Less than Few: Self-Shot Video Instance Segmentation

Pengwan Yang[1,2], Yuki M. Asano[1,2], Pascal Mettes[2], and Cees G. M. Snoek[1,2]

[1] QUVA Lab, University of Amsterdam, Amsterdam, The Netherlands
[2] Institute of Informatics, University of Amsterdam, Amsterdam, The Netherlands
yangpengwan2016@gmail.com

Abstract. The goal of this paper is to bypass the need for labelled examples in few-shot video understanding at run time. While proven effective, in many practical video settings even labelling a few examples appears unrealistic. This is especially true as the level of details in spatio-temporal video understanding and with it, the complexity of annotations continues to increase. Rather than performing few-shot learning with a human oracle to provide a few densely labelled support videos, we propose to automatically learn to find appropriate support videos given a query. We call this self-shot learning and we outline a simple self-supervised learning method to generate an embedding space well-suited for unsupervised retrieval of relevant samples. To showcase this novel setting, we tackle, for the first time, video instance segmentation in a self-shot (and few-shot) setting, where the goal is to segment instances at the pixel-level across the spatial and temporal domains. We provide strong baseline performances that utilize a novel transformer-based model and show that self-shot learning can even surpass few-shot and can be positively combined for further performance gains. Experiments on new benchmarks show that our approach achieves strong performance, is competitive to oracle support in some settings, scales to large unlabelled video collections, and can be combined in a semi-supervised setting. Code: https://github.com/PengWan-Yang/self-shot

1 Introduction

The goal of this paper is to decrease the reliance on humans to provide labelled examples in few-shot video understanding. While impressive few-shot video classification [38,11,56], localization [26,71,69] and detection [25,70] results have been reported, in many practical video settings even labelling a few examples may appear unrealistic. This is especially true as the level of spatio-temporal video understanding and with it, the complexity of annotations continues to increase. Consider for example the problem of video instance segmentation [68,9,64], where datasets for example contain around 1.6K annotated frames for just a single object class. We deem it unlikely that an interacting user, that is looking to segment a "query" video with unknown instances, is willing to provide pixel-precise annotations masks for all objects in a frame for each video in the support set, despite this being a setting which is typical for more classical, image-based few-shot learning scenarios. Thus, rather than relying on a human oracle to provide a few densely labeled support videos, we propose to automatically learn to find appropriate support videos given a query.

For this, we introduce the notion of *self-shot* learning, in which the need for labelled video clips at test-time is abolished. Instead, one is provided with a large unlabelled pool of videos from which samples potentially relevant to the query video can be retrieved and utilized in a strictly unsupervised fashion. We address this by adapting a simple self-supervised learning method [60] to generate an embedding space well-suited for unsupervised retrieval of relevant samples. To showcase this novel setting, we go beyond just bounding box detection and temporal localization and instead tackle, for the first time, *video instance segmentation* in a self-shot (and few-shot) setting, where the goal is to segment instances at the pixel-level across the spatial and temporal domains.

Overall, we make three contributions in this paper:

1. We propose the setting of self-shot learning. While annotations are used during training (similar to few-shot), at test-time, new classes are evaluated *without* any annotations, but with access to an unlabelled dataset.
2. We investigate this new setting for a particularly annotation-heavy scenario, that of video instance segmentation, for which we propose new splits to establish a self-shot (and few-shot) benchmark.
3. Finally, we provide strong baseline performances that utilize a novel transformer-based model and show that self-shot learning can even surpass few-shot and can be positively combined for further performance gains.

2 Related work

Video few-shot learning. There is limited related work on the few-shot learning setup for videos. Initial works have explored few-shot learning for the task of video classification [38,73]. For example, OSS-Metric Learning [38] measures similarity of pairs of video to enable few-shot video classification. Yang *et al*. [67] introduce few-shot action localization in time, where a few positive labelled and several negative labelled support videos steer the localization via an end-to-end meta-learning strategy. Xu *et al*. [66] and Zhang *et al*. [71] also perform few-shot temporal action localization with the assistance of video-level class annotations. To further free the need for labels, a new research line is emerging, called few-shot common action localization, where the common action in a long untrimmed query video is localized in time [26,69] or both in time and in space [25,70] based on a few support videos containing the same action. Wang *et al*. [62] segment objects that simultaneously exist in multiple individual videos. However, all of the input videos need to contain exactly the same object instances, which is not necessary in our self-shot setting where relevant support videos can be self-retrieved. Closest to our work is few-shot spatio-temporal action localization by Yang *et al*. [70], who adopt a transformer-based action detection architecture and extend to localizing actions at pixel level. They propose a mask head upon the action detection boxes to perform the binary classification for the pixels inside each detected box. In this paper, we propose the new task of self-shot video instance segmentation that operates on objects instead of actions, predicts the segmentation directly, removes the need for having predefined (labelled) support videos, and encapsulates the few-shot setting as a special case.

Video zero-shot learning. Various video tasks have been investigated from a zero-shot perspective. Zhang *et al.* [72] can localize the unseen activities in time in a long untrimmed video based on the label embeddings. Spatio-temporal action localization is also explored in zero-shot setting by linking actions to relevant objects [34,45,46], or by leveraging trimmed videos used for action classification [35]. Wang *et al.* [63] achieve zero-shot video object segmentation by proposing a novel attentive graph neural network which can iteratively fuse information over video graphs. Lu *et al.* [44] can distinguish foreground/background in a zero-shot manner, but it relies on supervised prior knowledge, *e.g.* class activation maps obtained from a pre-trained image classifier. Dave *et al.* [22] can segment the moving objects in videos, even ones unseen in training. Just like the zero-shot setting, we also aim to segment unseen object instances in videos, without any labelled support videos. But we try to leverage self-retrieved (free) support videos to boost the performance.

Self-supervised learning. Self-supervision has been proposed as a method to obtain feature representations without labels. This has been accomplished by geometric pretext tasks [28,50,52], clustering [5,13,14,27] or more recently contrastive [36,19,49,17,65] and teacher-student approaches [15,29]. These have also been extended to the video domain [24,53,54,58,39,4,2,8,30,37]. In this work we use self-supervised learning to construct an embedding space well-suited for retrieving semantically relevant videos to support video instance segmentation. Note that this use of support samples is fundamentally different to how it has been used in other self-supervised works such as [23] or [55], where they are only used as random subsets to aid contrastive learning. Instead, support samples are the goal of our self-shot method. To this end, we compare the use of noise contrastive methods [65] and the differentiable ranking loss [60] extended to the video domain.

Video instance segmentation (VIS). VIS [68,9] requires classifying, segmenting, and tracking instances over all frames in a given video. With the introduction of the YouTube-VIS dataset, which contains dense pixel-level annotations across consecutive frames [68], considerable progress has been made in tackling this challenging task. State-of-the-art methods typically develop sophisticated pipelines and rely on heavy supervision and complex heuristic rules to associate the instances across frames. As two representative methods, MaskTrack R-CNN [68] extends the Mask R-CNN model [32] with a pair-wise identity branch to solve the instance association problem in VIS, while MaskProp [9] introduces a multi-stage framework [16] for propagating instance masks in time. In contrast, VisTR [64] builds a DETR-based pipeline [12] for the VIS task in a query-based end-to-end fashion, which can supervise and segment the instances across frames as sequences. In this paper we adopt the spirit of VisTR to treat instance segmentation in a query video as a sequence prediction problem. Different from the usual VIS task, our self-support video instance segmentation task focuses on recognizing, segmenting, and tracking the instances in a query video containing novel classes – from just a few retrieved support videos and without knowing any annotations.

3 Problem definition and benchmarking

3.1 Task definition

Our goal is video instance segmentation in a query video without having access to any labelled training examples with the same instances as in the query. Instead, we consider a self-shot scenario where we have an unlabelled pool of videos that we can use to help guide the instance segmentation. To that end, we denote a set of seen classes as $\mathcal{S}$ and a disjoint set of unseen classes as $\mathcal{U}$, where $\mathcal{S} \cap \mathcal{U} = \emptyset$. Let $\mathcal{D}_{\mathcal{S}} = \{(x, y)|x \in \mathcal{X}, y \in \mathcal{Y}^{\mathcal{S}}\}$ represent the set of labelled training data on seen classes, where x is the pixel-wise feature embeddings from the visual space $\mathcal{X}$, y is the corresponding pixel-wise label in the label space $\mathcal{Y}^{\mathcal{S}}$ of seen classes. $\mathcal{D}_{\mathcal{U}}$ denotes the set of unlabelled videos on unseen classes. Our self-shot learning shares with few-shot and zero-shot learning the same goal to learn a model from $\mathcal{D}_{\mathcal{S}}$ and predict the label of each pixel for videos in $\mathcal{D}_{\mathcal{U}}$. However, they differ in their objective and expected data availability:

Few-shot learning. For each unseen class $c \in \mathcal{U}$, a handful of predefined support videos $\mathcal{V}_c^k$ are provided, where k is small. Then for each query video $Q_c \in \mathcal{D}_{\mathcal{U}}$, the small set of support videos containing exactly the same class $\mathcal{V}_c^k$ function as guidance videos to predict a segmentation for the unseen object class c.

Zero-shot learning. In the most conventional zero-shot strategy [72,34,46,40], all class labels $\mathcal{C}$=$\mathcal{S} \cup \mathcal{U}$ are provided and mapped through semantic embeddings to vector representations $\{v_c|c \in \mathcal{C}\}$. Then a joint visual-text perspective helps the model learned on the seen classes generalize to the unseen classes.

Self-shot learning. Instead of predefined supports or semantic class labels, self-shot learning relies on an unsupervised manner to retrieve support videos $\{\mathcal{V}_{Q_c}|\mathcal{V}_{Q_c} \in \mathbb{S}\}$ for each query video $Q_c, c \in \mathcal{U}$ from a collection of unlabelled videos $\mathbb{S}$. It leverages the discovered support videos as guidance for predicting an instance segmentation. Self-shot learning can be viewed as a framework to obtain noisy few-shot examples without the need for human annotations.

3.2 Datasets

Since self-shot video instance segmentation is a new task, we set up two benchmarks through the reorganization of two existing (many-shot) video instance segmentation datasets, namely YouTube-VIS [68] (2021 version) and OVIS [57].

Self-VIS. YouTube-VIS contains 2,985 videos in the training set where the instance mask annotations are publicly available. The annotated instances cover 40 instance categories and a minority of the videos have instances of more than one classes. To build a setting with videos containing one singe instance class, we discard videos with multiple instance classes and obtain a total of 2,123 videos. We randomly select 30 classes for training and 10 classes for validation and testing.

Self-OVIS. Occluded VIS (OVIS) provides 607 videos with annotated instance masks. Among the 25 instance categories in OVIS, 17 are for training and 8 for validation and testing. With more instances of multiple classes per video, and more frequent occlusions, the setting of OVIS is much harder than the one of YouTue-VIS. More details are provided in Table A in the Appendix. During training, the query video and support videos are randomly paired according to the common instances present, while the pairs are fixed for validation and testing for reproducibility.

YouTube-8M Segments. The YouTube-8M Segments dataset is a subset of the YouTube-8M dataset proposed in the same paper [1]. It contains about 237K 5-second videos extracted from around 50K source videos and while it contains annotations, we do not use any of the labels. Instead, we adopt YouTube-8M Segments as our unlabelled video database $\mathbb{S}$ for self-shot retrieval and call the self-shot benchmarks

4 Finding support videos through self-shot learning

The purpose of self-shot learning is to retrieve videos from a large, unlabelled video dataset that will aid in performing inference on the query video, specifically for the task of instance segmentation in this paper. To this end, we train an encoder that will yield an embedding space well suited for video retrieval by adopting components of self-supervised representation learning methods MoCo v1 to v3 [31,18,20], multiple-instance NCE [47] and self-supervised ranking [60].

For self-shot learning, we are given an unlabelled video collection $\mathbb{S}$. Each clip is encoded by two visual encoders, Φ and $\tilde{\Phi}$, where $\tilde{\Phi}$ is updated as the exponential moving-average of Φ as in [31,20]. With this setup we evaluate self-shot retrieval with two different losses: noise-contrastive instance discrimination and ranking.

The contrastive loss $\mathcal{L}_{\text{NCE}}$ in our case is given by setting positive pairs to be different temporal crops of a single video, while negative pairs are constructed from other instances of the dataset. Let $V_i \in \mathbb{S}$ denote a single unlabelled video and let $v \in V_i$ denote one of its temporal crops. Then we naturally arrive at the following multiple-instance NCE [47] formulation:

$$\mathcal{L}_{\text{NCE}}(v) = -\log \frac{\sum_{v^+} \exp\langle \Phi(v) \cdot \tilde{\Phi}(v^+) \rangle_\tau}{\sum_{(v^+ \cup v^-)} \exp\langle \Phi(v) \cdot \tilde{\Phi}(v^+) \rangle_\tau)}, \tag{1}$$

where $\langle \cdot, \cdot \rangle_\tau$, is a temperature-scaled dot-product, and v^+ is the positive set defined as $V_i \setminus v$ and v^- is the negative set, corresponding to crops from other videos in $\mathbb{S}$.

We further experiment with transplanting a differentiable ranking loss from [60], to the setting of using two encoders. The ranking loss is a less aggressive form of enforcing self-invariance than the NCE loss, and is given by learning an embedding space in which a set of positive videos is ranked *above* a set of negative videos, when comparing distances in feature space. More precisely,

$$\mathcal{L}_{\text{Rank}}(v) = -\log \sum_{v^+} \frac{R_{\Phi(v)}(\tilde{\Phi}(v), \tilde{\Phi}(v^+))}{R_{\Phi(v)}(\tilde{\Phi}(v), \tilde{\Phi}(v^+) \cup \tilde{\Phi}(v^-))}, \tag{2}$$

where $R_a(b, c)$ is a differentiable function to rank video b among all videos in the set $\{c\}$ with respect to the query video a [10,60].

Once the feature spaces are learned, the final step is **retrieving relevant support videos using only the query itself**. For query video q, we use the self-supervised trained encoder Φ and a simple k-nearest neighbor (kNN) approach. The self-shot support videos for query q is obtained as:

$$\text{self-shot}(q) = k\text{NN}(\Phi(q), \Phi(\mathbb{S})). \tag{3}$$

Note that this simple kNN approach allows us to use self-shot learning as a *plug-and-play* component, which can be used to replace supervised support videos or to extend these in a semi-supervised manner. This setup is generally applicable, we focus on video instance segmentation as testbed in this paper, due to the hefty annotation demand for supervised settings.

Implementation details. We follow the uniform frame sampling method in [3] for mapping a video to a sequence of tokens of a Vision Transformer of size B (ViT-B). In each mini-batch, we use 160 5×224×224 video segments from 32 videos. The patch-size of ViT-B is 16×16. We keep the memory queue [31] and the length is 1280. We train for 40 epochs with the AdamW optimizer [43], with an initial learning rating of 10^{-4}, which we decay by 10 at epoch 25. Further details are provided in the Appendix.

5 A Self-Shot and Few-Shot VIS Transformer

Equipped with self-shot learning, we examine a research problem with a high annotation cost: video instance segmentation. Hence our goal is to segment and track the object instances of interest in a query video. We do however not assume access to training videos labelled with the same instance classes, temporal boundaries, or mask annotations as support for the query video at test-time. In essence, the query video is on its own and we are instead given a collection of unlabelled videos. We seek to find a few unlabelled support videos from this collection through self-shot learning. While we aim for self-shot learning, few-shot video instance segmentation using a few support videos has not yet been investigated either. Hence, we first introduce a baseline sequence-to-sequence transformer approach to solve video instance segmentation given semantically similar (support) videos. At the core of our approach is a common instance segmentation transformer, which contains three stages with three functions: (i)

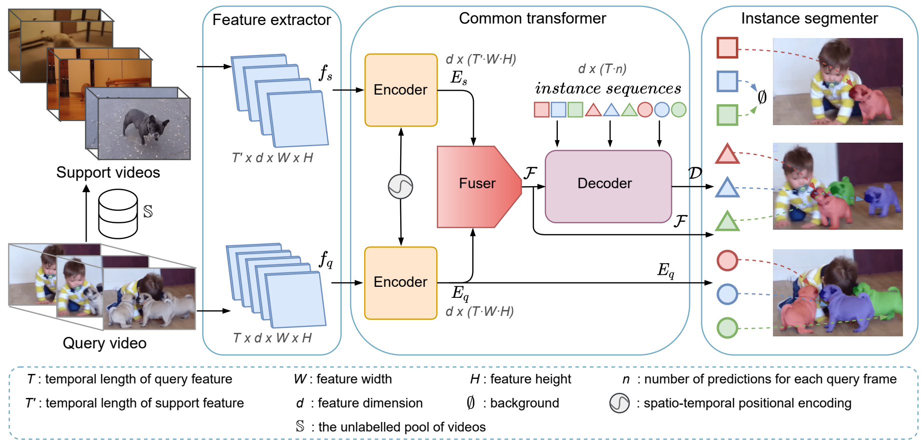

Fig. 1: Overview of the self-shot and few-shot VIS transformer. In the feature extraction stage, given a query video and a handful of (self-shot) support videos, the backbone extracts features of individual image frames, then the image features are concatenated in the frame order to form clip-level features for the query and support videos. In the common transformer, the encoder models the pixel-level similarity for the query and support features respectively, the fuser leverages the similarity between the query feature and the support feature, and the decoder learns the similarity between instances along the time dimension. In the prediction stage, the instance sequences are inferred in the query video

encoding and extracting of features for query and support videos, (ii) learning of pixel-level similarities for the query features by leveraging cross-attention, and (iii) prediction of instance mask sequences across space and time through decoding. Each step is detailed below.

Feature extraction. We adopt a modified ResNet-50 [33] with a bigger receptive field for feature extraction, with the complete architecture given in the Appendix. We let a single query video and a few support videos go through the backbone to extract the pixel-level image frame feature sequences. Assume that the query video contains T frames and the support videos contain T' frames in total. The backbone generates a lower-resolution activation map for each frame in the query video and support videos, then the frame features are concatenated to form video clip level features for the query and support videos. The query video feature is denoted as $f_q \in \mathbb{R}^{T\times d\times W\times H}$ and the support feature is $f_s \in \mathbb{R}^{T'\times d\times W\times H}$. The weights of the backbone are shared between the query and support videos.

Stage 1: Spatio-temporal transformer encoder. We first feed the extracted video features into the transformer encoder structure and flatten the spatial and temporal dimensions of f_q and f_s in 2D feature maps of size $d\times(T{\cdot}W{\cdot}H)$ and $d\times(T'{\cdot}W{\cdot}H)$. Since the image-encoder based backbone is permutation-invariant, we append spatio-temporal positional encodings [7,51] to the inputs as the instance segmentation task requires precise spatial and temporal information.

Specially, for all spatio-temporal coordinates of each dimension, we independently use $d/3$ sine and cosine functions with different frequencies. We then concatenate them to get the final d channel encoding. The spatio-temporal positional encodings are added to both the query feature and support feature in each encoder layer. The output of this transformer encoder structure for the query branch is $E_q \in \mathbb{R}^{d\times(T\cdot W\cdot H)}$, and the output for the support branch is $E_s \in \mathbb{R}^{d\times(T'\cdot W\cdot H)}$. The encoder weights are also shared.

Stage 2: Query-support fuser. Given encoded query and support videos, we seek to discover similarities in space and time by integrating the support branch into the query branch, by utilizing the attention mechanism. Let $\mathbb{MA}$ denote the multi-headed attention with linear projection function $Q(\cdot)$, $K(\cdot)$, $V(\cdot)$ as described in [61]. We first cross-enhance the fuser inputs E_q and E_s through multi-head attention, as shown in Figure A in the Appendix: $f_{q\leftarrow s} = \mathrm{LN}(E_q + \mathbb{MA}(Q(E_q), K(E_s), V(E_s))$, and similarly for $f_{s\leftarrow q}$. Here, LN denotes the layer normalization operation [6]. Next, the support branch is fused into the query branch to get the fused feature $\widetilde{F}$:

$$\widetilde{F} = \mathrm{LN}(f_{q\leftarrow s} + \mathbb{MA}(Q(f_{q\leftarrow s}), K(f_{s\leftarrow q}), V(f_{s\leftarrow q})). \tag{4}$$

In addition, a feed-forward network (2-layer MLP) is applied to $\widetilde{F}$ in a residual fashion for increased modelling ability, yielding output of the fuser: $\mathcal{F} = \mathrm{LN}(\widetilde{F} + \mathrm{FFN}(\widetilde{F}))$.

Stage 3: Decoding and predicting. The spatio-temporal decoder aims to decode the most discriminative pixel features that can represent the instances of each frame. We introduce a fixed number of input embeddings to represent the instance features across time and space, which we call *instance sequences*. Assuming that the model decodes n instances per frame, the number of instances for the T frames in the query video is $N{=}n{\cdot}T$. The *instance sequences* are learned by the spatio-temporal decoder and take the output of the query-support fuser $\mathcal{F}$ and *instance sequences* as input, to outputs N instance features, denoted as $\mathcal{D}$, as shown in Figure 1. Finally, the instance segmenter predicts the mask sequence for each instance. For each frame in the query video, we feed the instance features $\mathcal{D}$ and the fused feature $\mathcal{F}$ into an attention module to obtain the attention maps. The attention maps are then concatenated with the encoded query feature E_q and the fused feature $\mathcal{F}$, followed by a deformable convolution layer [21]. In this way, we obtain the mask features for each instance of the different frames in the query video. We denote the mask feature for instance i of frame t is $g_{i,t} \in \mathbb{R}^{1\times a\times W_0\times H_0}$, where a is the channel number, W_0 and H_0 are the feature width and height. Finally, the instance segmenter uses the accumulated features to output the mask sequence for each instance (see Appendix for details). and outputs the mask sequence $m_i \in \mathbb{R}^{1\times 1\times T\times W_0\times H_0}$ for the instance i directly.

Training loss. To score predicted instances with respect to the ground truth, we introduce an optimal bipartite matching between predicted and ground truth instances, in the spirit of [12,64] (see Appendix for details). Given the optimal assignment, the next step is to compute the training loss $\mathcal{L}_{\text{train}}$, which is a linear combination of a negative log-likelihood for *foreground/background* prediction, a box loss and mask loss for the instance sequences:

$$\begin{aligned}\mathcal{L}_{\text{train}}(y, \hat{y}) = \sum_{i=1}^{n} [(-\log \hat{p}_{\hat{\sigma}(i)}(c_i)) + \\ \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}(i)}) + \lambda_{\text{mask}} \cdot \mathcal{L}_{\text{mask}}(m_i, \hat{m}_{\hat{\sigma}(i)})],\end{aligned} \tag{5}$$

here c_i=*foreground*, and $\hat{\sigma}$ is the optimal assignment, $\hat{p}_{\sigma(i)}(c_i)$ denotes the probability of c_i with index $\sigma(i)$, b_i denotes the ground truth box sequences. This training loss is used to train the whole video instance segmentation framework end-to-end. For the bounding box loss we employ the generalize IoU loss as prescribed in [59], while we use a linear combination of the dice loss [48] and focal loss [41] for the mask loss. The full loss equations are provided in the Appendix. As a result we obtain an end-to-end framework that is guided by support videos and able to segment instances in a query video.

Implementation details. As the largest video length in Self-VIS is 32, we take 32 as the query video clip length T. The support video clip length is set to 24. If the original video is too short or too long, we pad it with the last frame or cut it at a random position. All videos are resized to a 320×280 resolution before they are fed into the backbone. The model predicts 10 instances for each query frame, thus the total instance number is 320. In the common transformer structure, we use 6 encoder, 3 fuser, 6 decoder layers of width 288 with 8 attention heads. The model is trained with AdamW [43], setting the initial common transformer's learning rate to 10^{-4}, the backbone's learning rate to 10^{-5}. The model is trained for 20 epochs, with the learning rate decay by 10 at 14 epochs. We initialize our backbone with the weights pretrained on the COCO dataset [42]. Further details are provided in the Appendix.

The main evaluation metric is average precision, with the video Intersection over Union (IoU) of the mask sequences as threshold. The IoU threshold is set to 0.5 unless specified otherwise.

6 Results

Self-shot evaluation. We first evaluate the potential of self-shot learning on the introduced video instance segmentation benchmarks. We use the introduced transformer and compare to both oracle upper bounds and self-supervised baselines for obtaining support videos as input to the transformer.

In Table 1, we compare a broad range of feature spaces for providing relevant support videos given a query video. We first find that utilizing random videos from the unlabelled dataset as support already provides a non-trivial instance segmentation performance (row *a*), but still with a considerable gap to the oracle baseline where each support video is manually curated to have matching instance classes with the query video. Self-shot learning brings large benefits over randomly picking support videos (rows *(c)-(f)*). We first establish a baseline of using ImageNet-pretrained features for obtaining relevant support samples

Table 1: Self-shot evaluation. The unsupervised support videos come from the Youtube-8M Segments dataset and video instance segmentation performance is evaluated on the test set of Self-VIS. For comparability, we include baselines using random videos or fully supervised oracle videos as support. Inference-time support-increase is evaluated in the $5+(n)$ columns where extra n support videos are used at inference. The strongest self-support is competitive with a 1-shot oracle-support

		Support				
		1	5	5+(1)	5+(3)	5+(5)
Retrieval based on						
(a)	Random videos	44.3	44.9	44.8	45.1	45.1
(b)	Oracle/labels	53.2	56.6	56.1	57.6	57.8
Self-shot variants						
(c)	Inception fixed	46.9	48.3	48.5	48.2	48.4
(d)	Inception MoCo	49.4	51.6	51.9	51.7	51.9
(e)	Video MIL-NCE	50.1	52.5	52.6	52.9	53.3
(f)	Video Rank	51.4	54.3	54.6	55.2	55.4

in row *(c)*, as well as finetuning these features using a non-parametric instance retrieval loss [65] using MoCo [31] in row *(d)*, which adds around 2.5-3.5% in performance. Next, we utilize a more sophisticated MIL-NCE loss [47] and a rank-based retrieval loss [60] to learn *video-clip* embeddings (see Appendix for details). With this we find that row *(f)* in Tab. 1 achieves strong gains of 6-10% in absolute performance compared to the random baseline and more than 3-7% compared to frozen ImageNet features and use this for subsequent experiments. Besides the finding that all features obtained in a self-supervised fashion improve over the supervised frame-based features ones we also establish that the self-supervised task does matter too, as we find the ranking loss well-suited for retrieving relevant support videos. We conclude that the strongest self-support can almost close the gap with the oracle-support baseline, even though self-shot support videos are not guaranteed to have matching classes.

Self-shot versus zero-shot learning. To further quantify the effectiveness of self-shot learning for finding support videos, we compare to zero-shot learning on two different tasks: video instance segmentation and temporal action localization. For video instance segmentation, all experiments run on the Self-VIS dataset and as zero-shot baselines we utilize the methods from Dave *et al*. [22] and Lu *et al*. [44]. Dave *et al*. can segment moving objects in videos, even the ones unseen in training. Lu *et al*. can distinguish foreground/background in a zero-shot manner. For temporal action localization, we follow the setup of Yang *et al*. [69], which also provides the temporal action localization pipeline and the reorganized dataset derived from Thumos14. Zhang *et al*. [72] provide the zero-shot method for temporal action localization. They can localize the unseen activities in a long untrimmed video based on the label embeddings, which means class labels are needed during inference. In Table 2, we report the video instance segmentation and temporal action localization results. We find that on both

Table 2: Self-shot versus zero-shot learning for video instance segmentation and temporal action localization. For Temporal action localization, we follow the setup of Yang *et al*. [69] on the Thumos14 dataset. The metric is video-mAP with an overlap threshold of 0.5. We find that our self-shot perspective is better suited for segmentation and localization in videos than zero-shot baselines

	Self-VIS			**Thumos14**		
	0	1	5	0	1	5
Zero-shot learning [22]	47.4	-	-	-	-	-
Zero-shot learning [44]	48.1	-	-	-	-	-
Zero-shot learning [72]	-	-	-	43.4	-	-
Self-shot learning	-	51.4	54.3	-	45.8	47.3
Self-shot learning k+(5)	-	54.6	55.4	-	47.7	48.0

Table 3: VIS transformer ablation under self-shot setting. The decoder achieves competitive performance by itself. It improves when the encoder processes the videos. A considerable performance gain happens when the fuser passes messages from the support branch to the query branch. Performance is best with all three modules

Encoder	**Fuser**	**Decoder**	**Self-VIS**		**Self-OVIS**	
			1 self-shot	5 self-shots	1 self-shot	5 self-shots
		✓	40.7	40.8	13.7	14.4
✓		✓	42.4	42.2	14.3	15.7
	✓	✓	49.1	51.5	19.6	21.0
✓	✓	✓	**51.4**	**54.3**	**20.6**	**23.7**

settings our self-shot approach improves over the zero-shot alternatives, indicating that automatically obtaining support videos in an unsupervised manner and using them for their respective video task obtains favorable results over a semantic transfer of information from seen to unseen visual classes. In the Appendix, we also provide results where we use zero-shot learning to help find support videos, which is also not as effective as self-shot learning.

Transformer ablation. In Table 3, we show the effect of the three components in our video instance segmentation transformer when using both one and five self-shot support videos. We find that all three components matter for maximizing segmentation performance. Especially the introduced fuser module is important and adds 8.4% 1-shot performance compared to the baseline, while the encoder adds 1.7%. When combined, the encoder yields an additional 2.3% gain on top of the fuser-only baseline, showing that having sufficient encoding capacity before fusing leads to better performance. In the appendix we also establish that the proposed baseline is competitive both against previous works adapted for the novel video instance segmentation setting, as well as against established methods for an image segmentation setting. Overall, we conclude that the proposed transformer is effective for video instance segmentation using a few support videos.

Table 4: Video difficulty ablation. (a): Performance when varying the number of instances on Self-VIS. High scores can be obtained when the common instances are not too many (no more than 3), segmentation of more than 4 instances in a query video remains challenging. **(b)**: Performance when varying the number of classes on Self-OVIS. When the common instances come from multiple classes, segmentation becomes harder

(a) Instance number in query video

	Instance number				
	1	2	3	4	≥5
1 self-shot	52.1	51.7	49.5	44.7	41.6
5 self-shots	54.8	54.6	52.9	50.2	45.5

(b) Class number in query video

	Class number			
	1	2	3	≥4
1 self-shot	23.4	20.7	17.1	8.6
5 self-shots	26.2	23.3	19.4	9.4

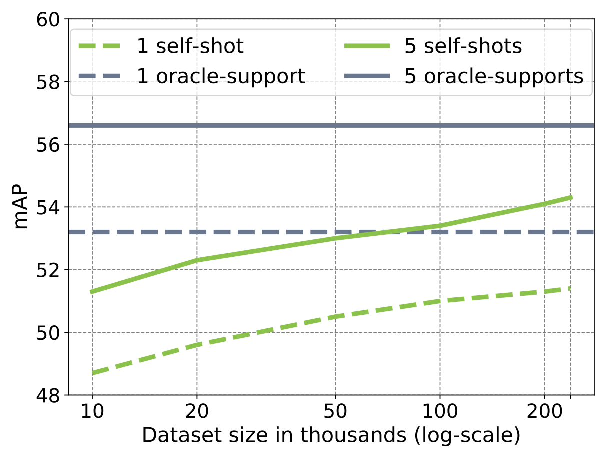

Fig. 2: Scalability of self-shot learning. Performance scales positively with the increase in unlabelled videos available and 5 self-supports outperforms using 1 oracle-support for >75K videos

Video difficulty ablation. Next, we ablate the effect of the number of instances and number of classes in the query video on the segmentation performance in Table 4. As each video in the Self-VIS dataset contains instances from just a single class, we use this to analyze the stability of our model with regard to the number of instances. The result is shown in Table 4a, and we find performance only mildly declines for a moderate increase, up to 3, in number of instances in the query video. Next, we use the more difficult Self-OVIS dataset to study the robustness against more diverse videos that contain more instances from multiple classes. As shown in Table 4b, the method still works well with more than one instance per video, though performance naturally declines with this added difficulty for the task.

Scaling unsupervised support. The quality of the self-shot learning is bounded by the quality of the videos in the unlabelled video collection. In Figure 2

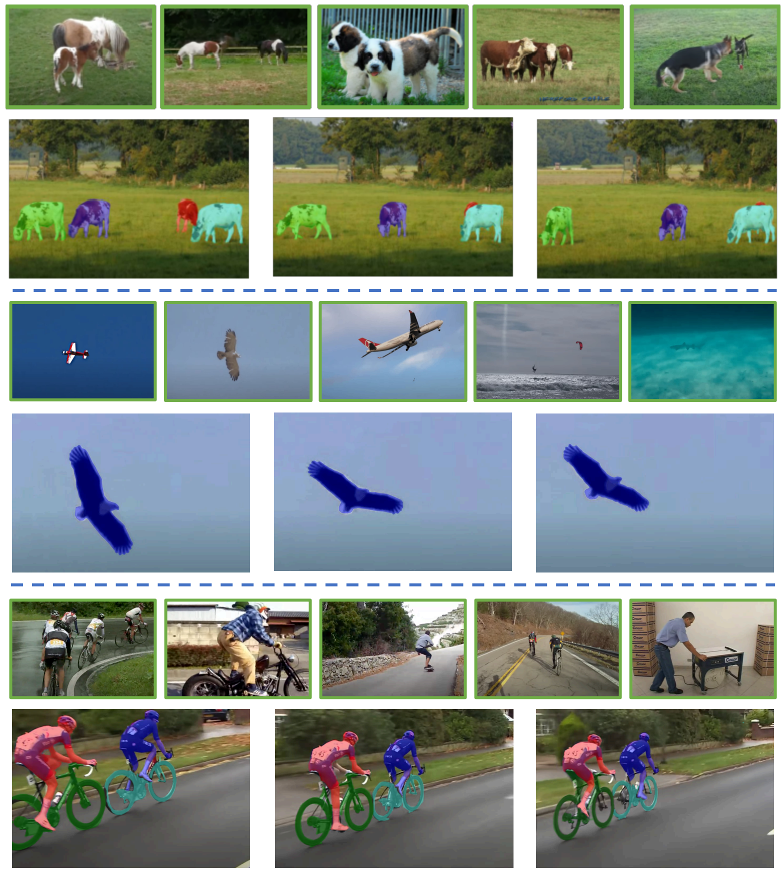

Fig. 3: Qualitative examples. Three examples of self-shot videos (top) and the resulting instance segmented query video (bottom)

we show how self-shot video instance segmentation scales with unlabelled dataset size. We find that at around 75K videos, our self-shot approach with 5 videos already *outperforms the 1-shot oracle-support* baseline. While we cannot perform experiments using even larger dataset sizes, we can see that even at 237K videos, the performance is still rising steeply on a typical log-datasize scale. Based on this result, oracle-support from a labelled dataset of limited size might not even present a top-line for the same number of support videos and as we have shown in Table 1 can be further boosted with inference-time increase in number of supports. Thus, this shows that using larger unlabelled video datasets and conducting retrieval presents an effective method for scalable video instance segmentation. In Figure 3 we provide qualitative examples of self-shot learning.

Semi-shot learning. In the final experiment, we show results for combining self-shot and oracle support videos. In this setting, we are given a few oracle

Table 5: Semi-shot learning. We show that the performance increases when adding additional self-shot videos to the oracle-support videos, arriving at a semi-shot alternative. We find that such a hybrid setting can quickly close the gap to few-shot learning with ground truth support videos, further highlighting the potential of self-shot learning for video instance segmentation

		# Oracle-support					
		0	1	2	3	4	5
# Self-shot	0		53.2	53.9	55.0	55.7	56.6
	1	51.4	53.6	54.5	55.1	56.0	
	2	52.3	54.2	54.8	55.9		
	3	52.8	54.7	55.2			
	4	53.6	55.1				
	5	54.3					

support videos and use these to help find more support videos in a self-shot manner, arriving at a semi-shot alternative. In Table 5, we show using a single oracle video and 4 self-shot videos boosts performance by 1.9% and achieves 55.1% mAP, thus almost reaching the performance when using 5 oracle-support videos, all without any further annotation requirements.

7 Discussion

Limitations. While we have proposed and explored the task of self-shot video instance segmentation, we have done so in a sequential fashion: The support set generation is detached from the segmentation pipeline. While this allows for better analysis of the method, performance can likely be further improved by training in an end-to-end fashion. We are also limited by the unlabelled video dataset size because of our computational resources. As shown in Figure 2, larger dataset sizes will likely further highlight the benefit of utilizing self-shot learning.

Conclusions. This paper proposed the task of self-shot learning. We have analysed and proposed this in the most annotation intense setting, that of video instance segmentation where existence of oracle support can be considered unrealistic. For this we develop a novel transformer based instance segmentation baseline and outline how to obtain support videos automatically from an unlabelled pool through self-supervision. Experiments show that our approach achieves strong performance, can already outperform oracle support in some settings, is scalable, and can be combined in a semi-supervised setting.

Acknowledgments: This work is financially supported by Qualcomm Technologies Inc., the University of Amsterdam and the allowance Top consortia for Knowledge and Innovation (TKIs) from the Netherlands Ministry of Economic Affairs and Climate Policy.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arXiv (2016) 5
2. Alwassel, H., Mahajan, D., Korbar, B., Torresani, L., Ghanem, B., Tran, D.: Self-supervised learning by cross-modal audio-video clustering. In: NeurIPS (2020) 3
3. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. arXiv (2021) 6
4. Asano, Y.M., Patrick, M., Rupprecht, C., Vedaldi, A.: Labelling unlabelled videos from scratch with multi-modal self-supervision. In: NeurIPS (2020) 3
5. Asano, Y.M., Rupprecht, C., Vedaldi, A.: Self-labelling via simultaneous clustering and representation learning. In: ICLR (2020) 3
6. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. STAT (2016) 8
7. Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V.: Attention augmented convolutional networks. In: ICCV (2019) 7
8. Benaim, S., Ephrat, A., Lang, O., Mosseri, I., Freeman, W.T., Rubinstein, M., Irani, M., Dekel, T.: Speednet: Learning the speediness in videos. In: CVPR. pp. 9922–9931 (2020) 3
9. Bertasius, G., Torresani, L.: Classifying, segmenting, and tracking object instances in video with mask propagation. In: CVPR (2020) 1, 3
10. Brown, A., Xie, W., Kalogeiton, V., Zisserman, A.: Smooth-ap: Smoothing the path towards large-scale image retrieval. In: ECCV (2020) 6
11. Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C.: Few-shot video classification via temporal alignment. In: CVPR (2020) 1
12. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020) 3, 8
13. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: ECCV (2018) 3
14. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: NeurIPS (2020) 3
15. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. arXiv (2021) 3
16. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., et al.: Hybrid task cascade for instance segmentation. In: CVPR (2019) 3
17. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. ICML (2020) 3
18. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv (2020) 5
19. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR (2021) 3
20. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv (2021) 5
21. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV (2017) 8
22. Dave, A., Tokmakov, P., Ramanan, D.: Towards segmenting anything that moves. In: ICCV Workshops (2019) 3, 10, 11

23. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In: ICCV (2021) 3
24. Feichtenhofer, C., Fan, H., Xiong, B., Girshick, R., He, K.: A large-scale study on unsupervised spatiotemporal representation learning. In: CVPR. pp. 3299–3309 (2021) 3
25. Feng, Y., Ma, L., Liu, W., Luo, J.: Spatio-temporal video re-localization by warp lstm. In: CVPR (2019) 1, 2
26. Feng, Y., Ma, L., Liu, W., Zhang, T., Luo, J.: Video re-localization. In: ECCV (2018) 1, 2
27. Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., Cord, M.: Learning representations by predicting bags of visual words. In: CVPR. pp. 6928–6938 (2020) 3
28. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018) 3
29. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: A new approach to self-supervised learning. In: NeurIPS (2020) 3
30. Han, T., Xie, W., Zisserman, A.: Self-supervised co-training for video representation learning. In: NeurIPS (2020) 3
31. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020) 5, 6, 10
32. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017) 3
33. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 7
34. Jain, M., van Gemert, J.C., Mensink, T., Snoek, C.G.M.: Objects2action: Classifying and localizing actions without any video example. In: ICCV (2015) 3, 4
35. Jain, M., Ghodrati, A., Snoek, C.G.M.: ActionBytes: Learning from trimmed videos to localize actions. In: CVPR (2020) 3
36. Kalantidis, Y., Sariyildiz, M.B., Pion, N., Weinzaepfel, P., Larlus, D.: Hard negative mixing for contrastive learning. In: NeurIPS (2020) 3
37. Kim, D., Cho, D., Kweon, I.S.: Self-supervised video representation learning with space-time cubic puzzles. In: AAAI. pp. 8545–8552 (2019) 3
38. Kliper-Gross, O., Hassner, T., Wolf, L.: One shot similarity metric learning for action recognition. In: International Workshop on Similarity-Based Pattern Recognition (2011) 1, 2
39. Korbar, B., Tran, D., Torresani, L.: Cooperative learning of audio and video models from self-supervised synchronization. NeurIPS (2018) 3
40. Li, P., Wei, Y., Yang, Y.: Consistent structural relation learning for zero-shot segmentation. NeurIPS (2020) 4
41. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017) 9
42. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014) 9
43. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2017) 6, 9
44. Lu, X., Wang, W., Shen, J., Tai, Y.W., Crandall, D.J., Hoi, S.C.: Learning video object segmentation from unlabeled videos. In: CVPR (2020) 3, 10, 11
45. Mettes, P., Snoek, C.G.M.: Spatial-aware object embeddings for zero-shot localization and classification of actions. In: ICCV (2017) 3

46. Mettes, P., Thong, W., Snoek, C.G.M.: Object priors for classifying and localizing unseen actions. IJCV (2021) 3, 4
47. Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos. In: CVPR (2020) 5, 10
48. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3DV (2016) 9
49. Misra, I., van der Maaten, L.: Self-supervised learning of pretext-invariant representations. In: CVPR (2020) 3
50. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV. pp. 69–84. Springer (2016) 3
51. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Shazeer, N., Ku, A., Tran, D.: Image transformer. In: ICML (2018) 7
52. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.: Context encoders: Feature learning by inpainting. In: CVPR (2016) 3
53. Patrick, M., Asano, Y.M., Huang, B., Misra, I., Metze, F., Henriques, J., Vedaldi, A.: Space-time crop & attend: Improving cross-modal video representation learning. In: ICCV (2021) 3
54. Patrick, M., Asano, Y.M., Kuznetsova, P., Fong, R., Henriques, J.F., Zweig, G., Vedaldi, A.: On compositions of transformations in contrastive self-supervised learning. In: ICCV. pp. 9577–9587 (2021) 3
55. Patrick, M., Huang, P.Y., Asano, Y., Metze, F., Hauptmann, A., Henriques, J., Vedaldi, A.: Support-set bottlenecks for video-text representation learning. In: ICLR (2021) 3
56. Perrett, T., Masullo, A., Burghardt, T., Mirmehdi, M., Damen, D.: Temporal-relational crosstransformers for few-shot action recognition. In: CVPR (2021) 1
57. Qi, J., Gao, Y., Hu, Y., Wang, X., Liu, X., Bai, X., Belongie, S., Yuille, A., Torr, P.H., Bai, S.: Occluded video instance segmentation. arXiv (2021) 4
58. Qian, R., Meng, T., Gong, B., Yang, M.H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal contrastive video representation learning. In: CVPR. pp. 6964–6974 (2021) 3
59. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: CVPR (2019) 9
60. Varamesh, A., Diba, A., Tuytelaars, T., Van Gool, L.: Self-supervised ranking for representation learning. NeurIPS (2020) 2, 3, 5, 6, 10
61. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS (2017) 8
62. Wang, L., Hua, G., Sukthankar, R., Xue, J., Niu, Z., Zheng, N.: Video object discovery and co-segmentation with extremely weak supervision. TPAMI (2016) 2
63. Wang, W., Lu, X., Shen, J., Crandall, D.J., Shao, L.: Zero-shot video object segmentation via attentive graph neural networks. In: ICCV (2019) 3
64. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. In: CVPR (2021) 1, 3, 8
65. Wu, Z., Xiong, Y., Stella, X.Y., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR (2018) 3, 10
66. Xu, H., Sun, X., Tzeng, E., Das, A., Saenko, K., Darrell, T.: Revisiting few-shot activity detection with class similarity control. arXiv (2020) 2
67. Yang, H., He, X., Porikli, F.: One-shot action localization by learning sequence matching network. In: CVPR (2018) 2

68. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: CVPR (2019) 1, 3, 4
69. Yang, P., Hu, V.T., Mettes, P., Snoek, C.G.M.: Localizing the common action among a few videos. In: ECCV (2020) 1, 2, 10, 11
70. Yang, P., Mettes, P., Snoek, C.G.M.: Few-shot transformation of common actions into time and space. In: CVPR (2021) 1, 2
71. Zhang, D., Dai, X., Wang, Y.F.: Metal: Minimum effort temporal activity localization in untrimmed videos. In: CVPR (2020) 1, 2
72. Zhang, L., Chang, X., Liu, J., Luo, M., Wang, S., Ge, Z., Hauptmann, A.: Zstad: Zero-shot temporal activity detection. In: CVPR (2020) 3, 4, 10, 11
73. Zhu, L., Yang, Y.: Compound memory networks for few-shot video classification. In: ECCV (2018) 2