Anchor-Constrained Viterbi for Set-Supervised Action Segmentation

Jun Li
Oregon State University
liju2@oregonstate.edu

Sinisa Todorovic
Oregon State University
sinisa@oregonstate.edu

Abstract

This paper is about action segmentation under weak supervision in training, where the ground truth provides only a set of actions present, but neither their temporal ordering nor when they occur in a training video. We use a Hidden Markov Model (HMM) grounded on a multilayer perceptron (MLP) to label video frames, and thus generate a pseudo-ground truth for the subsequent pseudo-supervised training. In testing, a Monte Carlo sampling of action sets seen in training is used to generate candidate temporal sequences of actions, and select the maximum posterior sequence. Our key contribution is a new anchor-constrained Viterbi algorithm (ACV) for generating the pseudo-ground truth, where anchors are salient action parts estimated for each action from a given ground-truth set. Our evaluation on the tasks of action segmentation and alignment on the benchmark Breakfast, MPII Cooking2, Hollywood Extended datasets demonstrates our superior performance relative to that of prior work.

1. Introduction

This paper is about action segmentation by labeling video frames with action classes under weak, set-level supervision in training. In the set-supervised training, the ground truth is a set of actions present in a training video, but their temporal ordering, the number of action instances, and their start and end frames are unknown. This is an important problem arising in many recent applications, such as, those dealing with big video datasets with automatically generated set-level annotations (*e.g.*, via word-based video retrievals from Youtube), for which human annotations are not available (*e.g.* because scaling human annotations over numerous video retrieval results is difficult). Our main challenge is that the provided ground truth – being a set with arbitrarily ordered distinct labels of action classes – does not provide sufficient constraints for a reliable learning of action segmentation.

Prior work typically adopts the following framework. In training, an action model is used to label every video frame, and thus generate a pseudo-ground truth for the subsequent pseudo-supervised training of the model. In testing, a Monte Carlo sampling is first used to generate candidate temporal sequences of actions, and then the learned model is applied to identify the best scoring candidate sequence as the solution of action segmentation. Differences among existing approaches mostly lie in how the pseudo-ground truth is generated. For example, in [27], binary classifiers are independently trained for each action to label frames using multi-instance learning. Such a training, however, cannot learn temporal spans of and transitions between actions, so these parameters are heuristically set in [27] for an HMM inference on test videos. The approach in [20] casts the set-supervised training as the NP-hard "all-color shortest path" problem [1], where frame labeling of a training video is constrained such that every action label from the ground-truth set appears at least once. For generating the pseudo-ground truth, they infer an HMM using a greedy two-step algorithm called set-constrained Viterbi (SCV). The SCV first produces an initial action segmentation by running the vanilla Viterbi, and then flips low-scoring frame labels to actions that have been missed in the initial segmentation but are present in the ground-truth set. Thus, the SCV produces an approximate solution to the NP-hard problem by projection to the legal domain. There are many shortcomings of [20] related to the heuristic, greedy flipping of frame labels.

In this paper, we improve the most critical step of the above framework – generation of the pseudo-ground truth. As in [20], we also pose the set-supervised training as the "all-color shortest path" problem [1]. Our key difference is that we formulate a more effective differentiable approximation to this NP-hard problem that allows for an end-to-end training, reduces training complexity, and ultimately leads to significant performance gains over [20].

As illustrated in Fig. 1, all legal action segmentations of a training video can be represented by distinct paths in a directed video segmentation graph. A legal (or valid) path includes every action from the ground-truth set at least once. Our goal is to efficiently identify the highest-scoring valid path. For this scoring, we use an HMM grounded via a two-layer MLP onto video frames, as shown in Fig. 2. In-
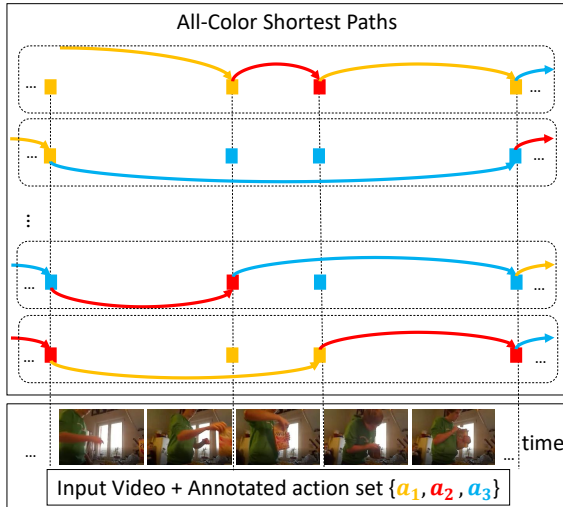
Figure 1. We cast the problem of generating pseudo-ground truth as finding an optimal "all-color shortest path" [1] in which every action from the ground-truth set occurs at least once. A video has numerous "all-color shortest paths". Our key contribution is an efficient approximation to this NP-hard problem.

Figure 2. We use an HMM to find the MAP "all-color shortest path" frame labeling of a training video, which is then taken as the pseudo-ground truth. For the MAP inference in training, we specify the new Anchor-Constrained Viterbi (ACV) algorithm, and diversity regularization which minimizes correlations of detected actions along the video. The pseudo-ground truth is used for computing the cross-entropy loss and regularization in the subsequent fully-supervised training of our HMM and MLP, enabling end-to-end training.

stead of using the solution-by-projection of [20], we specify an Anchor-Constrained Viterbi (ACV) algorithm that efficiently approximates the MAP "all-color shortest path". Efficiency comes from significantly reducing the number of valid action segmentations by considering only those that include salient action parts, called anchor segments or anchors. Importantly, the ACV enables an end-to-end training, as we use the generated pseudo-ground truth for computing the cross-entropy loss and regularization in the subsequent fully-supervised training of our HMM and MLP.

Given a training video, our approach begins by estimating saliency scores of all actions from the ground-truth set at every frame using the MLP. Then, for each action, we select the most salient frame in the video to represent a center of that action's anchor segment. Finally, we approximate a globally optimal action segmentation by running the standard Viterbi algorithm over the densely connected video graph whose nodes are frames and colored edges represent action segments (see Fig. 1). Efficiency is achieved by considering only those paths in the video graph that pass through the anchor edges (i.e., segments).

The resulting pseudo-ground truth is used in the second stage of our training, where we jointly learn parameters of the HMM and MLP using the cross entropy loss on the previously generated frame-wise pseudo-ground truth. We regularize this learning with a diversity loss aimed at maximizing a distance between saliency scores of distinct actions detected along the video. This is motivated by the following reasoning. For distinct actions, we expect that temporal patterns of their respective saliency scores across the frames
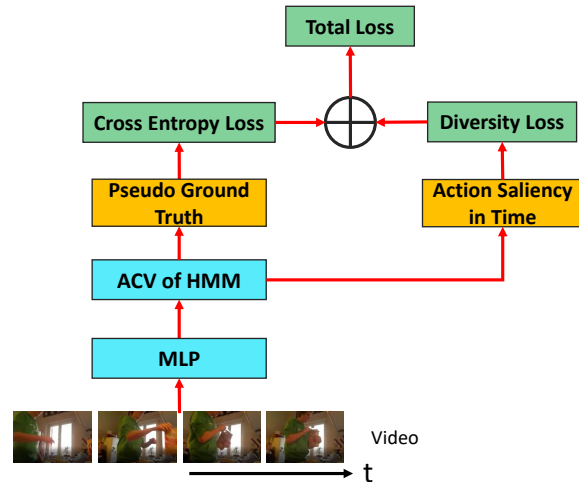
are different. Thus, our diversity loss ensures that the saliencies of every action are sufficiently discriminative in time so as to facilitate action segmentation.

In evaluation, we address action segmentation and action alignment on the Breakfast, MPII Cooking2, Hollywood Extended datasets. Our experiments demonstrate that we outperform the state of the art on both tasks.

In the following, Sec. 2 reviews related literature, Sec. 3 defines our problem and models, Sec. 4 specifies our ACV and regularization, Sec. 5 describes our inference on test videos, and Sec. 6 presents our results.

2. Related Work

This section puts our work in the context of recent approaches to weakly supervised action segmentation. Fully supervised action segmentation requires frame-wise annotations (*e.g.*, [6, 13, 16, 17, 18, 24, 29, 33, 37, 38, 42, 46]), and hence is beyond our scope. Unsupervised action segmentation is also beyond our scope (*e.g.* [15, 31]), since our inference on test videos requires annotations of action sets present in training videos.

Transcript-supervised Learning. The temporal ordering of actions present in training videos is assumed known in [2, 3, 4, 5, 8, 10, 11, 14, 19, 26, 28]. For example, Huang *et al*. [8] extended the connectionist temporal classification by taking into account a temporal coherence of features in

consecutive frames. Bojanowski *et al.* [2] relaxed discriminative clustering based on a conditional gradient (Frank-Wolfe) algorithm for action alignment. Other approaches [19, 26, 28] formulated action segmentation with a statistical language model. Our training setting is less constrained, and hence more challenging. This motivates our simpler model choice in comparison with more complex deep architectures of prior work (*e.g.*, recurrent network in [19, 28]).

Set-supervised Learning. The set of actions present in training videos is assumed known in [9, 21, 22, 23, 25, 32, 34, 35, 36, 40, 41, 45, 43, 44, 7]. For example, Shou *et al.* [32] specified the outer-inner-contrastive loss for learning an action boundary detector, Nguyen *et al.* [23] defined a background-aware loss to distinguish actions from the background, and Paul *et al.* [25] proposed an action affinity loss for multi-instance learning. As shown in [20, 27], all of these approaches are specifically designed to address videos with sparse and rather few action occurrences. In contrast, we consider videos with dense actions and significantly more action occurrences. Similar to our approach, in [7], temporal segments in the video are adaptively identified and labeled, but they do not use an HMM and do not sample valid sequences of actions for inference in testing.

In [20], an HMM and a Viterbi-like algorithm are also used for estimating the frame-wise pseudo-ground truth, as in our approach. Their Set-Constrained Viterbi (SCV) algorithm uses a greedy post-processing step for ensuring that all actions from the set-level ground truth are included in the frame-wise pseudo-ground truth. This post-processing is heuristic, and thus suboptimal. In contrast, we formulate the ACV algorithm that efficiently approximates the "all-color shortest path" and does not require post-processing. The ACV first estimates salient anchor frames and constructs an anchor-constrained graph, and then runs the vanilla Viterbi algorithm for optimization on the anchor-constrained graph. Unlike SCV, our ACV integrates both the estimation of salient anchors and anchor-constrained Viterbi in a unified end-to-end training.

3. Our Problem Statement and Models

3.1. Problem Formulation.

A training video with length T is represented by a sequence of frame features, $\boldsymbol{x} = [x_1, ..., x_t, ..., x_T]$, extracted in an unsupervised manner as in [20, 27]. Each training video is annotated with a ground-truth set of actions $C = \{c_1, \cdots, c_M\} \subseteq \mathcal{C}$, where $\mathcal{C}$ is the set of all actions, $\mathcal{C} = \{1, 2, ..., |\mathcal{C}|\}$. Our goal is to find an optimal action segmentation of every training video, $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$, where $\hat{\boldsymbol{c}} = [\hat{c}_1, ..., \hat{c}_n, ..., \hat{c}_{\hat{N}}]$ is the predicted temporal sequence of actions, $\hat{c}_n \in \mathcal{C}$, and $\hat{\boldsymbol{l}} = [\hat{l}_1, \cdots, \hat{l}_{\hat{N}}]$ are their corresponding temporal lengths. $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$ serves as our pseudo-ground truth for the subsequent pseudo-supervised training.

3.2. HMM

We use an HMM to estimate the MAP $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$ as

$$
\begin{aligned}
(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}}) &= \arg\max_{N, \boldsymbol{c}, \boldsymbol{l}} \ p(\boldsymbol{c}, \boldsymbol{l}|\boldsymbol{x}) = \arg\max_{N, \boldsymbol{c}, \boldsymbol{l}} \ p(\boldsymbol{c})p(\boldsymbol{l}|\boldsymbol{c})p(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{l}) \\
&= \arg\max_{N, \boldsymbol{c}, \boldsymbol{l}} \left[p(c_1) \prod_{n=1}^{N-1} p(c_{n+1}|c_n) \right] \cdot \left[\prod_{n=1}^{N} p(l_n|c_n) \right] \\
&\quad \cdot \left[\prod_{t=1}^{T} p(x_t|c_{n(t)}) \right],
\end{aligned}
\tag{1}
$$

where $p(c_1)$ is assumed equal for all action classes.

We study two versions of the HMM model defined in (1). One is the initial HMM, $\mathcal{H} = \mathcal{H}^0$, computed directly from the ground-truth action set, and the other is a refined HMM, $\mathcal{H}^i$, iteratively updated after i training iterations starting from the initial $\mathcal{H}^0$. In the following, our notation for $\mathcal{H}^0$ and $\mathcal{H}^i$ uses 0 and i in the superscript to indicate the training iterations, respectively.

The Initial HMM. The initial transition probability in (1) is defined as

$$
p^0(c_{n+1}|c_n) = \frac{\#(c_{n+1}, c_n)}{\#(c_n)},
\tag{2}
$$

where $\#(\cdot)$ is the number of actions or pairs of actions in the ground truth.

The initial action length in (1) is modeled as the Poisson distribution:

$$
p^0(l|c) = \frac{(\lambda_c^0)^l}{l!} e^{-\lambda_c^0},
\tag{3}
$$

where λ_c^0 is the expected temporal length of action $c \in \mathcal{C}$. For every c, we estimate λ_c^0 such that the total length of all actions from C is close to the video's length T. More formally, we estimate λ_c^0 by minimizing the following quadratic objective:

$$
\text{minimize} \quad \sum_v \left(T_v - \sum_{c \in C_v} \lambda_c^0 \right)^2,
$$
$$
\text{s.t.} \quad \text{for every } c: \ \lambda_c^0 > l_{\min},
\tag{4}
$$

where v is the index of videos, $l_{\min}$ is the minimum action length ($l_{\min} = 50$ frames in our experiments).

The initial likelihood at frame t in (1) is estimated as:

$$
p^0(x_t|c) \propto \frac{p(c|x_t)}{p^0(c)}, \qquad p^0(c) = \frac{\sum_v T_v \cdot 1(c \in C_v)}{\sum_v T_v}.
\tag{5}
$$

where $p(c|x_t)$ is a softmax score of the MLP, and $1(\cdot)$ is the indicator function. $p^0(c)$ in (5) is a percentage of the video footage having c in the ground truth.

The Refined HMM is initialized to $\mathcal{H}^0$ as in (2)–(5), and then for each training video updated in ith training iteration based on the MAP assignment $(\hat{c}^i, \hat{l}^i)$ for that video as

$$p^i(c_{n+1}|c_n) = p^{i-1}(c_{n+1}|c_n)$$
$$+ \frac{1}{V}\left(\frac{\#(\hat{c}_n^i, \hat{c}_{n+1}^i)}{\#(\hat{c}_n^i)} - p^{i-1}(c_{n+1}|c_n)\right), \quad (6)$$

$$\lambda_c^i = \lambda_c^{i-1} + \frac{1}{V}\left(\frac{\sum_{n=1}^{\hat{N}} \hat{l}_n^i \cdot 1(c = \hat{c}_n^i)}{\sum_{n=1}^{\hat{N}} 1(c = \hat{c}_n^i)} - \lambda_c^{i-1}\right), \quad (7)$$

$$p^i(c) = p^{i-1}(c) + \frac{1}{V}\left(\frac{\sum_{n=1}^{\hat{N}} \hat{l}_n^i \cdot 1(c = \hat{c}_n^i)}{T} - p^{i-1}(c)\right). \quad (8)$$

where V is the total number of training videos. We use $\frac{1}{V}$ as our learning rate, since after we compute (6)–(8) for all V training videos, the updates become properly normalized. In (6), $\#(\hat{c}_n^i)$ is the number of all segments labeled with $\hat{c}_n^i \in \mathcal{C}$, and $\#(\hat{c}_n^i, \hat{c}_{n+1}^i)$ is the total number of pairs of two consecutive video segments labeled with $\hat{c}_n^i \in \mathcal{C}$ and $\hat{c}_{n+1}^i \in \mathcal{C}$, respectively. The update is done for each training video. This means that our choice of the update rate as $\frac{1}{V}$ for the updates in (6)–(8) amounts to averaging over the pseudo-ground truth for all previously trained videos. In our experiments, the small update rate $\frac{1}{V}$ gives the "stable" and reliable updates in training.

3.3. The MLP Network

As in [27], our two-layer MLP scores frames for all action classes. The MLP's hidden layer $h \in \mathbb{R}^{n_h}$, $n_h = 256$, is input to the binary classifier $f_c(x)$, for every $c \in \mathcal{C}$, as

$$h(x) = \text{ReLU}(W^1 x + b^1), \quad (9)$$

$$f_c(x) = \sigma(W_c^\top h(x) + b_c), \ c \in \mathcal{C}, \quad (10)$$

where $W_c \in \mathbb{R}^{n_h}$, $b_c \in \mathbb{R}^1$, and σ is the sigmoid function.

$p(c|x)$ in (5) is estimated as a softmax score of the MLP's binary classifier $f_c(x)$ at frame x. To ensure reliable $p(c|x)$, we pretrain the MLP as in [27]. Specifically, we pretrain each $f_c(x)$, $c \in \mathcal{C}$, using multi-instance learning.

4. Our Set-Supervised Training

Our training consists of two stages. First, we generate the MAP pseudo-ground truth $(\hat{c}, \hat{l})$ for every training video with our ACV algorithm. Second, the pseudo-ground truth is used to estimate the cross-entropy loss and regularization for updating the HMM and MLP. Below, we first formulate our ACV, and then the loss and regularization.

4.1. Anchor-Constrained Viterbi

Given a training video and its ground truth C, our goal is to predict the MAP $(\hat{c}, \hat{l})$, such that every action $c \in C$

occurs at least once in $\hat{c}$. As shown in [1], this "all-color shortest path" problem is NP-hard. Our ACV algorithm is an efficient approximation that consists of two related steps, as illustrated in Fig. 3, where the first step constructs an anchor-constrained graph aimed at representing all valid action segmentations, and the second step finds the MAP valid segmentation in the graph.

The first step computes saliency scores $S \in \mathbb{R}^{M \times T}$ of all actions c in the ground truth C, $M = |C|$, at every frame $t = 1, \ldots, T$ using a sliding window of length $(2\tau + 1)$ as

$$S[c,t] = \sum_{u=-\tau}^{\tau} [\log f_c(x_{t+u}) - \min_{c' \in C} \log f_{c'}(x_{t+u})], \quad (11)$$

where $f_c(x_t)$ is given by (10), and $\tau = 15$ frames in our experiments. By subtracting $\min_{c'} \log f(c')$ in (11), we specify frame saliency as a difference between the highest and lowest softmax scores, instead of considering only the highest softmax scores. We have empirically observed that the latter leads to a high false-positive rate, as some frames have significantly higher softmax scores than others.

For every $c \in C$, we select its most salient frame t_c and take it to represent a center of c's anchor segment. This gives $M = |C|$ anchor segments in the video. The anchors are later used to constrain the search for an optimal "all-color shortest path" by considering only candidate action segmentations that include the anchors. A time interval centered at every t_c, $[t_c - \frac{\alpha}{2}\lambda_c, t_c + \frac{\alpha}{2}\lambda_c]$ is estimated as c's anchor segment, where λ_c is the expected length of action c given by (4), and $\alpha \in (0,1]$ is a constant. We empirically found that $\alpha = 0.6$ is optimal. Since $\alpha \in (0,1]$, the anchor segments are set to have smaller lengths than the expected lengths of the corresponding actions in order to allow some flexibility of action boundary learning. For every two anchor segments that happen to overlap, we keep the more salient anchor, and for the other action we iteratively select a new anchor as that action's second best or less salient frame until there is no overlap.

The anchor segments are used to construct a densely connected video segmentation graph, as illustrated in Fig. 3. In the graph, directed edges represent candidate actions along the video such that each edge is constrained to include one and only one of the anchor segments. Nodes in the graph represent frames that are candidate boundaries (or cuts) between actions, such that no cut can occur within the anchor segments. A directed path in the graph represents a candidate action segmentation. Therefore, by construction, every path in the anchor-constrained graph includes all actions from the ground truth C at least once, and thus represents a valid candidate for approximating our "all-color shortest path" problem.

The second step can be viewed as efficiently finding an optimal path in the anchor-constrained graph (see Fig. 3). For this, we use the Viterbi algorithm that predicts the MAP
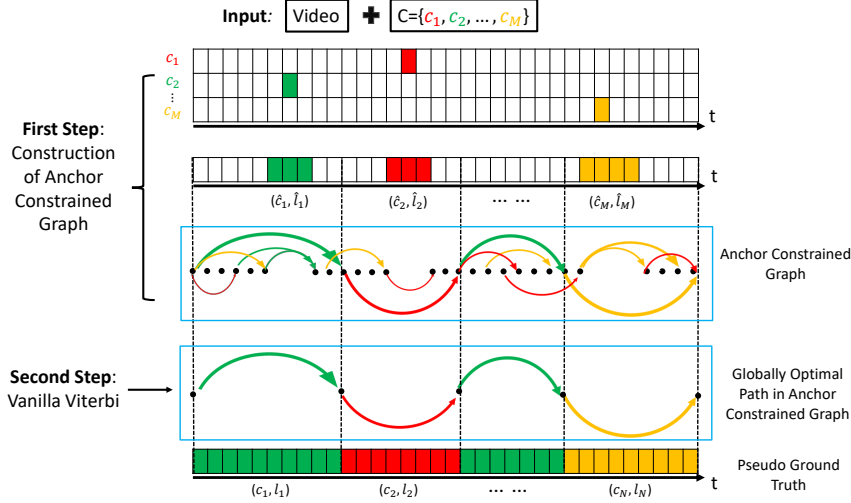
Figure 3. Our ACV consists of two steps. First, we compute saliency scores of videos frames for all actions c from the ground truth C, and for each $c \in C$ select the most salient frame, called anchor. Each anchor of action c is taken as a center of the corresponding anchor segment whose length is proportional to the expected length λ_c of c. The anchor segments are used to construct an anchor-constrained graph which efficiently represents all valid action segmentations by directed paths in the graph. Second, we run the Viterbi algorithm to efficiently identify the MAP path in the anchor-constrained graph.

$(\hat{c}, \hat{l})$ in this graph. From (1)–(8) and (10), the Viterbi is formalized as

$$(\hat{c}, \hat{l}) = \arg\max_{\substack{N, c, l \\ c \in C^N}} \sum_{n=1}^{N-1} \log p(c_{n+1}|c_n) + \sum_{n=1}^{N} \log p(l_n|c_n)$$
$$+ \sum_{t=1}^{T} \log p(x_t|c_{n(t)}). \quad (12)$$

under the constraint that all segments l_n in (12) must be selected from the edges of the anchor-constrained graph.

The Viterbi begins by computing: (i) Frame likelihoods $p(x_t|c)$ for $t = 1, \ldots, T$, using the MLP softmax scores as in (5) and (10), and (ii) Transition probabilities $p(c|c')$ for all $c, c' \in \mathcal{C}$ as in (2). Then, the Viterbi uses a recursion to efficiently compute $(\hat{c}_i, \hat{l}_i)$ in (12). Let $v(c, t)$ denote the maximum score for all action sequences c_t ending with action $c \in C$ at video frame t, where a total sum of the mean action lengths along each sequence until t is less than $1.5T$, $(\sum_{c' \in c_t} \lambda_{c'}) < 1.5T$. From (12), $v(c, t)$ can be recursively estimated as

$$v(c, t) = \max_{\substack{t' < t \\ c' \neq c \\ c' \in C}} \Big[v(c', t') + \log p(t - t'|c) + \log p(c|c')$$
$$+ \sum_{k=t'+1}^{t} \log p(x_k|c) \Big], \quad (13)$$

where $p(t-t'|c)$ is the action length likelihood given by (3), $p(c|c')$ is the transition probability from c' to c as in (2),

$p(x_k|c)$ is the likelihood at frames $k = (t'+1), \ldots, t$ given by (5). During the recursion, we enforce that time intervals $(t - t')$ must respect edges in the anchor-constrained graph (i.e., must include anchor segments). Finally, the optimal path can be back-tracked from $\max_{c \in C} v(c, T)$, resulting in an optimal action segmentation $(\hat{c}, \hat{l})$.

In our implementation, we improve efficiency by discarding early unrealistic candidate solutions with too many action changes along the path – i.e., every subpath c_t whose accumulated mean length exceeds 1.5 times the current temporal length, $\sum_{c \in c_i} \lambda_c > 1.5t$.

The predicted action segmentation is used as a pseudo-ground truth to estimate the cross-entropy loss for pseudo fully-supervised training of the HMM and MLP, as explained in the following section.

4.2. The Cross-Entropy Loss

The ACV inference result, $(\hat{c}, \hat{l})$, incurs the following binary cross-entropy loss over all action classes as

$$\mathcal{L}_{CE} = -\frac{1}{T} \Big[\sum_{t=1}^{T} \sum_{c \in \mathcal{C}} 1(\hat{c}_t = c) \log p(c|x_t)$$
$$+ 1(\hat{c}_t \neq c) \log(1 - p(c|x_t)) \Big], \ \hat{c}_t \in \hat{c} \quad (14)$$

where $p(c|x_t)$ is the MLP's softmax score for class c at frame t, given by (10).

4.3. Diversity Loss

We regularize our learning by using the action saliency scores estimated for all actions c at every frame, $\boldsymbol{S}[c, 1{:}T] =$

$[\boldsymbol{S}[c,1], ..., \boldsymbol{S}[c,t], ...\boldsymbol{S}[c,T]]$, where $\boldsymbol{S}[c,t]$ is given by (11). Distinct actions are expected to exhibit different saliency patterns along the video in $\boldsymbol{S}[c,1{:}T]$. Therefore, we specify a diversity loss, $\mathcal{L}_{DIV}$, for regularizing our learning so as to maximize a distance between every pair of $\boldsymbol{S}[c,1{:}T]$ and $\boldsymbol{S}[c',1{:}T]$, $c \neq c'$.

For a training video with the ground truth C, we define $\mathcal{L}_{DIV}$ over all pairs of distinct actions $(c,c') \in C \times C$, $c \neq c'$, as an average normalized correlation between $\boldsymbol{S}[c,1{:}T]$ and $\boldsymbol{S}[c',1{:}T]$ as

$$\mathcal{L}_{DIV} = \frac{1}{M(M-1)} \sum_{\substack{(c,c') \in C \times C \\ c \neq c'}} \frac{\boldsymbol{S}[c,1{:}T]^{\top}\boldsymbol{S}[c',1{:}T]}{\|\boldsymbol{S}[c,1{:}T]\|_2 \|\boldsymbol{S}[c',1{:}T]\|_2}. \tag{15}$$

The normalization in (15) is necessary, since the number of actions varies across training videos.

Our total loss is a sum of $\mathcal{L}_{CE}$ and $\mathcal{L}_{DIV}$:

$$\mathcal{L} = \mathcal{L}_{CE} + \beta \mathcal{L}_{DIV}, \tag{16}$$

where we experimentally found that $\beta = 0.4$ is optimal.

4.4. Complexity of Our Training

Complexity of ACV is $O(T^2|\mathcal{C}|^2)$, where T is the length of a training video, and $|\mathcal{C}|$ is the number of all action classes. This mainly comes from the Viterbi in ACV, whose complexity is $O(T^2|\mathcal{C}|^2)$. Complexity of the first step of ACV is $O(T|\mathcal{C}|) < O(T^2|\mathcal{C}|^2)$. Complexity of the diversity loss is $O(T|\mathcal{C}|^2)$. Therefore, the total complexity of our training is $O(T^2|\mathcal{C}|^2)$. The same training complexity is reported in [20].

5. Inference on a Test Video

In inference on a test video, we use a Monte Carlo sampling, as in [20, 27]. Given a test video of length T, we first randomly select one action set C from the ground truths seen in training. Then, we sequentially sample from C a temporal sequence of actions $\boldsymbol{c}$ until $\sum_{c \in \boldsymbol{c}} \lambda_c > T$. We repeat this sequential sampling when $\boldsymbol{c}$ is discarded for not including all actions from C. In this way, we end up generating K=1000 valid sequences $\mathbb{C} = \{\boldsymbol{c}\}$. For every valid $\boldsymbol{c} \in \mathbb{C}$, we use the standard dynamic programming to infer optimal action segmentation in the test video by maximizing the HMM's posterior, $(\boldsymbol{c}, \hat{\boldsymbol{l}}) = \arg\max_{\boldsymbol{l}} p(\boldsymbol{c}, \boldsymbol{l}|\boldsymbol{x})$. Among the K sequences in $\mathbb{C}$, we select the action segmentation with the maximum posterior as our final solution, $(\boldsymbol{c}^*, \boldsymbol{l}^*) = \arg\max_{\boldsymbol{c} \in \mathbb{C}} p(\boldsymbol{c}, \hat{\boldsymbol{l}}|\boldsymbol{x})$.

Complexity of our inference is $O(T^2|\mathcal{C}|K)$, where T is the length of a test video, $|\mathcal{C}|$ is the number of action classes, K is the number of generated action sequences. The same complexity is reported in [20].

6. Experiments

Datasets. As in [20, 27], for evaluation, we address the tasks of action segmentation and action alignment on three datasets, including Breakfast [12], Hollywood Extended (Ext) [2], and MPII Cooking 2 [30]. *Breakfast* consists of 1,712 videos of breakfast cooking comprising 48 actions. There are on average 6.9 action instances in each video. As in [12], a mean of frame accuracy (Mof) over the same 4-fold cross validation is reported. *Hollywood Ext* consists of 937 video clips showing a total of 16 actions. There are on average 2.5 action instances in each video. As in [14], we perform the same 10-fold cross validation and report the mean intersection over detection (IoD) averaged over the 10 folds, defined as IoD $= |GT \cup D|/|D|$, where GT and D denote the ground truth and detected action segments with the largest overlap. *MPII Cooking 2* consists of 273 videos showing cooking activities. There are 67 action classes. Following [29], the same training and testing split is used. We report the midpoint hit metric, i.e. a ratio of the midpoint of a correctly detected segment in the ground-truth segment, as in [29].

Features. For a fair comparison, our approach is evaluated on the same unsupervised video features as in [20, 27]. For all three datasets, we extract frame-wise Fisher vectors of improved dense trajectories [39] extracted over a sliding window of 20 frames. Our features are 64-dimensional projections by PCA of the Fisher vectors.

Training. Our training starts from pretraining binary classifiers in (10) with a multi-instance learning, as in [27]. Then we train our model for a total number of 100,000 iterations, where we randomly select one video in each training iteration. The learning rate is initialized to 0.01 and reduced to 0.001 at the 10,000th iteration. The refined HMM's parameters are updated after each iteration as in (6)–(8).

Ablations. We consider the following variants of our approach for evaluating the effect of each component:

- ACV = Our full approach with the diversity loss, updated transition probabilities, updated mean lengths, updated class priors given by (6)–(8).

- ACVinitial = ACV with the initial transition probabilities, initial mean lengths, initial class priors given by (2), (4), (5).

- ACVnoreg = ACV without any regularization.

- ACV+Npair= ACV without the diversity regularization, but with the N-pair loss regularization.

6.1. Action Segmentation

This section presents our action segmentation on the three datasets. As shown in Tab. 1, our ACV outperforms the state of the art by 3.2% on Breakfast, 1.0% on Cooking 2, and 3.2% on Hollywood Ext respectively. In Tab. 1, we also compare ACV with prior work that has access

Model	Breakfast (*Mof*)	Cooking2 (*midpoint*)	Holl.Ext (*IoD*)
(Set-supervised)			
Action Set [27]	23.3	10.6	9.3
SCT [7]our features	26.6	14.3	17.7
SCV [20]	30.2	14.5	17.7
Our ACV	**33.4**	**15.5**	**20.9**
(Transcript-supervised)			
OCDC [2]	8.9	-	-
HTK [14]	25.9	20.0	8.6
CTC [8]	21.8	-	-
ECTC [8]	27.7	-	-
HMM+RNN [26]	33.3	-	11.9
TCFPN [5]	38.4	-	18.3
NN-Viterbi [28]	43.0	-	-
D3TW [3]	45.7	-	-
CDFL [19]	50.2	-	25.8

Table 1. (Top) Our ACV outperforms the state-of-the-art set-level supervised approaches in terms of Mof, Midpoint, IoD. (Bottom) We compare ACV with weakly supervised approaches which have access to the true temporal ordering of actions in training. Our ACV achieves comparable results and outperforms some weakly supervised approaches. The dash means "not reported". When SCT [7] uses different 2048-dimensional frame features for Breakfast, their Mof of 30.4 is still inferior to ours.

to stronger supervision in training, where ground truth is a transcript that additionally specifies the temporal ordering of actions. ACV outperforms some recent transcript-supervised approaches.

Fig. 4 qualitatively illustrates our action segmentation on a test video *P15_cam01_P15_sandwich* from Breakfast. As can be seen, in general, ACV can detect true actions present in videos, but may miss the true locations of their start and end frames.

Figure 4. Qualitative result of action segmentation on a sample test video *P15_cam01_P15_sandwich* from Breakfast. Top row: ACV result. Bottom row: ground truth with the color-coded action sequence {cut_bun, smear_butter, put_toppingOnTop}. We mark the background frames in white. In general, ACV can detect true actions present in videos, but may miss their true locations.

Ablation — Initial vs Refined. Tab. 2 shows the results our ACV with different HMM parameters. Our ACV with updated HMM parameters outperforms the variant with initial HMM parameters – ACVinitial. The table also shows the upper-bound performance of ACV (ACV+ground

Model parameters	Breakfast (*Mof*)	Cooking 2 (*midpoint*)	Holl.Ext (*IoD*)
ACV+ground truth	36.3	16.8	22.1
ACVinitial	31.9	15.0	19.1
ACV	33.4	15.5	20.9

Table 2. Evaluation of ACV when using the HMM with the ground-truth parameters, initially estimated parameters, and updated refined parameters.

truth) when the mean action length λ_c, transition probability $p(c|c')$, and class priors $p(c)$ are estimated directly from ground-truth labels of video frames. As can be seen in Tab. 2, our iterative updating of model parameters, given by (6), in ACV is reasonable, since ACV+ground truth outperforms ACV by only 2.9% on Breakfast.

Ablation — Regularization. Tab. 3 compares our ACV with the other variants: ACVnoreg and ACV+Npair, where the latter uses the same loss as in [20]. As the N-pair loss needs two videos, as in [20], we sample two videos that share common actions in each iteration. In contrast, our ACV follows a more general training strategy where only one randomly selected video is needed. The table shows that ACV gives the best performance.

Model	Breakfast (*Mof*)	Cooking 2 (*midpoint hit*)	Holl.Ext (*IoD*)
ACVnoreg	32.1	15.1	19.9
ACV+Npair	32.4	15.1	20.2
ACV	**33.4**	**15.5**	**20.9**

Table 3. Evaluation of ACV w/out and w/ regularization.

Ablation — Anchor length. Tab. 4 shows the effect of input parameter α which controls the length of anchor segments. As can be seen, ACV with $\alpha = 0.6$ performs the best. This supports our design choice to allow flexibility in estimation of action lengths in the Viterbi (as opposed when $\alpha = 1$). Fig. 5 visualizes our sensitivity to the choice of α on the test video *P07_webcam01_P07_juice* from Breakfast. As can be seen, ACV with $\alpha = 0.6$ gives the best result.

Model	Breakfast (*Mof*)	Cooking 2 (*midpoint*)	Holl.Ext (*IoD*)
$\alpha = 0.4$	32.5	15.2	20.1
$\alpha = 0.6$	**33.4**	**15.5**	**20.9**
$\alpha = 0.8$	31.7	15.0	19.5
$\alpha = 1.0$	30.5	14.7	18.7

Table 4. Sensitivity of ACV to anchor-segment length.

Tab. 5 shows accuracy of the first step of our ACV in terms of intersection over detection (IoD). That is we estimate IoD between the estimated anchor segments and
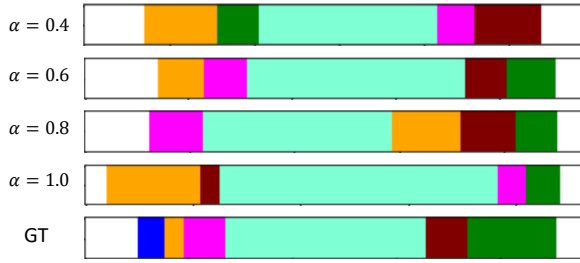
Figure 5. The effect of choosing different lengths of anchor segments controlled by α on the test video *P07_webcam01_P07_juice* from Breakfast. The predicted and ground-truth (GT) action sequences are color-coded {take_plate, take_knife, cut_orange, squeeze_orange, take_glass, pour_juice}. The background frames are marked white. ACV with $\alpha = 0.6$ gives the best performance.

Iteration	0k	20k	40k	60k	80k	100k
$\alpha = 0.4$	19.1	27.5	30.0	35.5	37.0	37.8
$\alpha = 0.6$	19.4	28.7	32.2	36.6	39.0	39.2
$\alpha = 0.8$	18.7	27.3	29.9	35.0	36.8	37.1
$\alpha = 1.0$	18.1	25.1	29.4	33.7	35.2	35.9

Table 5. IoD evaluation of salient anchors during training on Breakfast dataset.

ground truth action intervals. The table shows that $\alpha = 0.6$ gives the highest IoD, and that accuracy improves as the number of training iterations grows.

6.2. Action Alignment Given Action Sets

In the task of action alignment, we have access to the unordered set of actions C truly present in a test video. Thus, our Monte Carlo sampling can be constrained to uniformly sample only actions in the given ground-truth set C. In comparison to the state of the art, Tab. 6 shows that ACV improves action alignment in terms of Mof, Midpoint, and IoD by 4.3% on Breakfast, 1.1% on Cooking 2, 5.3% on Hollywood Ext, respectively. ACV also achieves comparable results to recent approaches that use the stronger transcript-level supervision in both training and action alignment. Fig. 6 shows a qualitative result of action alignment on the test video *0181* from Hollywood Ext. dataset. In general, ACV can successfully align the actions with video frames, but might incorrectly detect action boundaries.

7. Conclusion

In this paper, we have addressed set-supervised action segmentation and alignment. Our key contribution is a new Anchor-Constrained Viterbi (ACV) algorithm aimed at generating a frame-wise pseudo-ground truth, which is subsequently used for fully-supervised training of our HMM

Model	Breakfast (*Mof*)	Cooking 2 (*midpoint*)	Holl.Ext. (*IoD*)
(Set-supervised)			
Action Set [27]	28.4	10.6	24.2
SCV [20]	40.8	15.1	35.5
Our ACVnoreg	43.3	15.8	38.2
Our ACV	**45.1**	**16.2**	**40.8**
(Transcript-supervised)			
ECTC [8]	∼35	-	∼41
HTK [14]	43.9	-	42.4
OCDC [2]	-	-	43.9
HMM+RNN [26]	-	-	46.3
TCFPN [5]	53.5	-	39.6
NN-Viterbi [28]	-	-	48.7
D3TW [3]	57.0	-	50.9
CDFL [19]	63.0	-	52.9

Table 6. Evaluation of action alignment that has access to ground-truth action sets of test videos. Our ACV outperforms the state of the art on the three datasets. Also, ACV gives comparable results some transcript-supervised prior work uses stronger supervision in training. The dash means that the result is not reported in the respective paper.

Figure 6. Qualitative result of action alignment on a sample test video *0181* from Hollywood Ext. dataset. Top row: ACV result. Bottom row: ground truth with the color-coded action sequence {OpenCarDoor, DriveCar, DriveCar}. We mark the background frames in white. In general, ACV successfully aligns the actions with video frames, but may incorrectly detect action boundaries.

and MLP models. ACV is an efficient approximation to the NP-hard all-color shortest path problem. Efficiency comes from our estimation of anchor segments that constrain the domain of valid candidate solutions in which we efficiently find the MAP action segmentation. Our approach outperforms state of the art on three benchmark datasets for both action segmentation and alignment, without increasing complexity in training and testing. Various ablation studies demonstrate effectiveness of individual components of our model.

References

[1] Yunus Can Bilge, Dogukan Çagatay, Begüm Genç, Mecit Sari, Hüseyin Akcan, and Cem Evrendilek. All colors shortest path problem. *CoRR*, abs/1507.06865, 2015. 1, 2, 4

[2] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision*, pages 628–643. Springer, 2014. 2, 3, 6, 7, 8

[3] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3546–3555, 2019. 2, 7, 8

[4] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. *arXiv preprint arXiv:1901.02598*, 2019. 2

[5] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6508–6516, 2018. 2, 7, 8

[6] Eyrun Eyjolfsdottir, Steve Branson, Xavier P Burgos-Artizzu, Eric D Hoopfer, Jonathan Schor, David J Anderson, and Pietro Perona. Detecting social actions of fruit flies. In *European Conference on Computer Vision*, pages 772–787. Springer, 2014. 2

[7] Mohsen Fayyaz and Jurgen Gall. SCT: Set constrained temporal transformer for set supervised action segmentation. In *CVPR*, 2020. 3, 7

[8] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *European Conference on Computer Vision*, pages 137–153. Springer, 2016. 2, 7, 8

[9] H Jhuang, H Garrote, E Poggio, T Serre, and T Hmdb. A large video database for human motion recognition. In *Proc. of IEEE International Conference on Computer Vision*, volume 4, page 6, 2011. 3

[10] Oscar Koller, Hermann Ney, and Richard Bowden. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3793–3802, 2016. 2

[11] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4297–4305, 2017. 2

[12] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 6

[13] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016. 2

[14] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 2, 6, 7, 8

[15] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12066–12074, 2019. 2

[16] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017. 2

[17] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *European Conference on Computer Vision*, pages 36–52. Springer, 2016. 2

[18] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6742–6751, 2018. 2

[19] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2, 3, 7, 8

[20] Jun Li and Sinisa Todorovic. Set-constrained viterbi for set-supervised action segmentation. *arXiv preprint arXiv:2002.11925*, 2020. 1, 2, 3, 6, 7, 8

[21] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1298–1307, 2019. 3

[22] Ziyi Liu, Le Wang, Qilin Zhang, Zhanning Gao, Zhenxing Niu, Nanning Zheng, and Gang Hua. Weakly supervised temporal action localization through contrast based evaluation networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3899–3908, 2019. 3

[23] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5502–5511, 2019. 3

[24] Bingbing Ni, Vignesh R Paramathayalan, and Pierre Moulin. Multiple granularity analysis for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 756–763, 2014. 2

[25] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018. 3

[26] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 754–763, 2017. 2, 3, 7, 8

[27] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5987–5996, 2018. 1, 3, 4, 6, 7, 8

[28] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7386–7395, 2018. 2, 3, 7, 8

[29] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1194–1201. IEEE, 2012. 2, 6

[30] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 119(3):346–373, 2016. 6

[31] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8368–8376, 2018. 2

[32] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: weakly-supervised temporal action localization in untrimmed videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–171, 2018. 3

[33] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1961–1970, 2016. 2

[34] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3544–3553. IEEE, 2017. 3

[35] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 3

[36] Chen Sun, Sanketh Shetty, Rahul Sukthankar, and Ram Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 371–380. ACM, 2015. 3

[37] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1250–1257. IEEE, 2012. 2

[38] Nam N Vo and Aaron F Bobick. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2641–2648, 2014. 2

[39] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 6

[40] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4325–4334, 2017. 3

[41] Yunlu Xu, Chengwei Zhang, Zhanzhan Cheng, Jianwen Xie, Yi Niu, Shiliang Pu, and Fei Wu. Segregated temporal assembly recurrent networks for weakly supervised multiple action detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9070–9078, 2019. 3

[42] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2678–2687, 2016. 2

[43] Tan Yu, Zhou Ren, Yuncheng Li, Enxu Yan, Ning Xu, and Junsong Yuan. Temporal structure mining for weakly supervised action detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5522–5531, 2019. 3

[44] Yuanhao Zhai, Le Wang, Ziyi Liu, Qilin Zhang, Gang Hua, and Nanning Zheng. Action coherence network for weakly supervised temporal action localization. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3696–3700. IEEE, 2019. 3

[45] Xiao-Yu Zhang, Changsheng Li, Haichao Shi, Xiaobin Zhu, Peng Li, and Jing Dong. Adapnet: Adaptability decomposing encoder-decoder network for weakly supervised action recognition and localization. *IEEE Transactions on Neural Networks and Learning Systems*, 2020. 3

[46] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017. 2