Computer Vision and Image Understanding 192 (2020) 102886

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Guess where? Actor-supervision for spatiotemporal action localization[☆]

Victor Escorcia[a,*], Cuong D. Dao[a], Mihir Jain[b], Bernard Ghanem[a], Cees Snoek[c]

[a] *King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia*
[b] *Qualcomm AI Research[1], Qualcomm Technologies Netherlands B.V., 1098 XH Amsterdam, Netherlands*
[c] *University of Amsterdam, 1012 WX Amsterdam, Netherlands*

ARTICLE INFO

Communicated by M. Piccardi

Keywords:
Actor-supervision
Spatiotemporal action localization
Action understanding
Video analysis
Weakly-supervised

ABSTRACT

This paper addresses the problem of spatiotemporal localization of actions in videos. Compared to leading approaches, which all learn to localize based on carefully annotated boxes on training video frames, we adhere to a solution only requiring video class labels. We introduce an actor-supervised architecture that exploits the inherent compositionality of actions in terms of actor transformations, to localize actions. We make two contributions. First, we propose actor proposals derived from a detector for human and non-human actors intended for images, which are linked over time by Siamese similarity matching to account for actor deformations. Second, we propose an actor-based attention mechanism enabling localization from action class labels and actor proposals. It exploits a new actor pooling operation and is end-to-end trainable. Experiments on four action datasets show actor supervision is state-of-the-art for action localization from video class labels and is even competitive to some box-supervised alternatives.

1. Introduction

The goal of this paper is to localize and classify actions like *skateboarding* or *walking with dog* in video by means of its enclosing spatiotemporal tube, as depicted in Fig. 1. Empowered by action proposals (Jain et al., 2014; Weinzaepfel et al., 2015; Zhu et al., 2017), deep learning (Gkioxari and Malik, 2015; Saha et al., 2016) and carefully labeled datasets containing spatiotemporal annotations (Soomro et al., 2012; Rodriguez et al., 2008; Xu et al., 2015), progress on this challenging topic has been considerable (Kalogeiton et al., 2017; Hou et al., 2017a). However, the dependence on deep learning and spatiotemporal boxes is also hampering further progress, as annotating tubes inside video is tedious, costly and error prone (Mettes et al., 2016). We strive for action localization without the need for spatiotemporal video supervision.

Others have also considered action localization without spatiotemporal supervision (Siva and Xiang, 2011; Mettes et al., 2017; Li et al., 2018). Recently, Li et al. (2018) proposed a deep learning based model for action classification with an attention LSTM. The attention component highlights regions in the video that correspond to high-responses of certain action class labels. Unfortunately, this scheme does not ensure high-localization accuracy as the model may learn to attend only to discriminative parts of the action, such as the legs and the skateboard for the action skateboarding, but not the entire actor. Siva and Xiang (2011) and Mettes et al. (2017) circumvent this issue and aim to retrieve the entire actor by relying on human detectors, trained on images. These approaches learn a classifier using a multiple instance learning framework. This framework selects the best candidate proposal in the video guided by multiple cues, in particular the detected human actors, which is then used to learn an action classifier. These works are shallow and were not designed to exploit the representation learning principle of deep learning architectures. Our work unifies these alternatives. It infuses the pragmatic and arguably more accurate scheme of localization from detected actors into a novel end-to-end trainable deep architecture.

In this work, we introduce an actor-supervised architecture that exploits the relevance of actors to steer the localization of actions in videos without using spatiotemporal annotations of the training videos. Instead of using the detected actors to select among candidate regions *a posteriori* (Siva and Xiang, 2011; Mettes et al., 2017), we exploit the detections to define the candidate proposals *a priori*. Based on them, our architecture learns to rank the potential actor tubes from action labels at the video level. Our technical contributions are twofold. First, we introduce actor proposals; a means to generate candidate tubes that are likely to contain an action and that do not require any action video annotations for training. We derive our proposals from a detector for human and non-human actors, intended for images, combined with Siamese similarity matching to account for actor deformations over

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.cviu.2019.102886.

[*] Corresponding author.
E-mail address: victor.escorcia@kaust.edu.sa (V. Escorcia).

[1] Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

https://doi.org/10.1016/j.cviu.2019.102886
Received 19 December 2018; Received in revised form 29 July 2019; Accepted 2 December 2019
Available online 9 December 2019

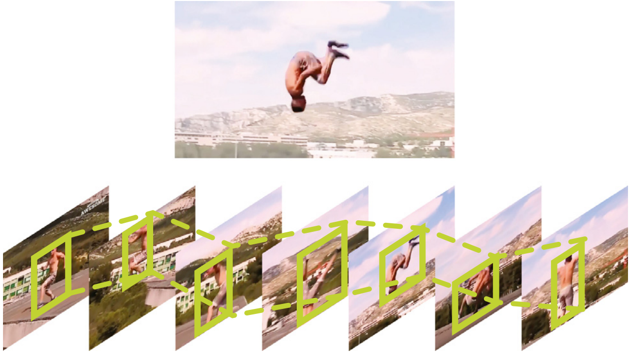

Fig. 1. We propose actor-supervision as a means for weakly-supervised action localization in video space and time. During the learning stage, our method relies on action labels at the video level only. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

time. Second, we introduce actor attention; an end-to-end architecture that selects the most suited (actor) proposals. It exploits a new differentiable operation, actor pooling, which summarizes the visual information spanned by an actor. In this way, our attention mechanism is not only derived from the action class, but it also considers the actors. Experiments on four human and non-human action datasets show that our actor proposals and actor attention register an absolute (and relative) improvements up to 8.4% (23.7%) in Recall and 10.4% (27.5%) in mAP, respectively. Taken together, our actor supervision is the state-of-the-art for action localization from video class labels and is even competitive to some box-supervised alternatives.

2. Related work

Typical approaches for action localization first generate spatiotemporal action proposals and then classify them with the appropriate action label. We discuss work related to these two aspects of action localization and group them by the amount of supervision needed.

2.1. Action proposals

Supervised action proposals generate box proposals and classify them per action for each individual frame. In addition to video-level class labels, bounding-box ground-truth for each action instance across all the frames is required. In Gkioxari and Malik (2015) and Weinzaepfel et al. (2015), the box proposals come from object proposals (Uijlings et al., 2013; Zitnick and Dollár, 2014), and a two-stream convolutional network (conv-net) is learned to classify these boxes into action classes. More recently, action boxes are generated by video extensions of modern object detectors (Liu et al., 2016; Ren et al., 2015), as in Saha et al. (2016, 2017b), Kalogeiton et al. (2017), Saha et al. (2017a), Hou et al. (2017b), Zhu et al. (2017), He et al. (2018) and Sun et al. (2018). For all these works, once the action boxes per frames are established they are linked together to create action proposals per video, for example via dynamic programming based on the Viterbi algorithm (Gkioxari and Malik, 2015).

Unsupervised action proposals do not require any class labels or bounding box ground-truth. A sliding window sampling is unsupervised but has an exponentially large search space. More efficient methods generate action proposals by grouping super-voxels based on low-level cues such as color or image gradients (Jain et al., 2014, 2017; Oneata et al., 2014). Clustering of motion trajectories is also an effective choice to hierarchically build proposals (van Gemert et al., 2015; Chen and Corso, 2015; Puscas et al., 2015).

Weakly-supervised action proposals do not rely on box-level ground-truth for all the video frames (Kläser et al., 2012; Lan et al., 2011; Tran and Yuan, 2012; Yu and Yuan, 2015). Instead, they exploit object detectors trained on images to get detections. Yu and Yuan (2015) use a human detector and motion scores to locate boxes and compute an actionness score for each of them. Linking of boxes is formulated as a maximum set convergence problem. Kläser et al. (2012) rely on an upper-body detector per frame and links them in a tube by tracking optical flow feature points. For the linking of our human and non-human actor detectors, we prefer a similarity based tracker (Tao et al., 2016; Bertinetto et al., 2016), which is more robust to deformations and can recover from loose and imprecise detection boxes.

Full supervision results in more precise boxes but scales poorly as the number of action classes grows. Unsupervised proposals are more scalable, but boxes are often less precise. Our approach achieves the best of both worlds. We obtain box precision by using an actor detector and then link the boxes from consecutive frames by Siamese similarity matching, making them robust to deformations. At the same time, our approach is action-class agnostic and hence more scalable.

2.2. Proposal classification

Supervised classification is the default in the action localization literature. Methods train classifiers using box-supervision for each action class and apply it on each action proposal for each test video, *e.g.* Gkioxari and Malik (2015), Weinzaepfel et al. (2015), Saha et al. (2016), Hou et al. (2017b), Kalogeiton et al. (2017), Saha et al. (2017a,b) and Duarte et al. (2018), Xie et al. (2018). Others, who rely on unsupervised or weakly-supervised action proposals, also train their action proposal classifiers in a supervised fashion using bounding-box ground-truth across frames (Jain et al., 2017; van Gemert et al., 2015; Chen and Corso, 2015).

Unsupervised classification has been addressed as well. Puscas et al. (2015) classify their unsupervised proposals using tube-specific and class agnostic detectors, trained via two-stage transductive learning. Soomro and Shah (2017) start with supervoxel segmentation and automatically discover action classes by discriminative clustering. It localizes actions by knapsack optimization. Jain et al. (2015), classify action proposals in a zero-shot fashion by encoding them into a semantic word embedding spanned by object classifiers. Mettes and Snoek (2017), capture actors, relevant object detections and their spatial relations in a word embedding. All the training happens on images and text, no videos are needed.

Weakly-supervised classification refrains from box-supervision for classifying action proposals. A considerable reduction in annotation effort may be achieved by replacing boxes with point annotations and unsupervised action proposals (Mettes et al., 2016), but it still demands manual labor. An alternative is to rely on human body parts (Ma et al., 2013) or human detectors trained on image benchmarks (Russakovsky et al., 2015; Lin et al., 2014) to steer the localization in video; either by defining the search space of most likely action locations, *e.g.* Siva and Xiang (2011), or by selecting the most promising action proposal (Mettes et al., 2017).

We also rely on human (and non-human) actor detectors but exploit them to generate a limited set of actor proposals. Among those, we select the best ones per action, based on an actor attention mechanism that only requires action class labels. Without the need for box annotations per video frame, we achieve results not far behind the supervised methods and much better than unsupervised methods.

3. Actor-supervision architecture

To deal with the inherent difficulty of spatiotemporal action localization without box supervision, we introduce actor supervision. We exploit the fact that actors are precursors of actions. Actions result from an actor going through certain transformations, while possibly interacting with other actors and/or objects in the process. This means that actors not only locate the action in the video, but also one can learn to rank the potential actor locations for a given action class.

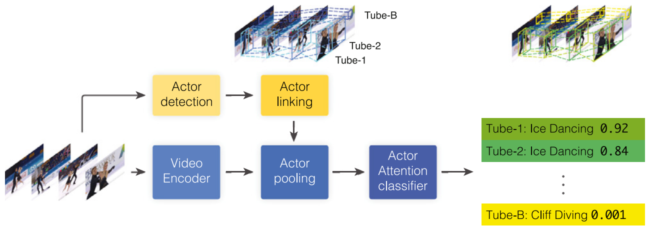

Fig. 2. Actor-supervised architecture. Blocks in yellow generate actor proposals where it is likely to find actions in the videos. Blocks in blue illustrate our action attention module which classifies the action occurring in each actor proposal and sorts them based on the relevance of the actor class. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Based on these premises, we design a novel end-to-end architecture for spatiotemporal action localization guided by actor supervision.

Fig. 2 illustrates the two pillars of our architecture, namely *actor proposals* and *actor attention*. In a nutshell, our approach enables the localization of actions with minimal supervision by (i) infusing the concept of actors in the architecture, through existing knowledge and progress in object detection and object tracking; and (ii) introducing a powerful attention mechanism suitable for learning a meaningful representation of actions. In the following subsections we disclose full details of each pillar.

3.1. Actor proposals

Our actor proposals receive a video stream and generate a set of tubes, parameterized as a sequence of boxes $\mathcal{T} = \{\mathcal{B}_i\}$. The tubes outline the most likely spatiotemporal regions where an action may occur based on the presence of an actor. It contains two modules: actor detection and actor linking, as shown in Fig. 3 and detailed next.

Actor detection. This module generates spatial locations where the actor of interest appears in the video. Respecting the requirements of our setup, this module adopts a pre-trained conv-net for object detection which predicts bounding boxes over all the frames of the video. Despite the huge progress in object detection, the predictions are still imperfect due to false positive errors or missed detections of the actor. Missed detections typically occur when the actor undergoes a significant deformation, which is common in actions. For example, when *performing a cartwheel* in floor gymnastics, the shape of the actor changes when he/she is flipping upside down. In these cases, actor deformations, characteristic of the performance of the action, may involve significant visual changes that do not fit the canonical model of the object category of the actor.

Actor linking. This stage carefully propagates the predictions of our actor detector throughout the video to generate an actor proposal tube. It complements the detector by filling the gaps left during the performance of the action, without demanding any annotation to tune the detector. In this way, our module is more robust to missed detections and consistently retrieves complete actor tubes associated with actors. We attain this goal with the aid of a robust similarity-based tracker along with a scheme to filter and select the boxes enabling detection boxes and tracker coordination. The similarity tracker exploits the temporal coherence between neighboring frames in the video, generating a box-sequence for every given box. In practice, we employ a pre-trained similarity function learned by a Siamese network which strengthens the matching of the actor between a small neighborhood in adjacent frames. Once it is learned, this similarity function is transferable and remains robust against deformations of the actor (Tao et al., 2016; Bertinetto et al., 2016). The filtering and selection scheme selects the best scoring detection boxes and sequentially feeds them to the tracker, which propagates them into box-sequences $\mathcal{B}_i$, also called tubes. This scheme also filters out the candidate detections, similar to the boxes generated by the tracker, reducing the amount of computation.

Section 4.2 describes the implementation details about the conv-net architectures used to generate our actor proposals.

3.2. Actor attention

The second pillar of our approach is responsible for assigning action labels to the actor proposals. It takes into account the visual appearance inside the actor proposals, and scores them based on action classification models trained on video-level class labels only. The outcome of this module is a set of ranked proposals where it is likely to find particular actions in the video. Fig. 2 illustrates the inner components of our actor attention which are detailed next.

Video encoder. The encoder transforms the video stream into a suitable space where our attention module can discern among different actions. In practice, we use a conv-net, which encodes video frames as response maps that also comprise spatial information. Without loss of generality, an input video with T frames and shape $T \times 3 \times W \times H$ produces a tensor of shape $T \times C \times W' \times H'$, where C is the number of response maps in the last layer of the video encoder. W' and H' correspond to scaled versions of the original width and height, respectively, due to the pooling layers or convolutions with long stride.

Actor pooling. We introduce a new pooling operation that takes as input the response maps from the video encoder and the set of actor proposals, and outputs a fixed size representation for each actor proposal. This module identifies the regions associated with each actor proposal in the response maps, and extracts a smooth representation for them. Our operation extends the bilinear interpolation layer (Jaderberg et al., 2015; Johnson et al., 2016), which operates over feature maps of images and bounding boxes, to deal with feature maps of videos and spatiotemporal tubes. Concretely, given an input feature map U of shape $T \times C \times W' \times H'$ and a set of actor proposals A of shape $T \times P \times 4$, which represents coordinates of the bounding boxes of P actor proposals of length T. We interpolate the features of U to produce an output feature map V of shape $T \times P \times C \times X \times Y$ where X, Y are the hyper-parameters representing the size of the desired output features of each actor box. For each actor box, we perform bilinear interpolation by projecting the bounding box onto the corresponding $U_{t,:,:,:}$ and computing a uniform sampling grid of size $X \times Y$, inside the actor box, associating each element of V with real-valued coordinates into U. We obtain V by convolving with a sampling kernel $k(d) = \max(0, 1 - |d|)$:

$$V_{t,p,c,i,j} = \sum_{i'=1}^{W'} \sum_{j'=1}^{H'} U_{t,c,i',j'}\, k\left(i' - x_{t,p,i,j}\right) k\left(j' - y_{t,p,i,j}\right) \qquad (1)$$

Finally, we average pool the contribution of all the output features belonging to the same actor proposal, which gives us a tensor Z of shape $P \times C \times X \times Y$ corresponding to the final output of our actor pooling.

Fig. 4 illustrates the inner details of actor pooling. Note that the bilinear kernel k ensures that only the four pixels adjacent to the point $(x_{t,p,i,j}, y_{t,p,i,j})$, which belongs to the sampling grid defined by each actor box, contributes to the final representation. Recently (Hou et al., 2017a) introduced the Tube of Interest Pooling which extends the 2D-RoI operation (Girshick, 2015) to spatiotemporal actors. In contrast, our actor pooling relies on bilinear interpolation. While the RoI operation only considers the maximum value over a bin cell, the bilinear kernel k considers four pixels for each sampling point. Thus, our layer yields a smooth representation for the actors with less sparse gradients during backpropagation. We showcase the benefits of the bilinear sampling inside a weakly-supervised experimental setup in Section 4.3.

Actor proposal classification. We classify each actor proposal according to a pre-defined set of action classes. This module learns to map the fixed size representation of each actor proposal into the space of actions of interest. In practice, we employ a fully connected layer where the number of outputs corresponds to the number of classes A. During training, the main challenge is to learn an appropriate mapping of the actor representation into the action space without the obligation of explicit annotations for each actor proposal. For this reason, we propose an attention mechanism over the actor proposals that bootstraps the action labels at the video level. In this way, we encourage the network

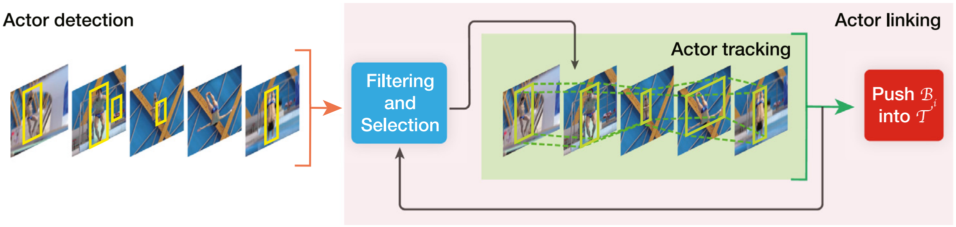

Fig. 3. Actor proposals. We generate actor proposals by detecting the most likely actor locations with the aid of an object detector. Our actor linking module selects the most relevant detections and carefully tracks them throughout the video using a Siamese similarity network, which robustly overcomes possible actor deformations. After an actor tube $\mathcal{B}_i$ is formed, we filter out detections with high similarity with the boxes of the tube. Notably, our actor linking can handle the miss detected actor boxes in the fourth frame without requiring annotations for fine tuning.

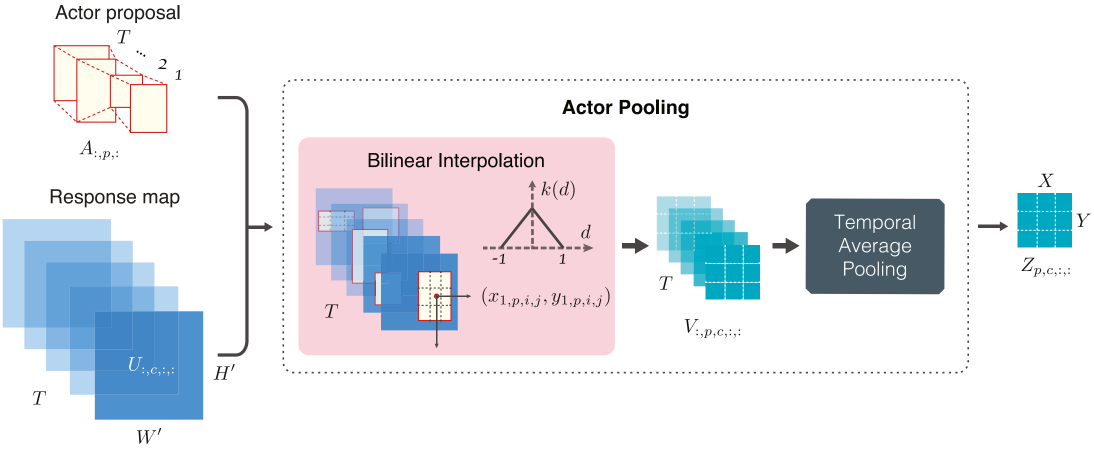

Fig. 4. Actor pooling computes a smooth fixed sized representation of each actor proposal. It crops the feature maps around each actor box and aligns a sampling grid $X \times Y$ to compute the representation via bilinear interpolation with the kernel $k(d)$. Finally, it appropriately forms the actor representation of each actor tube with a temporal average pooling. We illustrate the procedure for a single actor proposal over one slice of the feature maps.

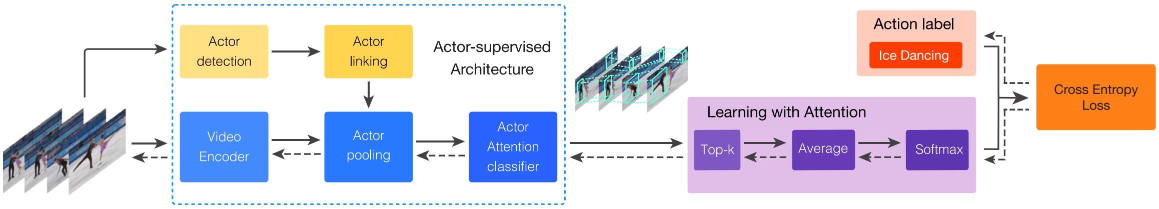

Fig. 5. Our actor-supervised architecture is the first end-to-end approach for weakly-supervised spatiotemporal action localization that adjusts the parameters of the video encoder as training progresses. The block diagram illustrates all the operations of our approach during training. The dashed lines represent the flow of data during backpropagation. Notably, we do not inquiry for any supervisory signal different than action labels at the video label to achieve the spatiotemporal localization of actions.

to learn the action classifier by focusing on the actors that contribute to an appropriate classification.

In this work, we explore the use of an attention mechanism based on top-k selection. It encourages the selection of the k most relevant actor proposals per class that contribute to perform a correct classification. In practice, we choose the top-k highest scores from the fully connected layer for each action category, and average them to form a single logit vector for each video. Subsequently, we apply a softmax activation on the logits of each video. Fig. 5 illustrates the particular instantiation described before. Note that our architecture classifies each actor independently, it could easily be extended for multi-class videos. In such case, we would change the softmax nonlinearity by a sigmoid during training.

Learning. We train our actor attention using the cross-entropy loss between the output of the softmax and the video label. It is relevant to highlight that we do not use any spatiotemporal information about the actions for learning the parameters of our model. In practice, we fit the parameters of the actor attention by employing back-propagation and stochastic gradient descent. In the case of the top-k selection module, we use a binary mask during the back-propagation representing the subgradients of the selection process.

Having defined our actor-supervised architecture, we are now ready to report experiments.

4. Experiments

4.1. Datasets and evaluation

We validate our approach on four public benchmarks for spatiotemporal action localization in videos[2].

UCF-Sports (Lan et al., 2011). This dataset consists of 150 videos from TV sport channels representing 10 action categories such as *weightlifting, diving, golf-swing,*. We employ the evaluation protocol established by Lan et al. (2011), but without using the box annotations in the training set.

[2] Datasets used in this paper were downloaded and experimented on by primary author

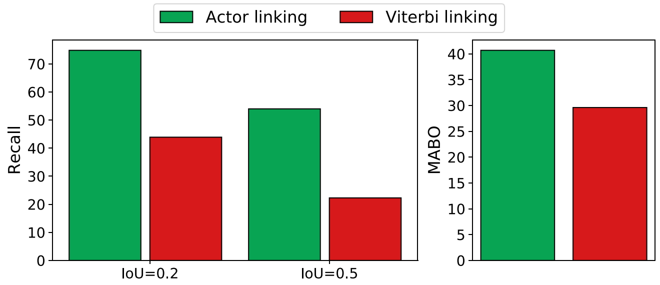

Fig. 6. Actor linking outperforms the Viterbi linking algorithm by Gkioxari and Gkioxari and Malik (2015), used to connect sparse detection in time. We attribute its success to the use of similarity-based matching to handle deformations of the actor that the Viterbi linking is unable to fix. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

JHMDB (Jhuang et al., 2013). This dataset showcases 21 action categories such as *push-up, shooting, etc.*; and consists of 928 videos from movies and consumer videos from internet portals. Unless stated otherwise, we employ the standard evaluation protocol using three splits. We refrain from using the box annotations in the training set to tune our model.

THUMOS13 (Soomro et al., 2012). This dataset incorporates untrimmed videos and multiple action instances per video. It consists of a subset of 3294 videos derived from UCF101 featuring 24 action categories. We use the training and testing partition from split 1 of UCF101 for evaluating our approach. Note that we do not rely on the spatiotemporal box annotations of the training set.

A2D (Xu et al., 2015). The actor–action dataset comprises 3782 videos from Youtube designed to model the relationship between actors and actions in videos. This dataset considers actions, such as *flying, jumping, climbing, etc.,* as performed by various actors, such as *ball, cat, baby, etc.* Again, we do not use the spatiotemporal annotations of the training set.

Evaluation. Following the standard protocol for action localization, we report the intersection over union (IoU) to measure the degree to which a candidate tube is associated with a given spatiotemporal action ground-truth annotation. Depending on the task and dataset of interest, we report the result in terms of Recall or mean Average Precision (mAP). To evaluate action classification performance, we employ the evaluation setup typical for action localization using mean average precision (mAP) over all available classes, given an overlap threshold of 0.2 (THUMOS13) and 0.5 (UCF-Sports, JHMDB).

4.2. Implementation details

Actor proposals. We use a single-shot multi-box detector (Liu et al., 2016) to detect an actor of interest in every frame. We used a public pretrained detector trained on all MS-COCO categories (Huang et al., 2017; Lin et al., 2014) and limit the detections used to the actors of interest according to the action categories defined in each dataset. The base network of the actor detector is InceptionV2 (Ioffe and Szegedy, 2015) – up to $inception(5b)$ pre-trained on ILSVCR-12 (Russakovsky et al., 2015) – followed by six feature layers and box predictors (Liu et al., 2016; Huang et al., 2017). We only track the detections selected by our actor linking forward-and-backward over the entire video using a multi-scale fully-convolutional Siamese-tracker (Bertinetto et al., 2016) trained on the ALOV dataset (Smeulders et al., 2014). The base network of the tracker corresponds to the first four convolutional blocks of VGG-16, followed by a cross-correlation layer (as in Bertinetto et al. (2016)) operating over 4 scales. The actor selector ignores detections predicted by the actor detector greedily when those have a high spatial affinity with boxes generated by the tracker. In practice, we use an overlap threshold of 0.7.

Actor attention. We only consider the RGB stream to encode the visual appearance of the videos. Our video encoder corresponds to the convolutional stages of VGG-16, for fair comparison with previous work (Li et al., 2018), pre-trained on ILSVCR-12 (Russakovsky et al., 2015). The grid size for the bilinear interpolation of actor pooling is 3×3. During training, our attention module focuses on the $k = 3$ most relevant actors out of 10 actor tubes for classifying the video. We train our entire actor attention module end-to-end from RGB streams to video labels, adjusting the parameters of the visual encoder as training progresses, as opposed to using pre-computed features like (Li et al., 2018). Due to memory constraints, we employ segment partition introduced by Wang et al. (2016) to allocate more than one video per batch. For each video, we analyze 16 equally spaced frames each time. We set the learning rate to 1e-2 and employ a momentum factor of 0.99 to train our model in a single GPU with a batch size of four videos. At test time, we used at most ten actor-proposals per video and remove tubes with overlap greater to 0.6 via NMS.

4.3. Results

Actor linking versus Viterbi linking. In our first experiment we validate the relevance of our actor linking with respect to the more traditional Viterbi linking (Gkioxari and Malik, 2015) for the generation of actor proposals from the predictions of our actor detector. As shown in Fig. 6 our actor linking achieves an improvement in Recall of +19.9% and +21.6% for 0.2 and 0.5 IoU in THUMOS13. We attribute these results to the capability of the similarity-based matching to accommodate for deformation of the actor, that the Viterbi linking approach is unable to fix. Previous approaches *e.g.* Gkioxari and Malik (2015), Kläser et al. (2012) and Saha et al. (2016) employ supervision at the level of boxes and length of the tubes to overcome this issue. This clearly limits their application under the weakly-supervised setup evaluated in this work. We conclude that actor linking, by similarity-based matching, aids spatiotemporal action localization with weak supervision.

Actor proposals versus others. Table 1 compares our actor proposals with previous supervised and unsupervised action proposals. Compared to the action proposals by Yu and Yuan (2015), our approach achieves an improvement of +34.2% in terms of Recall on THUMOS13. This result evidences the benefit of our actor detection and linking scheme. Interestingly, our approach improves upon previous unsupervised work by +1.7% and +8.4% in terms of Recall on UCF-Sports and THUMOS13, respectively. These methods (Jain et al., 2017; van Gemert et al., 2015) are based on grouping techniques over low-level primitives such as color and motion, which reaffirms our intuition about the relevance of actors as a strong semantic cue for the localization of the actions. Fig. 7 illustrates the recall of our actors proposals for a varying number of proposal in comparison with previous unsupervised approaches. It provides further evidence on the quality of our proposals, especially when considering only a limited number of proposals.

The state of the art for action proposals generation (Zhu et al., 2017; Weinzaepfel et al., 2015) are fully supervised approaches based on a mix of convolutional-recurrent networks and supervised instance level tracking, respectively. Our approach achieves competitive results or even outperforms them in JHMDB-split1, following the evaluation protocol of Weinzaepfel et al. (2015), without relying on additional action box supervision. Although supervised approaches offer proposals with better quality in two out of the three datasets studied, they do so at the expense of extra annotations. It that sense, these methods have a limited scalability potential, and it opens up a spot for our actor proposals.

Non-human actor proposals. We also analyze the quality of our proposals for generating spatiotemporal tubes for non-human actors. For this experiment we assume that the class of the actors are known at test time and evaluate the quality of the localization accordingly. Fig. 8

Table 1
Action proposal comparison in terms of Recall. Weinzaepfel et al. (2015) and Zhu et al. (2017) use video supervision from action boxes and action labels, while the rest do not use any video supervision. Our actor proposals achieve better Recall compared to previous unsupervised and weakly-supervised methods.

	UCF-Sports	THUMOS13		JHMDB
	IoU = 0.5	IoU = 0.5	IoU = 0.1	IoU = 0.5
Weinzaepfel et al. (2015)	98.8	–	–	94.6
Zhu et al. (2017)	96.8	61.4	–	–
Yu and Yuan (2015)	–	–	54.5	–
van Gemert et al. (2015)	89.4	35.5	–	–
Jain et al. (2017)	91.9	32.8	–	–
Our work	**93.6**	**43.9**	**88.7**	**97.4**

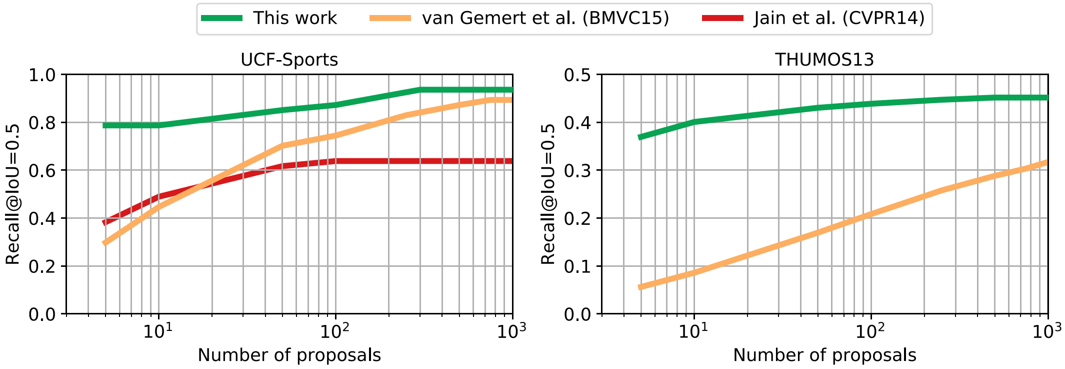

Fig. 7. Actor proposals outperform unsupervised action proposals. We attribute its success to the use of actors as semantic cue relevant for the grounding of actions. Notably, we retrieve relevant action tubes from a much smaller pool, which is advantageous in the context of retrieval and spatiotemporal localization of actions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

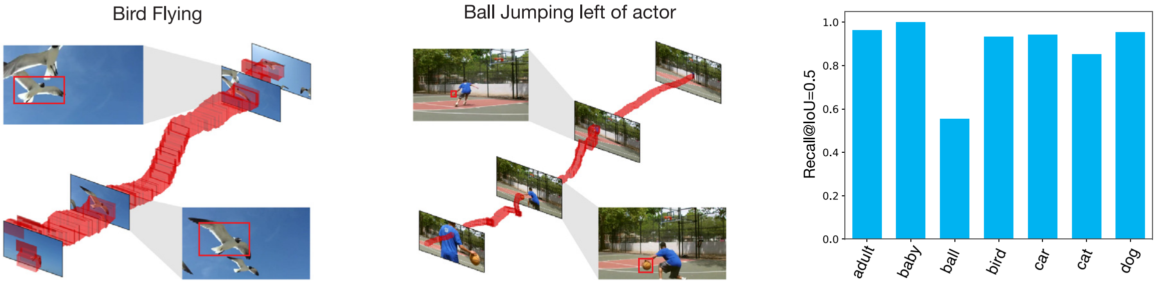

Fig. 8. Non-human actor proposals on A2D. Qualitative visualizations of action proposals for *Bird* and *Ball*. The recalls at IoU = 0.5 are consistently high for all the actor classes except for *Ball*, which is understandable due to its common shape and small size, which invite many occlusions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

summarizes our findings including two qualitative results taken from the A2D dataset. The leftmost example shows we are able to generate proposals for highly articulated actors like *birds*. The example in the center exemplifies a common failure case for the *ball* actor. In this case, the ball changes significantly in appearance during the execution of the action, including several full occlusions caused by the human. From our quantitative analysis, we appreciate that the actor with the highest recall is *baby*, which is not directly represented in the training set of our actor detector. The analysis also reveals that the gap in recall at 0.5 IoU between adult actors and animal actors like *bird* and *dog* is at most +2%. Except for *ball*, the recall for all the actors using at most 50 actor proposals is greater than 85%. Therefore, we conclude that our method is general and applicable to both human and non-human actors.

Impact of actor attention. We validate the benefits of our actor attention stream, and ablate the operations of our novel actor pooling layer.

How to pool the spatial information of the actors? Table 2(a) compares the proposed bilinear interpolation vs. the RoI pooling strategy proposed by Hou et al. (2017b), by replacing the former with the latter in our actor pooling module. The result shows an improvement of +7.9% in mAP when bilinear interpolation is used. This could be a consequence of the smoother representation with less sparse gradients mentioned in Section 3.2, while training in a regime without box level supervision.

How to aggregate the temporal information of the actor? Table 2(b) compares different types of temporal aggregation function of the actor representation after bilinear interpolation. We observed that *Early pooling*, averaging the temporal information of V after the bilinear sampling, performs the best. Followed closely by *Late pooling*, computing the likelihood of each box in the actor tube independently and average. In contrast, *No pooling*, using fixed size representation of a group of frames V, and *3D-conv*, a 3D conv kernel followed by early pooling, do not perform as well in THUMOS13. This result could be a consequence of some degree of overfitting as coarse level information is enough to classify the actions of interest and modeling precise details of the actors is not necessary in the current benchmarks. We used early pooling for the rest of the experiments in the paper.

Benefit of end-to-end training. Finally, we showcase the impact of fine-tuning the video encoder in an end-to-end fashion, as training

Table 2
Actor pooling ablations and the benefit of our end-to-end actor attention stream in THUMOS13. We ablate the spatial and temporal operations of our actor pooling layer. (a) Bilinear interpolation of the actor information adds 7.9% compared to RoI as done in Hou et al. (2017b). Similarly, (b) early pooling the temporal information of each actor performs the best among other types of temporal pooling. Finally, (c) Fine-tuning the video encoder adds 4.4% in mAP as opposed to a fixed visual representation as done in Li et al. (2018).

(a)		(b)		(c)	
Spatial pooling ablation	mAP@0.2	Temporal pooling ablation	mAP@0.2	Fine-tune video encoder	mAP@0.2
RoI as Hou et al. (2017a)	37.9	3D-conv	37.9		41.4
Bilinear Interp, *Our paper*	**45.8**	No pooling, V	42.68	✓	**45.8**
		Late pooling	44.7		
		Early pooling, Z	**45.8**		

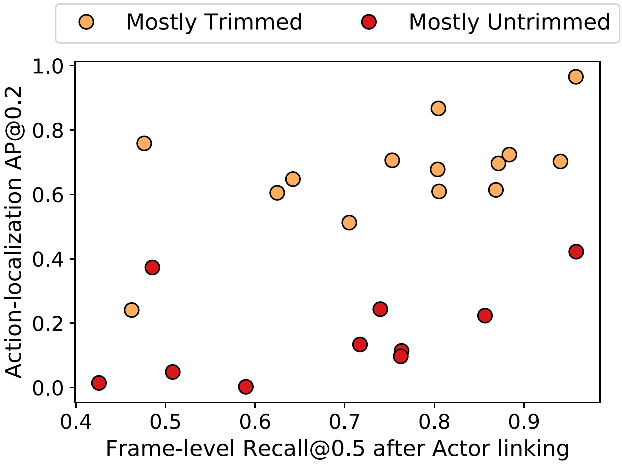

Fig. 9. Per-class correlation between action localization, AP at IoU = 0.2, and actor linking (detection + tracking) frame-level performance, Recall IoU = 0.5. Among all action classes AP is not correlated to the frame-level recall. We noticed that for two groups: (i) action classes that are relatively untrimmed, presented in red, and (ii) trimmed classes, presented in yellow; better performance in frame-level detection translates into better action localization performance. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

progresses, in Table 2(c) on THUMOS13. When the video encoder is fine-tuned we achieve an improvement of 4.4% in mAP, compared to when employing a fixed visual representation, as done in Li et al. (2018).

Impact of detection and tracking on action localization. Fig. 9 shows per-class correlation between action localization AP (at IoU = 0.2) and actor linking (detection + tracking) frame-level performance, Recall IoU = 0.5. We noticed that among all action classes AP is not correlated to the frame-level recall. We observe two clusters of classes, one mostly containing the untrimmed classes (in red) and the other mostly containing trimmed classes (in yellow). For each cluster, the action localization performance is directly proportional to that of actor detection and tracking.

Actor-supervision versus others. Table 3 compares multiple approaches with a varying degree of supervision. Based on these results, we note that actor supervision achieves the state-of-the-art among all approaches that localize action from an action class label only. It improves upon (Li et al., 2018) by +8.1% and upon (Mettes et al., 2017) by +10.4% on the THUMOS13 and UCF-Sports benchmarks, respectively. Compared to Mettes et al. (2017), our approach gives more relevance to the actors during the localization stage instead of using them as cues to improve the ranking of existing action proposals. We hypothesize that our attention mechanism is more effective than the one in Li et al. (2018), because actors are a more powerful cue for guiding the localization of actions than individual pixels. Table 4 illustrates the results of our approach for a broad range of IOU thresholds in contrast with the state of the art approach for weakly-supervised action localization of Mettes et al. (2017). It is evident that our architecture achieves better localization accuracy, especially on higher IOU values. Interestingly, actor-supervision outperforms this multiple instance learning approach in the most challenging dataset, THUMOS13, over all the IOU values reported.

Actor-supervision also outperforms several approaches with varying levels of supervision on the challenging THUMOS13 benchmark (Yu and Yuan, 2015; van Gemert et al., 2015; Mettes et al., 2016). Compared to Weinzaepfel et al. (2015) and Hou et al. (2017b), our visual representation is limited to the RGB video stream. We suspect that flow information can boost the action classification and localization results as earlier shown by Carreira and Zisserman (2017), Gkioxari and Malik (2015) and Saha et al. (2016). Similarly, we compare actor-supervision on JHMDB with Gkioxari and Malik (2015), using their RGB stream only, and observe comparable results (35.8% ± 2.7 versus 37.9%). This proves the relevance of our approach in another challenging benchmark against a strong competitor aided by box-supervision during training.

The state-of-the-art in action localization is dominated by fully-supervised approaches resembling conv-net architectures well established for generic object detection (Kalogeiton et al., 2017; Saha et al., 2017b), which require strong levels of supervision. These approaches are unable to be trained in the weakly supervised regime presented in this paper. Interestingly, our approach not only outperforms other weakly-supervised methods but it also has an edge over some of the supervised approaches. Considering the poor scalability of fully-supervised approaches and the tremendous amount of progress in object detection, we envision that our work can inspire the community to seek other forms of supervision during the design and adaption of deep representations for localizing actions in videos. Figs. 10–11 illustrate action localization results of our approach on the THUMOS13 benchmark. The latter showcases that our method can deal with videos exhibiting multiple actors such as Ice Dancing. We also corroborate that the mAP of actions exhibiting multiple actors is similar to those of actions with a single actor.

4.4. Discussion

Our tubes spans the entire videos as most datasets for spatiotemporal localization are trimmed or nearly trimmed. Only THUMOS13 contains action tubes that does not expand the entire video. Yet THUMOS13 is nearly trimmed; as the instance temporal coverage, defined as the ratio of the length of an action tube by the length of the video, is 74% on average, and only five out of twenty-four classes have an average coverage lower than 50%. We observed that the AP@IoU = 0.2 in those five classes is 0.25x lower than the other classes. Interestingly, concurrent work on weakly-supervised temporal localization has emerged (Nguyen et al., 2018; Paul et al., 2018) without localizing the actors as presented here. Thus, our work is orthogonal and we envision follow-up work merging both approaches, accompanied by new datasets where the untrimmed aspect of spatiotemporal action instances is widespread.

Table 3
Comparison of spatiotemporal action localization approaches with decreasing amount of supervision. The top half shows supervised approaches, whereas the bottom half shows weakly-supervised approaches relying on action class labels only. Actor-supervision achieves state-of-the-art performance among the weakly-supervised approaches and sometimes even outperforms supervised methods.

	Action supervision			THUMOS13	UCF-Sports	JHMDB
	Boxes	Points	Labels	mAP@0.2	mAP@0.5	mAP@0.5
Kalogeiton et al. (2017)	✓		✓	77.2	92.7	73.7
Saha et al. (2017b)	✓		✓	73.5	–	72
Hou et al. (2017b)	✓		✓	47.1	86.7	76.9
Jain et al. (2017)	✓		✓	48.1	–	–
Weinzaepfel et al. (2015)	✓		✓	46.8	90.5	60.7 ± 2.7
Gkioxari and Malik (2015)	✓		✓	–	75.8	37.9[a]
van Gemert et al. (2015)	✓		✓	37.8	–	–
Yu and Yuan (2015)	✓		✓	26.5	–	–
Mettes et al. (2016)		✓	✓	34.8	–	–
Mettes et al. (2017)			✓	37.4	37.8	–
Li et al. (2018)			✓	36.9	–	–
Cinbis et al. (2014) (from Mettes et al. (2016))			✓	13.6	–	–
Sharma et al. (2015) (from Li et al. (2018))			✓	5.5[a]	–	–
Our paper			✓	**45.8**[a]	**48.2**[a]	35.8 ± 2.7[a]

[a] Denotes use of RGB frames solely, as we do.

Table 4
Localization accuracy for multiple IoU. Actor-supervision outperforms the state-of-the-art weakly-supervised approach of Mettes et al. (2017) for a broad range of stringent IoU thresholds.

IOU thresholds		0.1	0.2	0.3	0.4	0.5	0.6
UCF-sports	Mettes et al. (2017)	**87.7**	81.7	64.4	54.5	37.8	17.5
	This paper	84.2	**84.2**	**84.2**	**64.9**	**48.2**	**40.6**
THUMOS13	Mettes et al. (2017)	49.8	37.4	25.8	13.7	6.2	1.3
	This paper	**53.4**	**45.8**	**38.0**	**30.7**	**19.3**	**6.2**

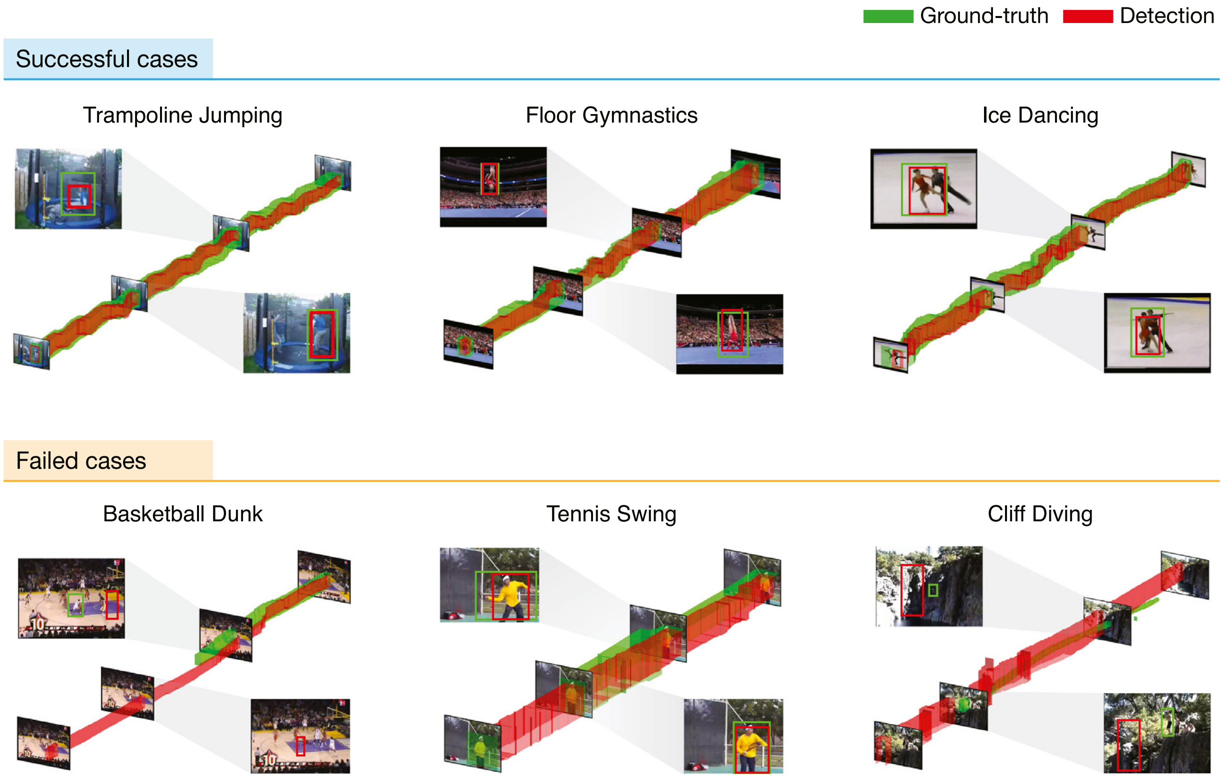

Fig. 10. Qualitative results on the THUMOS13 dataset. Top row shows three successful cases by visualizing the ground-truth and action tubes as well as two highlighted frames. These include action sequences that have deformations of actor as well as multiple actors with complex background. Bottom row visualizes three failed cases which show that crowded background, occlusions and temporally untrimmed action sequences are the most challenging scenarios. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

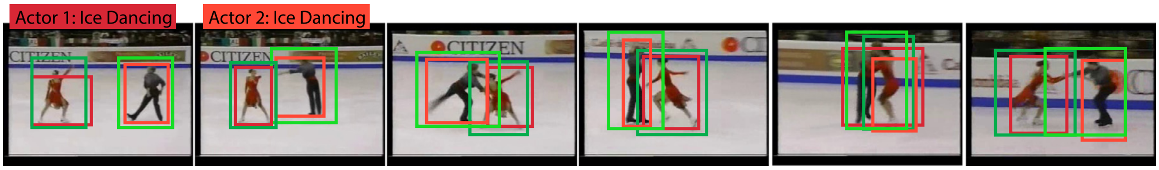

Fig. 11. Qualitative results with multiple actors, ground-truth tubes in green. Our approach can detect multiple actors and associate an action label for each of them. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Conclusion

This paper introduces a weakly-supervised approach for the spatiotemporal localization of actions, driven by actor supervision. We show that exploiting the inherent compositionality of actions, in terms of transformations of *actors*, disregards the dependence on spatiotemporal annotations of the training videos. In the proposal generation step, we introduce actor supervision in the form of an actor detector and similarity-based matching to locate the action in the video as a set of actor proposals. Then, our proposed actor attention learns to classify and rank these actor proposals for a given action class. This step also does not require any per frame box-level annotations. Instead, we design an attention based mechanism that chooses the most relevant actor proposal only for class labels at the video level. Moreover, we introduce a novel actor pooling operation that summarizes the representation of each actor in a more effective way than recent pooling strategies for the weakly-supervised setup of our interest. Our approach outperforms the state-of-the-art among weakly-supervised works and even achieves results that are better or competitive to some of the fully-supervised methods. In future work, we envision clever redesigns of our actor supervised approach to attain further improvement in the spatiotemporal localization of actions without action box annotations.

Acknowledgments

This publication is based upon work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2017-3405. We thank the team members of the IVUL from KAUST and Qualcomm AI Research for helpful comments and discussion. In particular, we appreciate the support of Amirhossein Habibian during the implementation of our Actor Linking.

Appendix

We complement the manuscript with the following items:

- A video summarizing our work. It showcases more qualitative results generated by our actor-supervised architecture. There you can appreciate that our approach works well in videos with multiple actors, involving considerable deformations and for non-human actors.
  Video (high-quality): http://bit.ly/2CkaIf6
  Video (compressed): http://bit.ly/2CkxIL2
  The codec of the video is H264-MPEG-4 AVC, its resolution is 1280 × 720 and the frame rate 60. We recommend using the VLC media player https://www.videolan.org/vlc/index.html.
- In Appendix A.1, we describe the inner details of our actor proposal algorithm and comment about its computational complexity.
- Appendix A.2 gives more details about the training of our actor-supervised architecture concerning initialization and training policy, details about the backpropagation, and robustness of our actor pooling operation.

A.1. Actor proposals

The Algorithm 1 describes all the interactions between the inner blocks involved for the generation of our actor proposals, described in the main paper.

Algorithm 1 Actor proposals generation

```
Input: maximum number of proposals $N$
Output: $\mathcal{T}$
$\mathcal{D} \leftarrow$ run actor detector over all frames
$\mathcal{T} \leftarrow \emptyset$
$i \leftarrow 0$
while $\mathcal{D} \neq \emptyset \wedge i < N$ do
    $b_i \leftarrow$ select actor with highest score from $\mathcal{D}$
    $\mathcal{B}_i \leftarrow$ actor tracker tracks $b_i$ forward and backward throughout the video
    Push $\mathcal{B}_i$ onto $\mathcal{T}$
    $\mathcal{D} \leftarrow$ filter actors in $\mathcal{D}$ with high similarity with boxes in $\mathcal{B}_i$
    $i \leftarrow i + 1$
```

Regarding the ***time complexity*** of our approach, our object detector and Siamese-tracker run at 32 and 60 FPS, respectively, on an Intel-Xeon E5-2687 with a GTX 1080. After direct contact with Jain et al. (2017), we establish that our method generates less number of proposals (100), is more accurate (+10% Recall at 0.5 IOU) and 7.91 times faster on THUMOS13. In conclusion, our action proposal approach generates more precise candidates and does so in less time.

A.2. Actor-supervised architecture

Backpropagation details. Fig. 5 illustrates the end-to-end characteristic of the actor attention stream of our Actor-supervised architecture. The modules involved during the training of our actor attention. These are (i) the video encoder; (ii) our actor pooling; (iii) the actor attention classifier; and (iv) the attention mechanism. The latter is composed by a top-k selection operational average among top-k scores per-class; and a softmax activation per video. This diagram also highlights the capability to be trained in a weakly-supervised setup, only from class labels at the video level. We can note that gradients flow backwards down to the raw video updating the parameters of our video encoder. This contrasts with the strategy of Li et al. (2018) that trains a recurrent module on top of pre-computed features from the last convolutional block of VGG16. In that sense, our work is the first deep architecture for weakly-supervised spatiotemporal action localization in videos trained from raw visual information.

On the other hand, it is relevant to mention that we did not employ any supervisory signal nor perform any updates on the parameters of our actor proposal. Note that the use of any additional supervisory signal to tune our actor proposals module, top stream in Fig. 5, goes against the weakly-supervised setup of our interest.

Training details of our actor-attention stream. We initialize the weights of our actor classifier module with Xavier technique (Glorot and Bengio, 2010), and our video encoder with the weights from a VGG-16 model pre-trained on Imagenet-ILSVCR-2012 (Russakovsky

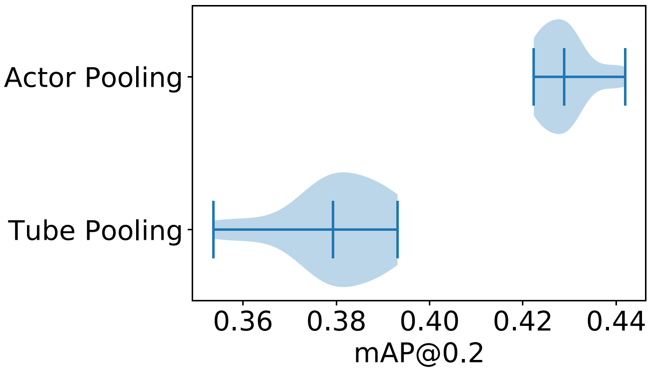

Fig. 12. The violin plot clearly shows that using actor pooling in regards of the tube pooling operation (Hou et al., 2017b) results in better localization performance. The middle line in each violin represents the median mAP and its shape is determined by the density function of the sample points in the experiment. The experiment was carried out in THUMOS13 using the training–testing partition from the split 2 of UCF101. More details of the experiment are provided in the text.

et al., 2015). Our actor attention stream is trained for 20 epochs annealing the learning rate by a factor of 0.75 after eleven epochs. We use a momentum factor of 0.99 and an initial learning rate of 0.01. As input pre-processing, we employ the segment based strategy suggested by Wang et al. (2016). In our case, we randomly sample 16 frames uniformly spaced per video. Additionally, we apply a random horizontal flipping of all the sampled frames. Finally, we normalize the input frames such that the intensity values lie on the range between $[-0.5, 0.5]$ using standard scaling with mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$ for the RGB channel.

***Robustness of actor pooling*.** As we mentioned in the main paper, the use of our actor pooling yields better results than the tube pooling operation (Hou et al., 2017b) in the weakly-supervised setup of our interest. Fig. 12 compares the robustness of our actor pooling operation in regards to the tube pooling operation. The violin plot, Fig. 12, summarizes the statistics of the top-15 results among forty experiments for each trial *i.e.* our actor supervised architecture using either actor pooling or tube pooling. One experiment constitutes a variation of the grid size (3×3 or 5×5), or the optimization hyper-parameters, such as learning rate, momentum, *etc.*, while maintaining the other hyper-parameters of our actor supervised architecture intact. In this manner, we can compare both operations beyond a single hyper-parameter configuration without introducing other confounding variables. For this experiment, we use the THUMOS13 dataset employing the training–testing partition from the split 2 of UCF101 (Soomro et al., 2012), ensuring that we do not overfit on the standard partition used for comparing different methods.

We verify that our actor pooling statistically improves upon the tube pooling operation with a p-value of 1%. Thus, we reaffirm that our actor pooling consistently achieves better localization performance than tube pooling beyond a single hyper-parameter configuration.

References

Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S., 2016. Fully-convolutional siamese networks for object tracking. In: ECCV Workshops.

Carreira, J., Zisserman, A., 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR.

Chen, W., Corso, J., 2015. Action detection by implicit intentional motion clustering. In: ICCV.

Cinbis, R.G., Verbeek, J., Schmid, C., 2014. Multi-fold MIL training for weakly supervised object localization. In: CVPR.

Duarte, K., Rawat, Y., Shah, M., 2018. VideoCapsulenet: A simplified network for action detection. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), NIPS. pp. 7610–7619.

van Gemert, J., Jain, M., Gati, E., Snoek, C., 2015. APT: Action localization proposals from dense trajectories. In: BMVC.

Girshick, R., 2015. Fast r-CNN. In: ICCV.

Gkioxari, G., Malik, J., 2015. Finding action tubes. In: CVPR.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Vol. 9. PMLR, pp. 249–256, URL http://proceedings.mlr.press/v9/glorot10a.html.

He, J., Ibrahim, M.S., Deng, Z., Mori, G., 2018. Generic tubelet proposals for action localization. In: WACV.

Hou, R., Chen, C., Shah, M., 2017a. Tube convolutional neural network (T-CNN) for action detection in videos. In: ICCV.

Hou, R., Chen, C., Shah, M., 2017b. Tube convolutional neural network (T-CNN) for action detection in videos. In: ICCV. pp. 5823–5832. http://dx.doi.org/10.1109/ICCV.2017.620.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In: CVPR.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML.

Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K., 2015. Spatial transformer networks. In: NIPS.

Jain, M., van Gemert, J., Jégou, H., Bouthemy, P., Snoek, C., 2014. Action localization with tubelets from motion. In: CVPR.

Jain, M., van Gemert, J., Jégou, H., Bouthemy, P., Snoek, C., 2017. Tubelets: Unsupervised action proposals from spatiotemporal super-voxels. IJCV.

Jain, M., van Gemert, J., Mensink, T., Snoek, C., 2015. Objects2action: Classifying and localizing actions without any video example. In: ICCV.

Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J., 2013. Towards understanding action recognition. In: ICCV.

Johnson, J., Karpathy, A., Fei-Fei, L., 2016. DenseCap: Fully convolutional localization networks for dense Captioning. In: CVPR.

Kalogeiton, V., Weinzaepfel, P., Ferrari, V., Schmid, C., 2017. Action tubelet detector for spatio-temporal action localization. In: ICCV.

Kläser, A., Marszałek, M., Schmid, C., Zisserman, A., 2012. Human focused action localization in video. In: Trends and Topics in Computer Vision.

Lan, T., Wang, Y., Mori, G., 2011. Discriminative figure-centric models for joint action localization and recognition. In: ICCV.

Li, Z., Gavrilyuk, K., Gavves, E., Jain, M., Snoek, C., 2018. Videolstm convolves, attends and flows for action recognition. CVIU.

Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: common objects in context. In: ECCV.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A., 2016. SSD: Single shot multibox detector. In: ECCV.

Ma, S., Zhang, J., Ikizler-Cinbis, N., Sclaroff, S., 2013. Action recognition and localization by hierarchical space-time segments. In: ICCV.

Mettes, P., van Gemert, J., Snoek, C., 2016. Spot on: Action localization from pointly-supervised proposals. In: ECCV.

Mettes, P., Snoek, C., 2017. Spatial-aware object embeddings for zero-shot localization and classification of actions. In: ICCV.

Mettes, P., Snoek, C., Chang, S., 2017. Localizing actions from video labels and pseudo-annotations. In: BMVC.

Nguyen, P., Liu, T., Prasad, G., Han, B., 2018. Weakly supervised action localization by sparse temporal pooling network. In: CVPR.

Oneata, D., Revaud, J., Verbeek, J., Schmid, C., 2014. Spatio-temporal object detection proposals. In: ECCV.

Paul, S., Roy, S., Roy-Chowdhury, A.K., 2018. W-TALC: weakly-supervised temporal activity localization and classification. In: ECCV.

Puscas, M., Sangineto, E., Culibrk, D., Sebe, N., 2015. Unsupervised tube extraction using transductive learning and dense trajectories. In: ICCV.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS.

Rodriguez, M.D., Ahmed, J., Shah, M., 2008. Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In: CVPR.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F., 2015. Imagenet large scale visual recognition challenge. IJCV.

Saha, S., Singh, G., Cuzzolin, F., 2017a. Amtnet: Action-micro-tube regression by end-to-end trainable deep architecture. In: ICCV.

Saha, S., Singh, G., Sapienza, M., Torr, P., Cuzzolin, F., 2016. Deep learning for detecting multiple space-time action tubes in videos. In: BMVC.

Saha, S., Singh, G., Sapienza, M., Torr, P., Cuzzolin, F., 2017b. Online real-time multiple spatiotemporal action localisation and prediction. In: ICCV.

Sharma, S., Kiros, R., Salakhutdinov, R., 2015. Action recognition using visual attention. In: NIPS Workshop.

Siva, P., Xiang, T., 2011. Weakly supervised action detection. In: BMVC.

Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M., 2014. Visual tracking: An experimental survey. TPAMI 36 (7), 1442–1468. http://dx.doi.org/10.1109/TPAMI.2013.230.

Soomro, K., Shah, M., 2017. Unsupervised action discovery and localization in videos. In: ICCV.

Soomro, K., Zamir, A.R., Shah, M., 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR, URL http://arxiv.org/abs/1212.0402.

Sun, C., Shrivastava, A., Vondrick, C., Murphy, K., Sukthankar, R., Schmid, C., 2018. Actor-centric relation network. In: ECCV.

Tao, R., Gavves, E., Smeulders, A.W., 2016. Siamese instance search for tracking. In: CVPR.

Tran, D., Yuan, J., 2012. Max-margin structured output regression for spatio-temporal action localization. In: NIPS.

Uijlings, J., van de Sande, K., Gevers, T., Smeulders, A., 2013. Selective search for object recognition. IJCV.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L., 2016. Temporal segment networks: Towards good practices for deep action recognition. In: ECCV.

Weinzaepfel, P., Harchaoui, Z., Schmid, C., 2015. Learning to track for spatio-temporal action localization. In: ICCV.

Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K., 2018. Rethinking spatiotemporal feature learning for video understanding. In: ECCV.

Xu, C., Hsieh, S.-H., Xiong, C., Corso, J., 2015. Can humans fly? Action understanding with multiple classes of actors. In: CVPR.

Yu, G., Yuan, J., 2015. Fast action proposals for human action detection and search. In: CVPR.

Zhu, H., Vial, R., Lu, S., 2017. TORNADO: A spatio-temporal convolutional regression network for video action proposal. In: ICCV.

Zitnick, L., Dollár, P., 2014. Edge boxes: Locating object proposals from edges. In: ECCV.