Efficient Video Action Detection with Token Dropout and Context Refinement

Lei Chen[1] Zhan Tong[2] Yibing Song[3] Gangshan Wu[1] Limin Wang[1,4*]
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]Ant Group [3]AI[3] Institute, Fudan University [4]Shanghai AI Lab
leichen1997@outlook.com zhantong.2023@gmail.com yibingsong.cv@gmail.com
gswu@nju.edu.cn lmwang@nju.edu.cn

Abstract

Streaming video clips with large-scale video tokens impede vision transformers (ViTs) for efficient recognition, especially in video action detection where sufficient spatiotemporal representations are required for precise actor identification. In this work, we propose an end-to-end framework for efficient video action detection (EVAD) based on vanilla ViTs. Our EVAD consists of two specialized designs for video action detection. First, we propose a spatiotemporal token dropout from a keyframe-centric perspective. In a video clip, we maintain all tokens from its keyframe, preserve tokens relevant to actor motions from other frames, and drop out the remaining tokens in this clip. Second, we refine scene context by leveraging remaining tokens for better recognizing actor identities. The region of interest (RoI) in our action detector is expanded into temporal domain. The captured spatiotemporal actor identity representations are refined via scene context in a decoder with the attention mechanism. These two designs make our EVAD efficient while maintaining accuracy, which is validated on three benchmark datasets (i.e., AVA, UCF101-24, JHMDB). Compared to the vanilla ViT backbone, our EVAD reduces the overall GFLOPs by 43% and improves real-time inference speed by 40% with no performance degradation. Moreover, even at similar computational costs, our EVAD can improve the performance by 1.1 mAP with higher resolution inputs. Code is available at https://github.com/MCG-NJU/EVAD.

1. Introduction

Vision transformers (ViTs) have improved a series of recognition performance, including image classification [9, 46, 20] and object detection [5, 8, 40]. The image patches are regarded as tokens for ViT inputs for self-attention computations. When recognizing a video clip, we notice that the tokens are from each frame and thus formulate a large-scale input for ViTs. These video tokens introduce heavy computations during training and inference, especially in the self-attention computation step. While attempts have been made to reduce vision tokens [36, 28, 12, 47] for fast computations, it is still challenging for video action detection (VAD) [41, 16, 26] to balance accuracy and efficiency. This is because in VAD we need to localize actors in each frame and recognize their corresponding identities. For each actor, the temporal motion in video sequences shall be maintained for consistent identification. Meanwhile, the scene context ought to be kept to differentiate from other actors. Sufficient video tokens representing both actor motions and scene context will preserve VAD accuracy, which leaves a

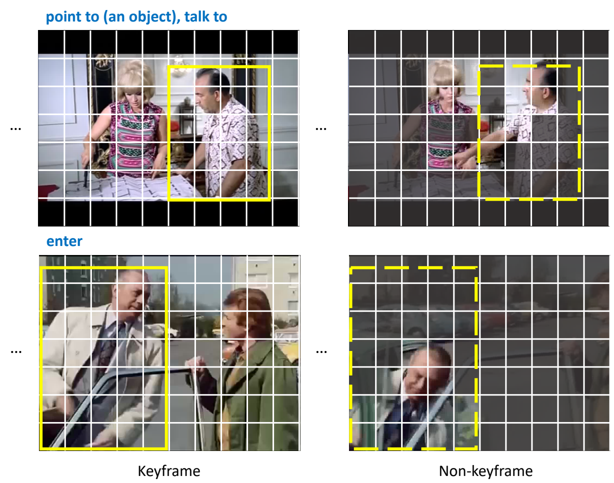

Figure 1: Spatiotemporal token dropout from a keyframe-centric perspective. We maintain tokens from the keyframe of a video clip, preserve a small amount of tokens from non-keyframes based on actor motions, and drop out the remaining tokens in this clip. On the first row, we preserve tokens relevant to the waving hand in non-keyframes as they benefit recognizing the action ‘point to’. On the second row, we drop out tokens irrelevant to the action ‘entering’ in non-keyframes for efficient recognition.

*Corresponding author.

research direction on how to select them for efficient VAD.

In this paper, we preserve video tokens representing actor motions and scene context, while dropping out irrelevant tokens. Based on the temporal coherency of video clips, we propose a spatiotemporal token dropout from a keyframe-centric perspective. For each video clip, we can select one keyframe representing the scene context where all the tokens shall be maintained. Meanwhile, we select tokens from other frames representing actor motions. Moreover, we drop out the remaining video tokens in this clip. Fig. 1 shows two examples. On the first row, we maintain all the tokens in the keyframe, preserve the tokens in non-keyframe relating to the eyes and mouths associating to the action 'talk to', the waving hand of the right person associating to the action 'point to', and drop out the remaining video tokens. On the second row, we maintain all the tokens in the keyframe, preserve the tokens in non-keyframes relating to the movement of entering the car from outside which is associated with the action 'enter', and drop out the remaining video tokens. Our spatiotemporal token dropout maintains the relevant actor motions and scene context from the coherent video clips, while discarding the remaining irrelevant ones for efficient computations.

We develop spatiotemporal token dropout via a keyframe-centric token pruning module within the ViT encoder backbone. The keyframe is either uniformly sampled or manually predefined in the video clips. We select the middle frame of the input clip with box annotations as the keyframe by default. The feature map of the selected keyframe is retained completely for actor localization. Besides, we extract an attention map enhanced by this keyframe. This attention map guides token dropout in the non-keyframes. After localization, we need to classify each localized bounding box (bbx) for actor identification. As video tokens are incomplete in non-keyframes, the classification performance is inferior in the bbx regions where tokens have been dropped out. Nevertheless, the inherent temporal consistency in video benefits us to refine both actor and scene context from the remaining video tokens. We expand the localized bbxs in the temporal domain for RoIAlign [17] to capture token features related to actor motions. Then, we introduce a decoder to refine actor features guided by the remaining video tokens in this clip. The decoder concatenates the actor and token features, and performs self-attention to produce enriched actor features for better identification. We find that after token dropout, the degraded action classification can be effectively recovered by using remaining video tokens for context refinement. The recovered performance is the same as that using the whole video tokens for action classification. Through this context refinement, we can maintain the VAD performance using reduced video tokens for efficient computations.

We conduct extensive experiments on three popular action detection datasets (AVA [16], UCF101-24 [41], JHMDB [19]) to show the advantages of our proposed EVAD. In the ViT-B encoder backbone, for instance, we employ a keyframe-centric token pruning module three times with a keep rate ρ of 0.7 (i.e,. the dropout rate is 0.3). The encoder outputs 34% of the original video tokens, which reduces GFLOPs by 43% and increases throughput by 40% while achieving on-par performance. On the other hand, under the similar computation cost (i.e., a similar amount of tokens), we can take video clips in a higher resolution for performance improvement. For example, we can improve detection performance by 1.1 mAP when increasing the resolution from 256 to 288 on the short side with GFLOPs reduced by 22% and throughput increased by 10%. We also provide visualizations and ablation studies to show how our EVAD performs spatiotemporal token dropout to eliminate action irrelevant video tokens.

2. Related Works

2.1. Spatio-temporal Action Detection

Current state-of-the-art methods [14, 13, 51, 44, 33, 6] commonly adopt a two-stage pipeline with two separated backbones, i.e., a 2D backbone for actor localization on keyframes and a 3D backbone for video feature extraction. Some previous approaches [23, 42] simplified the pipeline by training two backbones in an end-to-end manner, which suffer from heavy complexity and optimization difficulty. Most recent methods [15, 7, 58, 54] utilize a unified backbone to perform action detection. VAT [15] is a transformer-style action detector to aggregate the spatiotemporal context around the target actors. WOO [7] and TubeR [58] are query-based action detectors following the detection frameworks of [43, 5] to predict actor bounding boxes and action classes. STMixer [54] is a one-stage query-based detector to adaptively sample discriminative features. Several newly transformer-based methods [10, 52, 45] apply a ViT variant backbone and achieve competitive results following the two-stage pipeline. Moreover, there are also emerging methods [2, 25, 56] that focus on the training paradigm of deep networks for action detection task.

2.2. Spatio-temporal Redundancy

Spatial redundancy. The success of vision transformers has inspired various works [32, 18, 36, 39, 28, 38, 12] to explore the spatial redundancy of intermediate tokens. DynamicViT [36] observes that accurate image recognition is based on a subset of most informative tokens and designs a dynamic token sparsification framework to prune redundant tokens. EViT [28] calculates the attentiveness of the class token to each token and identifies the top-k tokens using the attentiveness value. ATS [12] introduces a differentiable adaptive token sampler for adaptively sampling significant

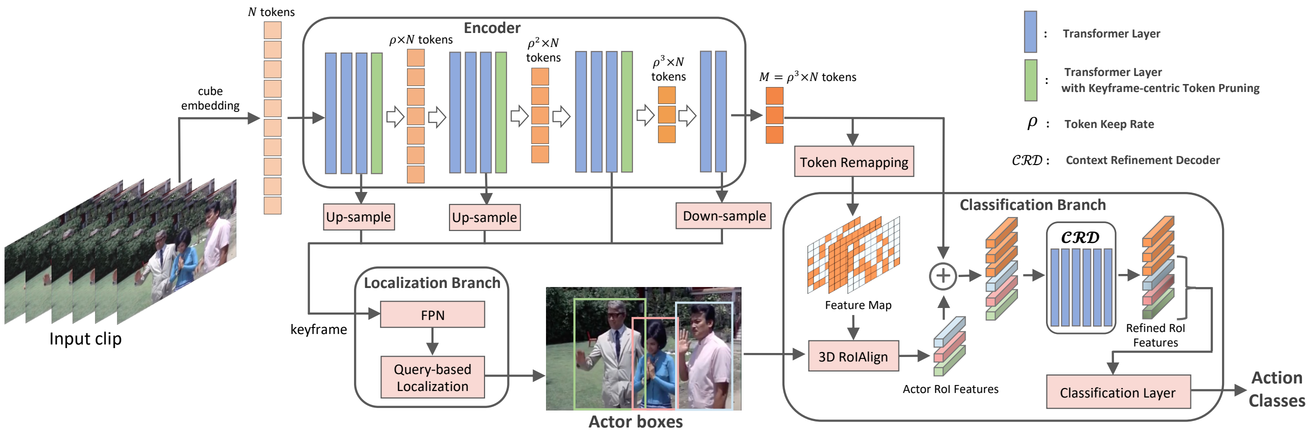

Figure 2: **Pipeline of EVAD.** Our detector consists of three parts: an encoder with keyframe-centric token pruning for efficient video feature extraction, a query-based localization branch using multiscale features of the keyframe for actor boxes prediction, and a classification branch conducting actor spatiotemporal feature refinement and relational modeling between actor RoI features and compact context tokens from ViT encoder.

tokens based on the image content.

Spatio-temporal redundancy. Due to the high redundancy of video datasets, extensive studies [29, 59, 55, 24, 49, 60, 11] have focused on developing efficient video recognition. Some recent approaches [3, 1, 4, 31, 35, 34, 47, 48] present efficient schemes specialized for video transformers. For the reduction of video tokens, MAR [35] devises a handcrafted masking strategy to discard a proportion of patches. K-centered [34] proposes a patch-based sampling that outperforms the traditional frame-based sampling. STTS [47] conducts token selection sequentially in temporal and spatial dimensions within the transformer. VideoMAE V2 [48] designs a dual masking strategy for efficient pre-training.

In contrast to these methods, we are the first to investigate redundancy in video action detection. We consider the correlation of keyframe and adjacent frames to drop out redundant tokens, and implement an efficient transformer-based action detector in an end-to-end manner.

3. Method

3.1. Overall EVAD Architecture

Video transformers overview. Video transformers [1, 45] perform tokenization by dividing the video sequence into $T/2 \times H/16 \times W/16$ cubes, each of size $2 \times 16 \times 16$. Then, project each cube to a 3D token using cude embedding. All tokens are added with positional encoding and fed into a sequentially-stacked transformer encoder consisting of a multi-head self-attention (MHSA) and a feed-forward network (FFN). The encoder output is used for classification after global average pooling. Alternatively, a learnable [CLS] token is added before the encoder for final prediction.

Our pipeline. The pipeline of end-to-end video action detection proposed in this paper is based on the vanilla ViT, as shown in Fig. 2. To alleviate the computational bottleneck caused by joint space-time attention, we devise an efficient video action detector (EVAD) with an encoder-decoder architecture with respect to the characteristics of action detection. EVAD enables the encoder with token pruning to remove the redundant tokens, and the decoder to refine actor spatiotemporal features. Following the setting in WOO [7], we utilize multiple intermediate spatial feature maps of the keyframe in the encoder for actor localization, and the spatiotemporal feature map output from the last encoder layer for action classification. **Token Pruning**. We design a keyframe-centric token pruning module to progressively reduce the redundancy of video data and ensure that few and promising tokens are delivered for action localization and classification. The specific token pruning module will be detailed in the next subsection. **Localization**. We up-sample or down-sample intermediate ViT feature maps of the keyframe to produce multiscale feature maps and send them to feature pyramid network (FPN) [30] for multiscale fusion. The localization branch predicts n candidate boxes in the keyframe by a query-based method, same as in [43, 7]. **Classification**. We remap the compact context tokens from ViT encoder into a spatiotemporal feature map with a regular structure. Then, we conduct 3D RoIAlign on the restored feature map using extended prediction boxes from the localization branch to obtain n actor RoI features. Subsequently, we utilize a context refinement decoder (CRD) for actor feature refinement and relational modeling between actor RoI features and compact context from the encoder, and the refined RoI features are passed through a classification layer for final prediction.

3.2. Keyframe-centric Token Pruning

The high spatiotemporal redundancy of video with similar semantic information between adjacent frames makes it

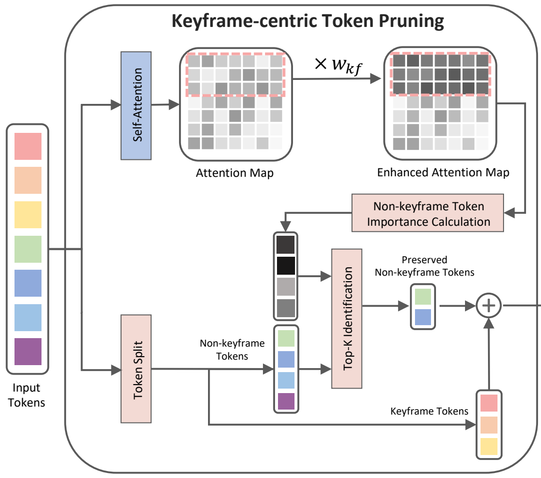

Figure 3: **Structure of keyframe-centric token pruning.** We conduct token pruning within the original encoder layer. Specifically, we calculate the importance scores of non-keyframe tokens using a keyframe-enhanced attention map. Then, we preserve the top-k important non-keyframe tokens concatenated with keyframe tokens as the results.

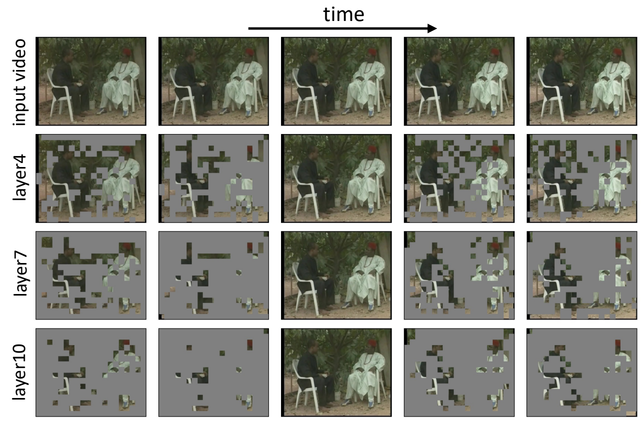

Figure 4: **Visualization of preserved tokens by encoder layers with token pruning.** On the third column, the keyframe tokens are all retained for accurate actor localization. The number of tokens in other frames progressively reduces, whereas important cues such as people and chairs can be effectively retained.

possible to perform token pruning with a high dropout rate on video transformers. In this paper, we design a keyframe-centric token pruning module, as shown in Fig. 3. We split all spatiotemporal tokens into keyframe tokens and non-keyframe tokens. And we preserve all keyframe tokens for accurate actor localization in the keyframe.

Non-keyframe token pruning. For the importance measure of non-keyframe tokens, we refer to the approach of EViT [28] in image classification, using the pre-computed attention map to represent the importance of each token without additional learnable parameters and nontrivial computational costs. As shown in the top part of Fig. 3, we first average the num_heads dimension of the attention map to obtain an $N \times N$ matrix, which represents the attentiveness between tokens (omitting the batch size). For example, $attn(i, j)$ denotes that the token i considers the importance of token j. Since there is no classification token specifically for video-level recognition, we calculate the average importance score of each token by $I_j = \frac{1}{N}\sum_{i=1}^{N} attn(i, j)$. Then we identify the top $(N * \rho - N_1)$ tokens from the N_2 non-keyframe tokens in descending order by importance scores, where N, N_1, N_2 represent the number of all, keyframe and non-keyframe tokens respectively, and ρ represents token keeping rate. Normally, the keyframe contains the most accurate semantic information for the current sample, and other frames away from the keyframe incur nontrivial information bias. Thus, it is practical to conduct token pruning guided by the keyframe. To this end, we insert a *Keyframe Attentiveness Enhancement* step between acquiring the attention map and calculating the importance of each non-keyframe token. As presented in Fig. 3, we apply a greater weight value to the keyframe queries, thereby retaining tokens with higher correlations to keyframe tokens. The importance score of each token is updated as follows:

$$I_j = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} w_{kf} \cdot \mathrm{attn}(i, j), & i \in (0, N_1) \\ \mathrm{atth}(i, j), & i \in (N_1, N) \end{cases} \tag{1}$$

where we assume the first N_1 tokens belong to the keyframe and the weight value w_{kf} is a hyper-parameter will be ablated in Sec. 4.2. In other words, we discard some tokens that only have a high response with non-keyframes, which may not be high-quality tokens. Only if the non-keyframe becomes the keyframe of previous/next samples, these highly responsive tokens are high-quality. By dropping out these redundant tokens with a high response to non-keyframes, we can further reduce the number of tokens. After the execution of token pruning, we send the preserved tokens to the follow-up FFN of the encoder layer.

The first token pruning is started at 1/3 of encoder layers to ensure that the model is capable of high-level semantic representation. Then, we perform token pruning every 1/4 of the total layers, discarding the redundant tokens and keeping the effective ones. As shown in Fig. 4, we visualize the preserved tokens of each pruning layer in ViT-B, and the model is able to retain important cues such as people and chairs. We present more visualizations in Appendix § D. After multiple pruning, the number of tokens is drastically reduced, which enables the model to save computation costs and accelerate training and inference process.

3.3. Video Action Detection

Actor localization branch. Benefit from preserving all keyframe tokens in Sec. 3.2, we can obtain multiple complete keyframe feature maps. We then up-sample or down-sample those keyframe feature maps for generating hierarchical features from the plain ViT. We then introduce a query-based actor localization head, inspired by Sparse R-CNN [43], to detect actors in the keyframe. The details of the localization head are provided in Appendix § B. Finally, the outputs of actor localization branch are n prediction boxes in the keyframe and corresponding actor confidence scores.

Action classification branch. Different from conventional feature extraction, EVAD produces M discrete video tokens. We need to restore the spatiotemporal structure of video feature map and then can perform location-related operations such as RoIAlign. We initialize a blank feature map shaped as $(T/2, H/16, W/16)$, fill the preserved tokens into this feature map according to their corresponding spatiotemporal positions, and pad the rest with zeros.

Next, we use the boxes generated by the localization branch to extract actor RoI features via 3D RoIAlign for subsequent action prediction. Due to actor movement or camera translation, actor spatial position is changed across frames, and using the keyframe box for 3D RoIAlign cannot obtain partial actor feature deviated from the box. Directly extending the scope of the box to cover whole motion trajectory might harm actor feature representation by introducing background or other interfering information. Nevertheless, in EVAD feature extraction stage, the interfering ones in the video are progressively eliminated, thus we can properly extend the box scope to add the deviated feature. The results in Sec. 4.2 demonstrate the ability of EVAD to cover large motion with extended boxes, while its counterpart ViTs impairs actor native feature representation.

We observe that directly applying token pruning methods of vision classification can greatly reduce the number of tokens involved in the calculations, but have a negative impact on final detection performance. Video action detection requires localizing and classifying the actions of all actors, but token pruning algorithms lead to incoherent actor features in space-time. As shown in Fig. 4, the static appearance of people sitting in the chairs is not retained completely in every frame. In the encoder, pair-wise self-attention is capable of modeling global dependency among tokens. The semantic information of the dropout tokens within the actor regions can be incorporated into some preserved tokens. Thus, we can recover the removed actor features from the preserved video tokens. To this end, we design a context refinement decoder to refine actor spatiotemporal representation. Concretely, we concatenate n actor RoI features with M video tokens and feed them into a sequentially-stacked transformer decoder consisting of MHSA and FFN. Guiding by the preserved tokens, actor features can enrich themselves with actor representation and motion information from other frames. Also, without token pruning, the decoder can be used as a kind of relational modeling modules [42, 57, 15, 44, 33] to capture inter-actor and actor-context interaction information. The actor feature refining and relational modeling capabilities of our decoder will be ablated in Sec. 4.2.

The n refined actor features output by the decoder are retrieved and passed through a classification layer to make final action prediction.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our EVAD on three common datasets for video action detection: AVA [16], UCF101-24 [41] and JHMDB [19]. AVA is a large-scale benchmark and contains 299 15-minute videos, divided into 211k training clips and 57k validation clips. The videos are annotated at 1FPS for boxes and labels. Following the standard evaluation protocol, we report our performance on 60 common action classes. UCF101-24 is a subset of UCF101. It includes 3,207 videos from 24 sports classes. Each video contains a single action class. Following the common practice, we report the performance on split-1. JHMDB contains 928 trimmed videos from 21 action classes. We report the average results over all three splits.

Evaluation criteria. We evaluate the performances with frame level mean Average Precision (mAP) under IoU threshold of 0.5 without multiple scales and flips for fair comparisons. We measure the throughput on a single A100 GPU with a batch size of 8 to estimate the average number of images that can be processed in one second. We specify a sample of video clip containing 16 frame images by default.

Our implementation details are described in Appendix § B.

4.2. Ablation Studies

We conduct in-depth ablation studies to investigate the effectiveness of our design in EVAD. All results are reported on AVA v2.2 with a 16-frame VideoMAE ViT-B backbone pre-trained and fine-tuned on Kinetics-400 [22].

Inference w/o re-training. EVAD does not add any additional learnable parameters for token pruning. Thus, we directly add our keyframe-centric token pruning module on the off-the-shelf pre-trained ViTs and show the performance of the models with different token keep rates in Fig. 5. We observe that directly applying token pruning negatively affects the model performance, and we speculate that the model needs re-training to adapt to a dynamic number of tokens at different layers. Then, we incorporate our token pruning in the training process, and retrain EVAD with token keep rate of 0.7 and infer on multiple keep rates. As

cls. head	ρ	mAP	#param
linear (baseline)	1.0	26.4	169M
WOO [7]	1.0	29.1	314M
our decoder	1.0	30.5	185M
linear	0.7	22.5	169M
our decoder	0.7	30.5	185M

(a) **Classification branch**

case	mAP	GFLOPs
ρ=1.0	30.5	223.8
random	29.9	158.9
CLS token	30.5	134.2
GAP	30.5	134.2

(b) **Pruning strategy**

case	mAP
keyframe	30.5
unified pruning	30.0

(c) **Keyframe**

case	mAP	GFLOPs
preserved tokens	30.5	134.2
all tokens	30.2	156.1

(d) **Decoder input**

depth	1	2	4	6	8
mAP	27.2	28.0	29.7	30.5	30.3

(e) **Decoder depth**

extend (x,y)	(0, 0)	(0.2, 0.1)	(0.4, 0.2)	(0.6, 0.3)
ρ=1.0	30.5	30.5	30.2	30.0
ρ=0.7	30.5	30.5	30.7	30.4

(f) **RoI extension**

w_{kf}	1	2	4	6	8
mAP	31.4	31.7	31.8	31.7	31.4

(g) **Attentiveness enhancement**

Table 1: **Ablation experiments on AVA v2.2.** All models here adopt 16-frame vanilla ViT-B as backbone and keep rate ρ is set to 0.7 as the default setting unless otherwise specified.

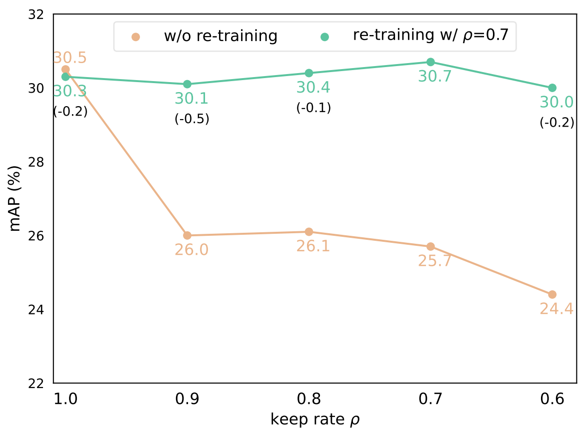

Figure 5: **Comparison on the performance of EVAD without re-training and re-training with keep rate ρ=0.7.** The numbers in '()' indicate the gap of the corresponding values to re-training with the target keep rate.

shown in Fig. 5, the result on each keep rate is nearly comparable to the re-training performance with the target keep rate. This illustrates that our token pruning module allows to adapt to different levels of computation costs after re-training once.

Classification branch. To illustrate the necessity of refining actor spatiotemporal features when using token dropout for efficient action detection, we conduct two groups of experiments, i.e., the encoder with keyframe-centric token pruning (ρ=0.7) and without token pruning (keep rate ρ=1.0), and examine the effect of context refinement decoder for actor feature refinement and relational modeling. Without token pruning, the decoder is used to conduct actor-context modeling to capture action interaction information. It enables +4.1 mAP gains over the baseline and outperforms the well-designed WOO head with fewer parameters. When token pruning is applied, partial actor features are eliminated during the pruning process, and the linear model is much worse than the baseline. Nevertheless, the strong feature refining capability of decoder compensates this gap and achieves a comparable performance with ρ=1.0.

Localization branch. To explore the localization capability of EVAD and its impact on the overall performance, we directly use the boxes from an off-the-shelf detector [37], which is commonly adopted in previous two-stage pipeline [14, 51, 44]. Our results shows that though utilizing the off-the-shelf detector can lead to an increase of +1.7 mAP, this comes at the cost of additional computational complexity, such as 246 GFLOPs for Faster R-CNN, and results in a more complex pipeline. The localization branch in our EVAD is lightweight with the cost of only 13.5 GFLOPs, and enjoys a simple end-to-end manner.

Pruning strategy. We then ablate different token pruning strategies with ρ=0.7. (1) Random masking performs a random sampling of non-keyframe tokens before entering the encoder. (2) Token pruning based on CLS token, as in [28], uses a learnable CLS token pre-trained on video recognition [22] responsible for calculating the importance of non-keyframe tokens. (3) GAP denotes pruning guided by global average pooling of all token attentive values. As shown in Table 1b, both attention-based approaches outperform random masking and achieve comparable performance with ρ=1.0, demonstrating that those models have the ability to preserve complete action semantics. We choose the latter with higher flexibility as the default setting.

Keyframe. In Sec 3.2, we emphasize the necessity of preserving all keyframe tokens. In Table 1c, we compare the two strategies of keeping all keyframe tokens and unified pruning by treating keyframe tokens as normal tokens, the latter discarding some important tokens in the keyframe and leading to performance degradation.

Decoder input. To verify if some discarded tokens are important for action detection, we save those discarded ones

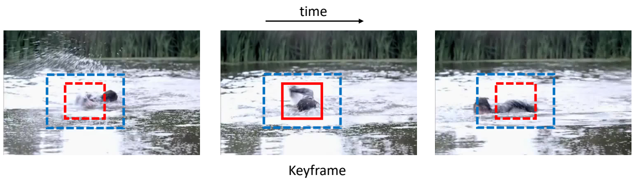

Figure 6: **RoI extension illustration.** The red boxes denote the prediction box in the keyframe and are copied to adjacent frames. The blue boxes denote the extended boxes to capture large motion.

when performing token pruning and feed all tokens into the decoder. As shown in Table 1d, using all tokens does not improve the performance, indicating that the tokens preserved by the model contain sufficient action semantics.

Decoder depth. The results in Table 1e show that stacking 6 layers in the decoder has the highest mAP, and more than 6 layers cause performance degradation due to over-fitting, so we use depth=6 as the default setting.

RoI extension. Due to human large motion, the box copied from the keyframe cannot cover the whole motion trajectory. Intuitively, we can solve it by properly extending the box scope, as shown in Fig. 6, extending the scope of the red box in the keyframe to cover the swimming person. In Table 1f, the model without token pruning shows a decreasing trend in performance as the proportion of box extending increases. Instead, the token pruning method achieves the best performance when the extended ratio reaches (0.4,0.2), where the box is $1.68\times$ of the original box, in which our pruning mechanism eliminates the interference information introduced by the extension. It is worth mentioning that our method increases 15.7% AP on 'swim' category. We observe that large displacements frequently happen in 'swim', and our token pruning combined with RoI extension can alleviate the effect of human movements to some extent.

Attentiveness enhancement. In Table 2b, there is a significant decrease in performance when the ρ is reduced from 0.7 to 0.6. It indicates that the number of tokens retained at ρ=0.6 is insufficient to contain complete information. In Sec. 3.2, we analyze the portion of preserved tokens only has high responses to non-keyframes, which could be redundant to the current sample. Thus, we conduct experiments on ρ=0.6 to explore the feasibility of further reducing the number of tokens in Table 1g. We ablate the attention weight of keyframe queries, where w_{kf}=4 yields the best result. It shows that the keyframe plays a greater role in identifying the importance of non-keyframe tokens, leading to a lower redundancy of preserved tokens.

Discussion on the preserved tokens. To explore what the preserved tokens correspond to, we compute the ratio of the

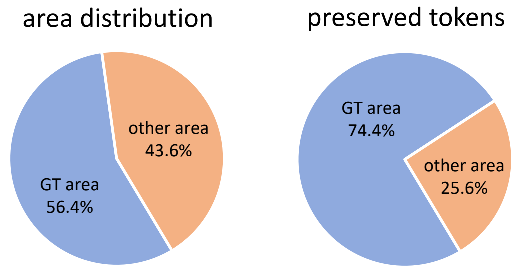

Figure 7: **Study on the relationship between the preserved tokens and GT bounding boxes.** 'GT area' denotes the union of all ground-truth bounding boxes in a single sample, and 'preserved tokens' in 'GT area' represents the proportion of preserved tokens inside the GT area. The results are averaged over AVA validation set.

preserved tokens inside the GT area on AVA validation set, as shown in Fig. 7. We observe that the average GT area is 56.4% of the input image, and 74.4% of the preserved tokens belong to the GT area (56.4%). This evidence indicates that, the proposed token pruning strategy allows our EVAD to focus on tokens in the GT area even without explicit box supervision.

4.3. Efficiency Analysis

To verify the efficiency of EVAD, we measure several facets of computation, inference speed and training cost at multiple resolutions.

First, we compare saved computation and improved inference speed with different keep rates, measured by GFLOPs and throughput, respectively, as shown in Table 2. When the keep rate ρ is in the range of [0.7, 1.0), the performance is comparable to ρ=1.0, which indicates that high spatiotemporal redundancy exists in action detection and our method can effectively remove redundancy and maintain critical cues. The model with ρ=0.7 achieves the best performance-efficiency trade-off for both resolutions. For the short side of 224, the detection mAP improves 0.2%, GFLOPs decrease by 40%, and throughput increases by 38%; for the short side of 288, mAP improves 0.2%, GFLOPs decrease by 43%, and throughput increases by 40%. Further reducing the keep rate degrades the performance because the number of tokens kept is insufficient to contain complete semantic information.

Then, we fix the keep rate ρ at 0.7 and compare the performance and efficiency gains of EVAD at higher resolutions, as shown in Table 3. We observe that: (1) EVAD with ρ=0.7 can obtain comparable performance to the model with ρ=1.0 at the same resolution with significantly lower computation and faster inference speed. (2) EVAD from a higher resolution provides a stable performance boost with relatively lower computation and faster speed. For instance, EVAD can improve performance by 1.1% when increasing

keep rate ρ	mAP	GFLOPs	throughput
1.0	30.5	223.8	298
0.9	30.6	186.3 (-17%)	328 (+10%)
0.8	30.5	157.0 (-30%)	377 (+27%)
0.7	30.7	134.2 (-40%)	411 (+38%)
0.6	30.2	116.3 (-48%)	427 (+43%)

(a) Comparison on the resolution of short side 224.

keep rate ρ	mAP	GFLOPs	throughput
1.0	32.1	424.5	171
0.9	32.2	346.3 (-18%)	179 (+5%)
0.8	32.1	287.4 (-32%)	216 (+26%)
0.7	32.3	242.8 (-43%)	240 (+40%)
0.6	31.4	208.9 (-51%)	263 (+54%)

(b) Comparison on the resolution of short side 288.

Table 2: **Comparison on the performance and efficiency variances of EVAD with different token keep rates.** The green numbers indicate the gap of the corresponding values to EVAD with ρ=1.0.

keep rate ρ	image size	mAP	GFLOPs	throughput
1.0	224	30.5	223.8	298
0.7	256	31.4	182.3	334
1.0	256	31.2	311.9	219
0.7	288	32.3	242.8	240
1.0	288	32.1	424.5	171

Table 3: **Comparison on the performance and efficiency improvements of EVAD with higher resolutions.**

model	backbone	mAP	GFLOPs	throughput
WOO [7]	SFR101	28.3	252	147
WOO [7]‡	ViT-B	30.0	378	176
TubeR [58]	CSN-152	33.6	240	64*
VideoMAE [45]	ViT-B	31.8	180+246	N/A
VideoMAE [45]	ViT-L	39.3	597+246	N/A
EVAD, ρ=0.7	ViT-B	32.3	243	240
EVAD, ρ=0.7	ViT-L	39.7	737	153

Table 4: **Comparison on the performance and efficiency of EVAD and other state-of-the-art methods.** ‡ denotes the results re-implemented by us for fair comparison. *: the code provided by TubeR does not include the implementation of its long-term context head, so the actual throughput will be less than we measured. 'N/A' indicates that the two-stage pipeline utilized in VideoMAE is not applicable to measure throughput.

the resolution from 256 to 288, while GFLOPs decrease by 22% and throughput increases by 10%.

Next, we compare the performance and efficiency of EVAD with other state-of-the-art methods in Table 4. We first observe that vanilla ViT allows WOO to achieve faster inference speed compared to traditional CNN-based backbone, and our EVAD can further accelerate the inference process (240 img/s vs. 176 img/s). Moreover, with a larger ViT-L backbone, EVAD can maintain a comparable inference speed to WOO with a CNN backbone (153 img/s vs. 147 img/s), and obtain significant performance improvements (39.7 mAP vs. 28.3 mAP). Although the previous art TubeR has fewer GFLOPs, it works slowly due to its heavy CSN-152 backbone. Compared to VideoMAE, EVAD can achieve comparable performance with fewer GFLOPs and allow for real-time inference in an end-to-end manner.

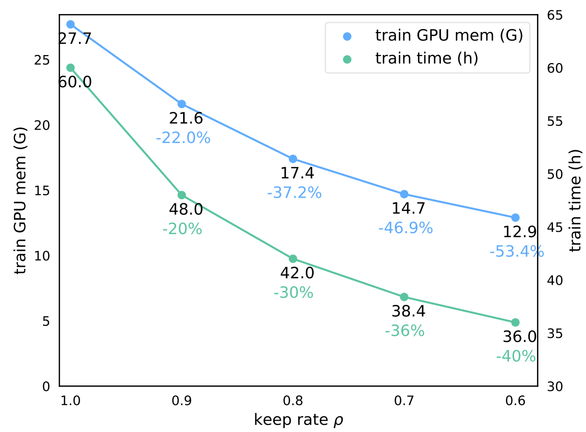

Figure 8: **Comparison on GPU memory and training time required of EVAD training with different token keep rates.** The blue and green numbers indicate the gap of the corresponding values to EVAD with ρ=1.0.

Finally, we show the significant impacts of EVAD during the training phase. In Fig. 8, as the keep rate decreases, GPU memory and training time consequently reduces during the EVAD training. When the keep rate ρ is 0.7, we can save 47% of GPU memory and 36% of training time, while maintaining the similar performance as EVAD with ρ=1.0.

In summary, EVAD can achieve significant improvements in saving computations, improving inference speed and reducing training costs. It can be treated as a simple yet efficient baseline applicable to many other action detection methods.

4.4. Comparison with the state-of-the-art methods

We compare our EVAD with the state-of-the-art methods on AVA v2.2 in Table 5a. The result on AVA v2.1 is provided in Appendix § C. To the best of our knowledge, the prevailing best model on AVA is VideoMAE, a two-stage model that requires an off-the-shelf person detector to pre-compute person proposals. Using the same pre-trained backbone, we can obtain comparable performance to VideoMAE, with mAP of 32.3 vs. 31.8 on ViT-B and 39.7 vs. 39.3 on ViT-L, and surpass other two-stage models. Compared to end-to-end models such as WOO and TubeR, we also have a significant performance gain, while benefiting

model	e2e	$T \times \tau$	backbone	pre-train	GFLOPs	mAP
SlowFast [14]	✗	32×2	SF-R101-NL	K600	393	29.0
ACAR-Net [33]	✗	32×2	SF-R101-NL	K600	N/A	31.4
AIA [44]	✗	32×2	SF-R101	K700	N/A	32.3
MeMViT [52]	✗	16×4	MViTv2-S	K400	305	29.3
MeMViT [52]	✗	32×3	MViTv2-B	K700	458	34.4
WOO [7]	✓	32×2	SF-R101-NL	K600	252	28.3
WOO [7]‡	✓	16×4	ViT-B	K400	378	30.0
TubeR [58]	✓	32×2	CSN-152	IG+K400	240	33.6
MaskFeat↑312 [50]	✗	40×3	MViTv2-L	K400	3074	37.5
VideoMAE [45]	✗	16×4	ViT-B	K400	426	31.8
VideoMAE [45]	✗	16×4	ViT-L	K700	843	39.3
STMixer [54]	✓	16×4	ViT-L	K700	N/A	39.5
EVAD, ρ=0.7	✓	16×4	ViT-B	K400	**243**	**32.3**
EVAD, ρ=0.7	✓	16×4	ViT-B†	K710+K400	**243**	**37.7**
EVAD, ρ=0.7	✓	16×4	ViT-L	K700	**737**	**39.7**

(a) **Comparison on AVA v2.2.**

model	e2e	$T \times \tau$	backbone	JHMDB	UCF24
ACT [21]	✓	6×1	VGG	65.7	69.5
MOC [27]*	✓	7×1	DLA32	70.8	78.0
AVA [16]*	✗	20×1	I3D-VGG	73.3	76.3
ACRN [42]	✓	20×1	S3D-G	77.9	-
CA-RCNN [53]	✗	32×2	R50-NL	79.2	-
YOWO [23]	✓	16×1	3D-X101	80.4	75.7
WOO [7]	✓	32×2	SF-R101-NL	80.5	-
TubeR [58]*	✓	32×2	I3D	80.7	81.3
TubeR [58]	✓	32×2	CSN-152	-	83.2
AIA [44]	✗	32×1	R50-C2D	-	78.8
ACAR-Net [33]	✗	32×1	SF-R50	-	84.3
STMixer [54]	✓	32×2	SF-R101-NL	86.7	83.7
baseline (Ours)	✓	16×4	ViT-B	88.1	83.9
EVAD, ρ=1.0	✓	16×4	ViT-B	88.5	84.9
EVAD, ρ=0.6	✓	16×4	ViT-B	**90.2**	**85.1**

(b) **Comparison on JHMDB and UCF101-24.**

Table 5: **Comparison with the state-of-the-art on three benchmarks.** ✓ denotes an end-to-end approach using a unified backbone, and ✗ denotes the use of two separated backbones, one of which is Faster R-CNN-R101-FPN (246 GFLOPs [37]) to pre-compute person proposals. ‡ denotes the results re-implemented by us for fair comparison. $T \times \tau$ refers to the frame number and corresponding sample rate. Methods marked with * leverage optical flow input. †: ViT-B backbone pre-trained on K710 and fine-tuned on K400 from VideoMAE V2 [48].

from the structural nature of transformers and token pruning with a low keep rate, we have a faster inference speed than CNN-based models and are more friendly to real-time action detection. Moreover, we can achieve slightly better performance than STMixer. STMixer is the most recent end-to-end model that designs a decoder to sample discriminative features. Orthogonal to it, our EVAD is devised for efficient video feature extraction, and further combining the two may yield better detection performance.

To demonstrate the generalizability of EVAD, we further verify our model on JHMDB and UCF101-24. As shown in Table 5b, we achieve state-of-the-art performance on both datasets with comparable improvement. Different from the experimental results on AVA, using the keep rate of 0.6 can lead to similar or better results than the model without token pruning on these two datasets. We consider the reason is that the scenarios of JHDMB and UCF01-24 are simpler and do not require complex relational modeling of actor-actor and actor-object, and hence can preserve a fewer number of tokens. In addition, the JHDMB dataset is small and using a lower keep rate for training might alleviate over-fitting and learn a better feature representation. We also provide the video-mAP results on both datasets in Appendix § C.

5. Conclusion and Future Work

Motivated by expensive computational costs of transformers with a video sequence and high spatiotemporal redundancy in video action detection, we design an Efficient Video Action Detector (EVAD) by dropping out spatiotemporal tokens and refining scene context to enable efficient transformer-based action detection. EVAD can achieve comparable performance to the vanilla ViT while considerably reducing computational costs and expediting inference speed, and it can reach the state-of-the-art on three popular action detection datasets. We hope that EVAD can serve as an efficient end-to-end baseline for future studies.

One limitation of our approach is that EVAD requires retraining once to take the benefits of reduced computations and faster inference from removing redundancy. A potential research approach is to explore transformer-adaptive token pruning algorithms. Moreover, we follow the end-to-end framework of WOO to verify the efficiency and effectiveness of EVAD, but WOO is still a 'two-stage' pipeline, which sequentially conducts actor localization and action classification modules. In future work, we aim to integrate those two modules into a unified head, which can reduce the inference time of passing through the detector head and hence amplify the efficiency benefits of EVAD.

Acknowledgements. This work is supported by National Key R&D Program of China (No. 2022ZD0160900), National Natural Science Foundation of China (No. 62076119, No. 61921006, No. 62072232)), Fundamental Research Funds for the Central Universities (No. 020214380091, No. 020214380099), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[2] Anurag Arnab, Xuehan Xiong, Alexey Gritsenko, Rob Romijnders, Josip Djolonga, Mostafa Dehghani, Chen Sun, Mario Lučić, and Cordelia Schmid. Beyond transfer learning: Co-finetuning for action localisation. *arXiv preprint arXiv:2207.03807*, 2022.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, 2021.

[4] Adrian Bulat, Juan Manuel Perez Rua, Swathikiran Sudhakaran, Brais Martinez, and Georgios Tzimiropoulos. Space-time mixing attention for video transformer. *Advances in Neural Information Processing Systems*, 2021.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020.

[6] Lei Chen, Zhan Tong, Yibing Song, Gangshan Wu, and Limin Wang. Cycleacr: Cycle modeling of actor-context relations for video action detection. *arXiv preprint arXiv:2303.16118*, 2023.

[7] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. Watch only once: An end-to-end video action detection framework. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[8] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[11] Mohsen Fayyaz, Emad Bahrami, Ali Diba, Mehdi Noroozi, Ehsan Adeli, Luc Van Gool, and Jurgen Gall. 3d cnns with adaptive temporal feature resolutions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[12] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European Conference on Computer Vision*, 2022.

[13] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[15] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[16] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE/CVF International Conference on Computer Vision*, 2017.

[18] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[19] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *IEEE/CVF International Conference on Computer Vision*, 2013.

[20] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *Advances in Neural Information Processing Systems*, 2021.

[21] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4405–4413, 2017.

[22] 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.

[23] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. *arXiv preprint arXiv:1911.06644*, 2019.

[24] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[25] Akash Kumar and Yogesh Singh Rawat. End-to-end semi-supervised learning for video action detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[26] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. Multisports: A multi-person video dataset of spatio-temporally localized sports actions. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[27] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *European Conference on Computer Vision*, 2020.

[28] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022.

[29] Jintao Lin, Haodong Duan, Kai Chen, Dahua Lin, and Limin Wang. Ocsampler: Compressing videos to one clip with single-step sampling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13884–13893, 2022.

[30] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[31] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[32] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers. *arXiv preprint arXiv:2110.03860*, 2021.

[33] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[34] Seong Hyeon Park, Jihoon Tack, Byeongho Heo, Jung-Woo Ha, and Jinwoo Shin. K-centered patch sampling for efficient video recognition. In *European Conference on Computer Vision*, 2022.

[35] Zhiwu Qing, Shiwei Zhang, Ziyuan Huang, Xiang Wang, Yuehuan Wang, Yiliang Lv, Changxin Gao, and Nong Sang. Mar: Masked autoencoders for efficient action recognition. *arXiv preprint arXiv:2207.11660*, 2022.

[36] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in Neural Information Processing Systems*, 2021.

[37] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[38] Cedric Renggli, André Susano Pinto, Neil Houlsby, Basil Mustafa, Joan Puigcerver, and Carlos Riquelme. Learning to merge tokens in vision transformers. *arXiv preprint arXiv:2202.12015*, 2022.

[39] Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. In *Advances in Neural Information Processing Systems*, 2021.

[40] Hwanjun Song, Deqing Sun, Sanghyuk Chun, Varun Jampani, Dongyoon Han, Byeongho Heo, Wonjae Kim, and Ming-Hsuan Yang. Vidt: An efficient and effective fully transformer-based object detector. *arXiv preprint arXiv:2110.03921*, 2021.

[41] 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.

[42] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *European Conference on Computer Vision*, 2018.

[43] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[44] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *European Conference on Computer Vision*, 2020.

[45] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022.

[46] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021.

[47] Junke Wang, Xitong Yang, Hengduo Li, Li Liu, Zuxuan Wu, and Yu-Gang Jiang. Efficient video transformers with spatial-temporal token selection. In *European Conference on Computer Vision*, 2022.

[48] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[49] Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang. Adaptive focus for efficient video recognition. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[50] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[51] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross B. Girshick. Long-term feature banks for detailed video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[52] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[53] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu. Context-aware rcnn: A baseline for action detection in videos. In *European Conference on Computer Vision*, 2020.

[54] Tao Wu, Mengqi Cao, Ziteng Gao, Gangshan Wu, and Limin Wang. Stmixer: A one-stage sparse action detector. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[55] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[56] Fanyi Xiao, Kaustav Kundu, Joseph Tighe, and Davide Modolo. Hierarchical self-supervised representation learning for movie understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[57] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[58] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Bing Shuai, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, et al. Tuber: Tubelet transformer for video action detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[59] Yin-Dong Zheng, Zhaoyang Liu, Tong Lu, and Limin Wang. Dynamic sampling networks for efficient action recognition in videos. *IEEE Transactions on Image Processing*, 29:7970–7983, 2020.

[60] Yuan Zhi, Zhan Tong, Limin Wang, and Gangshan Wu. Mgsampler: An explainable sampling strategy for video action recognition. In *IEEE/CVF International Conference on Computer Vision*, 2021.