Learning Actionness via Long-range Temporal Order Verification

Dimitri Zhukov[1,2], Jean-Baptiste Alayrac[3], Ivan Laptev[1,2], and Josef Sivic[1,2,4]

[1] Département d'informatique de l'ENS, ENS, CNRS, PSL University, Paris, France
[2] INRIA, Paris, France
[3] DeepMind
[4] CIIRC, Prague, Czech Republic

Abstract. Current methods for action recognition typically rely on supervision provided by manual labeling. Such methods, however, do not scale well given the high burden of manual video annotation and a very large number of possible actions. The annotation is particularly difficult for temporal action localization where large parts of the video present no action, or *background*. To address these challenges, we here propose a self-supervised and generic method to isolate actions from their background. We build on the observation that actions often follow a particular temporal order and, hence, can be predicted by other actions in the same video. As consecutive actions might be separated by minutes, differently to prior work on the arrow of time, we here exploit long-range temporal relations in 10-20 minutes long videos. To this end, we propose a new model that learns actionness via a self-supervised proxy task of order verification. The model assigns high actionness scores to clips which order is easy to predict from other clips in the video. To obtain a powerful and action-agnostic model, we train it on the large-scale unlabeled HowTo100M dataset with highly diverse actions from instructional videos. We validate our method on the task of action localization and demonstrate consistent improvements when combined with other recent weakly-supervised methods.

Keywords: temporal order, action localization, video recognition.

1 Introduction

Learning from web videos is becoming increasingly popular in computer vision as such videos are available in large quantities, and cover diverse activities and scenes. In particular, instructional videos have been recently explored as a rich source for many tasks and goal-driven sequences of actions [2, 13, 23, 31, 40, 41]. While the quantity and diversity of video data appears crucial for training current recognition models [4, 22, 23], the manual annotation of actions in large-scale video data requires large efforts [4] and may not scale well to the large number of possible actions. This is particularly true for the task of temporal action localization where sparse actions should be isolated in video streams from the large portion of "background" with no actions. For example, action frames in

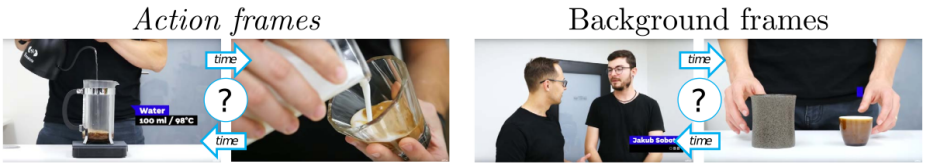


Fig. 1. We show pairs of action frames and background frames from the same video. Can you predict the order of frames within each pair? While the order is relatively easy to guess for actions, the same task is more difficult for the background. We use this observation and exploit the predictability of temporal order as a measure of actionness[‖]

the CrossTask dataset [41] with typical instructional videos represent only 25.9% of the total video length.

In our work we address the above challenges and aim to develop a self-supervised approach for separating a large and diverse set of actions from their background. We observe that actions typically do not happen in isolation and are often surrounded by other related actions. Moreover, action sequences often demonstrate a consistent order (taking off a car wheel should be preceded by lifting the car), hence, many actions can be identified by the predictability of their order with respect to other actions in the same video. On the contrary, the order of background frames is often hard to predict, hence, the low predictability for the order could be used to signify the background. We illustrate this idea with a quiz in Figure 1.

Temporal order has been explored as a supervisory signal by a number of recent works [9, 24, 34, 39]. The goal of such methods, however, is to learn video representations by verifying the order for a short range of consequtive frames. We here address a different task and learn actionness [5] by exploiting long-range relations between video clips on the scale of minutes. To this end, we propose a new model and a method to learn actionness scores via a self-supervised proxy task of order verification. The model assigns high actionness scores to clips which order is easy to predict from other clips in the video. Our method is self-supervised and requires no manual annotation. Given this property, we use a very large HowTo100M dataset [23] with diverse and unlabeled instructional videos to learn an action-agnostic model for actionness. We show interesting insights of our method and demonstrate improved performance of action localization when combining our model with recent weakly-supervised approaches.

Contributions. This work makes the following contributions: (1) We develop a new model for action-agnostic action/background classification and propose to learn it via a self-supervised proxy task of long-range order verification. (2) We demonstrate a successful application of our method to the tasks of frame-wise action/background classification and action proposal generation evaluated on datasets with instructional videos COIN [31] and CrossTask [41]; (3) We further

[‖]**Quiz answers:** Action frames are shown in correct temporal order; Background frames are shown in reverse temporal order.

demonstrate the benefit of our model for action localization by combining it with recent weakly-supervised methods of step localization on the CrossTask dataset; and (4) We provide ablation studies that give insights about our approach.

2 Related work

2.1 Self-supervised learning

Our work exploits the natural source of supervision in videos: the temporal order between frames. The proposed method is, hence, related to a large body of recent work on self-supervised learning, where the supervisory signal is obtained directly from the data and does not require manual annotation. The variety of recently proposed self-supervised tasks in the image domain include prediction of image rotation [11], spotting artifacts [14], image colorization [36, 17], cross-channel prediction [37], inpainting [27] and predicting relative position of patches [6, 25]. In the video domain, motion has been used as a cue for learning video representations in [1, 26, 33, 7]. More related to our work, previous methods [9, 18, 24, 34, 35] explore the temporal order, either by predicting the exact order of consecutive frames [18, 35] or verifying their partial order [9, 24, 34]. Our work builds on these ideas but brings two important innovations. First, in contrast to previous work that exploits temporal order to learn local video representations, we address a different task of action/background classification. As actions are often separated by minutes, our task requires reasoning about *long-range* temporal order, as opposed to *short-range* frame permutations explored by previous methods. Our second innovation is, hence, a new method that exploits long-range order verification for video clips and enables to model relations between actions in 10-20 minutes long videos.

2.2 Learning from instructional videos

Instructional videos have recently been in the focus of numerous works in the context of action localization [2, 28, 41], joint learning of object states and actions [3], joint modeling of video and language [23, 30] and visual reference resolution [12, 13]. Some of this work exploits specific properties of instructional videos, such as the approximate temporal alignment between narrations and the visual content [2, 23, 30, 41], and the order consistency [2, 28, 41]. Similarly, we rely on the partial order between actions. Our novelty is to use the order verification as a proxy task to discover most relevant parts of the video. To demonstrate the value of our approach, we combine it with the previous methods [22, 23, 41] for the task of weakly-supervised step localization in instructional videos and demonstrate consistent improvements.

2.3 Action proposals

We apply our method to generate action proposals. Action proposals is an essential part of many methods for action detection, explored by a number of recent

papers [8, 10, 15, 19–21, 38]. A popular approach to generate action proposals is to estimate an *actionness* score for each temporal unit and then apply some sort of temporal grouping and non-maxima suppression. The notion of actionness was first introduced in [5] as a confidence measure of intentional bodily movement of biological agents. Most works [21, 19, 20, 10, 38] address actionness with supervised methods based on manual annotation of a known and limited set of action classes. This is done by training a binary classifier for estimating actionness score as first proposed in the context of spatial action detection [32]. Contrary to this approach, we aim to learn an actionness score without manual supervision by relying on generic assumptions about action order. Our definition of actions is narrower than in [5]. In particular, we only consider *goal-oriented actions*, necessary to perform specific manipulation tasks. This definition excludes actions such as gesticulation and conversations.

3 Unsupervised learning of actionness score

Given a large corpus of instructional videos depicting complex tasks, our goal is to automatically discover which segments of the videos are the most relevant for the successful completion of the tasks. We refer to these relevant segments as actions. Example of a complex task would be "*building a shelve*" and a relevant action would be "*drilling a hole*". Formally, we learn an *actionness* scoring function S that takes as input a video clip $x \in \mathbb{R}^{T \times H \times W \times 3}$ containing T frames of height H and width W and outputs a score $S(x) \in \mathbb{R}$ that is high when x corresponds to a relevant action and is low otherwise, *e.g.* on background scenes that are not relevant for completing the task.

We propose to learn S in a self-supervised manner through the pretext task of long range temporal order verification, which consists in predicting whether or not a set of video clips spanning a long temporal interval are in the correct order. The intuition is that one needs to isolate the relevant segments of the video that allow to best identify the correct ordering of events to be able to solve that task. We use that observation to train our actionness model S.

This section formally describes our method for joint order verification and actionness prediction. Section 3.1 introduces the model used. Section 3.2 details the training procedure, that allows to train the actionness score S via order verification.

3.1 Models for actionness and order verification

We represent each video clip x by a d-dimensional feature vector $h = f(x) \in \mathbb{R}^d$ obtained from a pre-trained video network f that we keep fixed throughout the work. In practice we use averaged pooled I3D representation [4] pretrained in [22], with $d = 1024$. For simplicity, we only refer to the feature vector h in the following (implicitly assuming h is associated with a video clip x). The actionness score $S(x)$ is estimated by a linear function on h, *i.e.*, $S(x) = w^\top h + b$, where $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are learnt weights and biases, respectively. To predict the

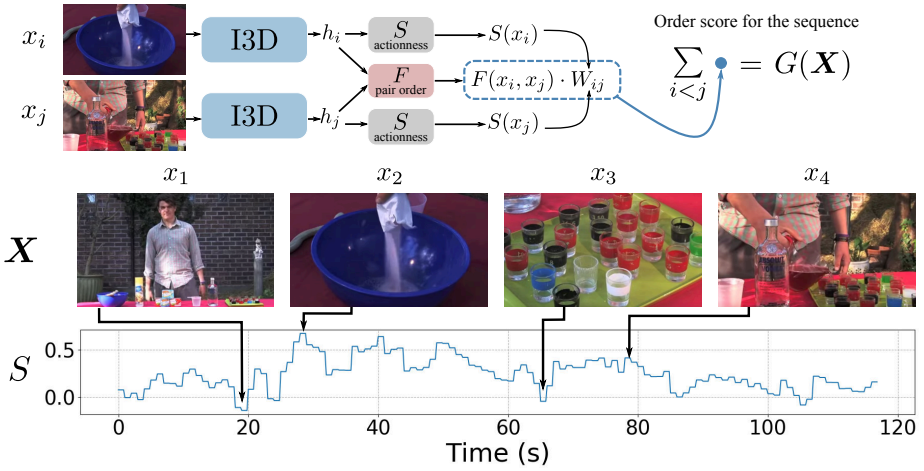


Fig. 2. Given a sequence of clips $\boldsymbol{X} = (x_1, x_2, ..., x_M)$ extracted from the same video as input, our model produces two types of outputs: confidence $S(x_i)$ that video clip x_i displays an action, and confidence $F(x_i, x_j)$ that x_i occurs before x_j in the original video. We combine these scores together to produce an order score $G(\boldsymbol{X})$ that reflects the model confidence that the sequence $\boldsymbol{X}$ is displayed in the correct order. We generate training data for G for *free* by simply maintaining or reverting order of videos. By doing so, the model automatically learns to put more weight $W_{ij} \propto \exp(S(x_i) + S(x_j))$ to clips from which it is easy to predict whether clip x_i has happened before clip x_i. We argue that clips with such properties are more likely to be actions that allow S to become an *actionness* score. **Top.** Architecture for producing the actionness score S and its training. **Bottom.** Evolution of the learnt actionness score S throughout a video. Note how the model learned to put higher weights to frames that correspond to actions such as *adding sugar* (x_2) or *pouring* (x_4) and low score on clips that are not relevant to the completion of the task (*i.e.*, *background*) such as a *man standing still* (x_1) or a clip only showing *empty glasses* (x_3)

order of clips, we also introduce a model F that takes as input two clips x and x' and outputs the confidence $F(x, x') \in \mathbb{R}$ that x happens before x'. We force F to be antisymmetric (*i.e.*, $F(x, x') = -F(x', x)$), by defining $F(x, x') = a^\top(h - h')$, where $a \in \mathbb{R}^d$ and $h' = f(x')$. Our choice of simple linear models is practicaly motivated by the fact that in our experiments we did not see improvements from using more complex models. Next, we describe the training strategy used to train S and F using order verification.

3.2 Training with ordering verification

Actionness through order verification. Our goal is to learn actionness score in an unsupervised manner. As explained earlier we believe that actions contain more information than background in terms of predicting what happens before or

what may come next in instructional videos as they carry more information about the global temporal structure of the video than the background. In this section, we use that observation to automatically differentiate actions from background. In short, the idea is to train a network to predict if a set of clips are in the correct order, a task for which it's trivial to get *free* supervision as correct ordering is naturally present in the video. In order to do so, we allow the model to softly select which pairs of clips from the set are best to perform that prediction, *i.e.*, those that are most informative in terms of their relative order in the video. Hence, by learning to predict order through weighted relative ordering of pairs of clips, we expect the model to pay more attention on important actions and therefore learn a good actionness score S. Details are given next.

Order verification task. As illustrated in Figure 2, given a sequence $\boldsymbol{X}$ of M video clips $\boldsymbol{X} = (x_i)_{i=1}^M$ randomly sampled from the *same* video, the task of order verification is to predict whether or not the clips in $\boldsymbol{X}$ are in the correct order, *i.e.*, the same order as the original video.

Ground truth generation. We create positives and negatives for the verification task by simply randomly sampling M clips from a video and either **(i)** create a positive sequence by sorting clips in the order of their appearence in the video (label $y = 1$) or **(ii)** reverse completely the original sequence (negative label $y = 0$).

Order verification prediction. We seek to predict if the sequence $\boldsymbol{X}$ is in the correct order through our pairwise model F that can predict the relative ordering of a given pair of clips (x_i, x_j). To make a more accurate prediction, it is better to aggregate scores over many pairs from the entire sequence $\boldsymbol{X}$. For this reason, we predict the confidence $G(\boldsymbol{X})$ that $\boldsymbol{X} = (x_i)_{i=1}^M$ is in the correct order as a weighted average of all pair-wise predictions:

$$G(\boldsymbol{X}) = \sum_{i<j} W_{ij} F(x_i, x_j). \tag{1}$$

Note that the sum indices $i < j$ are to make sure that (i) we only compare pairs in the order given by the sequence and (ii) we don't compare a clip to itself. Finally the weights W_{ij} are defined with a softmax over all pairs of clips:

$$W_{ij} = \frac{\exp(S(x_i) + S(x_j))}{\sum\limits_{i'<j'} \exp(S(x_{i'}) + S(x_{j'}))}, \tag{2}$$

where $S(x_i)$ and $S(x_j)$ are the actionness score of our model for clips x_i and x_j, respectively. Because of the softmax (2) we have $\sum_{i<j} W_{ij} = 1$ and $W_{ij} \geq 0$. Hence the weights W_{ij} can be seen as a way to softly select the contribution of every individual pair (x_i, x_j) since they control the contribution of the individual order pairwise prediction $F(x_i, x_j)$ in the global order score $G(\boldsymbol{X})$ (see (1)). This contribution to $G(\boldsymbol{X})$ is proportional to the sum $S(x_i)+S(x_j)$ of actionness score of both clips x_i *and* x_j. This is to match our intuition that *both* clips should be depicting a relevant action to facilitate the order prediction. By learning G we will therefore indirectly learn S as described by the training objective below.

Training objective. Given a sequence $\boldsymbol{X}$ and associated label y indicating whether or not the sequence of clips is in the correct order, we use the binary cross-entropy loss $\mathcal{L}$ as follows

$$\mathcal{L}(\boldsymbol{X}, y) = -y \log(\sigma(G(\boldsymbol{X}))) - (1-y) \log(1-\sigma(G(\boldsymbol{X}))), \tag{3}$$

where σ is the sigmoid function. This loss will enforce that when $\boldsymbol{X}$ is in the correct order (*i.e.*, $y = 1$), then $G(\boldsymbol{X})$ should be maximized. To do so, the actionness model S needs to be *high* on clips x_i and x_j from which it's *easier* to predict their correct ordering (meaning high value of $F(x_i, x_j)$) so that their contribution in $G(\boldsymbol{X})$ will be maximal. Conversely, when $\boldsymbol{X}$ is in the incorrect order (*i.e.*, $y = 0$), then $G(\boldsymbol{X})$ should be minimized. Again, to do so the actionness model S needs to be *high* on clips x_i and x_j from which it's *easier* to predict their incorrect ordering (meaning low value of $F(x_i, x_j)$) so that their contribution in $G(\boldsymbol{X})$ will be maximal. Following the standard procedure, we minimize the expected loss $\mathcal{L}$ on our training dataset.

4 Experiments

The experimental setup is described in Section 4.1. We then provide in Section 4.2 an ablation study of highlighting the most important components of our method. Finally, in Section 4.3, we show how we can use our learned actionness score for applications such as action proposals or weakly supervised action localization.

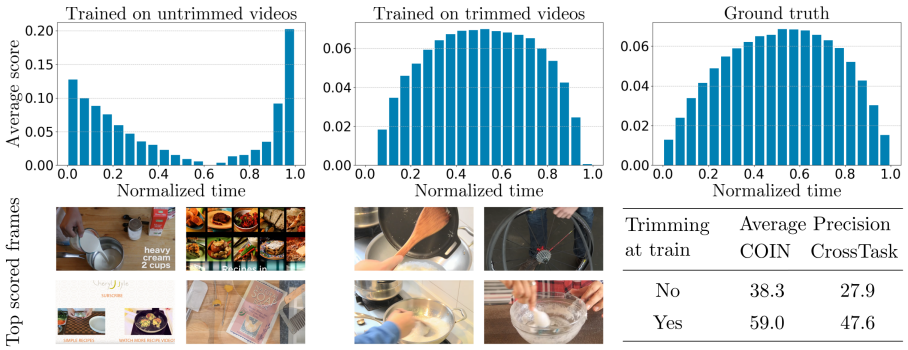
4.1 Experimental setup

Input processing. Given a clip x containing T frames, we extract $h(x)$ using the I3D backbone from [22], pre-trained on HowTo100M for the task of joint embedding of videos and subtitles. We extract the features at `Mixed_5c` in a fully convolutional manner and perform a global average pooling to have a single feature vector $h(x)$ of dimension 1024. Note that this network was trained without requiring manual annotations. To reduce computation cost during training, we pre-extract these features and directly work in feature space. In particular, this means that we do not finetune the I3D backbone for our task.
Training dataset. Due to the large variety of actions in instructional videos, the variety of their visual appearance as well as the order in which they are performed in videos, we require a large amount of data for our self-supervised task. For this reason, we train our model on HowTo100M [23], a large dataset containing more than 1.2 million videos depicting around 23,000 different tasks and was collected without manual annotation.
Training details. We optimize the objective (3) using the Adam optimizer [16] on a single GPU with batch size 1024, initial learning rate of 10^{-4} that we decay by a factor 0.9 at every epoch. We train for a total of 15 epochs. In addition, since HowTo100M is biased towards some specific domains (*e.g.* 40% of videos

are cooking), we resample the data, using the task taxonomy of HowTo100M. Precisely, we consider all subcategories of depth 3, *i.e.*, subcategories of the principal categories, such as Food and Cars, and sample equal number of videos from each subcategory. We also remove subcategories with less than 3000 videos.
Evaluation datasets. We use two instructional video datasets, COIN [31] and CrossTask [41]. Both datasets contain untrimmed videos that have been temporally annotated with action labels corresponding to the different steps of the task. In the following, time intervals without any action labels are considered as background. We use the official COIN test subset of 2797 videos for evaluation. Since there is no official test subset of CrossTask, we randomly split it into a training (2062 videos) and test sets (688 videos). In addition, we made sure to discard all COIN and CrossTask videos from our training set.
Evaluation tasks. We use three different tasks for evaluation, detailed next.
Background vs. action classification. In this task, videos from the respective test sets are split into non overlapping 0.2s segments. For each segment we assign a binary label: positive (action) if the segment overlaps with an annotated action interval and negative (background) otherwise. The goal is to classify each segment as an action or a background. We use average precision (AP) as an evaluation metric for this task.
Action temporal proposal generation. The task is to generate a set of proposals, that overlap well with ground truth action intervals. To generate proposals from the outputs of the network $S(x)$, we use the Temporal Actionness Grouping (TAG) method [38]. We set the Intersection over Union (IoU) threshold for non-maximum suppression to 0.8. We follow the evaluation protocol of [20] using the implementation provided in [19] to compute Average Recall (AR) at multiple IoU thresholds: $[0.5 : 0.05 : 0.95]$. Finally, we report AR as a function of the Average Number (AN) of proposals per video AR@AN as done in [20].
Action step localization. Third task is step localization on CrossTask. Given an ordered list of actions for a video, the goal is to assign each action to exactly one frame. We use the same evaluation protocol as in [41] and report average recall.

4.2 Ablations studies

This section ablates the important components of our method. We report performance using *background vs action classification* task on COIN and CrossTask.
Video trimming. When first training our method, we notice that the model could learn to be good at the self-supervised ordering task by putting high score on intro and outro segments of videos. This can be intuitively explained by the fact that the beginning and end of instructional videos are distinctive: they start with some typical introduction and often finish with credits. This effect is demonstrated on the left part of Figure 3: when trained on untrimmed videos, the actionness score $S(x)$ is high for beginning and end of videos simply because the model can easily discriminate if a frame is from intro (beginning) or outro (end) and hence can safely select these frames to predict relative ordering of pairs. Top scoring frames illustrate that the model picks up on typical credit frames. This led to bad performance as these segments do not contain relevant actions. To

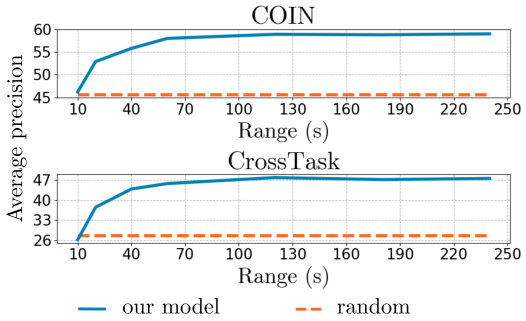


Trimming at train	Average Precision COIN	CrossTask
No	38.3	27.9
Yes	59.0	47.6

Fig. 3. *Actionness scores at different temporal locations of COIN videos test set.* (**left**) scores of the actionness model S, trained on untrimmed videos. (**center**) scores of the model S, trained on trimmed videos. (**right**) Distribution of actions in the videos. When trained on untrimmed videos, the score concentrates in the beginning and in the end of the video. When trained on untrimmed videos, the score distribution is closer to the ground truth action label distribution which leads to significant increase in background vs. action classification performance

alleviate this effect, we employ a simple but effective strategy: during training we trim off 30% of frames in the beginning and 30% of frames in the end of the video. Note that we only do that at training and do not trim test videos to keep the evaluation protocol consistent. Figure 3 shows that the temporal distribution of scores that we obtain by this technique better matches the ground truth one, as expected. In particular it's interesting to note that our model is still able to assign frames to actions even for frames that are in the beginning or the end when relevant (*i.e.*, this trimming did not handicaped the model for true actions that happen early or late in the video). Finally, the table in Figure 3 demonstrates the effectiveness of this approach on our two evaluation datasets.

Long vs. short range order verification. The driving hypothesis of this work is that learning actionness score S is possible thanks to the long range order consistency between actions in instructional videos. We claim that this is not true for short-term ordering between frames as used in previous work [24], as in that case order verification can be done via low level visual cues regardless of whether images depict actions or background. We verify that claim by training our model at different temporal scales. Formally, we set d to be the temporal window length in which we are going to sample $M = 5$ segments of length $d/10$. In other words, d corresponds to the maximum distance we can have between two clips from $\boldsymbol{X}$, and is a good measure of the *range* at which we perform the order verification task. Figure 4 (left) shows the results for different values of d. For small values of d, the model shows poor performance, compared to larger values. This demonstrates that our method works better on the scale of several minutes (long range), rather than 10-20 seconds (short range).



Number of segments	Average precision	
	COIN	CrossTask
3	59.4	43.1
4	59.9	46.0
5	59.5	47.2
6	59.0	47.2
7	58.2	46.5
8	57.6	45.6
9	57.1	45.1

Fig. 4. Left: long vs. short range. Action vs. background average precision of our model on COIN (top) and on CrossTask (bottom) as a function of the range d (in seconds) spanned by the sequence $\boldsymbol{X}$. Duration of each sampled clip equals $d/10$. Performance increases with the range used for the order prediction task, confirming our hypothesis that long range action dependencies are better to learn about actionness. **Right: number of segments.** Average precision of our model on COIN and CrossTask for different number of segments M sampled per video

Number of segments per video. In Figure 4 (right), we study the effect on performance when training our model with different number of segments M per video. Results are given for $M \geq 3$ ($M = 2$ makes training of S impossible since there are no pairs to select from). We observe that overall the method is not too sensitive to M. However, only sampling 3 segments per video may often lead to a situation where all selected segments are background, hence selecting larger value for M leads to better results. Figure 4 also shows a decline in precision for $M > 5$ on both datasets. This can be explained by the fact that for large values of M there is a high probability of having at least one pair of segments, for which the temporal order is easy to guess. This may decrease the ability of our model to learn temporal order for other more subtle pairs. Based on that study, we use $M = 5$ in all of our other experiments.

4.3 Actionness score for practical applications

Action vs. Background classification. In Table 1, we compare against five methods: **(i)** chance baseline, **(ii)** chance baseline with trimming, **(iii)** hand detector, **(iv)** optical flow and **(v)** a supervised model. **(i)** simply assigns random uniform action scores to segments in the video. Following our observation from Section 4.2 about trimming, **(ii)** does the same as **(i)** but also assigns to background the segments that occur in the first and last 30% portion of the video. **(iii)** assigns the actionness score to the maximum score of a hand detector [29] computed for each frame. This baseline is based on the assumption that the actions correlate with the appearance of hands. **(iv)** assumes that the actions correlate with motion and estimates an actionness score as an average magnitude

Table 1. Action vs. Background. Frame-wise average precision of background separation on COIN and CrossTask datasets

Dataset	(i) Chance	(ii) Chance (trim@0.3)	(iii) Hand Detector	(iv) Optical Flow	Ours	(v) Supervised
COIN	45.6%	53.5%	50.6%	47%	**59.0%**	70.7%
CrossTask	27.5%	32.6%	28.4%	30.3%	**47.6%**	56.2%

of optical flow at each frame. Finally for the supervised topline **(v)**, we train in a supervised manner a linear layer on top of our feature representation $h(x)$ for the binary action vs. background classification task. As we also use a linear layer for S, **(v)** provides an upper bound of performance, that can possibly be achieved with our approach. Methods **(ii-iv)** provide simple, yet meaningful baselines in the absence of existing unsupervised methods for the considered task. The results are shown in Table 1. Baselines **(ii-iv)** show only a marginal improvement over Chance, which illustrates the difficulty of the task. **(ii)** provides the strongest baseline on both COIN and CrossTask, highlighting the importance of intro and outro segments for action vs. background classification. Our method shows an improvement over the baselines on both datasets.

Temporal Action Proposals. Following the evaluation protocol described in Section 4.1, we compare to 6 different methods: **(i)** chance baseline, **(ii)** chance baseline with trimming, **(iii)** hand detector, **(iv)** optical flow, **(v)** supervised and **(vi)** temporal prior. **(i-v)** are the same as for the Action vs. Background classification task. For **(vi)**, we use a temporal prior to generate action segments: it consists in sampling proposal start and length from a prior distribution, obtained from ground truth action intervals. In details, we compute the empirical distribution of the normalized start time of actions as well as their duration. We then randomly sample segments from that distribution by first sampling a start time and then a duration. Note that this baseline has access to more annotation than our proposed approach, since we don't have access to any temporal annotation. Figure 5 shows the average recall for CrossTask and COIN, as a function of average number of proposals per video (AN). On both datasets we outpeform baselines **(i-iv)** for most values of AN. More interestingly, we also significantly outpeform the temporal prior **(vi)** approach despite using less annotation information. Finally it is worth noting that the gap between our method and the supervised **(v)** topline is not large (less than between our method and **(vi)**).

Step localization. In this experiment, we explore how our actionness model can improve weakly supervised action localization. This task is particularly relevant since presence of background is one of the main challenges for weakly supervised action localization methods. To do so, we augment various action localization methods from [41], [23] and [22] with our actionness score $S(x)$ and evaluate on the CrossTask dataset following the protocol described in [41]. These methods work in a similar way, described next. First, step classifiers are applied to every frame of the video. Then, each step is assigned to exactly one short clip, using a dynamic programming to solve: $(t_1^*, ..., t_K^*) = \arg \max_{t_1 < ... < t_K} \sum_{t=1}^{T} \sum_{k=1}^{K} f_k(x_t)$,

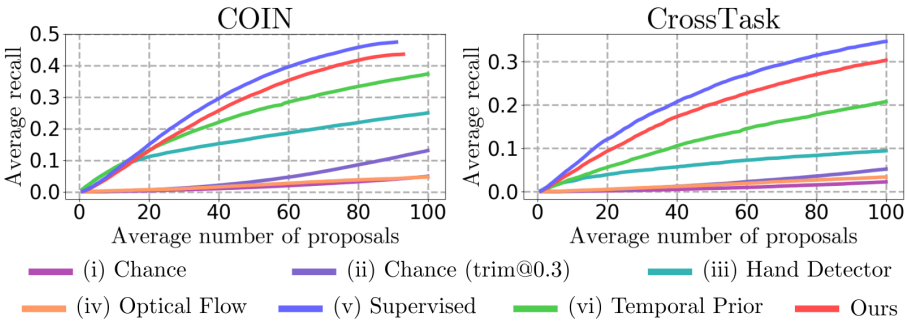


Fig. 5. Action proposal. Average recall versus average number of proposals per video

where $f_k(x_t)$ is the output of the step classifier k for the t-th input clip x_t, and t_1, ..., t_K are the clips ids assigned to steps 1, ..., K, respectively. [41], [23] and [22] differ only in the form of the classifier $f_k(x)$. We augment these methods with our actionness score by simply adding it to the objective during inference:

$$(t_1^*, ..., t_K^*) = \arg \max_{t_1 < ... < t_K} \sum_{t=1}^{T} \left[(1-\alpha) \sum_{k=1}^{K} f_k(x_t) + \alpha S(x_t) \right], \tag{4}$$

where $S(x_t)$ is the actionness score of our model for clip x_t and α is a combination parameter. Intuitively, the role of our score is to lower the confidence of background clips and increase the score on foreground action clips.

Results are provided in Table 2. α is selected independently for each method, which equals 0.1 for [23] and 0.8 for other methods. We use the same value of α for all test tasks. Combining the baseline method with our score improves the performance in every case. The gap in performance is particularly large for the method from Zhukov *et al.* [41]. This can be in part attributed to the fact that this method does not try to model the background for a given frame (indeed a simple constraint imposes that the score should sum to one across time for all actions without trying to explicitly lower score on detected background frames). Adding our score to the outputs of the model resolves this problem and leads to a large improvement (+4.6% recall). Other methods in Table 2 rely instead on a joint video and text embedding pretrained on the large scale HowTo100M dataset to score every segment of the video against the text embeddings of the action description. These text-video embeddings approach lead to much stronger base model. However, given a frame, there is no guarantee that the model will explicitly be looking for actions as scores can still be high if its visual content partially matches the description of an action. For example, if the action is *season steak*, the presence of a *steak* and the object *salt* in the frame can increase the similarity score even if the action is not visible. Interestingly, combining these much stronger base models with our score still improves performance by

	+ actionness	Make Kimchi Rice	Pickle Cucumber	Make Banana Ice Cream	Grill Steak	Jack Up Car	Make Jello Shots	Change Tire	Make Lemonade	Add Oil to Car	Make Latte	Build Shelves	Make Taco Salad	Make French Toast	Make Irish Coffee	Make Strawberry Cake	Make Pancakes	Make Meringue	Make Fish Curry	Average
[41]	✗	13.3	18.0	23.4	23.1	16.9	16.5	30.7	21.6	4.6	19.5	35.3	10.0	32.3	13.8	29.5	37.6	**43.0**	13.3	22.4
[41]	✓	**18.4**	**24.9**	**25.6**	**24.1**	**19.0**	**29.6**	**33.8**	**30.0**	**7.7**	**23.7**	**45.0**	**13.4**	**36.1**	**23.7**	**34.3**	**41.9**	42.0	**15.8**	**27.0**
[23]	✗	33.5	27.1	**36.6**	37.9	24.1	35.6	32.7	35.1	30.7	28.5	43.2	19.8	**34.7**	33.6	**40.4**	**41.6**	**41.9**	**27.4**	33.6
[23]	✓	**33.6**	**28.6**	35.4	**38.5**	**25.0**	**37.3**	**35.1**	**41.2**	**30.9**	**30.1**	**45.1**	**21.4**	33.7	**34.3**	39.1	41.2	40.3	26.3	**34.0**
I3D [22]	✗	28.7	**37.9**	**42.8**	36.3	22.0	42.9	**27.4**	43.1	30.8	32.7	42.8	27.5	**34.0**	**33.7**	44.3	48.0	**46.0**	33.9	36.4
I3D [22]	✓	**31.1**	37.2	42.6	**37.4**	**23.5**	**43.4**	27.1	**43.4**	**32.2**	**35.9**	**46.0**	**29.4**	33.9	36.6	**45.6**	**49.7**	45.2	**37.0**	**37.6**
S3D [22]	✗	31.5	36.0	46.5	38.5	25.2	45.0	33.3	**48.1**	**38.4**	**37.0**	48.1	34.2	**38.7**	41.9	**44.6**	**48.2**	**52.2**	**38.0**	40.3
S3D [22]	✓	**34.1**	**40.0**	**48.7**	**40.3**	**30.7**	**46.1**	**34.5**	45.9	38.1	35.9	**50.0**	**35.4**	38.1	**42.6**	42.6	45.9	51.6	37.8	**41.0**

Table 2. Step localization results on CrossTask with and without our actionness score

a significant margin (+1.2% recall and +0.7% recall for the best S3D model), hence setting a new state-of-the-art on the CrossTask benchmark [41].

4.4 Qualitative results

To provide more insight about the kind of signal captured by our model S, we provide clips from HowTo100M with high and low actionness scores in Figure 6. In order to obtain these examples, we run our model on 50,000 randomly sampled videos from HowTo100M. To show the variety of different tasks, we illustrate the top 10 highest scoring clips within each of the four largest HowTo100M categories: Food and Entertaining, Home and Garden, Hobbies and Craft and Cars & Other Vehicles. Finally, we also give the 10 lowest scoring clips across all categories to illustrate the type of background discovered by our model.

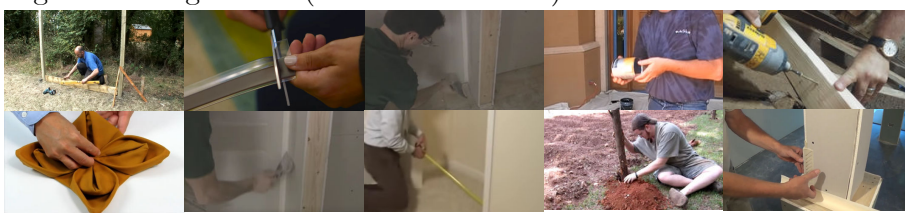
5 Conclusion

In this paper, we have presented a self-supervised method that can separate actions from background without resorting to any form of manual annotation. It does so by leveraging the assumption that frames that depict key actions are more informative when it comes to predict what may come next or what happens in the past. Equipped with our method, we managed to improve the state-of-the-art on a challenging action localization benchmark. As future work we notably plan to investigate if our method would generalize to broader domains than instructional videos. Another potential direction would be to jointly discriminate background and actions while also clustering similar actions together, thus paving the way for unsupervised action discovery.

Acknowledgements. This work was partially supported by the European Regional Development Fund under project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15 003/0000468), Louis Vuitton ENS Chair on Artificial Intelligence, the MSR-Inria joint lab, and the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

Highest scoring clips (Food and Entertainment)

Highest scoring frames (Home and Garden)



Highest scoring frames (Hobbies and Craft)

Highest scoring frames (Cars & Other Vehicles)

Lowest scoring frames (all categories)

Fig. 6. First four rows show highest actionness scoring clips from the top 4 categories of HowTo100M: Food and Entertaining, Home and Garden, Hobbies and Craft and Cars & Other Vehicles. Bottom row illustrates the lowest scoring clips according to our actionness score S (see Supplementary material for more examples)

References

1. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
2. Alayrac, J.B., Bojanowski, P., Agrawal, N., Laptev, I., Sivic, J., Lacoste Julien, S.: Unsupervised learning from narrated instruction videos. In: CVPR (2016)
3. Alayrac, J.B., Sivic, J., Laptev, I., Lacoste-Julien, S.: Joint discovery of object states and manipulation actions. In: ICCV (2017)
4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
5. Chen, W., Xiong, C., Xu, R., Corso, J.J.: Actionness ranking with lattice conditional ordinal random fields. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2014)
6. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
7. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: Temporal cycle-consistency learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
8. Escorcia, V., Heilbron, F.C., Niebles, J.C., Ghanem, B.: Daps: Deep action proposals for action understanding. In: The European Conference on Computer Vision (ECCV) (September 2016)
9. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
10. Gao, J., Chen, K., Nevatia, R.: Ctap: Complementary temporal action proposal generation. In: The European Conference on Computer Vision (ECCV) (September 2018)
11. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (April 2018)
12. Huang, D.A., Lim, J.J., Fei-Fei, L., Niebles, J.C.: Unsupervised visual-linguistic reference resolution in instructional videos. In: CVPR (2017)
13. Huang, D.A., Ramanathan, V., Mahajan, D., Torresani, L., Paluri, M., Fei-Fei, L., Niebles, J.C.: Finding "it": Weakly-supervised reference-aware visual grounding in instructional video. In: CVPR (2018)
14. Jenni, S., Favaro, P.: Self-supervised feature learning by learning to spot artifacts. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
15. Ji, J., Cao, K., Niebles, J.C.: Learning temporal action proposals with fewer labels. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
16. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
18. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)

19. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: Bmn: Boundary-matching network for temporal action proposal generation. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
20. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. In: The European Conference on Computer Vision (ECCV) (September 2018)
21. Liu, Y., Ma, L., Zhang, Y., Liu, W., Chang, S.F.: Multi-granularity generator for temporal action proposal. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
22. Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos (2020)
23. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
24. Misra, I., Zitnick, C., Hebert, M.: Shuffle and learn: Unsupervised learning using temporal order verification. In: ECCV (September 2016)
25. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: The European Conference on Computer Vision (ECCV) (September 2016)
26. Pathak, D., Girshick, R., Dollar, P., Darrell, T., Hariharan, B.: Learning features by watching objects move. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
27. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
28. Sener, F., Yao, A.: Unsupervised learning and segmentation of complex activities from video. In: CVPR (2018)
29. Shan, D., Geng, J., Shu, M., Fouhey, D.F.: Understanding human hands in contact at internet scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
30. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
31. Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J.: Coin: A large-scale dataset for comprehensive instructional video analysis. In: CVPR (2019)
32. Wang, L., Qiao, Y., Tang, X., Van Gool, L.: Actionness estimation using hybrid fully convolutional networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
33. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
34. Wei, D., Lim, J.J., Zisserman, A., Freeman, W.T.: Learning and using the arrow of time. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
35. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
36. Zhang, R., Isola, P., Efros, A.: Colorful image colorization. In: The European Conference on Computer Vision (ECCV) (September 2016)

37. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
38. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
39. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
40. Zhou, L., Chenliang, X., Corso, J.J.: Towards automatic learning of procedures from web instructional videos. In: AAAI (2018)
41. Zhukov, D., Alayrac, J.B., Cinbis, R.G., Fouhey, D., Laptev, I., Sivic, J.: Cross-task weakly supervised learning from instructional videos. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)