Semi-Weakly-Supervised Learning of Complex Actions from Instructional Task Videos

Yuhan Shen
Northeastern University
shen.yuh@northeastern.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

Abstract

We address the problem of action segmentation in instructional task videos with a small number of weakly-labeled training videos and a large number of unlabeled videos, which we refer to as Semi-Weakly-Supervised Learning (SWSL) of actions. We propose a general SWSL framework that can efficiently learn from both types of videos and can leverage any of the existing weakly-supervised action segmentation methods. Our key observation is that the distance between the transcript of an unlabeled video and those of the weakly-labeled videos from the same task is small yet often nonzero. Therefore, we develop a Soft Restricted Edit (SRE) loss to encourage small variations between the predicted transcripts of unlabeled videos and ground-truth transcripts of the weakly-labeled videos of the same task. To compute the SRE loss, we develop a flexible transcript prediction (FTP) method that uses the output of the action classifier to find both the length of the transcript and the sequence of actions occurring in an unlabeled video. We propose an efficient learning scheme in which we alternate between minimizing our proposed loss and generating pseudo-transcripts for unlabeled videos. By experiments on two benchmark datasets, we demonstrate that our approach can significantly improve the performance by using unlabeled videos, especially when the number of weakly-labeled videos is small.[1].

1. Introduction

Many of humans everyday tasks are procedural, where a task consists of a sequence of actions that must be followed to achieve the desired goal. Therefore, there has been an explosion of instructional videos on the web, teaching how to perform tasks, such as cooking recipes, repairing devices, assembling furnitures, performing emergency first aid, etc. [1, 10, 15, 23, 38, 56, 66, 67]. Automatic learning of procedural tasks from instructional videos has important applications, such as teaching intelligent agents to perform complex tasks, constructing large knowledge bases of compact instructions, and automatic performance evaluation for executing tasks. Over the past several years, we have seen great advances on different aspects of learning from instructions [1, 4, 8, 15, 19, 20, 33, 35–37, 49, 50, 56, 66, 67].

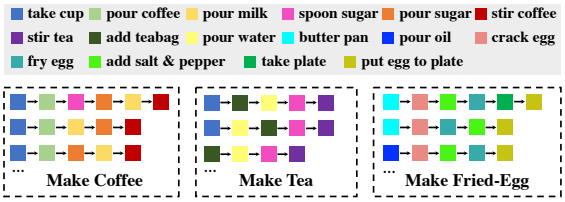

Figure 1. Transcript variation within each task for three different tasks.

A major challenge in learning from instructional videos is that videos are long and have many actions, therefore annotation is costly and complex. This poses a major challenge for scaling the learning to a large number of tasks and videos. Therefore, while a few fully-supervised methods have studied learning from densely-annotated videos [24, 27, 45, 49, 51, 53, 64, 66], the majority of existing works have focused on using less supervision. Specifically, weakly-supervised methods assume that each training video is accompanied with its transcript (ordered list of actions) [5, 7, 12, 30, 35, 44, 67] or action-set (unordered list of actions) [16, 31, 32, 36, 43]. While using weak supervision reduces the annotation cost by removing the need for specifying temporal boundaries of actions, it still requires annotators to watch entire videos. On the other hand, unsupervised methods remove the need for annotation by using unlabeled videos and leveraging the similarity of videos of the same task. However, existing similarity constraints, e.g., videos following the same sequence of actions or the same pairwise action ordering, are limiting and often violated in videos (see Figure 1). This has led to performance of unsupervised methods significantly lagging behind that of the weakly-supervised algorithms.

Paper Contributions. Motivated by the above discussion, we study a new action segmentation problem in which we assume having access to a small number of weakly-labeled

[1]Code available at https://github.com/Yuhan-Shen/SWSL.

training videos and a large number of unannotated videos (with only task labels) from multiple tasks. We refer to this setting as Semi-Weakly-Supervised Learning (SWSL) of actions, whose goal is to learn a video segmentation model using both types of training videos. Using unlabeled videos allows us to effectively regularize learning from a small number of weakly-labeled videos, which would be insufficient for learning action segmentation/classifier using current methods. On the other hand, using weakly-labeled videos allows us to guide learning from unlabeled videos by leveraging a few transcripts of each task.

We propose an SWSL method to find the parameters of a video feature learning module and an action classifier by simultaneously learning from weakly-labeled and unlabeled videos. Our key observation is that transcripts of unlabeled videos often have small but nonzero distances to the transcripts of the weakly-labeled videos from the same task, accounting for small variations by which the task could be accomplished. Therefore, we develop a differentiable Soft Restricted Edit (SRE) loss, which allows us to predict a transcript for an unlabeled video that is close to ground-truth transcripts of the weakly-labeled videos of the same task, yet could be different from them. To compute the SRE loss, we develop a flexible transcript prediction (FTP) method that uses the output of the action classifier to find both the length of the transcript and the sequence of actions occurring in an unlabeled video. Motivated by prior works on self-training [28, 62, 63], we propose a learning scheme in which we alternate between i) minimizing our proposed loss (sum of the weakly-supervised and SRE losses) on both types of videos; ii) adding a few most confident unlabeled videos and their pseudo-transcripts to the weakly labeled set. An advantage of our method is that it can use any existing weakly-supervised method. By experiments on two benchmark datasets of Breakfast [23] and CrossTask [67], we demonstrate the effectiveness of our approach.

2. Related Works

Action Segmentation. Depending on the supervision type, existing works on action segmentation in instructional videos can be divided into three categories. First, fully-supervised methods assume that frame-wise annotations of actions in videos are given [24, 27, 45, 49, 51, 53, 64, 66]. Second, weakly-supervised methods assume each training video comes with an ordered or unordered list of its actions [5, 7, 30, 31, 35, 36, 43, 44, 67] or its summary [39, 60]. Third, unsupervised learning methods exploit the common structure, cross-modality consistency or temporal information of videos of the same task to discover and localize actions [14, 15, 18, 25, 48, 50]. In this paper, we propose the new setup of semi-weakly-supervised learning from instructional videos, which has not been explored yet.

Weakly-supervised action segmentation methods mostly use the transcripts to learn a mapping from video features to framewise action class probabilities, so the major difference among prior works is the choice of mapping functions and loss functions. In the paper, we leverage two existing weakly-supervised methods [30, 54]. Specifically, [30] uses a GRU layer with a fully-connected layer as the mapping function, while [54] uses a deep convolutional neural network. As for the loss functions, [30] uses a constrained discriminative forward loss (CDFL) to distinguish the valid frame labelings, consistent with the ground-truth transcripts, from invalid labelings. [54] has a module to predict the class and length of segments and uses the mutual consistency (MuCon) loss to enforce the consistency of the frame-wise probabilities and predicted segments.

Semi-supervised learning (SSL) approaches aim to learn from both labeled and unlabeled data [21, 34, 41, 46, 61, 63, 65]. In video understanding, SSL has been studied for temporal action proposals, human pose estimation, salient object detection, action recognition, etc. [42, 52, 59, 62] There are two major directions in SSL: self-training and consistency regularization. Self-training based methods [28, 62, 63] first train a model using supervised methods and then predict pseudo-labels for unlabeled data. Consistency regularization, first proposed by [3] and extended in several works including temporal ensembling [26] and mean teacher [57], minimizes the discrepancy between predictions of perturbed input data.

Sequence Alignment. Our work is related to sequence alignment. Dynamic Time Warping (DTW) is a classic algorithm to measure the distance between two temporal sequences [47]. Recent variants of DTW include differentiable approximations [9, 19] and allowing skipping outliers [13]. Weak sequence alignment algorithm (WSA) [50] performs one-to-one alignment while allowing some items to be unmatched, and is extended to be differentiable. However, all of these works require the alignment to strictly obey a temporal order. Order-preserving Wasserstein distance [55] can tackle local temporal distortion via optimal transport, but it is multiple-to-multiple or one-to-multiple alignment and not differentiable. Our work is motivated by edit distance, which measures the distance between two strings. Levenshtein distance [2,29] is a specific-type of edit distance that allows deletion, insertion and substitution, and can be computed by Needleman-Wunsch [40] algorithm. [22] extends the inputs of Needleman-Wunsch algorithm from strings to time series and makes it differentiable, but it does not allow adjacent transposition. Restricted Edit Distance and Damerau–Levenshtein distance [11, 17] allow the transposition of two adjacent characters, but neither of them are differentiable. Our proposed SRE loss is an extension of restricted edit distance, which can be applied to temporal sequences and is differentiable.

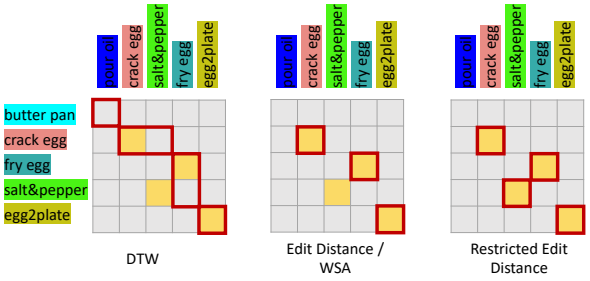

Figure 2. Alignment of two transcripts by different methods.

Illustrative Example: To better highlight the difference between the Restricted Edit distance and other sequence alignment methods, Figure 2 shows the alignment between two transcripts by different methods. DTW strictly aligns every entry in the two sequences, leading to many false alignments. While Edit Distance or WSA can obtain a one-to-one alignment and skip some unmatched items, they strictly follow temporal ordering and cannot handle the transposition from "*crack egg*" to "*add salt and pepper*". In contrast, the Restricted Edit distance can handle outlier elements and the transposition of adjacent items.

3. Problem Statement

In the semi-weakly-supervised learning of actions, we assume there are N weakly-labeled videos, $\{\boldsymbol{X}_n^w\}_{n=1}^N$, and M unlabeled videos, $\{\boldsymbol{X}_m^u\}_{m=1}^M$. The videos come from multiple tasks and each video consists of a sequence of actions required to achieve the underlying task. Let O denote the number of tasks and A denote the number of action classes across all videos. Our goal is to learn a model that segments a test video into different actions and recognizes the action of each segment and the underlying task of the video.

More specifically, for weakly-labeled training videos, we assume we have triplets $\{(\boldsymbol{X}_n^w, \boldsymbol{G}_n^w, y_n^w)\}_{n=1}^N$ of video features, transcripts and task labels,

$$\begin{aligned} \boldsymbol{X}_n^w &= \big(\boldsymbol{x}_{n,1}^w, \boldsymbol{x}_{n,2}^w, \dots, \boldsymbol{x}_{n,T_n^w}^w\big), \\ \boldsymbol{G}_n^w &= \big(\boldsymbol{g}_{n,1}, \boldsymbol{g}_{n,2}, \dots, \boldsymbol{g}_{n,L_n}\big), \\ y_n^w &\in \{1, \dots, O\}, \end{aligned} \tag{1}$$

where $\boldsymbol{x}_{n,i}^w \in \mathbb{R}^d$ is the d-dimensional feature of the i-th frame in the n-th weakly-labeled video and T_n^w is the length of the n-th video. Also, $\boldsymbol{G}_n^w$ is the video transcript (weak label), with the one-hot encoding $\boldsymbol{g}_{n,l} \in \{0,1\}^A$ denoting the l-th action in the n-th video, L_n is the length of the transcript, and y_n^w is the task label. For each task $o \in \{1, \dots, O\}$, we denote the set of transcripts of all weakly-labeled videos from the task by $\mathcal{G}_o$, i.e.,

$$\mathcal{G}_o = \{\boldsymbol{G}_n^w \,|\, \text{if } \; y_n^w = o, \forall n\}. \tag{2}$$

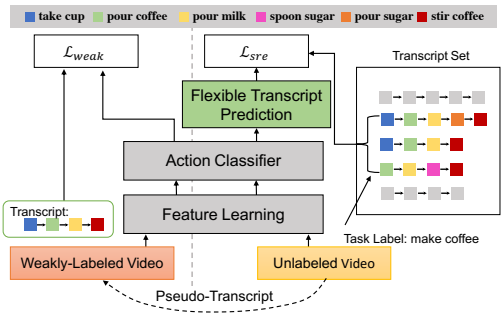

Figure 3. Our proposed framework for learning from both weakly-labeled and unlabeled instructional videos of multiple tasks.

On the other hand, for unlabeled videos, we have pairs $\{(\boldsymbol{X}_m^u, y_m^u)\}_{m=1}^M$ of video features and task labels,

$$\begin{aligned} \boldsymbol{X}_m^u &= \big(\boldsymbol{x}_{m,1}^u, \boldsymbol{x}_{m,2}^u, \dots, \boldsymbol{x}_{m,T_m^u}^u\big), \\ y_m^u &\in \{1, \dots, O\}, \end{aligned} \tag{3}$$

where $\boldsymbol{x}_{m,i}^u$ is the feature of the i-th frame in the m-th unlabeled video, and T_m^u is the length of the m-th video.

4. Semi-Weakly-Supervised Action Learning

4.1. Overview of Proposed Framework

We propose a general framework for jointly learning from (small) weakly-labeled and (large) unlabeled videos. As shown in Figure 3, our framework consists of two branches for learning from both types of training videos while using a shared action classifier. For weakly-labeled videos, in our proposed framework, we can flexibly use any existing weakly-supervised method. Let $\mathcal{L}_{weak}$ denote the associated loss, which is introduced in Sec. 4.2.

For unlabeled videos, given the video features as inputs, we use the action classifier to output a frame-wise probability matrix $\boldsymbol{P} \in [0,1]^{T \times A}$ that captures the probability of each frame belonging to each action. To predict the transcript of each unlabeled video, we propose a Flexible Transcript Prediction (FTP) algorithm that takes $\boldsymbol{P}$ as input and outputs the transcript of the video and the associated segmentation. We use the key observation that the predicted transcript of an unlabeled video should have a small distance to the transcripts of the weakly-labeled videos of the same task, corresponding to small variations by which a task could be accomplished. Therefore, we propose the Soft Restricted Edit (SRE) distance, a differentiable loss that allows insertion, deletion, substitution and adjacent transposition for computing the distance between the predicted transcript and the training transcript set. Thus, we can predict transcripts of unlabeled videos that are sufficiently close to those of the weakly-labeled videos, instead of enforcing the predicted transcript to coincide with the training transcripts.

As training proceeds, we generate pseudo-transcripts for the unlabeled videos and gradually add the videos with the most confident transcripts into the weakly-labeled set. This self-training strategy is introduced in Sec. 4.5.

4.2. Weakly-Supervised Action Segmentation

To learn from weakly-labeled training videos, whose set is denoted by $\mathcal{W}$, weakly-supervised action segmentation learns a mapping $\mathcal{F}_\Theta : \mathbb{R}^{T\times d} \to \mathbb{R}^{T\times A}$ from the features of each video to framewise action class probabilities, $\boldsymbol{P}$, and uses the provided transcript, $\boldsymbol{G}$, for supervision, i.e.,

$$\min_{\Theta} \sum_{\mathcal{W}} \mathcal{L}_{weak}(\boldsymbol{P}, \boldsymbol{G}), \text{ where } \boldsymbol{P} = \mathcal{F}_\Theta(\boldsymbol{X}).^{2} \quad (4)$$

As an advantage of our framework, we can leverage any existing weakly-supervised method and use unlabeled videos to significantly improve its performance, especially when the number of weakly-labeled videos is small. In the paper, we use two state-of-the-art methods, MuCon [54] and CDFL [30], which we reviewed in Sec. 2.

4.3. Flexible Transcript Prediction (FTP)

Given the framewise probability matrix $\boldsymbol{P} \in [0,1]^{T\times A}$ of a video, where T is the number of frames and A is the number of action classes, our goal is to predict the transcript probability, denoted by $\boldsymbol{Q} \in [0,1]^{L\times A}$, where L is the predicted transcript length, i.e., the number of actions in the video. The i-th row of $\boldsymbol{Q}$ indicates the probabilities that the entry i in the transcript will be each of the A actions. Notice that both the transcript length and the sequence of actions are unknown. Therefore, we propose a Flexible Transcript Prediction (FTP) algorithm to estimate both.

To tackle the problem, first notice that, given a fixed transcript length L, we can find the segmentation boundary points $(t_0, t_1, t_2, \dots, t_L)$ and the classes of segments $(a_1, a_2, \dots, a_L)$ by solving

$$\max_{\{t_i\},\{a_i\}} \prod_{i=1}^{L} \prod_{j=t_{i-1}+1}^{t_i} p_{j,a_i} \quad \text{s.t.} \quad t_0 = 0,\ t_L = T. \quad (5)$$

In other words, we simultaneously search for the segmentation and the assignment of each segment $(t_{i-1}+1, t_i)$ to an action class a_i that gives the maximum likelihood. Given that the transcript length, L, is itself unknown, we modify the problem to also search for a transcript of sufficiently small length. Thus, we add L as a penalty to the negative log-likelihood in (5) and solve

$$\min_{\{t_i\},\{a_i\},L} -\frac{1}{T}\sum_{i=1}^{L}\sum_{j=t_{i-1}+1}^{t_i} \log p_{j,a_i} + \lambda L \quad (6)$$
$$\text{s.t.} \quad t_0 = 0,\ t_L = T,\ \ell_{min} \le L \le \ell_{max},$$

[2]For simplicity of notation, we have dropped the subscript i when referring to $\boldsymbol{P}_i$ and $\boldsymbol{G}_i$ for $i \in \mathcal{W}$.

Algorithm 1: Flexible Transcript Prediction (FTP)

input : Probability matrix $\boldsymbol{P} \in [0,1]^{T\times A}$

1 Compute cumulative sum of negative log-likelihood: $s_{t,a} = \sum_{t'\le t} -\log q_{t',a}$;

2 Compute the minimal cost of each segment: $c_{t_1,t_2} = \min_a(s_{t_2,a} - s_{t_1-1,a}), 1 \le t_1 \le t_2 \le T$;

3 Dynamic program: $d_{t,l} = \min_{t'<t}(d_{t',l-1} + c_{t'+1,t})$, $t \in \{1,\dots,T\}, l \in \{1,\dots,\ell_{max}\}$;

4 Optimal length: $L = \text{argmin}_{\ell_{min}\le l\le \ell_{max}} \frac{d_{T,l}}{T} + \lambda l$;

5 Back-tracking: $t_L = T$, $t_{i-1} = \text{argmin}_{t<t_i}(d_{t,i-1} + c_{t+1,t_i}), i \in \{L,\dots,1\}$;

output: Segmentation boundary points $(t_0,\dots,t_L)$.

where (ℓ_{min}, ℓ_{max}) is a predefined range for the transcript length (which can be estimated from weakly-labeled videos), and λ is a regularization hyperparameter.

To solve (6), we develop a dynamic programming-based method, shown in Algorithm 1. For each possible segment (t_1, t_2), we define a cost $c_{t_1,t_2} = \min_a \sum_{t=t_1}^{t_2} -\log p_{t,a}$, which corresponds to the inner summation of the first term in (6) and represents the negative log-likelihood of assigning the segment to its most likely action. We can calculate c_{t_1,t_2} efficiently by precomputing the cumulative sum of negative log-likelihood. We then dynamically update $d_{t,l}$, which computes the minimal cost value for the first t frames and l segments, via step 3, where we find an optimal boundary point t' that minimizes the cost of splitting the first t' frames into $l-1$ segments and setting $(t'+1, t)$ as a segment. Here, $d_{T,l}$ denotes the minimal cost of splitting the video into l segments, so we find the optimal length L that minimizes (6). Finally, we recover the segmentation boundary points by back-tracking.

After solving (6), we compute the probability matrix $\boldsymbol{Q}$ of the predicted transcript using the geometric average of the probabilities within each predicted segment, i.e.,

$$q_{i,a} = \Big(\prod_{t=t_{i-1}+1}^{t_i} p_{t,a}\Big)^{\frac{1}{t_i - t_{i-1}}}. \quad (7)$$

Computational Complexity. In Algorithm 1, the complexity of step 1 is $\mathcal{O}(T^2)$, step 2 is $\mathcal{O}(T^2A)$, step 3 is $\mathcal{O}(T^2\ell_{max})$, step 4 is $\mathcal{O}(\ell_{max})$ and step 5 is $\mathcal{O}(L)$. Thus, the overall complexity is $\mathcal{O}(T^2(\ell_{max} + A))$. We can also significantly improve the complexity by a factor of $1/S^2$ through applying a temporal average pooling with stride S to $\boldsymbol{P}$, which we investigate in the experiments.

4.4. Soft Restricted Edit (SRE) Distance

We develop the differentiable Soft Restricted Edit (SRE) loss that finds the distance between two sequences while allowing the operations of insertion, deletion, substitution and adjacent transposition. This allows us to obtain

a predicted transcript for an unlabeled video, $\boldsymbol{Q} = (\boldsymbol{q}_1, \cdots, \boldsymbol{q}_L)$, that has a small distance to the transcripts of the weakly-labeled videos of the same task. More specifically, with $\mathcal{G}$ denoting the set of transcripts $\boldsymbol{G} = (\boldsymbol{g}_1, \cdots, \boldsymbol{g}_{L'}) \in \{0,1\}^{L' \times A}$ of the weakly-labeled videos of the same task, we propose to minimize $\mathcal{L}_{sre}$ defined as

$$\mathcal{L}_{sre}(\mathcal{G}, \boldsymbol{Q}) \triangleq \min_{\boldsymbol{G} \in \mathcal{G}} \mathrm{SRE}(\boldsymbol{G}, \boldsymbol{Q}). \tag{8}$$

To motivate our method for computing SRE loss, we start with the toy example of restricted edit distance between two strings, and then generalize it to arbitrary sequences.

Motivating Example. Consider the two strings of $\boldsymbol{x} =$ ('*p*', '*a*', '*r*', '*s*', '*e*') and $\boldsymbol{y} =$ ('*e*', '*r*', '*a*', '*s*', '*e*', '*r*'). Our goal is to find the least number of operations to convert one string into the other, where the admissible edit operations are *insertion, deletion, substitution* and *transposition of two adjacent characters*.[3] As shown in Figure 4 (left), the restricted edit distance between the two strings is 3.

To compute the restricted edit distance between two strings $\boldsymbol{x}$ and $\boldsymbol{y}$, of lengths L_x and L_y, we use dynamic programming. We compute a cumulative cost matrix $\boldsymbol{E} = [e_{i,j}] \in \mathbb{R}^{L_x+1 \times L_y+1}$ whose entry $e_{i,j}$ is the distance between the first $i-1$ entries of $\boldsymbol{x}$ and the first $j-1$ entries of $\boldsymbol{y}$, see Figure 4 (right). As a result, the last entry of $\boldsymbol{E}$, i.e., e_{L_x+1,L_y+1} corresponds to the restricted edit distance between the two sequences. We can recursively compute $e_{i,j}$ as the minimum among

$$\begin{cases} e_{i-1,j} + 1 \ \text{(deletion)} \\ e_{i,j-1} + 1 \ \text{(insertion)} \\ e_{i-1,j-1} + \mathbb{1}(\boldsymbol{x}_{i-1} \neq \boldsymbol{y}_{j-1}) \ \text{(substitution)} \\ e_{i-2,j-2} + 1, \text{if } (\boldsymbol{x}_{i-2} = \boldsymbol{y}_{j-1}, \boldsymbol{x}_{i-1} = \boldsymbol{y}_{j-2}) \ \text{(transposition)} \end{cases} \tag{9}$$

where $\mathbb{1}(\cdot)$ is an indicator function, which is one when its argument is true and is zero otherwise. Notice that the last entry in (9) is the cost of adjacent transposition. In our toy example, see Figure 4 (right), if we want to update the $(4,4)$-*th* entry of $\boldsymbol{E}$ (the blue box), we check if the 2*nd* and 3*rd* characters in the two strings are the same but swapped. Since they are, we move two steps back and take the value of the $(2,2)$-*th* entry (the orange box), which is 1, and increment it by one for the adjacent transposition cost.

SRE Forward Propagation. We make the restricted edit distance to be differentiable (for efficient classifier and video feature learning) and extend it to handle the more general case of computing distances between sequences of vectors. In our case, the two sequences are the ground-truth transcript of a weakly-labeled video, $\boldsymbol{G} = (\boldsymbol{g}_1, \cdots, \boldsymbol{g}_{L'})$, and the predicted transcript of an unlabeled video, $\boldsymbol{Q} = (\boldsymbol{q}_1, \cdots, \boldsymbol{q}_L)$. To achieve this, we modify (9) as

$$e_{i,j} = \min_\beta \begin{cases} e_{i-1,j} + c_D, \\ e_{i,j-1} + c_I, \\ e_{i-1,j-1} + \delta_{i-1,j-1}, \\ e_{i-2,j-2} + \delta_{i-2,j-1} + \delta_{i-1,j-2} + c_T \ (\forall i,j \geq 3), \end{cases} \tag{10}$$

where i) we change the cost of edit operations from 1 to positive constant costs of c_D, c_I, c_T for deletion, insertion and adjacent transposition, respectively; ii) for substitution, we use a continuous distance function δ between two vectors, instead of the indicator function; iii) to make the distance differentiable, we replace the minimum operation with soft-min, defined as $\min_\beta\{\alpha_1, \alpha_2, \ldots\} = -\beta \log \sum_k e^{-\alpha_k/\beta}$, where β is a smoothing hyperparameter. The Soft Restricted Edit (SRE) loss will be the last entry of cumulative matrix $\boldsymbol{E}$,

$$\mathrm{SRE}(\boldsymbol{G}, \boldsymbol{Q}) = e_{L'+1,L+1}. \tag{11}$$

Algorithm 2 summarizes the forward propagation steps for computing the SRE loss.

In the paper, we compute the distance $\delta_{i,j}$ between the entry i in the weakly-labeled transcript $\boldsymbol{G}$ and the entry j in the predicted transcript $\boldsymbol{Q}$ using the inner product

$$\delta_{i,j} = -\langle \boldsymbol{g}_i, \log \boldsymbol{q}_j \rangle, \tag{12}$$

which is the negative log-likelihood that the j-th action in the predicted transcript is the same as the i-th action in weakly-labeled transcript.

SRE Backward Propagation. Given that an input to the SRE distance is $\boldsymbol{Q}$, which is obtained using the framewise probabilities from the action classifier, not only we can find the best alignment between the predicted transcript and weakly-labeled transcripts, but also supervise the learning of video features and action classifier to get a better alignment. Hence, to update Θ (parameters of the feature learning and classifier modules), we need to compute

$$\nabla_\Theta \mathrm{SRE}(\boldsymbol{G}, \boldsymbol{Q}) = \left(\frac{\partial \boldsymbol{Q}}{\partial \Theta}\right)^T \nabla_{\boldsymbol{Q}} \mathrm{SRE}(\boldsymbol{G}, \boldsymbol{Q}). \tag{13}$$

Let $J \triangleq \mathrm{SRE}(\boldsymbol{G}, \boldsymbol{Q})$. We first define two intermediate variables $r_{i,j} = \frac{\partial J}{\partial \delta_{i,j}}$ and $h_{i,j} = \frac{\partial J}{\partial e_{i,j}}$. As we show in the

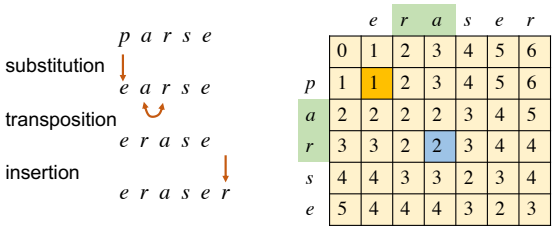

Figure 4. Left: restricted edit distance between two strings. Right: Cumulative cost matrix. The last entry is the distance between sequences.

[3]The main difference between restricted edit distance and the edit distance [22, 29] is that we allow adjacent transposition.

Algorithm 2: Forward Propagation for SRE

input : Pairwise cost matrix $\boldsymbol{\Delta} = [\delta_{i,j}] \in \mathbb{R}^{L' \times L}$; $c_D, c_I, c_T, \beta \geq 0$.

$e_{i,1} = (i-1) \cdot c_D, i \in \{1, 2, \ldots, L'+1\}$

$e_{1,j} = (j-1) \cdot c_I, j \in \{2, \ldots, L+1\}$

for $i \leftarrow 2$ **to** $L'+1$ **do**

  for $j \leftarrow 2$ **to** $L+1$ **do**

    update $e_{i,j}$ via (10);

output: SRE Loss, $\mathcal{L}_{sre} = e_{L'+1,L+1}$

Algorithm 3: Backward Propagation for SRE

input : Pairwise cost $\boldsymbol{\Delta} = [\delta_{i,j}] \in \mathbb{R}^{L' \times L}$; cumulative cost $\boldsymbol{E} = [e_{i,j}] \in \mathbb{R}^{L'+1 \times L+1}$; $c_I, c_D, c_T, \beta \geq 0$.

$h_{L'+1,L+1} = 1$

for $i \leftarrow L'+1$ **to** 1 **do**

  for $j \leftarrow L+1$ **to** 1 **do**

    update $a_{i,j}, b_{i,j}, h_{i,j}$ via Eq. (13) in the supplementary material;

Update $r_{i,j}$ via (4.4) and set $\boldsymbol{R} = [r_{i,j}]$.

output: $\nabla_{\boldsymbol{Q}} \text{SRE}(\boldsymbol{G}, \boldsymbol{Q}) = \left(\frac{\partial \boldsymbol{\Delta}(\boldsymbol{G},\boldsymbol{Q})}{\partial \boldsymbol{Q}}\right)^T \boldsymbol{R}$

supplementary materials, we can efficiently compute $r_{i,j}$ using $h_{i,j}$, and two auxiliary gradient matrices with entries defined as $a_{i,j} = \frac{\partial e_{i+1,j+1}}{\partial e_{i,j}}$ and $b_{i,j} = \frac{\partial e_{i+2,j+2}}{\partial e_{i,j}}$, where

$$r_{i,j} = h_{i+1,j+1} \cdot a_{i,j} + h_{i+1,j+2} \cdot b_{i-1,j} + h_{i+2,j+1} \cdot b_{i,j-1}.$$

We also show that $h_{i,j}, a_{i,j}, b_{i,j}$ can be recursively updated starting from the last row and column of $\boldsymbol{E}$. Algorithm 3 shows the backward propagation for computing the gradient with respect to $\boldsymbol{Q}$. Once finished, we can back-propagate the gradient w.r.t. Θ by using (13).

4.5. Training and Inference

To jointly learn from both weakly-labeled and unlabeled videos, we propose to minimize

$$\mathcal{L}_{swsl} = \sum_n \mathcal{L}_{weak}(\boldsymbol{X}_n^w, \boldsymbol{G}_n^w) + \rho \sum_m \mathcal{L}_{sre}(\mathcal{G}_{y_m^u}, \boldsymbol{Q}_m^u), \quad (14)$$

over the parameters of the feature learning and action classifier modules as well as searching over the predicted transcripts, shown in Figure 3, where $\rho \geq 0$ is a hyper-parameter. The first term in (14) minimizes the distance between each weakly-labeled video and its ground-truth transcript, while the second term minimizes the distance between the predicted transcript of each unlabeled video and the set of weakly-labeled transcripts of the same task.

Motivated by works on self-training [28, 62, 63], we propose a training strategy which alternates between minimizing $\mathcal{L}_{swsl}$ and moving some of the unlabeled videos and their predicted transcripts to the weakly-supervised set. We show that this approach works better than optimizing (14) once. More specifically, we minimize $\mathcal{L}_{swsl}$ using both the current weakly-labeled videos and unlabeled videos. We then generate pseudo-transcripts for unlabeled videos and add the unlabeled videos with the most confident pseudo-transcripts to the weakly-labeled set (see below for details) and retrain the model by minimizing $\mathcal{L}_{swsl}$, and so on. In the last iteration, all unlabeled videos have been moved to the weakly-labeled set, therefore, we minimize $\mathcal{L}_{weak}$.

For self-training, given an unlabeled video, we compute $\mathcal{L}_{weak}$ between the video and all transcripts in the weakly-labeled set, and find the transcript $\boldsymbol{G}^* \in \{0,1\}^{L' \times A}$ with the minimum loss. We use the transcript probability $\boldsymbol{Q} \in [0,1]^{L \times A}$ produced by FTP, and compute the restricted edit distance between $\boldsymbol{G}^*$ and $\boldsymbol{Q}$ to find their alignment $\boldsymbol{M} \in \{0,1\}^{L' \times L}$, where $m_{i,j} = 1$ indicates that the i-th action in the transcript occurs in the j-th segment in $\boldsymbol{Q}$. We then generate the one-hot pseudo-transcript for the unlabeled video using $\tilde{\boldsymbol{Q}} = \boldsymbol{M}^T \boldsymbol{G}^*$. Finally, we choose a few videos with the smallest loss values and add them along with their pseudo-transcript to the weakly-labeled set.

Remark 1 *A conventional self-training approach has two major differences with our method. First, it only minimizes the first term of (14), i.e., $\rho = 0$. Second, it generates a pseudo-label for an unlabeled video using the most probable transcript from the training videos. However, this enforces the unlabeled videos to have exactly the same transcript as weakly-labeled videos, which is limiting, especially when the number of weakly-labeled videos is small. In the experiment, we show that this strategy does not perform well compared to our method.*

5. Experiments

5.1. Experimental Setup

Datasets. For our experiments, we use two benchmark datasets of *Breakfast* [23] and *CrossTask* [67]. *Breakfast* consists of 1,712 videos of people making breakfast with 10 cooking tasks. The dataset contains 48 actions in total and on average, about 7 action instances are performed in each video. We use the existing 4 training/testing data splits in the dataset and report the average performance on the 4 splits. *CrossTask* consists of videos from 18 primary tasks. We follow [35] and use the 14 cooking-related tasks, which contains 2,552 videos and 80 classes of actions. We use the same training/testing split as in [35], with 90% training and 10% testing videos. For each dataset, we randomly split the training set into two subsets: one subset with weak labels (action transcripts and task labels), and the other subset with only task labels, and evaluate the performance on testing set. We investigate the effect of using different ratios of the number of weakly-labeled videos to the total number of videos used, including 1%, 2%, 5%, 10%, 20%.

Evaluation. For a comprehensive analysis of the results, we report multiple evaluation metrics. Following most works in action segmentation [25, 30, 35, 54], we compute Mean over Frame (MoF), i.e., frame-wise accuracy, and intersection over union (IoU). Since in *CrossTask*, over 70% of frames are background, simply predicting all frames as background will lead to a high MoF. Thus, to avoid this undesired effect, we also report the F1-score [14, 50] on *CrossTask*.

Implementation Details. To show that our proposed framework can leverage any existing weakly-supervised action segmentation methods, we use MuCon [54] and CDFL [30]. To be consistent with prior works, on *Breakfast*, we use the 2048-dimensional RGB+Flow I3D features [6] for MuCon, and the 64-dimensional improved dense trajectory features [58] for CDFL. On *CrossTask*, we use the 3200-dimensional released features [67] for MuCon and, following [35], reduce the feature dimension to 64 via PCA for CDFL. The average number of frames on *Breakfast* is over 2000, so we apply a temporal average pooling with stride 8 before inputting the frame-wise probabilities into our Flexible Transcript Prediction (FTP) module. We start the training for several epochs using weakly-labeled videos, and then use both weakly-labeled and unlabeled videos to optimize (14). We generate pseudo-transcripts for the unlabeled videos in three rounds, each round adding 1/3 of the unlabeled videos and their learned pseudo-transcripts to the weakly-labeled set. We keep the same setups as in the original works on MuCon and CDFL [30, 54]. Due to space limitation, other details such as hyperparameter values are reported in the supplementary materials.

Baselines. We mainly compare four methods: i) Weakly-Supervised Learning (WSL): we apply $\mathcal{L}_{weak}$ (MuCon or CDFL loss) on weakly-labeled videos only; ii) Semi-Weakly-Supervised Learning (SWSL): we apply $\mathcal{L}_{swsl}$ for all videos without generating pseudo-transcripts (i.e., without self-training); iii) WSL+Self: we use self-training in weakly-supervised methods, where we minimize only $\mathcal{L}_{weak}$ and iteratively generate pseudo-transcripts of unlabeled videos and add them to the weakly-labeled set and re-train the model; iv) SWSL+Self: our final approach, where we minimize our proposed $\mathcal{L}_{swsl}$ loss, while iteratively increasing the weakly-labeled set using self-training.

5.2. Experimental Results

Action Segmentation Performance. Tables 1 and 2 show the average performances of different methods on Breakfast and CrossTask when we use, respectively, MuCon [54] and CDFL [30] as the backbone weakly-supervised module in our framework. First, notice that using unlabeled videos always improves the performance of WSL. In particular, in Table 1, when the number of labeled videos is extremely small (1%), our method (SWSL+Self) improves the MoF by 14.0%, IoU by 16.3% on Breakfast, and MoF by 9.9%, IoU by 3.3% and F1 by 6.3% on CrossTask. Additionally, SWSL+Self significantly improves over WSL+Self, e.g., improves MoF by 2.7%, IoU by 3.7% on Breakfast, and MoF by 7.6%, IoU by 1.9%, F1 by 0.9% on CrossTask. This shows the effectiveness of our proposed $\mathcal{L}_{sre}$ loss and simultaneously learning the model from both weakly-labeled and unlabeled videos. Finally, notice that combining self-training with our method (SWSL+Self vs SWSL) improves MoF by 1.9%, IoU by 8.0% on Breakfast, and MoF by 0.4%, IoU by 1.3%, F1 by 2.6% on CrossTask.

	WP	UP	Breakfast		CrossTask		
			MoF	IoU	MoF	IoU	F1
WSL	1%	0	11.0	13.5	38.2	14.6	2.6
SWSL	1%	99%	23.1	21.8	47.7	16.6	6.3
WSL+Self	1%	99%	22.3	26.1	40.5	16.0	8.0
SWSL+Self	1%	99%	**25.0**	**29.8**	**48.1**	**17.9**	**8.9**
WSL	2%	0	12.9	14.8	44.0	15.8	5.3
SWSL	2%	98%	21.2	23.5	**52.3**	**18.9**	8.3
WSL+Self	2%	98%	26.5	27.8	41.7	16.0	9.9
SWSL+Self	2%	98%	**26.7**	**30.6**	44.6	17.8	**11.3**
WSL	5%	0	23.1	25.8	42.3	16.1	8.3
SWSL	5%	95%	28.1	23.5	**51.3**	18.0	10.3
WSL+Self	5%	95%	**32.7**	31.0	45.3	16.5	11.4
SWSL+Self	5%	95%	32.5	**31.7**	50.6	**18.3**	**11.5**
WSL	10%	0	28.0	28.8	42.1	16.7	9.9
SWSL	10%	90%	33.9	31.2	48.3	17.4	9.7
WSL+Self	10%	90%	**36.7**	32.1	45.0	16.5	11.6
SWSL+Self	10%	90%	36.3	**33.4**	**49.0**	**18.0**	**12.1**
WSL	20%	0	35.2	33.4	44.4	17.7	11.0
SWSL	20%	80%	36.7	33.6	46.7	17.3	10.7
WSL+Self	20%	80%	38.6	34.7	46.3	17.0	11.5
SWSL+Self	20%	80%	**39.8**	**36.1**	**54.5**	**19.3**	**11.8**
WSL	*100%*	*0*	*48.5*†	*39.1**	*48.4**	*21.0**	*16.7**

Table 1. Performance on Breakfast and CrossTask, when using MuCon as the backbone weakly-supervised module in our framework. (WP: **W**eakly-labeled video **P**ercentage. UP: **U**nlabeled video **P**ercentage. † reported in [54]; ∗ obtained in our experiments.)

Transcript Prediction. Table 3 shows the normalized edit distance (smaller is better) between the predicted transcript and the ground-truth transcript on the test set. Notice that our SWSL+Self method performs better than WSL+Self in all cases. Given that the difference between the two methods is in using our proposed $\mathcal{L}_{sre}$ loss, the improvement shows that encouraging the transcript of unlabeled videos to have a small distance to the transcripts of weakly-labeled videos (instead of enforcing them to coincide) is beneficial for transcript prediction, especially when the number of labeled videos is small. Besides, the improvement is more remarkable on CrossTask than Breakfast, because CrossTask has more diverse transcripts and benefits more from allowing flexible transcripts.

Comparison with Soft Edit distance. Table 4 compares the performance of our proposed Soft Restricted Edit (SRE) distance with Soft Edit (SE) distance, which does not allow adjacent transposition. Notice that, generally, SRE performs better than SE, especially when the number of weakly-labeled videos is small. This comes from the fact

	WP	UP	Breakfast		CrossTask		
			MoF	IoU	MoF	IoU	F1
WSL	1%	0	10.9	16.9	20.7	8.6	3.0
SWSL	1%	99%	13.4	20.4	24.5	**10.7**	4.4
WSL+Self	1%	99%	18.6	21.8	**27.8**	10.6	9.2
SWSL+Self	1%	99%	**32.4**	**29.5**	21.8	9.2	**9.9**
WSL	2%	0	10.9	17.4	20.5	8.6	5.3
SWSL	2%	98%	16.3	22.4	**26.9**	**11.4**	7.8
WSL+Self	2%	98%	27.6	26.2	22.3	9.6	9.5
SWSL+Self	2%	98%	**35.4**	**30.0**	21.4	9.1	**10.1**
WSL	5%	0	13.4	19.7	20.4	8.7	5.1
SWSL	5%	95%	24.4	25.7	**23.5**	**10.4**	7.9
WSL+Self	5%	95%	37.1	**31.8**	20.7	8.8	8.3
SWSL+Self	5%	95%	**39.6**	31.3	22.6	9.1	**11.3**
WSL	10%	0	20.4	20.9	23.2	9.0	7.8
SWSL	10%	90%	24.7	24.2	23.2	**9.8**	9.0
WSL+Self	10%	90%	38.3	31.4	20.5	8.5	10.0
SWSL+Self	10%	90%	**40.4**	**32.4**	**24.0**	9.3	**11.7**
WSL	20%	0	31.7	26.4	23.6	9.0	8.1
SWSL	20%	80%	33.9	28.9	22.6	**9.4**	10.9
WSL+Self	20%	80%	42.3	33.0	22.0	8.5	12.4
SWSL+Self	20%	80%	**43.5**	**33.0**	**24.8**	9.0	**13.2**
WSL	*100%*	*0*	*50.2*†	*35.9**	*31.5**	*13.2**	*18.8**

Table 2. Performance on Breakfast and CrossTask, when using CDFL as the backbone weakly-supervised module in our framework. (WP: **W**eakly-labeled video **P**ercentage. UP: **U**nlabeled video **P**ercentage. † reported in [30]; ∗ obtained in our experiments.)

WP	Breakfast		CrossTask	
	WSL+Self	SWSL+Self	WSL+Self	SWSL+Self
1%	0.437	**0.422**	0.538	**0.499**
2%	0.419	**0.403**	0.510	**0.468**
5%	0.358	**0.335**	0.488	**0.458**
10%	0.364	**0.318**	0.481	**0.424**
20%	0.322	**0.312**	0.451	**0.406**

Table 3. Normalized edit distance (smaller is better) between the predicted transcript and ground-truth transcript on the test set for MuCon.

WP	MoF		IoU	
	SE	SRE	SE	SRE
1%	24.3	**25.0**	29.2	**29.8**
2%	25.3	**26.7**	29.9	**30.6**
5%	31.1	**32.5**	30.8	**31.7**
10%	33.5	**36.3**	**33.4**	**33.4**
20%	39.2	**39.8**	35.8	**36.1**

Table 4. Comparison on Soft Edit distance (disallow adjacent transposition) and our proposed Soft Restricted Edit distance (allow adjacent transposition) for MuCon on Breakfast with respect to different **W**eakly-labeled video **P**ercentage (WP).

that when we have fewer labeled videos, the size of the weakly-labeled transcript set would be small, therefore, it is more likely that the order of adjacent actions in unlabeled videos would be different from that in the transcript set.

Training Effect. Figure 5 shows the performance of two methods (SWSL+Self and SWSL) on the Breakfast test set as a function of training epochs. We train the model using weakly-labeled videos for 60 epochs and then add unlabeled videos for training. For SWSL+Self, we generate pseudo-transcripts for unlabeled videos and add a subset of them to the weakly-labeled set in epoch 80, 100, 120. Notice that at epoch 60, adding unlabeled videos significantly improves the performance. For SWSL (right), the model converges

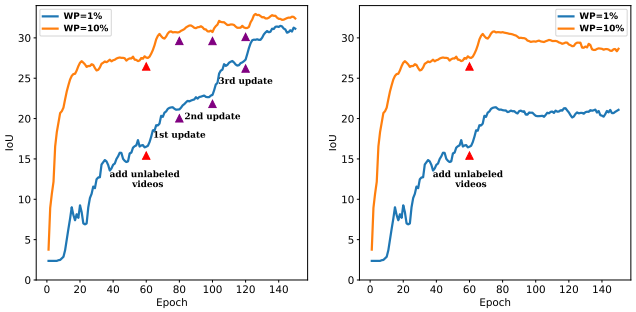

Figure 5. IoU of different methods on the Breakfast test set as a function of the number of training epochs. Left: SWSL+Self. Right: SWSL.

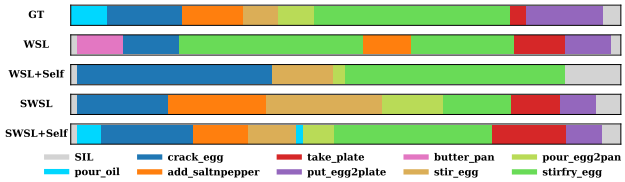

Figure 6. Segmentation results for a test video from the task 'make scrambled egg' in the Breakfast dataset.

or even overfits after epoch 80. But if we generate pseudo-transcripts for unlabeled videos (left), the performance still improves after each update. Besides, the improvement is more significant when we have 1% weakly-labeled videos, which shows the effectiveness of our method in the case of an extremely small number of weakly-labeled videos. See supplementary materials for the plots of other methods.

Qualitative Results. Figure 6 shows the segmentation of a test video from 'make scrambled egg' by different methods. This is a challenging case where the transcript of the test video is not present in the weakly-labeled training videos. WSL or WSL+Self will produce transcripts very different from the ground-truth, while the transcript predicted by SWSL or SWSL+Self is more close to the ground-truth. Furthermore, compared with SWSL, the boundary localization of SWSL+Self is more accurate, which shows the advantages of self-training.

6. Conclusions

We studied the new problem of semi-weakly supervised action learning from instructional videos. We proposed a Soft Restricted Edit distance that leverages unlabeled videos for training by encouraging the transcript of unlabeled videos to be close, yet possibly different, from those of the weakly-labeled videos of the same task. Our experiments on two datasets showed that our proposed framework significantly improves the performance.

Acknowledgements

This work is sponsored by DARPA PTG (HR00112220001), NSF (IIS-2115110), ARO (W911NF2110276) and ONR (N000141812132). Content does not necessarily reflect the position/policy of the Government. No official endorsement should be inferred.

References

[1] J. B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[2] Alexandr Andoni and Krzysztof Onak. Approximating edit distance in near-linear time. *SIAM Journal on Computing*, 2012. 2

[3] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 2014. 2

[4] Jing Bi, Jiebo Luo, and Chenliang Xu. Procedure planning in instructional videos via contextual modeling and model-based policy learning. *IEEE International Conference on Computer Vision*, 2021. 1

[5] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. *European Conference on Computer Vision*, 2014. 1, 2

[6] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 7

[7] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2

[8] Xiaobin Chang, Frederick Tung, and Greg Mori. Learning discriminative prototypes with dynamic time warping. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1

[9] M. Cuturi and M. Blondel. Soft-dtw: a differentiable loss function for time-series. *International Conference on Machine learning*, 2017. 2

[10] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[11] Fred J Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 1964. 2

[12] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1

[13] Nikita Dvornik, Isma Hadji, Konstantinos G Derpanis, Animesh Garg, and Allan D Jepson. Drop-dtw: Aligning common signal between sequences while dropping outliers. *Neural Information Processing Systems*, 2021. 2

[14] E. Elhamifar and D. Huynh. Self-supervised multi-task procedure learning from instructional videos. *European Conference on Computer Vision*, 2020. 2, 7

[15] E. Elhamifar and Z. Naing. Unsupervised procedure learning via joint dynamic summarization. *International Conference on Computer Vision*, 2019. 1, 2

[16] Mohsen Fayyaz and Jurgen Gall. Sct: Set constrained temporal transformer for set supervised action segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[17] Ryan Gabrys, Eitan Yaakobi, and Olgica Milenkovic. Codes in the damerau distance for deletion and adjacent transposition correction. *IEEE Transactions on Information Theory*, 2017. 2

[18] Karan Goel and Emma Brunskill. Learning procedural abstractions and evaluating discrete latent temporal structure. *International Conference on Learning Representation*, 2019. 2

[19] Isma Hadji, Konstantinos G. Derpanis, and Allan D. Jepson. Representation learning via global temporal alignment and cycle-consistency. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2

[20] Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram N. Syed, Andrey Konin, M. Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1

[21] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. *Neural Information Processing Systems*, 2019. 2

[22] Satoshi Koide, Keisuke Kawano, and Takuro Kutsuna. Neural edit operations for biological sequences. *Advances in Neural Information Processing Systems*, 2018. 2, 5

[23] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human. *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1, 2, 6

[24] H. Kuehne, J. Gall, and T. Serre. An end-to-end generative framework for video segmentation and recognition. *IEEE Winter Conference on Applications of Computer Vision*, 2016. 1, 2

[25] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 7

[26] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *International Conference on Learning Representations*, 2017. 2

[27] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[28] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on challenges in representation learning, ICML*, 2019. 2, 6

[29] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10, 1966. 2, 5

[30] J. Li, P. Lei, and S. Todorovic. Weakly supervised energy-based learning for action segmentation. *International Conference on Computer Vision*, 2019. 1, 2, 4, 7, 8

[31] Jun Li and Sinisa Todorovic. Set-constrained viterbi for set-supervised action segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[32] J. Li and S. Todorovic. Anchor-constrained viterbi for set-supervised action segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1

[33] Zhe Li, Yazan Abu Farha, and Jurgen Gall. Temporal action segmentation from timestamp supervision. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1

[34] Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff. Deep contextualized acoustic representations for semi-supervised speech recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020. 2

[35] Z. Lu and E. Elhamifar. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. *International Conference on Computer Vision*, 2021. 1, 2, 6, 7

[36] Z. Lu and E. Elhamifar. Set-supervised action learning in procedural task videos via pairwise order consistency. *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2

[37] 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. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[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. *International Conference on Computer Vision*, 2019. 1

[39] Z. Naing and E. Elhamifar. Procedure completion by learning from partial summaries. *British Machine Vision Conference*, 2020. 2

[40] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 1970. 2

[41] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. *IEEE International Conference on Computer Vision*, 2015. 2

[42] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[43] A. Richard, H. Kuehne, and J. Gall. Action sets: Weakly supervised action segmentation without ordering constraints. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2

[44] A. Richard, H. Kuehne, A. Iqbal, and J. Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2

[45] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1, 2

[46] Devendra Singh Sachan, Manzil Zaheer, and Ruslan Salakhutdinov. Revisiting lstm networks for semi-supervised text classification via mixed objective function. *AAAI Conference on Artificial Intelligence*, 2019. 2

[47] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26, 1978. 2

[48] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[49] Fadime Sener and Angela Yao. Zero-shot anticipation for instructional activities. *International Conference on Computer Vision*, 2019. 1, 2

[50] Y. Shen, L. Wang, and E. Elhamifar. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 7

[51] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta. Asynchronous temporal fields for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[52] Ankit Singh, Omprakash Chakraborty, Ashutosh Varshney, Rameswar Panda, Rogerio Feris, Kate Saenko, and Abir Das. Semi-supervised action recognition with temporal contrastive learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[53] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for finegrained action detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2

[54] Yaser Souri, Mohsen Fayyaz, Luca Minciullo, Gianpiero Francesca, and Juergen Gall. Fast Weakly Supervised Action Segmentation Using Mutual Consistency. *PAMI*, 2021. 2, 4, 7

[55] Bing Su and Gang Hua. Order-preserving wasserstein distance for sequence matching. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[56] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[57] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 2017. 2

[58] H. Wang and C. Schmid. Action recognition with improved trajectories. *International Conference on Computer Vision*, 2013. 7

[59] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Changxin Gao, and Nong Sang. Self-supervised learning for

semi-supervised temporal action proposal. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[60] C. Xu and E. Elhamifar. Deep supervised summarization: Algorithm and application to learning instructions. *Neural Information Processing Systems*, 2019. 2

[61] Jie Yan, Yan Song, Li-Rong Dai, and Ian McLoughlin. Task-aware mean teacher method for large scale weakly labeled semi-supervised sound event detection. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020. 2

[62] Pengxiang Yan, Guanbin Li, Yuan Xie, Zhen Li, Chuan Wang, Tianshui Chen, and Liang Lin. Semi-supervised video salient object detection using pseudo-labels. *IEEE International Conference on Computer Vision*, 2019. 2, 6

[63] Hai Ye and Lu Wang. Semi-supervised learning for neural keyphrase generation. *Empirical Methods in Natural Language Processing*, 2018. 2, 6

[64] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 2018. 1, 2

[65] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. *IEEE International Conference on Computer Vision*, 2019. 2

[66] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. *AAAI*, 2018. 1, 2

[67] D. Zhukov, J. B. Alayrac, R. G. Cinbis, D. Fouhey, I. Laptev, and J. Sivic. Cross-task weakly supervised learning from instructional videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 6, 7