Drop-DTW: Aligning Common Signal Between Sequences While Dropping Outliers

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work, we consider the problem of sequence-to-sequence alignment for signals containing outliers. Assuming the absence of outliers, the standard Dynamic Time Warping (DTW) algorithm efficiently computes the optimal alignment between two (generally) variable-length sequences. While DTW is robust to temporal shifts and dilations of the signal, it fails to align sequences in a meaningful way in the presence of outliers that can be arbitrarily interspersed in the sequences. To address this problem, we introduce Drop-DTW, a novel algorithm that aligns the common signal between the sequences while automatically dropping the outlier elements from the matching. The entire procedure is implemented as a single dynamic program that is efficient and fully differentiable. In our experiments, we show that Drop-DTW is a robust similarity measure for sequence retrieval and demonstrate its effectiveness as a training loss on diverse applications. With Drop-DTW, we address temporal step localization on instructional videos, representation learning from noisy videos, and cross-modal representation learning for audio-visual retrieval and localization. In all applications, we take a weakly- or unsupervised approach and demonstrate state-of-the-art results under these settings.

1 Introduction

The problem of sequence-to-sequence alignment is central to many computational applications. Aligning two sequences (e.g., temporal signals) entails computing the optimal pairwise correspondence between the sequence elements while preserving their match orderings. For example, video [1] or audio [2] synchronization are important applications of sequence alignment in the same modality, while the alignment of video to audio [3] represents cross-modal task. Dynamic Time Warping (DTW) [2] is a standard algorithm for recovering the optimal alignment between two variable length sequences. It efficiently solves the alignment problem by finding all correspondences between the two sequences, while being robust to temporal variations in the execution rate and shifts.

A major issue with DTW is that it enforces correspondences between *all* elements of both sequences and thus cannot properly handle sequences containing outliers. That is, given sequences with interspersed outliers, DTW enforces matches between the outliers and clean signal, which is prohibitive in many applications. A real-world example of sequences containing outliers is instructional videos. These are long, untrimmed videos depicting a person performing a given activity (e.g., making a latte) following a pre-defined set of ordered steps (e.g., a recipe). Typically, only a few frames in the video correspond to the instruction steps, while the rest of the video frames are unrelated to the main activity (e.g., the person talking); see Figure 1 for an illustration. In this case, matching the outlier frames to instruction steps will not yield a meaningful alignment. Moreover, match score between such sequences, computed by DTW, will be negatively impacted by the "false" correspondences and therefore cannot be used reliably for downstream tasks, e.g., retrieval or representation learning.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Figure 1: Aligning instructional videos. Left: Both video sequences (top and bottom) depict the three main steps of "making latte"; however, there are unrelated video segments, i.e., outliers, inbetween the steps. DTW aligns all the frames with each other and creates false correspondences where outliers are matched to signal (red links). Right: In contrast, Drop-DTW identifies the common signal between two videos, drops unrelated frames (crossed out), and aligns only the instruction steps, leaving only correct correspondences (green links).

In this paper, we introduce Drop-DTW to address the problem of matching sequences that contain interspersed outliers. While various improvements to DTW have been previously proposed (e.g., [4, 5, 1, 6]), Drop-DTW is the first to augment DTW with the ability to skip through irrelevant parts of the signal during alignment. This is achieved by identifying the common signal between the sequences, defining the outlier elements to be dropped, and aligning the remaining subsequences. This process is illustrated in Figure 1 (right) and compared to standard DTW on the (left). Drop-DTW casts sequence alignment as an optimization problem with a novel cost component explicitly encouraging outlier-free matches. It is efficiently realized using a dynamic program that naturally admits a differentiable approximation and can be efficiently used at training and inference time.

Contributions.

- We propose an extension of DTW that is able to identify and align the common signal between sequences, while excluding interspersed outliers from the alignment.
- The proposed Drop-DTW formulation naturally admits a differentiable approximation and we therefore demonstrate its usage as a training loss function.
- We demonstrate the utility of Drop-DTW, for both training and inference for multi-step localization in instructional videos, using only ordered steps as weak supervision. We achieve state-of-the-art results on the CrossTask [7] dataset and are the first to tackle the COIN [8] and YouCook2 [9] datasets, given only ordered steps (i.e., no framewise labels are used).
- We employ Drop-DTW as a loss function for weakly-supervised video representation learning on the PennAction dataset [10], modified to have interspersed outliers, and unsupervised audio-visual representation learning on the AVE dataset [11]. Compared to the baselines, Drop-DTW yields superior representations as measured by its performance on various downstream tasks.

We are committed to releasing the code upon acceptance.

2 Related work

Sequence alignment. The use of aligned sequences in learning has recently seen a growing interest across various tasks [12, 4, 5, 13, 1, 6, 14, 15]. While some methods start from aligned paired signals [12, 16], others directly learn to align unsynchronized signals [13, 5, 1, 6]. One approach tackles the alignment problem locally by maximizing the number of one-to-one correspondences using a soft nearest neighbors method [13]. More closely related are methods [4, 5, 1, 14] that seek a global alignment between sequences by relying on Dynamic Time Warping (DTW) [2]. To use DTW for end-to-end learning, differentiable approximations of the discrete operations (i.e., the min operator) in DTW have been explored [4, 1]. One of the main downsides of extant DTW-based approaches is that they require clean, tightly cropped, data with matching endpoints. In contrast to existing methods, Drop-DTW extends DTW and its differentiable variants with the ability to handle outliers in the sequences by allowing the alignment process to automatically handle outliers (i.e., match the signal and drop outliers). While previous work [6, 17] targeted rejecting outliers limited to the start or end of the sequences, our proposed Drop-DTW is designed to handle outliers interspersed in the

sequence, including at the endpoints. As shown in our experiments, this ability to skip outliers during alignment can in turn be used at inference time to localize the start and end times of subsequences.

Representation learning. Self- and weakly-supervised approaches are increasingly popular for representation learning as they eschew the need of dense labels for training. Such approaches are even more appealing for video-based tasks that typically require inordinately large labeling efforts. A variety of (label-free) proxy tasks have emerged for the purpose of video representation learning [18–35]. More closely related are methods using sequence alignment as a proxy task [5, 13, 1, 6, 14, 15]. Our approach has the added advantage of handling interspersed outliers during alignment. Also related are methods that leverage multimodal aspects of video, e.g., video-audio [36, 37] or video-language alignment [38–41]. Our step localization application builds on strong representations learned via the task of aligning vision and language using narrated instructional videos [39]. However, we augment these representations with a more global and robust approach to the alignment process, thereby directly enabling fine-grained applications, such as multi-step localization.

Step localization. Temporal step localization consists of determining the start and end times of one or more activities present in a video. These methods can be broadly categorized into two classes based on the level of training supervision involved. Fully supervised approaches (e.g., [42, 43, 8]) rely on fine-grained temporal labels indicating the start and end times of activities. Weakly supervised approaches eschew the need for framewise labels. Instead, they use a video-level label corresponding to the activity category [44, 45], or sparse time stamps that provide the approximate (temporal) locations of the activities. More closely related are methods that rely on the order of instruction steps in a clip to yield framewise step localization [46–48, 5, 7]. In contrast, we do not rely on abstract step labels and demonstrate that our approach applies to any ordered sequence of embedded steps, such as embedded descriptions in natural language.

3 Technical approach

3.1 Preliminaries: Dynamic time warping

Dynamic Time Warping (DTW) [2] computes the optimal alignment between two sequences subject to certain constraints. Let $X = [x_1, ..., x_N]^\top \in \mathbb{R}^{N \times d}$ and $Z = [z_1, ..., z_K]^\top \in \mathbb{R}^{K \times d}$ be the input sequences, where N, K are the respective sequence lengths and d is the dimensionality of each sequence element. The valid alignments between sequences are defined as a binary matrix, $M \in \{0, 1\}^{K \times N}$, of the pairwise correspondences, where $M_{i,j} = 1$ if z_i is matched to x_j, and 0 otherwise. Note that Z corresponds to the first (row) index of M and X the second (column). Matching an element z_i to element x_j has a cost $C_{i,j}$ which is typically a measure of dissimilarity between the elements. DTW finds the alignment M^* between sequences Z and X that minimizes the overall matching cost:

$$M^* = \underset{M \in \mathcal{M}}{\arg\min} \langle M, C \rangle = \underset{M \in \mathcal{M}}{\arg\min} \sum_{i,j} M_{i,j} C_{i,j}, \tag{1}$$

where $\langle M, C \rangle$ is the Frobenius inner product and $\mathcal{M}$ is the set of all feasible alignments that satisfy the following constraints: monotonicity, continuity, and matching endpoints ($M_{1,1} = M_{K,N} = 1, \forall M \in \mathcal{M}$). Fig. 2 (a) provides an illustration of feasible and optimal alignments. The cost of aligning two sequences with DTW is defined as the cost of the optimal matching: $c^* = \langle M^*, C \rangle$.

In the case of sequences containing outliers, the matching endpoints and continuity constraints are too restrictive. Thus, we introduce Drop-DTW that admits more flexibility in the matching process.

3.2 Subsequence alignment with Drop-DTW

Drop-DTW limits consideration of feasible alignments, $\mathcal{M}$, to those adhering to the monotonicity constraint only. Consequently, unlike DTW, elements can be dropped from the alignment process. Dropping an element x_i means there is no element in Z that has been matched to x_i. This is captured in the i-th column of the correspondence matrix, M, containing only zeros, i.e., $M_{:,i} = \mathbf{0}$.

To account for unmatched elements in the alignment objective, we extend the set of costs beyond pairwise matching, $C_{i,j}$, used in DTW, with novel drop costs $d_i^z \in \mathbb{R}^K$ and $d_j^x \in \mathbb{R}^N$, for elements

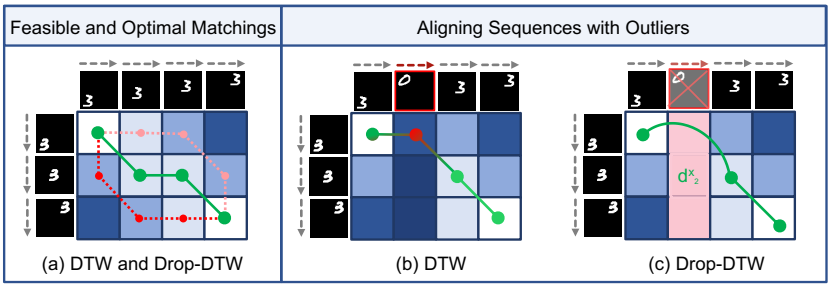

Figure 2: Optimal alignment with DTW and Drop-DTW. Aligning two different videos where digit "3" moves across the square frame. The colored matrices represent the pairwise matching costs, C, with darker cell indicating higher cost, $C_{i,j}$. The paths on the grid are alignment paths, while the points on them indicate a pairwise match between the corresponding row and column elements. **(a)** All three paths are feasible DTW paths, while only one of them (in green) is optimal. **(b)** When sequence X contains an outlier (i.e., digit "0"), DTW uses it in the alignment and incurs a high cost (red point). **(c)** In contrast, Drop-DTW skips the outlier (while paying the cost d_2^x) and only keeps the relevant matches.

Algorithm 1 Subsequence alignment with Drop-DTW.

Inputs: $C \in \mathbb{R}^{K \times N}$ - pairwise match cost matrix, d^x - drop costs for elements in x.
▷ *initializing dynamic programming tables*
$D^+_{0,0} = 0; D^+_{i,0} = \infty; D^+_{0,j} = \infty;$ $\quad i \in [\![K]\!], j \in [\![N]\!]$ ▷ *match table*
$D^-_{0,0} = 0; D^-_{i,0} = \infty; D^-_{0,j} = \sum_{k=1}^{j} d^x_k;$ $\quad i \in [\![K]\!], j \in [\![N]\!]$ ▷ *drop table*
$D_{0,0} = 0; D_{i,0} = D^-_{i,0}; D_{0,j} = D^-_{0,j};$ $\quad i \in [\![K]\!], j \in [\![N]\!]$ ▷ *optimal solution table*
for $i = 1, \ldots, K$ **do** ▷ *iterating over elements in Z*
  for $j = 1, \ldots, N$ **do** ▷ *iterating over elements in X*
    $D^+_{i,j} = C_{i,j} + \min\{D_{i-1,j-1}, D_{i,j-1}, D^+_{i-1,j}\}$ ▷ *consider matching z_i to x_j*
    $D^-_{i,j} = d^x_j + D_{i,j-1}$ ▷ *consider dropping x_j*
    $D_{i,j} = \min\{D^+_{i,j}, D^-_{i,j}\}$ ▷ *select the optimal action*
  end for
end for
$M^* = \text{traceback}(D)$ ▷ *compute the optimal alignment by tracing back the minimum cost path*
Output: $D_{K,N}$, M^*

$x_j \in X$ and $z_i \in Z$, respectively. The optimal matching can be then defined as follows:

$$M^* = \arg\min_{M \in \bar{\mathcal{M}}} \langle M, C \rangle + P_x(M) \cdot (d^z_i) + P_y(M) \cdot \left(d^z_j\right), \tag{2}$$

where $\cdot$ denotes the inner product, $\bar{\mathcal{M}}$ is the set of binary matrices satisfying just the monotonicity constraint, and $P_x(M)$ is a vector with the i-th element equal to one if the $M_{:,i} = \mathbf{0}$ and zero otherwise; $P_y(M)$ is defined similarly, but on columns.

For clarity, in Algorithm 1, we describe the dynamic program for efficiently computing the optimal alignment and its cost with dropping elements limited to X, i.e., $d^z_i = \infty$. Our general Drop-DTW algorithm that drops elements from *both* sequences, X and Z, is given in the supplemental.

3.3 Definition of match and drop costs

Drop costs. As a starting point, consider a drop cost that is constant across all elements in X and Z: $d^x_j = d^z_i = s$. Setting the constant s too low (relative to the match costs, C) will lead to a high frequency of drops and thus matching only a small signal fraction. In contrast, setting s too high may result in retaining the outliers, i.e., no drops. To avoid such extreme outcomes, we define the drop cost on a per-instance basis with values comparable to those of the match cost, C. In particular, we define the drop costs as the $p \in [0, 100]$ top percentile of the values contained in the cost matrix, C:

$$s = \text{percentile}(\{C_{i,j} \mid 1 \leq i \leq K, 1 \leq j \leq N\}, p). \tag{3}$$

Defining the drop cost as a function of the top percentile match costs has the advantage that adjusting p allows one to inject a prior belief on the outlier rate in the input sequences.

Match costs. The pairwise match costs, $C_{i,j}$, are typically defined based on the dissimilarity between elements x_j and z_i. In this paper, we consider two different ways to define the pairwise costs, depending on the application. In general, when dropping elements from both X and Z is permitted, we consider the following symmetric match cost:

$$C^s_{i,j} = 1 - \cos(z_i, x_j). \tag{4}$$

Alternatively, when one of the sequences, i.e., Z, is known to contain only signal, we follow [1], who show that the following asymmetric matching cost is useful during representation learning:

$$C^a_{i,j} = -\log(\text{softmax}_1(Z^\top X/\gamma))_{i,j}, \tag{5}$$

where $\text{softmax}_1(\cdot)$ defines a standard softmax operator applied over the first tensor dimension.

3.4 Drop-DTW as a differentiable loss function

Differentiable approximation. To make the matching process differentiable, we replace the hard $\min$ operator in Alg. 1 with the following differentiable approximation introduced in [1]:

$$\text{smoothMin}(\boldsymbol{x}; \gamma) = \boldsymbol{x} \cdot \text{softmax}(\boldsymbol{x}/\gamma) = \frac{\boldsymbol{x} \cdot e^{\boldsymbol{x}/\gamma}}{\sum_j e^{x_j/\gamma}}, \tag{6}$$

where $\gamma > 0$ is a hyper-parameter controlling the trade-off between smoothness and the error of the approximation.

Loss function. The differentiable Drop-DTW is a sequence match cost suitable for representation learning (cf. [1]). The corresponding Drop-DTW loss function is defined as follows:

$$\mathcal{L}_{\text{DTW}}(Z, X) = \text{Drop-DTW}(Z, X) = D_{K,N}(Z, X), \tag{7}$$

where $D_{K,N}(Z, X)$ is the optimal match cost between Z and X computed in Alg. 1.

4 Experiments

The key novelty of Drop-DTW is the ability to robustly match sequences with outliers, by identifying the outliers and limiting matching to the inliers. In Sec. 4.1, we first present a controlled experiment using synthetic data to demonstrate the robustness of Drop-DTW. Next, we show the strength of Drop-DTW on a range of applications, including multi-step localization (Sec 4.2), representation learning from noisy videos (Sec 4.3), and audio-visual alignment for retrieval and localization (Sec 4.4).

4.1 Controlled synthetic experiments

Synthetic dataset. We use the MNIST dataset [49] to generate videos of moving digits (cf. [50]). Each video in our dataset depicts a single digit moving along a given trajectory. For each digit-trajectory pair, we generate two videos: (i) the digit moves along the full trajectory (termed MNIST-full) and (ii) the digit performs a random sub-part of the trajectory (termed TMNIST-part). Each video frame is independently encoded with a shallow ConvNet [49] trained for digit recognition. We elaborate on the the dataset and the framewise embedding network in the supplemental. In the following sections, we use these datasets for the purpose of sequence retrieval and subsequence localization.

4.1.1 Video retrieval

In this experiment, we use TMNIST-part as queries and look for the most similar videos in TMNIST-full. A correct retrieval is defined as identifying the video in TMNIST-full containing the same digit-trajectory pair as the one in the query. We use the Recall@1 metric to report performance. To analyze the strength of Drop-DTW in this controlled setting, we also introduce temporal noise in the query sequence by randomly blurring a subset of the video frames. We compare Drop-DTW to standard DTW [2] in this experiment. In all cases, we use the alignment score obtained from each algorithm as a matching cost between sequences. For this experiment, we use the symmetric matching costs defined in (4) and we empirically set the drop costs to a constant, $d^x = d^z = 0.3$.

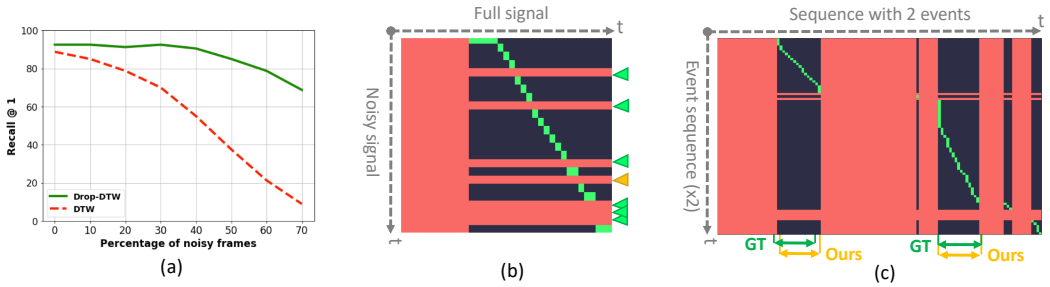

Figure 3: Drop-DTW for retrieval and event localization on TMNIST. (a) We consider queries, Z, from TMNIST-part with interspersed noise, for indexing into TMNIST-full. Drop-DTW is more robust to interspersed noise than other alignment algorithms. (b) An example query-signal correspondence matrix illustrates outlier removal, where red rows (cols) depict dropped frames in Z (X, resp.), and green (yellow) arrows denote correct (incorrect) query outlier identification. (c) Replicating the query and dataset signal in time demonstrates temporal localization by Drop-DTW despite signal repetitions and interspersed outliers (see main text).

Figure 3 (a) compares the performance of the two alignment algorithms with various levels of noise. We can see that DTW is very sensitive to interspersed noise. In contrast, Drop-DTW remains robust across the range of noise levels. Concretely, for the highest noise level in Fig 3(a), we see that Drop-DTW is $8\times$ better than DTW, which speaks decisively in favor of Drop-DTW.

4.1.2 Subsequence localization

To demonstrate Drop-DTW's ability to identify start and end times we consider test sequences, X, formed by concatenating M clips from TMNIST-part, where exactly $N < M$ of these clips are from a selected digit-trajectory class. We consider queries of the form $\hat{Z} = [Z^{(1)}, \ldots, Z^{(N)}]$, where each of the $Z^{(n)}$'s are instances of this same digit-trajectory class in TMNIST-full. Ideally, aligning $\hat{Z}$ and X should put a subsequence of each of the $Z^{(n)}$'s in correspondence with a matching subsequence in X (see Fig. 3 (c)). By construction, there should be N such matches, interspersed with outliers. Here, we use $N = 2$, $M \in \{5, \ldots, 7\}$, and randomly generate 1000 such sequences X.

We use Drop-DTW to align the generated pairs, $\hat{Z}$ and X, and select the longest N contiguously matched subsequences. We recover start and end times as the endpoints of these N longest matched intervals. We find the localization accuracy on this data is $97.1 \pm 0.28\%$ and IoU is $91.2 \pm 0.71\%$ (see metrics definition in Sec. 4.2). These results demonstrate the efficacy of Drop-DTW for localizing events in long untrimmed videos. Note that we do not compare our method to any baseline as Drop-DTW is the only alignment-based method capable of solving this task.

4.2 Multi-step localization

We now evaluate Drop-DTW on multi-step localization in more realistic settings. For this task, we are given as input: (i) a long untrimmed video of a person performing a task (e.g., "making salad"), where the steps involved in performing the main task are interspersed with irrelevant actions (e.g., intermittently washing dishes), and (ii) an ordered set of textual descriptions of the main steps (e.g., "cut tomatoes") in this video. The goal is to temporally localize each step in the video.

Datasets. For evaluation, we use the following three recent instructional video datasets: CrossTask [7], COIN [8], and YouCook2 [9]. Both CrossTask and COIN include instructional videos of different activity types (i.e., tasks), with COIN being twice the size of CrossTask in the number of videos and spanning 10 times more tasks. YouCook2 is the smallest of the three datasets and focuses on cooking tasks. While all datasets provide framewise labels for the start and end times of each step in a video, we take a weakly-supervised approach and only use the ordered step information in our algorithm.

Metrics. We evaluate the performance of Drop-DTW on step localization using three increasingly strict metrics. **Recall** [7] is defined as the number of steps correctly assigned to the correct ground truth time interval divided by the total number of steps and is the least strict metric out of the three considered. **Framewise accuracy (Acc.)** [8] is defined as the ratio between the number of frames assigned the correct step label (including background) and the total number of frames. Finally,

Table 1: Ablation study of training and inference methods for step localization on CrossTask [7]. The first column describes the loss function(s) used for training, while the first row gives an inference procedure. The second row specifies the measured metric, and the rest of the table reports the results. Here, CT (in row 1), corresponds to the evaluation algorithm provided by CrossTask. In column 1, * means our re-implementation.

Method	CT	Drop-DTW		drop *then* DTW	
	Recall ↑	Acc. ↑	IoU ↑	Acc.↑	IoU ↑
MIL-NCE* [39]	39.1	66.9	20.9	60.1	13.9
SmoothDTW	29.6	48.5	9.0	44.1	7.1
Drop-DTW	33.8	56.6	14.2	48.5	11.9
$\mathcal{L}_{\text{clust}}$	43.4	70.2	29.8	60.5	21.5
SmoothDTW + $\mathcal{L}_{\text{clust}}$	43.1	70.2	30.5	60.8	22.2
Drop-DTW + $\mathcal{L}_{\text{clust}}$	**48.9**	**71.3**	**34.2**	66.2	26.2

Intersection over Union (IoU) [9] is defined as the sum of the intersections between the predicted and ground truth time intervals of each step divided by the sum of their unions. IoU is the most challenging of the three metrics as it more strictly penalizes misalignments.

Training and Inference. We start from (pretrained) strong vision and language embeddings obtained from training on a large instructional video dataset [39]. We further train a two-layer fully-connected network on top of the visual embeddings alone to align videos with a list of corresponding step (language) embeddings using the Drop-DTW loss,(7). Notably, the number of different step descriptions in a dataset is orders of magnitude smaller than the number of video clips, which leads to degenerate alignments when training with the Drop-DTW loss, (7), i.e., most clips are matched to one step. To regularize the training, we introduce an additional clustering loss, $\mathcal{L}_{\text{clust}}$, defined in the supplementary. $\mathcal{L}_{\text{clust}}$ is an order-agnostic discriminative loss that encourages video embeddings to cluster around the embeddings of steps present in the video. Finally, we use $\mathcal{L}_{\text{DTW}}$ with asymmetric costs, (5), and 30%-percentile drop costs in combination with $\mathcal{L}_{\text{clust}}$ during training.

At test time, we are given a video and the corresponding ordered list of step descriptions. To temporarily localize each of the steps in the video, we align the learned vision and (pre-trained) language embeddings using Drop-DTW.

4.2.1 Ablation study

We first investigate the role of each component of the proposed loss for step localization on CrossTask (and report ablations on YouCook2 and COIN in the supplemental). We train the video embeddings with various alignment losses, with or without regularization, and compare with the pre-trained embeddings. At inference time, we use the learned embeddings to calculate pairwise similarities between all video clips and steps, and use the similarities to assign a step label to each video clip. Notably, with Drop-DTW we can infer the start and end times of each step directly *during* the alignment as our algorithm supports skipping outliers. For a further comparison, we also consider dropping outliers *before* alignment. In this case, a clip is dropped if the drop cost (defined in Section 3.3) is less than the cost of matching the clip to any step. The alignment is then performed on the remaining clips.

As seen in Table 1, using the clustering loss alone on top of the pre-trained embeddings provides a strong baseline, as it effectively uses an unordered set of steps to learn the video representations. Interestingly, combining SmoothDTW with $\mathcal{L}_{\text{clust}}$ to introduce order information in training does not improve (and can decrease) performance. We conjecture that the outliers present in the sequences can negatively impact the training signal. In contrast, training with Drop-DTW yields a significant boost in performance on all the metrics. This suggests that filtering out the background is key to training with step order supervision on instructional videos. Notably, using only an alignment loss without regularization leads to poor localization performance due to the degenerate alignment problem discussed above. Comparing the performance between Drop-DTW and the "drop then DTW" heuristic clearly shows the superiority of formulating dropping as part of the optimization.

Table 2: Step localization results on CrossTask [7], COIN [8], and YouCook2 [9].

Method		CrossTask			COIN			YouCook2		
		Recall ↑	Acc. ↑	IoU ↑	Recall ↑	Acc. ↑	IoU ↑	Recall ↑	Acc. ↑	IoU ↑
Non-DTW Methods	MaxMargin [38]	33.6	-	-	-	-	-	-	-	-
	UniVL [41]	42.0	-	-	-	-	-	-	-	-
	NN-Viterbi [48]	-	-	-	-	21.2	-	-	-	-
	ISBA [47]	-	-	-	-	34.3	-	-	-	-
	ProcNet [9]	-	-	-	-	-	-	-	-	37.5
	MIL-NCE [39]	39.1	66.9	20.9	33.0	50.2	23.3	70.7	63.7	43.7
DTW-based Methods	SmoothDTW [1]	43.1	70.2	30.5	37.7	52.7	27.7	75.3	66.0	47.5
	D^3TW [5]	43.2	70.4	30.6	37.9	52.8	27.7	75.3	66.4	47.1
	OTAM [6]	43.8	70.6	30.8	37.2	52.6	27.2	75.4	66.9	46.8
	Drop-DTW	**48.9**	**71.3**	**34.2**	**40.8**	**54.8**	**29.5**	**77.4**	**68.4**	**49.4**

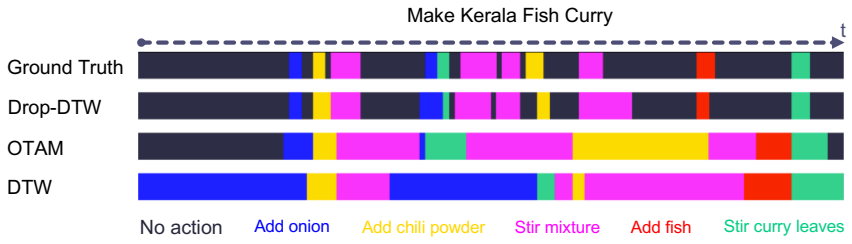

Figure 4: Step localization with DTW variants. Rows two to four show step assignment results when the same alignment method is used for training and inference. Drop-DTW allows to identify interspersed unlabelled clips and much more closely approximates the ground truth.

4.2.2 Comparison to baselines

We now compare the performance of Drop-DTW to various alignment methods [1, 5, 6], as well as previous reported results on the datasets [38, 9, 47, 48, 39, 41]. In particular, we compare to the following baselines:

- *SmoothDTW* [1]: Learns fine-grained representations in a contrastive setting using a differentiable approximation of DTW. Drop-DTW uses the same approximation, while supporting the ability to skip irrelevant parts of sequences.

- *D^3TW* [5]: Uses a different differentiable formulation of DTW, as well as a discriminative loss.

- *OTAM* [6]: Extends D^3TW with the ability to handle outliers strictly present around the endpoints of a sequence.

- *MIL-NCE* [39]: Learns strong vision and language representations by locally aligning videos and narrations using a noise contrastive loss in a multi-instance learning setting. Notably, we learn our embeddings on top of representations trained with this loss on the HowTo100M dataset [38]. There is a slight difference in performance between our re-implementation and [38], which is due to the difference in the input clip length and frame rate.

The results in Table 2 speak decisively in favor of our approach, where we outperform all the baselines on all datasets across all the metrics. Notably, although we are training a shallow fully-connected network on top of the MIL-NCE's [39] frozen pretrained visual representations, we still outperform it with sizeable margins due to the global perspective taken in aligning the two modalities. Also, note that the results of all other DTW-based methods [1, 5, 6] are on-par due to their collective inability to handle interspersed background frames during alignment, as shown in Fig. 4. This figure also motivates the use of Drop-DTW as the inference method for step localization. It uses the drop capability to 'label' irrelevant frames as background, without ever learning a background model, and does it rather accurately according to Fig. 4, which other methods are not designed to handle.

4.3 Drop-DTW for representation learning

In this section, we demonstrate the utility of the Drop-DTW loss, (7), for representation learning. We train the same network used in previous work [13, 1] using the alignment proxy task on PennAction [10]. Similar to previous work [13, 1], we evaluate the quality of the learned representations (i.e.,

embeddings) on the task of video alignment using Kendall's Tau metric. Notably, the PennAction videos are tightly temporally cropped around the actions, making it suitable for an alignment-based proxy task. To assess the robustness of Drop-DTW to the presence of outliers, we also contaminate PennAction sequences by randomly introducing $p\%$ frames from other activities to each sequence. Note that this type of data transformation is similar to a realistic scenario where a video of a baseball game sequentially captures both baseball pitch and baseball swing actions, as well as crowd shots. For various amounts of outlier frames, we contaminate PennAction videos as specified above and train a network to perform sequence alignment with the SmoothDTW [1] or Drop-DTW loss. Note that no regularization loss is needed in this case and the drop cost is defined according to (3). Fig. 5 demonstrates the strong performance of Drop-DTW as a loss for representation learning, which once again proves to be more resilient to outliers.

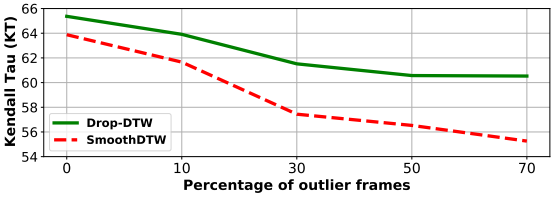

Figure 5: Representation learning with Drop-DTW. Video alignment results using Kendall's Tau metric on PennAction [10] with an increasing level of outliers introduced in training.

Table 3: Unsupervised audio-visual cross-modal localization on AVE [11]. A2V: visual localization from audio segment query; V2A: audio localization from visual segment query.

Model	A2V ↑	V2A ↑
OTAM [6]	37.5	32.7
SmoothDTW [1]	39.8	33.9
Drop-DTW (ours)	**41.1**	**35.8**
Supervised [11]	44.8	34.8

4.4 Drop-DTW for audio-visual localization

To show the benefits of Drop-DTW for unsupervised audio-visual representation learning, we adopt the cross-modality localization task from [11]. Given a trimmed signal from one modality (e.g., audio), the goal is localize the signal in an untrimmed sequence in the other modality (e.g., visual). While the authors of [11] use supervised learning with clip-wise video annotations indicating the presence (or absence) of an audio-visual event, we train our models completely *unsupervised*. We use a margin-based loss that encourages audio-visual sequence pairs from the same video to be closer in the shared embedding space than audio-visual pairs from different videos, and measure the matching cost between the sequences with a matching algorithm. When applying Drop-DTW, we use symmetric match costs, (4), and 70%-percentile drop costs defined in Section 3.3. Otherwise, we closely follow the experimental setup from [11] and adopt their encoder architecture and evaluation protocol; for additional details, please see the supplemental.

In Table 3, we compare different alignment methods used as a sequence matching cost to train the audio-visual model. We can see that Drop-DTW outperforms both SmoothDTW and OTAM due to its ability to drop outliers from both sequences. Interestingly, Drop-DTW outperforms the fully-supervised baseline on visual localization, given an audio query. We hypothesize that this is due to our triplet loss formulation, that introduces learning signals from non-matching audio-visual signals, while supervised training in [11] only considers audio-visual pairs from the same video.

4.5 Discussion and limitations

Even though Drop-DTW can remove arbitrary sequence elements as part of the alignment process, the final alignment is still subject to the monotonicity constraint, which is based on the assumption that the relevant signals are strictly ordered. While not relevant for the applications presented in this paper, training from partial ordered signals using Drop-DTW is not currently addressed and is an interesting subject for future research.

5 Conclusion

In summary, we introduced an extension to the classic DTW algorithm, which relaxes the constraints of matching endpoints of paired sequences and the continuity of the path cost. This relaxation allows our method to handle interspersed outliers during sequence alignment. The proposed algorithm is efficiently implemented as a dynamic program and naturally admits a differentiable approximation. We showed that Drop-DTW can be used both as a flexible matching cost between sequences and a loss during training. Finally, we demonstrated the strengths of Drop-DTW across a range of applications.

References

[1] I. Hadji, K. G. Derpanis, and A. D. Jepson, "Representation learning via global temporal alignment and cycle-consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[2] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken processing recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 26, 1978.

[3] T. Halperin, A. Ephrat, and S. Peleg, "Dynamic temporal alignment of speech to lips," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3980–3984, 2019.

[4] M. Cuturi and M. Blondel, "Soft-DTW: A differentiable loss function for time-series," in *International Conference on Machine Learning (ICML)*, 2017.

[5] C. Chang, D. Huang, Y. Sui, L. Fei-Fei, and J. C. Niebles, "D3TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] K. Cao, J. Ji, Z. Cao, C. Chang, and J. C. Niebles, "Few-shot video classification via temporal alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] D. Zhukov, J.-B. Alayrac, R. G. Cinbis, D. Fouhey, I. Laptev, and J. Sivic, "Cross-task weakly supervised learning from instructional videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[8] Y. Tang, D. Ding, Y. Rao, Y. Zheng, D. Zhang, L. Zhao, J. Lu, and J. Zhou, "COIN: A large-scale dataset for comprehensive instructional video analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] L. Zhou, C. Xu, and J. J. Corso, "Towards automatic learning of procedures from web instructional videos," in *AAAI Conference on Artificial Intelligence*, 2018.

[10] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2013.

[11] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu, "Audio-visual event localization in unconstrained videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[12] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine, "Time-contrastive networks: Self-supervised learning from video," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[13] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "Temporal cycle-consistency learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] X. Chang, F. Tung, and G. Mori, "Learning discriminative prototypes with dynamic time warping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[15] X. Cai, T. Xu, J. Yi, J. Huang, and S. Rajasekaran, "DTWNet: A dynamic time warping network," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[16] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, "Actor and observer: Joint modeling of first and third-person videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] Y. Sakurai, C. Faloutsos, and M. Yamamuro, "Stream monitoring under the time warping distance," in *International Conference on Data Engineering (ICDE)*, 2007.

[18] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: Unsupervised learning using temporal order verification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[19] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-supervised video representation learning with odd-one-out networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] H. Lee, J. Huang, M. Singh, and M. Yang, "Unsupervised representation learning by sorting sequences," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[21] U. Büchler, B. Brattoli, and B. Ommer, "Improving spatiotemporal self-supervision by deep reinforcement learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[22] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, "Self-supervised spatiotemporal learning via video clip order prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] D. Wei, J. J. Lim, A. Zisserman, and W. T. Freeman, "Learning and using the arrow of time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] J. J. Yu, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, 2016.

[25] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *AAAI Conference on Artificial Intelligence*, 2018.

[26] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, "Tracking emerges by colorizing videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[27] J. Janai, F. Güney, A. Ranjan, M. J. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[28] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[29] X. Wang, A. Jabri, and A. A. Efros, "Learning correspondence from the cycle-consistency of time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] A. Jabri, A. Owens, and A. A. Efros, "Space-time correspondence as a contrastive random walk," in *NeurIPS*, 2020.

[31] S. Benaim, A. Ephrat, O. Lang, I. Mosseri, W. T. Freeman, M. Rubinstein, M. Irani, and T. Dekel, "SpeedNet: Learning the speediness in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[32] D. Jayaraman and K. Grauman, "Slow and steand feature analysis: Higher order temporal coherence in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[33] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating visual representations from unlabeled video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] J. Wang, J. Jiao, L. Bao, S. He, Y. Liu, and W. Liu, "Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] T. Han, W. Xie, and A. Zisserman, "Video representation learning by dense predictive coding," in *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, 2019.

[36] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[37] R. Arandjelovic and A. Zisserman, "Objects that sound," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[38] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[39] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-End Learning of Visual Representations from Uncurated Instructional Videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "VideoBERT: A joint model for video and language representation learning," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[41] H. Luo, L. Ji, B. Shi, H. Huang, N. Duan, T. Li, J. Li, T. Bharti, and M. Zhou, "UniVL: A unified video and language pre-training model for multimodal understanding and generation," *arXiv preprint arXiv:2002.06353*, 2020.

[42] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[43] M. Ma, H. Fan, and K. M. Kitani, "Going deeper into first-person activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[44] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[45] P. X. Nguyen, D. Ramanan, and C. C. Fowlkes, "Weakly-supervised action localization with background modeling," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[46] D. Huang, L. Fei-Fei, and J. C. Niebles, "Connectionist temporal modeling for weakly supervised action labeling," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[47] L. Ding and C. Xu, "Weakly-supervised action segmentation with iterative soft boundary assignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[48] A. Richard, H. Kuehne, A. Iqbal, and J. Gall, "NeuralNetwork-Viterbi: A framework for weakly supervised video learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[49] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[50] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *International Conference on Machine Learning (ICML)*, 2015.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 4
   (b) Did you describe the limitations of your work? [Yes] See Section 4.5
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] **(i)** We use publically available datasets, **(ii)** we provide implementation details in the supplemental and **(iii)** plan to release code upon decision. See Line-59
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and supplemental.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We did include error bars where possible, e.g., see Section 4.1.2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplemental.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4
   (b) Did you mention the license of the assets? [Yes] See supplemental
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See supplemental
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]