TubeR: Tubelet Transformer for Video Action Detection

Jiaojiao Zhao[1]*, Yanyi Zhang[2]*, Xinyu Li[3]*, Hao Chen[3], Bing Shuai[3], Mingze Xu[3], Chunhui Liu[3],
Kaustav Kundu[3], Yuanjun Xiong[3], Davide Modolo[3], Ivan Marsic[2], Cees G.M. Snoek[1], Joseph Tighe[3]
[1]University of Amsterdam [2]Rutgers University [3]AWS AI Labs

Abstract

We propose TubeR: a simple solution for spatio-temporal video action detection. Different from existing methods that depend on either an off-line actor detector or hand-designed actor-positional hypotheses like proposals or anchors, we propose to directly detect an action tubelet in a video by simultaneously performing action localization and recognition from a single representation. TubeR learns a set of tubelet-queries and utilizes a tubelet-attention module to model the dynamic spatio-temporal nature of a video clip, which effectively reinforces the model capacity compared to using actor-positional hypotheses in the spatio-temporal space. For videos containing transitional states or scene changes, we propose a context aware classification head to utilize short-term and long-term context to strengthen action classification, and an action switch regression head for detecting the precise temporal action extent. TubeR directly produces action tubelets with variable lengths and even maintains good results for long video clips. TubeR outperforms the previous state-of-the-art on commonly used action detection datasets AVA, UCF101-24 and JHMDB51-21. Code will be available on GluonCV(https://cv.gluon.ai/).

1. Introduction

This paper tackles the problem of spatio-temporal human action detection in videos [3, 17, 39], which plays a central role in advanced video search engines, robotics, and self-driving cars. Action detection is a compound task, requiring the localization of per-frame person instances, the linking of these detected person instances into action tubes and the prediction of their action class labels. Two approaches for spatio-temporal action detection are prevalent in the literature: frame-level detection and tubelet-level detection. Frame-level detection approaches detect and classify the action independently on each frame [14, 29, 32], and then link per-frame detections together into coherent action tubes. To compensate for the lack of temporal information, several methods simply repeat 2D proposals [12, 15, 35] or offline person detections [9, 28, 37, 43] over time to obtain spatio-temporal features (Figure 1 top left).

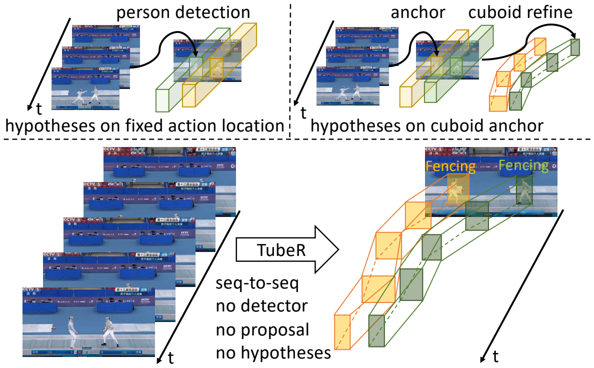

Figure 1. TubeR takes as input a video clip and directly outputs tubelets: sequences of bounding boxes and their action labels. TubeR runs end-to-end without person detectors, anchors or proposals.

Alternatively, tubelet-level detection approaches [16, 19, 26, 33, 45, 49], directly generate spatio-temporal volumes from a video clip to capture the coherence and dynamic nature of actions. They typically predict action localization and classification jointly over spatio-temporal hypotheses, like 3D cuboid proposals [16, 19] (Figure 1 top right). Unfortunately, these 3D cuboids can only capture a short period of time, also when the spatial location of a person changes as soon as they move, or due to camera motion. Ideally, this family of models would use flexible spatio-temporal tubelets that can track the person over a longer time, but the large configuration space of such a parameterization has restricted previous methods to short cuboids. In this work we present a tubelet-level detection approach that is able to simultaneously localize and recognize action tubelets in a flexible manner, which allows tubelets to change in size and location over time (Figure 1 bottom). This allows our system to leverage longer tubelets, which aggregate visual information of a person and their actions over longer periods of time.

We draw inspiration from sequence-to-sequence modelling in natural language processing (NLP), particularly machine translation [21, 24, 36, 40], and its application to object detection, DETR [4]. Being a detection framework,

*Equally contributed and work done while at AWS AI Labs

DETR can be applied as a frame-level action detection approach trivially, but the power of the transformer framework, on which DETR is built, is its ability to generate complex structured outputs over sequences. In NLP, this typically takes the form of sentences but in this work we use the notion of decoder queries to represent people and their actions over video sequences, without having to restrict tubelets to fixed cuboids.

We propose a tubelet-transformer, we call TubeR, for localizing and recognizing actions from a single representation. Building on the DETR framework [4], TubeR learns a set of tubelet queries to pull action-specific tubelet-level features from a spatio-temporal video representation. Our TubeR design includes a specialized spatial and temporal tubelet attention to allow our tubelets to be unrestricted in their spatial location and scale over time, thus overcoming previous limitations of methods restricted to cuboids. TubeR regresses bounding boxes within a tubelet jointly across time, considering temporal correlations between tubelets, and aggregates visual features over the tubelet to classify actions. This core design already performs well, outperforming many previous model designs, but still does not improve upon frame-level approaches using offline person detectors. We hypothesize that this is partially due to the lack of more global context in our query based feature as it is hard to classify actions referring to relationships such as 'listening-to' and 'talking-to' by only looking at a single person. Therefore, we introduce a context aware classification head that, along with the tubelet feature, takes the full clip feature from which our classification head can draw contextual information. This design allows the network to effectively relate a person tubelet to the full scene context where the tubelet appears and is shown to be effective on its own in our results section. One limitation of this design is the context feature is only drawn from the same clip our tubelet occupies. It has been shown [43] to be important to also include long term contextual features for the final action classification. Thus, we introduce a memory system inspired by [44] to compress and store contextual features from video content around the tubelet. We feed this long term contextual memory to our classification head using the same feature injection strategy and again show this gives an important improvement over the short term context alone. We test our full system on three popular action detection datasets (AVA [15], UCF101-24 [34] and JHMDB51-21 [18]) and show our method can outperform other state-of-the-art results.

In summary, our contributions are as follows:

1. We propose TubeR: a tubelet-level transformer framework for human action detection.
2. Our tubelet query and attention based formulation is able to generate tubelets of arbitrary location and scale.
3. Our context aware classification head is able to aggregate short-term and long-term contextual information.
4. We present state-of-the-art results on three challenging action detection datasets.

2. Related Work

Frame-level action detection. Spatio-temporal action detection in video has a long tradition, *e.g*. [3, 15, 17, 28, 29, 37, 39, 42]. Inspired by object detection using deep convolution neural networks, action detection in video has been considerably improved by frame-level methods [29, 31, 32, 42]. These methods perform localization and recognition per-frame and then link frame-wise predictions to action tubes. Specifically, they apply 2D positional hypotheses (anchors) or an offline person detector on a keyframe for localizing actors, and then focus more on improving action recognition. They incorporate temporal patterns by an extra stream utilizing optical flow. Others [12, 15, 35] apply 3D convolution networks to capture temporal information for recognizing actions. Feichtenhofer *et al*. [9] present a slowfast network to even better capture spatio-temporal information. Both Tang *et al*. [37] and Pan *et al*. [28] propose to explicitly model relations between actors and objects. Recently, Chen *et al*. [5] propose to train actor localization and action classification end-to-end from a single backbone. Different from these frame-level approaches, we target on tubelet-level video action detection, with a unified configuration to simultaneously perform localization and recognition.

Tubelet-level action detection. Detecting actions by taking a tubelet as a representation unit [23, 26, 33, 45, 49] has been popular since it was proposed by Jain *et al*. [17]. Kalogeiton *et al*. [19] repeat 2D anchors per-frame for pooling ROI features and then stack the frame-wise features to predict action labels. Hou *et al*. [16] and Yang *et al*. [45] depend on carefully-designed 3D cuboid proposals. The former directly detects tubelets and the later progressively refines 3D cuboid proposals across time. Besides box/cuboid anchors, Li *et al*. [26] detect tubelet instances by relying on center position hypotheses. Hypotheses-based methods have difficulties to process long video clips, as we discussed in the introduction. We add to the tubelet tradition by learning a small set of tubelet queries to represent the dynamic nature of tubelets. We reformulate the action detection task as a sequence-to-sequence learning problem and explicitly model the temporal correlations within a tubelet. Our method is capable to handle long video clips.

Transformer-based action detection. Vaswani *et al*. [40] proposed the transformer for machine translation, which soon after became the most popular backbone for sequence-to-sequence tasks, *e.g*., [21, 24, 36]. Recently, it has also demonstrated impressive advances in object detection [4, 50], image classification [6, 46] and video recognition [7, 10, 47]. Girdhar *et al*. [13] propose a video action transformer network for detecting actions. They apply a region-proposal-network for localization. The transformer is utilized for further improving action recognition by aggregating features from the spatio-temporal context around actors. We propose a unified solution to simultaneously localize and recognize actions.

3. Action Detection by TubeR

In this section, we present our TubeR that takes as input a video clip and directly outputs a tubelet: a sequence of bounding boxes and the action label. The TubeR design takes inspiration from the image-based DETR [4] but reformulates the transformer architecture for sequence-to-sequence(s) modeling in video (Figure 2).

Given a video clip $I \in \mathbb{R}^{T_{\text{in}} \times H \times W \times C}$ where T_{in}, H, W, C denote the number of frames, height, width, and colour channels, TubeR first applies a 3D backbone to extract video feature $F_{\text{b}} \in \mathbb{R}^{T' \times H' \times W' \times C'}$, where T' is the temporal dimension and C' is the feature dimension. A transformer encoder-decoder is then utilized to transform the video feature into a set of tubelet-specific feature $F_{\text{tub}} \in \mathbb{R}^{N \times T_{\text{out}} \times C'}$, with T_{out} the output temporal dimension and N the number of tubelets. In order to process long video clips, we use temporal down-sampling to make $T_{\text{out}} < T' < T_{\text{in}}$, which reduces our memory requirement. In this case, TubeR generates sparse tubelets. For short video clips we remove the temporal down-sampling to make sure $T_{\text{out}}{=}T'{=}T_{\text{in}}$, which results in dense tubelets. Tubelet regression and associated action classification can be achieved simultaneously with separated task heads as:

$$y_{\text{coor}} = f(F_{\text{tub}}); y_{\text{class}} = g(F_{\text{tub}}), \tag{1}$$

where f denotes the tubelet regression head and $y_{\text{coor}} \in \mathbb{R}^{N \times T_{\text{out}} \times 4}$ stands for the coordinates of N tubelets, each of which is across T_{out} frames (or T_{out} sampled frames for long video clips). Here g denotes the action classification head, and $y_{\text{class}} \in \mathbb{R}^{N \times L}$ stands for the action classification for N tubelets with L possible labels.

3.1. TubeR Encoder

Different from the vanilla transformer encoder, the TubeR encoder is designed for processing information in the 3D spatio-temporal space. Each encoder layer is made up of a self-attention layer (SA), two normalization layers and a feed forward network (FFN), following [40]. We only put the core attention layers in all equations below.

$$F_{\text{en}} = \text{Encoder}(F_{\text{b}}), \tag{2}$$

$$\text{SA}(F_{\text{b}}) = \text{softmax}(\frac{\sigma_q(F_{\text{b}}) \times \sigma_k(F_{\text{b}})^T}{\sqrt{C'}}) \times \sigma_v(F_{\text{b}}), \tag{3}$$

$$\sigma(*) = \text{Linear}(*) + \text{Emb}_{\text{pos}}, \tag{4}$$

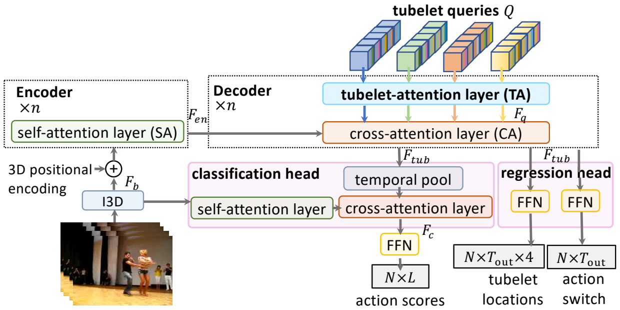

Figure 2. The overall structure of TubeR. Both encoder and decoder contain n stacked modules. We only show the key components in the encoder and decoder modules. The encoder models the spatio-temporal features from the backbone F_{b} by self-attention layers (see Section 3.1). The decoder transforms a set of tubelet queries Q and generates tubelet-level features F_{tub}. We utilize tubelet-attention layers to model the relations between box query embeddings within a tubelet (see Section 3.2). Finally, we apply the context aware classification head and action switch regression head to predict tubelet labels and coordinates (see Section 3.3).

where F_{b} is the backbone feature and $F_{\text{en}} \in \mathbb{R}^{T'H'W' \times C'}$ denotes the C' dimensional encoded feature embedding. The $\sigma(*)$ is the linear transformation plus positional embedding. Emb_{pos} is the 3D positional embedding [47]. The optional temporal down-sampling can be applied to the backbone feature to shrink the input sequence length to the transformer for better memory efficiency.

3.2. TubeR Decoder

Tubelet query. Directly detecting tubelets is quite challenging based on anchor hypotheses. The tubelet space along the spatio-temporal dimension is huge compared to the single-frame bounding box space. Consider for example Faster-RCNN [30] for object detection, which requires for each position in a feature map with spatial size $H' \times W'$, $K (=9)$ anchors. There are in total $KH'W'$ anchors. For a tubelet across T_{out} frames, it would require $(KH'W')^{T_{\text{out}}}$ anchors to maintain the same sampling in space-time. To reduce the tubelet space, several methods [16, 45] adopt 3D cuboids to approximate tubelets by ignoring the spatial action displacements in a short video clip. However, the longer the video clip is, the less accurately a 3D cuboid hypotheses represents a tubelet. We propose to learn a small set of tubelet queries $Q{=}\{Q_1, ..., Q_N\}$ driven by the video data. N is the number of queries. The i-th tubelet query $Q_i{=}\{q_{i,1}, ..., q_{i,T_{\text{out}}}\}$ contains T_{out} box query embeddings $q_{i,t} \in \mathbb{R}^{C'}$ across T_{out} frames. We learn a tubelet query to represent the dynamics of a tubelet, instead of hand-designing 3D anchors. We initialize the box embeddings identically for a tubelet query.

Tubelet attention. In order to model relations in the tubelet queries, we propose a tubelet-attention (TA) module which contains two self-attention layers (shown in Figure 2). First we have a *spatial self-attention layer* that processes the spatial relations between box query embeddings within a frame *i.e.* $\{q_{1,t}, ..., q_{N,t}\}$, t=$\{1, ..., T_{\text{out}}\}$. The intuition of this layer is that recognizing actions benefits from the interactions between actors, or between actors and objects in the same frame. Next we have our *temporal self-attention layer* that models the correlations between box query embeddings across time within the same tubelet, *i.e.* $\{q_{i,1}, ..., q_{i,T_{\text{out}}}\}$, i=$\{1, ..., N\}$. This layer facilitates a TubeR query to track actors and generate action tubelets that focus on single actors instead of a fixed area in the frame. TubeR decoder applies the tubelet attention module to tubelet queries Q for generating the tubelet query feature $F_{\text{q}} \in \mathbb{R}^{N \times T_{\text{out}} \times C'}$:

$$F_q = \text{TA}(Q). \tag{5}$$

Decoder. The decoder contains a tubelet-attention module and a cross-attention (CA) layer which is used to decode the tubelet-specific feature F_{tub} from F_{en} and F_{q}:

$$\text{CA}(F_q, F_{\text{en}}) = \text{softmax}(\frac{F_q \times \sigma_k(F_{\text{en}})^T}{\sqrt{C'}}) \times \sigma_v(F_{\text{en}}), \tag{6}$$

$$F_{\text{tub}} = \text{Decoder}(F_q, F_{\text{en}}). \tag{7}$$

$F_{\text{tub}} \in \mathbb{R}^{N \times T_{\text{out}} \times C'}$ is the tubelet specific feature. Note that with temporal pooling, $T_{\text{out}} < T_{\text{in}}$, TubeR produces sparse tubelets; For T_{out}=T_{in}, TubeR produces dense tubelets.

3.3. Task-Specific Heads

The bounding boxes and action classification for each tubelet can be done simultaneously with independent task-specific heads. Such design maximally reduces the computational overheads and makes our system expandable.

Context aware classification head. The classification can be simply achieved with a linear projection.

$$y_{\text{class}} = \text{Linear}_{\text{c}}(F_{\text{tub}}), \tag{8}$$

where $y_{\text{class}} \in \mathbb{R}^{N \times L}$ denotes the classification score on L possible labels, one for each tubelet.

Short-term context head. It is known that context is important for understanding sequences [40]. We further propose to leverage spatio-temporal video context to help video sequence understanding. We query the action specific feature F_{tub} from some context feature F_{context} to strengthen F_{tub}, and get the feature $F_{\text{c}} \in \mathbb{R}^{N \times C'}$ for the final classification:

$$F_{\text{c}} = \text{CA}(\text{Pool}_t(F_{\text{tub}}), \text{SA}(F_{\text{context}})) + \text{Pool}_t(F_{\text{tub}}). \tag{9}$$

When we set F_{context}=F_{b} for utilizing the short-term context in the backbone feature, we call it short-term context head. A self-attention layer is first applied to F_{context}, then a cross-attention layer utilizes F_{tub} to query from F_{context}. The Linear$_{\text{c}}$ is applied to F_{c} for final classification.

Long-term context head. Inspired by [41, 43, 47] which explore long-range temporal information for video understanding, we propose a long-term context head. To utilize long-range temporal information but under certain memory budget, we adopt a two-stage decoder for long-term context compression as described in [44]:

$$\text{Emb}_{\text{long}} = \text{Decoder}(\text{Emn}_{n1}, \text{Decoder}(\text{Emb}_{n0}, F_{\text{long}}). \tag{10}$$

The long-term context $F_{\text{long}} \in \mathbb{R}^{T_{\text{long}} \times H'W' \times C'}$ (T_{long}=$(2w + 1)T'$) is a buffer that contains the backbone feature extracted from a long $2w$ adjacent clips concatenated along time. In order to compress the long-term video feature buffer to an embedding Emb$_{\text{long}}$ with a lower temporal dimension, we apply two stacked decoders with two token embedding Emn$_{n0}$ and Emn$_{n1}$. Specifically, we first apply a compressed token Emb$_{n_0}$ ($n_0 < T_{\text{long}}$) to query important information from F_{long} and get an intermediary compressed embedding with temporal dimension n_0. Then we further utilize another compressed token Emb$_{n_1}$ ($n_1 < n_0$) to query from the intermediary compressed embedding and get the final compressed embedding Emb$_{\text{long}}$. Emb$_{\text{long}}$ contains the long-term video information but with a lower temporal dimension n_1. Then we adopt a cross-attention layer to F_{b} and Emb$_{\text{long}}$ to generate a long-term context feature $F_{\text{lt}} \in \mathbb{R}^{T' \times H' \times W' \times C'}$:

$$F_{\text{lt}} = \text{CA}(F_{\text{b}}, \text{Emb}_{\text{long}}), \tag{11}$$

we set $F_{\text{context}} = F_{\text{lt}}$ in Eq. 9 to utilize the long-term context for classification.

Action switch regression head. The T_{out} bounding boxes in a tubelet are simultaneously regressed with an FC layer as:

$$y_{\text{coor}} = \text{Linear}_{\text{b}}(F_{\text{tub}}), \tag{12}$$

where $y_{\text{coor}} \in \mathbb{R}^{N \times T_{\text{out}} \times 4}$, N is the number of action tubelets, and T_{out} is the temporal length of an action tubelet. To remove non-action boxes in a tubelet, we further include an FC layer for deciding whether a box prediction depicts the actor performing the action(s) of the tubelet, we call action switch. The action switch allows our method to generate action tubelets with a more precise temporal extent. The probabilities of the T_{out} predicted boxes in a tubelet being visible are:

$$y_{\text{switch}} = \text{Linear}_{\text{s}}(F_{\text{tub}}), \tag{13}$$

where $y_{\text{switch}} \in \mathbb{R}^{N \times T_{\text{out}}}$. For each predicted tubelet, each of its T_{out} bounding boxes obtain an action switch score.

3.4. Losses

The total loss is a linear combination of four losses:

$$\begin{aligned}\mathcal{L} = \lambda_1 \mathcal{L}_{\text{switch}}(y_{\text{switch}}, Y_{\text{switch}}) + \lambda_2 \mathcal{L}_{\text{class}}(y_{\text{class}}, Y_{\text{class}}) \\ + \lambda_3 \mathcal{L}_{\text{box}}(y_{\text{coor}}, Y_{\text{coor}}) + \lambda_4 \mathcal{L}_{\text{iou}}(y_{\text{coor}}, Y_{\text{coor}}),\end{aligned} \tag{14}$$

where y is the model output and Y denotes the ground truth. The action switch loss $\mathcal{L}_{\text{switch}}$ is a binary cross entropy loss. The classification loss $\mathcal{L}_{\text{class}}$ is a cross entropy loss. The $\mathcal{L}_{\text{box}}$ and $\mathcal{L}_{\text{iou}}$ denote the per-frame bounding box matching error. It is noted when $T_{\text{out}} < T_{\text{in}}$, the tubelet is sparse and the coordinate ground truth Y_{coor} are from the corresponding temporally down-sampled frame sequence. We used the Hungarian matching similar to [4] and more details can be found in the supplementary. We empirically set the scale parameter as $\lambda_1{=}1$, $\lambda_2{=}5$, $\lambda_3{=}2$, $\lambda_4{=}2$.

4. Experiments

4.1. Experimental Setup

Datasets. We report experiments on three commonly used video datasets for action detection. **UCF101-24** [34] is a subset of UCF101. It contains 24 sport classes in 3207 untrimmed videos. We use the revised annotations for UCF101-24 from [32] and report the performance on split-1. **JHMDB51-21** [18] contains 21 action categories in 928 trimmed videos. We report the average results over all three splits. **AVA** [15] is larger-scale and includes 299 15-minute movies, 235 for training, and the remaining 64 for validating. Box and label annotations are provided on per-second sampled keyframes. We evaluate on AVA with both annotation versions v2.1 and v2.2.

Evaluation criteria. We report the video-mAP at different IoUs on UCF101-24 and JHMDB51-21. As AVA only has keyframe annotations, we report frame-mAP@IoU=0.5 following [15] using a single, center-crop inference protocol.

Implementation details. We pre-train the backbone on Kinetics-400 [20]. The encoder and decoder contain 6 blocks on AVA. For the smaller UCF101-24 and JHMDB51-21, we reduce the numbers of blocks to 3 to avoid overfitting. We empirically set the number of tubelet query N to 15. During **training**, we use the bipartite matching [11] based on the Hungarian algorithm [22] between predictions and the ground truth. We use the AdamW [27] optimizer with an initial learning rate $1e{-}5$ for the backbone and $1e{-}4$ for the transformers. We decrease the learning rate 10× when the validation loss saturates. We set $1e{-}4$ as the weight decay. Scale jittering in the range of (288, 320) and color jittering are used for data augmentation. During **inference**, we always resize the short edge to 256 and use a single center-crop (1-view). We also tested the horizontal flip trick to create 2-view inference. For fair comparisons with previous methods on UCF101-24 and JHMDB51-21, we also test a two-stream setting with optical flow following [49].

4.2. Ablations

We perform our ablations on both UCF101-24 and AVA 2.1 to demonstrate the effectiveness of our designs on different evaluation protocols. Only RGB inputs are considered.

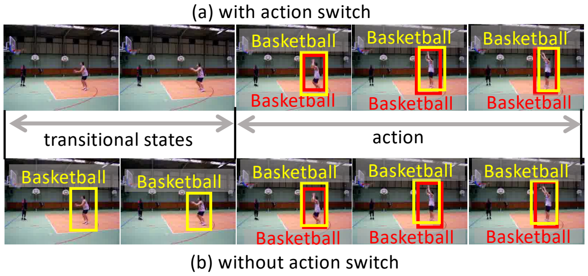

Figure 3. Visualizations of action switch on UCF101-24. Best view in color. The red box and label represent the ground truth. Yellow indicates our detected tubelets. With the action switch (top row), TubeR avoids misclassification for the transitional states.

For UCF101-24 with per-frame annotations, we report **video-mAP** at IoU=0.5. A standard backbone I3D-VGG [15] is utilized and the input length is set to 7 frames if not specified. For AVA 2.1 with 1-fps annotation, we only take the model prediction on keyframes and report **frame-mAP** at IoU=0.5. We use a CSN-50 backbone [38] with a single view evaluation protocol if not specified.

Benefit of tubelet queries. We first show the benefit of the proposed tubelet query sets. Each query set is composed of T_{out} per-frame query embeddings (see section 3.2), which predict the spatial location of the action on their respective frames. We compare this to using a single query embedding that represents a whole tubelet and must regress T_{out} box locations for all frames in the clip. Our results are shown in Table 1a. Compared to using a single query embedding, our tubelets query set improves performance by +4.1% video mAP on UCF101-24, showing that modeling action detection as a sequence-to-sequence task effectively leverages the capabilities of transformer architectures.

Effect of tubelet attention. In Table 1b, we show using our tubelet attention module helps improve video-*mAP* on UCF101-24 by 0.9% and 0.3% on AVA. The tubelet attention saves about 10% memory ($4,414$MB) compared to the typical self-attention implementation ($5,026$MB) during training (16 frames input with batch size of 1).

Benefit of action switch. We report the effectiveness of our action switch head in Table 1c. On UCF101-24 the action switch increases the video-mAP from 53.8% to 57.7% by precisely determining the temporal start and end point of actions. Without action switch, TubeR misclassifies transitional states as actions, like the example shown in Figure 3 (bottom row). As only the frame-level evaluation can be done on AVA, the advantage of the action switch is not shown by the frame-mAP. Instead, we demonstrate its effect in Figure 4 and Figure 5. The action switch produces tubelets with precise temporal extent for videos with shot changes.

Effect of short and long term context head. We report the impact of our context aware classification head with both short and long-term features in Table 1d. The context head

	UCF101-24	*AVA*
single query	48.8	26.2
tubelet query set	52.9	27.4

(a) **Analysis on tubelet query.** Our tubelet query set design allows for each query to focus on the spatial location of the action on a specific frame.

	UCF101-24	*AVA*
self-attention	52.9	27.4
tubelet attention	53.8	27.7

(b) **Effect of tubelet attention.** With tubelet attention modeling relations within a tubelet and across tubelets improves.

	UCF101-24	*AVA*
w/o switch	53.8	27.7
w/ switch	57.7	27.7

(c) **Benefit of action switch.** Action switch produces a more precise temporal extent, which can only be shown by video-mAP.

	UCF101-24	*AVA*
FC head	57.8	23.4
+ short-term context	58.4	27.7
+ long-term context	-	28.8

(d) **Effectiveness of short- and long-term context.** The short-term context and long-term context help with performance, more noticeable on AVA.

	UCF101-24	*AVA*
8	53.9	24.4
16	58.2	26.9
32	58.4	27.7

(e) **Length of input clip.** Longer input video leads to a better performance on both UCF101-24 and AVA.

w	# of clips	duration (s)	mAP
-	1	2.1	27.7
2	5	10.6	28.4
3	7	14.9	28.8
5	11	23.5	28.6

(f) **Long-term context length** analysis on AVA. The right amount of long-term context helps improve frame-mAP on AVA.

Table 1. Ablation studies on UCF101-24 and AVA 2.1. The proposed tubelet query, tubelet attention, the action switch and context-awareness generally improve model performance. The proposed TubeR works well on long clips with shot changes. We report video-mAP@IoU=0.5 for UCF101-24 and frame-mAP@IoU=0.5 for AVA.

brings a decent performance gain (+4.3%) on AVA. This is probably because the movie clips in AVA contain shot changes and so the network benefits from seeing the full context of the clip. On UCF101-24, the videos are usually short and without shot changes. The context does not bring a significant improvement on UCF101-24.

Length of input clip. We report results with variable input lengths in Table 1e. We compare with input length of 8, 16 and 32 on both UCF101-24 and AVA with CSN-152 as backbone. TubeR is able to handle long video clips as expected. We notice that our performance on UCF101-24 saturates faster than on AVA, probably because UCF101-24 does not contain shot changes that requires longer temporal context for classification.

Length of long-term context. This ablation is only conducted on AVA as videos on UCF101-24 are too short to use long-term context. Table 1f shows that the right amount of long-term context helps performance, but overwhelming the amount of long-term context harms performance. This is probably because the long-term feature contains both useful information and noise. The experiments show that about 15s context serves best. Note that the context length varies per dataset, but can be easily determined empirically.

4.3. Frame-Level State-of-the-Art

AVA 2.1 Comparison. We first compare our results with previously proposed methods on AVA 2.1 in Table 2. Compared to previous end-to-end models, with comparable backbone (I3D-Res50) and the same inference protocol, the proposed TubeR outperforms all. TubeR outperforms the most recent end-to-end works WOO [5] by 0.9% and VTr [13] by 1.2%. This demonstrates the effectiveness of our designs.

Compared to previous work using an offline person detector, the proposed TubeR is also more effective under the same inference protocols. This is because TubeR generates tubelet-specific features without assumptions on location, while the two-stage methods have to assume the actions occur at a fixed location. It is also worth mentioning that the TubeR with CSN backbones outperforms the two-stage model with the same backbone by +4.4%, demonstrating that the gain is not from the backbone but our TubeR design. TubeR even outperforms the methods with multi-view augmentations (horizontal flip, multiple spatial crops and multi-scale). TubeR is also considerably faster than previous models, we have attempted to collect the reported FLOPs from previous works (Table 2). Our TubeR has 8% fewer FLOPs than the most recently published end-to-end model [5] with higher accuracy. Tuber is also 4× more efficient than the two-stage model [9] with noticeable performance gain. Thanks to our sequence-to-sequence design, the heavy backbone is shared and we do not need temporal iteration for tubelet regression.

We finally present the highest number reported in the literature, regardless of the inference protocol, pre-training dataset and additional information used. TubeR still achieves the best performance, even better than the model using additional object bounding-boxes as input [37].The results show that the proposed sequence-to-sequence model with tubelet specific feature is a promising direction for action detection.

AVA 2.2 Comparison. The results are shown in Table 3. Under the same single-view protocol, TubeR is considerably better than previous methods, including the most recent work with an end-to-end design (WOO [5] +5.1%) and the two-stage work with strong backbones (MViT [7] +4.7%). A fair comparison between TubeR and a two-stage model [48] with the same backbone CSN-152, shows TubeR gains +5.5% frame-mAP. It demonstrates TubeR's superior performance comes from our design rather than the backbone.

UCF101-24 Comparison. We also compare TubeR with the

Model	Detector	Input	Backbone	Pre-train	Inference	GFLOPs	mAP
Comparison to end-to-end models							
I3D [15]	✗	32 × 2	I3D-VGG	K400	1 view	NA	14.5
ACRN [35]	✗	32 × 2	S3D-G	K400	1 view	NA	17.4
STEP [45]	✗	32 × 2	I3D-VGG	K400	1 view	NA	18.6
VTr [13]	✗	64 × 1	I3D-VGG	K400	1 view	NA	24.9
WOO [5]	✗	8 × 8	SF-50	K400	1 view	142	25.2
TubeR	✗	16 × 4	I3D-Res50	K400	1 view	132	**26.1**
TubeR	✗	16 × 4	I3D-Res101	K400	1 view	246	**28.6**
Comparison to two-stage models							
Slowfast-50 [9]	F-RCNN	16 × 4	SF-50	K400	1 view	308	24.2
X3D-XL [8]	F-RCNN	16 × 5	X3D-XL	K400	1 view	290	26.1
CSN-152*	F-RCNN	32 × 2	CSN-152	IG + K400	1 views	342	27.3
LFB [43]	F-RCNN	32 × 2	I3D-101-NL	K400	18 views	NA	27.7
ACAR-NET [28]	F-RCNN	32 × 2	SF-50	K400	6 views	NA	28.3
TubeR	✗	32 × 2	CSN-50	K400	1 view	78	**28.8**
TubeR	✗	32 × 2	CSN-152	IG + K400	1 view	120	**31.7**
Comparison to best reported results							
WOO [5]	✗	8 × 8	SF-101	K400+K600	1 view	246	28.0
SF-101-NL [9]	F-RCNN	32 × 2	SF-101+NL	K400+K600	6 views	962	28.2
ACAR-NET [28]	F-RCNN	32 × 2	SF-101	K400+K600	6 views	NA	30.0
AIA [37]	F-RCNN	32 × 2	SF-101	K400+K700	18 views	NA	31.2
TubeR	✗	32 × 2	SF-101	K400+K700	1 view	240	**31.6**
TubeR	✗	32 × 2	CSN-152	IG + K400	2 view	240	**32.0**

Table 2. **Comparison on AVA v2.1** validation set. Detector shows if additional detector is required; * denotes the results we tested. IG denotes the IG-65M dataset, SF denotes the slowfast network. The FLOPs for two-stage models are the sum of Faster RCNN-R101-FPN FLOPs (246 GFLOPs [4]) plus classifier FLOPs multiplied by view number. TubeR performs more effectively and efficiently.

Model	backbone	pre-train	inference	mAP
Single-view				
X3D-XL [8]	X3D-XL	K600+ K400	1 view	27.4
CSN-152 [48]	CSN-152	IG + K400	1 view	27.9
WOO [5]	SF-101	K600+ K400	1 view	28.3
M-ViT-B-24 [7]	MViT-B-24	K600+ K400	1 view	28.7
TubeR	CSN-50	IG + K400	1 view	29.2
TubeR	CSN-152	IG + K400	1 view	33.4
Multi-view				
SlowFast-101 [9]	SF-101	K600+ K400	6 views	29.8
ACAR-Net [28]	SF-101	K700+ K400	6 views	33.3
AIA (obj) [37]	SF-101	K700+ K400	18 views	32.2
TubeR	CSN-152	IG + K400	2 views	**33.6**

Table 3. **Comparison on AVA v2.2** validation set. IG denotes the IG-65M, SF denotes the slowfast. TubeR achieves the best result.

state-of-the-art using frame-mAP@IoU=0.5 on UCF101-24 (see the first column with numbers in Table 4). Compared to existing methods, TubeR acquires better results with comparable backbones, for both RGB-stream and two-stream settings. Further with a CSN-152 backbone, TubeR gets 83.2 frame-mAP, even better than two-stream methods. Though TubeR targets on tubelet-level detection, it performs well on frame-level evaluation on both AVA and UCF101-24.

4.4. Video-Level State-of-the-Art

We also compare TubeR with various settings to state-of-the-art reporting video-mAP on UCF101-24 and JHMDB51-21 in Table 4. For fair comparisons, TubeR with a 2D backbone gains +4.4% video-mAP@IoU=0.5 compared to the recent state-of-the-art [26] on UCF101-24 without using optical flow, which demonstrates that TubeR learning tubelet queries is more effective compared to using positional hypotheses. Compared to TacNet [33] which proposes a transition-aware context network to distinguish transitional

		UCF101-24				*JHMDB51-21*	
	Backbone	f-mAP	0.20	0.50	0.50:0.95	0.20	0.50
RGB-stream							
MOC [26]	DLA34	72.1	78.2	50.7	26.2	-	-
TubeR	Res50	79.5	81.2	55.1	28.1	-	-
T-CNN [16]	C3D	41.4	47.1	-	-	78.4	76.9
TubeR	I3D	80.1	82.8	57.7	28.6	79.7	78.3
TubeR	CSN-152	**83.2**	**83.3**	**58.4**	**28.9**	**87.4**	**82.3**
Two-stream							
TacNet [33]	VGG	72.1	77.5	52.9	24.1	-	-
2in1 [49]	VGG		78.5	50.3	24.5	-	74.7
ACT [19]	VGG	67.1	77.2	51.4	25.0	74.2	73.7
MOC [26]	DLA34	78.0	82.8	53.8	28.3	77.3	77.2
STEP [45]	I3D	75.0	76.6	-	-	-	-
I3D [15]	I3D	76.3	-	59.9	-	-	78.6
*CFAD [25]	I3D	72.5	81.6	**64.6**	26.7	**86.8**	**85.3**
TubeR	I3D	**81.3**	**85.3**	60.2	**29.7**	81.8	80.7

Table 4. **Comparison on UCF101-24 and JHMDB51-21** with video-*mAP*. TubeR achieves better results compared to most state-of-arts. f-mAP denotes the frame mAP@IoU=0.5. *CFAD is pre-trained on K600 but others on K400.

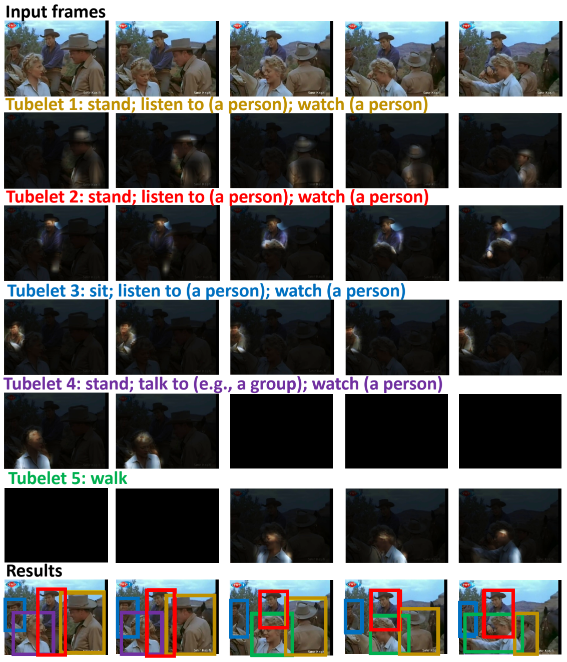
Input frames
Tubelet 1: stand; listen to (a person); watch (a person)
Tubelet 2: stand; listen to (a person); watch (a person)
Tubelet 3: sit; listen to (a person); watch (a person)
Tubelet 4: stand; talk to (e.g., a group); watch (a person)
Tubelet 5: walk
Results

Figure 4. Visualization of tubelet specific feature with attention rollout. Each tubelet covers a separated action instance. Best viewed in color.

states, TubeR with action switch performs better even with a one-stream setting. When incorporating optical flow inputs, the TubeR with I3D further boosts the video-level results. It is noted that TubeR pretrained on K400 even outperforms CFAD pretrained on K600 on some metrics. We test TubeR inference speed on UCF101-24 by following CFAD. To directly generate a tubelet without an offline linker, TubeR runs at 156 fps. Faster than CFAD (130fps) and most existing SOTA methods (40-53 fps). The result illustrates our design is effective and efficient for video-level action detection.

4.5. Visualization

We first provide visualizations (Figure 4) of the tubelet-specific features by overlaying the tubelet-specific feature activation over the input frames using attention rollout [1]. The example in Figure 4 is challenging as it contains multiple people and concurrent actions. The visualization show that: 1. Our proposed TubeR is able to generate highly discriminative tubelet-specific features. Different actions in this case are clearly separated in different tubelets. 2. Our action switch works as expected and initiates/cuts the tubelets when the action starts/stops. 3. Our TubeR generalizes well to scale changes (the brown tubelet). 4. The generated tubelets are tightly associated with tubelet specific feature as expected.

We further show our TubeR performs well in various scenarios. TubeR works well on videos with shot changes (Figure 5 top); TubeR is able to detect an actor moving with distance (Figure 5 middle); and TubeR is robust to action detection even for small people (Figure 5 bottom).

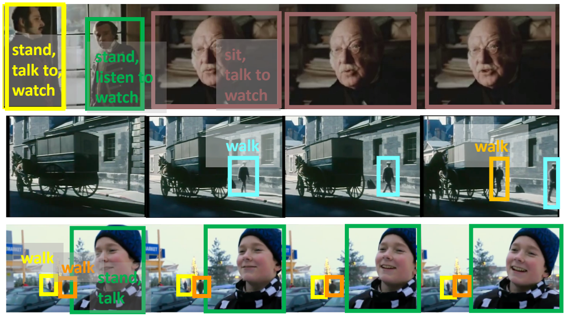

Figure 5. Results visualization, with different colors to label different tubelets. Each action tubelet contains its action labels and boxes per frame. We only show the action labels on the first frame of an action tubelet. Some challenging cases are shown. Top: shot changes; Middle: actors moving with distance; Bottom: multiple actors with small and large scales. Best viewed in color.

5. Discussion and Conclusion

Limitations. Although proposed for long videos, we noticed two potential limitations that stop us from feeding in very long videos in one shot.
1. We observe that 90% of computation (FLOPs) and 67% of memory usage was used by our 3D backbone. This heavy backbone restricts us from applying TubeR on long videos. Recent works show that transformer encoders can be used for video embedding [2, 7, 47] and are less memory and computationally hungry. We will explore these transformer based embeddings in future work.
2. If we were to process a long video in one pass we'd need enough queries to cover the maximum number of different actions per-person in that video. This would likely require a large number of queries which would cause memeory issues in our self attention layers. A possible solution is to generate person tubelets, instead of action tubelets, so that we do not need to split tubelets when a new action happens. Then we would only need a query for each person instance.
Potential negative impact. There are real-world applications of action detection technology such as patient or elderly health monitoring, public safety, Augmented/Virtual Reality, and collaborative robots. However, there could be unintended usages and we advocate responsible usage and complying with applicable laws and regulations.
Conclusion. This paper introduces TubeR, a unified solution for spatio-temporal video action detection in a sequence-to-sequence manner. Our design of tubelet-specific features allows TubeR to generate tubelets (a set of linked bounding boxes) with action predictions for each of the tubelets. TubeR does not rely on positional hypotheses and therefore scales well to longer video clips. TubeR achieves state-of-the-art performance and better efficiency compared to previous works.

References

[1] Samira Abnar and Willem H Zuidema. Quantifying attention flow in transformers. In *ACL*, 2020. 8

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv:2103.15691*, 2021. 8

[3] Liangliang Cao, Zicheng Liu, and Thomas S Huang. Cross-dataset action detection. In *CVPR*, 2010. 1, 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2, 3, 5, 7

[5] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. Watch only once: An end-to-end video action detection framework. In *ICCV*, 2021. 2, 6, 7

[6] 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. In *ICLR*, 2021. 2

[7] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv:2104.11227*, 2021. 2, 6, 7, 8

[8] Christoph Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 7

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *CVPR*, 2019. 1, 2, 6, 7

[10] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees GM Snoek. Actor-transformers for group activity recognition. In *CVPR*, 2020. 2

[11] Dobrik Georgiev and Pietro Lió. Neural bipartite matching. *arXiv:2005.11304*, 2020. 5

[12] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. A better baseline for ava. *arXiv:1807.10066*, 2018. 1, 2

[13] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019. 2, 6, 7

[14] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *CVPR*, 2015. 1

[15] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 1, 2, 5, 7

[16] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *ICCV*, 2017. 1, 2, 3, 7

[17] Mihir Jain, Jan van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees GM Snoek. Action localization with tubelets from motion. In *CVPR*, 2014. 1, 2

[18] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, 2013. 2, 5

[19] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, 2017. 1, 2, 7

[20] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv:1705.06950*, 2017. 5

[21] Aisha Urooj Khan, Amir Mazaheri, Niels da Vitoria Lobo, and Mubarak Shah. MMFT-BERT: Multimodal fusion transformer with bert encodings for visual question answering. *arXiv:2010.14095*, 2020. 1, 2

[22] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955. 5

[23] Dong Li, Zhaofan Qiu, Qi Dai, Ting Yao, and Tao Mei. Recurrent tubelet proposal and recognition networks for action detection. In *ECCV*, 2018. 2

[24] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *ICCV*, 2019. 1, 2

[25] Yuxi Li, Weiyao Lin, John See, Ning Xu, Shugong Xu, Ke Yan, and Cong Yang. Cfad: Coarse-to-fine action detector for spatiotemporal action localization. In *ECCV*, 2020. 7

[26] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *ECCV*, 2020. 1, 2, 7

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2017. 5

[28] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *CVPR*, 2021. 1, 2, 7

[29] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *ECCV*, 2016. 1, 2

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 3

[31] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. *arXiv:1608.01529*, 2016. 2

[32] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*, 2017. 1, 2, 5

[33] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. Tacnet: Transition-aware context network for spatio-temporal action detection. In *CVPR*, 2019. 1, 2, 7

[34] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012. 2, 5

[35] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, 2018. 1, 2, 7

[36] Chiranjib Sur. Self-segregating and coordinated-segregating transformer for focused deep multi-modular network for visual question answering. *arXiv:2006.14264*, 2020. 1, 2

[37] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *ECCV*, 2020. 1, 2, 6, 7

[38] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, 2019. 5

[39] Du Tran and Junsong Yuan. Max-margin structured output regression for spatio-temporal action localization. In *NIPS*, 2012. 1, 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. In *NIPS*, 2017. 1, 2, 3, 4

[41] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 4

[42] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, 2015. 2

[43] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019. 1, 2, 4, 7

[44] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. In *NeurIPS*, 2021. 2, 4

[45] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *CVPR*, 2019. 1, 2, 3, 7

[46] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv:2101.11986*, 2021. 2

[47] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr-switch: Video transformer without convolutions. In *ICCV*, 2021. 2, 3, 4, 8

[48] Yanyi Zhang, Xinyu Li, and Ivan Marsic. Multi-label activity recognition using activity-specific features and activity correlations. In *CVPR*, 2021. 6, 7

[49] Jiaojiao Zhao and Cees GM Snoek. Dance with flow: Two-in-one stream action detection. In *CVPR*, 2019. 1, 2, 5, 7

[50] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2