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. 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-
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 \( N \times M (> 30\%) \) 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 provide 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 (*e.g.*, frame labels are provided) or a weakly-supervised (*e.g.*, only ordered sub-actions are provided) manner. While TAS relies on a predefined 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.
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.

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 sub-actions, e.g., hammer throw in the second row comprises swing, rotate body and throw.

<table>
<thead>
<tr>
<th></th>
<th>TSN</th>
<th>RGB</th>
<th>RGB + Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>sample</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>uniform</td>
<td>83.97</td>
<td>82.22</td>
<td>91.01</td>
</tr>
<tr>
<td>aligned</td>
<td>88.83</td>
<td>86.22</td>
<td>93.80</td>
</tr>
</tbody>
</table>

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

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 has 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
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 \( \{f_1, ..., f_t\} \in \mathbb{R}^{1 \times d_f} \). Each feature \( f_t \) is then refined by a Soft-Pattern-Strengthen (SPS) unit. The SPS unit maintains a parametric pattern miner \( \phi \) 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 \( f'_t = f_t + \text{SPS}(f_t, \phi) \).

The parsing process of TransParser works as follows. Given refined features of action frames \( \{f'_1, ..., f'_n\} \), we at first compute the responses \( \{\alpha_1, ..., \alpha_n\} \) of them and the patterns stored in \( \phi \). The representative pattern \( (i.e. \text{argmax}_j \alpha_{t,j}) \) 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 \( f_t \), the SPS unit will maintain a pattern miner, parameterized as \( \phi = [\phi_1, ..., \phi_m] \) 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 \( f_t \) and the miner \( \phi \) respectively as query and (key, value) pairs:

\[
\alpha_t = \text{softmax}(f_t \cdot W_Q) \cdot (\phi \cdot W_K)^T, \quad (1)
\]

\[
r_t = \alpha_t \cdot (\phi \cdot W_V), \quad (2)
\]

where \( \cdot \) stands for matrix multiplication, and the output \( r_t \) can be intuitively regarded as a residual of \( f_t \) helpful to distinguish subtle differences. Following [50] in practice we use multi-head attention consisting of two groups of \( \{W_Q, W_K, W_V\} \), and the final \( r_t \) is obtained by feeding the two features \( r_t^{(1)} \) and \( r_t^{(2)} \) from each group respectively into a single fc layer. The output feature \( f'_t \) is computed by identifying the starting frames \( \{v_{t_1}, ..., v_{t_k}\} \) of sub-actions.
inative patterns compared to trivial solutions (e.g. for action categories. For each action instance \( A \), 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:

\[
L_{\text{global}} = \text{NLL}\left( \frac{1}{n} \sum_{i=1}^{n} (W \cdot f'_i), l_A \right),
\]

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

6. Experiments

6.1. Evaluation Metrics

Once the parsing process is finished, we can obtain a series of predicted start frames denoted by \( T_{\mathcal{P}} = \{s_1, s_2, \ldots, s|\mathcal{P}| \} \). Assuming the ground-truth sub-actions start at \( T_{\mathcal{G}} = \{t_1, t_2, \ldots, t|\mathcal{G}| \} \), we can determine the correctness of each prediction if its distance from the nearest ground-truth is smaller than a certain threshold \( d \). The absolute frame distance from 0 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

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 each 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 \( \theta_c \), 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 \( \{f_i \}_{i=1}^N \) 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.
Table 2: Temporal action parsing results on the proposed TAPOS dataset measured by average F1-score.

<table>
<thead>
<tr>
<th># of SPS</th>
<th>local loss</th>
<th>avg. F1</th>
<th>avg. Recall</th>
<th>avg. Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>×1</td>
<td>✓</td>
<td>0.2897</td>
<td>0.6950</td>
<td>0.1832</td>
</tr>
<tr>
<td>×2</td>
<td>✓</td>
<td>0.3996</td>
<td>0.5354</td>
<td>0.3189</td>
</tr>
</tbody>
</table>

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

| &lt;ref&gt;Table 4&lt;/ref&gt;: Performances of TSN [53] on action classification using different sampling schemes.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>uniform</td>
<td>83.97</td>
<td>82.22</td>
<td>91.01</td>
<td>88.15</td>
</tr>
<tr>
<td>ISBA</td>
<td>80.95</td>
<td>79.61</td>
<td>88.88</td>
<td>85.80</td>
</tr>
<tr>
<td>CTM</td>
<td>82.51</td>
<td>82.33</td>
<td>89.83</td>
<td>88.11</td>
</tr>
<tr>
<td>TCN</td>
<td>81.79</td>
<td>81.10</td>
<td>90.00</td>
<td>87.16</td>
</tr>
<tr>
<td>TransParser</td>
<td>84.80</td>
<td>83.30</td>
<td>91.62</td>
<td>89.26</td>
</tr>
</tbody>
</table>

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.

**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 (~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., $\phi_k$. It is interesting to observe that different rows of the pattern miner $\phi$ in SPS are responsive for different sub-actions, e.g. approach running, landing, and tumbling. Note that both $\phi_3$ and $\phi_{16}$ 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


[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


