Efficient and Effective Weakly-Supervised Action Segmentation via Action-Transition-Aware Boundary Alignment

Angchi Xu[1], Wei-Shi Zheng[1,2,3,*]
[1]School of Computer Science and Engineering, Sun Yat-sen University, China
[2]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
[3]Guangdong Province Key Laboratory of Information Security Technology, Guangzhou, China
xuangch@mail2.sysu.edu.cn, wszheng@ieee.org

Abstract

Weakly-supervised action segmentation is a task of learning to partition a long video into several action segments, where training videos are only accompanied by transcripts (ordered list of actions). Most of existing methods need to infer pseudo segmentation for training by serial alignment between all frames and the transcript, which is time-consuming and hard to be parallelized while training. In this work, we aim to escape from this inefficient alignment with massive but redundant frames, and instead to directly localize a few action transitions for pseudo segmentation generation, where a transition refers to the change from an action segment to its next adjacent one in the transcript. As the true transitions are submerged in noisy boundaries due to intra-segment visual variation, we propose a novel Action-Transition-Aware Boundary Alignment (ATBA) framework to efficiently and effectively filter out noisy boundaries and detect transitions. In addition, to boost the semantic learning in the case that noise is inevitably present in the pseudo segmentation, we also introduce video-level losses to utilize the trusted video-level supervision. Extensive experiments show the effectiveness of our approach on both performance and training speed.[1]

1. Introduction

Action segmentation aims to partition a long untrimmed video into several segments and classify each segment into an action category [8, 12, 13, 15, 16, 29, 32, 35, 38, 44]. It is an important yet challenging task for instructional or procedural video understanding. Although fully-supervised action segmentation (FSAS) methods [1, 6, 13, 14, 27] have achieved great progress, they require frame-wise dense annotation, which is labor-intensive and time-consuming to collect. As a result, many works [5, 9, 25, 28, 34, 38, 46] explore the weakly-supervised action segmentation (WSAS) only requiring the *transcript* annotation, which refers to the ordered list of actions occurring in the video without their start and end times. The transcripts are less costly to obtain and can be accessed directly from video narrations or other meta data [2, 9, 18, 28, 30, 33, 47].

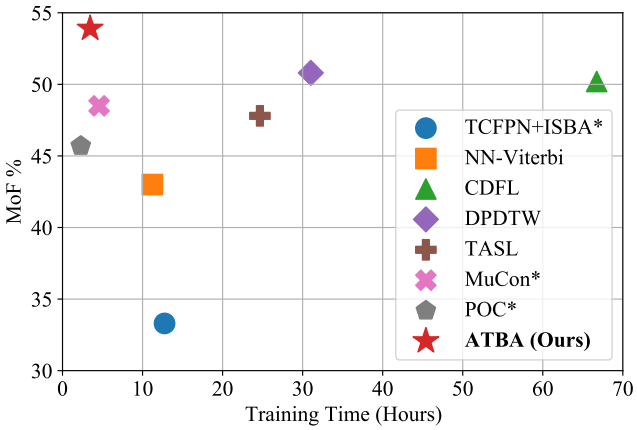

Figure 1. Comparison of performance and training time of WSAS methods on the Breakfast dataset. MoF-The main metric of the task, the higher the better. *-Alignment-free methods. Our ATBA achieves the best performance with a very short training time.

Most of previous WSAS methods have to infer the pseudo segmentation (pseudo frame-wise labels) for training via a sequence alignment process between the video and given transcript, such as Viterbi [22, 23, 25, 28, 33, 34] or Dynamic Time Warping (DTW) [4, 5]. These alignment algorithms are usually designed in a recursive form which needs to be performed serially frame-by-frame and hard to be parallelized, resulting in very slow training process.

In this work, we argue that the frame-by-frame alignment is NOT necessary, since the pseudo segmentation is

*Corresponding author.
[1]Code is available at https://github.com/iSEE-Laboratory/CVPR24_ATBA.

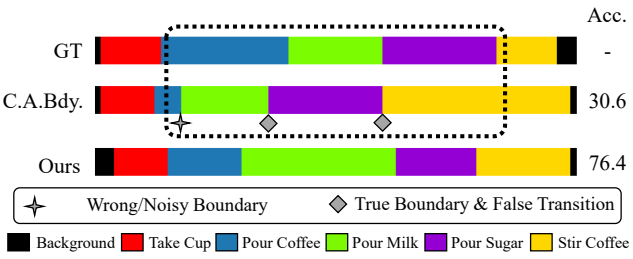

Figure 2. The necessity of proposed ATBA. The example is *P54-webcam01-P54-coffee* in Breakfast dataset. GT-The ground-truth segmentation. C.A.Bdy.-Only class-agnostic boundary detection is applied (Exp.1 of Table 3). Acc.-The accuracy of pseudo segmentation. In the video clip around the "star" point, the coffee pot undergoes a change from being picked up to tilted pouring within the segment "*Pour Coffee*", and this noisy visual change is incorrectly detected. In addition, although two boundaries are correctly detected by the "C.A.Bdy." (diamonds), they correspond to incorrect transitions due to one false positive error (star), resulting in complete dislocation of segments within the dashed box. Best viewed in color.

fundamentally determined by the locations of a *small* number of *action transitions* (*i.e.*, the change from an action segment to its *next adjacent* action segment in the transcript). Hence, the pseudo segmentation generation can be viewed as a transition detection problem, implying the way to more efficient designs. Intuitively, action transitions are often accompanied by significant visual changes, and there are already many approaches that can detect class-agnostic action *boundaries* based on these changes [12, 20]. However, due to the intra-segment visual variation and sub-segments under finer granularity, there are numerous *noisy* boundaries not corresponding to any transitions. Moreover, as the class-agnostic way cannot guarantee correct correspondence between the boundaries and transitions, even slight errors in the detection can result in severe deviation (Fig. 2).

To overcome the above noisy boundary issue, we propose an efficient and effective framework for WSAS, termed ***A**ction-**T**ransition-Aware **B**oundary **A**lignment* (ATBA), which directly detects the transitions for faster and effective pseudo segmentation generation. To tolerate the noisy boundaries, the ATBA generates *more* class-agnostic boundaries than the number of transitions as candidates, and then determines a subset from candidates that *optimally* matches all desired transitions via a drop-allowed alignment algorithm. Furthermore, to fortify the semantic learning under the inevitable noise in pseudo segmentation, we also introduce video-level losses to make use of the *trusted* video-level supervision. Our ATBA is efficient, because the number of generated candidates will be proportional to the length of the transcript, and therefore the complexity of alignment is now *independent* of the very long video length. Moreover, other computations required by ATBA, *i.e.*, measuring *how likely a frame is to be a boundary* and *how likely a candidate corresponds to a desired transition*, are both built on a convolution-like algorithm inspired by [20], which can be parallelized on GPUs efficiently.

For inference, we directly adopt the results from the trained frame-wise classifier, without the need for any alignment processing with retrieved or predicted transcript (the ground-truth transcript is not available during inference) like previous WSAS methods [4, 5, 25, 28, 34, 38], which also improves the inference efficiency.

In summary, our contributions are as follows. (1) We propose to directly localize the action transitions for efficient pseudo segmentation generation during training, without the need of time-consuming frame-by-frame alignment. (2) For robustness to noisy boundaries, we propose a novel ATBA framework to effectively determine boundaries corresponding to each transition. Video-level losses are also introduced to regularize the semantic learning involving the unavoidable noise in the pseudo segmentation. Experiments are conducted on three popular datasets to evaluate our approach: Breakfast [21], Hollywood Extended [2] and CrossTask [47]. Our ATBA achieves state-of-the-art or comparable results with one of the fastest training speed (Fig. 1), demonstrating the effectiveness of ours.

2. Related Work

Weakly-supervised action segmentation methods learn to partition a video into several action segments from training videos only annotated by transcripts [2, 4, 5, 9, 18, 22, 23, 25, 28, 33, 34, 38, 39, 46]. Despite different optimization objectives, most of them generate the pseudo segmentation for training by solving alignment objectives between two sequences (video and transcript) via Connectionist Temporal Classification (CTC) [18], Viterbi [22, 23, 25, 28, 33, 34] or Dynamic Time Warping (DTW) [4, 5].

Specifically, [18] proposes an extended version of CTC to evaluate all valid alignments between videos and transcripts, which additionally takes the visual similarities of frames into account. Inspired by speech recognition, [22, 23, 33] all use the Hidden Markov Model (HMM) to model the relationship between videos and actions. [9] always generates uniform segmentation but iteratively adjusts the boundaries by inserting repeating actions into the transcript. [34] proposes an alignment objective based on explicit context and length models, which can be solved by Viterbi, and the solved optimal alignment would serve as pseudo labels to train the frame-wise classifier. [25] and [28] both focus on novel learning objectives, but still require the optimal pseudo segmentation produced by Viterbi. [4, 5] both learn from the contrast of aligning the video to the ground-truth transcript and negative transcripts, where the alignment is performed by DTW. Whether using CTC, Viterbi or DTW,

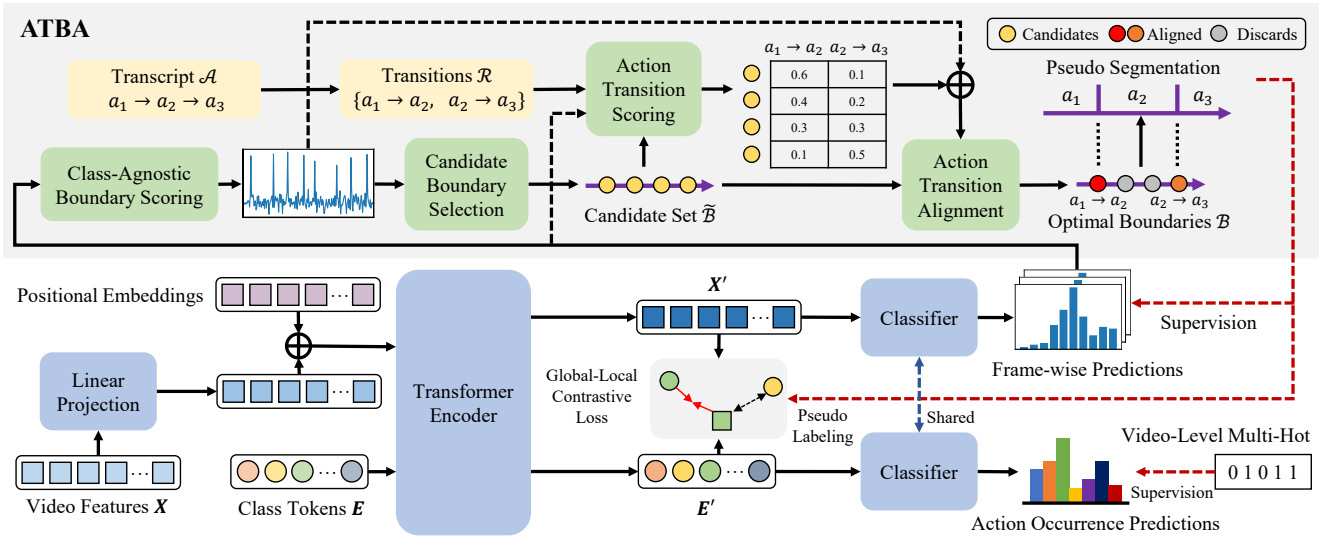

Figure 3. The overall framework. We propose an Action-Transition-Aware Boundary Alignment (ATBA) framework, which takes the class-agnostic boundary pattern and action transition pattern together into account to efficiently generate pseudo labels. The trusted video-level supervision is also utilized to further enhance the performance.

the above approaches except [9] require frame-by-frame serial calculation, which are inefficient.

Recently, some efficient methods are proposed with alignment-free design. [38] learns from the mutual consistency between two forms of a segmentation (*i.e.*, frame-wise classification and category/length pairs). [29] proposes a loss to enforce the output order of any two actions to be consistent with the transcript.[2] In this work, we also propose an efficient framework with different technical roadmap, and our performance is better at the same level of training speed.

3. Approach

3.1. Problem Statement

Action segmentation is a task of partitioning a video into several temporal segments with action labels, which is equivalent to predicting the action categories of *each* frame. Formally, given a sequence of T d-dimensional frame-wise features $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_T] \in \mathbb{R}^{T \times d}$ for a video with T frames, the goal is to predict a sequence of actions $\hat{\mathcal{Y}} = [\hat{y}_1, ..., \hat{y}_T]$, where $\hat{y}_t \in \mathcal{C}$ and $\mathcal{C} = \{1, 2, ..., |\mathcal{C}|\}$ is the set of action categories across the dataset (including the *background*). Under the setting of WSAS, the frame-wise ground-truth $\mathcal{Y} = [y_1, ..., y_T]$ is NOT available during training. Instead, the ordered list of actions called transcript $\mathcal{A} = [a_1, ..., a_M]$ is provided (including the background segments), where $a_m \in \mathcal{C}$ and M is the total number of action segments in the video. The action transitions of $\mathcal{A}$ are naturally formulated as $\mathcal{R} = \{(a_r, a_{r+1})\}_{r=1}^{M-1}$.

[2]POC [29] is a set-supervised method but can be extended to transcript supervision naturally, so we cite its corresponding results for comparison.

3.2. Overview

Our proposed framework is illustrated in Fig. 3. At first, the input sequence $\boldsymbol{X}$ is further encoded by a temporal network to generate more task-relevant representations $\boldsymbol{X}' \in \mathbb{R}^{T \times d'}$ (Sec. 3.3), then a classifier shared along the temporal axis followed by a category softmax activation will predict the frame-wise class probabilities $\boldsymbol{P} = [\boldsymbol{p}_1, ..., \boldsymbol{p}_T] \in \mathbb{R}^{T \times |\mathcal{C}|}$ from $\boldsymbol{X}'$. After that, the Action-Transition-Aware Boundary Alignment (ATBA) module takes $\boldsymbol{P}$ and the transcript $\mathcal{A}$ as input to infer the pseudo frame-wise labels $\widetilde{\mathcal{Y}} = [\widetilde{y}_1, ..., \widetilde{y}_T]$, where $\widetilde{y}_t \in \mathcal{C}$ (Sec. 3.4). Finally, $\widetilde{\mathcal{Y}}$ is used back to supervise $\boldsymbol{P}$ by a standard cross entropy:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{T} \sum_{t=1}^{T} \sum_{c=1}^{|\mathcal{C}|} \mathbb{I}(\widetilde{y}_t = c) \log \boldsymbol{P}_{t,c}, \tag{1}$$

where $\mathbb{I}(\cdot)$ is the indicator function which returns 1 if the condition is satisfied and 0 otherwise. Note that the pseudo frame-wise labels are inferred at each iteration for a batch of data and used immediately for current training at the same iteration. The additional video-level losses are stated in Sec. 3.5, and the training and inference processes are described in Sec. 3.6.

3.3. Temporal Network

We employ a slightly modified pre-norm [42] Transformer [40] encoder with learnable positional embeddings as the temporal network for feature learning. Following [11, 45],

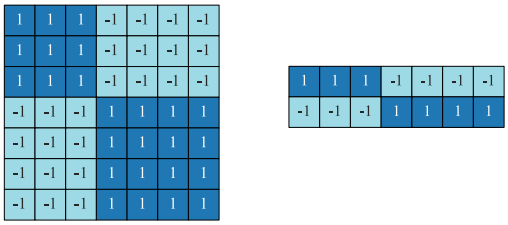

Figure 4. (a) A 7×7 template for class-agnostic boundary scoring. (b) A 2×7 template for action transition scoring.

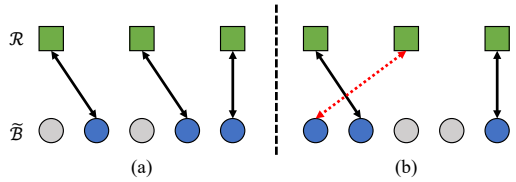

Figure 5. Illustration of the alignment between action transitions $\mathcal{R}$ and candidate boundaries $\widetilde{\mathcal{B}}$. Blue circles are aligned and gray ones are dropped. (a) A valid alignment. (b) An invalid alignment. The red dashed arrow violates the ordering consistency. Best viewed in color.

the vanilla *full* self-attention is replaced with a pyramid hierarchical *local* attention to better adapt to the action segmentation task (see supplemental material for more details).

3.4. Action-Transition-Aware Boundary Alignment

The ATBA is the core component of our framework. It generates the pseudo frame-wise labels $\widetilde{\mathcal{Y}}$ by inferring the boundaries corresponding to $M-1$ action transitions from the predicted class probabilities $\boldsymbol{P}$. Once they are found, $\widetilde{\mathcal{Y}}$ can be naturally generated by assigning the action labels of $\mathcal{A}$ one-by-one into the intervals between these boundaries.

Briefly, the ATBA first generates a set of candidate boundaries via a class-agnostic way, and then finds $M-1$ points from this set that optimally match all transitions of $\mathcal{A}$ via a dynamic programming (DP) algorithm. We describe the details in the following.

- Class-Agnostic Boundary Scoring. Firstly, the ATBA calculates the class-agnostic boundary scores $\mathcal{V}^{\mathrm{b}} = [v_1^{\mathrm{b}}, ..., v_T^{\mathrm{b}}]$ for *each* timestamp. We employ a pattern-matching-based scoring method proposed by a generic event boundary detection approach termed UBoCo [20], which considers the pattern of each frame's neighborhood. Specifically, for timestamp t, a pairwise similarity matrix $\boldsymbol{\Gamma}^{(t)} \in \mathbb{R}^{w^{\mathrm{b}} \times w^{\mathrm{b}}}$ is calculated within a local window with size w^{b} centered at t, whose (i,j)-entry represents the class-agnostic similarity between i-th and j-th frames inside the window, with values ranging from -1 to 1 (see the computational details in supplementary material). Clearly, if t is a boundary, the frame feature should change dramatically at t and keep stable elsewhere, so its $\boldsymbol{\Gamma}^{(t)}$ should show the pattern of that the values in the upper-left and lower-right areas are close to 1 and otherwise close to -1. Hence a template $\boldsymbol{\Omega}^{\mathrm{b}} \in \mathbb{R}^{w^{\mathrm{b}} \times w^{\mathrm{b}}}$ like Fig. 4(a) is designed to capture this pattern and output the class-agnostic boundary score for each t by a correlation operation:

$$v_t^{\mathrm{b}} = \frac{1}{w^{\mathrm{b}} \times w^{\mathrm{b}}} \sum_{i=1}^{w^{\mathrm{b}}} \sum_{j=1}^{w^{\mathrm{b}}} \boldsymbol{\Omega}_{i,j}^{\mathrm{b}} \boldsymbol{\Gamma}_{i,j}^{(t)}. \tag{2}$$

- Candidate Boundary Selection. After calculating $\mathcal{V}^{\mathrm{b}}$, we select a set of K candidate boundaries $\widetilde{\mathcal{B}} = \{b_k\}_{k=1}^{K}$, where $1 < b_1 < ... < b_K \leq T$ and $K > M-1$. The selection is performed by a simple greedy strategy with non maximum suppression inspired by [12], *i.e.*, each time we select one timestamp with current highest score v_t^{b}, and invalidate its neighborhood to avoid selecting multiple timestamps corresponding to one same boundary, until the number of selected timestamps reach an upper bound or all remaining timestamps are invalid. The radius of the invalid interval is set adaptively to $\frac{\mu T}{M}$, where $\mu \in [0,1]$ is a hyper-parameter, and in practice, the upper bound for K is set to $\lambda(M-1)$, where $\lambda \in \mathbb{N}_+$ is also a hyper-parameter.

- Action Transition Scoring. As mentioned in Sec. 1, the class-agnostic scores $\mathcal{V}^{\mathrm{b}}$ are not enough to detect action transitions. To this end, we then calculate an action transition score matrix $\boldsymbol{V}^{\mathrm{a}} \in \mathbb{R}^{K \times (M-1)}$, where $\boldsymbol{V}_{k,r}^{\mathrm{a}}$ measures the possibility that the k-th candidate boundary corresponds to the r-th transition, *i.e.*, separates the r-th and $(r+1)$-th action segments. This matrix is also calculated via pattern matching. Clearly, if candidate b_k corresponds to the r-th transition, the classifier's activation for class a_r should drop sharply after b_k, while rise for class a_{r+1}. Hence, a template $\boldsymbol{\Omega}^{\mathrm{a}} \in \mathbb{R}^{2 \times w^{\mathrm{a}}}$ with temporal size w^{a} like Fig. 4(b) is employed to detect this pattern around b_k:

$$\begin{aligned} \boldsymbol{V}_{k,r}^{\mathrm{a}} &= \frac{1}{2w^{\mathrm{a}}} \sum_{i=1}^{2} \sum_{j=1}^{w^{\mathrm{a}}} \boldsymbol{\Omega}_{i,j}^{\mathrm{a}} \boldsymbol{P}_{\mathrm{ind}^{\mathrm{a}}(b_k,j),\ a_{r+i-1}}, \\ \mathrm{ind}^{\mathrm{a}}(b_k, j) &= b_k - \lfloor \frac{w^{\mathrm{a}}}{2} \rfloor + j - 1, \end{aligned} \tag{3}$$

where $\mathrm{ind}^{\mathrm{a}}(b_k, j)$ is the index transform from the index j of the local window centered at b_k to the global timestamp index. Finally, the class-agnostic scores $\mathcal{V}^{\mathrm{b}}$ are added back to $\boldsymbol{V}^{\mathrm{a}}$ to produce the final score matrix $\boldsymbol{V} \in \mathbb{R}^{K \times (M-1)}$:

$$\boldsymbol{V}_{k,r} = \boldsymbol{V}_{k,r}^{\mathrm{a}} + v_{b_k}^{\mathrm{b}}. \tag{4}$$

The above equation means that the class-agnostic boundary score of the k-th boundary $v_{b_k}^{\mathrm{b}}$ is added to *all* transition scores corresponding to the k-th boundary $\boldsymbol{V}_{k,r}^{\mathrm{a}}, \forall r$.

- Action Transition Alignment. The final step is to find exact $M-1$ optimal boundaries from the candidates $\widetilde{\mathcal{B}}$ based

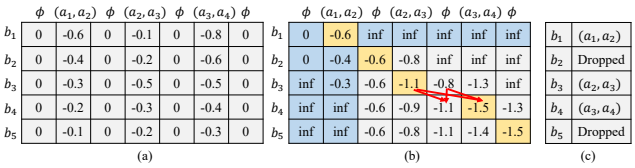

Figure 6. An example of action transition alignment between $\widetilde{\mathcal{B}}$ of length 5 and $\mathcal{R}$ of length 3. (a) The cost matrix $\boldsymbol{\Delta}$. (b) The cumulative cost matrix $\boldsymbol{D}$. Blue areas are directly initialized. Yellow areas correspond to optimal alignment. Red arrows are allowed path directions. (c) The optimal alignment. Best viewed in color.

on the score matrix $\boldsymbol{V}$. It is equivalent to seek an one-to-one alignment with lowest cost between the transitions $\mathcal{R}$ and candidates $\widetilde{\mathcal{B}}$ while requiring $K - M + 1$ candidates to be dropped (Fig. 5). Formally, if we denote the optimal aligned boundary set as $\mathcal{B} = \{b_{k_r}\}_{r=1}^{M-1}$, where b_{k_r} corresponds to the r-th transition and $1 \leq k_1 < ... < k_{M-1} \leq K$. It should satisfy:

$$\mathcal{B} = \arg\min_{\mathcal{B}'} \psi(\mathcal{B}'), \quad \psi(\mathcal{B}') = -\sum_{r=1}^{M-1} \boldsymbol{V}_{k'_r,r}, \tag{5}$$

where $\psi(\mathcal{B}')$ is the cost function of an alignment and $\mathcal{B}'$ is any feasible aligned boundary set such as $\mathcal{B}$.

To solve Eq. (5), we propose a DP algorithm, of which an example is shown in Fig. 6. To allow some candidates in $\widetilde{\mathcal{B}}$ to be dropped, inspired by [36], we first expand the transition sequence $\mathcal{R}$ by inserting the *empty* symbol ϕ interleaved with transition symbols, obtaining $\mathcal{R}' = [\phi, (a_1, a_2), \phi, (a_2, a_3), \phi, ..., \phi, (a_{M-1}, a_M), \phi]$. Now the candidate boundary matched to ϕ will be discarded. Then, a cost matrix $\boldsymbol{\Delta} \in \mathbb{R}^{K \times (2(M-1)+1)}$ is constructed (Fig. 6(a)), whose (k, r')-th entry is the cost of aligning the b_k with the r'-th symbol in $\mathcal{R}'$. For even number of r' (transition symbols), it is clear that $\boldsymbol{\Delta}_{k,r'} = -\boldsymbol{V}_{k,r'/2}$ from Eq. (5). Odd r' means dropping the candidate, and the cost is set to 0.

A valid alignment is represented by a path from the top-left to bottom-right of $\boldsymbol{\Delta}$ with constraint on directions. Specifically, the allowed start positions are $(1, 1)$ and $(1, 2)$, respectively corresponding to two cases where b_1 is dropped and b_1 is matched with the first transition. Similarly, there are two allowed end positions: $(K, 2(M-1))$ and $(K, 2(M-1)+1)$. Then for the position (k, r') on the path, if r' is odd, its previous position is either $(k-1, r')$ or $(k-1, r'-1)$. It means that if b_k is dropped, the b_{k-1} is also dropped or matched with the previous transition. On the other hand, the previous position is either $(k-1, r'-1)$ or $(k-1, r'-2)$ if r' is even, with the similar meaning. These allowed directions are indicated by red arrows in Fig. 6(b).

The DP algorithm generally finds the optimal solution by computing a cumulative cost matrix $\boldsymbol{D}$ with the same shape as $\boldsymbol{\Delta}$ (Fig. 6(b)), where $\boldsymbol{D}_{k,r'}$ represents the *minimum* cumulative cost of all valid paths *ending* at (k, r'). It is computed by the following recursive equation, considering paths *coming from* valid directions aforementioned:

$$\boldsymbol{D}_{k,r'} = \boldsymbol{\Delta}_{k,r'} + \begin{cases} \min(\boldsymbol{D}_{k-1,r'}, \boldsymbol{D}_{k-1,r'-1}), & r' \text{is odd}, \\ \min(\boldsymbol{D}_{k-1,r'-1}, \boldsymbol{D}_{k-1,r'-2}), & r' \text{is even}. \end{cases} \tag{6}$$

The above equation is not applicable to the first row and the first two columns of $\boldsymbol{D}$. They are directly initialized before the computation (blue areas of Fig. 6(b)). We set the cost of invalid positions to ∞ (inf) and others are copied from $\boldsymbol{\Delta}$.

The cumulative cost of the optimal alignment is $\min(\boldsymbol{D}_{K,2(M-1)}, \boldsymbol{D}_{K,2(M-1)+1})$. The complete alignment, *i.e.*, the optimal boundary set $\mathcal{B}$ can be obtained by starting from the corresponding position of the above `min` operation and backtracking (Fig. 6(b/c)). More details can be found in the supplementary material.

- Complexity. Clearly, the class-agnostic boundary scoring and action transition scoring can be implemented by the `unfold` operator and matrix operations, thus can be parallelized on GPUs. The action transition alignment must be performed serially with a time complexity of $O(KM)$ or $O(\lambda M^2)$. Compared with the previous alignment methods such as Viterbi ($O(T^2M)$ [25, 28, 34]) and DTW ($O(TM)$), it decreases from a function involving video length T to one of *only* transcript length M ($M << T$).

3.5. Video-Level Losses

Inevitably, the inferred pseudo segmentation contains some degree of noise, which can be unexpectedly fit by the network. To improve the semantic robustness, we propose to jointly train a video-level multi-label classification task that predicts *whether each action appears in the video*, which is *precisely* indicated by the transcript. Specifically, inspired by the common practice of Transformer [7, 10, 19, 24, 43], the input sequence $\boldsymbol{X}$ is augmented by a set of $|\mathcal{C}|$ learnable class tokens which are fed into the same network and responsible for predicting action occurrence. The outputs, denoted as $\{\boldsymbol{e}'_c\}_{c=1}^{|\mathcal{C}|}$, are then used to predict the action occurrence probability via $\xi_c = \sigma(\boldsymbol{w}_c^T \boldsymbol{e}'_c + \epsilon_c)$ for each category c, where $\boldsymbol{w}_c$ and ϵ_c are the weight vector and bias scalar for class c in the classifier *shared* with frame-wise classification. $\sigma(\cdot)$ is the sigmoid activation. These predictions are supervised by the binary cross entropy:

$$\mathcal{L}_{\text{vid}} = -\frac{1}{|\mathcal{C}|} \sum_{c=1}^{|\mathcal{C}|} [y_c^{\text{vid}} \log \xi_c + (1 - y_c^{\text{vid}}) \log(1 - \xi_c)], \tag{7}$$

where $y_c^{\text{vid}} \in \{0, 1\}$ is the binary label indicating whether action c appears in the video. Through the shared network and the interactions between tokens, the semantics learned from video-level training can benefit the main task.

Moreover, we treat the class tokens with global information as the *prototypes* of each action and attempt to align the

frame features to them for semantic consistency. Specifically, we first calculate the feature centroid of each category c that appears in the video by averaging the output frame features with the corresponding pseudo labels:

$$\overline{\boldsymbol{x}}_c = \frac{\sum_{t=1}^{T} \mathbb{I}(\widetilde{y}_t = c)\boldsymbol{x}'_t}{\sum_{t=1}^{T} \mathbb{I}(\widetilde{y}_t = c)}, \; c \in \text{Set}(\mathcal{A}), \tag{8}$$

where the $\text{Set}(\cdot)$ operator converts a list into an unordered set and removes duplicates. These centroids will move towards the corresponding $\boldsymbol{e}'_c$ under the guidance of a global-local contrastive loss in the form of InfoNCE [17, 31]:

$$\mathcal{L}_{\text{glc}} = -\frac{1}{|\text{Set}(\mathcal{A})|} \sum_{c \in \text{Set}(\mathcal{A})} \log \frac{\exp(\langle \overline{\boldsymbol{x}}_c \rangle^T \langle \boldsymbol{e}'_c \rangle / \tau)}{\sum_{c'=1}^{|\mathcal{C}|} \exp(\langle \overline{\boldsymbol{x}}_c \rangle^T \langle \boldsymbol{e}'_{c'} \rangle / \tau)}, \tag{9}$$

where $\langle \boldsymbol{x} \rangle = \boldsymbol{x}/\|\boldsymbol{x}\|_2$ is the l_2-normalization operator, and τ is the temperature hyper-parameter.

3.6. Training and Inference

- Training. Since the pseudo labeling always requires a relatively good initialization, we utilize a two-stage training strategy. The first stage only uses the reliable labels, *i.e.*, the video-level labels, to pretrain the network. Hence the loss function is $\mathcal{L}_{\text{I}} = \mathcal{L}_{\text{vid}}$. The loss for the second stage further includes the frame-wise losses on both classification and representation using pseudo labels: $\mathcal{L}_{\text{II}} = \alpha\mathcal{L}_{\text{vid}} + \beta\mathcal{L}_{\text{cls}} + \gamma\mathcal{L}_{\text{glc}}$, where α, β and γ are all hyper-parameters.

- Inference. For inference, we obtain the action label $\hat{y}_t$ for frame t directly from the frame-wise class probabilities: $\hat{y}_t = \arg\max_c \boldsymbol{P}_{t,c}$. Note that we do not require any alignment processing during inference, which is performed by previous WSAS methods for smoother predictions. However, since the ground-truth transcript is not available during inference, these methods either run alignment with every transcript from the training set [4, 5, 25, 28, 34], or predict the transcript via another model [38], which makes the pipeline more time-consuming and less practical.

4. Experiments

4.1. Experimental Setup

- Datasets. We perform experiments on three datasets. The **Breakfast** [21] dataset contains 1712 videos of breakfast cooking with 48 different actions. On average, each video has 6.8 segments and 7.3% frames are background. The **Hollywood Extended** [2] dataset contains 937 videos taken from movies with 16 categories of daily actions such as *walk* or *sit*. On average, each video has 5.9 segments and 60.9% frames are background. The **CrossTask** [47] dataset contains videos from 18 primary tasks. Following [28], 14 cooking-related tasks are selected, which have 2552 videos and 80 action categories. On average, each video has 14.4 segments and 74.8% frames are background. For Breakfast, we use the released 4 training/test splits and report the average. For Hollywood, we perform a 10-fold cross-validation. For CrossTask, we use the released training/test split. These evaluation protocols are consistent with previous methods.

- Metrics. To evaluate our method, we use 4 standard metrics, following [5, 9, 25, 28]. (1) The *Mean-over-Frames* (**MoF**) is the percentage of frames whose labels are correctly predicted. (2) The *Mean-over-Frames without Background* (**MoF-Bg**) is the MoF over non-background frames. It is more suitable than MoF for the datasets with high background rate such as Hollywood and CrossTask. (3) The *Intersection-over-Union* (**IoU**) is defined as $|I \cap I^*|/|I \cup I^*|$ while (4) the *Intersection-over-Detection* (**IoD**) is $|I \cap I^*|/|I|$, where I^* and I are the ground-truth (GT) segment and the predicted segment *with the same class*, respectively. For each GT segment, the highest IoU/IoD with *one* predicted segment are preserved, and the average of all GT segments is reported. Note that the definition of the IoU/IoD in [28] is different from other works [5, 9, 25], and will be indicated with special symbols when used.

- Input Features. As the input features $\boldsymbol{X}$ for Breakfast and Hollywood, we use the 2048-dimensional RGB+flow I3D features [3] adopted by MuCon [38] and most FSAS methods [13, 26, 27, 45], while some WSAS methods [5, 25, 28, 34] still use the iDT features [41]. Since recent studies [37, 38] have found that they perform worse with more advanced I3D features, we still report the performance with the iDT features for them following [38]. For CrossTask, the officially released 3200-dimensional features are adopted, but we do not perform dimension reduction via PCA as with [28]. For GPU memory efficiency, we perform a $10\times$ temporal downsampling for Breakfast and $5\times$ for Hollywood. During inference, the output is upsampled to match the original video/ground-truth length.

- Implementation Details. We adopt a 6-layer Transformer [40] with single-head self-attention as the temporal network, and the latent dimension d' is 256. For the ATBA, we set $w^{\text{b}} = 7$ and $\lambda = 4$ for all datasets. w^{a} is set to 31 for Breakfast/CrossTask and 23 for Hollywood. μ is set to 0.3 for Breakfast/Hollywood and 0.1 for CrossTask. The loss weights α, β are set to 1.0 and γ is 0.1. τ in Eq. (9) is set to 0.2. Besides, to alleviate the issue that the predictions on Hollywood/CrossTask are overly dominated by background, we lower the sample weights of *pseudo* background frames to 0.8 in Eq. (1) for these datasets.

- Multiple Runs. Due to the alternating nature of learning from weak supervision, there are often fluctuations in the results of WSAS methods [28, 38]. Hence, following [38], we report the average and standard deviation over 5 runs with different random seeds for better evaluation.

Breakfast				
Method	MoF±std	MoF-Bg±std	IoU±std	IoD±std
HMM+RNN [33]	33.3	-	-	-
[33]+Length [23]	36.7	-	-	-
TCFPN+ISBA [9]	38.4/36.4±1.0*	38.4	24.2	40.6
NN-Viterbi [34]	43.0/39.7±2.4*	-	-	-
D3TW [4]	45.7	-	-	-
CDFL [25]	50.2/48.1±2.5*	48.0	33.7	45.4
DPDTW [5]	50.8	-	35.6	45.1
TASL [28] ♠	47.8	-	35.2†	46.1†
MuCon [38] ♠	48.5±1.8	50.3*	40.9*	54.0*
POC [29] ♠	45.7	-	38.3†	-
AdaAct [46]	51.2	48.3	36.3	46.4
ATBA ♠ (Ours)	**53.9**±1.2	**54.4**±1.2	**41.1**±0.7 **39.5**±0.8†	**61.7**±1.1 **55.9**±1.0†
Hollywood Extended				
Method	MoF±std	MoF-Bg±std	IoU±std	IoD±std
HMM+RNN [33]	-	-	11.9	-
[33]+Length [23]	-	-	12.3	-
TCFPN+ISBA [9]	28.7	34.5	12.6	18.3
D3TW [4]	33.6	-	-	-
CDFL [25]	45.0	40.6	19.5	25.8
DPDTW [5]	**55.6**	25.6♯	**33.2**	43.3
TASL [28]	42.1♯	27.2♯	23.3†♯	33.0†♯
MuCon [38] ♠	-	**41.6**	13.9*	-
ATBA ♠ (Ours)	47.7±2.5	40.2±1.6	30.9±1.6 **28.5**±1.6†	**55.8**±0.8 **44.9**±0.5†
CrossTask				
Method	MoF±std	MoF-Bg±std	IoU±std	IoD±std
NN-Viterbi [34] ♠	26.5*	-	10.7†*	24.0†*
CDFL [25] ♠	31.9*	-	11.5†*	23.8†*
TASL [28] ♠	40.7	27.4♯	14.5†	**25.1**†
POC [29] ♠	42.8	17.6♯	15.6†	-
ATBA ♠ (Ours)	**50.6**±1.3	**31.3**±0.7	**20.9**±0.4 **15.7**±0.3†	**44.6**±0.7 24.6±0.4†

Table 1. Comparisons of ours with other WSAS methods on three datasets. std is the standard deviation of multiple runs (if any). †-The metric is computed by the definition of [28]. *-Results are reported by other works. Please refer to the supplementary material for detailed sources. ♯-Results are obtained by us via rerunning the open sources (The TASL [28] does not follow the common 10-fold evaluation protocol for Hollywood so we re-produce the results). ♠-The reported results are the average of multiple runs. Best results are in bold, second best are underlined.

4.2. Comparison with the State-of-the-Art

- Performance. In Table 1, we compare our proposed method with previous WSAS methods. Our ATBA achieves state-of-the-art (SOTA) by a clear margin (+2.7% MoF) on the Breakfast [21], demonstrating the effectiveness of focusing on action transitions. Moreover, comparing the standard deviation with other methods, it can be found that our method is also relatively stable.

Our ATBA also achieves comparable performance on the Hollywood [2]. The reason why our method does not show significant advantage is probably because this dataset is collected from movies and so contains many shot changes, resulting in more noisy boundaries. Note that although the DPDTW [5] achieves significantly high MoF, this metric is severely biased in case of highly imbalanced categories (60.9% frames are background). We report the MoF-Bg metric for it, on which it performs poorly, proving that it cannot recognize real actions very well. In contrast, the performance of our method is more balanced.

For the more difficult dataset CrossTask [47] with the most segments and highest background rate, our approach also outperforms previous methods whether or not the metric involves background (+7.8% MoF and +3.9% MoF-Bg).

Method	Tr.A.	MoF	Training (Hours)	Inference (Seconds)
TCFPN+ISBA [9]	✗	33.3	12.75*	**0.01***
NN-Viterbi [34]	V	43.0	11.23*	56.25*
CDFL [25]	V	50.2	66.73*	62.37*
DPDTW [5]	D	50.8	31.02♯	0.69♯
TASL [28]	V	47.8	24.66♯	54.99♯
MuCon [38]	✗	48.5	4.57	3.03
POC [29]	✗	45.7	**2.28**♯	**0.01**♯
ATBA (Ours)	B	**53.9**	3.45	**0.01**

Table 2. Comparison of accuracy, training and inference time on the Breakfast. The training time is measured as the entire training duration on the split 1, and the inference time is measured as the average time for inferring a video from the test set of the split 1. Tr.A.-The alignment algorithm adopted during training (✗-No Alignment. V-Viterbi. D-DTW. B-Boundary Alignment). *-Measured by [38]. ♯-Measured by us. Best results are in bold, second best are underlined.

- Efficiency. Besides of the performance, we also compare the training time of our approach with previous WSAS methods to show the efficiency of ours. Following [38], we train our model on an Nvidia GeForce GTX 1080Ti GPU, and the training time is measured as the wall time over the whole training phase, during which any irrelevant operations such as intermediate evaluation and saving checkpoints are removed. As all the WSAS methods directly load pre-computed features, the time measurement also does not include the time to extract features from raw videos. As shown in Table 2, our ATBA achieves the best performance with the second shortest training time, which demonstrates the effectiveness of our design. Specifically, the training speed of ours is on average 10 times (varying from 3 to 20) faster than methods performing frame-by-frame alignment [5, 25, 28, 34]. Compared to the alignment-free methods with comparable training speed [29, 38], ours achieves better performance.

We also compare the time to infer a single test video. As mentioned in Sec. 3.6, ours does not require any alignment processing, which makes it the fastest during inference.

Exp.	$\mathcal{V}^{\mathrm{b}}$	Cost Mat. $\boldsymbol{V}^{\mathrm{a}}$	Cost Mat. $\boldsymbol{V}$	P.L.	MoF	IoU	IoD
1	✓			38.6	32.3	23.1	52.5
2	✓	✓		51.7	41.3	30.2	47.8
3	✓	○	✓	**67.9**	**54.0**	**41.1**	**62.3**

Table 3. Ablation studies of ATBA on the Breakfast. Exp.-Different experiment configurations. Cost Mat.-Different choices for the cost matrix in action transition alignment. ○-According to Eq. (4), $\boldsymbol{V}$ contains $\boldsymbol{V}^{\mathrm{a}}$. P.L.-The accuracy of pseudo labels during training. Exp.1 means that only the class-agnostic boundary detection is adopted to localize action transitions. Exp.3 is our default setting. Best results are in bold.

Exp.	$\mathcal{L}_{\mathrm{vid}}$	$\mathcal{L}_{\mathrm{glc}}$	D.Cls.	P.L.	MoF	IoU	IoD
1				67.5	45.9	40.5	61.9
2	✓			66.6	52.4	38.8	61.4
3	✓	✓		67.9	**54.0**	**41.1**	**62.3**
4	✓	✓	✓	**68.1**	53.7	40.1	61.7

Table 4. Ablation studies of video-level losses on the Breakfast. Exp.-Different experiment configurations. D.Cls.-Different classifiers for frame-wise prediction and action occurrence prediction. P.L.-The accuracy of pseudo labels during training. Exp.3 is our default setting. Best results are in bold.

4.3. Ablation Studies

- Effect of ATBA. In Table 3, we conducted an in-depth evaluation of our proposed ATBA. From Exp.1, the pseudo label quality and the evaluation performance are both very poor when only using the class-agnostic boundary scores $\mathcal{V}^{\mathrm{b}}$ to directly select $M-1$ boundaries for transitions via the greedy strategy stated in Sec. 3.4, showing that it is critical to take the class-specific transition pattern into account as with our design to suppress the noisy boundaries for more precise pseudo segmentation. Fig. 2 provides an intuitive example. In addition, comparing Exp.2 and 3, it's better to involve the class-agnostic boundary scores in the action transition alignment. We think it is because that some candidates can have very low boundary scores (unlikely to be a boundary) as the candidate selection only depends on ranking, and when the transition patterns are not discriminative at the beginning of training, these candidates may be unexpectedly aligned if the boundary scores are not involved.

- Effect of Video-Level Losses. In Table 4, we evaluate the $\mathcal{L}_{\mathrm{vid}}$ and $\mathcal{L}_{\mathrm{glc}}$. Comparing Exp.1 and 2, the model with $\mathcal{L}_{\mathrm{vid}}$ achieves higher performance (+6.5% MoF) at the same level of pseudo label accuracy, demonstrating that the video-level supervision can promote the learning of precise action semantics. In Exp.3, the explicit representation alignment ($\mathcal{L}_{\mathrm{glc}}$) further improves the quality of frame features and thus the performance (+1.6% MoF). Moreover, Exp.4 shows that whether or not to share the classifier between frame-/video-level classification has little impact.

We also provide the analysis on the effect of important hyper-parameters in the supplementary material, including w^{b}, w^{a}, μ and λ in the ATBA.

4.4. Qualitative Results

We show the qualitative inference result compared with two recent open source WSAS methods, *i.e.*, MuCon [38] and TASL [28], in Fig. 7. Our ATBA achieves significantly more accurate result in this challenging video with many segments. The result of TASL shows order reversal ("*Crake Egg*" and "*Spoon Flour*") and hallucination ("*Butter Pan*"). The MuCon predicts the correct action ordering, but the result deviates severely. In contrast, ours successfully predicts an accurate segmentation, indicating the action semantics are well learned. More qualitative results are provided in supplementary material.

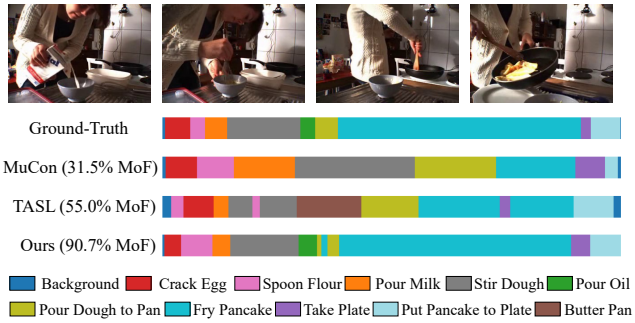

Figure 7. Qualitative results on the Breakfast. The example test video is *P15-stereo01-P15-pancake*. We compare our inference result with two recent methods. Best viewed in color.

5. Conclusion

In this work, we propose to directly localize action transitions for efficient pseudo segmentation generation in the WSAS task, thus avoiding the time-consuming frame-by-frame alignment. Due to the presence of noisy boundaries, a novel Action-Transition-Aware Boundary Alignment (ATBA) framework is proposed to efficiently and effectively filter out noise and detect transitions. Moreover, we also design some video-level losses to utilize video-level supervision to improve the semantic robustness. Extensive experiments show the effectiveness of our ATBA.

6. Acknowledgment

This work was supported partially by the NSFC (U21A20471), National Key Research and Development Program of China (2023YFA1008503), Guangdong NSF Project (No. 2023B1515040025, 2020B1515120085).

References

[1] Emad Bahrami, Gianpiero Francesca, and Juergen Gall. How much temporal long-term context is needed for action segmentation? *arXiv preprint arXiv:2308.11358*, 2023. 1

[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 *Proceedings of the European Conference on Computer Vision*, pages 628–643. Springer, 2014. 1, 2, 6, 7

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 6

[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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3555, 2019. 1, 2, 6, 7

[5] Xiaobin Chang, Frederick Tung, and Greg Mori. Learning discriminative prototypes with dynamic time warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8395–8404, 2021. 1, 2, 6, 7

[6] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9454–9463, 2020. 1

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5

[8] Guodong Ding and Angela Yao. Leveraging action affinity and continuity for semi-supervised temporal action segmentation. In *European Conference on Computer Vision*, pages 17–32. Springer, 2022. 1

[9] 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. 1, 2, 3, 6, 7

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5

[11] Dazhao Du, Bing Su, Yu Li, Zhongang Qi, Lingyu Si, and Ying Shan. Efficient u-transformer with boundary-aware loss for action segmentation. *arXiv preprint arXiv:2205.13425*, 2022. 3

[12] Zexing Du, Xue Wang, Guoqing Zhou, and Qing Wang. Fast and unsupervised action boundary detection for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2022. 1, 2, 4

[13] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3575–3584, 2019. 1, 6

[14] Shang-Hua Gao, Qi Han, Zhong-Yu Li, Pai Peng, Liang Wang, and Ming-Ming Cheng. Global2local: Efficient structure search for video action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16805–16814, 2021. 1

[15] Reza Ghoddoosian, Isht Dwivedi, Nakul Agarwal, Chiho Choi, and Behzad Dariush. Weakly-supervised online action segmentation in multi-view instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13780–13790, 2022. 1

[16] Reza Ghoddoosian, Isht Dwivedi, Nakul Agarwal, and Behzad Dariush. Weakly-supervised action segmentation and unseen error detection in anomalous instructional videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10128–10138, 2023. 1

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 6

[18] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *Proceedings of the European Conference on Computer Vision*, pages 137–153. Springer, 2016. 1, 2

[19] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *Advances in neural information processing systems*, 34: 18590–18602, 2021. 5

[20] Hyolim Kang, Jinwoo Kim, Taehyun Kim, and Seon Joo Kim. Uboco: Unsupervised boundary contrastive learning for generic event boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20073–20082, 2022. 2, 4

[21] 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. 2, 6, 7

[22] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 1, 2

[23] Hilde Kuehne, Alexander Richard, and Juergen Gall. A hybrid rnn-hmm approach for weakly supervised temporal action segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):765–779, 2018. 1, 2, 7

[24] Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. General multi-label image classification with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16478–16488, 2021. 5

[25] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceed-*

ings of the IEEE/CVF International Conference on Computer Vision, pages 6243–6251, 2019. 1, 2, 5, 6, 7

[26] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 6

[27] Daochang Liu, Qiyue Li, AnhDung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. *arXiv preprint arXiv:2303.17959*, 2023. 1, 6

[28] Zijia Lu and Ehsan Elhamifar. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8085–8095, 2021. 1, 2, 5, 6, 7, 8

[29] Zijia Lu and Ehsan Elhamifar. Set-supervised action learning in procedural task videos via pairwise order consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19903–19913, 2022. 1, 3, 7

[30] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 1

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 6

[32] Rahul Rahaman, Dipika Singhania, Alexandre Thiery, and Angela Yao. A generalized and robust framework for timestamp supervision in temporal action segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 279–296. Springer, 2022. 1

[33] 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. 1, 2, 7

[34] 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. 1, 2, 5, 6, 7

[35] Yuhan Shen and Ehsan Elhamifar. Semi-weakly-supervised learning of complex actions from instructional task videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3344–3354, 2022. 1

[36] Yuhan Shen, Lu Wang, and Ehsan Elhamifar. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2021. 5

[37] Yaser Souri, Alexander Richard, Luca Minciullo, and Juergen Gall. On evaluating weakly supervised action segmentation methods. *arXiv preprint arXiv:2005.09743*, 2020. 6

[38] Yaser Souri, Mohsen Fayyaz, Luca Minciullo, Gianpiero Francesca, and Juergen Gall. Fast weakly supervised action segmentation using mutual consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6196–6208, 2021. 1, 2, 3, 6, 7, 8

[39] Yaser Souri, Yazan Abu Farha, Fabien Despinoy, Gianpiero Francesca, and Juergen Gall. Fifa: Fast inference approximation for action segmentation. In *Proceedings of the DAGM German Conference on Pattern Recognition*, pages 282–296. Springer, 2022. 2

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 6

[41] 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

[42] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020. 3

[43] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4310–4319, 2022. 5

[44] Di Yang, Yaohui Wang, Antitza Dantcheva, Quan Kong, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. Lac-latent action composition for skeleton-based action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13679–13690, 2023. 1

[45] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. *arXiv preprint arXiv:2110.08568*, 2021. 3, 6

[46] Runzhong Zhang, Suchen Wang, Yueqi Duan, Yansong Tang, Yue Zhang, and Yap-Peng Tan. Hoi-aware adaptive network for weakly-supervised action segmentation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 1722–1730, 2023. 1, 2, 7

[47] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3537–3545, 2019. 1, 2, 6, 7