Uncertainty-DTW for Time Series and Sequences

Lei Wang[⋆,†,§] and Piotr Koniusz[⋆,§,†]

[†]Australian National University [§]Data61/CSIRO
[§]firstname.lastname@data61.csiro.au

Abstract. Dynamic Time Warping (DTW) is used for matching pairs of sequences and celebrated in applications such as forecasting the evolution of time series, clustering time series or even matching sequence pairs in few-shot action recognition. The transportation plan of DTW contains a set of paths; each path matches frames between two sequences under a varying degree of time warping, to account for varying temporal intra-class dynamics of actions. However, as DTW is the smallest distance among all paths, it may be affected by the feature uncertainty which varies across time steps/frames. Thus, in this paper, we propose to model the so-called aleatoric uncertainty of a differentiable (soft) version of DTW. To this end, we model the heteroscedastic aleatoric uncertainty of each path by the product of likelihoods from Normal distributions, each capturing variance of pair of frames. (The path distance is the sum of base distances between features of pairs of frames of the path.) The Maximum Likelihood Estimation (MLE) applied to a path yields two terms: (i) a sum of Euclidean distances weighted by the variance inverse, and (ii) a sum of log-variance regularization terms. Thus, our uncertainty-DTW is the smallest weighted path distance among all paths, and the regularization term (penalty for the high uncertainty) is the aggregate of log-variances along the path. The distance and the regularization term can be used in various objectives. We showcase forecasting the evolution of time series, estimating the Fréchet mean of time series, and supervised/unsupervised few-shot action recognition of the articulated human 3D body joints.

Keywords: time series, aleatoric uncertainty, few-shot, actions

1 Introduction

Dynamic Time Warping (DTW) [6] is a method popular in forecasting the evolution of time series, estimating the Fréchet mean of time series, or classifying generally understood actions. The key property of DTW is its sequence matching transportation plan that allows any two sequences that are being matched to progress at different 'speeds' not only in the global sense but locally in the temporal sense. As DTW is non-differentiable, a differentiable 'soft' variant of DTW, soft-DTW [7], uses a soft-minimum function which enables backpropagation.

The role of soft-DTW is to evaluate the (relaxed) DTW distance between a pair of sequences $\boldsymbol{\Psi} \equiv [\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_\tau] \in \mathbb{R}^{d' \times \tau}$, $\boldsymbol{\Psi}' \equiv [\boldsymbol{\psi}'_1, ..., \boldsymbol{\psi}'_{\tau'}] \in \mathbb{R}^{d' \times \tau'}$ of lengths τ and τ', respectively. Under its transportation plan $\mathcal{A}_{\tau,\tau'}$, each path $\boldsymbol{\Pi} \in \mathcal{A}_{\tau,\tau'}$ is evaluated to

⋆ Equal contribution. Code: `https://github.com/LeiWangR/uDTW`.

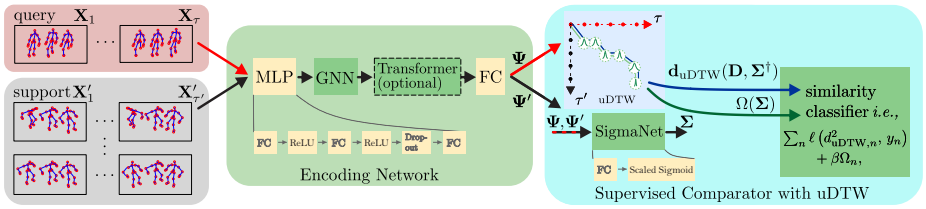


Fig. 1: Supervised few-shot action recognition of the articulated human 3D body joints with the uncertainty-DTW (uDTW). Frames from a query and support sequences are split into short-term temporal blocks $\mathbf{X}_1, ..., \mathbf{X}_\tau$ and $\mathbf{X}'_1, ..., \mathbf{X}'_{\tau'}$ of length M given stride S. We pass all skeleton coordinates via Encoding Network to obtain feature tensors $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}'$, which are directed to the Supervised Comparator with uDTW. For each query-support pair $(\boldsymbol{\Psi}_n, \boldsymbol{\Psi}'_n)$, uDTW computes the base-distance matrix $\mathbf{D}_n$ reweighted by uncertainty $\boldsymbol{\Sigma}_n^\dagger$ to compare $\tau \times \tau'$ blocks, and SigmaNet generates underlying block-wise uncertainty parameters $\boldsymbol{\Sigma}_n$. uDTW finds the warping path with the smallest distance, and returns its Ω_n penalty (uncertainty aggregated along the path).

ascertain the path distance, and the smallest distance is 'selected' by the soft minimum:

$$d^2_{\text{DTW}}(\boldsymbol{\Psi}, \boldsymbol{\Psi}') = \text{SoftMin}_\gamma \left([\langle \boldsymbol{\Pi}, \boldsymbol{D}(\boldsymbol{\Psi}, \boldsymbol{\Psi}') \rangle]_{\boldsymbol{\Pi} \in \mathcal{A}_{\tau,\tau'}} \right), \tag{1}$$

where $\text{SoftMin}_\gamma(\boldsymbol{\alpha}) = -\gamma \log \sum_i \exp(-\alpha_i/\gamma)$ is the soft minimum, $\gamma \geq 0$ controls its relaxation (hard *vs*. soft path selection), and $\boldsymbol{D} \in \mathbb{R}_+^{\tau \times \tau'} \equiv [d^2_{\text{base}}(\boldsymbol{\psi}_m, \boldsymbol{\psi}'_n)]_{(m,n) \in \mathcal{I}_\tau \times \mathcal{I}_{\tau'}}$ contains pair-wise distances between all possible pairings of frame-wise feature representations of sequences $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}'$, and $d^2_{\text{base}}(\cdot,\cdot)$ may be the squared Euclidean distance.

However, the path distance $\langle \boldsymbol{\Pi}, \boldsymbol{D}(\boldsymbol{\Psi}, \boldsymbol{\Psi}') \rangle$ of path $\boldsymbol{\Pi}$ ignores the observation uncertainty of frame-wise feature representations by simply relying on the Euclidean distances stored in $\boldsymbol{D}$. Thus, we resort to the notion of the so-called aleatoric uncertainty known from a non-exhaustive list of works about uncertainty [28, 18, 15, 14, 17].

Specifically, to capture the aleatoric uncertainty of the Euclidean distance (or regression, *etc*.), one should tune the observation noise parameter of sequences. Instead of the homoscedastic model (constant observation noise), we opt for the so-called heteroscedastic aleatoric uncertainty model (the observation noise may vary with each frame/sequence). To this end, we model each path distance by the product of likelihoods of Normal distributions (we also investigate other distributions in Appendix Sec. F).

Our (soft) uncertainty-DTW takes the following generalized form:

$$\begin{cases} d^2_{\text{uDTW}}(\boldsymbol{D}, \boldsymbol{\Sigma}^\dagger) = \text{SoftMin}_\gamma \Big(\underbrace{[\langle \boldsymbol{\Pi}, \boldsymbol{D} \odot \boldsymbol{\Sigma}^\dagger \rangle]_{\boldsymbol{\Pi} \in \mathcal{A}_{\tau,\tau'}}}_{\boldsymbol{w}} \Big) & (2) \\ \Omega(\boldsymbol{\Sigma}) = \text{SoftMinSel}_\gamma \Big(\boldsymbol{w}, [\langle \boldsymbol{\Pi}, \log \boldsymbol{\Sigma} \rangle]_{\boldsymbol{\Pi} \in \mathcal{A}_{\tau,\tau'}} \Big), & (3) \end{cases}$$

where $\boldsymbol{D} \equiv \boldsymbol{D}(\boldsymbol{\Psi}, \boldsymbol{\Psi}')$, $\boldsymbol{\Sigma} \equiv \boldsymbol{\Sigma}(\boldsymbol{\Psi}, \boldsymbol{\Psi}')$ and $\boldsymbol{\Sigma}^\dagger = \text{inv}(\boldsymbol{\Sigma})$,

where $\odot$ is the Hadamard product, $\boldsymbol{\Sigma}^{\dagger}(\boldsymbol{\Psi}, \boldsymbol{\Psi}')$ is the element-wise inverse of matrix $\boldsymbol{\Sigma} \in \mathbb{R}_{+}^{\tau \times \tau'} \equiv [\sigma^2(\boldsymbol{\psi}_m, \boldsymbol{\psi}'_n)]_{(m,n) \in \mathcal{I}_\tau \times \mathcal{I}_{\tau'}}$ which contains pair-wise variances between all possible pairings of frame-wise feature representations from sequences $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}'$. $\mathrm{SoftMin}_\gamma(\boldsymbol{\alpha}) = \sum_i \alpha_i \frac{\exp(-(\alpha_i - \mu_\alpha)/\gamma)}{\sum_j \exp(-(\alpha_j - \mu_\alpha)/\gamma)}$ with μ_α (the mean over coefficients of $\boldsymbol{\alpha}$) subtracted from each coefficient α_i to attain stability of the softmax (into which we feed $(\alpha_i - \mu_\alpha)$). Moreover, $\mathrm{SoftMinSel}_\gamma(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_i \beta_i \frac{\exp(-(\alpha_i - \mu_\alpha)/\gamma)}{\sum_j \exp(-(\alpha_j - \mu_\alpha)/\gamma)}$ is a soft-selector returning $(\beta_{i^*}\colon i^* = \arg\min_i \alpha_i)$ if γ approaches zero.

Eq. (2) yields the uncertainty-weighted time warping distance $d^2_{\mathrm{uDTW}}(\boldsymbol{D}, \boldsymbol{\Sigma}^{\dagger})$ between sequences $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}'$ because $\boldsymbol{D}$ and $\boldsymbol{\Sigma}^{\dagger}$ are both functions of $(\boldsymbol{\Psi}, \boldsymbol{\Psi}')$.

Eq. (3) provides the regularization penalty $\Omega(\boldsymbol{\Sigma})$ for sequences $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}'$ (as $\boldsymbol{\Sigma}$ is a function of $(\boldsymbol{\Psi}, \boldsymbol{\Psi}')$) which is the aggregation of log-variances along the path with the smallest distance, *i.e.*, path matrix $((\boldsymbol{\Pi}_{i^*} \in \{0,1\}^{\tau \times \tau'})\colon i^* = \arg\min_k w_k)$ if $\gamma = 0$, and vector $\boldsymbol{w}$ contains path-aggregated distances for all possible paths of the plan $\mathcal{A}_{\tau,\tau'}$.

Contributions. The celebrated DTW warps the matching path between a pair of sequences to recover the best matching distance under varying temporal within-class dynamics of each sequence. The recovered path, and the distance corresponding to that path, may be suboptimal if frame-wise (or block-wise) features contain noise (frames that are outliers, contain occlusions or large within-class object variations, *etc*.)

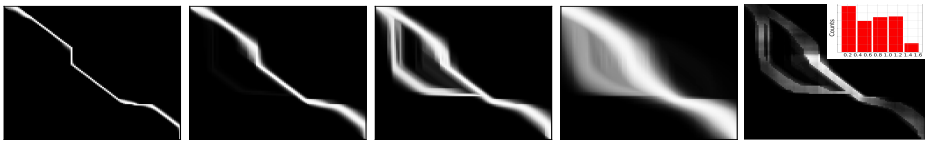
To this end, we propose several contributions:

i. We introduce the uncertainty-DTW, dubbed as uDTW, whose role is to take into account the uncertainty of in frame-wise (or block-wise) features by selecting the path which maximizes the Maximum Likelihood Estimation (MLE). The parameters (such as variance) of a distribution (*i.e.*, the Normal distribution) are thus used within MLE (and uDTW) to model the uncertainty.
ii. As pairs of sequences are often of different lengths, optimizing the free-form variable of variance is impossible. To that end, we equip each of our pipelines with SigmaNet, whose role is to take frames (or blocks) of sequences, and generate the variance end-to-end (the variance is parametrized by SigmaNet).
iii. We provide several pipelines that utilize uDTW for (1) forecasting the evolution of time series, (2) estimating the Fréchet mean of time series, (3) supervised few-shot action recognition, and (4) unsupervised few-shot action recognition.

Notations. $\mathcal{I}_\tau$ is the index set $\{1, 2, ..., \tau\}$. Concatenation of α_i into a vector $\boldsymbol{\alpha}$ is denoted by $[\alpha_i]_{i \in \mathcal{I}_I}$. Concatenation of α_{ij} into matrix $\mathbf{A}$ is denoted by $[\alpha_{ij}]_{(i,j) \in \mathcal{I}_I \times \mathcal{I}_J}$. Dot-product between two matrices equals the dot-product of vectorized $\boldsymbol{\Pi}$ and $\boldsymbol{D}$, that is $\langle \boldsymbol{\Pi}, \boldsymbol{D} \rangle \equiv \langle \mathrm{vec}(\boldsymbol{\Pi}), \mathrm{vec}(\boldsymbol{D}) \rangle$. Mathcal symbols are sets, *e.g.*, $\mathcal{A}$ is a transportation plan, capitalized bold symbols are matrices, *e.g.*, $\boldsymbol{D}$ is the distance matrix, lowercase bold symbols are vectors, *e.g.*, $\boldsymbol{w}$ contains weighted distances. Regular fonts are scalars.

1.1 Similarity learning with uDTW

In further chapters, based on the distance in Eq. (2) and the regularization term in Eq. (3), we define specific loss functions for several problems such as forecasting the evolution of time series, clustering time series or even matching sequence pairs in few-shot



(a) sDTW$_{\gamma=0.01}$ (b) sDTW$_{\gamma=0.1}$ (c) uDTW$_{\gamma=0.01}$ (d) uDTW$_{\gamma=0.1}$ (e) uDTW uncert.

Fig. 2: Plots (a)-(d) show paths of sDTW and uDTW (in white) for a pair of sequences. We power-normalized pixels of plots (by the power of 0.1) to see also darker paths better. With higher γ that controls softness, in (b) & (d) more paths become 'active' (fuzzy effect). In (c), uDTW has two possible routes *vs*. sDTW (a) due to uncertainty modeling. In (e), we visualise uncertainty $\boldsymbol{\Sigma}$. We binarize plot (c) and multiply it by the $\boldsymbol{\Sigma}$ to display uncertainty values on the path (white pixels = high uncertainty). The middle of the main path is deemed uncertain, which explains why an additional path merges in that region with the main path. See also the histogram of values of $\boldsymbol{\Sigma}$.

action recognition. Below is an example of a generic similarity learning loss:

$$\arg\min_{\mathcal{P}} \sum_n \ell\left(d^2_{\text{uDTW}}(\boldsymbol{D}(\boldsymbol{\Psi}_n, \boldsymbol{\Psi}'_n), \boldsymbol{\Sigma}^\dagger(\boldsymbol{\Psi}_n, \boldsymbol{\Psi}'_n)),\ \delta_n\right) + \beta\Omega(\boldsymbol{\Sigma}(\boldsymbol{\Psi}_n, \boldsymbol{\Psi}'_n)), \tag{4}$$

or

$$\arg\min_{\mathcal{P}, \boldsymbol{\Sigma}>0} \sum_n \ell\left(d^2_{\text{uDTW}}(\boldsymbol{D}(\boldsymbol{\Psi}_n, \boldsymbol{\Psi}'_n), \boldsymbol{\Sigma}^\dagger),\ \delta_n\right) + \beta\Omega(\boldsymbol{\Sigma}), \tag{5}$$

where $\boldsymbol{\Psi}_n = f(\mathbf{X}_n; \mathcal{P})$ and $\boldsymbol{\Psi}'_n = f(\mathbf{X}'_n; \mathcal{P})$ are obtained from some backbone encoder $f(\cdot; \mathcal{P})$ with parameters $\mathcal{P}$ and $(\mathbf{X}_n, \mathbf{X}'_n) \in \mathcal{X}$ is a sequence pair to compare with the similarity label $\delta_n \in \{0, 1\}$ (where $\delta_n = 0$ if $y_n = y'_n$ and $\delta_n = 1$ otherwise), (y_n, y'_n) is a pair of class labels for $(\boldsymbol{\Psi}_n, \boldsymbol{\Psi}'_n)$, and $\beta \geq 0$ controls the penalty for high matching uncertainty. Figure 2 illustrates the impact of uncertainty on uDTW.

Note that minimizing Eq. (5) w.r.t. $(\mathcal{P}, \boldsymbol{\Sigma})$ assumes that $\boldsymbol{\Sigma} \in \mathbb{R}_+^{\tau \times \tau'}$ is a free variable to minimize over (derivation in Section 1.2). However, as sequence pairs vary in length, *i.e.*, $\tau \neq \tau'$, optimizing one global $\boldsymbol{\Sigma}$ is impossible (its size changes). Thus, for problems we tackle, we minimize loss functions with the distance/penalty in Eq. (4) and (5) where $\boldsymbol{\Sigma}$ is parametrized by $(\boldsymbol{\Psi}_n, \boldsymbol{\Psi}'_n)$:

$$d^2_{\text{uDTW}\bullet}(\boldsymbol{\Psi}, \boldsymbol{\Psi}') \equiv d^2_{\text{uDTW}}(\boldsymbol{D}(\boldsymbol{\Psi}, \boldsymbol{\Psi}'), \boldsymbol{\Sigma}^\dagger(\boldsymbol{\Psi}, \boldsymbol{\Psi}')), \tag{6}$$

$$\Omega_\bullet(\boldsymbol{\Psi}, \boldsymbol{\Psi}') \equiv \Omega(\boldsymbol{\Sigma}(\boldsymbol{\Psi}, \boldsymbol{\Psi}')). \tag{7}$$

To that end, we devise a small MLP unit $\sigma(\cdot; \mathcal{P}_\sigma)$ or $\sigma(\cdot, \cdot; \mathcal{P}_\sigma)$ and obtain:

$$\boldsymbol{\Sigma} = 0.5 \cdot [(\sigma^2(\boldsymbol{\psi}_m; \mathcal{P}_\sigma) + \sigma^2(\boldsymbol{\psi}'_n; \mathcal{P}_\sigma))]_{(m,n) \in \mathcal{I}_\tau \times \mathcal{I}_{\tau'}} \tag{8}$$

or

$$\boldsymbol{\Sigma}' = [\sigma^2(\boldsymbol{\psi}_m, \boldsymbol{\psi}'_n; \mathcal{P}_\sigma)]_{(m,n) \in \mathcal{I}_\tau \times \mathcal{I}_{\tau'}}, \tag{9}$$

where Eq. (8) uses additive variance terms generated for individual frames $\boldsymbol{\psi}_m$ and $\boldsymbol{\psi}'_n$, whereas (9) is a jointly generated variance for $(\boldsymbol{\psi}_m, \boldsymbol{\psi}'_n)$.

1.2 Derivation of uDTW

We proceed by modeling an arbitrary path $\boldsymbol{\Pi}_i$ from the transportation plan of $\mathcal{A}_{\tau,\tau'}$ as the following Maximum Likelihood Estimation (MLE) problem:

$$\underset{\{\sigma_{mn}\}_{(m,n)\in\boldsymbol{\Pi}_i}}{\arg\max} \prod_{(m,n)\in\boldsymbol{\Pi}_i} p(\|\boldsymbol{\psi}_m - \boldsymbol{\psi}'_n\|, \sigma^2_{mn}), \tag{10}$$

where p may be some arbitrary distribution, σ are distribution parameters, and $\|\cdot\|$ is an arbitrary norm. For the Normal distribution $\mathcal{N}$ which relies on the squared Euclidean distance $\|\cdot\|_2^2$, we have:

$$\underset{\{\sigma_{mn}\}_{(m,n)\in\boldsymbol{\Pi}_i}}{\arg\max} \prod_{(m,n)\in\boldsymbol{\Pi}_i} \mathcal{N}(\boldsymbol{\psi}_m; \boldsymbol{\psi}'_n, \sigma^2_{mn}) \tag{11}$$

$$= \underset{\{\sigma_{mn}\}_{(m,n)\in\boldsymbol{\Pi}_i}}{\arg\max} \log \prod_{(m,n)\in\boldsymbol{\Pi}_i} \frac{1}{(2\pi)^{\frac{d'}{2}}\sigma^{d'}} \exp\Big(-\frac{\|\boldsymbol{\psi}_m - \boldsymbol{\psi}'_n\|_2^2}{\sigma^2_{mn}}\Big) \tag{12}$$

$$= \underset{\{\sigma_{mn}\}_{(m,n)\in\boldsymbol{\Pi}_i}}{\arg\max} \sum_{(m,n)\in\boldsymbol{\Pi}_i} -\frac{d'}{2}\log(2\pi) - d'\log(\sigma) - \frac{\|\boldsymbol{\psi}_m - \boldsymbol{\psi}'_n\|_2^2}{\sigma^2_{mn}} \tag{13}$$

$$= \underset{\{\sigma_{mn}\}_{(m,n)\in\boldsymbol{\Pi}_i}}{\arg\min} \sum_{(m,n)\in\boldsymbol{\Pi}_i} d'\log(\sigma) + \frac{\|\boldsymbol{\psi}_m - \boldsymbol{\psi}'_n\|_2^2}{\sigma^2_{mn}}, \tag{14}$$

where d' is the length of feature vectors $\boldsymbol{\psi}$. Having recovered uncertainty parameters $\{\sigma_{mn}\}_{(m,n)\in\boldsymbol{\Pi}_i}$, we obtain a combination of penalty terms and reweighted squared Euclidean distances:

$$\beta\Omega_{\boldsymbol{\Pi}_i} + d^2_{\boldsymbol{\Pi}_i} = \sum_{(m,n)\in\boldsymbol{\Pi}_i} \beta\log(\sigma_{mn}) + \frac{\|\boldsymbol{\psi}_m - \boldsymbol{\psi}'_n\|_2^2}{\sigma^2_{mn}}, \tag{15}$$

where $\beta \geq 0$ (generally $\beta \neq d'$) adjusts the penalty for large uncertainty. Separating the uncertainty penalty $\log(\sigma_{mn})$ from the uncertainty-weighted distance (both aggregated along path $\boldsymbol{\Pi}_i$) yields:

$$\begin{cases} d^2_{\boldsymbol{\Pi}_i} = \big\langle \boldsymbol{\Pi}_i, \boldsymbol{D}(\boldsymbol{\Psi}, \boldsymbol{\Psi}') \odot \boldsymbol{\Sigma}^{\dagger} \big\rangle \\ \Omega_{\boldsymbol{\Pi}_i} = \langle \boldsymbol{\Pi}_i, \log\boldsymbol{\Sigma} \rangle \,, \end{cases} \tag{16}$$

where $\boldsymbol{D} \in \mathbb{R}_+^{\tau\times\tau'} \equiv \big[\frac{d_2^2(\boldsymbol{\psi}_m, \boldsymbol{\psi}'_n)}{\sigma^2_{mn}}\big]_{(m,n)\in\mathcal{I}_\tau\times\mathcal{I}_{\tau'}}$ and $\boldsymbol{\Sigma} \in \mathbb{R}_+^{\tau\times\tau'} \equiv [\sigma^2_{mn}]_{(m,n)\in\mathcal{I}_\tau\times\mathcal{I}_{\tau'}}$. Derivations for other distributions, *i.e.*, Laplace or Cauchy, follow the same reasoning.

2 Related Work

Different flavors of Dynamic Time Warping. DTW [6], which seeks a minimum cost alignment between time series is computed by dynamic programming in quadratic time, is not differentiable and is known to get trapped in bad local minima. In contrast, soft-DTW (sDTW) [7] addresses the above issues by replacing the minimum over alignments with a soft minimum, which has the effect of inducing a 'likelihood' field over all

possible alignments. However, sDTW has been successfully applied in many computer vision tasks including audio/music score alignment [31], action recognition [39, 4], and end-to-end differentiable text-to-speech synthesis [10]. Despite its successes, sDTW has some limitations: (i) it can be negative when used as a loss (ii) it may still get trapped in bad local minima. Thus, soft-DTW divergences (sDTW div.) [3], inspired by sDTW, attempts to overcome such issues.

Other approaches inspired by DTW have been used to improve the inference or adapt to modified or additional constraints, *i.e.*, OPT [38] and OWDA [40] treat the alignment as the optimal transport problem with temporal regularization. TAP [39] directly predicts the alignment through a lightweight CNN, thus is does not follow a principled transportation plan, and is not guaranteed to find a minimum cost path.

Our uDTW differs from these methods in that the transportation plan is executed under the uncertainty estimation, thus various feature-level noises and outliers are less likely to lead to the selection of a sub-optimal cost path.

Alignment-based time series problems. Distance between sequences plays an important role in time series retrieval [40], forecasting [7, 3], classification [7, 3, 9, 49], clustering [12, 35], *etc*. Various temporal nuisance noises such as initial states, different sampling rates, local distortions, and execution speeds make the measurement of distance between sequences difficult. To tackle these issues, typical feature-based methods use RNNs to encode sequences and measure the distance between corresponding features [34]. Other existing methods [43, 45, 20] either encode each sequence into features that are invariant to temporal variations [1, 26] or adopt alignment for temporal correspondence calibration [38]. However, none of these methods is modeling the aleatoric uncertainty. As we model it along the time warping path, the observation noise may vary with each frame or block.

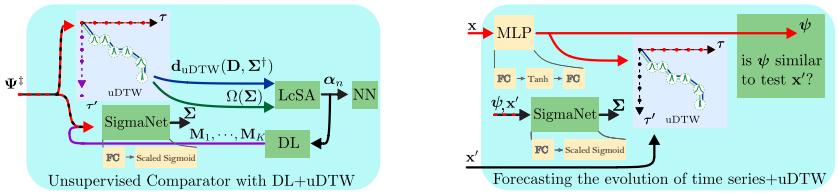
Few-shot action recognition. Most existing few-shot action recognition methods [44, 47, 46] follow the metric learning paradigm. Signal Level Deep Metric Learning [30] and Skeleton-DML [29] one-shot FSL approaches encode signals into images, extract features using a deep residual CNN and apply multi-similarity miner losses. TAEN [2] and FAN [41] encode actions into representations and apply vector-wise metrics.

Most methods identify the importance of temporal alignment for handling the non-linear temporal variations, and various alignment-based models are proposed to compare the sequence pairs, *e.g.*, permutation-invariant spatial-temporal attention reweighted distance in ARN [50], a variant of DTW used in OTAM [4], temporal attentive relation network [32], a two-stage temporal alignment network (TA2N) [22], a temporal CrossTransformer [33], a learnable sequence matching distance called TAP [39].

In all cases, temporal alignment is a well-recognized tool, however lacking the uncertainty modeling, which impacts the quality of alignment. Such a gap in the literature inspires our work on uncertainty-DTW.

3 Pipeline Formulations

Below we provide our several pipeline formulations for which uDTW is used as an indispensable component embedded with the goal of measuring the distance for warped paths under uncertainty.



(a) (b)

Fig. 3: In (a) is the unsupervised comparator for unsupervised few-shot action recognition. The unsupervised head is wired with the Encoding Network from Figure 1, and trained from scratch without labels. In (b) is the pipeline for forecasting the evolution of time series (a.k.a. multistep-ahead prediction).

3.1 Few-shot Action Recognition

For both supervised and unsupervised few-shot pipelines, we employ the Encoder Network (EN) and the Supervised Comparator (similarity learning) as in Figure 1, or Unupervised Comparator (based on dictionary learning) as in Figure 3a.

Encoding Network (EN). Our EN contains a simple 3-layer MLP unit (FC, ReLU, FC, ReLU, Dropout, FC), GNN, with transformer [11] and FC. The MLP unit takes M neighboring frames, each with J skeleton body joints given by Cartesian coordinates (x, y, z), forming one temporal block[1]. In total, depending on stride S, we obtain some τ temporal blocks (each block captures the short temporal dependency), whereas the long temporal dependency will be modeled by uDTW. Each temporal block is encoded by the MLP into a $d \times J$ dimensional feature map. Subsequently, query feature maps of size τ and support feature maps of size τ' are forwarded to a simple linear GNN model, and transformer, and an FC layer, which returns $\boldsymbol{\Psi} \in \mathbb{R}^{d' \times \tau}$ query feature maps and $\boldsymbol{\Psi}' \in \mathbb{R}^{d' \times \tau'}$ support feature maps. Such encoded feature maps are passed to the Supervised Comparator with uDTW.

Specifically, let support maps $\boldsymbol{\Psi}' \equiv [f(\boldsymbol{X}'_1; \mathcal{P}), ..., f(\boldsymbol{X}'_{\tau'}; \mathcal{P})]$ and query maps $\boldsymbol{\Psi} \equiv [f(\boldsymbol{X}_1; \mathcal{P}), ..., f(\boldsymbol{X}_{\tau}; \mathcal{P})]$ (where $\boldsymbol{\Psi}, \boldsymbol{\Psi}' \in \mathbb{R}^{d' \times \tau}$), for query and support frames per block $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^{3 \times J \times M}$. We define $f(\mathbf{X}; \mathcal{P}) = \text{FC}(\text{Transf}(\text{S}^2\text{GC}(\text{MLP}(\mathbf{X}; \mathcal{P}_{MLP}); \mathcal{P}_{S^2GC}); \mathcal{P}_{Transf}); \mathcal{P}_{FC})$, where $\mathcal{P} \equiv [\mathcal{P}_{MLP}, \mathcal{P}_{S^2GC}, \mathcal{P}_{Transf}, \mathcal{P}_{FC}, \mathcal{P}_{SN}]$ is the set of parameters of EN, where $\mathcal{P}_{SN}$ are parameters of SigmaNet, and S^2GC is a Simple Spectral Graph Convolution (S^2GC) [51] whose details are in Sec. H.3 of the Appendix.

Supervised Few-shot Action Recognition. For the N-way Z-shot problem, we have one query feature map and $N \times Z$ support feature maps per episode. We form a mini-batch containing B episodes. We have query feature maps $\{\boldsymbol{\Psi}_b\}_{b \in \mathcal{I}_B}$ and support feature maps $\{\boldsymbol{\Psi}'_{b,n,z}\}_{b \in \mathcal{I}_B, n \in \mathcal{I}_N, z \in \mathcal{I}_Z}$. Moreover, $\boldsymbol{\Psi}_b$ and $\boldsymbol{\Psi}'_{b,1,:}$ share the same class (drawn from N classes per episode), forming the subset $C^\ddagger \equiv \{c_1, ..., c_N\} \subset \mathcal{I}_C \equiv \mathcal{C}$. To be precise, labels $y(\boldsymbol{\Psi}_b) = y(\boldsymbol{\Psi}'_{b,1,z}), \forall b \in \mathcal{I}_B, z \in \mathcal{I}_Z$ while $y(\boldsymbol{\Psi}_b) \neq y(\boldsymbol{\Psi}'_{b,n,z}), \forall b \in \mathcal{I}_B, n \in \mathcal{I}_N \setminus \{1\}, z \in \mathcal{I}_Z$. Thus the similarity label $\delta_1 = 0$, whereas $\delta_{n \neq 1} = 1$. Note that the selection of $C^\ddagger$ per episode is random. For the N-way Z-shot protocol, the Supervised

[1] We use temporal blocks as they were shown more robust than frame-wise FSAR [50] models.

Comparator is minimized w.r.t. $\mathcal{P}$ ($\boldsymbol{\Psi}_b$ and $\boldsymbol{\Psi}'$ depend on $\mathcal{P}$) as:

$$\arg\min_{\mathcal{P}} \sum_{b\in\mathcal{I}_B} \sum_{n\in\mathcal{I}_N} \sum_{z\in\mathcal{I}_Z} \left(d^2_{\text{uDTW}\bullet}(\boldsymbol{\Psi}_b, \boldsymbol{\Psi}'_{b,n,z}) - \delta_n\right)^2 + \beta\Omega_\bullet(\boldsymbol{\Psi}_b, \boldsymbol{\Psi}'_{b,n,z}). \tag{17}$$

Unsupervised Few-shot Action Recognition. Below we propose a very simple unsupervised variant with so-called Unsupervised Comparator. The key idea is that with uDTW, invariant to local temporal speed changes can be used to learn a dictionary which, with some dictionary coding method should outperform at reconstructing the sequences. This means we can learn an unsupervised comparator by projecting sequences onto the dictionary space. To this end, let the protocol remain as for the supervised few-shot learning with the exception that class labels are not used during training, and only support images in testing are labeled for sake of evaluation the accuracy by deciding which support representation each query is the closest to in the nearest neighbor sense.

Firstly, in each training episode, we combine the query sequences $\boldsymbol{\Psi}_b$ with the support sequences $\boldsymbol{\Psi}'_{b,n,z}$ into episode sequences denoted as $\boldsymbol{\Psi}^\ddagger_{b,n}$ where $b\in\mathcal{I}_B$ enumerates over B episodes, and $n\in\mathcal{I}_{(N\cdot Z+1)}$. For the feature coding, we use Locality-constrained Soft Assignment (LCSA) [25, 19, 21] and a simple dictionary update based on the least squares computation.

For each episode $b\in\mathcal{I}_B$, we iterate over the following three steps:

i. The LCSA coding step which expresses each $\boldsymbol{\Psi}^\ddagger_{b,n}$ as $\boldsymbol{\alpha}_{b,n}\in\mathbb{R}^K_+$ that assign $\boldsymbol{\Psi}^\ddagger_{b,n}$ into a dictionary with K sequences $\boldsymbol{M}_1, ..., \boldsymbol{M}_K\in\mathbb{R}^{d'\times\tau'}$ (dictionary anchors):

$$\forall_{k,n},\ \alpha_{k,b,n} = \begin{cases} \dfrac{\exp\left(-\frac{1}{\gamma'}d^2_{\text{uDTW}\bullet}\left(\boldsymbol{\Psi}^\ddagger_{b,n}, \boldsymbol{M}_k\right)\right)}{\sum\limits_{l\in\mathcal{M}(\boldsymbol{\Psi}^\ddagger_{b,n};K')} \exp\left(-\frac{1}{\gamma'}d^2_{\text{uDTW}\bullet}\left(\boldsymbol{\Psi}^\ddagger_{b,n}, \boldsymbol{M}_l\right)\right)} & \text{if } \boldsymbol{M}_k\in\mathcal{M}\left(\boldsymbol{\Psi}^\ddagger_{b,n}; K'\right), \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where $0 < K' \leq K$ is a subset size for K' nearest anchors of $\boldsymbol{\Psi}^\ddagger_{b,n}$ retrieved by operation $\mathcal{M}(\boldsymbol{\Psi}^\ddagger_{b,n}; K')$ (based on uDTW) from $\boldsymbol{M}_1, ..., \boldsymbol{M}_K$, τ' is set to the mean of τ (over training set), and $\gamma'=0.7$ is a so-called smoothing factor;

ii. The dictionary update step updates $\boldsymbol{M}_1, ..., \boldsymbol{M}_K$ given $\boldsymbol{\alpha}_{b,n}$ from Eq. (18):

`for i=1,...,dict_iter:`

$$\forall_k,\ \boldsymbol{M}_k := \boldsymbol{M}_k - \lambda_{\text{DL}} \sum_{n=1}^{NZ+1} \nabla_{\boldsymbol{M}_k} d^2_{\text{uDTW}\bullet}\left(\boldsymbol{\Psi}^\ddagger_{b,n}, \sum_{l=1}^{K} \alpha_{l,b,n}\boldsymbol{M}_l\right), \tag{19}$$

where `dict_iter` is set to 10 and $\lambda_{\text{DL}}=0.001$;

iii. The main loss for the Feature Encoder update step is given as ($\lambda_{\text{EN}}=0.001$):

$$\mathcal{P} := \mathcal{P} - \lambda_{\text{EN}} \sum_{n=1}^{NZ+1} \nabla_{\mathcal{P}} d^2_{\text{uDTW}\bullet}\left(\boldsymbol{\Psi}^\ddagger_{b,n}, \boldsymbol{M}'\right) + \beta\Omega_\bullet\left(\boldsymbol{\Psi}^\ddagger_{b,n}, \boldsymbol{M}'\right), \tag{20}$$

$$\text{where } \boldsymbol{M}' = \sum_{l=1}^{K} \alpha_{l,b,n}\boldsymbol{M}_l.$$

During testing, we use the learnt dictionary, pass new support and query sequences via Eq. (18) and obtain $\boldsymbol{\alpha}$ codes. Subsequently, we compare the LCSA code of the query sequence with LCSA codes of support sequences via the histogram intersection kernel. The closest match in the support set determines the test label of the query sequence.

3.2 Time Series Forecasting and Classification

One of key applications of DTW and sDTW is learning with time series, including forecasting the evolution of time series as in Figure 3b and time series classification.

Forecasting the Evolution of Time Series. Let $\mathbf{x} \in \mathbb{R}^t$ and $\mathbf{x}' \in \mathbb{R}^{\tau-t}$ be the training and testing parts of one time series corresponding to timesteps $1, ..., t$ and $t+1, ..., \tau$, respectively. The goal is to learn encoder $f(\mathbf{x}; \mathcal{P}) \in \mathbb{R}^{\tau-t}$ which will be able to take $\mathbf{x}$ as input, learn to translate it to $\mathbf{x}'$. Figure 3b show the full pipeline. We took the Encoding Network from the original soft-DTW pipeline [7]. Our training objective is:

$$\arg\min_{\mathcal{P}} \sum_{n \in \mathcal{I}_N} d^2_{\text{uDTW}^\bullet}(\boldsymbol{\psi}_n, \mathbf{x}'_n) + \beta \Omega_\bullet(\boldsymbol{\psi}_n, \mathbf{x}'_n), \tag{21}$$

where $\boldsymbol{\psi} = f(\mathbf{x}; \mathcal{P})$ and N is the number of training time series, $\mathcal{P} \equiv [\mathcal{P}_{MLP}, \mathcal{P}_{SN}]$ is the set of parameters of EN and SigmaNet. In order to obtain $\boldsymbol{\Sigma}$, vectors $\boldsymbol{\psi}$ and $\mathbf{x}'$ are passed via SigmaNet. After training, at the test time, for a previously unseen testing sample $\mathbf{x}$, $f(\cdot)$ has to predict the remaining part of the time series given by $\mathbf{x}'$.

Time Series Classification. Below we follow the setting for this classical task according to the original soft-DTW paper [7], and define the **nearest centroid** classifier. We estimate the Fréchet mean of training time series of each class separately. We do not use any Encoding Network but the raw features. Let $\mathbf{x} \in \mathbb{R}^\tau$ be training samples and $\boldsymbol{\mu} \in \mathbb{R}^{\tau'}$ be class prototypes (τ' is set to average of τ across all classes). We have:

$$\forall_c, \ \arg\min_{\mathcal{P}} \sum_{n \in \mathcal{I}_{N_c}} d^2_{\text{uDTW}^\bullet}(\mathbf{x}_n, \boldsymbol{\mu}_c) + \beta \Omega_\bullet(\mathbf{x}_n, \boldsymbol{\mu}_c), \tag{22}$$

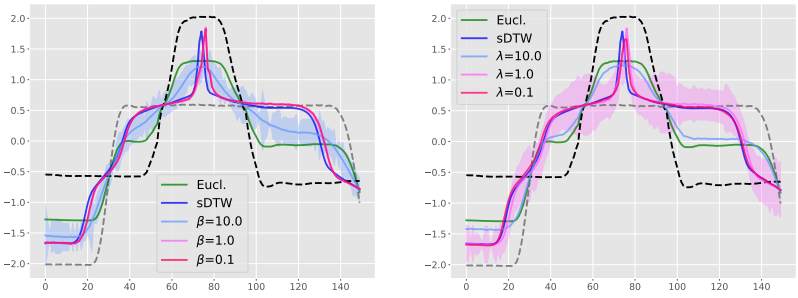
where N_c is the number of samples for class $c \in \mathcal{I}_C$ and $\mathcal{P} \equiv [\mathcal{P}_{SN}, \boldsymbol{\mu}_c]$. During testing, we apply $\arg\min_{c \in \mathcal{I}_C} d^2_{\text{uDTW}^\bullet}(\mathbf{x}, \boldsymbol{\mu}_c) + \beta \Omega_\bullet(\mathbf{x}, \boldsymbol{\mu}_c)$ for $\mathbf{x}$ to find its nearest neighbor and label it. The variances of $\mathbf{x}$ are recovered through SigmaNet while variances of $\boldsymbol{\mu}_c$ were obtained during training (adding both yields $\boldsymbol{\Sigma}$ of testing sample). As in soft-DTW paper [7], we use uDTW to directly find the **nearest neighbor** of $\mathbf{x}$ across training samples to label $\mathbf{x}$ (for uncertainty, we use SigmaNet from the nearest centroid task).

4 Experiments

Below we apply uDTW in several scenarios such as (i) forecasting the evolution of time series, (ii) clustering/classifying time series, (iii) supervised few-shot action recognition, and (iv) unsupervised few-shot action recognition.

Datasets. The following datasets are used in our experiments:

i. *UCR* archive [8] is a dataset for time series classification archive. This dataset contains a wide variety of fields (astronomy, geology, medical imaging) and lengths, and can be used for time series classification/clustering and forecasting tasks.



(a) β (where $\lambda=1$) (b) λ (where $\beta=10$)

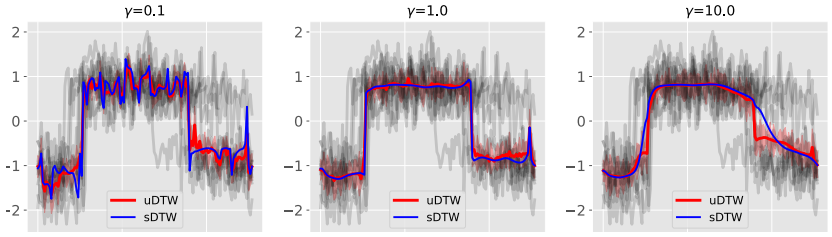
Fig. 4: Interpolation between two time series (grey and black dashed lines) on the Gun Point dataset. We compute the barycenter by solving $\arg\min_{\boldsymbol{\mu},\boldsymbol{\sigma}_\mu} \sum_{n=1}^{2} d^2_{\text{uDTW}}\big(\boldsymbol{D}, \boldsymbol{\Sigma}^\dagger\big) + \beta\Omega\big(\boldsymbol{\Sigma}\big) + \lambda\Omega'\big(\boldsymbol{\Sigma}\big)$ where $\boldsymbol{D}=(\mathbf{x}_n\mathbf{1}^\top-\mathbf{1}\boldsymbol{\mu}^\top)^2$ and $\boldsymbol{\Sigma}=\mathbf{1}\mathbf{1}^\top+\mathbf{1}\boldsymbol{\sigma}_\mu^\top$ where $\mathbf{x}_n$ is the given n-th time series. $\beta\geq 0$ controls the penalty for high matching uncertainty, Ω' is defined as in Eq. (3) but element-wise $\log\boldsymbol{\Sigma}$ is replaced by element-wise $(\boldsymbol{\Sigma}-1)^2$ so that $\lambda\geq 0$ favours uncertainty to remain close to one. β and λ control the uncertainty estimation and yield different barycenters than the Euclidean (green color) and sDTW (blue color) distances. As Ω and Ω' act similar, we only use Ω in our experiments.

ii. *NTU RGB+D (NTU-60)* [36] contains 56,880 video sequences and over 4 million frames. NTU-60 has variable sequence lengths and high intra-class variations.
iii. *NTU RGB+D 120 (NTU-120)* [24], an extension of NTU-60, contains 120 action classes (daily/health-related), and 114,480 RGB+D video samples captured with 106 distinct human subjects from 155 different camera viewpoints.
iv. *Kinetics* [16] is a large-scale collection of 650,000 video clips that cover 400/600/700 human action classes. It includes human-object interactions such as *playing instruments*, as well as human-human interactions such as *shaking hands* and *hugging*. We follow approach [48] and use the estimated joint locations in the pixel coordinate system as the input to our pipeline. As OpenPose produces the 2D body joint coordinates and Kinetics-400 does not offer multiview or depth data, we use a network of Martinez et al. [27] pre-trained on Human3.6M [5], combined with the 2D OpenPose output to estimate 3D coordinates from 2D coordinates. The 2D OpenPose and the latter network give us (x, y) and z coordinates, respectively.

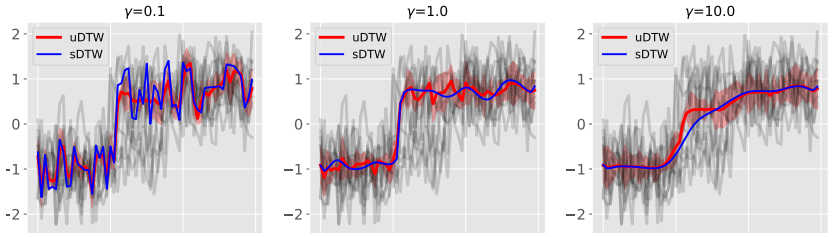
4.1 Fréchet Mean of Time Series

Below, we visually inspect the Fréchet mean for the Euclidean, sDTW and our uDTW distance, respectively.

Experimental setup. We follow the protocol of soft-DTW paper [7]. For each dataset in UCR, we choose a class at random, pick 10 time series from the selected class to compute its barycenter. We use L-BFGS [23] to minimise the proposed uDTW barycenter objective. We set the maximum number of iterations to 100.



(a) CBF



(b) Synthetic Control

Fig. 5: Comparison of barycenter based on sDTW or uDTW on CBF and Synthetic Control. We visualize uncertainty around the barycenters in red color for uDTW. Our uDTW generates reasonable barycenters even when higher γ values are used, *e.g.*, $\gamma = 10.0$. Higher γ value leads to smooth barycenter but introducing higher uncertainty.

Qualitative results. We first perform averaging between two time series (Figure 4). We notice that averaging under the uDTW yields substantially different results than those obtained with the Euclidean and sDTW geometry.

Figure 5 shows the barycenters obtained using sDTW and our uDTW. We observe that our uDTW yields more reasonable barycenters than sDTW even when large γ are used, *e.g.*, for $\gamma = 10$ (right column of plots in Figure 5), the change points of red curve look sharper. We also notice that both uDTW and sDTW with low smoothing parameter $\gamma = 0.1$ can get stuck in some bad local minima, but our uDTW has fewer sharp peaks compared with sDTW (barycenters of uDTW are improved by the uncertainty measure). Moreover, higher γ values smooth the barycenter but introducing higher uncertainty (see uncertainty visualization around the barycenters by comparing, *e.g.*, $\gamma = 0.1$ *vs.* $\gamma = 10.0$). With $\gamma = 1$, the barycenters of sDTW and uDTW match well with the time series. More visualizations can be found in Appendix Sec. D.

4.2 Classification of Time Series

In this section, we devise the nearest neighbor and nearest centroid classifiers [13] with uDTW, as detailed in Section 3. For the K-nearest neighbor classifier, we used softmax for the final decision. See Appendix Sec. H.4 for details.

Experimental setup. We use 50% of the data for training, 25% for validation and 25% for testing. We report $K = 1, 2$ and 3 for the nearest neighbor classifier.

Table 1: Classification accuracy (mean±std) on UCR archive by the nearest neighbor and the nearest centroid classifiers. In the column we indicate which distance was used for computing the class prototypes. K is the number of nearest neighbors in this context.

	Nearest neighbor			**Nearest centroid**
	$K=1$	$K=3$	$K=5$	
Euclidean	71.2±17.5	72.3±18.1	73.0±16.7	61.3±20.1
DTW [6]	74.2±16.6	75.0±17.0	75.4±15.8	65.9±18.8
sDTW [7]	76.2±16.6	77.2±15.9	78.0±16.5	70.5±17.6
sDTW div. [3]	78.6±16.2	79.5±16.7	80.1±16.5	70.9±17.8
uDTW	80.0±15.0	81.2±17.8	83.3±16.2	72.2±16.0

Quantitative results. Table 1 shows a comparison of our uDTW versus Euclidean, DTW, sDTW, and sDTW div. Unsurprisingly, the use of uDTW for barycenter computation improves the accuracy of the nearest centroid classifier, and it outperforms sDTW div. by $\sim 2\%$. Moreover, uDTW boosts results for the nearest neighbor classifier given K=1, 2 and 3 by 1.4%, 1.7% and 3.2%, respectively, compared to sDTW div.

4.3 Forecasting the Evolution of Time Series

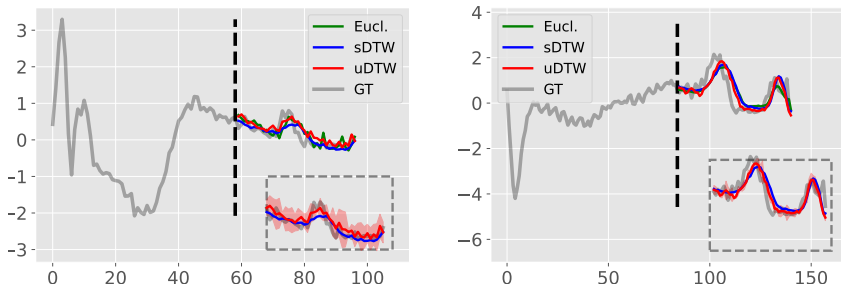
Experimental setup. We use the training and test sets pre-defined in the UCR archive. For both training and test, we use the first 60% of timesteps of series as input and the remaining 40% as output, ignoring the class information.

Qualitative results. The visualization of the predictions are given in Figure 6. Although the predictions under the sDTW and uDTW losses sometimes agree with each other, they can be visibly different. Predictions under uDTW can confidently predict the abrupt and sharp changes. More visualizations can be found in Appendix Sec. E.

Quantitative results. We also provide quantitive results to validate the effectiveness of uDTW. We use ECG5000 dataset from the UCR archive which is composed of 5000 electrocardiograms (ECG) (500 for training and 4500 for testing) of length 140. To better evaluate the predictions, we use 2 different metrics (i) MSE for the predicted errors of each time step (ii) DTW, sDTW div. and uDTW for comparing the 'shape' of time series. We use such shape metrics for evaluation as the length of time series generally varies, and the MSE metric may lead to biased results which ignore the shape trend of time series. We then use the Student's t-test (with significance level 0.05) to highlight the best performance in each experiment (averaged over 100 runs). Table 2 shows that our uDTW achieves almost the best performance on both MSE and shape evaluation metrics (lower score is better).

4.4 Few-shot Action Recognition

Below, we use uDTW as a distance in our objectives for few-shot action recognition (AR) tasks. We implement supervised and unsupervised pipelines (which is also novel).



(a) ECG200 (b) ECG5000

Fig. 6: Given the first part of a time series, we train 3 multi-layer perception (MLP) to predict the remaining part, we use the Euclidean, sDTW or uDTW distance per MLP. We use ECG200 and ECG5000 in UCR archive, and display the prediction obtained for the given test sample with either of these 3 distances and the ground truth (GT). Oftentimes, we observe that uDTW helps predict the sudden changes well.

Table 2: Time series forecasting results evaluated with MSE, DTW, sDTW div. and uDTW metrics on ECG5000, averaged over 100 runs (mean±std). Best method(s) are highlighted in bold using Student's t-test. Column-wise distances indicate the distance used during training. Row-wise distances indicate the distance used to compare prediction with the groundtruth at the test time (lower values are better).

	MSE	DTW	sDTW div.	uDTW
Euclidean	**32.1±1.62**	20.0±0.18	15.3±0.16	14.4±0.18
sDTW [7]	38.6±6.30	**17.2±0.80**	22.6±3.59	32.1±2.25
sDTW div. [3]	24.6±1.37	38.9±5.33	**20.0±2.44**	15.4±1.62
uDTW	23.0±1.22	**16.7±0.08**	**16.8±1.62**	**8.27±0.79**

Experimental setup. For NTU-120, we follow the standard one-shot protocols [24]. Base on this protocol, we create a similar one-shot protocol for NTU-60, with 50/10 action classes used for training/testing respectively (see Appendix Sec. C for details). We also evaluate the model on both 2D and 3D Kinetics-skeleton. We split the whole Kinetics-skeleton into 200 actions for training (the rest is used for testing). We choose Matching Nets (MatchNets) and Prototypical Net (ProtoNet) as baselines as these two models are very popular baselines, and we adapt these methods to skeleton-based action recognition. We reshape and resize each video block into 224×224 color image, and pass this image into MatchNets and ProtoNet to learn the feature representation per video block. We compare uDTW *vs*. Euclidean, sDTW, sDTW div. and recent TAP.

Quantitative results. Table 3, 4 and 5 show that our uDTW performs better than sDTW and sDTW div. on both supervised and unsupervised few-shot action recognition. On Kinetics-skeleton dataset, we gain 2.4% and 4.4% improvements on 3D skeletons for

Table 3: Evaluations on NTU-60.

#classes	10	20	30	40	50
			Supervised		
MatchNets [42]	46.1	48.6	53.3	56.3	58.8
ProtoNet [37]	47.2	51.1	54.3	58.9	63.0
TAP [39]	54.2	57.3	61.7	64.7	68.3
Euclidean	38.5	42.2	45.1	48.3	50.9
sDTW [7]	53.7	56.2	60.0	63.9	67.8
sDTW div. [3]	54.0	57.3	62.1	65.7	69.0
uDTW	56.9	61.2	64.8	68.3	72.4
			Unsupervised		
Euclidean	20.9	23.7	26.3	30.0	33.1
sDTW [7]	35.6	45.2	53.3	56.7	61.7
sDTW div. [3]	36.0	46.1	54.0	57.2	62.0
uDTW	37.0	48.3	55.3	58.0	63.3

Table 4: Evaluations on NTU-120.

#classes	20	40	60	80	100
			Supervised		
MatchNets [42]	20.5	23.4	25.1	28.7	30.0
ProtoNet [37]	21.7	24.0	25.9	29.2	32.1
TAP [39]	31.2	37.7	40.9	44.5	47.3
Euclidean	18.7	21.3	24.9	27.5	30.0
sDTW [7]	30.3	37.2	39.7	44.0	46.8
sDTW div. [3]	30.8	38.1	40.0	44.7	47.3
uDTW	32.2	39.0	41.2	45.3	49.0
			Unsupervised		
Euclidean	13.5	16.3	20.0	24.9	26.2
sDTW [7]	20.1	25.3	32.0	36.9	40.9
sDTW div. [3]	20.8	26.0	33.2	37.5	42.3
uDTW	22.7	28.3	35.9	39.4	44.0

Table 5: Evaluations on 2D and 3D Kinetics-skeleton.

	Supervised		**Unsupervised**	
	2D	3D	2D	3D
Euclidean	21.2	23.1	12.7	13.3
TAP [39]	32.9	36.0	-	-
sDTW [7]	34.7	39.6	23.3	28.3
sDTW div. [3]	35.0	40.1	24.0	28.9
uDTW	35.5	42.0	25.9	32.7

supervised and unsupervised settings. On supervised setting, we outperform TAP by $\sim$ 4% and 2% on NTU-60 and NTU-120 respectively. Moreover, we outperform sDTW by $\sim$ 2% and 3% on NTU-60 and NTU-120 for the unsupervised setting. More evaluations on few-shot action recognition are in Appendix Sec. F.

5 Conclusions

We have introduced the uncertainty-DTW which handles the uncertainty estimation of frame- and/or block-wise features to improve the path warping of the celebrated soft-DTW. Our uDTW produces the uncertainty-weighted distance along the path and returns the regularization penalty aggregated along the path, which follows sound principles of classifier regularization. We have provided several pipelines for time series forecasting, and supervised and unsupervised action recognition, which use uDTW as a distance. Our simple uDTW achieves better sequence alignment in several benchmarks.

References

1. Abid, A., Zou, J.: Autowarp: Learning a warping distance from unlabeled time series using sequence autoencoders. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018) 6
2. Ben-Ari, R., Shpigel Nacson, M., Azulai, O., Barzelay, U., Rotman, D.: Taen: Temporal aware embedding network for few-shot action recognition. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 2780–2788 (2021) 6
3. Blondel, M., Mensch, A., Vert, J.P.: Differentiable divergences between time series. In: Banerjee, A., Fukumizu, K. (eds.) Proceedings of The 24th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 130, pp. 3853–3861. PMLR (13–15 Apr 2021) 6, 12, 13, 14
4. Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C.: Few-shot video classification via temporal alignment. In: CVPR (2020) 6
5. Catalin, Ionescu, Dragos, Papava, Vlad, Olaru, Cristian, Sminchisescu: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Transactions on Pattern Analysis & Machine Intelligence (2014) 10
6. Cuturi, M.: Fast global alignment kernels. In: International Conference on Machine Learning (ICML) (2011) 1, 5, 12
7. Cuturi, M., Blondel, M.: Soft-dtw: a differentiable loss function for time-series. In: International Con- ference on Machine Learning (ICML) (2017) 1, 5, 6, 9, 10, 12, 13, 14
8. Dau, H.A., Keogh, E., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The UCR Time Series Classification Archive (October 2018), https://www.cs.ucr.edu/~eamonn/time_series_data_2018/ 9
9. Dempster, A., Schmidt, D.F., Webb, G.I.: Minirocket: A very fast (almost) deterministic transform for time series classification. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. p. 248–257. KDD '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3447548.3467231 6
10. Donahue, J., Dieleman, S., Binkowski, M., Elsen, E., Simonyan, K.: End-to-end adversarial text-to-speech. In: International Conference on Learning Representations (2021) 6
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020) 7, 27
12. García-García, D., Parrado Hernández, E., Díaz-de María, F.: A new distance measure for model-based sequence clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(7), 1325–1331 (2009). https://doi.org/10.1109/TPAMI.2008.268 6
13. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics, Springer New York Inc., New York, NY, USA (2001) 11
14. Hüllermeier, E., Waegeman, W.: Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Mach. Learn. **110**(3), 457–506 (2021). https://doi.org/10.1007/s10994-021-05946-3 2
15. Indrayan, A.: Medical biostatistics. Chapman & Hall/CRC,, Boca Raton :, 2nd ed. edn. (c2008), http://www.loc.gov/catdir/toc/ecip0723/2007030353.html 2
16. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset (2017) 10

17. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017) 2
18. Kiureghian, A.D., Ditlevsen, O.: Aleatory or epistemic? does it matter? Structural Safety **31**(2), 105–112 (2009). https://doi.org/https://doi.org/10.1016/j.strusafe.2008.06.020, risk Acceptance and Risk Communication 2
19. Koniusz, P., Mikolajczyk, K.: Soft assignment of visual words as linear coordinate coding and optimisation of its reconstruction error. In: 2011 18th IEEE International Conference on Image Processing. pp. 2413–2416 (2011). https://doi.org/10.1109/ICIP.2011.6116129 8
20. Koniusz, P., Wang, L., Cherian, A.: Tensor representations for action recognition. TPAMI (2020) 6
21. Koniusz, P., Yan, F., Mikolajczyk, K.: Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. Computer Vision and Image Understanding **117**(5), 479–492 (2013). https://doi.org/https://doi.org/10.1016/j.cviu.2012.10.010 8
22. Li, S., Liu, H., Qian, R., Li, Y., See, J., Fei, M., Yu, X., Lin, W.: TTAN: two-stage temporal alignment network for few-shot action recognition. CoRR (2021) 6
23. Liu, D.C., Nocedal, J.: On the limited memory bfgs method for large scale optimization. Mathematical Programming **45**, 503–528 (1989) 10
24. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019). https://doi.org/10.1109/TPAMI.2019.2916873 10, 13
25. Liu, L., Wang, L., Liu, X.: In defense of soft-assignment coding. In: 2011 International Conference on Computer Vision. pp. 2486–2493 (2011). https://doi.org/10.1109/ICCV.2011.6126534 8
26. Lohit, S., Wang, Q., Turaga, P.: Temporal transformer networks: Joint learning of invariant and discriminative time warping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 6
27. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3d human pose estimation. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2659–2668 (2017). https://doi.org/10.1109/ICCV.2017.288 10
28. Matthies, H.G.: Quantifying uncertainty: Modern computational representation of probability and applications. In: Ibrahimbegovic, A., Kozar, I. (eds.) Extreme Man-Made and Natural Hazards in Dynamics of Structures. pp. 105–135. Springer Netherlands, Dordrecht (2007) 2
29. Memmesheimer, R., Häring, S., Theisen, N., Paulus, D.: Skeleton-dml: Deep metric learning for skeleton-based one-shot action recognition (2021) 6
30. Memmesheimer, R., Theisen, N., Paulus, D.: Signal level deep metric learning for multimodal one-shot action recognition (2020) 6
31. Mensch, A., Blondel, M.: Differentiable dynamic programming for structured prediction and attention. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 3462–3471. PMLR (10–15 Jul 2018) 6
32. Mina, B., Zoumpourlis, G., Patras, I.: Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. In: Sidorov, K., Hicks, Y. (eds.) Proceedings of the British Machine Vision Conference (BMVC). pp. 130.1–130.14. BMVA Press (September 2019). https://doi.org/10.5244/C.33.130 6
33. Perrett, T., Masullo, A., Burghardt, T., Mirmehdi, M., Damen, D.: Temporal-relational crosstransformers for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 475–484 (June 2021) 6
34. Ramachandran, P., Liu, P.J., Le, Q.V.: Unsupervised pretraining for sequence to sequence learning (2018) 6

35. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **26**(1), 43–49 (1978). https://doi.org/10.1109/TASSP.1978.1163055 6
36. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+d: A large scale dataset for 3d human activity analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (June 2016) 10
37. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. pp. 4077–4087 (2017) 14
38. Su, B., Hua, G.: Order-preserving optimal transport for distances between sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(12), 2961–2974 (2019). https://doi.org/10.1109/TPAMI.2018.2870154 6
39. Su, B., Wen, J.R.: Temporal alignment prediction for supervised representation learning and few-shot sequence classification. In: International Conference on Learning Representations (2022) 6, 14
40. Su, B., Zhou, J., Wu, Y.: Order-preserving wasserstein discriminant analysis. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9884–9893 (2019). https://doi.org/10.1109/ICCV.2019.00998 6
41. Tan, S., Yang, R.: Learning similarity: Feature-aligning network for few-shot action recognition. In: International Joint Conference on Neural Networks (IJCNN). pp. 1–7 (2019) 6
42. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 3630–3638 (2016) 14
43. Wang, L.: Analysis and Evaluation of Kinect-based Action Recognition Algorithms. Master's thesis, School of the Computer Science and Software Engineering, The University of Western Australia (Nov 2017) 6
44. Wang, L., Huynh, D.Q., Koniusz, P.: A comparative review of recent kinect-based action recognition algorithms. IEEE Transactions on Image Processing **29**, 15–28 (2020) 6
45. Wang, L., Huynh, D.Q., Mansour, M.R.: Loss switching fusion with similarity search for video classification. ICIP (2019) 6
46. Wang, L., Koniusz, P.: Self-Supervising Action Recognition by Statistical Moment and Subspace Descriptors, p. 4324–4333. Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3474085.3475572 6
47. Wang, L., Koniusz, P., Huynh, D.Q.: Hallucinating IDT descriptors and I3D optical flow features for action recognition with cnns. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019) 6
48. Yan, S., Xiong, Y., Lin, D.: Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In: AAAI (2018) 10
49. Yang, C.H.H., Tsai, Y.Y., Chen, P.Y.: Voice2series: Reprogramming acoustic models for time series classification. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 11808–11819. PMLR (18–24 Jul 2021) 6
50. Zhang, H., Zhang, L., Qi, X., Li, H., Torr, P., Koniusz, P.: Few-shot action recognition with permutation-invariant attention. In: European Conference on Computer Vision (ECCV) (2020) 6, 7
51. Zhu, H., Koniusz, P.: Simple spectral graph convolution. In: International Conference on Learning Representations (ICLR) (2021) 7