TSP: Temporally-Sensitive Pretraining of Video Encoders for Localization Tasks

Humam Alwassel Silvio Giancola Bernard Ghanem
King Abdullah University of Science and Technology (KAUST)
{humam.alwassel,silvio.giancola,bernard.ghanem}@kaust.edu.sa
http://humamalwassel.com/publication/tsp

Abstract

Due to the large memory footprint of untrimmed videos, current state-of-the-art video localization methods operate atop precomputed video clip features. These features are extracted from video encoders typically trained for trimmed action classification tasks, making such features not necessarily suitable for temporal localization. In this work, we propose a novel supervised pretraining paradigm for clip features that not only trains to classify activities but also considers background clips and global video information to improve temporal sensitivity. Extensive experiments show that using features trained with our novel pretraining strategy significantly improves the performance of recent state-of-the-art methods on three tasks: Temporal Action Localization, Action Proposal Generation, and Dense Video Captioning. We also show that our pretraining approach is effective across three encoder architectures and two pretraining datasets. We believe video feature encoding is an important building block for localization algorithms, and extracting temporally-sensitive features should be of paramount importance in building more accurate models. The code and pretrained models are available on our project website.

1. Introduction

Video understanding is thriving in the computer vision community, and it manifests in several challenging tasks such as action classification [13, 34, 41, 69], activity localization [16, 36, 86], and video captioning [25, 53, 71, 88]. Yet, the success of video research has been lagging behind that of its counterpart in the image domain. In many aspects, this is due to the exponentially larger amount of data in videos compared to images, not fitting in commodity hardware. Image encoders have the privilege to process batches of complete images at once, thus exploiting the rich contextual information from all pixels. Empowered by such capability, many image models are trained in an end-to-end manner for complex tasks such as object detection [59, 60, 67], semantic segmentation [10, 23, 26], and image captioning [4, 84, 48]. In contrast, the long and variable length of *untrimmed* videos makes it impractical to encode a complete video on current hardware accelerators [76]. While a few recent localization works [44, 87] attempt to train end-to-end for *untrimmed* video tasks, such as temporal action localization, they need to resort to aggressive spatial and temporal downsampling to remain computationally practical. Instead, most state-of-the-art localization methods for *untrimmed* videos choose to learn models atop *precomputed* clip features [27, 42, 81, 85].

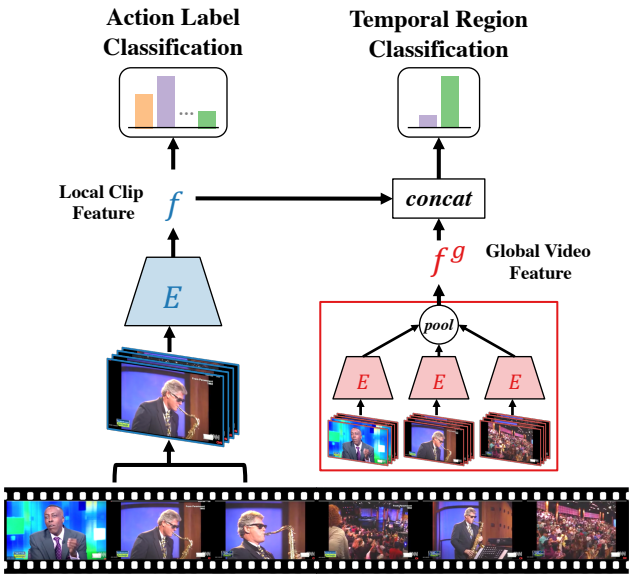

Figure 1: **Temporally-Sensitive Pretraining (TSP)**. We train video encoders to be temporally-sensitive through a novel supervised pretraining paradigm. A fixed-sized clip is sampled from an untrimmed video and passed through the encoder to obtain a local clip feature (blue). A global video feature (red) is pooled from the local features of all clips in the untrimmed video. The local and global features are used to train the encoder on the task of classifying the label of foreground clips (action label) and classifying whether a clip is inside or outside the action (temporal region).

In this work, we focus on improving the precomputed features used for temporal localization tasks, which we de-

fine as tasks that require predictions related to the time dimension of the video. Specifically, we target three important localization problems: Temporal Action Localization (TAL), Action Proposal Generation (Proposals), and Dense Video Captioning (Dense-Captioning). State-of-the-art methods for these localization tasks use features extracted from video encoders typically pretrained for the task of *Trimmed Action Classification* (TAC) on large-scale datasets, such as Kinetics [32] and Sports-1M [31]. However, this pretrained representation is not necessarily suitable for localization tasks. In particular, we observe that TAC-pretrained features tend to be temporally-insensitive, *i.e*. background (no action) segments can have quite similar representations to foreground (action) segments from the same untrimmed video. We provide an analysis study of TAC-pretrained features in Section 5 that shows evidence of the high cosine similarity between features of background and foreground clips. These temporally-insensitive features make it harder for the localization algorithm to learn the target task, and thus, negatively impact the final performance.

To circumvent these drawbacks, we propose a novel, supervised pretraining paradigm for video clip representation that not only trains to classify foreground activities but also considers background clips and global video information to improve temporal sensitivity. We refer to our pretraining approach as *Temporally-Sensitive Pretraining* (TSP). Figure 1 gives an overview of TSP. We conduct extensive experiments to show that features extracted by clip encoders pretrained with TSP are more discriminative, and that training state-of-the-art localization algorithms atop TSP features results in significant performance gains on three temporal localization tasks: TAL, Proposals, and Dense-Captioning. Moreover, TSP gives consistent performance boosts regardless of the video encoder architecture, pretraining dataset, or the localization algorithm learned atop our features. Interestingly, we observe that localization performance on short instances greatly improves when using TSP pretrained features. This aligns well with our hypothesis that temporally-sensitive features allow localization algorithms to draw sharper contrast between foreground and background context in long untrimmed videos.

Contributions. (I) We propose TSP, a temporally-sensitive supervised pretraining task for video encoders. TSP trains an encoder to explicitly discriminate between foreground and background clips in untrimmed videos. **(II)** We show with comprehensive experiments that using features pretrained with the TSP task significantly improves performance across three video localization problems. Additionally, we show the generalization capability of our pretraining strategy on three encoder architectures and two pretraining datasets. We also demonstrate consistent performance gains for multiple localization algorithms trained on the same target problem. **(III)** We provide an extensive analysis study of our features. Interestingly, we observe that TSP pretraining boosts temporal action localization performance on short action instances. The study also demonstrates that our features are in fact temporally-sensitive and can encode background clips differently from foreground clips.

2. Related Work

Action recognition. Large-scale video datasets, such as UCF-101 [66], Sports-1M [31], and Kinetics [32], have accelerated the development of action classification models. Simonyan and Zisserman [65] introduced a two-stream encoder to represent appearance with RGB frames and motion with stacked optical flow vectors. Wang *et al*. [74] proposed the Temporal Segment Network (TSN) encoder to capture long-term temporal information. Pretrained on TAC, TSN along with other recent architectures (*e.g*. R(2+1)D [70], I3D [8], and C3D [68]) have become the *de facto* feature extractors for temporal action localization (TAL) [57], action segmentation [15], and event captioning [80]. Since TAC-pretraining is not necessarily suitable for these localization tasks, we propose a pretraining that learns from both foreground and background clips in untrimmed videos.

Temporal action localization and proposal generation. Many algorithms have been developed for TAL [2, 11, 21, 62, 83]. While the majority has been on fully-supervised TAL [22, 38, 40, 44, 47], recent works have also studied TAL under weak supervision [37, 46, 54, 55, 63], single-frame supervision [49], and self-supervision [30]. The first generation of algorithms applied complex action classifiers in a sliding window fashion [14, 52]. To alleviate the expensive cost of sliding an action classifier over long videos, the second generation of algorithms [6, 18, 42, 43, 64, 85] followed a two-stage approach that first learns action proposals to limit the number of candidates passed to the action classifier. A third set of algorithms jointly learn action proposals and action classifiers in one stage [9, 79, 81, 87]. A few works [44, 87] learn TAL end-to-end by drastically downsampling videos to be computationally practical, *e.g*. PBR-Net [44] uses only 3 frames per second on ActivityNet and SSN [87] uses only 9 clips per proposal. In contrast, most state-of-the-art methods build atop precomputed features from TAC-pretrained encoders. Since experiments show that such features are not best suited for TAL and Proposals, we propose to replace them with temporally-sensitive pretrained features that can significantly boost performance.

Dense video captioning. Krishna *et al*. [35] introduced the task of Dense-Captioning along with the ActivityNet Captions benchmark. Dense-Captioning aims at both localizing and textually describing all events in a video. This problem branched out from video captioning [75, 82, 56], where a full video is captioned without localizing events. [35] uses a variant of DAPs [12] to generate proposals and

employs an LSTM-based captioning module to describe these proposals. Subsequent works use bidirectional attentive fusion [72], masked transformers [89], and reinforcement learning [39, 51, 77]. A line of multi-modal Dense-Captioning methods combine visual cues with signals from audio [58], speech/subtitles [61], or both [27, 28]. Similar to TAL and Proposals, Dense-Captioning algorithms rely on temporally-insensitive TAC-pretrained features, which do not perform as well as the TSP pretrained ones.

3. Technical Approach

3.1. Traditional Pretraining Strategies

Since it is impractical to fit entire *untrimmed* videos into commodity GPUs without drastically downsampling space or time, current state-of-the-art localization algorithms share a common practice in that they do not fine-tune their video encoders directly on the target task (*e.g.* TAL). Instead, they use *pretrained* encoders as *fixed* feature extractors [27, 42, 81, 85]. Trimmed action classification (TAC) has been the traditional approach to pretrain these encoders. The TAC task aims to classify clips from short videos, where the action spans the entire video. While TAC has been successful in providing features that discriminate between different action classes, it often fails to distinguish between the action instance and its nearby background context. For example, recent diagnostic studies [1] have shown that state-of-the-art TAL methods are quite sensitive to the context around action instances and that their inability to distinguish between an action and its temporal background context is the main roadblock to improving localization performance. We argue that the features used in these state-of-the-art localization methods, pretrained on TAC, are a source of such confusion. Thus, we propose to depart from the traditional strategy and render the features temporally-sensitive through a novel pretraining task.

3.2. How to Incorporate Temporal Sensitivity?

A limiting aspect of TAC-pretrained encoders is that they only learn from positive samples (foreground/action clips). Intuitively, learning from negative samples (background/no action clips) is expected to improve the temporal discriminative ability of these encoders. Given an untrimmed video, a good encoder for localization problems should be able to distinguish between the semantics of different actions as well as between actions and their background context. Intuitively, clip features that have an idea of whether the clip is inside or outside an action can directly help localization methods find better activity/proposal boundaries for TAL and Proposals and find better captions for Dense-Captioning. Thus, we propose to pretrain encoders on the task of (1) classifying the label of foreground clips and (2) classifying whether a clip is inside or outside the action.

3.3. Temporally-Sensitive Pretraining (TSP)

Input data. We pretrain our model using untrimmed videos with temporal annotations. The encoder is learned in an end-to-end fashion from the raw video input. In particular, given an untrimmed video, we sample a fixed-size input clip $\mathbf{X}$ of size $3\times L\times H\times W$, where 3 refers to the RGB channels, L is the number of frames, and H and W are the frame height and width. We assign $\mathbf{X}$ two labels: (1) the action class label $\mathbf{y}^c$ if this clip is from a foreground segment, and (2) the binary temporal region label $\mathbf{y}^r$ that indicates if the clip is from a foreground/action ($\mathbf{y}^r = 1$) or background/no action ($\mathbf{y}^r = 0$) region of the video.

Local and global feature encoding. Let E be the video encoder that transforms a clip $\mathbf{X}$ into a feature vector f of size F. We refer to f as the local clip feature. Let $\{\mathbf{X}_i\}$ be the set of clips from an untrimmed video. We refer to the max-pooled feature $f^g = \max(E(\mathbf{X}_i))$ as the global video feature (GVF). Given only a short clip $\mathbf{X}$, it is challenging to classify whether $\mathbf{X}$ is inside or outside an action. The challenge stems from the fact that we only have access to local context, while the task we wish to solve inherently requires global understanding of the video content. To overcome this challenge, we combine the GVF with the local clip feature to better learn the task. We can think of the GVF as a conditioning vector for deciding foreground *vs*. background. We study other GVF pooling functions in the appendix.

Two classification heads. We employ two classification heads to pretrain the encoder. Specifically, the first head (action label head) consists of a fully-connected (FC) layer $\mathbf{W}^c$ of size $F \times C$, where C is the number of action classes in the dataset. $\mathbf{W}^c$ transforms the local features f to an action label logits vector $\hat{\mathbf{y}}^c$. The second head (temporal region head) is an FC layer $\mathbf{W}^r$ of size $2F\times 2$, which takes as input the concatenation of the local and global features, $f \oplus f^g$, to produce a temporal region logits vector $\hat{\mathbf{y}}^r$.

Loss. We optimize our loss for each input clip $\mathbf{X}$:

$$\text{loss} = \begin{cases} \alpha^r \mathcal{L}(\hat{\mathbf{y}}^r, \mathbf{y}^r) + \alpha^c \mathcal{L}(\hat{\mathbf{y}}^c, \mathbf{y}^c), & \text{if } \mathbf{y}^r = 1 \\ \alpha^r \mathcal{L}(\hat{\mathbf{y}}^r, \mathbf{y}^r), & \text{otherwise,} \end{cases} \quad (1)$$

where $\mathcal{L}$ is the cross-entropy loss and (α^c, α^r) are trade-off coefficients to weigh the losses of the two heads. The loss is the sum of the two head losses when the clip is from the foreground, *i.e.* $\mathbf{y}^r = 1$, and is the loss from the second head when the clip is from the background.

Optimization details. Temporally annotated video datasets have a natural imbalance between the temporal duration of foreground *vs*. background. To mitigate this imbalance, we subsample clips from videos in such a way that we train on the same number of foreground and background clip samples. We initialize our encoder weights with those pretrained on Kinetics-400 [32]. Many of the recent video ar-

chitectures have publicly released their Kinetics-pretrained weights, and we make use of these models in our experiments. Ideally, we wish to backpropagate the loss (Equation 1) through the GVF portion of our model. However, and as mentioned earlier, it is impractical to treat entire *untrimmed* videos in commodity GPUs. Thus, we freeze the GVF during training, *i.e*. we precompute the GVF of each video from the Kinetics-pretrained initialized encoder.

4. Experiments

4.1. Experimental Settings

Pretraining datasets. To pretrain with our TSP strategy, we need a dataset of untrimmed videos with temporal boundary annotations. Thus, we leverage two standard datasets: ActivityNet v1.3 [7] and THUMOS14 [29]. ***ActivityNet***: This dataset has 20K untrimmed videos and 200 activity classes. It is split into training, validation, and testing subsets, where the testing subset labels are withheld for an annual challenge. Following standard practices, we use the training subset (10024 videos) to train and the validation subset (4926 videos) to test. ***THUMOS14***: This dataset has 1010 validation and 1574 testing videos annotated with 101 sport-related action classes at the video-level. Among these videos, only 200 validation and 213 testing videos have temporal annotations for 20 sport actions. We use these 200 validation videos to train and the 213 testing videos to test.

Encoder architectures. We conduct experiments using two architectures: ResNet3D and R(2+1)D [70]. We select these backbones for their recognized good performance, speed, and efficiency. ***ResNet3D***: This is the 3D version of the 2D ResNet [24] CNN for images. ResNet3D is composed of a series of 3D convolution layers with residual skip connections. In our experiments and for simplicity, we consider the 18-layer variant of ResNet3D. ***R(2+1)D***: This encoder is also a ResNet-based backbone. It decomposes each spatio-temporal 3D convolution kernel into a 2D (spatial) and a 1D (temporal) convolution. Compared to ResNet3D, R(2+1)D is more efficient and light-weight, and it has been shown to maintain high performance on video tasks. In our experiments, we use the 18 and 34-layer versions of R(2+1)D.

Implementation details. In order to cope with the diversity of video formats present in ActivityNet and THUMOS14, we re-encode all videos in MP4 format with a constant frame rate of 30 fps. We sample clips of $L = 16$ frames with a stride of 2 frames, such that each clip covers a temporal receptive field of approximately one second. While keeping the aspect ratio fixed, frames are resized such that the smallest dimension is 128 pixels and then cropped to $H \times W = 112 \times 112$ pixels, randomly in training but deterministically centered during testing. The videos are split into temporally contiguous segments, representing foreground (action) and background (no action) content. We select 5 clips per segment, sampled randomly (temporal jittering) during training and uniformly in testing. We set $\alpha^c = \alpha^r = 1$ in Equation (1), and use a distributed SGD optimizer with different learning rates per module: 10^{-4} for the video encoder and a grid search among [0.002, 0.004, 0.006, 0.008, 0.01] for the two classification heads. We train for 8 epochs with a batch size of 32 clips per GPU. We use two V100 GPUs and scale the learning rate linearly with the number of GPUs. We use a linear learning rate warm up strategy over the first 2 epochs and decay factor of $\gamma = 0.01$ at epochs 4 and 6. We select the best model among learning rates and training epochs based on the average validation clip accuracy of the two classification heads.

Baselines. We compare our pretraining approach with TAC pretraining. In particular, we consider the following baselines: *TAC on Kinetics*, *TAC on ActivityNet*, and *TAC on THUMOS14*. The models from the second and third baselines are finetuned from a Kinetics-pretrained model.

Target tasks and evaluation metrics. We consider three localization tasks to evaluate TSP pretrained features: TAL on both ActivityNet and THUMOS14, Proposals on ActivityNet, and Dense-Captioning on ActivityNet Captions [35]. For the TAL tasks, the performance is measured using the mean Average Precision (mAP) metric, where a predicted temporal segment is considered a true positive, if it satisfies a temporal Intersection over Union (tIoU) threshold with a ground truth instance of the correct action label. Following standard practice, we use the average mAP over tIoUs $[0.5 : 0.05 : 0.95]$ as the main metric for ActivityNet and the mAP at tIoU=0.5 (mAP@0.5) for THUMOS14. For the Proposals task, the main evaluation metric is the area under the curve (AUC) of the average recall (AR) *vs*. average number of proposals per video. Following common practice in ActivityNet, we limit the number of proposals to 100 per video when computing the AUC. We also report AR at 1, 10, and 100 proposals as additional metrics. Following common practice in the Dense-Captioning task, we use BLEU@3, BLEU@4, and METEOR averaged over tIoUs $[0.3, 0.5, 0.7, 0.9]$ to evaluate performance.

Algorithms for the target tasks. In order to showcase the benefits of TSP pretrained features compared to the baselines, we retrain a variety of state-of-the-art algorithms for each target task atop features extracted from TSP pretrained encoders as well as the baseline encoders. We select the algorithms based on (1) their strong performance on the target tasks and (2) the availability of open-sourced code. Here, we briefly discuss each algorithm and how we apply it to our features. *It is essential to note that we do not innovate in any of these algorithms, and we use their default hyperparameter settings unless otherwise stated below*. We simply swap the visual features they originally use with ours or those of the encoder baselines we compare against. ***G-TAD*** [81]: We

Table 1: **Effects of TSP on target tasks.** We compare features pretrained with our TSP task *vs*. those pretrained with *TAC on Kinetics* and *TAC on ActivityNet*. We use R(2+1)D-34 encoders and pretrain on ActivityNet. We use G-TAD [81], BMN [42], and BMT [27] as algorithms for the ActivityNet TAL, Proposals, and Dense-Captioning tasks, respectively. The column corresponding to the main evaluation metric for each task is highlighted in grey and the best performance is in bold. TSP significantly outperforms the baselines on all tasks.

Video Task	Temporal Action Localization				Action Proposal Generation				Dense Video Captioning		
Feature Pretraining	0.5	0.75	0.95	Avg.	AR@1	AR@10	AR@100	AUC	BLEU@3	BLEU@4	METEOR
TAC on Kinetics	48.54	34.24	7.85	33.32	34.19	57.52	75.56	67.91	3.42	1.58	8.17
TAC on ActivityNet	49.76	34.87	8.65	34.08	34.67	57.89	75.65	68.08	3.63	1.74	8.21
TSP w/o GVF	**51.45**	36.87	9.11	35.75	34.97	**59.35**	76.47	68.88	3.75	1.83	8.42
TSP on ActivityNet	51.26	**37.12**	**9.29**	**35.81**	**34.99**	58.96	**76.63**	**69.04**	**4.16**	**2.02**	**8.75**

use G-TAD for TAL on both ActivityNet and THUMOS14. G-TAD originally uses a Kinetics-pretrained TSN [74] encoder to extract RGB and Flow features, then trains on their concatenation. For G-TAD on THUMOS14, we increase the very small default learning rate by $\times 10$ (*i.e*. to 0.0004) to speed up the training. ***BMN*** [42]: BMN is used for both Proposals and TAL on ActivityNet. BMN did not release code for THUMOS14, and it uses the same precomputed features as G-TAD. ***P-GCN*** [85]: We employ P-GCN for TAL on THUMOS14. P-GCN did not release code for ActivityNet. P-GCN extracts features from an RGB and Flow I3D Kinetics-pretrained encoder. Then, two RGB and Flow localization models are trained *independently* and their results are combined at inference time. We keep the Flow model unchanged and only retrain the RGB model with our features. ***BMT*** [27]: BMT is used for the Dense-Captioning task on the ActivityNet Captions dataset. BMT uses visual and audio features. The visual features are the summation of RGB and Flow features from I3D Kinetics-pretrained encoders, and the audio features are from a VGG-like encoder pretrained on AudioSet [19]. We keep the audio features as is and replace the visual features with ours.

4.2. Ablation Study

Here, we extensively ablate TSP along four dimensions: target localization task, encoder architecture, localization algorithm, and pretraining dataset.

Study 1: Effects of TSP on target tasks. This study aims to compare features pretrained with TSP *vs*. those pretrained with the baselines, *TAC on Kinetics* and *TAC on ActivityNet*, on multiple target tasks. Specifically, we pretrain with the ActivityNet dataset and use an R(2+1)D-34 for the baseline encoders as well as our own. We use G-TAD, BMN, and BMT as the algorithms for the ActivityNet TAL, Proposals, and Dense-Captioning tasks, respectively. Table 1 summarizes the results. ***Observations:*** **(I)** *TAC on ActivityNet* outperforms *TAC on Kinetics* for all three tasks. This makes sense given the fact that the former baseline is pretrained on the same dataset used in the target tasks. However, TSP features consistently show the best performance across all tasks. Specifically, TSP outperforms both baselines by at least +1.73% in average mAP on TAL, +0.96% in AUC on Proposals, and +0.54% in average METEOR on Dense-Captioning. These significant gains underscore the effectiveness of TSP pretraining in encoding better temporal representations for untrimmed videos. **(II)** On the TAL task, TSP features significantly boost performance at high tIoU thresholds (*e.g*. mAP@0.75 is 37.12% for TSP *vs*. 34.87% for *TAC on ActivityNet*). Better mAP at high tIoUs signifies tighter temporal predictions around the ground truth action instances. This indicates that TSP pretrained features can encode better boundary contrast between the action and its nearby background context. **(III)** While TSP pretraining without the GVF (*TSP w/o GVF* in the table) outperforms the baselines, using GVF for the second classification head consistently boosts performance across all tasks (*e.g*. 8.75% *vs*. 8.42% in average METEOR on Dense-Captioning). This validates our design choice and shows the importance of GVF in helping the local features be more temporally-sensitive. Given this observation, we omit *TSP w/o GVF* from the remaining ablation studies. **(IV)** While the Dense-Captioning experiment is conducted on the same pretraining videos, the ActivityNet Captions temporal annotations [35] used for training the Dense-Captioning methods do not necessarily align with the ActivityNet temporal action annotations used for our pretraining. Nevertheless, TSP still provides an improvement over the baselines. **(V)** Table 2 studies the contribution of each TSP classification head to the target task performance. We observe that the performance boost comes mostly from the temporal region head, validating the importance of pretraining on foreground and background clips to attain temporal-sensitivity.

Table 2: **Contribution of each TSP classification head to the target task performance.** We pretrain R(2+1)D-34 on ActivityNet and test the features on ActivityNet TAL using G-TAD [81].

Feature Pretraining	0.5	0.75	0.95	Avg
TSP w/o Temporal Region	49.76	34.87	8.65	34.08
TSP w/o Action Label	51.23	36.79	**9.91**	35.72
TSP	**51.26**	**37.12**	9.29	**35.81**

Study 2: TSP for different video encoders. This experiment explores TSP pretraining for different video architec-

Table 3: **TSP for different video encoders.** We pretrain ResNet3D-18, R(2+1)D-18, and R(2+1)D-34 on ActivityNet and compare the features on the ActivityNet TAL task using G-TAD [81] as the TAL algorithm. Our TSP features consistently outperform the baselines for every encoder type, indicating the generalizability of our pretraining to different backbone architectures.

Backbone Architecture	ResNet3D-18				R(2+1)D-18				R(2+1)D-34			
Feature Pretraining	0.5	0.75	0.95	Avg.	0.5	0.75	0.95	Avg.	0.5	0.75	0.95	Avg.
TAC on Kinetics	47.97	33.21	**8.96**	32.78	47.57	33.11	8.10	32.46	48.54	34.24	7.85	33.32
TAC on ActivityNet	48.71	34.22	8.82	33.40	49.00	34.56	**9.42**	33.87	49.76	34.87	8.65	34.08
TSP on ActivityNet	**49.81**	**34.81**	8.63	**34.10**	**50.07**	**35.61**	8.96	**34.71**	**51.26**	**37.12**	**9.29**	**35.81**

tures. Specifically, we pretrain ResNet3D-18, R(2+1)D-18, and R(2+1)D-34 on ActivityNet and compare the features on the ActivityNet TAL task using G-TAD as the TAL algorithm (refer to Table 3). ***Observations:*** **(I)** We observe similar performance trends among the different pretraining strategies regardless of the encoder type. In particular, our TSP features successfully outperform the baselines for every encoder. This indicates the generalization capability of the TSP pretraining to different backbones. **(II)** Aligned with observations made by previous works [70], R(2+1)D-18 exhibits better performance compared to ResNet3D-18 (average mAP of 34.71% *vs*. 34.10%). **(III)** Not only does the deeper R(2+1)D-34 pretrained with the TSP strategy achieve better performance compared to R(2+1)D-18, but interestingly, the performance gap between TSP and *TAC on Kinetics* widens with the deeper encoder (+2.25% for R(2+1)D-18 *vs*. +2.49% for R(2+1)D-34). Similarly, TSP performance gap with *TAC on ActivityNet* increases from +0.84% for R(2+1)D-18 to +1.73% for R(2+1)D-34. This suggests that our pretraining can potentially show even larger gains for more sophisticated and deeper encoders.

Study 3: TSP with other localization algorithms. We investigate here whether TSP features can consistently improve performance on the target task, regardless of the localization algorithm used. To that end, we conduct the same TAL on ActivityNet experiment from Study 1 (cf. Table 1) but with the BMN algorithm instead of G-TAD. Table 4 summarizes the results using BMN. ***Observations:*** **(I)** Our TSP features used with BMN show similar performance gains as when they are used with G-TAD, with at least a 0.92% gap in average mAP with the TAC-based pretrainings. This demonstrates that our features are more discriminative for the task and that they can benefit different algorithms. **(II)** Both BMN and G-TAD originally use the same features (TSN pretrained on Kinetics) and have a 0.24% gap in average mAP. However, when both are trained using TSP features, BMN bridges the performance gap with G-TAD to be only 0.14%. This highlights the importance of having temporally-sensitive video features for localization tasks.

Study 4: TSP on different datasets. Here, we study two aspects of TSP: its applicability to other pretraining datasets (*i.e*. TSP pretrained on THUMOS14 and tested for TAL on THUMOS14), and its transferability across datasets (*i.e*. TSP pretrained on ActivityNet and tested for TAL on THUMOS14). Specifically, we pretrain R(2+1)D-34 on THUMOS14 and on ActivityNet, then apply P-GCN and G-TAD atop TSP features for the TAL task on THUMOS14. Table 5 compares the two TSP features with the baselines, *TAC on Kinetics* and *TAC on THUMOS14*. ***Observations:*** **(I)** THUMOS14 is different from ActivityNet in two key aspects: THUMOS14 is much smaller, and it has a higher background to foreground ratio (*i.e*. actions are sparser in THUMOS14). Despite these differences, TSP on THUMOS14 improves over the TAC-based baselines by significant margins, regardless of the localization algorithm. Specifically when using P-GCN, TSP on THUMOS14 features improve

Table 4: **TSP with other localization algorithms.** We conduct the same TAL on ActivityNet experiment from Table 1 but with the BMN algorithm instead of G-TAD. Our TSP features achieve the best performance when used with BMN as well.

Feature Pretraining	0.5	0.75	0.95	Avg.
TAC on Kinetics	49.95	35.31	8.61	34.46
TAC on ActivityNet	50.78	35.40	7.96	34.75
TSP on ActivityNet	**51.23**	**36.78**	**9.50**	**35.67**

Table 5: **TSP on different datasets.** We pretrain R(2+1)D-34 on THUMOS14 and on ActivityNet, and use P-GCN [85] and G-TAD [81] for the TAL task on THUMOS14. TSP features are applicable to and transferable across different datasets.

(a) **P-GCN**. Results are reported for the RGB model / RGB+Flow models.

Feature Pretraining	0.3	0.5	0.7
TAC on Kinetics	52.4 / 65.9	37.8 / 49.0	15.6 / 22.9
TSP on ActivityNet	54.2 / 65.4	39.4 / 51.0	14.7 / 22.2
TAC on THUMOS14	54.4 / 66.4	38.7 / 50.0	16.1 / 23.3
TSP on THUMOS14	**58.0 / 69.1**	**44.2 / 53.5**	**18.5 / 26.0**

(b) **G-TAD**

Feature Pretraining	0.3	0.4	0.5	0.6	0.7
TAC on Kinetics	50.6	43.2	34.5	24.1	15.5
TSP on ActivityNet	53.4	45.9	37.0	26.7	16.1
TAC on THUMOS14	52.6	45.5	35.8	26.2	15.6
TSP on THUMOS14	**59.6**	**52.0**	**43.2**	**32.2**	**21.1**

Table 6: **SOTA comparison for TAL and Dense-Captioning.** We compare TSP with SOTA methods for (a) TAL on ActivityNet, (b) TAL on THUMOS14, and (c) Dense-Captioning on ActivityNet Captions. We use G-TAD [81], P-GCN [85], and BMT [27] as the algorithms trained atop our features for each task, respectively. TSP achieves SOTA performance on (a) and (b) and is competitive on (c).

(a) TAL on ActivityNet

Method	0.5	0.75	0.95	Avg.
C-TCN [38]	47.60	31.90	6.20	31.10
P-GCN [85]	48.26	33.16	3.27	31.11
BMN [42]	50.07	34.78	8.29	33.85
GTAN [47]	52.61	34.14	8.91	34.31
PBRNet [44]	**53.96**	34.97	8.98	35.01
G-TAD [81]	50.36	34.60	9.02	34.09
TSP (ours)	51.26	**37.12**	**9.29**	**35.81**

(b) TAL on THUMOS14

Method	0.3	0.4	0.5	0.6	0.7
G-TAD [81]	54.5	47.6	40.2	30.8	23.4
TAL-Net [9]	53.2	48.5	42.8	33.8	20.8
Zhao *et al*. [86]	53.9	50.7	45.4	38.0	28.5
PBRNet [44]	58.5	54.6	51.3	41.8	**29.5**
TSA-Net [22]	65.6	61.4	53.0	**42.4**	28.8
P-GCN [85]	63.6	57.8	49.1	–	–
TSP (ours)	**69.1**	**63.3**	**53.5**	40.4	26.0

(c) Dense-Captioning

Method	B@3	B@4	M
Bi-SST [72]	2.27	1.13	6.10
DVC [39]	2.27	0.73	6.93
MFT [77]	2.82	1.24	7.08
MDVC [28]	2.60	1.07	7.31
SDVC [51]	2.94	0.93	**8.82**
BMT [27]	3.84	1.88	8.44
TSP (ours)	**4.16**	**2.02**	8.75

Table 7: **SOTA comparison for Proposals on ActivityNet**. We use BMN atop our features. TSP significantly improves over BMN original performance and is competitive with SOTA.

Method	[45]	[86]	[5]	[40]	[17]	BMN [42]	**TSP**
AR@100	74.54	75.27	76.73	76.65	**78.63**	75.01	76.63
AUC	66.43	66.51	68.05	68.23	**69.93**	67.10	69.04

Table 8: **SOTA SSL comparison.** We compare TSP with XDC for TAL on THUMOS14. Both use R(2+1)D-18 and G-TAD.

Feature Pretraining	0.3	0.4	0.5	0.6	0.7
TAC on Kinetics	45.4	38.9	30.5	19.7	11.5
TAC on THUMOS14	48.0	41.6	33.3	23.7	14.6
XDC on IG-Kinetics [3]	51.5	44.9	37.2	28.7	**20.0**
TSP on THUMOS14	**57.1**	**50.2**	**41.0**	**30.4**	19.7

the RGB model results by at least 5.5% in mAP@0.5. Moreover, combining the predictions of our newly-trained RGB model with that of the original (unchanged) Flow modality boosts the overall performance by at least 3.5% in mAP@0.5. **(II)** Using TSP features pretrained on ActivityNet (TSP on ActivityNet) outperforms both *TAC on Kinetics* and *TAC on THUMOS14* in mAP@0.5. This shows that TSP features are transferable across TAL datasets.

4.3. State-of-the-Art (SOTA) Comparison

While the previous ablations shed light on the generalization of TSP across multiple tasks, video encoders, algorithms, and datasets, this subsection puts our results in perspective and compares them with SOTA algorithms for each localization task. We report the comparative results in Tables 6 and 7. Note that we build TSP upon the best-performing *publicly available* code for each task, namely G-TAD [81], P-GCN [85], BMN [42], and BMT [27]. In ***TAL on ActivityNet*** (Table 6(a)), we reach SOTA performance with TSP. We achieve 35.81% in average mAP, a boost of 0.80% w.r.t. the previous SOTA PBRNet [44] and a boost of 1.72% w.r.t. our baseline G-TAD [81]. Moreover, TSP (with RGB features only) outperforms SOTA methods [38, 42, 44, 81, 85] that use RGB and Flow features. In ***TAL on THUMOS14*** (Table 6(b)), we achieve 53.5% in mAP@0.5, a boost of 0.5% w.r.t. the previous SOTA TSA-Net [22] and a boost of 4.4% w.r.t. our baseline P-GCN [85]. The results display different improvements on both datasets, focusing on higher tIoU for ActivityNet and lower tIoU on THUMOS14. We argue that this discrepancy originates from the different activity densities in both datasets. In ***Action Proposal Generation*** (Table 7), we reach 69.04% in AUC, a boost of 1.96% w.r.t. our baseline BMN [42], but fall short of RapNet [17] (−0.89%). In ***Dense Video Captioning*** (Table 6(c)), we reached 8.75% in average METEOR, a 0.31% improvement over the baseline BMT [27], but fall short of SDVC [51] (−0.07%). We argue that SDVC [51] uses a reinforcement learning paradigm that optimizes for the METEOR metrics directly, trading off BLEU performances to overfit on METEOR. In contrast, the TSP-empowered BMT model achieves balanced performances in both BLEU and METEOR metrics.

4.4. Comparison with Self-Supervised Encoders

Recent self-supervised learning (SSL) methods have shown impressive performance on video tasks such as action classification [33, 50, 73, 78]. Here, we compare TSP features with SOTA SSL features for temporal localization tasks. Specifically, we compare with XDC [3], a recent SOTA SSL method that learns video and audio features via cross-modal deep clustering. Table 8 compares TSP and XDC features for TAL on THUMOS14 under the same settings: R(2+1)D-18 encoder and G-TAD algorithm. Although XDC impressively outperforms the supervised TAC baselines, it falls short of TSP performance by 2.8% in mAP@0.5. While it is expected that SSL requires more video data for pretraining than supervised pretraining, it is worthwhile to point out that XDC pretrains on 65M videos from IG-Kinetics [20], *i.e*. 260 times more videos than TSP.

Table 9: **Performance as a function of action length**. We report the performance of TAL on ActivityNet for different action lengths. TSP performs significantly better on Extra Short (XS) and Short (S) actions. XS and S make up about 70% of all actions.

Instance Length	XS	S	M	L	XL
% of the Dataset	53.7	16.2	16.8	9.7	4.0
TAC on Kinetics	16.4	41.3	53.2	**68.4**	72.3
TAC on ActivityNet	17.5	42.0	53.1	67.5	**72.5**
TSP on ActivityNet	**19.3**	**44.2**	**53.9**	67.8	71.3

5. Feature Analysis

We further analyze TSP pretrained features on ActivityNet.

DETAD analysis. Following DETAD [1], we analyze the TAL on ActivityNet performance (average mAP) for five different groups of activities based on their length (Table 9): Extra Short (XS: (0s, 30s]), Short (S: (30s, 60s]), Medium (M: (60s, 120s]), Long (L: (120s, 180s]), and Extra Long (XL: $>$ 180s). The extra short instances, XS, are known to be the most challenging to localize [1], and they represent more than half of the annotated instances (53.7%). Their temporal extent is limited as is the information available to recognize the activity. Such instances might be hidden among a significant amount of background. It is clear that TAC performs well in localizing long activities, in particular because they are predominant in their corresponding videos. Yet, TAC achieves the worst performance on the challenging shorter activities. We argue that their localization is more sensitive to the classification of each single clip, since TAC is unaware of what an activity does *not* look like in its temporal surrounding. In contrast, TSP features outperform the TAC ones for the short activity instances (XS and S). We believe our learned clip feature is more aware of background, and thus more perceptive of temporal activity boundaries for localization. As a trade-off, it appears that TSP does not perform as well on extra long activities. We believe those long activities might include intermediate clips with content leaning toward a background activity, thus misleading the localization and resulting in slightly worse performance. Nevertheless, the XL activities merely represent 4.0% of the dataset, so the overall impact on performance is insignificant.

Feature similarity among video clips. Here, we analyze the similarity between video clip features within the same video. We expect the clip features from the same activity to be very similar (consensus), yet very different from the background clips in its temporal surrounding (sharpness). Figure 2 visualizes the cosine similarity between clips of the same video using *TAC on Kinetics vs*. TSP features (more examples are in the *supplementary material*). In (a), *TAC on Kinetics* shows a high similarity between the activity and the background. This will inevitably make localization more difficult. In comparison, TSP better discriminates between background and activity. In (b), it appears that *TAC on Kinetics* is trying to split the activity in two. By learning what background is and what it is not, TSP homogenizes the similarity between all clip features in the foreground activity. In (c), the TSP video encoder increases the differences between background and foreground features. It homogenizes the features within both activities (bottom left and top right corners), yet it does not enforce background features to be similar, resulting in an increase in dissimilarity within the background (see the apparent diagonal in the central square). In (d), *TAC on Kinetics* displays a high similarity for the clips of the foreground activity, yet they might look similar to the remaining background. TSP learns obvious dissimilarity between background and foreground. The TAC pretraining is unaware of the existence of background clips. As a result, it might recognize the class of some actions but is unable to localize them precisely. In contrast, TSP makes the encoder aware of the existence of background, and so the clip features across the video tend to be more informative for localization. Thus, TSP improves the encoder's discriminative ability, reduces the smoothing over the temporal axis, and leads to sharper localization.

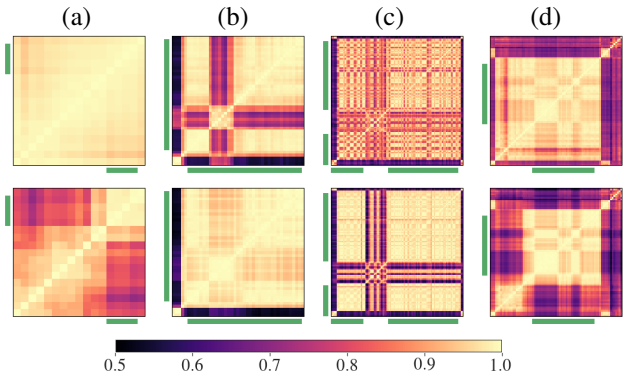

Figure 2: **Feature similarity**. Each column shows the similarity matrices for clips in a single video using *TAC on Kinetics* (top) and TSP (bottom) features. The green lines represent the temporal extent of ground truth actions. Better viewed in color.

6. Conclusion

We present TSP, a novel temporally-sensitive supervised pretraining for video encoders, which not only trains to classify actions, but also considers background clips and global information to gain temporal sensitivity. We show that TSP features improve SOTA methods on the TAL, Proposals, and Dense-Captioning tasks. We argue TSP features can be preferred over other features to build more accurate models.

Acknowledgments. This work is supported the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2017-3405.

References

[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *ECCV*, 2018. 3, 8

[2] Humam Alwassel, Fabian Caba Heilbron, and Bernard Ghanem. Action search: Spotting targets in videos and its application to temporal action localization. In *ECCV*, 2018. 2

[3] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020. 7

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 1

[5] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *ECCV*, 2020. 7

[6] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *CVPR*, 2017. 2

[7] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 4

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

[9] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, 2018. 2, 7

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1

[11] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. In *ICCV*, 2017. 2

[12] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *ECCV*, 2016. 2

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 1

[14] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, 2013. 2

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

[16] Jiyang Gao, Kan Chen, and Ram Nevatia. Ctap: Complementary temporal action proposal generation. In *ECCV*, 2018. 1

[17] Jialin Gao, Zhixiang Shi, Guanshuo Wang, Jiani Li, Yufeng Yuan, Shiming Ge, and Xi Zhou. Accurate temporal action proposal generation with relation-aware pyramid network. In *AAAI*, 2020. 7

[18] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, 2017. 2

[19] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017. 5

[20] Deepti Ghadiyaram, Matt Feiszli, Du Tran, Xueting Yan, Heng Wang, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019. 7

[21] Bernard Ghanem, Juan Carlos Niebles, Cees Snoek, Fabian Caba Heilbron, Humam Alwassel, Victor Escorcia, Ranjay Khrisna, Shyamal Buch, and Cuong Duc Dao. The activitynet large-scale activity recognition challenge 2018 summary. *arXiv preprint arXiv:1808.03766*, 2018. 2

[22] Guoqiang Gong, Liangfeng Zheng, and Yadong Mu. Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. In *ICME*, 2020. 2, 7

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[25] Jingyi Hou, Xinxiao Wu, Wentian Zhao, Jiebo Luo, and Yunde Jia. Joint syntax representation learning and visual cue translation for video captioning. In *ICCV*, 2019. 1

[26] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 1

[27] Vladimir Iashin and Esa Rahtu. A better use of audio-visual cues: Dense video captioning with bi-modal transformer. In *BMVC*, 2020. 1, 3, 5, 7

[28] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *CVPRW*, 2020. 3, 7

[29] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 2017. 4

[30] Mihir Jain, Amir Ghodrati, and Cees G. M. Snoek. Actionbytes: Learning from trimmed videos to localize actions. In *CVPR*, 2020. 2

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

[32] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 3

[33] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 7

[34] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *ICCV*, 2019. 1

[35] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 2, 4, 5

[36] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017. 1

[37] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *AAAI*, 2020. 2

[38] Xin Li, Tianwei Lin, Xiao Liu, Wangmeng Zuo, Chao Li, Xiang Long, Dongliang He, Fu Li, Shilei Wen, and Chuang Gan. Deep concept-wise temporal convolutional networks for action localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020. 2, 7

[39] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Jointly localizing and describing events for dense video captioning. In *CVPR*, 2018. 3, 7

[40] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *AAAI*, 2020. 2, 7

[41] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 1

[42] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, 2019. 1, 2, 3, 5, 7

[43] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018. 2

[44] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *AAAI*, 2020. 1, 2, 7

[45] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *CVPR*, 2019. 7

[46] Ziyi Liu, Le Wang, Qilin Zhang, Zhanning Gao, Zhenxing Niu, Nanning Zheng, and Gang Hua. Weakly supervised temporal action localization through contrast based evaluation networks. In *ICCV*, 2019. 2

[47] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *CVPR*, 2019. 2, 7

[48] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, 2017. 1

[49] Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou. SF-Net: Single-Frame Supervision for Temporal Action Localization. In *ECCV*, 2020. 2

[50] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 7

[51] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined dense video captioning. In *CVPR*, 2019. 3, 7

[52] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Efficient action localization with approximately normalized fisher vectors. In *CVPR*, 2014. 2

[53] Boxiao Pan, Haoye Cai, De-An Huang, Kuan-Hui Lee, Adrien Gaidon, Ehsan Adeli, and Juan Carlos Niebles. Spatio-temporal graph for video captioning with knowledge distillation. In *CVPR*, 2020. 1

[54] Alejandro Pardo, Humam Alwassel, Fabian Caba Heilbron, Ali Thabet, and Bernard Ghanem. Refineloc: Iterative refinement for weakly-supervised action localization. In *WACV*, 2021. 2

[55] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, 2018. 2

[56] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *CVPR*, 2019. 2

[57] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *CVPR*, 2018. 2

[58] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *ICCV*, 2019. 3

[59] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1

[60] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1

[61] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. Dense procedure captioning in narrated instructional videos. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. 3

[62] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 2017. 2

[63] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, 2018. 2

[64] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016. 2

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

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

[67] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, 2020. 1

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

[69] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, 2019. 1

[70] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 2, 4, 6

[71] Bairui Wang, Lin Ma, Wei Zhang, Wenhao Jiang, Jingwen Wang, and Wei Liu. Controllable video captioning with pos sequence guidance based on gated fusion network. In *ICCV*, 2019. 1

[72] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*, 2018. 3, 7

[73] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, 2019. 7

[74] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 2, 5

[75] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, 2018. 2

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

[77] Yilei Xiong, Bo Dai, and Dahua Lin. Move forward and tell: A progressive generator of video descriptions. In *ECCV*, 2018. 3, 7

[78] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, 2019. 7

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

[80] Huijuan Xu, Kun He, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Text-to-clip video retrieval with early fusion and re-captioning. In *AAAI*, 2019. 2

[81] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020. 1, 2, 3, 4, 5, 6, 7

[82] Chenggang Yan, Yunbin Tu, Xingzheng Wang, Yongbing Zhang, Xinhong Hao, Yongdong Zhang, and Qionghai Dai. Stat: spatial-temporal attention mechanism for video captioning. *TMM*, 2019. 2

[83] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016. 2

[84] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, 2016. 1

[85] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, 2019. 1, 2, 3, 5, 6, 7

[86] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *ECCV*, 2020. 1, 7

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

[88] Qi Zheng, Chaoyue Wang, and Dacheng Tao. Syntax-aware action targeting for video captioning. In *CVPR*, 2020. 1

[89] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018. 3