Intra- and Inter-Action Understanding via Temporal Action Parsing

Dian Shao Yue Zhao Bo Dai Dahua Lin
CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
{sd017, zy317, bdai, dhlin}@ie.cuhk.edu.hk

Abstract

Current methods for action recognition primarily rely on deep convolutional networks to derive feature embeddings of visual and motion features. While these methods have demonstrated remarkable performance on standard benchmarks, we are still in need of a better understanding as to how the videos, in particular their internal structures, relate to high-level semantics, which may lead to benefits in multiple aspects, e.g. *interpretable predictions and even new methods that can take the recognition performances to a next level. Towards this goal, we construct TAPOS, a new dataset developed on sport videos with manual annotations of sub-actions, and conduct a study on temporal action parsing on top* [1]*. Our study shows that a sport activity usually consists of multiple sub-actions and that the awareness of such temporal structures is beneficial to action recognition. We also investigate a number of temporal parsing methods, and thereon devise an improved method that is capable of mining sub-actions from training data without knowing the labels of them. On the constructed TAPOS, the proposed method is shown to reveal* ***intra****-action information,* i.e. *how action instances are made of sub-actions, and* ***inter****-action information,* i.e. *one specific sub-action may commonly appear in various actions.*

1. Introduction

Action understanding is a central topic in computer vision, which benefits a number of real-world applications, including video captioning [54], video retrieval [17, 41] and vision-based robotics [32]. Although over the past decade, remarkable progress has been made on action classification [56, 51, 48, 13], action localization [58, 31], and action segmentation [27, 23, 6], the insight into actions themselves remains lacking, as few works have analyzed actions at a finer-granularity, such as exploring their internal structures (intra-action understanding), discovering their mutual relationships (inter-action understanding).

One hindrance for intra- and inter-action understanding is a well-annotated dataset, which provides annotations that penetrate into actions, besides action labels as in most existing datasets [45, 24, 20, 21]. However, such a dataset is hard to collect, especially when labeling sub-actions. Specifically, action labels provided by humans are sometimes ambiguous and inconsistent, *e.g. open/close fridge* are treated as the same action while *pour milk/oil* belong to different actions. Such an issue could become severer when we deal with sub-actions, as compared to actions, sub-actions share more subtle differences between each other. Moreover, sub-actions belonging to the same category could not only appear in different instances of some action, but also instances of different actions. Although previous attempts have alleviated these issues by restricting the types of both actions and sub-actions within relatively more formatted cases, *e.g.* instructional [47, 33] and cooking [40, 22, 46, 10] videos. In more general cases, ensuring a consistent labeling scheme across sub-actions may be infeasible, considering the scale of a dataset.

Fortunately, we observe that humans are sensitive to *boundaries* of sub-actions, even without knowing their categories. We thus provide intra-action annotations in the form of high-quality temporal segmentations, instead of sub-action labels. The temporal segmentations divide actions into segments of different sub-actions, implicitly revealing actions' internal structures. The constructed dataset, which we refer to as **Temporal Action Parsing of Olympics Sports (TAPOS)**, contains over $16K$ action instances in 21 Olympics sport classes. We focus on Olympics sport actions as they have consistent and clean backgrounds, and diverse internal structures and rich sub-actions. These characteristics would encourage models to exploit the action themselves rather than the background scenes.

On top of TAPOS, we notice a temporal segmental network (TSN) [53] can obtain significant performance gains when the segments are aligned with temporal structures, instead of being evenly divided. Motivated by the study, we propose to investigate actions by temporally parse an action instance into segments, each of which covers a complete sub-action, where categories of these sub-actions are un-

[1] Project website: https://sdolivia.github.io/TAPOS/

known. *e.g*. parsing an instance of *triple jump* into six segments, whose semantics could be characterized as *run-up*, three *jumps*, and then a *reset*. While conceptually simple, temporal action parsing (TAP) is challenging in several aspects. Firstly, there are no pre-defined sub-action classes, and the associations among segments, *i.e*. which segments belong to the same class, are also unknown. Consequently, the possible number of distinct classes could be as large as $N * M$ ($> 30k$ in TAPOS), where N is the number of action instances and M is the average number of segments in an instance. This characteristic of TAP highlights its difference with tasks having pre-defined classes such as *Action Segmentation* [40, 46, 10, 14], since it is infeasible to turn TAP into these tasks by enumerating over possible class assignments. Moreover, compared to action boundaries, at the finer granularity of sub-actions, the transition between consecutive segments is often quite smoother, making it difficult to localize their boundaries.

We further develop an improved framework for temporal action parsing on TAPOS, inspired by recently proposed Transformer [50]. The proposed framework, referred to as TransParser, adopts two stacked transformer as its core, where frames of action instances are used as the queries, and parameters in a memory bank are served as keys and values. While TransParser outperforms baselines on temporal action parsing, its structure also enables it to discover semantic similarities of sub-action segments within one action class and across different action classes, in an unsupervised way. By investigating TransParser, we could also reveal additional intra-action information (*e.g*. which sub-action is the most discriminative one for some action class) and inter-action information (*e.g*. which sub-action commonly appears in different action classes).

The contribution of this work can be briefly summarized into three aspects: 1) a new dataset TAPOS which provides a class label for each action instance as well as its temporal structure; 2) a new task, namely Temporal Action Parsing, that encourages the exploration of the internal structures of actions; 3) an improved framework for temporal action parsing, which provides additional abilities for further intra- and inter-action understanding.

2. Related Work

Datasets. Being an important task in computer vision, various datasets have been collected for action understanding, which could be roughly divided into three categories. Datasets in the first category provides only class labels, including early attempts (*e.g*. KTH [25], Weizmann [3], UCFSports [39], Olympic [34]) of limited scale and diversity, and succeeding benchmarks (*e.g*. UCF101 [45], HMDB51 [24], Sports1M [20], and Kinetics [21]) that better fit the need of deep learning methods. However, despite of increasing numbers of action instances being covered, more sophisticated annotations are not provided by these datasets. In the second category, datasets provide boundaries of actions in untrimmed videos. Specifically, videos in THUMOS'15 [15] contain action instances of 20 sport classes. And daily activities are included in ActivityNet [7] and Charades [43]. Other datasets in this category include HACS [57] and AVA [16]. Although these datasets are all annotated with temporal boundaries, they focus on the location of an action in an untrimmed video. Instead, we intend to provide boundaries inside action instances, revealing their internal structures.

Our proposed dataset belongs to the third category, where fine-grained annotations for action instances are provided. Most of the existing datasets in this category focus on instructional videos, such as cooking videos in 50 Salads [46], Breakfast [22], and MPIICooking [40], as well as surgical videos in JIGSAWS [14]. Compared to these datasets, TAPOS mainly focuses on instances of Olympics sport actions for two reasons. First, Olympics actions have rich sub-actions, and loosely formatted but diverse internal structures, so that models are encouraged to exploit inside actions in a data-driven way. Moreover, instances of the same Olympics action have consistent and clean backgrounds, making models focus on the action itself.

Tasks. Various methods have been proposed for vision-based action recognition [56, 42, 51, 35, 44, 48, 13, 4, 49, 11, 5, 8, 30], where they are asked to predict a single class label for a given video instance. Temporal action localization [55, 58, 31], on the other hand, aims at identifying temporal locations of action instances in an untrimmed video. Another line of research focuses on a detailed understanding of the internal structures of action instances, especially along the temporal dimension. Specifically, in the task of action recognition, some researchers [36, 52] implicitly learn the temporal structures of complex activities to promote performances, but the quality of estimated temporal structures is not assessed. In temporal action parsing (TAP), we emphasize the importance of such temporal structures, and provide annotations for quality assessment. The most related task to TAP is temporal action segmentation (TAS) [26, 22, 1, 28, 9, 38, 29]. TAS aims at labeling each frame of an action instance within a set of pre-defined sub-actions, which can be done in a fully-supervised [9, 27] (*e.g*. frame labels are provided) or a weakly-supervised [29, 38, 37] (*e.g*. only ordered sub-actions are provided) manner. While TAS relies on a pre-defined set of sub-actions, assuming all samples contain only these classes, TAP offers only the boundaries between sub-actions, which are significantly weaker supervisions. We empirically found in our experiments that methods for TAS cannot well estimate the temporal structures under the setting of TAP, indicating TAP poses new challenges.

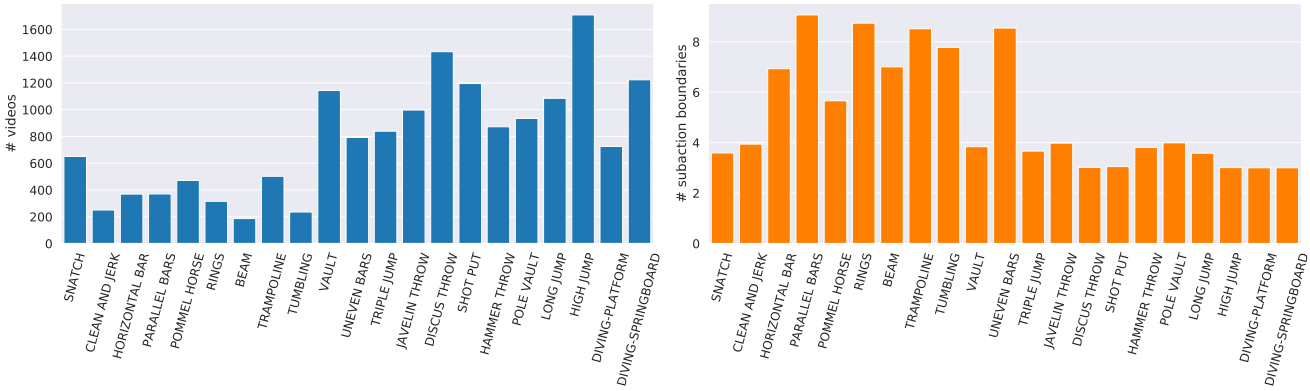

Figure 1: Statistics of the dataset: the left histogram depicts the average number of timestamps per class; the right one illustrates the average number of annotations for sub-action boundaries.

3. Dataset

To encourage intra- and inter-action understanding, we construct a new dataset, referred to as Temporal Action Parsing of Olympics Sports (TAPOS). Specifically, samples in TAPOS are all action instances of Olympics sports, so that instances belonging to the same sport tend to have a consistent background. Moreover, samples in TAPOS are ensured to cover a complete action instance with no shot changes. These two characteristics of TAPOS make it a suitable dataset for models that focus on the action itself, as potential distracters are explicitly avoided. For each sample, we provide annotations of two levels of granularity, namely the action labels (*e.g*. *triple-jump*, *shot-put*, etc), and the ranges of sub-actions (*e.g*. *run-up*, *jump* and *landing* in *triple-jump*), in the form of temporal timesteps. It's worth noting that labels of sub-actions are not provided. While sub-actions such as *run-up* could be further decomposed into stages at a finer granularity, in this paper we restrict our annotations to have only two-level granularities, leaving finer annotations as future work. We start by introducing the collection process, followed by dataset statistics and dataset analysis.

3.1. Dataset Collection

To obtain samples in TAPOS, we at first collect a set of videos from public resources (*e.g*. Youtube). Each collected video will be divided into a set of shots utilizing techniques for shot detection [2] to obtain instances within a single shot. For action labels and ranges of sub-actions, we apply a two-round annotation process using crowdsourcing services. In the first round, irrelevant shots and shots containing incomplete action instances are filtered out, and remaining shots are labeled with action classes. Subsequently, every remaining shot will be assigned to three annotators, so that we could cross-validate the annotations. Each annotator will mark the boundaries of consecutive sub-actions independently. Before the second round, we will provide annotators with instructional descriptions and illustrative samples, guiding them to divide shots without knowing sub-action labels. Finally, we filter out shots with a number of temporal timestamps less than 3.

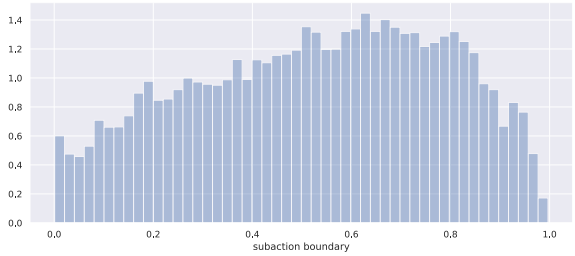

Figure 2: Probability distribution of sub-action boundary occurrence across the video. Length of each video is normed to one.

3.2. Dataset Statistics

TAPOS contains $16,294$ valid instances in total, across 21 action classes. These instances have a duration of 9.4 seconds on average. The number of instances within each class is different, where the largest class *high jump* has over $1,600$ instances, and the smallest class *beam* has 200 instances. The average number of sub-actions also varies from class to class, where *parallel bars* has 9 sub-actions on average, and *long jump* has 3 sub-actions on average, as shown in Figure 1. Finally, as Figure 2 shows, start and end points of sub-actions could temporally be any way for a single instance. While the number of instances within each class reflects the natural distribution of action classes, the variance in instances including their time durations, numbers of sub-actions in them and locations of sub-actions has reflected the natural diversity of actions' inner structures, facilitating more sophisticated investigations on actions.

All instances are split into train, validation and test sets, of sizes 13094, 1790, and 1763, respectively. When splitting instances, we ensure that instances belonging to the same video will appear only in one split.

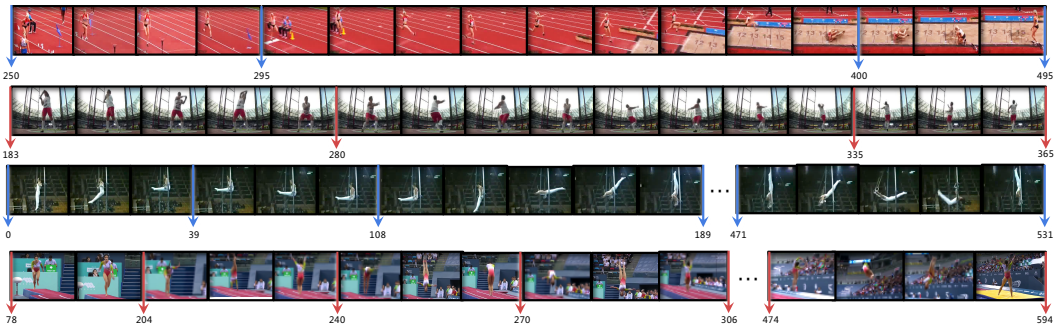

Figure 3: Some samples from the proposed TAPOS dataset. From top to bottom, action classes are: triple jump, hammer throw, rings and tumbling. Temporal boundaries of sub-actions are annotated, complex actions are therefore composed of temporally adjacent subactions, *e.g.*, hammer throw in the second row comprises *swing*, *rotate boby* and *throw*.

TSN	RGB		RGB + Flow	
sample	Top-1 Acc.	Avg. Acc.	Top-1 Acc.	Avg. Acc.
uniform	83.97	82.22	91.01	88.15
aligned	88.83	86.22	93.80	91.73

Table 1: Comparison of performance on action classification using different sampling for TSN. Both the top-1 accuracy and overall accuracy is reported.

Figure 4: Similar sub-actions are shared by irrelevant actions, *e.g.*, jump in *beam* and *triple jump* (the first pair), somersault in *uneven bars* and *diving* (the second pair).

4. Analysis of Sub-actions

The success of feature-based methods, *i.e.* those that directly connect visual feature embeddings to action classes, leads to a question: *"do we need to take a step further into the temporal structures?"*. In this section we present a brief study for this question.

Indeed, the utility of temporal structures have been demonstrated, sometimes in an implicit way, in previous works. Specifically, Wang *et al.* [53] found that dividing a video into segments helps action recognition. However, this work only considers even segmentation that is not necessarily aligned with the inherent structure. Feichtenhofer *et al.* [12] observed that certain patterns emerge automatically from the internal activations of a model for action classification. These discoveries indicate that each class of actions often contain temporal structures in certain ways. This corroborates with our intuition that an action is often composed of stages at finer granularity, *e.g.* the entire process of a *long jump* consists of a *run-up*, a *jump*, and a *landing*. We refer to segments of an action as *sub-actions*.

Next, we further investigate how the decomposition of an action into sub-actions influence action understanding. In the first study, we compared temporal segmental networks [53] on TAPOS, with two configurations: (1) with segments of even durations, and (2) with segments aligned with annotated sub-actions. Table 1 shows that the latter configuration outperforms the former by a large margin, which implies that the use of temporal structures, in particular the segmentation of sub-actions, can significantly help the performance of action recognition. In the second study, we carefully examine the connections between sub-actions in different action classes. As shown in Figure 4, sub-actions in different action classes can be similar, even for those actions that appear to be quite different in the first place. These findings suggest that to effectively discriminate between such classes, one may need to go beyond a local scope and resort to a more global view, *e.g.* looking into how sub-actions evolve from one to another.

5. Temporal Action Parsing

In this section, we first briefly introduce the setting of temporal action parsing, and then discuss our framework.

5.1. Task Definition

Formally, let $A = \{v_1, ..., v_n\}$ denotes a certain action instance of n frames, and $S_1, ..., S_k$ be its corresponding sub-actions so that $A = \{S_1, ..., S_k\}$, where $S_i = \{v_{t_i}, ..., v_{t_{i+1}-1}\}$. A can then be represented by a set of middle-level representations, each derived from one of its sub-action. The goal of an action parsing model is thus to

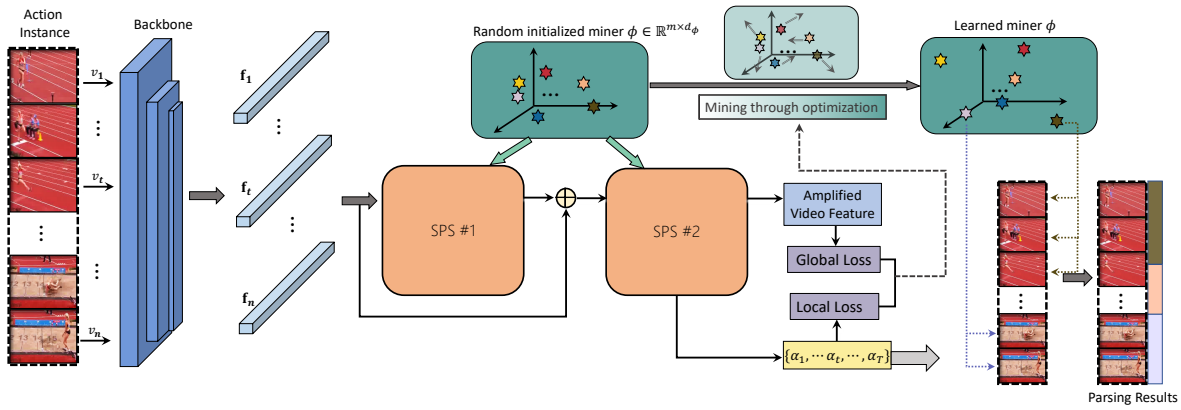

Figure 5: An overview of the proposed TransParser. Given a sequence of video frames, we first obtain the frame-level feature $\{\mathbf{f}_1, \cdots, \mathbf{f}_n\}$. Lying in the core of the TransParser are two stacked Soft-Pattern-Strengthen (SPS) Units which maintain a pattern miner ϕ and use a soft-attention mechanism to produce amplified feature from $\mathbf{f}_t$. We use two losses, *i.e.* a local loss to promote agreement between frames within a sub-action while suppressing that across sub-actions and a global loss to predict action label as a regularization. Throughout optimization, different representations for different sub-actions are automatically learned in the pattern miner. During inference, the calculated attention weight at the last SPS Unit can be used to obtain the temporal action parsing results.

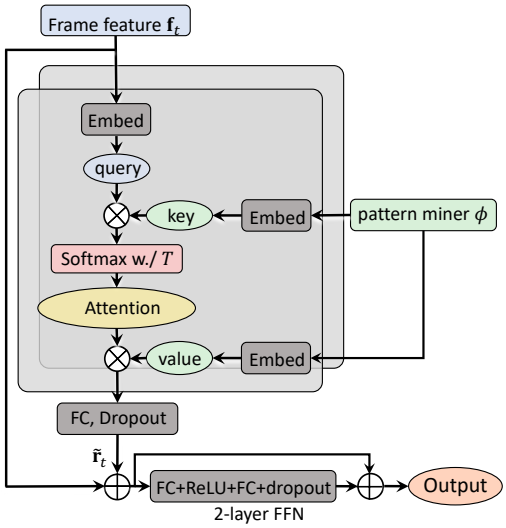

Figure 6: An instantiation of SPS Unit. It maintains a pattern miner ϕ and takes frame feature $\mathbf{f}_t$ as input. A multi-head soft-attention is conducted thereon. The final output $\tilde{\mathbf{r}}_t$ is added to $\mathbf{f}_t$ to achieve amplification.

identify the starting frames $\{v_{t_1}, ..., v_{t_k}\}$ of sub-actions.

5.2. TransParser for Temporal Action Parsing

To decompose an action instance into a set of sub-actions without knowing the possible sub-action categories, we develop a data-driven way to discover the distinct patterns of different sub-actions, as shown in Fig. 5. Specifically, given frames of an action instance $\{v_1, ..., v_n\}$, we at first apply a BNInception network [19] to extract per-frame features $\{\mathbf{f}_1, ..., \mathbf{f}_n; \mathbf{f}_t \in \mathbb{R}^{1 \times d_f}\}$. Each feature $\mathbf{f}_t$ is then refined by a Soft-Pattern-Strengthen (SPS) unit. The SPS unit maintains a parametric pattern miner $\boldsymbol{\phi}$ to learn distinct characteristics of sub-actions, which could be used to regularize the input feature, amplifying its discriminative patterns. The refinement could be described as $\mathbf{f}'_t = \mathbf{f}_t + \mathrm{SPS}(\mathbf{f}_t, \boldsymbol{\phi})$.

The parsing process of TransParser works as follows. Given refined features of action frames $\{\mathbf{f}'_1, ..., \mathbf{f}'_n\}$, we at first compute the responses $\{\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_n\}$ of them and the patterns stored in $\boldsymbol{\phi}$. The representative pattern (*i.e.* $\mathrm{argmax}_j\, \boldsymbol{\alpha}_t|_j$) in each response is selected thereafter, and once two consecutive frames $(t, t+1)$ have different representatives, TransParser marks the start of a new sub-action at $(t+1)$-th frame.

Soft-Pattern-Strengthen (SPS) Unit. Some components of the SPS unit are inspired by Transformer [50], which has been actively studied in the language domain but rarely explored in the field of action understanding. Being designed to amplify the discriminative patterns in the input feature $\mathbf{f}_t$, the SPS unit will maintain a pattern miner, parameterized as $\boldsymbol{\phi} = [\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_m]$ of size $m \times d_\phi$, to discover informative patterns of frame features during training. Such a process is conducted by a soft-attention that treats the input feature $\mathbf{f}_t$ and the miner $\boldsymbol{\phi}$ respectively as query and (key, value) pairs:

$$\boldsymbol{\alpha}_t = \mathrm{softmax}((\mathbf{f}_t \cdot \mathbf{W}_Q) \cdot (\boldsymbol{\phi} \cdot \mathbf{W}_K)^T), \quad (1)$$

$$\mathbf{r}_t = \boldsymbol{\alpha}_t \cdot (\boldsymbol{\phi} \cdot \mathbf{W}_V), \quad (2)$$

where $\cdot$ stands for matrix multiplication, and the output $\mathbf{r}_t$ can be intuitively regarded as a residual of $\mathbf{f}_t$ helpful to distinguish subtle differences. Following [50] in practice we use multi-head attention consisting of two groups of $\{\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V\}$, and the final $\tilde{\mathbf{r}}_t$ is obtained by feeding the two features $\mathbf{r}_t^{(1)}$ and $\mathbf{r}_t^{(2)}$ from each group respectively into a single fc layer. The output feature $\mathbf{f}'_t$ is computed by

$\mathrm{FFN}(\mathbf{f}_t + \widetilde{\mathbf{r}}_t)$, where FFN stands for a small feed-forward net as in [50]. Amplification is achieved via adding $\widetilde{\mathbf{r}}_t$ to $\mathbf{f}_t$.

We utilize the combination of two losses to learn the TransParser, with the ground-truths of temporal segmentations and labels of action instances. 1) *local loss*: to help the pattern miner ϕ capture informative patterns in features of action frames, a semantic loss is applied to maximize the agreement between frames within a sub-action while suppressing that across sub-actions:

$$\mathcal{L}_{\text{local}} = \frac{\mathcal{L}_{\text{sim}} + \lambda}{\mathcal{L}_{\text{dissim}}}, \tag{3}$$

$$\mathcal{L}_{\text{sim}} = \text{avg}(\sum_{t_1,t_2 \in S_i \forall i} ||\boldsymbol{\alpha}'_{t_1} - \boldsymbol{\alpha}'_{t_2}||_2), \tag{4}$$

$$\mathcal{L}_{\text{dissim}} = \text{avg}(\sum_{t_1 \in S_i, t_2 \in S_j, i \neq j} ||\boldsymbol{\alpha}'_{t_1} - \boldsymbol{\alpha}'_{t_2}||_2), \tag{5}$$

where $\boldsymbol{\alpha}'_t$ is the response computed on $\mathbf{f}'_t$ according to Eq.(1), as the refined feature $\mathbf{f}'_t$ contains amplified discriminative patterns compared to $\mathbf{f}_t$. λ is a regularizer to prevent trivial solutions (*e.g.* all $\boldsymbol{\alpha}$s are collapsed to be the same one hot vector). 2) *global loss*: We further add a global classification loss as a regularization, suggesting that refined features of action frames still need to be representative to action categories. For each action instance A with n frames:

$$\mathcal{L}_{\text{global}} = \text{NLL}(\frac{1}{n}\sum_{t=1}^{n}(\mathbf{W} \cdot \mathbf{f}'_t), l_A), \tag{6}$$

where $\mathbf{W}$ is the weight of a classifier, and l_A is the label. For conciseness, in practice we apply a second SPS unit to obtain $\boldsymbol{\alpha}'_t$ for the local loss. Succeedingly, we also use $\mathbf{f}''_t$ from the second unit in the global loss, which is more discriminative than $\mathbf{f}'_t$.

6. Experiments

6.1. Evaluation Metrics

Once the parsing process is finished, we can obtain a series of predicted start frames denoted by $\mathcal{T}_{\mathcal{P}} = \{s_1, s_2, \cdots, s_{|\mathcal{P}|}\}$. Assuming the ground-truth sub-actions start at $\mathcal{T}_{\mathcal{G}} = \{t_1, t_2, \cdots, t_{|\mathcal{G}|}\}$, we can determine the correctness of each prediction if its distance from the nearest ground-truth is smaller than a certain threshold d. d can either be in absolute frame number Δt or relative percentage $\Delta t/T$. The number of correct predictions is written as $|\mathcal{T}_{\mathcal{P}} \widetilde{\cap}_d \mathcal{T}_{\mathcal{G}}|$, where $\widetilde{\cap}_d$ can be regarded as an operation of intersection with respect to certain tolerance. We report the recall, precision, and F1 score, defined by:

$$\text{Recall@}d = \frac{|\mathcal{T}_{\mathcal{P}} \widetilde{\cap}_d \mathcal{T}_{\mathcal{G}}|}{|\mathcal{T}_{\mathcal{G}}|}, \qquad \text{Prec@}d = \frac{|\mathcal{T}_{\mathcal{P}} \widetilde{\cap}_d \mathcal{T}_{\mathcal{G}}|}{|\mathcal{T}_{\mathcal{P}}|}, \tag{7}$$

$$\text{F1@}d = \frac{2 \times \text{Recall@}d \times \text{Prec@}d}{\text{Recall@}d + \text{Prec@}d} \tag{8}$$

6.2. Baseline methods

Due to the connections between temporal action parsing and other tasks, such as temporal action segmentation [6, 18, 26] and action detection [31], we select representative methods from these tasks and adapt them to temporal action parsing for comparison with several modifications.

Action boundary detection. We resort to a sequence model, temporal convolution network (TCN) [26, 31] particularly, to estimate the emerge of action state changes. Given a snippet of T frames, a two-layer temporal convolution network is constructed on top to densely predict a scalar for every frame. Following [31], the annotated temporal boundary along with its k neighboring frames is labeled as 1 and the rest are set to be 0. The network is optimized using a weighted Binary Cross Entropy loss due to the imbalance between positive (*i.e.* sub-action change point) and negative samples. During inference, the sub-action is detected once the output is over a certain threshold θ_c, *e.g.* 0.5.

Weakly-supervised temporal action segmentation. Temporal action segmentation aims at labeling each frame of an action instance with a set of pre-defined sub-actions. In the weakly-supervised setting, only a list of sub-actions in the order of occurrence without precise temporal location is provided. We select two representative methods via Iterative Soft Boundary Assignment (ISBA) [6] and Connectionist Temporal Modeling (CTM) [18]. For ISBA, we generate pseudo-labels by extracting frame-level features $\{\mathbf{f}_i\}_{i=1}^{N}$ and pre-grouping them into K clusters. For CTM, the original training objective is to maximize the log likelihoods of the *pre-defined* target labeling. In our case, the loss is changed to the sum of log likelihoods for all *possible* labelings, in that all k distinctive randomly sampled sub-actions could be a possible solution. During inference, we use the simple best path decoding, *i.e.* concatenating the most active outputs at every timestamp.

6.3. Quantitative results

Parsing results of different methods. We use the aforementioned three metrics, namely Recall@d, Prec@d, and F1@d to evaluate the parsing performance of different methods. We vary the relative distance from 0.05 to 0.5 at a step of 0.05 and the absolute frame distance from 5 to 50 at a step of 5. The average F1-score across different distance thresholds is reported in Table 2. We can see that our method outperforms all baseline methods by a large margin. We also calculate the overlap of the second- and third-round annotation against the first-round to be the *human* performance. We see that 1) high consistency exists between human annotators; 2) there is still a huge gap compared with human performance, leaving great room for improvement.

A detailed comparison of F1 score, precision and recall are shown in Figure 7. As we can see, (1) Both TCN and CTM methods have exceedingly high recall but low

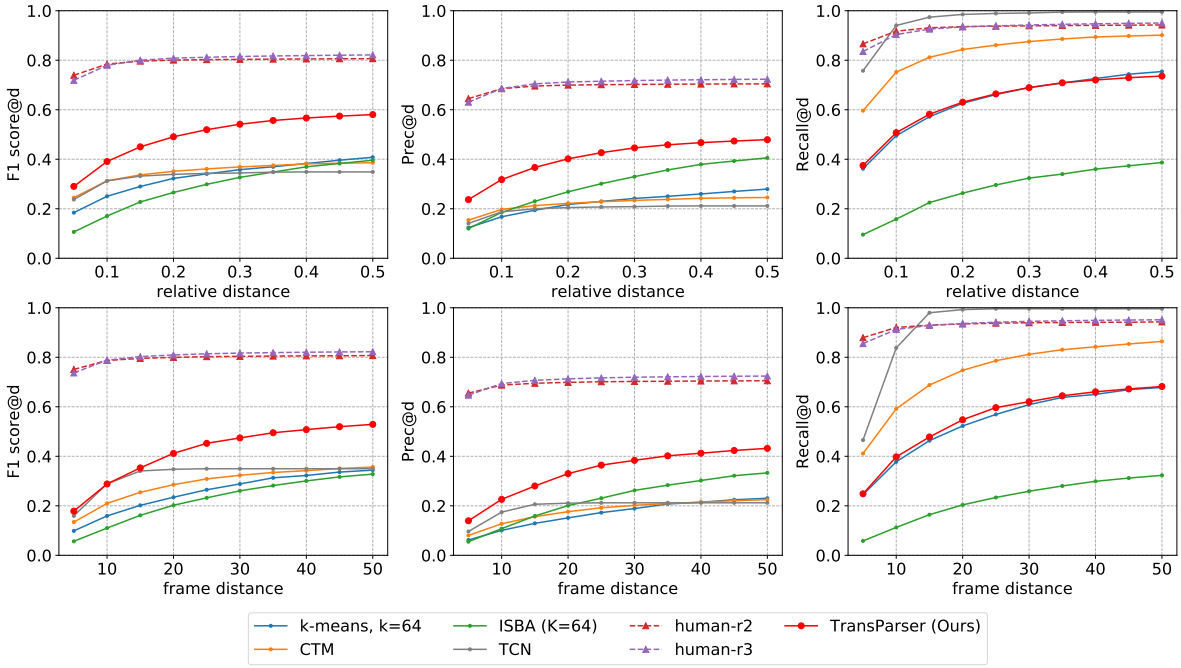

Figure 7: Comparison of different methods as well as human performance in terms of F1-score, precision and recall at different tolerance levels of relative distance and frame distance.

	avg. F1-score (rel.)	avg. F1-score (abs.)
k-means ($k = 64$)	0.3302	0.2881
ISBA ($k = 64$)	0.2892	0.2604
CTM	0.3502	0.3194
TCN	0.3303	0.3483
TransParser	**0.4745**	**0.3981**
Human (r2)	0.7948	0.8031
Human (r3)	0.8012	0.8158

Table 2: Temporal action parsing results on the proposed TAPOS dataset measured by average F1-score.

# of SPS	local loss	avg. F1	avg. Recall	avg. Precision
$\times 1$		0.2897	0.6950	0.1832
$\times 1$	✓	0.3996	0.5354	0.3189
$\times 2$	✓	0.4210	0.5548	0.3393

Table 3: Temporal action parsing results of TransParser under different settings. The average F1, recall and precision are calculated across $d \in \{5, 10, \cdots, 50\}$.

precision, showing that these methods suffer from severe over-parsing, indicating that they focus too much on local difference; However, ISBA performs poorly on recalls but yield higher precision than CTM and TCN, indicating that such a coarse-to-fine manner may be trapped and cannot exploit intra-action information. (2) The performance of our method consistently increases when relaxing the distance threshold, while the baseline methods quickly saturate.

Variants of the proposed TransParser. In this part, we validate the effectiveness of the designs behind TransParser. The results are summarized in Table 3. If the local loss is dropped, we observe an increase of recall at the cost of a significant decrease in precision. This reveals the crucial role of local semantic loss to encourage consistency between intra-stage frames and suppress that between inter-stage frames. We can also see that increasing the number of SPS Units improves the performance, showing that discriminative differences can be amplified. Increasing the number of SPS Units to over 2 does not yield further improvements.

TSN	RGB		RGB + Flow	
sample	Top-1 Acc.	Avg. Acc.	Top-1 Acc.	Avg. Acc.
uniform	83.97	82.22	91.01	88.15
ISBA	80.95	79.61	88.88	85.80
CTM	82.51	82.33	89.83	88.11
TCN	81.79	81.10	90.00	87.16
TransParser	**84.80**	**83.30**	**91.62**	**89.26**

Table 4: Performances of TSN [53] on action classification using different sampling schemes.

TransParser-based sampling benefits action recognition. We train a TSN by sampling frames based on the parsing results from Table 2. we can see from Table 4 that compared to other baseline methods the parsing results by TransParser can benefit action classification by a notable improvement ($\sim 1\%$) over the uniform sampling strategy.

6.4. Qualitative results

In this part, we present some qualitative analysis to gain better knowledge of TransParser. First, results on TAPOS

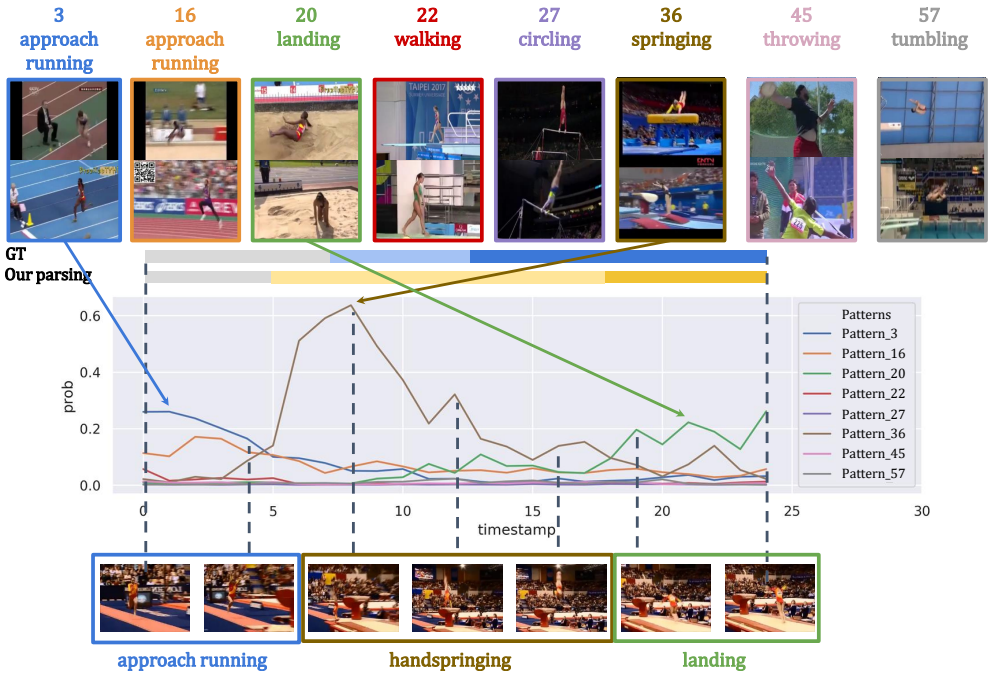

Figure 8: The semantic meaning of selected rows from the pattern miner is illustrated on top. Given a video of a vault (bottom), we visualize the response of activated rows in the miner and provide the predicted parsing results. The figure is best viewed in color.

Figure 9: Qualitative analysis on 50Salads [46]. Patterns from top-left to bottom-right: *place tomato/cheese into bowl*, *add pepper/oil*, *mix dressing/ingredients*, and *cut lettuce/cheese*. Instances in the same box are annotated to be different actions in 50 Salads, but share similar motion patterns as predicted by the TransParser.

are shown in Figure 8. Particularly, we retrieve video frames with highest attention score $\alpha_{t,k}$ with respect to each row of the miner, *i.e.* ϕ_k. It is interesting to observe that different rows of the pattern miner ϕ in SPS are responsive for different sub-actions, *e.g.* approach running, landing, and tumbling. Note that both ϕ_3 and ϕ_{16} are most responsive to approach running but are visually different: the former is from long jump/triple jump and resembles sprinting; the latter is from high jump and is more similar to take-off. Further, certain sub-action occurs in various actions, *e.g.* the sub-action of throwing is common for both discus throwing and javelin throwing. Finally, given a complete instance of vaulting, three stages of approach running, springing onto the vault and landing are clearly observed.

In addition, we also include a qualitative analysis on 50Salads [46] dataset, which is commonly used for action segmentation. The results are shown in Figure 9. As we can see, the automatically mined patterns demonstrate different semantic meanings with human annotated sub-actions, which tend to focus on motion dynamics. For example, *add pepper* and *add oil* are labeled as different classes in 50Salads while they actually follow a similar motion pattern, as predicted by the TransParser.

7. Conclusion

In this paper we propose a new dataset called TAPOS, that digs into the internal structures of action instances, to encourage the exploration towards the hierarchical nature of human actions. In TAPOS, we provide each instance with not only a class label, but also a high-quality temporal parsing annotation at the granularity of sub-actions, which is found to be beneficial for sophisticated action understanding. We also propose an improved method, TransParser, for action parsing, which is capable of identifying underlying patterns of sub-actions without knowing the categorical labels. On TAPOS, TransParser outperforms existing parsing methods significantly. Moreover, with the help of automatically identified patterns, TransParser successfully reveals the internal structure of action instances, and the connections of different action categories.

Acknowledgements. This work is partially supported by SenseTime Collaborative Grant on Large-scale Multi-modality Analysis and the General Research Funds (GRF) of Hong Kong (No. 14203518 and No. 14205719).

References

[1] Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *ICCV*, pages 2127–2136, 2017. 2

[2] Evlampios Apostolidis and Vasileios Mezaris. Fast shot segmentation combining global and local visual descriptors. In *ICASSP*, pages 6583–6587. IEEE, 2014. 3

[3] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402. IEEE, 2005. 2

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2

[5] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *CVPR*, pages 3076–3086, 2017. 2

[6] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *CVPR*, pages 6508–6516, 2018. 1, 6

[7] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 2

[8] Hao-Shu Fang, Jinkun Cao, Yu-Wing Tai, and Cewu Lu. Pairwise body-part attention for recognizing human-object interactions. In *European Conference on Computer Vision*, pages 51–67, 2018. 2

[9] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *CVPR*, June 2019. 2

[10] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, pages 3281–3288. IEEE, 2011. 1, 2

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018. 2

[12] Christoph Feichtenhofer, Axel Pinz, Richard P Wildes, and Andrew Zisserman. What have we learned from deep representations for action recognition? In *CVPR*, pages 7844–7853, 2018. 4

[13] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016. 1, 2

[14] Yixin Gao, S Swaroop Vedula, Carol E Reiley, Narges Ahmidi, Balakrishnan Varadarajan, Henry C Lin, Lingling Tao, Luca Zappella, Benjamın Béjar, David D Yuh, et al. Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In *MICCAI Workshop: M2CAI*, volume 3, page 3, 2014. 2

[15] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/, 2015. 2

[16] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, pages 6047–6056, 2018. 2

[17] Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, and Stephen Maybank. A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):797–819, 2011. 1

[18] 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. 6

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456. JMLR, 2015. 5

[20] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 1, 2

[21] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 2

[22] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, pages 780–787, 2014. 1, 2

[23] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. *IEEE Winter Conference on Applications of Computer Vision*, Mar 2016. 1

[24] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011. 1, 2

[25] Ivan Laptev, Barbara Caputo, et al. Recognizing human actions: a local svm approach. In *International Conference on Pattern Recognition*, pages 32–36. IEEE, 2004. 2

[26] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *CVPR*, pages 156–165, 2017. 2, 6

[27] 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. 1, 2

[28] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *CVPR*, pages 6742–6751, 2018. 2

[29] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *ICCV*, pages 6243–6251, 2019. 2

[30] Yong-Lu Li, Siyuan Zhou, Xijie Huang, Liang Xu, Ze Ma, Hao-Shu Fang, Yanfeng Wang, and Cewu Lu. Transferable interactiveness knowledge for human-object interaction detection. In *CVPR*, 2019. 2

[31] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *European Conference on Computer Vision*, pages 3–19, 2018. 1, 2, 6

[32] Maja J Mataric. Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics. In *Imitation in animals and artifacts*, pages 391–422. MIT Press, 2002. 1

[33] 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. *arXiv preprint arXiv:1906.03327*, 2019. 1

[34] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European Conference on Computer Vision*, pages 392–405. Springer, 2010. 2

[35] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, pages 1817–1824, 2013. 2

[36] Hamed Pirsiavash and Deva Ramanan. Parsing videos of actions with segmental grammars. In *CVPR*, pages 612–619, 2014. 2

[37] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *CVPR*, pages 754–763, 2017. 2

[38] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *CVPR*, pages 5987–5996, 2018. 2

[39] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, pages 1–8. IEEE, 2008. 2

[40] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, pages 1194–1201. IEEE, 2012. 1, 2

[41] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In *European Conference on Computer Vision*, pages 200–216, 2018. 1

[42] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *CVPR*, 2020. 2

[43] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. 2

[44] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 2

[45] 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. 1, 2

[46] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738. ACM, 2013. 1, 2, 8

[47] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, pages 1207–1216, 2019. 1

[48] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 1, 2

[49] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1510–1517, 2018. 2

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 5, 6

[51] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013. 1, 2

[52] Limin Wang, Yu Qiao, and Xiaoou Tang. Latent hierarchical model of temporal structure for complex activity classification. *IEEE Transactions on Image Processing*, 23(2):810–822, 2013. 2

[53] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016. 1, 4, 7

[54] Yilei Xiong, Bo Dai, and Dahua Lin. Move forward and tell: A progressive generator of video descriptions. In *European Conference on Computer Vision*, pages 468–483, 2018. 1

[55] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, pages 5783–5792, 2017. 2

[56] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *CVPR*, 2020. 1, 2

[57] Hang Zhao, Zhicheng Yan, Lorenzo Torresani, and Antonio Torralba. Hacs: Human action clips and segments dataset for recognition and temporal localization. *arXiv preprint arXiv:1712.09374*, 2019. 2

[58] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2914–2923, 2017. 1, 2