Learning to Segment Actions from Visual and Language Instructions via Differentiable Weak Sequence Alignment

Yuhan Shen
Northeastern University
shen.yuh@northeastern.edu

Lu Wang
University of Michigan
wangluxy@umich.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

Abstract

We address the problem of unsupervised localization of task-relevant actions (key-steps) and feature learning in instructional videos using both visual and language instructions. Our key observation is that the sequences of visual and linguistic key-steps are weakly aligned: there is an ordered one-to-one correspondence between most visual and language key-steps, while some key-steps in one modality are absent in the other. To recover the two sequences, we develop an ordered prototype learning module, which extracts visual and linguistic prototypes representing key-steps. To find weak alignment and perform feature learning, we develop a differentiable weak sequence alignment (DWSA) method that finds ordered one-to-one matching between sequences while allowing some items in a sequence to stay unmatched. We develop an efficient forward and backward algorithm for computing the alignment and the loss derivative with respect to parameters of visual and language feature learning modules. By experiments on two instructional video datasets, we show that our method significantly improves the state of the art.

1. Introduction

Learning to perform procedural tasks by watching visual demonstrations or reading manuals is one of the complex capabilities of humans. Bringing this capability to machines allows us to design intelligent agents that autonomously learn to perform tasks or help humans/agents to achieve complex tasks and enables building massive instructional knowledge bases for education and autonomy. The explosion of data, on the other hand, has provided invaluable resources for automatic procedural task learning: there exist tens or hundreds of thousands of instructional videos on the web about how to cook different recipes, how to assemble or repair different devices, etc. [1, 49, 50, 32, 17, 45, 41, 25].

Given instructional videos of one or multiple tasks, the goal of procedure learning is to localize the key-steps (actions required to accomplish a task) in videos. Over the past several years, we have seen great advances on different aspects of learning from instructions [42, 1, 40, 21, 30, 36, 12, 34, 11, 31]. The majority of works address learning from weakly annotated videos [22, 3, 37, 38, 9, 50, 27, 7, 28], i.e., videos with ground-truth lists/sequences of key-steps or ground-truth summaries [47]. On the other hand, understanding instructional videos at the scale necessary to build large knowledge bases or assistive agents that respond to many instructions, requires unsupervised learning that does not require costly video annotations. This has motivated several works on unsupervised procedure learning [42, 40, 17, 26, 19, 16, 51, 1, 18, 15], which mostly rely on learning from visual data alone.

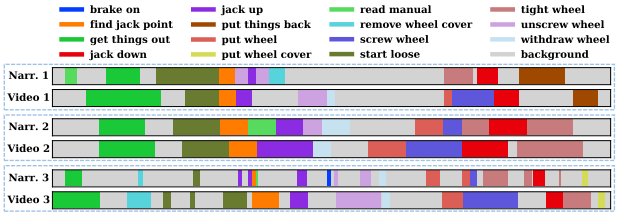


Figure 1: Key-steps in visual data and narrations for three videos from the task 'change tire'. Each color represents one key-step or background.

Learning from Visual and Language Instructions. Instructional videos are often accompanied with narrations, where visual demonstrations of many steps have language descriptions, see Figure 1. Indeed, these two modalities contain rich information that can be leveraged to more effectively discover key-steps. However, there are multiple challenges that we need to address to leverage this shared information. First, while the majority of key-steps appear in both modalities, some may only appear in visual data or narrations. For example, '*read manual*' appears in narration 1 but does not occur in the video, and '*screw wheel*' occurs in video 1 while being absent in the narration. Second, the two modalities are not necessarily aligned: for a visual demonstration of a key-step, the associated narration could happen before, during or after performing it (e.g., one may review one or a few steps using language before or after demonstrating them). Third, visual data and narrations

often contain substantial amount of background not related to the task, which do not necessarily occur at the same time[1] (see grey temporal regions in Figure 1). While few works have addressed unsupervised learning from both modalities [1, 42, 18], they rely on narration as the main modality or assume that visual and language descriptions have close temporal alignment. This limits their applicability to general cases where visual data and narrations are unaligned or when some key-steps are missing in one modality.

Paper Contributions. We address task-relevant action (key-step) localization and multimodal feature learning in instructional videos using visual and language data. Our key observation is that the *'sequences' of visual and linguistic key-steps are weakly aligned*. More specifically, there is a one-to-one correspondence between most visual and language key-steps, while some key-steps in one modality are absent in the other. Moreover, the ordering of the common key-steps in two modalities are similar. Thus, instead of assuming temporal alignment of key-steps in language and visual data, we assume weak alignment of key-step sequences once recovered, which allows for some steps to appear in only one modality and for the visual and language demonstration of a key-step to be temporally far.

To recover the sequences, we develop an *ordered prototype learning module*, which extracts visual and linguistic prototypes representing key-steps. On the other hand, to find weak alignment and perform feature learning, we develop a *differentiable weak sequence alignment (DWSA)* method that finds ordered one-to-one matching between sequences while allowing some items in a sequence to stay unmatched. We derive an efficient dynamic programming-based algorithm for computing the loss and alignment as well as an efficient backpropagation method for computing the gradient with respect to parameters of visual and language feature learning modules. By experiments on two instructional video datasets, we show that our method improves the state of the art by about 4.7% in F1 score.

2. Related Works

Procedure Learning. Existing works on learning from instructional videos can be divided into three categories. The first group of works assumes that annotations of key-steps in videos are given and the goal is to learn how to segment new videos [49] or anticipate future key-steps [41]. To reduce the costly and unscalable annotation requirement, the second group of works on weakly-supervised learning assumes that each video is accompanied with an ordered or unordered list of its key-steps, and the goals are to localize and learn models of key-steps in videos [22, 3, 37, 38, 9, 50, 27, 7, 28]. However, gathering error-free list of key-steps requires annotators to watch each video or manual intervention on noisy video meta data.

Unsupervised learning, which is the subject of our work, removes the annotation requirement by exploiting the common structure of videos to discover and localize key-steps. Many unsupervised methods have focused on learning from either narrations or visual demonstrations [40, 17, 26, 16, 12, 19, 14]. Hence, they cannot leverage the rich complementary information of the two modalities. On the other hand, [1, 30, 48, 42, 18] have addressed learning from multi-modal instructional data. However, they assume that each key-step appears in both modalities, rely on having (close) temporal alignment of a key-step in the two modalities, or mainly use narration to discover key-steps and then localize the discovered steps in visual data.

Sequence Alignment. Dynamic Time Warping (DTW) is a classic algorithm to measure the distance between two temporal sequences [39]. Cuturi and Blondel [8] extend DTW to a differentiable loss (soft-DTW) that enables training predictive and generative models for time series. Chang et al. [7] extend soft-DTW to a discriminative setting for weakly-supervised action segmentation. Cao et al. [5] propose an ordered temporal alignment module, using a variant of DTW, for few-shot video classification. However, all of the above works are based on one-to-many matchings and assume that each item in one sequence has a match in the other sequence. While [1] develops a Frank-Wolfe-based optimization algorithm for ordered alignment of multiple sequences, it does not allow for feature learning and is costly and initialization-dependent. [10] proposes a differentiable neural network for multiple sequence alignment, but is supervised and requires ground-truth alignments, which are not available in our setting. To the best of our knowledge, our DWSA is the first differentiable method that measures the cost of one-to-one alignment between sequences, allows some items in each sequence to be unmatched, and enables feature learning without access to ground-truth alignments. Our DWSA is to some extent similar to identifying regions of similarity among different genes [2].

Self-Supervised Representation Learning. Learning self-supervised video representations has become increasingly popular, due to the high cost of large-scale video annotations [33, 43, 6, 13, 35]. Our work is more related to self-supervised multimodal representation learning [32, 20, 44, 29, 31]. Some works assume that video clips and narrations are aligned [32] or close in time [31], and use such correspondences to train joint video-text embedding models. This could be limiting as demonstrations and narrations could be unaligned. Hu et al. [20] proposes a multi-modal clustering method to learn audio-visual embeddings, but the learned representations are fine-grained object-level representations and require aligned audio-image pairs.

[1] One might demonstrate two consecutive key-steps, while expressing opinions or advertising products in between narrations of the key-steps.

3. Learning to Segment Actions from Instructional Videos with Narrations

In this section, we develop a framework for unsupervised localization of key-steps and segmentation of instructional videos using visual and language data. It is worth mentioning that our framework can also handle using only visual data or narrations, as we show in the experiments.

3.1. Problem Statement

Assume we have N narrated videos from the same task. We denote the visual and language features of video n by

$$\mathcal{X}_n^v = \left(\boldsymbol{x}_{n,1}^v, \boldsymbol{x}_{n,2}^v, \dots\right), \quad \mathcal{X}_n^l = \left(\boldsymbol{x}_{n,1}^l, \boldsymbol{x}_{n,2}^l, \dots\right), \quad (1)$$

where $\boldsymbol{x}_{n,i}^v$ is the feature vector of segment i and $\boldsymbol{x}_{n,i}^l$ is the feature of verb-phrase i (notice that the number of segments and verb-phrases in a video could be different). Given that each segment or verb-phrase occurs during a time interval, we denote the middle time instant of the i-th segment and verb-phrase interval in the n-th video by $t_{n,i}^v$ and $t_{n,i}^l$, respectively. Our goal is to assign each video frame and verb-phrase to a key-step or background, hence, recover segmentation of videos and find frames and verb-phrases across videos that belong to the same key-step.

3.2. Proposed Framework

We model key-steps in visual data using visual prototypes $\{\boldsymbol{c}_k^v\}_{k=1}^{K_v}$ and model narration key-steps using linguistic prototypes $\{\boldsymbol{c}_k^l\}_{k=1}^{K_l}$. The number of visual and linguistic prototypes K_v and K_l are hyperparameters (in practice, we set $K_l > K_v$, since narration often contains more key-step descriptions). Our goals are to jointly learn the visual and linguistic prototypes, find their associations that result in recovering segmentations of videos, and learn representations that bring matched visual and linguistic prototypes closer. To do so, we propose a framework that consists of the following components, as shown in Figure 2: 1) a narration processing module that discovers verb-phrases from narrations and removes irrelevant ones; 2) visual-text[2] feature extraction and refinement; 3) two soft ordered prototype learning (SOPL) modules that learn visual and linguistic prototypes; 4) a differentiable weak sequence alignment (DWSA) loss that aligns the sequences of prototypes of two modalities and enables self-supervised feature learning.

3.2.1 Narration Processing

We use the subtitles automatically generated from YouTube to extract verb phrases. Following the pipeline in [11], we first adopt T-BRNN [46] to punctuate the subtitles. Next, we perform coreference resolution to resolve the pronouns via SpaCy[3]. We then run the dependency parser to discover verb phrases in the narrations. Unlike [1] that only keeps *verb+dobj* pairs, the format of our verb phrases is *verb+(prt)+dobj+(prep+pobj)*.[4] We keep more components because they are important to distinguish one key-step from another. For instance, the two key-steps of '*pass tie in front of knot*' and '*pass tie through knot*' in the task of 'tie a tie' lead to the same *verb+dobj* pair '*pass tie*', which is undesired. Finally, we remove some irrelevant verb-phrases that do not correspond to physical actions in videos by using their concreteness scores [4, 23]. The concreteness of a phrase is the highest concreteness of its words. We remove phrases with score lower than 3, e.g., '*keep interruptions to a minimum*' and '*avoid this problem*'. We also remove phrases that only contain stop words. Table 1 shows an example of extracted verb-phrases along with ground-truth key-steps. Notice that verb-phrases capture the majority of key-steps while still containing some noisy information and missing some key-steps.

Extracted verb phrases	key-steps in groundtruth
assemble your instrument	put case facing up
remove the reed	open case
put the thin end in your mouth	**put reed in mouth**
hold the lower section	
grease the cork	**grease corks**
twist the bell onto the lower section	**put on bell**
hold the upper section in your other hand	
twist the two sections	join lower joint and upper joint
twist the barrel onto the upper joint	
rest the bell in your leg	line up bridge key
attach the mouthpiece to the barrel	**put mouthpiece on barrel**
align the open flat side with a register key	
put the ligature	**put ligature on mouthpiece**
slip the reed with the flat side	
slide down the ligature	
center the reed on the mouthpiece	**put reed on mouthpiece**
center the reed with only a hair line	
touch the tip	
tighten the screws until snug	**tighten ligature screws**
ask your teacher for help	

Table 1: Extracted verb-phrases from a video of 'assemble clarinet'. Verb-phrases and steps in bold have similar semantics.

3.2.2 Visual+Text Feature Extraction and Learning

We extract unsupervised features from visual data and narrations and refine them using our method. We use the *unsupervised* pretrained joint visual-text embedding model[5] in [31] to extract embeddings: $\boldsymbol{x}_{n,i}^v$ is the output of the I3D network and $\boldsymbol{x}_{n,i}^l$ is the output of two fully-connected layers after the word2vec model, pretrained on Howto100M dataset [32]. To learn more discriminative features where visual and narration features of the same key-step are close, we use a feature learning module to refine the unsupervised

[2]We interchangeably use *textual* and *linguistic* in this paper.

[3]https://spacy.io

[4] prt: particle; dobj: direct object; prep: preposition; pobj: object of preposition. The components in the parenthesis are optional.

[5]This model is pretrained in an unsupervised manner without access to any ground-truth annotations. We keep this model fixed, i.e., do not fine-tune it, in our experiments.

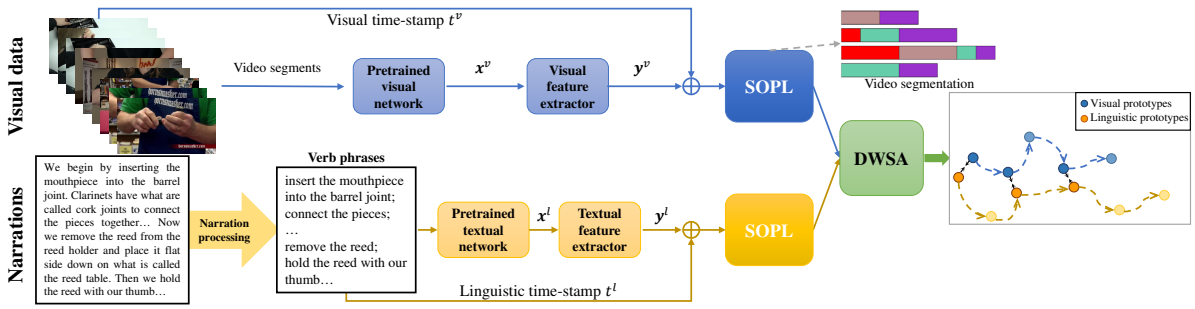


Figure 2: Overview of our framework for unsupervised learning from instructional videos and their narrations. The dashed arrows in the plot of prototypes denote their learned ordering and black arrows represent the alignment found by DWSA, which minimizes distances of aligned pairs for feature learning.

features of each modality. Let $f_{\Theta_v}(\cdot)$ and $f_{\Theta_l}(\cdot)$ denote, respectively, the feature learning functions for visual data and narrations, with parameters Θ_v and Θ_l. We denote the transformed visual and language features by

$$\boldsymbol{y}^v_{n,i} \triangleq f_{\Theta_v}(\boldsymbol{x}^v_{n,i}), \quad \boldsymbol{y}^l_{n,i} \triangleq f_{\Theta_l}(\boldsymbol{x}^l_{n,i}). \tag{2}$$

In the experiments, we discuss the exact architectures used for feature learning.

3.2.3 Soft Ordered Prototype Learning

To recover visual and linguistic prototypes, we use the following observations. First, given that videos come from the same task, the sequences of key-steps in different videos are similar, e.g., have a small edit distance from each other (see Figure 1). Moreover, the relative length of each key-step with respect to the video length is similar across videos (several works on weakly-supervised learning also rely on such length consistency [38, 27, 28]). Also, nearby frames or verb-phrases belong to the same key-step or background.

For simplicity of notation and removing repetitive descriptions, we drop the superscript/subscript v and l. We associate each feature prototype $\boldsymbol{c}$, representing a key-step, with a time-stamp prototype τ to enforce that nearby time-stamps in each video and similar time-stamps across videos (when lengths of all videos are normalized to be the same) should belong to the same key-step. To do so, we optimize the modified k-means objective function

$$\begin{aligned} &\min_{\{\boldsymbol{c}_k,\tau_k\}} \sum_n \sum_i -\beta \log \Big(\sum_k e^{-d_{nik}/\beta} \Big), \\ &d_{nik} \triangleq \|\boldsymbol{y}_{n,i} - \boldsymbol{c}_k\|^2 + \gamma \big(\frac{t_{n,i}}{T_n} - \tau_k\big)^2 \end{aligned} \tag{3}$$

where the term inside the second sum corresponds to the soft-min operation[6] with parameter $\beta \geq 0$, which allows us to learn features by backpropagating the gradient of our DWSA loss function. Also, T_n denotes the length of the video n and $\gamma \geq 0$ is a hyperparameter. Algorithm 1 shows the steps of the soft ordered prototype learning

[6]Soft-min is defined as $\min_\beta\{\alpha_1, \alpha_2, \ldots\} = -\beta \log \sum_k e^{-\alpha_k/\beta}$.

Algorithm 1: Soft Ordered Prototype Learning (SOPL)

Input : $\{(\boldsymbol{y}_{n,i}, t_{n,i})\}_{n,i}, K, \beta \geq 0$

Initialize prototypes, $\{\boldsymbol{c}_k, \tau_k\}_{k=1}^K$

for $iter \leftarrow 1$ **to** $p = 5$ **do**

  Compute $\{d_{nik}\}$ via (3) and soft assignments

  $s_{nik} = \frac{\exp(-d_{nik}/\beta)}{\sum_{j=1}^K \exp(-d_{nij}/\beta)}$

  Update prototypes

  $[\boldsymbol{c}_k, \tau_k] = \frac{\sum_n \sum_i s_{nik} [\boldsymbol{y}_{n,i},\ t_{n,i}/T_n]}{\sum_n \sum_i s_{nik}}$

Output: Feature and time prototypes $\{\boldsymbol{c}_k, \tau_k\}_{k=1}^K$

to solve (3) via gradient descent, by iteratively computing soft assignments and updating feature/time prototypes. To remove background segments, we use a background ratio parameter $b \in [0, 1]$, similar to [26]. We keep $1 - b$ fraction of the segments within each cluster that are closest to the prototype and consider other segments as background.

Remark 1 *Based on Algorithm 1, SOPL can be viewed as a network which receives initial prototypes as inputs and whose each layer outputs the updated prototypes.*[7] *This allows integration of SOPL with our proposed DWSA loss.*

The output of the algorithm provides ordering of feature prototypes $\{\boldsymbol{c}_k\}$ based on their learned time prototypes $\{\tau_k\}$. Let $\mathcal{O}^v$ and $\mathcal{O}^l$ denote the ordered visual and linguistic prototypes, respectively,

$$\begin{aligned} \mathcal{O}^v &\triangleq (\boldsymbol{c}^v_{i_1}, \boldsymbol{c}^v_{i_2}, \ldots), \text{ where } \tau^v_{i_1} \leq \tau^v_{i_2} \leq \ldots, \\ \mathcal{O}^l &\triangleq (\boldsymbol{c}^l_{j_1}, \boldsymbol{c}^l_{j_2}, \ldots), \text{ where } \tau^l_{j_1} \leq \tau^l_{j_2} \leq \ldots, \end{aligned} \tag{4}$$

which correspond to sequences of visual and linguistic key-steps learned from videos. Given that some key-steps may appear in only one modality, we develop a differentiable weak sequence alignment (DWSA) method to find the best ordered one-to-one matching between $\mathcal{O}^v$ and $\mathcal{O}^l$ while allowing some prototypes in a modality to stay unmatched.

[7]Note that s is computed from the pairwise distance between inputs and prototypes and it is not a trainable parameter.

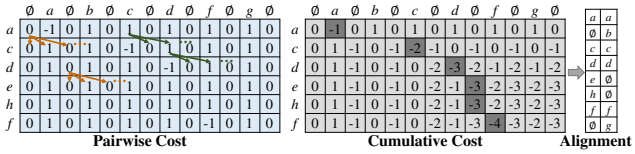


Figure 3: Illustration of our dynamic programming-based method for weak sequence alignment.

Remark 2 *The SOPL differs from [26] as we learn the feature and time prototypes simultaneously in a differentiable way for multi-modal feature learning, while [26] first applies kmeans on visual embeddings and then sorts clusters in temporal order based on their time-stamps.*

3.2.4 Diff. Weak Sequence Alignment (DWSA)

Each of the ordered visual and linguistic prototype sequences ($\mathcal{O}^v$ and $\mathcal{O}^l$) represents the common sequence of key-steps in each modality. However, these sequences are not necessarily aligned, as some steps in one modality might be missing in the other modality (see Figure 1). Thus, our goal is to find an ordered correspondence between $\mathcal{O}^v$ and $\mathcal{O}^l$ while allowing some prototypes in each sequence to be unmatched. More specifically, if item i in the visual sequence is aligned with item j in the linguistic sequence, then item $i+1$ can be associated with either items after j or with empty. To do so, we develop a differentiable weak sequence alignment (DWSA) loss between ordered visual and linguistic prototypes and propose to solve

$$\min_{\Theta_v,\Theta_l} \text{DWSA}(\mathcal{O}^v, \mathcal{O}^l, \ell(\cdot,\cdot)), \tag{5}$$

over the parameters of the visual and language feature learning modules. Here, $\ell(\cdot,\cdot)$ denotes the metric used for computing distances between items of the two sequences. In the paper, we use the cosine dissimilarity between visual and linguistic prototypes as distance, i.e.,

$$\ell(\boldsymbol{c}_i^v, \boldsymbol{c}_j^l) = 1 - \frac{\langle \boldsymbol{c}_i^v, \boldsymbol{c}_j^l \rangle}{\|\boldsymbol{c}_i^v\| \|\boldsymbol{c}_j^l\|}. \tag{6}$$

After solving (5) (see the next section for details), we use the learned prototypes from SOPL and assign each video segment to the closest learned visual prototype to obtain segmentation of videos into key-steps (similarly, we assign each verb-phrase to the closest linguistic prototype). Given that our method finds correspondences between visual and linguistic prototypes, we can also find all segments and verb-phrases that are clustered together.

4. Sequence Alignment via DWSA

We develop a differentiable weak sequence alignment method that i) finds a one-to-one ordered matching between two sequences while allowing some items in a sequence to be unmatched; ii) enables feature learning. We motivate the algorithm using a toy example of aligning symbolic sequences and then generalize it to arbitrary sequences.

Algorithm 2: Forward Propagation for DWSA

input : Cost matrix $\boldsymbol{\Delta}$; soft-min parameter $\beta \geq 0$

$d_{1,j} \leftarrow \delta_{1,j}, j \in \{1, 2, \ldots, 2q'+1\}$

for $i \leftarrow 2$ **to** q **do**

  for $j \leftarrow 1$ **to** $2q'+1$ **do**

    if j *is odd* **then**

      $d_{i,j} \leftarrow \delta_{i,j} + \min_\beta\{d_{i-1,1}, \ldots, d_{i-1,j}\}$

    else

      $d_{i,j} \leftarrow \delta_{i,j} + \min_\beta\{d_{i-1,1}, \ldots, d_{i-1,j-1}\}$

$\mathcal{L} \leftarrow \min_\beta\{d_{q,1}, \ldots, d_{q,2q'+1}\}$

output: DWSA Loss $= \mathcal{L}$

Assume we have two symbolic sequences $\mathcal{O} = (a, c, d, e, h, f)$ and $\mathcal{O}' = (a, b, c, d, f, g)$. Our goal is to find the best one-to-one alignment between two sequences that respects their orderings. This ideal ordering is shown in Figure 3 (right), where a's in two sequences are matched, b in sequence 2 is matched with empty $\varnothing$, c in two sequences are matched, and so on. To achieve this alignment, first, we take one of the sequences (here, $\mathcal{O}'$) and expand it by inserting empty before and after each symbol, $\mathcal{O}' = (\varnothing, a, \varnothing, b, \varnothing, c, \varnothing, d, \varnothing, f, \varnothing, g, \varnothing)$. We compute a pairwise matching cost matrix, shown in Figure 3 (left), as $\boldsymbol{\Delta}(\mathcal{O}, \mathcal{O}') = [\delta_{i,j}]$ whose (i,j)-th entry denoted by $\delta_{i,j}$ is the cost of aligning item i in $\mathcal{O}$ with item j in $\mathcal{O}'$. In our toy example, we set $\delta_{i,j} = -1$ for matching the same symbols, $\delta_{i,j} = 0$ for matching a symbol with empty and $\delta_{i,j} = +1$ for matching two distinct non-empty symbols. This means we prefer to align same symbols (cost of -1) over aligning a symbol with empty (cost of 0) over aligning two distinct symbols (cost of +1).

Our goal is to find a *valid* alignment path with minimum cost according to $\boldsymbol{\Delta}$ that satisfies the ordered one-to-one alignment: it can start from any entry in the first row of $\boldsymbol{\Delta}$ and keeps going downward ($\downarrow$) or right-downward ($\searrow$). More specifically, if the current alignment position is at (i,j) and j is odd (corresponding to empty), the next alignment position can be $(i+1, j')$ for any $j' \geq j$ (orange arrows in Fig. 3). This ensures that if a symbol of $\mathcal{O}$ is matched with empty in $\mathcal{O}'$, the next symbol of $\mathcal{O}$ can be paired with empty or the next symbol of $\mathcal{O}'$. On the other hand, if the current alignment position is at (i,j) and j is even (corresponding to symbols), the next alignment position can be $(i+1, j')$ for any $j' > j$ (green arrows in Fig. 3). This ensures that when two symbols in $\mathcal{O}$ and $\mathcal{O}'$ are matched, the next symbol in $\mathcal{O}$ can be paired with either empty or the next symbol in $\mathcal{O}'$, hence, preserving one-to-one correspondence.

Therefore, to find the minimum cost valid alignment, we use dynamic programming: we calculate a cumulative cost

matrix $\boldsymbol{D} = [d_{i,j}]$, whose first row is initialized by $d_{1,j} = \delta_{1,j}$ and its (i,j)-th entry is computed as

$$d_{i,j} = \begin{cases} \delta_{i,j} + \min_\beta \{d_{i-1,1}, \dots, d_{i-1,j}\}, j \text{ is odd} \\ \delta_{i,j} + \min_\beta \{d_{i-1,1}, \dots, d_{i-1,j-1}\}, j \text{ is even.} \end{cases} \quad (7)$$

After computing $\boldsymbol{D}$, the minimum value in the last row of $\boldsymbol{D}$ corresponds to the minimum alignment cost. We obtain the optimal alignment path by starting from this minimum entry location and by backtracking (see the supplementary materials for more details). Algorithm 2 shows the steps of the dynamic programming solution.

For the general case where we have two sequences $\mathcal{O} = (\boldsymbol{o}_1, \dots, \boldsymbol{o}_q)$ and $\mathcal{O}' = (\boldsymbol{o}'_1, \dots, \boldsymbol{o}'_{q'})$ of vectors, we construct the matrix of pairwise matching cost $\boldsymbol{\Delta}(\mathcal{O}, \mathcal{O}')$ as

$$\boldsymbol{\Delta}(\mathcal{O}, \mathcal{O}') \triangleq \left[e^{\delta_{i,j}} / \sum_{j'} e^{\delta_{i,j'}} \right], \quad \delta_{i,j} \triangleq \begin{cases} \ell(\boldsymbol{o}_i, \boldsymbol{o}'_j) & , j \text{: even}, \\ \delta_e & , j \text{: odd}, \end{cases} \quad (8)$$

where $\ell(\cdot,\cdot)$ is the cosine dissimilarity (other distance metrics could also be used) defined in (6) and δ_e is a predefined constant measuring the cost of matching with empty. In the experiments, we show that given the ability to learn features, our results are robust to δ_e.

Remark 3 *Notice we use a softmax function on each row of the distance matrix in* (8) *to discriminate between good and bad matchings and, more importantly, to avoid the trivial solution of collapsing all features to the same vector.*

Computing DWSA Gradient. When sequences are functions of learnable parameters, not only we can find the best alignment, but also learn features that lend themselves to better alignment. Indeed, this is the case in our setting, where the visual and linguistic sequences ($\mathcal{O}^v$ and $\mathcal{O}^l$) depend on parameters of the feature learning modules (Θ_v and Θ_l). Thus, to update Θ_v, we need to compute

$$\nabla_{\Theta_v} \text{DWSA}(\mathcal{O}, \mathcal{O}') = \left(\frac{\partial \mathcal{O}}{\partial \Theta_v} \right)^T \nabla_{\mathcal{O}} \text{DWSA}(\mathcal{O}, \mathcal{O}'), \quad (9)$$

which requires differentiating the loss w.r.t. $\mathcal{O}$ (similarly for Θ_l). As we show in the supplementary materials, differentiation can be efficiently computed by defining intermediate variables $g_{i,j}$ and recursively updating them starting from the last row and column of $\boldsymbol{\Delta}$. Algorithm 3 shows the recursion for computing the gradient w.r.t. $\mathcal{O}$ (once computed, the gradient w.r.t. Θ_v is given by (9)).

Computational Complexity. The complexity of each forward and backward pass in our method is $O(qq')$. This can be seen from Algorithm 2, where we scan over q rows and $2q'+1$ columns of the cost matrix. Notice that computing $d_{i,j}$ has $O(1)$ cost as we can reuse the minimum from the previous iteration, i.e., $\min_\beta\{d_{i-1,1}, \dots, d_{i-1,j}\} = \min_\beta\{\min_\beta\{d_{i-1,1}, \dots, d_{i-1,j-1}\}, d_{i-1,j}\}$. Similarly, in Algorithm 3, the precomputed sum values can be reused in Lines 5 and 7.

Algorithm 3: Backward Propagation for DWSA

input : Matching cost $\boldsymbol{\Delta} \in \mathbb{R}^{q \times 2q'+1}$; Cumulative cost $\boldsymbol{D}$; soft-min parameter $\beta \geq 0$

1 $g_{q,j} \leftarrow \frac{e^{-d_{q,j}/\beta}}{\sum_{r=1}^{2q'+1} e^{-d_{q,r}/\beta}}, j \in \{1, \dots, 2q'+1\}$
2 **for** $i \leftarrow q-1$ **to** 1 **do**
3 **for** $j \leftarrow 2q'+1$ **to** 1 **do**
4 **if** j *is odd* **then**
5 $g_{i,j} \leftarrow \sum_{r \geq j} g_{i+1,r} e^{(-d_{i,j} + d_{i+1,r} - \delta_{i+1,r})/\beta}$
6 **else**
7 $g_{i,j} \leftarrow \sum_{r > j} g_{i+1,r} e^{(-d_{i,j} + d_{i+1,r} - \delta_{i+1,r})/\beta}$
8 Set $\boldsymbol{G} = [g_{i,j}]$

output: $\nabla_{\mathcal{O}} \text{DWSA}(\mathcal{O}, \mathcal{O}') = \left(\frac{\partial \boldsymbol{\Delta}(\mathcal{O}, \mathcal{O}')}{\partial \mathcal{O}} \right)^T \boldsymbol{G}$

5. Experiments

5.1. Experimental setup

Datasets. We evaluate our proposed method on two instructional video datasets: ProceL [17] and CrossTask [50]. ProceL consists of 47.3 hours of videos from 12 tasks, where each task has about 60 videos. CrossTask has 213 hours of footage from 2,750 videos from 18 primary tasks.[8] In both datasets, each video has narrations and annotations of key-steps. We cannot use COIN [45] as it lacks narrations or Howto100M [32] as it lacks key-step annotations.

Evaluation metrics. We use the framewise F1 score as the primary metric and also report the framewise recall and precision. Similar to prior works [1, 26, 17], we run the Hungarian algorithm to find a global one-to-one matching between steps in the ground-truth and predictions. Recall is the ratio between the number of correctly predicted frames and number of frames with key-steps in ground-truth. Precision is the ratio between the number of correctly predicted frames and number of frames predicted as key-steps. F1 score is the harmonic mean of precision and recall.

To better demonstrate the undesired effect of including background on the evaluation metric, we also compute the mean-over-frames (MoF) [26], which is the percentage of frames for which the predictions, including background, are correct. Additionally, similar to prior works [1, 50, 18], we compute the step recall. Here, one assigns a single frame to each predicted step in a video and measures the ratio between the number of correct predictions (a predicted frame is correct when it falls into the correct ground-truth time interval) and the number of ground-truth key-steps.

Baselines. We compare our method with the following unsupervised baselines: *Uniform*, which distributes key-step assignments uniformly over all segments in each video; *Alayrac et al.* [1], which uses narrations and visual data;

[8] CrossTask also consists of 65 secondary tasks, whose key-steps are not annotated, hence, we cannot use them in our experiments.

	ProceL				CrossTask			
	F1 (%)	Recall (%)	Precision (%)	MoF (%)	F1 (%)	Recall (%)	Precision (%)	MoF (%)
Uniform	10.28	9.36	12.41	48.20	9.03	9.75	8.69	55.88
Alayrac et al. [1]	5.54	3.73	12.25	**55.77**	4.46	3.43	6.80	**64.18**
Kukleva et al. [26]	16.39	30.19	11.69	12.04	15.27	35.90	9.82	13.95
Elhamifar et al. [16]	14.00	26.70	9.49	5.61	16.30	**41.60**	10.14	13.72
Fried et al. [18]	–	–	–	–	–	28.80	–	31.80
Ours	**21.07**±0.25	**31.78**±0.37	**16.51**±0.09	25.79±0.22	**21.00**±0.09	35.46±0.14	**15.21**±0.07	40.99±0.07

Table 2: Performance comparison on ProceL and CrossTask. We report framewise F1, recall, precision and MoF averaged over tasks.

		Make Kimchi Rice	Pickle Cucumber	Make Banana Ice Cream	Grill Steak	Jack Up Car	Make Jello Shots	Change Tire	Make Lemonade	Add Oil to Car	Make Latte	Build Shelves	Make Taco Salad	Make French Toast	Make Irish Coffee	Make Strawberry Cake	Make Pancakes	Make Meringue	Make Fish Curry	Average
Weakly sup.	Specific [50]	13.2	17.6	19.3	19.3	9.7	12.6	30.4	16.0	4.5	19.0	29.0	9.1	29.1	*14.5*	22.9	29.0	32.9	7.3	18.6
	Sharing [50]	*13.3*	*18.0*	*23.4*	*23.1*	*16.9*	*16.5*	*30.7*	*21.6*	*4.6*	*19.5*	*35.3*	*10.0*	*32.3*	13.8	*29.5*	*37.6*	*43.0*	*13.3*	*22.4*
Unsup.	Uniform	4.2	7.1	6.4	7.3	17.4	7.1	14.2	9.8	3.1	10.7	22.1	5.5	9.5	7.5	9.2	9.2	19.5	5.1	9.7
	Alayrac et al. [1]	15.6	10.6	7.5	14.2	9.3	11.8	17.3	13.1	6.4	12.9	27.2	9.2	15.7	8.6	16.3	13.0	23.2	7.4	13.3
	Kukleva et al. [26]	12.3	13.1	13.9	17.2	14.8	11.9	13.2	10.1	7.6	**16.2**	24.1	6.3	17.0	11.1	16.1	15.5	18.1	11.5	13.9
	Elhamifar et al. [16]	7.5	5.1	9.5	5.8	5.1	**22.4**	**24.9**	5.3	8.2	9.6	6.8	**11.3**	9.8	**15.8**	4.6	6.6	11.5	8.7	9.9
	Fried et al. [18]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	10.6
	Ours	**18.5**	**16.3**	**25.9**	**22.4**	**17.8**	17.3	16.8	**15.8**	**14.3**	11.8	**32.5**	10.6	**24.3**	10.3	**28.6**	**27.4**	**33.2**	**11.7**	**19.8**

Table 3: Step recall (%) on CrossTask. We compare our unsupervised approach with unsupervised baselines and the weakly-supervised methods in [50].

Kukleva et al. [26], *Elhamifar et al.* [16] and *Fried et al.* [18], which use visual data (the unsupervised version of [18] uses only visual data and its weakly-supervised version uses both modalities). In the presented results in the main paper, similar to [26, 18], we set the number of visual clusters K^v to be the number of key-steps in the ground-truth. Given that [1, 16] measure performance as a function of the number of key-step $K \in \{7, 10, 12, 15\}$ in predictions, we report their best performance across K's. In the supplementary materials, we also report results as a function of K, which give similar conclusions.

5.2. Implementation details

Data preprocessing. As discussed in Section 3.2.2, we use the unsupervised pretrained visual-text embedding model in [31] to extract visual and textual features. To do so, similar to [31], we segment each video into intervals of 32 frames sampled at 10 fps (3.2 seconds) with 224×224 resolution. For each verb-phrase in narrations, we lowercase, perform tokenization and remove stop words as preprocessing. The feature dimension for both modalities is 512.

Feature learning. For visual and textual feature extractors, discussed in Section 3.2.2, we use a linear layer followed by normalization into unit length vectors, $\boldsymbol{y}^{\star}_{n,i} = \text{norm}(\boldsymbol{W}^{\star}\boldsymbol{x}^{\star}_{n,i} + \boldsymbol{b}^{\star})$, with $\star \in \{v, l\}$, where $\Theta_v = \{\boldsymbol{W}^v, \boldsymbol{b}^v\}$ and $\Theta_l = \{\boldsymbol{W}^l, \boldsymbol{b}^l\}$ are trainable parameters. In our experiments, more complex models did not improve results.

Hyperparameters. For feature learning, we train the feature extractors using Adam optimizer [24] with learning rate of 5e−4, weight decay of 0.02, and batch size of 30 videos for 30 epochs. On both datasets, the softmin parameter β in SOPL and DWSA is set to 0.001 and the cost of alignment with empty is set to $\delta_e = 1$. We set ($\gamma = 4, b = 0.2$) on ProceL and ($\gamma = 1, b = 0.4$) on CrossTask. We set the number of textual prototypes to be the number of visual prototypes plus 10. We repeat every experiment 20 times for different initializations of the visual and textual prototypes and report the mean and standard deviation. We show the robustness of our method to the hyperparameters in the next section and in the supplementary materials.

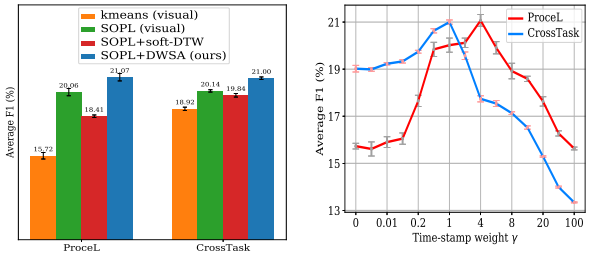


Figure 4: Ablation studies (left) and effect of timestamp weights γ (right). The bars show standard deviations.

5.3. Experimental results

Table 2 shows the average scores of different methods on ProceL and CrossTask datasets. Notice that [1] has lower framewise precision, recall and F1 than other methods but much higher MoF. This comes from the fact that it only predicts a single frame instead of intervals, which reduces the first three metrics (later, we also show the step recall, which only considers a single frame prediction). Having a high MoF is due to the fact that a large portion of each video is background, hence, predicting almost all frames as background achieves a high score (this can also be seen from the results of Uniform). While [26] does better than [16] on ProceL, the trend is opposite on CrossTask. This comes from the fact that [16] trains a deep network in a self-supervised fashion for key-step

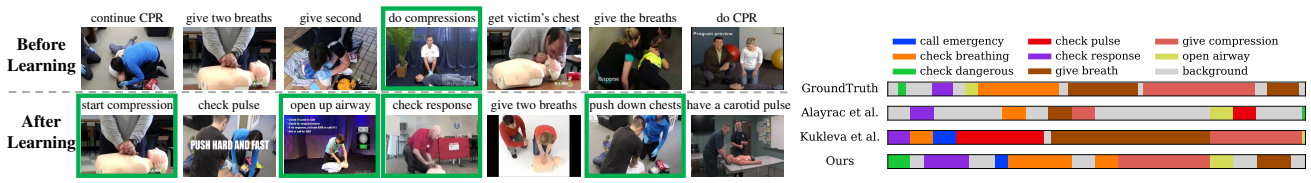


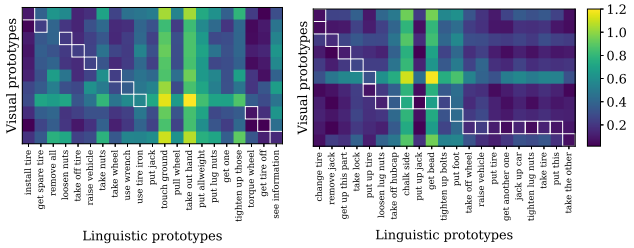
Figure 5: Left: Matched visual and textual prototypes before and after feature learning. For visualization, we use the video frame and phrase closest to the prototypes. Correct matchings are marked with green boxes. Right: Localization results for a video from the task 'perform cpr'.

localization, which benefits from the larger number of videos in CrossTask, obtaining the best recall on the dataset. Notice that our method performs significantly better than other algorithms, improving the F1 score by about 4.7% on both datasets. This verifies the importance of properly leveraging narrations for more effective key-step discovery.

Step detection performance. Table 3 shows the step recall of different tasks on CrossTask. We additionally compare our method with two weakly-supervised approaches in [50], *Specific* and *Sharing*[9], which use visual and narration data. Notice that our method outperforms unsupervised methods, obtaining 19.8% recall compared to 13.9% by [26]. Also, the performance of the weakly-supervised method, which could be considered as an upper bound on unsupervised methods, is close to ours. Interestingly, on 13 out of 18 tasks, our method achieves a higher step recall than other unsupervised methods and on 6 out of 18 tasks, performs better than weakly-supervised methods.

Effect of different components. Figure 4 (left) shows the effect of different components of our method. Kmeans and SOPL only use visual data, where the former does not enforce ordering consistency of prototypes ($\gamma = 0$), while the latter does. Notice that enforcing order consistency has a significant effect, improving performance by 4.3% and 1.2% on ProceL and CrossTask. SOPL+soft-DTW, which uses both visual and text data and performs feature learning performs worse than SOPL. This comes from the fact that poor matching between prototypes in different modalities obtained by soft-DTW adversely affects feature learning and performance. On the other hand, using DWSA boosts the performance of SOPL by about 1% thanks to obtaining better prototype matchings. Figure 4 (right) shows the effect of γ in (3). Notice that on both datasets, there is a range for which our method obtain stable results (for $\gamma \in [0.2, 10]$ on both datasets, our method outperforms other baselines). As expected, for very large γ, the performance drops as the method ignores the common information in visual and linguistic features (see the supplementary materials for analysis based on order consistency of tasks).

Qualitative analysis. Figure 5 (left) shows the visualization of matching between visual and textual prototypes before and after feature learning for the task 'perform CPR'. We show the closest frame and verb-phrase to each prototype. While matchings are poor before learning (e.g., 'give two breaths', 'get victim's chest' and 'give breaths' are associated with incorrect demonstrations), the quality of matching significantly improves after learning. Figure 5 (right) shows localization results of different methods. Notice that our method can localize 'check dangerous' thanks to using narrations, while successfully localizing other steps such as 'check response', 'check breathing', 'give compression' and 'give breath' (but missing its first occurrence). Finally, Figure 6 shows the learned distances via our DWSA and soft-DTW for the task 'change tire'. Soft-DTW aligns a single visual prototype with multiple linguistic prototypes (e.g., 'take off wheel', 'raise vehicle', 'put tire'), hence, learning similar embeddings for verb-phrases of distinct key-steps. On the other hand, DWSA learns one-to-one matching while allowing noisy/incorrect linguistic prototypes to stay unmatched. More qualitative examples are included in the supplementary materials.



(a) DWSA (b) Soft-DTW

Figure 6: Learned distances between visual and textual prototypes using DWSA and Soft-DTW. White boxes indicate the alignment path.

6. Conclusions

We proposed an unsupervised action segmentation method for instructional videos and narrations. We modeled visual and language key-steps by prototypes, recovered them by developing a soft ordered prototype learning module and developed a novel weak sequence alignment method to find correspondence between visual and linguistic prototype sequences. By experiments on two datasets, we showed the effectiveness of our method.

Acknowledgements

This work is partially supported by DARPA Young Faculty Award (D18AP00050), NSF (IIS-1657197), ONR (N000141812132) and ARO (W911NF1810300).

[9]They use transcripts as supervision. *Specific* learns a classifier for each step of a task, while *Sharing* incorporates sharing key-steps across tasks.

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.

[2] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 1997.

[3] 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.

[4] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 2014.

[5] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[6] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. *European Conference on Computer Vision*, 2018.

[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.

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

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

[10] Pelin Dogan, Boyang Li, Leonid Sigal, and Markus Gross. A neural multi-sequence alignment technique (neumatch). *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[11] H. Doughty, I. Laptev, W. Mayol-Cuevas, and D. Damen. Action modifiers: Learning from adverbs in instructional videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[12] Xinya Du, Bhavana Dalvi Mishra, Niket Tandon, Antoine Bosselut, Wen tau Yih, Peter Clark, and Claire Cardie. Be consistent! improving procedural text comprehension using label consistency. *Annual Meeting of the North American Association for Computational Linguistics*, 2019.

[13] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. Temporal cycle-consistency learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[14] E. Elhamifar. Sequential facility location: Approximate submodularity and greedy algorithm. *International Conference on Machine Learning*, 2019.

[15] E. Elhamifar and M. C. De-Paolis-Kaluza. Subset selection and summarization in sequential data. *Neural Information Processing Systems*, 2017.

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

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

[18] Daniel Fried, Jean-Baptiste Alayrac, Phil Blunsom, Chris Dyer, Stephen Clark, and Aida Nematzadeh. Learning to segment actions from observation and narration. *Annual Meeting of the Association for Computational Linguistics*, 2020.

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

[20] Di Hu, Feiping Nie, and Xuelong Li. Deep multimodal clustering for unsupervised audiovisual learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[21] D. Huang, S. Buch, L. Dery, A. Garg, L. Fei-Fei, and J. C. Niebles. Finding it?: Weakly-supervised reference-aware visual grounding in instructional videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[22] D. A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist temporal modeling for weakly supervised action labeling. *European Conference on Computer Vision*, 2016.

[23] Oana Ignat, Laura Burdick, Jia Deng, and Rada Mihalcea. Identifying visible actions in lifestyle vlogs. *Annual Meeting of the Association for Computational Linguistics*, 2019.

[24] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[25] 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.

[26] 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.

[27] J. Li, P. Lei, and S. Todorovic. Weakly supervised energy-based learning for action segmentation. *International Conference on Computer Vision*, 2019.

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

[29] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in Neural Information Processing Systems*, 2019.

[30] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *NAACL*, 2015.

[31] 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.

[32] 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.

[33] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. *European Conference on Computer Vision*, 2016.

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

[35] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[36] X. Puig, K. Ra, M. Boben, J. Li, , T. Wang, S. Fidler, and A. Torralba. Virtualhome: Simulating household activities via programs. *IEEE Conference on computer Vision and Pattern Recognition*, 2018.

[37] 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.

[38] 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.

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

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

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

[42] Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections. *IEEE International Conference on Computer Vision*, 2015.

[43] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. *International Conference on Robotics and Automation*, 2018.

[44] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *IEEE International Conference on Computer Vision*, 2019.

[45] 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.

[46] Ottokar Tilk and Tanel Alumäe. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. *Interspeech*, 2016.

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

[48] Shoou-I Yu, Lu Jiang, and Alexander Hauptmann. Instructional videos for unsupervised harvesting and learning of action examples. *ACM International Conference on Multimedia*, 2014.

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

[50] 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.

[51] D. Zhukov, J. B. Alayrac, I. Laptev, and J. Sivic. Learning actionness via long-range temporal order verification. *European Conference on Computer Vision*, 2020.