00 (1.70)*	89.50 (1.10)†	83.67 (0.80)
BeetleFly	70.00 (2.50)	65.50 (3.40)*	63.50 (2.10)*	64.20 (2.80)*	65.80 (2.70)*	70.67 (2.50)	71.50 (3.00)†	67.00 (3.10)*	70.33 (3.30)
BirdChicken	75.50 (2.20)	75.50 (4.10)	75.80 (3.10)	76.50 (2.60)	69.50 (4.40)*	60.50 (4.50)*	69.50 (3.40)*	62.50 (4.20)*	76.50 (2.70)
Chinatown	97.50 (0.30)	97.50 (0.80)	92.50 (1.90)*	93.55 (2.20)*	97.67 (0.70)	74.93 (0.90)*	83.50 (0.80)*	95.50 (1.30)*	98.00 (0.60)
Coffee	99.80 (2.70)†	99.50 (2.50)†	96.20 (2.00)	92.50 (2.10)*	89.28 (3.00)*	96.42 (3.60)	88.80 (2.70)*	81.50 (3.00)*	96.80 (3.50)
DistalPhalanxOutlineCorrect	72.20 (2.50)*	68.70 (1.10)*	64.50 (1.20)*	70.70 (1.50)*	72.46 (2.20)*	75.36 (2.60)	69.20 (2.80)*	70.70 (2.90)*	74.43 (1.50)
DistalPhalanxTW	59.00 (1.80)*	67.00 (1.90)†	55.50 (1.40)*	54.50 (1.50)*	59.71 (2.50)*	63.31 (3.90)	61.50 (1.80)	54.50 (2.30)*	62.80 (2.20)
ECG200	77.50 (1.30)*	84.90 (1.20)*	84.20 (1.10)*	84.80 (1.30)*	92.08 (0.90)†	88.40 (1.50)	89.10 (1.20)	85.20 (1.80)*	89.50 (1.50)
FaceFour	83.20 (0.70)*	86.40 (1.20)*	87.50 (1.00)*	86.20 (1.10)*	88.63 (0.80)*	93.18 (0.50)	84.20 (1.00)*	86.50 (1.20)*	93.50 (0.70)
Fungi	80.50 (1.30)*	82.00 (1.20)*	80.50 (1.40)*	79.20 (1.50)*	84.40 (0.80)*	94.08 (1.50)	79.70 (0.90)*	82.50 (0.90)*	93.20 (1.00)
GunPoint	91.00 (1.20)*	98.00 (0.80)	90.48 (1.30)*	91.20 (1.00)*	94.00 (1.00)*	97.33 (0.70)	94.33 (0.90)*	86.20 (1.00)*	98.00 (0.90)
Herring	53.50 (1.30)*	63.80 (1.40)†	56.30 (1.50)*	55.50 (1.30)*	53.12 (4.40)*	59.37 (6.60)	52.67 (4.70)*	60.70 (4.20)	60.50 (3.40)
ItalyPowerDemand	95.00 (1.00)†	94.65 (0.80)†	91.30 (4.00)	91.50 (4.20)	95.53 (1.90)†	92.03 (1.40)	93.33 (1.90)	95.00 (2.60)†	92.00 (1.40)
MoteStrain	83.00 (0.80)*	88.89 (1.30)	84.20 (2.20)*	85.30 (2.60)*	86.50 (1.00)*	88.57 (1.40)	88.67 (1.30)	79.30 (2.50)*	89.33 (0.70)
OliveOil	83.50 (2.20)*	46.67 (1.80)*	78.80 (2.00)*	77.30 (2.60)*	86.67 (1.50)	86.67 (2.40)	74.50 (1.80)*	79.30 (2.70)*	86.67 (1.70)
Plane	99.00 (0.50)	99.00 (0.80)	97.67 (1.10)*	97.30 (1.30)*	96.19 (1.50)*	98.09 (0.90)	98.20 (1.30)	96.00 (2.00)*	99.00 (1.00)
SmoothSubspace	83.50 (2.30)*	74.00 (2.50)*	73.50 (2.40)*	74.50 (2.70)*	99.03 (3.00)	99.33 (2.30)	91.33 (1.90)*	93.50 (2.00)*	98.67 (2.50)
SonyAIBORobotSurface1	73.60 (0.70)*	62.22 (0.90)*	66.30 (1.00)*	67.20 (1.30)*	68.22 (1.50)*	78.36 (1.00)	68.40 (1.50)*	77.20 (1.60)	77.20 (1.30)
SonyAIBORobotSurface2	83.10 (1.00)*	83.73 (1.00)	82.10 (1.00)*	82.20 (0.90)*	87.62 (0.80)†	80.07 (1.60)*	80.67 (1.50)*	84.30 (1.30)	84.70 (1.10)
ToeSegmentation2	84.50 (1.20)*	88.46 (1.20)*	86.30 (2.20)*	86.20 (1.70)*	87.69 (1.90)*	90.20 (2.00)	87.70 (1.10)*	84.00 (1.90)*	90.20 (1.50)
Average	81.72 (12.00)	81.21 (14.76)	79.65 (12.37)	79.77 (11.94)	83.04 (13.05)	83.48 (12.41)	80.14 (11.69)	80.54 (11.82)	85.60 (11.56)

Table 3

The classification results of k-NN with different distance measures on UCR archive datasets without adding outliers. For each dataset, the original train and test sets were merged before being divided randomly into 5 equal folds. The classifiers were executed on each dataset using 5-fold cross-validation, and the average accuracies in percentage on all the folds, along with the standard deviations (in brackets), are reported. The highest accuracy for each dataset is presented in **bold**. The symbol * indicates that the corresponding distance measure performed worse than POW, while the symbol † indicates that the corresponding distance measure performed better than POW with significance at the 5% level by McNemar's test [22].

contaminated by outliers. This table also includes the results from McNemar's test [54] conducted between the proposed distance and each of its competitors. We can observe that Drop-DTW and POW obtained the highest classification accuracies on most of the datasets. From the results of McNemar's test, we can also confirm that they significantly outperformed most of the remaining distance measures. One prominent reason is that Drop-DTW and POW include mechanisms to exclude abnormal elements when aligning the two time series. Consequently, they exhibit robustness in the presence of outliers. Other distance measures, such as those based on shape description and alignment weighting,

can also mitigate the impact of outliers. However, these approaches appeared to be less effective than the outlier-removing mechanisms of Drop-DTW and POW. Next, we examine the performance differences between DTW variants and OT-based distances. The results presented in Table 2 indicate that, in most datasets, OT-based approaches outperformed the DTW variants. The best performance was achieved with POW. This validates the importance of alignment flexibility when calculating distances, as it helps reduce the effects of local distortion and provides better matching between elements of the two time series.

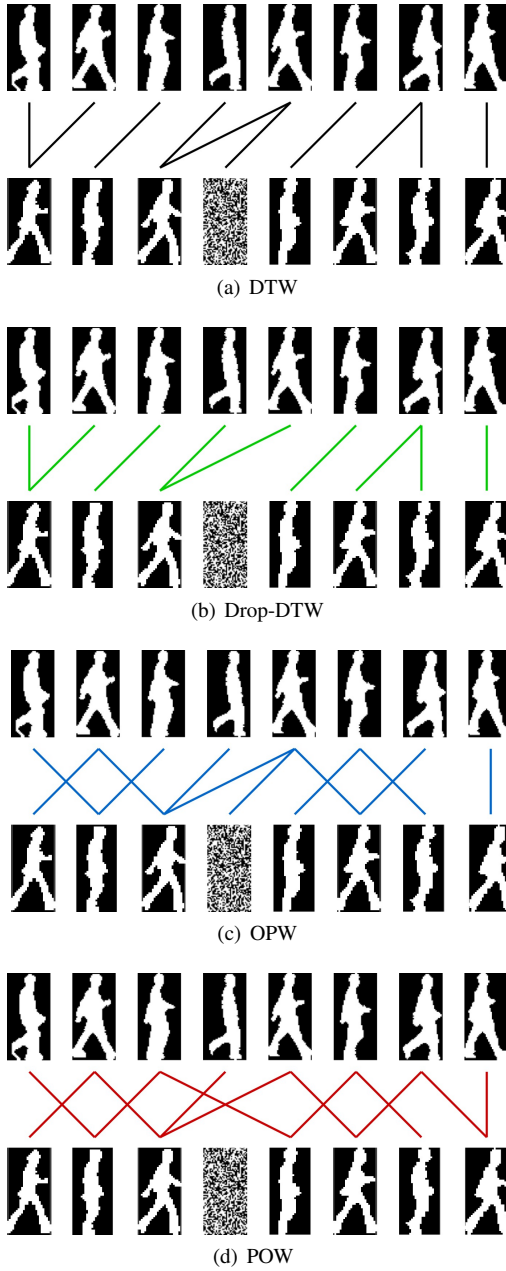


Figure 8: Alignments returned by several methods on two video sequences of different people performing the walking action taken from Weizman dataset. The solid lines show the frame-wise matching and the fourth frame of the second sequence is the outlier, which is significantly different from the other frames.

Following many prior works in time series classification [37, 61] we further constructed a critical difference (CD) diagram to compare the performances of k-NN with different distance measures in terms of statistical significance. Figure 7(a) shows the CD diagram, which depicts the mean rank corresponding to each distance and indicates whether two distances exhibit a statistically significant difference in performance based on the Wilcoxon signed-rank test ($\alpha = 0.05$) [22]. The conclusion is consistent with the findings from Table 2.

We also evaluated the performances of k-NN with different distance measures on the original UCR datasets (with no added outlier). For each dataset, we merged the original training and test sets and randomly re-divided them into five equal folds. Then, the classifiers were executed on each dataset using 5-fold cross-validation. The results are presented in Table 3, and the corresponding Critical Difference (CD) diagram is depicted in Figure 7(b). Notably, even in the absence of outliers, POW maintains its position as the best distance measure, albeit with performance differences that are not as significant as in the presence of outliers. OPW reaches better performances than both Drop-DTW and OPW (even if these differences are not statistically significant according to the Friedman test with $\alpha = 5\%$). Both POW and OPW are grounded in the Optimal Transport (OT) framework. However, the proposed regularization in POW effectively mitigates numerical instability issues, resulting in slightly better accuracy compared to OPW.

We then evaluate the performance of the aforementioned distance measures on the Weizmann dataset. Figure 8 illustrates the alignments generated by various measures for two videos featuring different subjects performing a walking action. Notably, Drop-DTW and POW demonstrate effective exclusion of abnormal frames from the alignment, thanks to their outlier-removing mechanisms. In contrast, methods like DTW and OPW align all frames, including the outliers. This inclusion of abnormal frames significantly impacts the distance calculation, leading to unreliable results. Furthermore, the figure illustrates that POW achieves better frame-wise matching compared to DTW and Drop-DTW. Indeed, unlike these methods that strictly prevent alignment between frames in one video sequence and those preceding the already aligned frames in the other sequence, POW offers flexibility inherited from Optimal Transport (OT). This flexibility is crucial in aligning frames from human action videos with periodic patterns, where strict constraints may lead to mismatches. As a consequence, alignment results of POW are more suitable for practical applications. Table 4 and Table 5 shows the classification results on the Weizmann dataset with and without adding outliers respectively, using k-NN with different distance measures. These tables also include the results of McNemar's test conducted between the proposed distance and each of its counterparts. Notably, POW consistently enables k-NN to achieve the highest classification accuracy when the dataset is contaminated by outliers, highlighting the advantages of the proposed distance. On the original dataset, OPW and POW exhibit competitive performances and significantly surpass the remaining baselines. Although OPW facilitates the 1-NN and 3-NN classifiers in achieving the highest accuracy, its performances with the remaining values of k do not exhibit significant differences compared to those of POW. Therefore, on this dataset, we can consider POW as a viable alternative to OPW, with the advantage of numerical stability.

Distance		1-NN	3-NN	5-NN	7-NN	15-NN	30-NN
Variants of DTW	DTW	69.50 (4.05)*	45.25 (5.14)*	72.00 (3.69)*	69.75 (5.58)*	64.70 (3.59)*	66.82 (4.21)*
	Soft-DTW	52.50 (4.54)*	58.25 (3.55)*	55.25 (4.17)*	54.25 (4.11)*	53.27 (3.92)*	55.21 (3.83)*
	SS-DTW	60.42 (3.50)*	59.55 (4.15)*	57.03 (4.11)*	59.25 (3.47)*	54.12 (3.99)*	56.62 (4.05)*
	GW-DTW	65.61 (4.15)*	52.61 (3.91)*	62.55 (4.37)*	60.15 (4.19)*	61.08 (3.94)*	63.28 (3.73)*
	Drop-DTW	79.50 (3.83)*	80.25 (4.16)*	77.75 (3.68)*	71.25 (3.76)*	73.24 (3.72)*	76.60 (4.19)*
OT-based distances	OPW	76.25 (3.56)*	81.34 (3.92)*	75.54 (3.48)*	78.46 (3.52)*	79.03 (4.17)*	79.62 (4.23)*
	WSK	79.67 (4.16)*	80.82 (4.13)*	79.32 (3.87)*	81.01 (3.82)	77.77 (3.93)*	78.82 (3.78)*
	OTW	75.54 (4.48)*	76.55 (4.16)*	77.05 (3.92)*	76.59 (4.18)*	76.42 (3.85)*	76.69 (4.01)*
	POW	92.00 (3.08)	95.25 (3.12)	86.75 (3.09)	81.25 (3.17)	84.12 (3.05)	83.86 (3.21)

Table 4

The classification results of k -NN with different distance measures on the Weizmann dataset with additional outliers. The classifiers were executed on each dataset five times with varying proportions of outliers, and the average accuracy in percentage, along with the standard deviations (in brackets), are reported. The highest accuracy for each k is presented in **bold**. The symbol * indicates that the corresponding distance measure performed worse than POW, while the symbol † indicates that the corresponding distance measure performed better than POW with significance at the 5% level by McNemar's test [22].

Distance		1-NN	3-NN	5-NN	7-NN	15-NN	30-NN
Variants of DTW	DTW	87.50 (4.35)*	82.50 (4.27)*	82.50 (3.88)*	80.40 (4.20)	77.30 (3.75)*	79.94 (4.45)*
	Soft-DTW	88.30 (4.48)*	82.10 (3.22)*	82.10 (3.05)*	80.80 (3.45)*	79.67 (3.13)*	79.67 (3.51)*
	SS-DTW	83.50 (3.77)*	79.45 (4.03)*	80.05 (4.27)*	79.58 (3.89)*	75.62 (4.08)*	76.33 (4.21)*
	GW-DTW	85.81 (4.35)*	82.55 (4.11)*	82.03 (4.65)*	80.37 (4.41)*	81.05 (3.87)*	79.32 (3.85)*
	Drop-DTW	89.33 (4.05)*	83.75 (4.71)*	79.57 (3.83)*	81.54 (4.05)*	79.44 (4.52)*	75.67 (3.94)*
OT-based distances	OPW	97.50 (3.42)	95.00 (3.86)	90.41 (2.37)	87.33 (3.48)	84.12 (4.35)	81.16 (4.52)
	WSK	87.42 (3.98)*	84.64 (4.42)*	84.22 (3.95)*	80.11 (3.52)*	76.38 (3.37)*	75.87 (3.48)*
	OTW	85.67 (4.18)*	85.85 (4.53)*	83.36 (3.84)*	82.87 (4.24)*	79.84 (3.94)*	77.67 (4.15)*
	POW	96.67 (3.44)	94.14 (3.17)	89.50 (3.31)	88.67 (3.63)	84.42 (4.10)	82.46 (3.83)

Table 5

The classification results of k -NN with different distance measures on the Weizmann dataset without adding outliers are presented. The dataset was divided into 5 equal folds, and the classifiers were executed on each dataset using 5-fold cross-validation. The average accuracy in percentage on all the folds, along with the standard deviations (in brackets), are reported. The highest accuracy for each k is presented in **bold**. The symbol * indicates that the corresponding distance measure performed worse than POW, while the symbol † indicates that the corresponding distance measure performed better than POW with significance at the 5% level by McNemar's test [22].

6.2. Multi-step localization

Datasets. In our multi-step localization experiments, we used three recent instructional video datasets: CrossTask² [80], COIN³ [67], and YouCook2⁴ [79]. CrossTask includes 2750 videos showing 18 different tasks, while YouCook2 has 2000 videos covering 89 recipes, and COIN contains 11827 videos with 778 procedures. We processed these datasets using a pretrained vision and language model to get embeddings for video frames and steps. Each dataset was divided into training, validation, and testing sets with a ratio of 60/20/20. Although all datasets provide frame-wise labels for the start and end times of each step in a video, we only considered a set of steps in a weakly-supervised mode. It is worth noting that our approach differs from previous work, where sequences of steps following the exact order of occurrences in the video were required.

Compared methods. We compare our methods with the following weakly supervised baselines:

- D³TW [16] that learns to align video frames with its step labels using a different differentiable formulation of DTW and a discriminative loss,
- OTAM [15], which is an extension of D³TW that allows skipping through outliers at the beginning and end of video sequences,
- Drop-DTW [24] that aligns frame and step embeddings using DTW framework equipped with mechanism to drop irrelevant frames,
- Graph2Vid [25], which capture all possible orders of steps in the given set using flow graphs and align them with frames using Drop-DTW.

Note that all the above methods require to specify ground distance between frame and step embeddings in advance. For a fair comparison, similarly to [24, 25], we use cosine distance as the ground distance for all the methods.

Evaluation methods. We assess the performance of the proposed methods and baselines using two distinct metrics. The first metric, *Framewise accuracy (Acc.)* [67], calculates the proportion of correctly labeled frames (excluding outliers) relative to the total number of frames. The second

²<https://github.com/DmZhukov/CrossTask>

³<https://coin-dataset.github.io/>

⁴<http://youcook2.eecs.umich.edu/>

Method	CrossTask			YouCook2			COIN		
	Acc.	IoU	Mc	Acc.	IoU	Mc	Acc.	IoU	Mc
D ³ TW	11.23	9.31	*	26.72	22.73	*	23.72	19.77	*
OTAM	18.86	10.84	*	33.20	26.45	*	28.66	21.18	*
Drop-DTW	67.34	19.91	*	48.92	31.16	†	50.74	23.25	*
Graph2Vid	68.12	20.24	*	48.64	32.31		51.44	22.92	*
POW	68.30	19.25	*	47.51	25.66		51.93	23.53	*
SRPOW	70.31	20.55		47.23	28.99		53.84	25.65	

Table 6

Results of different methods performing multi-step localization inference tasks on CrossTask, COIN, and YouCook2 datasets. The highest scores for each datasets are presented in **bold**. The column "Mc" shows the result of the McNemar's test. The mark * indicates that the corresponding method performed worse than SRPOW with statistical significant at the 5% level. If the mark is †, the corresponding method performed better than SRPOW with significance at the 5% level.

metric, *Intersection over Union (IoU)* [79], quantifies the overlap between predicted and actual time intervals for each step label. It is determined by dividing the sum of the intersections of these intervals by the sum of their unions. It is worth noting that IoU is the more stringent of these two metrics because it heavily penalizes any misalignment. In addition to the above two metrics, we also conduct the McNemar's tests [54] between the proposed method and each baselines. The significance level was set to $\alpha = 0.05$.

Parameter tuning. SRPOW has three parameters: the transported mass s ; the parameter λ for order regularization; and the parameter γ for segment regularization. During the training phase, we set s as the ratio between the number of outlier frames and the total number of frames since the ground truth labels for each frame were provided in advance. In the inference phase, we employed Algorithm 2 to automatically approximate the best transported mass s . The remaining two parameters were selected through a grid search. Note that, when $\gamma = 0$, SRPOW reduces to the original POW version. Therefore, in our experiments, we also set γ to zero for an ablation study. For the Graph2Vid method, we utilized the flow graphs provided by the authors⁵, which were constructed for the CrossTask dataset. For the two remaining datasets, we generated the flow graphs based on the step information provided in the ground truth, following the procedure proposed in [25]. Parameters for the remaining methods were set according to the original papers.

Results and discussion. Table 6 shows the results of different methods on the multi-step localization task. We also visualize the results for two video examples in Figures 9 and 10, respectively. It is evident that D³TW, solely based on DTW, performs poorly on all the datasets. Since DTW lacks a mechanism to remove outliers, D³TW cannot exclude irrelevant frames, such as people talking and advertisements, from the alignment. OTAM mitigates this issue with the ability to skip outlier frames at the beginning and end of video sequences. However, as outliers often intersperse within video sequences, OTAM appears to be ineffective. Drop-DTW outperforms both D³TW and OTAM on all the

⁵<https://github.com/SamsungLabs/Graph2Vid>

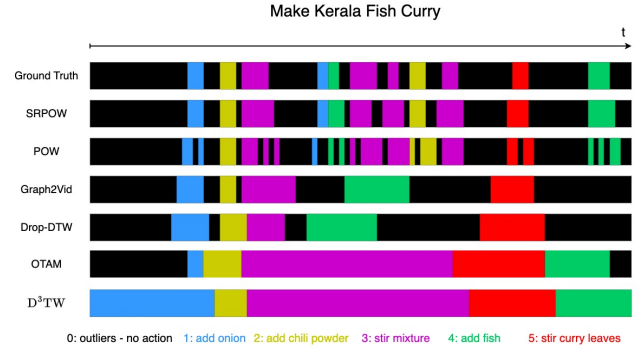


Figure 9: Step assignment results on an instructional video of making Kerala Fish Curry from CrossTask dataset of the proposed methods and baselines. Different colors correspond to different steps.

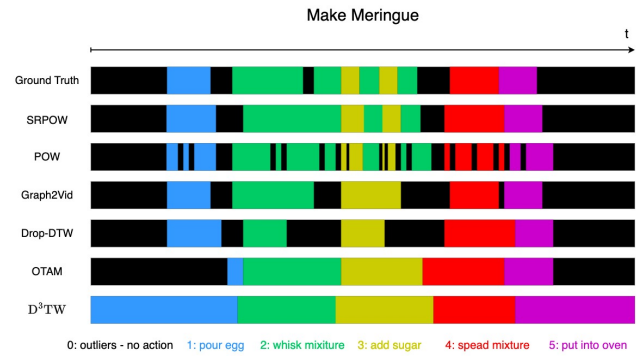


Figure 10: Step assignment results on an instructional video of making Meringue from CrossTask dataset of the proposed methods and baselines. Different colors correspond to different steps.

datasets. This result validates the effectiveness of its outlier-removing mechanism.

Graph2Vid extends Drop-DTW by constructing a flow graph to capture all possible orders of action steps in the given set for each video. Therefore, it can handle videos where people perform the same activities in different orders. Table 6 shows that Graph2Vid achieves slightly better results than Drop-DTW on CrossTask and surpasses all the competitors on YouCook2 in terms of the IoU metric. However, we note that Graph2Vid constructs the graphs assuming that each step occurs only once in each video. This assumption is unrealistic, as can be observed from the ground truth depicted in Figures 9 and Figure 10, where a step can be repeated several times in a video.

Similar to Drop-DTW, POW and its segment-regularized version are equipped with an outlier-removing mechanism whose parameter is selected automatically and adaptively from the data. Therefore, the proposed distance measures, as exhibited in Table 6, show significant improvements over D³TW and OTAM. By inheriting the flexible alignment ability from the OT framework, POW and SRPOW can

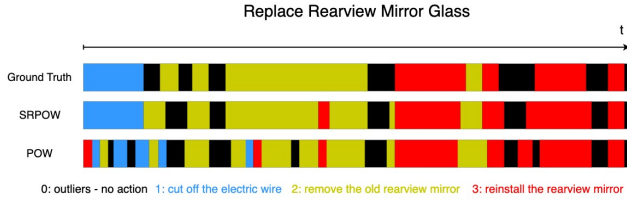


Figure 11: Step assignment results on an instructional video of replacing rearview mirror glass from COIN dataset of the POW with and without segment regularization. Different colors correspond to different steps.

assign frames to their corresponding steps, disregarding repetitions and the local swap in the order of the action steps. It is worth noting from Figures 9 and 10 that POW divides several steps with long durations in the videos into many fractions, which negatively affects its step assignment results. SRPOW with segment regularization, in contrast, produces much smoother results. Thus, its step assignment is more consistent with the ground truth than the original version of POW. Figure 11, which compares step assignment results between POW and SRPOW in a video example from the COIN dataset, further reinforces the assessment above.

7. Conclusion

In this paper, we introduce a novel distance measure called *Partial Ordered Wasserstein* (POW) for sequential data. Built upon the OT framework, POW possesses two particularly beneficial properties. First, POW can produce flexible alignments to handle local distortions in sequences. Second, it allows for limiting the amount of transported mass to prevent outliers from deteriorating the alignment and influencing distance calculations. We provide theoretical proof that, by properly setting the amount of transported mass, we can eliminate all outliers while retaining all normal elements in the alignment and calculation results.

To facilitate the computation of POW, we introduce an algorithm for automatic and adaptive selection of the best value for the portion of transported mass, based on sudden increases in the first derivative of the distance. We extensively study the applications of POW in time-series classification and multi-step localization tasks. Additionally, we propose a segment-regularized version of POW – SRPOW – to enhance its performance in multi-step localization. Extensive experiments on widely recognized benchmarks were conducted for both tasks, and the results confirm the advantages of our proposed distance measures over existing competitors.

Although POW has shown promising preliminary results, there are still some interesting open problems for future research. First, akin to existing OT-based distances like OPW [64], POW employs a regularization term to force transportation within elements possessing relatively similar positions in their respective sequences. However, under certain circumstances, two sequences may exhibit identical

shapes but differing evolution speeds. For instance, younger people often execute a specific physical activity more rapidly than their elder counterparts. In such scenarios, facilitating transportation between some elements with distant positions becomes imperative. Hence, we are currently exploring novel regularization techniques that can accommodate sequences with both similar and different evolution speeds.

Second, a noticeable trend in recent research involves integrating distance measures for sequential data into deep neural networks (DNNs). Given that POW is also grounded in the OT framework just like OTW, there also exists potential for its integration into DNNs. In addition, recent advancements in DNN architecture, such as subnets formed by groups of neurons, as proposed in [74], present an opportunity to enhance capacity and reduce computational workloads [72, 77, 76] of DNNs. Incorporating the proposed distances within such novel architecture represents a new and exciting avenue for our future research.

A. Proof of Lemma 1

In this appendix, we provide a detailed proof of Lemma 1. Recall that

$$T_{s^*}^* = \underset{T \in \Pi_{s^*}(u_N, u_M)}{\operatorname{argmin}} \sum_{i=1}^N \sum_{j=1}^M d_{i,j} t_{i,j} + \lambda \sum_{i=1}^N \sum_{j=1}^M \left(\frac{i}{N} - \frac{j}{M} \right)^2 t_{i,j}, \quad (27)$$

where $s^* = \frac{N^+}{N}$ is the total mass that corresponds to normal elements in sequence \mathbf{X} . We aim to show that $T_{s^*}^*$ has no entry $t_{i,j}^* > 0$ such that $i \in \mathcal{I}^-$. In other words, no mass that corresponds to outliers in \mathbf{X} is transported to \mathbf{Y} .

We prove this by contradiction. Specifically, we assume that there is at least one index $i^- \in \mathcal{I}^-$ such that $t_{i^-,k}^* > 0$. Then, we have the following:

$$\sum_{i \in \mathcal{I}^+} \sum_{j=1}^M t_{i,j}^* < s^*. \quad (28)$$

This inequality indicates that all the mass from normal elements in \mathbf{X} is not fully transported to \mathbf{Y} . Therefore, there exists an index $i^+ \in \mathcal{I}^+$ such that

$$\sum_{j=1}^M t_{i^+,j}^* < \frac{1}{N}. \quad (29)$$

Let $\epsilon = \min \left(t_{i^-,k}^*, \frac{1}{N} - \sum_{j=1}^M t_{i^+,j}^* \right)$. We then construct a new transport matrix $T_{s^*}^*$ from $T_{s^*}^*$ as follows:

$$t_{i,j} = \begin{cases} t_{i^-,k}^* - \epsilon & \text{if } i = i^-, j = k \\ t_{i^+,k}^* + \epsilon & \text{if } i = i^+, j = k \\ t_{i,j}^* & \text{otherwise} \end{cases} \quad (30)$$

It is obviously that $T_{s^*}^* \in \mathbb{R}_+^{N \times M}$. Furthermore, we have

$$\mathbf{1}_N^\top T_{s^*}^* \mathbf{1}_M = \sum_{i=1}^N \sum_{j=1}^M t_{i,j} = s^*. \quad (31)$$

This is because T_{s^*} is only different from $T_{s^*}^*$ at two entries $t_{i^-,k}$ and $t_{i^+,k}$. However, it is clear that

$$t_{i^-,k} + t_{i^+,k} = t_{i^-,k}^* - \epsilon + t_{i^+,k}^* + \epsilon = t_{i^-,k}^* + t_{i^+,k}^* \quad (32)$$

Therefore, summation of their entries remains the same. We also have

$$T_{s^*} \mathbf{1}_M \leq \mathbf{u}_N, \quad (33)$$

because

$$\sum_{j=1}^M t_{i,j} = \sum_{j=1}^M t_{i,j}^* \leq \frac{1}{N} \quad \forall i \notin \{i^-, i^+\}, \quad (34)$$

$$\sum_{j=1}^M t_{i^-,j} = \sum_{j=1}^M t_{i^-,j}^* - \epsilon \leq \frac{1}{N}, \quad (35)$$

$$\begin{aligned} \text{and } \sum_{j=1}^M t_{i^+,j} &= \sum_{j=1}^M t_{i^+,j}^* + \epsilon \\ &= \sum_{j=1}^M t_{i^+,j}^* + \min \left(t_{i^-,k}^*, \frac{1}{N} - \sum_{j=1}^M t_{i^+,j}^* \right) \\ &\leq \frac{1}{N}. \end{aligned} \quad (36)$$

T_{s^*} also satisfies

$$T_{s^*}^\top \mathbf{1}_N \leq \mathbf{u}_M, \quad (37)$$

because

$$\sum_{i=1}^N t_{i,j} = \sum_{i=1}^N t_{i,j}^* \leq \frac{1}{M} \quad \forall j \neq k, \quad (38)$$

$$\begin{aligned} \text{and } \sum_{i=1}^N t_{i,k} &= \sum_{\substack{i=1, \\ i \notin \{i^-, i^+\}}}^N t_{i,k} + t_{i^-,k} + t_{i^+,k} \\ &= \sum_{\substack{i=1, \\ i \notin \{i^-, i^+\}}}^N t_{i,k}^* + t_{i^-,k}^* - \epsilon + t_{i^+,k}^* + \epsilon \\ &= \sum_{i=1}^N t_{i,k}^* \leq \frac{1}{M}. \end{aligned} \quad (39)$$

From (31), (33), and (37), we have $T_{s^*} \in \Pi_{s^*}(\mathbf{u}_N, \mathbf{u}_M)$. Recall that T_{s^*} is only different from $T_{s^*}^*$ at two entries of index (i^-, k) and (i^+, k) , therefore, we have

$$\begin{aligned} &\sum_{i=1}^N \sum_{j=1}^M \left[d_{i,j} + \lambda \left(\frac{i}{N} - \frac{j}{M} \right)^2 \right] t_{i,j}^* \\ &- \sum_{i=1}^N \sum_{j=1}^M \left[d_{i,j} + \lambda \left(\frac{i}{N} - \frac{j}{M} \right)^2 \right] t_{i,j} \\ &= \left[d_{i^-,k} + \lambda \left(\frac{i^-}{N} - \frac{k}{M} \right)^2 \right] (t_{i^-,k}^* - t_{i^-,k}) \\ &+ \left[d_{i^+,k} + \lambda \left(\frac{i^+}{N} - \frac{k}{M} \right)^2 \right] (t_{i^+,k}^* - t_{i^+,k}) \\ &= \left[d_{i^-,k} + \lambda \left(\frac{i^-}{N} - \frac{k}{M} \right)^2 - d_{i^+,k} - \lambda \left(\frac{i^+}{N} - \frac{k}{M} \right)^2 \right] \epsilon \end{aligned}$$

$$\geq [d_{i^-,k} - d_{i^+,k} - \lambda] \epsilon. \quad (40)$$

Equation (40) is derived from the facts that $\left(\frac{i^-}{N} - \frac{k}{M} \right)^2 \geq 0$ and $\left(\frac{i^+}{N} - \frac{k}{M} \right)^2 \leq 1$. Recall that $d_{i^-,k} \geq C(d_{\max} + \lambda)$, while $d_{i^+,k} \leq d_{\max}$. Therefore, we have

$$[d_{i^-,k} - d_{i^+,k} - \lambda] \epsilon \geq 0. \quad (41)$$

This contradicts to the fact that $T_{s^*}^*$ is the optimal solution of the problem (27). As a result, the assumption is incorrect and we can conclude that $T_{s^*}^*$ has no entry $t_{i,j}^* > 0$ such that $i \in \mathcal{I}^-$. The proof is completed.

B. Proof of Lemma 2

This appendix provides a detailed proof of the Lemma 2. From definition of the derivative, we have the followings

$$\frac{\partial \text{POW}_{X,Y}(s^* - \epsilon)}{\partial s} = \lim_{\Delta s \rightarrow 0} \frac{\text{POW}_{X,Y}(s^* - \epsilon + \Delta s) - \text{POW}_{X,Y}(s^* - \epsilon)}{\Delta s}, \quad (42)$$

$$\frac{\partial \text{POW}_{X,Y}(s^* + \epsilon)}{\partial s} = \lim_{\Delta s \rightarrow 0} \frac{\text{POW}_{X,Y}(s^* + \epsilon + \Delta s) - \text{POW}_{X,Y}(s^* + \epsilon)}{\Delta s}, \quad (43)$$

where Δs is a very small mass that satisfies $0 \geq \Delta s < \epsilon$. From Lemma 1, we know that both $T_{s^* - \epsilon}^*$ and $T_{s^* - \epsilon + \Delta s}^*$ have no entries of index (i, j) such that $i \in \mathcal{I}^-$. Therefore, the additional mass Δs is transported from normal elements of \mathbf{X} to \mathbf{Y} . This implies that

$$\text{POW}_{X,Y}(s^* - \epsilon + \Delta s) - \text{POW}_{X,Y}(s^* - \epsilon) \leq \Delta s \times d_{\max}. \quad (44)$$

Substituting (44) into (42), we have

$$\frac{\partial \text{POW}_{X,Y}(s^* - \epsilon)}{\partial s} \leq \lim_{\Delta s \rightarrow 0} \frac{\Delta s \times d_{\max}}{\Delta s} = d_{\max}. \quad (45)$$

Similarly, it is obvious that $T_{s^* + \epsilon}^*$ starts to transport masses from outliers in \mathbf{X} to \mathbf{Y} as its total transported mass is now larger than s^* . Therefore, considering $T_{s^* + \epsilon + \Delta s}^*$, the additional mass Δs is definitely also transported from outliers. This implies that

$$\text{POW}_{X,Y}(s^* + \epsilon + \Delta s) - \text{POW}_{X,Y}(s^* + \epsilon) \geq \Delta s \times C(d_{\max} + \lambda). \quad (46)$$

Substituting (46) into (43), we have

$$\frac{\partial \text{POW}_{X,Y}(s^* + \epsilon)}{\partial s} \geq \lim_{\Delta s \rightarrow 0} \frac{\Delta s \times C(d_{\max} + \lambda)}{\Delta s} = C(d_{\max} + \lambda). \quad (47)$$

The proof is completed.

C. GCG algorithm for SRPOW

In this appendix, we drive the generalized conditional gradient (GCG) algorithm for solving optimization problem (26) of segment regularized - POW (SRPOW). In general, the GCG algorithm framework addresses the scenario of

Algorithm 3 : Generalized Conditional Gradient

- 1: Initialize $k = 0$ and $\mathbf{T}^0 \in \mathcal{B}$;
- 2: **while** not convergence **do**
- 3: With $\mathbf{G} \in \nabla f(\mathbf{T}^k)$, solve

$$\mathbf{T}^* = \underset{\mathbf{T} \in \mathcal{B}}{\operatorname{argmin}} \langle \mathbf{T}, \mathbf{G} \rangle_F + g(\mathbf{T});$$

- 4: Find the optimal step α^k

$$\alpha^k = \underset{0 \leq \alpha \leq 1}{\operatorname{argmin}} f(\mathbf{T}^k + \alpha \Delta \mathbf{T}) + g(\mathbf{T}^k + \alpha \Delta \mathbf{T})$$

with $\Delta \mathbf{T} = \mathbf{T}^* - \mathbf{T}^k$

- 5: $\mathbf{T}^{k+1} \leftarrow \mathbf{T}^k + \alpha^k \Delta \mathbf{T}$, set $k \leftarrow k + 1$;
- 6: **end while**

constrained minimization of composite functions defined as

$$\min_{\mathbf{T} \in \mathcal{B}} f(\mathbf{T}) + g(\mathbf{T}) \quad (48)$$

where $f(\cdot)$ is a potentially non-convex differentiable function, $g(\cdot)$ is a possibly non-differentiable convex function, and \mathcal{B} represents any convex and compact subset of \mathbb{R}^d . Details of GCG for solving the general problem (48) are presented in Algorithm 3.

The GCG algorithm has been shown in [9] that it converges towards a stationary point of (48). In the case of SRPOW, we set

$$f(\mathbf{T}) = \langle \mathbf{D}, \mathbf{T} \rangle_F + \gamma \|\mathbf{S}\mathbf{T}^\top\|_F^2 \quad (49)$$

$$\text{and } g(\mathbf{T}) = \lambda \sum_{i=1}^N \sum_{j=1}^M \left(\frac{i}{N} - \frac{j}{M} \right)^2 t_{i,j}, \quad (50)$$

and $\mathcal{B} = \Pi_s(\mathbf{u}_N, \mathbf{u}_M)$. Then algorithm 3 can be utilized to solve the problem (26) of SRPOW. Since $g(\mathbf{T})$ is differentiable, stronger convergence results can be obtained. We further denote $\Omega(\mathbf{T}) = \|\mathbf{S}\mathbf{T}^\top\|_F^2$ and find that $\Omega(\mathbf{T})$ is also differentiable with respect to \mathbf{T} . Computation of its gradient $\nabla \Omega(\mathbf{T}^k)$ is derived in more details in Appendix D. Therefore, step 3 of the algorithm 3 boils down to

$$\begin{aligned} \mathbf{T}^* = \underset{\mathbf{T} \in \Pi_s(\mathbf{u}_N, \mathbf{u}_M)}{\operatorname{argmin}} & \langle \mathbf{T}, \mathbf{D} + \gamma \nabla \Omega(\mathbf{T}^k) \rangle_F \\ & + \lambda \sum_{i=1}^N \sum_{j=1}^M \left(\frac{i}{N} - \frac{j}{M} \right)^2 t_{i,j} \end{aligned} \quad (51)$$

Interestingly, the problem at hand corresponds to the original POW problem. Therefore, we can utilize matrix scaling algorithm to effectively solve it.

D. Gradient computation of segment regularization

We have

$$\Omega(\mathbf{T}) = \|\mathbf{S}\mathbf{T}^\top\|_F^2 = \|\mathbf{T}\mathbf{S}^\top\|_F^2 = \operatorname{Tr}(\mathbf{T}\mathbf{S}^\top\mathbf{S}\mathbf{T}^\top). \quad (52)$$

Let \mathbf{J} be the $M \times N$ matrix whose entries are all 0 except one at position i, j , which is equal to 1. Let $\Delta \mathbf{T} = \epsilon \mathbf{J}$, where $\epsilon > 0$ is tiny. Then

$$\begin{aligned} \Omega(\mathbf{T} + \Delta \mathbf{T}) &= \operatorname{Tr}((\mathbf{T} + \Delta \mathbf{T})\mathbf{S}^\top\mathbf{S}(\mathbf{T} + \Delta \mathbf{T})^\top) \\ &= \operatorname{Tr}(\mathbf{T}\mathbf{S}^\top\mathbf{S}\mathbf{T}^\top) + \operatorname{Tr}(\Delta \mathbf{T}\mathbf{S}^\top\mathbf{S}\mathbf{T}^\top) \\ &\quad + \operatorname{Tr}(\mathbf{T}\mathbf{S}^\top\mathbf{S}\Delta \mathbf{T}^\top) + \operatorname{Tr}(\Delta \mathbf{T}\mathbf{S}^\top\mathbf{S}\Delta \mathbf{T}^\top) \\ &\approx \operatorname{Tr}(\mathbf{T}\mathbf{S}^\top\mathbf{S}\mathbf{T}^\top) + \operatorname{Tr}(\Delta \mathbf{T}\mathbf{S}^\top\mathbf{S}\mathbf{T}^\top) \\ &\quad + \operatorname{Tr}(\mathbf{T}\mathbf{S}^\top\mathbf{S}\Delta \mathbf{T}^\top) \\ &= \operatorname{Tr}(\mathbf{T}\mathbf{S}^\top\mathbf{S}\mathbf{T}^\top) + 2 \operatorname{Tr}(\mathbf{T}\mathbf{S}^\top\mathbf{S}\Delta \mathbf{T}^\top) \\ &= \Omega(\mathbf{T}) + 2 \langle \mathbf{T}\mathbf{S}^\top\mathbf{S}, \Delta \mathbf{T} \rangle \\ &= \Omega(\mathbf{T}) + 2\epsilon \langle \mathbf{T}\mathbf{S}^\top\mathbf{S}, \mathbf{J} \rangle. \end{aligned} \quad (53)$$

Comparing this result with $\Omega(\mathbf{T} + \epsilon \mathbf{J}) \approx \Omega(\mathbf{T}) + \epsilon \frac{\partial \Omega(\mathbf{T})}{\partial t_{i,j}}$, we obtain

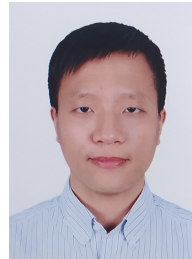
$$\begin{aligned} \frac{\partial \Omega(\mathbf{T})}{\partial t_{i,j}} &= 2 \langle \mathbf{T}\mathbf{S}^\top\mathbf{S}, \mathbf{J} \rangle \\ &= 2 \operatorname{Tr}((\mathbf{T}\mathbf{S}^\top\mathbf{S})^\top \mathbf{J}) \\ &= 2 \operatorname{Tr}(\mathbf{S}^\top\mathbf{T}^\top \mathbf{J}). \end{aligned} \quad (54)$$

References

- [1] Abanda, A., Mori, U., Lozano, J.A., 2019. A review on distance based time series classification. *Data Mining and Knowledge Discovery* 33, 378–412.
- [2] Aghabozorgi, S., Shirkhorshidi, A.S., Wah, T.Y., 2015. Time-series clustering—a decade review. *Information systems* 53, 16–38.
- [3] Albregtsen, F., et al., 2008. Statistical texture measures computed from gray level cooccurrence matrices. *Image processing laboratory, department of informatics, university of oslo* 5.
- [4] Altschuler, J., Niles-Weed, J., Rigollet, P., 2017. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *Advances in neural information processing systems* 30.
- [5] Arici, T., Celebi, S., Aydin, A.S., Temiz, T.T., 2014. Robust gesture recognition using feature pre-processing and weighted dynamic time warping. *Multimedia Tools and Applications* 72, 3045–3062.
- [6] Bai, L., Cui, L., Zhang, Z., Xu, L., Wang, Y., Hancock, E.R., 2020. Entropic dynamic time warping kernels for co-evolving financial time series analysis. *IEEE Transactions on Neural Networks and Learning Systems*.
- [7] Blondel, M., Mensch, A., Vert, J., 2021. Differentiable divergences between time series: Proceedings of the 24th international conference on artificial intelligence and statistics.
- [8] Bock, C., Togninalli, M., Ghisu, E., Gumbsch, T., Rieck, B., Borgwardt, K., 2019. A wasserstein subsequence kernel for time series, in: 2019 IEEE International Conference on Data Mining (ICDM), IEEE. pp. 964–969.
- [9] Bredies, K., Lorenz, D.A., Maass, P., 2007. A generalized conditional gradient method and its connection to an iterative shrinkage method. *Computational Optimization and Applications* 42, 173–193. doi:10.1007/s10589-007-9083-3.
- [10] Buza, K., 2018. Time series classification and its applications, in: Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, pp. 1–4.
- [11] Buza, K., Antal, M., 2021. Convolutional neural networks with dynamic convolution for time series classification, in: Advances in Computational Collective Intelligence: 13th International Conference, ICCCI 2021, Kallithea, Rhodes, Greece, September 29–October 1, 2021, Proceedings 13, Springer. pp. 304–312.

- [12] Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J., 2015. Activitynet: A large-scale video benchmark for human activity understanding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 961–970.
- [13] Caffarelli, L.A., McCann, R.J., 2010. Free boundaries in optimal transport and monge-ampere obstacle problems. *Annals of mathematics*, 673–730.
- [14] Cai, X., Xu, T., Yi, J., Huang, J., Rajasekaran, S., 2019. Dtnet: a dynamic time warping network. *Advances in neural information processing systems* 32.
- [15] Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C., 2020. Few-shot video classification via temporal alignment, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10618–10627.
- [16] Chang, C.Y., Huang, D.A., Sui, Y., Fei-Fei, L., Niebles, J.C., 2019. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3546–3555.
- [17] Chang, X., Tung, F., Mori, G., 2021. Learning discriminative prototypes with dynamic time warping, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8395–8404.
- [18] Chapel, L., Alaya, M.Z., Gasso, G., 2020. Partial optimal transport with applications on positive-unlabeled learning. *Advances in Neural Information Processing Systems* 33, 2903–2913.
- [19] Cuturi, M., 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26.
- [20] Cuturi, M., Blondel, M., 2017. Soft-dtw: a differentiable loss function for time-series, in: International conference on machine learning, PMLR. pp. 894–903.
- [21] Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E., 2019. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 1293–1305.
- [22] Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research* 7, 1–30.
- [23] Ding, L., Xu, C., 2018. Weakly-supervised action segmentation with iterative soft boundary assignment, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6508–6516.
- [24] Dvornik, M., Hadji, I., Derpanis, K.G., Garg, A., Jepson, A., 2021. Drop-dtw: Aligning common signal between sequences while dropping outliers. *Advances in Neural Information Processing Systems* 34, 13782–13793.
- [25] Dvornik, N., Hadji, I., Pham, H., Bhatt, D., Martinez, B., Fazly, A., Jepson, A.D., 2022. Flow graph to video grounding for weakly-supervised multi-step localization, in: European Conference on Computer Vision, Springer. pp. 319–335.
- [26] Dvornik, N., Hadji, I., Zhang, R., Derpanis, K.G., Wildes, R.P., Jepson, A.D., 2023. Stepformer: Self-supervised step discovery and localization in instructional videos, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18952–18961.
- [27] Eltrass, A.S., Tayel, M.B., Ammar, A.I., 2022. Automated ecg multi-class classification system based on combining deep learning features with hrv and ecg measures. *Neural Computing and Applications* 34, 8755–8775.
- [28] Faouzi, J., 2022. Time series classification: A review of algorithms and implementations. *Machine Learning (Emerging Trends and Applications)*.
- [29] Figalli, A., 2010. The optimal partial transport problem. *Archive for rational mechanics and analysis* 195, 533–560.
- [30] Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R., 2007. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence* 29, 2247–2253.
- [31] Gupta, A., Gupta, H.P., Biswas, B., Dutta, T., 2020. Approaches and applications of early classification of time series: A review. *IEEE Transactions on Artificial Intelligence* 1, 47–61.
- [32] Han, T., Xie, W., Zisserman, A., 2022. Temporal alignment networks for long-term video, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2906–2916.
- [33] Holder, C., Middlehurst, M., Bagnall, A., 2024. A review and evaluation of elastic distance functions for time series clustering. *Knowledge and Information Systems* 66, 765–809.
- [34] Hong, J.Y., Park, S.H., Baek, J.G., 2020. Ssdwt: Shape segment dynamic time warping. *Expert Systems with Applications* 150, 113291.
- [35] HORIE, M., KASAI, H., 2021. Optimal transport based sequence matching with grouped elements. *Proceedings of the Annual Conference of JSAI JSAI2021, 2G1GS2d01–2G1GS2d01*. doi:10.11517/pjsai.JSAI2021.0_2G1GS2d01.
- [36] Huang, D.A., Fei-Fei, L., Niebles, J.C., 2016. Connectionist temporal modeling for weakly supervised action labeling, in: Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14, Springer. pp. 137–153.
- [37] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A., 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery* 33, 917–963.
- [38] Iwana, B.K., Frinken, V., Uchida, S., 2020. Dtw-nn: A novel neural network for time series recognition using dynamic alignment between inputs and weights. *Knowledge-Based Systems* 188, 104971.
- [39] Jeong, Y.S., Jeong, M.K., Omata, O.A., 2011. Weighted dynamic time warping for time series classification. *Pattern recognition* 44, 2231–2240.
- [40] Jiang, Y., Qi, Y., Wang, W.K., Bent, B., Avram, R., Olgin, J., Dunn, J., 2020. Eventdtw: An improved dynamic time warping algorithm for aligning biomedical signals of nonuniform sampling frequencies. *Sensors* 20, 2700.
- [41] KAMURA, M., KASAI, H., 2022. A study of sequence matching method considering data transition. *Proceedings of the Annual Conference of JSAI JSAI2022, 2S6IS3d01–2S6IS3d01*. doi:10.11517/pjsai.JSAI2022.0_2S6IS3d01.
- [42] Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S., 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems* 3, 263–286.
- [43] Keogh, E.J., Pazzani, M.J., 2001. Derivative dynamic time warping, in: Proceedings of the 2001 SIAM international conference on data mining, SIAM. pp. 1–11.
- [44] Ko, D., Choi, J., Ko, J., Noh, S., On, K.W., Kim, E.S., Kim, H.J., 2022. Video-text representation learning via differentiable weak temporal alignment, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5016–5025.
- [45] Kulsoom, F., Narejo, S., Mehmood, Z., Chaudhry, H.N., Butt, A., Bashir, A.K., 2022. A review of machine learning-based human activity recognition for diverse applications. *Neural Computing and Applications* 34, 18289–18324.
- [46] Latorre, F., Liu, C., Sahoo, D., Hoi, S.C., 2023. Otw: Optimal transport warping for time series, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 1–5.
- [47] Lerogeron, H., Picot-Clément, R., Rakotomamonjy, A., Heutte, L., 2023. Approximating dynamic time warping with a convolutional neural network on eeg data. *Pattern Recognition Letters* 171, 162–169.
- [48] Li, H., Liu, J., Yang, Z., Liu, R.W., Wu, K., Wan, Y., 2020. Adaptively constrained dynamic time warping for time series classification and clustering. *Information Sciences* 534, 97–116.
- [49] Li, M., Zhu, Y., Zhao, T., Angelova, M., 2022. Weighted dynamic time warping for traffic flow clustering. *Neurocomputing* 472, 266–279.
- [50] Liero, M., Mielke, A., Savaré, G., 2018. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive

- measures. *Inventiones mathematicae* 211, 969–1117.
- [51] Lin, T., Ho, N., Jordan, M., 2019. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms, in: *International Conference on Machine Learning*, PMLR. pp. 3982–3991.
 - [52] Lines, J., Bagnall, A., 2015. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery* 29, 565–592.
 - [53] Ma, M., Fan, H., Kitani, K.M., 2016. Going deeper into first-person activity recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1894–1903.
 - [54] McNemar, Q., 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* 12, 153–157.
 - [55] Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A., 2020. End-to-end learning of visual representations from uncurated instructional videos, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9879–9889.
 - [56] Okawa, M., 2021. Time-series averaging and local stability-weighted dynamic time warping for online signature verification. *Pattern Recognition* 112, 107699.
 - [57] Petitjean, F., Ketterlin, A., Gançarski, P., 2011. A global averaging method for dynamic time warping, with applications to clustering. *Pattern recognition* 44, 678–693.
 - [58] Peyré, G., Cuturi, M., et al., 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning* 11, 355–607.
 - [59] Ren, Z., Lin, T., Feng, K., Zhu, Y., Liu, Z., Yan, K., 2023. A systematic review on imbalanced learning methods in intelligent fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*.
 - [60] Richard, A., Kuehne, H., Iqbal, A., Gall, J., 2018. Neuralnetwork-verbnet: A framework for weakly supervised video learning, in: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7386–7395.
 - [61] Ruiz, A.P., Flynn, M., Large, J., Middlehurst, M., Bagnall, A., 2021. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* 35, 401–449.
 - [62] Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing* 26, 43–49.
 - [63] Shen, Y., Wang, L., Elhamifar, E., 2021. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10156–10165.
 - [64] Su, B., Hua, G., 2019. Order-preserving optimal transport for distances between sequences. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 41, 2961–2974.
 - [65] Su, B., Wu, Y., 2019. Learning distance for sequences by learning a ground metric, in: *International Conference on Machine Learning*, PMLR. pp. 6015–6025.
 - [66] Su, B., Zhou, J., Wen, J.R., Wu, Y., 2021. Linear and deep order-preserving wasserstein discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 3123–3138.
 - [67] Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J., 2019. Coin: A large-scale dataset for comprehensive instructional video analysis, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1207–1216.
 - [68] Tang, Y., Song, Z., Zhu, Y., Yuan, H., Hou, M., Ji, J., Tang, C., Li, J., 2022. A survey on machine learning models for financial time series forecasting. *Neurocomputing* 512, 363–380.
 - [69] Theissler, A., Spinnato, F., Schlegel, U., Guidotti, R., 2022. Explainable ai for time series classification: a review, taxonomy and research directions. *IEEE Access*.
 - [70] Villani, C., 2009. Optimal transport. old and new, volume 338 of. *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*.
 - [71] Villani, C., 2021. Topics in optimal transportation. volume 58. American Mathematical Soc.
 - [72] Wu, W., Sun, W., Wu, Q.J., Yang, Y., Zhang, H., Zheng, W.L., Lu, B.L., 2020. Multimodal vigilance estimation using deep learning. *IEEE Transactions on Cybernetics* 52, 3097–3110.
 - [73] Xing, Z., Pei, J., Keogh, E., 2010. A brief survey on sequence classification. *ACM Sigkdd Explorations Newsletter* 12, 40–48.
 - [74] Yang, Y., Wu, Q.J., 2019. Features combined from hundreds of midlayers: Hierarchical networks with subnetwork nodes. *IEEE transactions on neural networks and learning systems* 30, 3313–3325.
 - [75] Yuan, J., Lin, Q., Zhang, W., Wang, Z., 2019. Locally slope-based dynamic time warping for time series classification, in: *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1713–1722.
 - [76] Zhang, W., Wu, Q.J., Yang, Y., 2023. Semisupervised manifold regularization via a subnetwork-based representation learning model. *IEEE transactions on cybernetics* 53, 6923–6936.
 - [77] Zhang, W., Wu, Q.J., Zhao, W.W., Deng, H., Yang, Y., 2022. Hierarchical one-class model with subnetwork for representation learning and outlier detection. *IEEE Transactions on Cybernetics*.
 - [78] Zhao, J., Itti, L., 2018. shapedtw: Shape dynamic time warping. *Pattern Recognition* 74, 171–184.
 - [79] Zhou, L., Xu, C., Corso, J., 2018. Towards automatic learning of procedures from web instructional videos, in: *Proceedings of the AAAI Conference on Artificial Intelligence*.
 - [80] Zhukov, D., Alayrac, J.B., Cinbis, R.G., Fouhey, D., Laptev, I., Sivic, J., 2019. Cross-task weakly supervised learning from instructional videos, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3537–3545.



Tung Doan received the B.S. degree in computer engineering from the Hanoi University of Science and Technology, in 2014, and the Ph.D. degree from the National Institute of Informatics, Japan, in 2021. He is currently a Lecturer at the Department of Computer Engineering, School of Information and Communication Technology, Hanoi University of Science and Technology. His current research interests include machine learning, optimal transport, and their applications on images and sequential data.



Tuan Phan is currently studying for his B.S. degree in Data Science and Artificial Intelligence at the Hanoi University of Science and Technology (HUST). His research interests include machine learning, optimal transport, and their applications on images and sequential data.



Phu Nguyen received the B.S. degree in Data Science and Artificial Intelligence from the Hanoi University of Science and Technology in 2023. His current research interests are optimal transport theory and its application in machine learning.



Khoat Than is currently an associate professor at Hanoi University of Science and Technology. He received Ph.D. degree from Japan Advanced Institute of Science and Technology in 2013. His recent research interests include deep generative models, continual learning, and learning theory.



Muriel Visani started her career as a research engineer at France Télécom R&D (Orange Labs), before embracing the academic path. She got her PhD in 2005 and, since 2006, she is an Associate Professor at La Rochelle University (France). Since 2023, she is detached at the French Military Center for Epidemiology and Public Health (CESPA). Her main research interests include, but are not limited to, machine learning and especially clustering or classification methods, mainly applied to image analysis, and possibly with user interaction.



Atsuhiko Takasu received his B.E., M.E., and Dr.Eng. in 1984, 1986, and 1989, respectively, from the University of Tokyo, Japan. He is a professor at the National Institute of Informatics, Japan. His research interests are data engineering and data mining. He is a member of the ACM, IEEE, IEICE, IPSJ, and JSAP.