Weakly-Supervised Action Segmentation and Alignment via Transcript-Aware Union-of-Subspaces Learning

Zijia Lu
Northeastern Univeristy
lu.zij@northeastern.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

Abstract

We address the problem of learning to segment actions from weakly-annotated videos, i.e., videos accompanied by transcripts (ordered list of actions). We propose a framework in which we model actions with a union of low-dimensional subspaces, learn the subspaces using transcripts and refine video features that lend themselves to action subspaces. To do so, we design an architecture consisting of a Union-of-Subspaces Network, which is an ensemble of autoencoders, each modeling a low-dimensional action subspace and can capture variations of an action within and across videos. For learning, at each iteration, we generate positive and negative soft alignment matrices using the segmentations from the previous iteration, which we use for discriminative training of our model. To regularize the learning, we introduce a constraint loss that prevents imbalanced segmentations and enforces relatively similar duration of each action across videos. To have a real-time inference, we develop a hierarchical segmentation framework that uses subset selection to find representative transcripts and hierarchically align a test video with increasingly refined representative transcripts. Our experiments on three datasets show that our method improves the state-of-the-art action segmentation and alignment, while speeding up the inference time by a factor of 4 to 13.[1]

1. Introduction

Localization and classification of human actions in long uncurated videos has been a major challenge in video understanding [54, 10, 11, 27, 65, 69, 66, 59, 19]. While many methods have studied the problem in a fully-supervised setting using dense supervision [50, 57, 29, 33, 55, 67], gathering framewise annotations is costly and cannot scale to massive amounts of video data, which are available today. As a result, there has been an increasing interest in methods that can learn from weakly-annotated videos. In particular, action transcripts, which refer to sequences of actions appearing in videos without specifying their beginning and ending times, are less costly to gather and can also be obtained from video narrations or other meta data [32, 1, 42]. This has motivated a variety of interesting approaches that learn to localize and classify actions using transcripts [21, 2, 48, 49, 9, 9, 73, 36, 4, 37].

Challenges. Despite tremendous advances, existing works on weakly-supervised action learning still face major challenges. In fact, a successful class of recent methods focuses on alternating between segmentation of the training videos using transcripts and retraining models with the obtained segmentations [49, 36]. However, training a model with the one estimated segmentation could ignore and discourage other likely segmentations and propagate the initial segmentation errors.

Moreover, existing methods often ignore the underlying low-dimensional structures of videos. In fact, it is well known that high-dimensional visual data, e.g., rigid and nonrigid motions or human actions, lie in low-dimensional subspaces [63, 12, 41, 3, 40, 38, 6]. Yet, leveraging such low-dimensional subspaces in the weakly-supervised setting has been mainly ignored, as the existing works work in the fully-supervised or fully-unsupervised regimes and cannot take advantage of weak supervision, e.g., transcripts.

On the other hand, inference on test videos that do not have transcripts is often extremely costly. This comes from the fact that existing methods require aligning the test video with every transcript in the training set to select the most likely transcript and the associated segmentation. This prevents methods from being applicable in real-time.

Paper Contributions. In this paper, we address the problem of weakly-supervised action segmentation by developing a Transcript-aware Action Subspace Learning (TASL) framework that models actions with a union of low-dimensional subspaces, learns the subspaces using weak supervision (transcripts) and refines video features that lend themselves to action subspaces. To do so, we design an architecture consisting of a feature learning module and a new

[1] Code is available at https://github.com/ZijiaLewisLu/ICCV21-TASL.

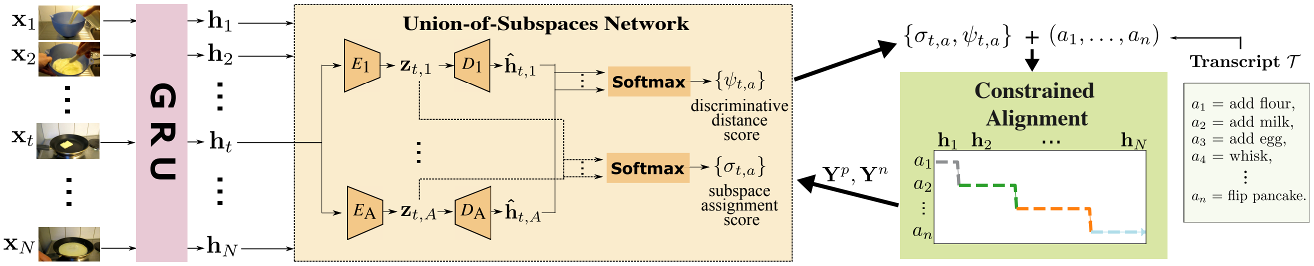

Figure 1: We propose a framework, referred to as Transcript-aware Action Subspace Learning (TASL), for weakly-supervised segmentation of videos. The framework consists of a Union-of-Subspaces Network (USN), which learns to embed actions into discriminative low-dimensional subspaces, and an efficient constrained video alignment algorithm that generates positive and negative soft alignments, which will be used for parameter learning.

Union-of-Subspaces Network (USN). USN is an ensemble of autoencoders, each modeling a low-dimensional action subspace, that captures variations of each action. As we show in the experiments, depending on the semantic similarity of actions (e.g., sharing verb or noun), the learned subspaces will be nearly orthogonal to each other in some directions (allowing discrimination), while intersecting in some other directions (capturing shared information).

For learning, we alternate between segmenting training videos using transcripts and learning models and features from segmentations. However, instead of learning a model to reproduce an obtained segmentation, we generate positive and negative soft alignment matrices using the optimal segmentation, which we will use for discriminative learning of subspaces. We introduce a constraint loss to prevent imbalanced segmentations and to enforce relatively similar duration of each action across videos.

To have real-time inference, we develop a hierarchical segmentation framework that uses subset selection to find representative transcripts of training videos. We will hierarchically align a test video with increasingly refined representative transcripts. Our experiments on three datasets show that our method improves the state of the art while speeding up inference by a factor of 4 to 13.

2. Related Works

Action Segmentation with Minimal Supervision. Large amounts of long untrimmed videos [56, 15, 73, 44, 53, 7] along with the high cost of framewise video annotations have motivated a large body of works in computer vision to localize and classify actions with minimum supervision.

Weakly-supervised methods learn from ordered or unordered list of actions in videos [21, 2, 48, 49, 9, 73, 36, 4, 37] or video summaries [64]. In particular, [30] interprets the problem as speech recognition problem, where the videos correspond to the audio signal and the action classes correspond to words, hence, learns a standard HMM-GMM model using a speech recognition toolkit. Building upon this idea, [47] replaces the GMM by a recurrent neural network, while still relying on an HMM for a coarse temporal modeling. Also, [24, 25, 47, 30] use a two-step optimization scheme that does not allow for direct, sequence-wise training. [39] uses the connectionist temporal classification (CTC) approach in combination with a statistical language model. As an extension of the CTC approach, [21] proposes ECTC that accounts for visual similarities between the frames to avoid degenerate segmentations. [9] trains a network on uniformly generated segmentations and iteratively inserting new actions into the segmentations based on the learned network, which are then used to retrain the network. Finally, [49] generates optimal segmentations using Viterbi decoding that will be used to train a classifier. [4] maximizes the likelihoods of all transcript-consistent segmentations and minimizes those of transcript-inconsistent ones. [36] has achieved state-of-the-art performance by optimizing valid segmentations that are generated by slightly shifting action boundaries of the optimal segmentations.

Several works have also studied the weakly-supervised learning from unordered list of actions appearing in videos. In particular, [48, 37] extend the Viterbi decoding to the set-supervised action segmentation problem, which alternates between estimating ordering of actions and learning a segmentation model. In contrast, [14] proposes to directly predict the actions and their lengths via a neural network. Finally, to completely remove the need for video annotations, several recent works have studied unsupervised action segmentations by leveraging the shared structure of videos from similar tasks [54, 10, 11, 31, 52, 1, 51, 17].

Subspace Learning. The goal of subspace clustering is to cluster data into underlying low-dimensional subspaces and learn parameters of subspaces. This has been addressed using iterative methods [61, 20, 71, 60, 18, 16], which alternate between estimating subspaces and clustering data, or spectral clustering-based methods that build affinities between data points often using sparse or low-rank representations [13, 40, 45, 35, 68, 70, 43, 5, 8]. Motivated by advances in deep learning, recent methods have studied unsupervised feature learning for subspace clustering [22, 72, 46]. *Given that subspace clustering is an unsupervised problem, existing methods cannot take advantage of the weak supervision, when available.* We propose a method that learns a union of subspaces using transcripts.

3. Transcript-Aware Multi-Subspace Learning

Problem Statement. Assume we have V videos and their action transcripts $\{(\mathcal{X}^v, \mathcal{T}^v)\}_{v=1}^{V}$, where $\mathcal{X}^v = (\boldsymbol{x}_1^v, \ldots, \boldsymbol{x}_{N_v}^v)$ denotes the collection of framewise unsupervised features for the video v, which has N_v frames. $\mathcal{T}^v = (a_1^v, \ldots, a_{n_v}^v)$ denotes its transcript, which is the ordered list of n_v actions in the video. We have $a_i^v \in \{1, 2, \ldots, A\}$, where A denotes the total number of actions across videos. The goal of weakly-supervised action learning is to learn an action segmentation model only using the transcripts of the training videos and to predict the actions of test videos. Depending on the information provided for a test video, inference can be divided into action alignment, where the video's transcript is known, and action segmentation, where the transcript is unknown. Indeed, action segmentation can be cast as action alignment by aligning the test video with the transcripts of training videos and selecting the one that has the minimum alignment cost. For simplicity, we drop the superscript and subscript v in notations (referring to video v), as it would be clear from the context.

Proposed Framework. To address the problem of weakly-supervised action segmentation, we develop the Transcript-aware Action Subspace Learning (TASL) framework. As shown in figure 1, TASL alternates between aligning training videos with transcripts using the model predictions and learning the model using current alignments. The proposed model consists of a GRU as a feature learning module and a Union-of-Subspaces Network (USN) to learn low-dimensional subspaces of actions via an ensemble of autoencoders. The outputs of USN are two scores capturing the similarity between each frame feature and each action subspace. Using the scores, we find the optimal alignment of a video by assigning frames to their closest subspaces while respecting the transcript. We then use the alignments to generate candidate valid and invalid frame labels, encoded via two soft alignment matrices $\boldsymbol{Y}^p, \boldsymbol{Y}^n$. The two matrices are then used in the discriminative network loss to enforce frame features have large/small embeddings on to the likely/unlikely subspaces while increasing distances between learned subspaces.

3.1. Discriminative USN Training

In this section, we introduce the designed network architecture and efficient discriminative loss for learning features and low-dimensional subspaces corresponding to actions.

Proposed Architecture. First, we use a recurrent network (here GRUs) as the feature learning module, $(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N) = \mathrm{GRU}\big((\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)\big)$, that captures temporal dependencies between framewise unsupervised features and transforms them into more discriminative features lying in low-dimensional subspaces corresponding to actions. To achieve such low-dimensional embeddings, we design a Union-of-Subspaces Network (USN) that consists of an ensemble of A autoencoders for A actions. The autoencoder a encodes the input feature vector $\boldsymbol{h}_t \in \mathbb{R}^p$ into a low-dimensional embedding vector $\boldsymbol{z}_{t,a} \in \mathbb{R}^d$ $(d \ll p)$, which will be decoded to $\hat{\boldsymbol{h}}_{t,a} \in \mathbb{R}^p$. More specifically,

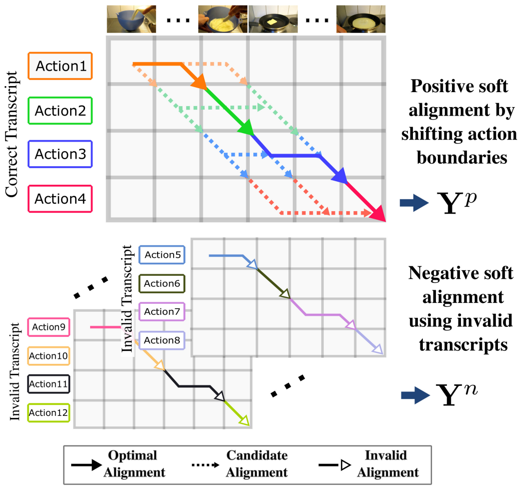

Figure 2: Proposed alignment algorithm: 1) perform constrained Viterbi decoding to obtain an optimal alignment, 2) generate positive and negative soft alignments based on the optimal alignment.

$$\boldsymbol{z}_{t,a} = \boldsymbol{W}_a^e \boldsymbol{h}_t + \mathrm{b}_a^e \in \mathbb{R}^d, \;\; \hat{\boldsymbol{h}}_{t,a} = \boldsymbol{W}_a^d \boldsymbol{z}_{t,a} + \mathrm{b}_a^d \in \mathbb{R}^p, \tag{1}$$

where $\{\boldsymbol{W}_a^e, \boldsymbol{W}_a^d, \mathrm{b}_a^e, \mathrm{b}_a^d\}$ are the learnable weights of the encoder and decoder of the action a, respectively. Here, $\{\boldsymbol{z}_{t,a}\}$ represent the d-dimension embeddings of $\{\boldsymbol{h}_t\}$ on the subspace of action a. With the linear decoder, $\{\hat{\boldsymbol{h}}_{t,a}\}$ are affine transformations of $\{\boldsymbol{z}_{t,a}\}$ using the same combination weights $\boldsymbol{W}_a^d$, thus, lie on the d-dimension subspace. Therefore, subspaces are described by the column spaces of $\{\boldsymbol{W}_a^d\}$. A feature $\boldsymbol{h}_t$ being close to $\hat{\boldsymbol{h}}_{t,a}$ implies that the frame t is close to the subspace a.

Given framewise labels from the alignments, one can naively learn a subspace for each action a by minimizing the distance $\|\hat{\boldsymbol{h}}_{t,a} - \boldsymbol{h}_t\|_2$ over all frames assigned to it. However, this has several drawbacks. First, minimizing the distance to one subspace will not necessarily increase the distances to other subspaces, which results in poor action segmentation performance. Moreover, as we learn both features and subspace parameters, minimizing the distances alone results in shrinking weights and features towards zero, hence, loosing distinctions between actions. Finally, the cost does not use the information in the embeddings, $\boldsymbol{z}_{t,a}$.

Proposed Discriminative Training. To address the above challenges, we develop a method that uses two complementary scores for discriminative training. Since $\|\boldsymbol{z}_{t,a}\|_2$ corresponds to the embedding norm of $\boldsymbol{h}_t$ onto the subspace a, we compute the *subspace assignment score* of $\boldsymbol{h}_t$ to subspace a by

$$\sigma_{t,a} = \frac{e^{\|\boldsymbol{Q}_a \boldsymbol{z}_{t,a}\|_2^2}}{\sum_{a'} e^{\|\boldsymbol{Q}_{a'} \boldsymbol{z}_{t,a'}\|_2^2}} \in [0,1]. \tag{2}$$

Since not every direction in a subspace is necessarily useful for recognition of the underlying action, e.g., directions corresponding to intersection with other subspaces, we use $\boldsymbol{Q}_a \in \mathbb{R}^{d' \times d}$ ($d' \leq d$) to allow learning discriminative features within each subspace a.

Given $\|\hat{\boldsymbol{h}}_{t,a} - \boldsymbol{h}_t\|_2$ as the distance between $\boldsymbol{h}_t$ and the subspace a, we define the *discriminative distance score*,

$$\psi_{t,a} = \frac{e^{-\|\hat{\boldsymbol{h}}_{t,a} - \boldsymbol{h}_t\|_2^2}}{\sum_{a'} e^{-\|\hat{\boldsymbol{h}}_{t,a'} - \boldsymbol{h}_t\|_2^2}} \in [0,1], \tag{3}$$

whose maximization for a subspace a enforces that $\boldsymbol{h}_t$ must be close to it and far from other subspaces.

Based on network outputs, our alignment algorithm will produce two soft label matrices (see the next subsection for details): 1) positive soft alignments $\boldsymbol{Y}^p \in [0,1]^{N \times A}$, whose each row is the probability distribution of frame t belonging to each action, and is computed based on optimal alignment of the video with its transcript; 2) negative soft alignments $\boldsymbol{Y}^n \in [0,1]^{N \times A}$, whose each row is the probability distribution of frame t to undesired actions. Thus, to learn the parameters of the GRU and USN, for each video, we define the loss

$$\begin{aligned} \mathcal{L}_{\text{video}} \triangleq \sum_{t=1}^{N} \sum_{a=1}^{A} \Big[& - y_{t,a}^p \big(\log(\sigma_{t,a}) + \rho \log(\psi_{t,a})\big) \\ & + y_{t,a}^n \big(\log(\sigma_{t,a}) + \rho \log(\psi_{t,a})\big) \Big], \end{aligned} \tag{4}$$

and minimize the average of this loss over all training videos with respect to the network parameters. Here, ρ controls the trade-off between the subspace assignment score σ and discriminative distance score ψ (here, $y_{t,a}^p$ and $y_{t,a}^n$ are the (t,a)-th elements of $\boldsymbol{Y}^p$ and $\boldsymbol{Y}^n$, respectively). The loss function aims to maximize the embedding norms and the closeness between $\boldsymbol{h}_t$ and the associated subspace based on the positive alignment $\boldsymbol{Y}^p$ while minimizing those to the incorrect subspaces based on $\boldsymbol{Y}^n$. Notice that these two scores have complementary effects, where ψ enforces learned subspaces to be more distinct, while σ prevents shrinking parameters and features towards zero. One can also choose different subspace dimensions for different actions, depending on the appearance and motion complexity of the action. In the experiments, we explore the effect of subspace dimensions on the performance of our method.

Remark 1 *While autoencoders have been used in unsupervised subspace clustering methods [22, 72, 46], their roles are fundamentally different than our USN. In such works a* single *autoencoder is used for feature learning for all data followed by applying a self-expressive layer [22] and spectral clustering on the similarities built from embeddings of the autoencoder. The self-expressive layer learns subspaces yet requires the entire dataset at each training iteration. In contrast, we use a GRU for feature learning and* multiple *autoencoders, one per action class, to learn subspaces. Moreover, our method can be trained with video batches and directly predict actions.*

3.2. Proposed Alignment Algorithm

In this section, we discuss our algorithm to find the optimal alignment of a transcript with a video then to form positive and negative soft alignments for network training.

Finding Optimal Transcript Alignment. Given the transcript $\mathcal{T} = (a_1, \ldots, a_n)$ of a video, our goal is to find the best alignment that assigns each frame to one action in the transcript in order. Notice that an alignment can be fully determined by finding the lengths of actions in the transcript. Let l_i denote the length of action a_i, where we must have $\sum_i l_i = N$. To find the optimal alignment, we obtain the subspace assignment scores $\sigma_{t,a}$ in (2) and search for $\{l_i\}_{i=1}^n$ that give the best total assignment score over the video via an optimization algorithm, i.e., we solve

$$\min_{\{l_i\}, \sum_i l_i = N} \sum_{i=1}^{n} \left[\gamma \mathcal{L}_{reg}(l_1, ..., l_n) + \sum_{t=L_i+1}^{L_i+l_i} -\log(\sigma_{t,a_i}) \right]. \tag{5}$$

Here, $L_i \triangleq \sum_{j=1}^{i-1} l_j$ is the total length of actions prior to a_i (we set $L_1 = 0$), $\mathcal{L}_{reg}$ is a regularization term preventing degenerate alignments and the hyperparameter γ sets a trade-off between negative likelihood and regularization.[2]

Given that our method alternates between learning subspaces and features and finding alignments, it is possible that the alignment assigns majority of the frames to one action and few frames to other actions in the transcript, or one action has drastically different durations across videos. Therefore, we design $\mathcal{L}_{reg}$ to prevent such undesired solutions. Let $p(a)$ denote the estimated frequency of action a to occur and $p_a(l)$ denote the probability of action a having length l. We define

$$\mathcal{L}_{reg} = \underbrace{\sum_{i=1}^{n} l_i \log\big(p(a_i)\big)}_{\triangleq \mathcal{L}_{reg}^1} + \underbrace{\sum_{i=1}^{n} - \log\big(p_{a_i}(l_i)\big)}_{\triangleq \mathcal{L}_{reg}^2}, \tag{6}$$

where $\mathcal{L}_{reg}^1$ penalizes imbalanced segmentation within a video by incurring a large cost when most frames are assigned to a frequent action. On the other hand, $\mathcal{L}_{reg}^2$ ensures each action has similar lengths across videos, as it is

[2] It is also possible to include $\log(\psi_{t,a_i})$ in (5), yet we found excluding it yields better performance.

often the case that the length of the same action is roughly consistent in topic-related videos. We model $p_a(l) = \lambda_a^l \exp(-\lambda_a)/l!$ by a Poisson distribution [49] with a parameter λ_a denoting the mean action length. We will estimate both $p(a)$'s and λ_a's at the same step when we learn the USN parameters (see below also for more details).

Remark 2 *Notice that $\mathcal{L}_{reg}^1$ and $\mathcal{L}_{reg}^2$ have complementary effects. While the alignments are balanced within a video, durations of the one action can be different across videos. On the other hand, while the same action's durations are roughly similar across videos, one could obtain imbalanced alignments within videos. Thus, the two terms ensure balancedness within and duration consistency across videos.*

Naively searching for the optimal alignment that solves (5) is exponentially complex, due to combinatorial number of possibilities for $\{l_i\}_{i=1}^n$. Therefore, we employ a constrained Viterbi decoding algorithm [49, 37] to efficiently solve the problem. Specifically, the objective function of (5) can be computed using recursive function evaluations

$$U(a_i, t) = \max_{l_i > 0} U(a_{i-1}, t - l_i) - \sum_{t'=t-l_i}^{t} \log(\sigma_{t',a_i}) + l_i \log(p(a_i)) - \log(p_{a_i}(l_i)), \quad (7)$$

where $U(a_n, N)$ corresponds to the minimum objective value. By backtracking through the recursion, we can find the optimal alignment $\{l_i^*\}$.

Constructing Positive and Negative Soft Alignments. One can use the optimal alignment $\{l_i^*\}$ directly to train the network. However, this has the drawback that a network with enough capacity could overfit to alignment errors at the early training stage. On the other hand, we observe that, given the optimal alignment $\{l_i^*\}$, alignments with small shifts in action boundaries can potentially be valid candidates as well (even humans have difficulty discerning the boundaries between actions). Moreover, those possible valid alignments must be preferred over invalid alignments that use different transcripts than the ground-truth, e.g., using $\mathcal{T}' = (a_1', \dots, a_n')$, where $a_i' \neq a_i$ for all i.

To allow our model to explore multiple candidate alignments and to better distinguish between alignments with valid/invalid transcripts, we generate positive and negative soft alignment matrices, $\boldsymbol{Y}^p \in [0,1]^{N \times A}$ and $\boldsymbol{Y}^n \in [0,1]^{N \times A}$. More specifically, starting from the optimal alignment $\{l_i^*\}$, we generate candidate valid alignments $\{\boldsymbol{R}_k^p\}$ and invalid alignments $\{\boldsymbol{R}_k^n\}$ using [36], as shown in figure 2. $\boldsymbol{R}_k^p \in \{0,1\}^{N \times A}$ is a discrete label matrix encoding k-th alignment and similarly for $\boldsymbol{R}_k^n$. Its (t,a) entry equals to 1 if frame t is assigned action a(each row has only one 1). To further incorporate the candidate alignments' likelihood, we propose to measure the likelihood score by computing the inner product

$$s(\boldsymbol{R}_k^p) \triangleq \langle \boldsymbol{R}_k^p, \boldsymbol{\Delta} \rangle, \quad \boldsymbol{\Delta} \triangleq \big[\log(\sigma_{t,a})\big] \in \mathbb{R}_-^{N \times A}, \quad (8)$$

which measures the likelihood of the k-th alignment path according to the learned subspace assignment scores $\sigma_{t,a}$. We then form the positive soft alignment matrix by computing the weighted average

$$\boldsymbol{Y}^p \triangleq \sum_k \alpha_k \boldsymbol{R}_k^p, \quad \alpha_k \triangleq \frac{\exp(s(\boldsymbol{R}_k^p))}{\sum_j \exp(s(\boldsymbol{R}_j^p))}. \quad (9)$$

Similarly, we compute the score of the k-th negative alignment $s(\boldsymbol{R}_k^n) \triangleq \langle \boldsymbol{R}_k^n, \boldsymbol{\Delta} \rangle$ and the negative soft alignment matrix $\boldsymbol{Y}^n$ as the weighted average of $\{\boldsymbol{R}_k^n\}$. We will use these two matrices to train our network via (4).

3.3. Learning and Inference

Our learning method alternates between the two steps of training the networks using positive/negative soft alignments and computing video alignments using the trained network. We initialize $p(a) = 1/A$ and $\lambda_a = 1$. At each iteration, we randomly sample one video and compute its optimal alignment with its transcript, $\{l_i^*\}$, as well as the soft alignment matrices, Y^p and Y^n. These matrices will be used to train the network. We use the optimal alignment $\{l_i^*\}$ from the current and previous iterations to update the estimation of $p(a)$, as the average number of frames across videos assigned to a, and λ_a, as the average length of the action a across videos, which will affect the constrained Viterbi decoding in the next iteration.

Inference via Representative Transcripts. During inference, for the *action alignment*, where we have the transcript of a test video, we run the alignment algorithm and choose the best $\boldsymbol{R}_k^{p*}$ given by the transcript that has the maximum likelihood score $s(\boldsymbol{R}_k^{p*})$ as the video alignment. For *action segmentation*, where the test video's transcript is unknown, for most datasets, we follow [49, 36] and run the alignment with every transcript of the training videos, each giving us a $\boldsymbol{R}_k^{p*}$ with its likelihood $s(\boldsymbol{R}_k^{p*})$. The $\boldsymbol{R}_k^{p*}$ with the highest likelihood score is chosen as the video alignment. However, this incurs a large computational cost if a dataset contains thousands of unique training transcripts (e.g., CrossTask [73] has 2,026 transcripts).

To handle large number of transcripts, we propose a hierarchical segmentation method: 1) We use the facility location subset selection algorithm [26] (see the supplementary materials for more details) to group all training transcripts into C groups based on the normalized edit distances between each pair of the transcripts, which is computed as $2 \times \mathit{edit}(\mathcal{T}_1, \mathcal{T}_2)/(|\mathcal{T}_1| + |\mathcal{T}_2|)$. Here, $\mathit{edit}(\cdot,\cdot)$ denotes the Levenshtein distance [34]. Thus, each group will also have a representative transcript. 2) We run the alignment algorithm between the test video and each of the C representative transcripts and find the best matching representative that yields the new $\boldsymbol{R}_k^{p*}$ with maximum likelihood score. 3)

The video is aligned with each of the transcripts in the group of the matched representative transcript and $\boldsymbol{R}_k^{p*}$ with maximum likelihood score is chosen as the final video alignment.

Remark 3 *Our method can be viewed as minimizing the average of the unified objective function*

$$\gamma\mathcal{L}_{reg}+\frac{1}{N}\sum_{t,a}\Big[(-y^p_{t,a}+y^n_{t,a})\times\big(\log(\sigma_{t,a})+\rho\log(\psi_{t,a})\big)\Big], \tag{10}$$

over all videos with respect to the model parameters and alignments. When training the network, we fix the label matrices $\boldsymbol{Y}^p$ and $\boldsymbol{Y}^n$ using the given alignment, and optimize the cost to learn the network parameters. For a learned model, hence, with fixed network outputs, we find the alignment using the proposed algorithm.

4. Experiments

We evaluate the performance of our proposed TASL method, against state-of-the-art weakly supervised action segmentation algorithms, NNV [49] and CDFL [36], on the Breakfast [28], Hollywood Extended [2] and CrossTask [73] datasets. We consider both *action segmentation*, where the transcripts of test videos are unknown, and *action alignment*, where the transcript of each test video is known.

Due to the alternating nature of learning from weak supervision, the performance of existing methods, including NNV and CDFL, changes for different initializations. Therefore, current works have reported the results of their best run [58]. For a fair comparison, we run all methods using their codes for 3 different initializations and report the best run results as '*Best*' performance in the tables. However, given that in the weakly-supervised setting, one cannot in practice distinguish between good and bad initializations, in addition, we report the averaged results over runs as the '*Average*' performance in the tables.

Due to space limitations, complexity analysis, metrics discussion, comparison with subspace clustering baselines and more results are provided in supplementary materials.

4.1. Experimental Setup

Datasets. We perform experiments on three large datasets. The *Breakfast* [28] dataset consists of 1,712 videos of people performing 10 different cooking activities. It has 48 different actions, including a 'background' class to denote non-action frames. On average a video has 6.9 actions and 7.3% background frames. The *Hollywood Extended*[2] dataset contains 937 videos of people performing actions such as *walk*, *sit* and *answer phone*. Overall there are 16 actions and on average 2.5 actions per video, while 60.9% of frames are background. The *CrossTask* [73] dataset contains videos from 18 primary tasks. We use the 14 cooking-related tasks, which include 2,552 videos and 80 different actions. Each video has 14.4 actions on average, while 74.8% of frames correspond to background.

For Breakfast, we use the four released training/test splits of the dataset. For Hollywood, we similarly partition the videos into four splits, each split with 10% videos for testing and 90% for training. On these two datasets, we report average results over splits and, similar to prior works, use the 64-dimensional improved dense trajectory features [62] released by [49]. For CrossTask, we use the released training/test split, with 90% training and 10% testing and downsample the released features to 64 dimensions via PCA for consistency with respect to the other datasets.

Evaluation Metrics. For evaluation, we use *1) Mean-over-frame (Mof)*, which is the percentage of frames for which the predicted action labels are correct. *2) Intersection over Union (IoU)*, defined as $\frac{1}{A}\sum_a |GT_a \cap D_a|/|GT_a \cup D_a|$, where GT_a is the set of frames belonging to action a and D_a is the set of frames classified as action a. *3) IoU-bg*, which is the same as IoU but excluding the background class. *4) Intersection over Detection (IoD)*, defined as $\frac{1}{A}\sum_a |GT_a \cap D_a|/|D_a|$ and *5) IoD-bg* which is the same as IoD but excluding the background class. Notice that IoU and IoD account for class imbalance. These metrics are consistent with prior works, yet differ from [9], which considers having some overlap as true detection (see the supplementary material for evaluation under the metrics of [9]).

Implementation Details. We consider: i) **TASL(3)**, where we set $\boldsymbol{Q}_a$ in (2) to identity and $d_a = 3$, i.e., we directly use the projection magnitude on each subspace to compute the assignment scores; ii) **TASL(10,3)**, where we learn $\boldsymbol{Q}_a \in \mathbb{R}^{3\times 10}$ in (2), i.e., learn a linear combination of projections on each subspace to compute the assignment scores. We set $\rho = 0.35$ for TASL(3), $\rho = 0.2$ for TASL(10,3) and set $\gamma = 1$ for both models. For inference, we perform hierarchical segmentation with $C = 20$ on CrossTask, as it contains 2,026 training transcripts. On average each group has 100 transcripts. We do not use it for main results on Breakfast and Hollywood for a fair comparison with prior works.

4.2. Experimental Results

Comparison of TASL with prior works. Table 1 and 2 show the performance of different methods for action segmentation and alignment, respectively. For TASL, we report the results of the best subspace setting on each dataset, which is TASL(3) for Breakfast and CrossTask and TASL(10, 3) for Hollywood (see the supplementary materials for results of both settings for all datasets). A larger subspace dimension is more suitable for Hollywood as its actions have more variations, such as *fight person* and *drive car*, whereas actions in Breakfast and CrossTask have more consistent patterns, such as *stir mixture*.

Notice that TASL achieves state-of-the-art performance on all datasets for both action segmentation and alignment tasks, demonstrating that USN effectively learns discrimi-

Breakfast	Mof	IoU	IoU-bg	IoD	IoD-bg
Best					
NNV [49]	42.9	32.2	29.1	32.1	31.8
CDFL [36]	**50.8**	35.7	33.6	46.8	45.7
TASL(ours)	49.9	**36.6**	**34.3**	**47.7**	**46.4**
Average					
NNV [49]	40.2	31.2	27.7	41.4	38.9
CDFL [36]	47.2	34.1	31.3	44.9	43.7
TASL(ours)	**47.8**	**35.2**	**32.6**	**46.1**	**44.5**
Hollywood	Mof	IoU	IoU-bg	IoD	IoD-bg
Best					
NNV [49]	44.4	23.2	13.1	34.5	17.8
CDFL [36]	40.7	22.2	15.1	36.1	19.0
TASL(ours)	**46.6**	**25.2**	**15.3**	**37.7**	**21.3**
Average					
NNV [49]	43.1	22.2	11.8	33.7	16.2
CDFL [36]	39.9	21.6	**14.1**	35.3	18.0
TASL(ours)	**43.7**	**23.4**	13.6	**35.7**	**18.3**
CrossTask	Mof	IoU	IoU-bg	IoD	IoD-bg
Best					
NNV [49]	27.0	11.0	8.5	24.4	10.1
CDFL [36]	32.5	11.8	7.7	24.0	9.6
TASL(ours)	**42.7**	**14.9**	**9.2**	**25.5**	**11.3**
Average					
NNV [49]	26.5	10.7	7.9	24.0	9.4
CDFL [36]	31.9	11.5	7.5	23.8	9.3
TASL(ours)	**40.7**	**14.5**	**8.9**	**25.1**	**11.0**

Table 1: Action Segmentation Performance on Three Datasets.

Breakfast	Mof	IoU	IoU-bg	IoD	IoD-bg
Best					
NNV [49]	59.5	47.0	47.7	61.7	65.0
CDFL [36]	**67.6**	50.5	51.3	65.1	**69.5**
TASL(ours)	65.8	**51.0**	**51.9**	**65.5**	69.1
Average					
NNV [49]	55.9	45.2	45.6	60.1	63.4
CDFL [36]	62.1	47.8	48.4	63.1	67.1
TASL(ours)	**64.1**	**49.9**	**50.7**	**64.7**	**68.2**
Hollywood	Mof	IoU	IoU-bg	IoD	IoD-bg
Best					
NNV [49]	61.5	35.9	26.4	51.3	41.5
CDFL [36]	60.2	36.9	**31.5**	51.1	40.9
TASL(ours)	**63.7**	**38.3**	30.7	**53.2**	**43.0**
Average					
NNV [49]	59.8	35.0	25.4	49.9	39.6
CDFL [36]	59.5	36.5	**30.7**	51.7	40.2
TASL(ours)	**62.2**	**37.7**	30.0	**52.4**	**41.7**
CrossTask	Mof	IoU	IoU-bg	IoD	IoD-bg
Best					
NNV [49]	34.6	15.3	11.4	27.5	14.0
CDFL [36]	46.7	17.2	11.5	28.0	14.5
TASL(ours)	**57.1**	**19.1**	**11.7**	**28.9**	**15.8**
Average					
NNV [49]	34.3	15.1	11.3	27.1	13.4
CDFL [36]	43.4	17.0	11.3	27.6	14.3
TASL(ours)	**54.6**	**18.8**	**11.5**	**28.2**	**15.2**

Table 2: Action Alignment Performance on Three Datasets.

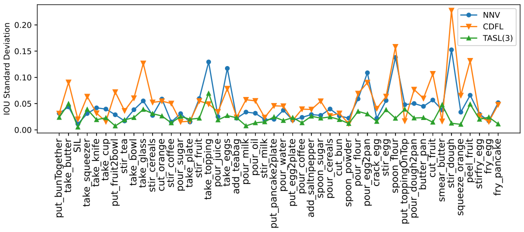

Figure 3: Standard deviation of IoU on Breakfast actions for alignment.

native action subspaces and can adjust subspace dimensions according to the complexity of a dataset. For the more difficult task of action segmentation, TASL exceeds CDFL by 1.1% and 1.2% at IoU and IoD respectively on Breakfast for 'Average'. On the more challenging CrossTask dataset, our method outperforms CDFL by 8.8% and 3% for MoF and IoU. On Hollywood, TASL significantly improves CDFL by 3.8% and 1.8% for MoF and IoU. However, it is still possible to adjust the dimension of each action subspace in our method for further improvement, as we show below.

Notice that on Breakfast, TASL obtains larger improvements over the state of the art for 'Average' than 'Best'. This is due to the fact that the low-dimensional subspace assumption for actions regularizes the training and makes our model more robust to random initializations. Figure 3 shows the standard deviation of IoU over multiple initializations on Breakfast for the alignment task, demonstrating that TASL obtains the lowest variance on most actions.

Subspace dimension effect. Table 3 (left) shows the performance of TASL can further improve by reducing the subspace dimension for background. The first row shows the previous results of TASL(10, 3) on Hollywood. We change TASL(10, 3) by allowing the background subspace dimension to be $d_{bg} \in \{1, 3, 10\}$ and Q_{bg} be identity, while keeping dimensions of other action subspaces intact. Since background occupies 60.9% of frames in Hollywood and contains large visual appearance variations, a larger d_{bg} allows us to capture its complex variations while preventing overfitting (see supplementary materials for similar results on Breakfast). In fact, with d_{bg}=10, we further improve IoD/IoD-bg of TASL than what we reported in Table 1 and 2. Table 3 (middle) shows the robustness of results for changing subspace dimensions of all actions for action segmentation on Breakfast. While TASL(3) has the best performance, other dimensions achieve competitive performance with less than 0.5% difference.

Inference with representative transcripts. Table 4 compares the average inference time on a test video and the average accuracy on Breakfast using our hierarchical segmentation. We set the number of groups $C = 20$ and, on average, each group contains about 10 transcripts. We compare the result of hierarchically segmentation ('Hier') and the result of using only the representative transcripts ('Rep'), i.e, returning the segmentation via the best matched representative transcript. Notice that using 'Rep' ('Hier'), the average segmentation time for one video improves by a factor of 13 (4). Morevover, Both 'Rep' and 'Hier' have only less than 1% drop on IoU and IoD, showing our method can be extended to real-time application with minor accuracy loss.

Ablation studies. Table 3 (right) shows the effect of each

Hollywood Ext.	IoD	IoD-bg
TASL(10,3)	35.7	18.3
d_{bg} = 1	35.1	17.4
d_{bg} = 3	35.5	18.0
d_{bg} = 10	**36.0**	**19.2**

Breakfast	IoU	IoU-bg	IoD	IoD-bg
TASL(3)	**35.2**	32.6	**46.1**	**44.5**
TASL(5)	35.1	**32.9**	**46.1**	44.0
TASL(10,3)	35.1	32.7	46.0	44.1

σ	ψ	$\boldsymbol{Y}$	$\mathcal{L}_{reg}$	IoU	IoU-bg	IoD	IoD-bg
×	○	✓	✓	11.0	3.1	22.0	7.8
✓	○	✓	✓	15.5	11.0	24.8	23.0
✓	✓	×	✓	27.1	23.8	37.5	36.1
✓	✓	✓	×	3.5	0.5	7.1	5.8
✓	✓	✓	✓	35.2	32.6	46.1	44.5

Table 3: Left: Effect of 'Background' subspace dimension for TASL(10, 3) on Hollywood. Middle: Effect of hierarchical segmentation for TASL(3) on Breakfast. Right: Effect of different components of our method, TAS(3), on Breakfast. Results in all tables are for the segmentation task.

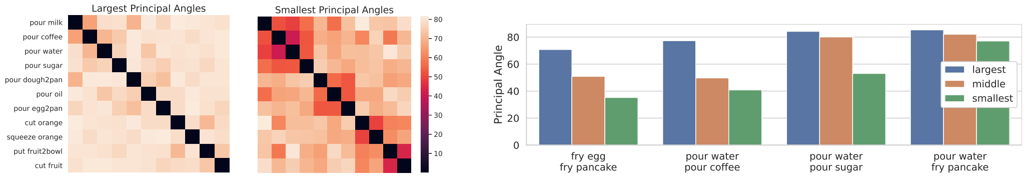

Figure 4: Left:the largest and smallest principal angles between the subspaces learned by TASL(3) for a subset of Breakfast actions. Right: principal angles between four pairs of action subspaces learned by TASL(3).

Breakfast	Inf. Time	IoU	IoU-bg	IoD	IoD-bg
TASL(3) - Rep	0.1s	35.0	32.0	45.9	43.1
TASL(3) - Hier	0.3s	35.1	32.1	46.1	43.7
TASL(3)	1.3s	35.2	32.6	46.1	44.5

Table 4: Effect of hierarchical segmentation process for TASL(3) on Breakfast for action segmentation.

TASL component. For the network loss in (4), we compare excluding scores and training with only the optimal alignment $\{l_i^*\}$ instead of soft alignments. For the Viterbi cost in (5), we investigate excluding $\mathcal{L}_{reg}$. We also test $\psi_{t,a} = \exp(-\|\hat{\boldsymbol{h}}_{t,a} - \boldsymbol{h}_t\|_2^2)$, denoted with ∘ sign, to better show the effect of our discriminative distance score in (3). The first and second rows show that optimizing $\sigma_{t,a}$ is important to learn discriminative subspaces. Moreover, simply minimizing the distance $\|\hat{\boldsymbol{h}}_{t,a} - \boldsymbol{h}_t\|$, instead of the discriminative distance score in (3), will shrink features towards zeros, thus, loses distinctive representations. On the other hand, training TASL without positive and negative soft alignments (third row) results in overfitting to poor initial segmentations. Finally, excluding $\mathcal{L}_{reg}$ leads to imbalanced segmentations and actions with inconsistent lengths, significantly reducing the accuracy.

Learned Subspace Angles. Figure 4 (left) shows the largest and smallest angles [23] between the learned subspaces for a subset of Breakfast actions. Notice that the largest angles are all nearly 90°, meaning that subspaces are mutually orthogonal in at least one dimension, which guarantees features of actions are discriminative. Moreover, the smallest angles show that our method captures semantic similarities between actions (e.g., the group of similar actions on the upper left block are about 'pouring', and the lower right ones are about 'fruit'/'orange'). Figure 4 (right) shows subspace angles between four action pairs. Notice that actions in *(fry egg, fry pancake) or (pour water, pour coffee)* are similar, thus two angles are small. Also, *(pour water, pour sugar)* are similar only in verb and have one small angle. On the other hand, *(pour water, fry pancake)* are different actions, thus all three angles are large (see supplementary materials for more comprehensive results).

Quantitative Results. Figure 5 shows action segmentation (top) and alignment (bottom) generated by NNV, CDFL and TASL(3) against the ground-truth (GT) on two videos from Breakfast. On both videos, TASL is more accurate at detecting actions and their boundaries. Specifically, TASL accurately classifies short (quick) actions, such as *pour milk* in video 1 and *add salt & pepper* or *pour oil* in video 2.

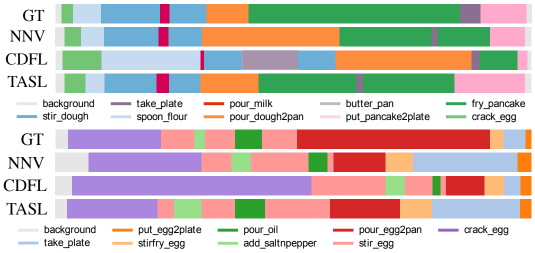

Figure 5: Results of NNV, CDFL, TASL(3) against ground-truth, on two Breakfast videos for action segmentation (top) and alignment (bottom).

5. Conclusions

We addressed learning to segment actions in videos using weakly-annotated data. We modeled actions by low-dimensional subspaces using an ensemble of autoencoders and proposed an efficient alignment algorithm by generating soft positive and negative alignments and introducing a regularization to prevent unbalanced segmentations within and across videos. We proposed an efficient method to significantly reduce the inference time. By experiments on Breakfast, Hollywood Extended and CrossTask datasets, we showed our method improves the state of the art.

Acknowledgements

This work is supported by NSF (IIS-2115110, IIS-1657197), DARPA Young Faculty Award (D18AP00050), ONR (N000141812132) and ARO (W911NF1810300, W911NF2110276).

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

[2] 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

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis. *Journal of the ACM*, 58(1):1–37, 2010. 1

[4] 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

[5] Ying Chen, Chun-Guang Li, , and Chong You. Stochastic sparse subspace clustering. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[6] Anoop Cherian, Basura Fernando, Mehrtash Harandi, and Stephen Gould. Generalized rank pooling for activity recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[7] 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. 2

[8] Zhiyuan Dang, Cheng Deng, Xu Yang, and Heng Huang. Multi-scale fusion subspace clustering using similarity constraint. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[9] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 6

[10] E. Elhamifar and D. Huynh. Self-supervised multi-task procedure learning from instructional videos. *European Conference on Computer Vision*, 2020. 1, 2

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

[12] E. Elhamifar and R. Vidal. Sparse subspace clustering. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1

[13] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 2

[14] Mohsen Fayyaz and Jurgen Gall. Sct: Set constrained temporal transformer for set supervised action segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[15] D. F. Fouhey, W. C. Kuo, A. A. Efros, and J. Malik. From lifestyle vlogs to everyday interactions. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[16] Z. Ghahramani and G. E. Hinton. The em algorithm for mixtures of factor analyzers. *Technical Report CRG-TR-96-1, Dept. Computer Science, Univ. of Toronto*, 1996. 2

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

[18] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 2

[19] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. *IEEE International Conference on Computer Vision*, 2017. 1

[20] J. Ho, M. H. Yang, J. Lim, K.C. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003. 2

[21] D. A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist temporal modeling for weakly supervised action labeling. *European Conference on Computer Vision*, 2016. 1, 2

[22] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid. Deep subspace clustering networks. *Neural Information Processing Systems*, 2017. 2, 4

[23] Andrew Knyazev and Merico Argentati. Principal angles between subspaces in an a-based scalar product: Algorithms and perturbation estimates. *Industrial and Applied Mathematics*, 2002. 8

[24] O. Koller, H. Ney, and R. Bowden. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[25] O. Koller, S. Zargaran, and H. Ney. Re-sign: Re-aligned end-to-end sequence modeling with deep recurrent cnn-hmms. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[26] Andreas Krause and Daniel Golovin. Submodular function maximization. Cambridge University Press, 2014. 5

[27] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. *IEEE International Conference on Computer Vision*, 2017. 1

[28] 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. 6

[29] 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

[30] H. Kuehne, A. Richard, and J. Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding Journal*, 2017. 2

[31] 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

[32] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1

[33] 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

[34] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710, 1965. 5

[35] C. G. Li and R. Vidal. Structured sparse subspace clustering: A unified optimization framework. *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[36] J. Li, P. Lei, and S. Todorovic. Weakly supervised energy-based learning for action segmentation. *International Conference on Computer Vision*, 2019. 1, 2, 5, 6, 7

[37] Jun Li and Sinisa Todorovic. Set-constrained viterbi for set-supervised action segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 5

[38] Sheng Li, Kang Li, and Yun Fu. Temporal subspace clustering for human motion segmentation. *IEEE International Conference on Computer Vision*, 2015. 1

[39] M. Lin, N. Inoue, and K. Shinoda. Ctc network with statistical language modeling for action sequence recognition in videos. *Thematic Workshops of the ACM Conference on Multimedia*, 2017. 2

[40] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 1, 2

[41] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *International Conference on Machine Learning*, 2009. 1

[42] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1

[43] S. Matsushima and M. Brbic. Neural information processing systems. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[44] 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. 2

[45] V. M. Patel, H. Van-Nguyen, and R. Vidal. Latent space sparse subspace clustering. *International Conference on Computer Vision*, 2013. 2

[46] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi. Deep subspace clustering with sparsity prior. *International Joint Conference on Artificial Intelligence*, 2016. 2, 4

[47] A. Richard, H. Kuehne, and J. Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[48] 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

[49] 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, 5, 6, 7

[50] 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

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

[52] Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections. *IEEE International Conference on Computer Vision*, 2015. 2

[53] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Intra- and inter-action understanding via temporal action parsing. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[54] 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

[55] 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

[56] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *European Conference on Computer Vision*, 2016. 2

[57] 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

[58] Yaser Souri, Alexander Richard, Luca Minciullo, and Juergen Gall. On evaluating weakly supervised action segmentation methods, 2020. 6

[59] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[60] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999. 2

[61] P. Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, 2000. 2

[62] H. Wang and C. Schmid. Action recognition with improved trajectories. *International Conference on Computer Vision*, 2013. 6

[63] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb. 2009. 1

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

[65] C. Xu, S.-H. Hsieh, C. Xiong, and J. J. Corso. Can humans fly? action understanding with multiple classes of actors. *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1

[66] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[67] 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

[68] C. You, D. P. Robinson, and R. Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[69] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[70] T. Zhang, P. Ji, M. Harandi, W. Huang, and H. Li. Neural collaborative subspace clustering. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[71] T. Zhang, A. Szlam, and G. Lerman. Median k-flats for hybrid linear modeling with many outliers. In *Workshop on Subspace Methods*, 2009. 2

[72] P. Zhou, Y. Hou, and J. Feng. Deep adversarial subspace clustering. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 4

[73] 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, 5, 6