Adapting Short-Term Transformers for Action Detection in Untrimmed Videos

Min Yang[1]   Huan Gao[2]   Ping Guo[3]   Limin Wang[1,4,✉]

[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]Inchitech   [3]Intel Labs China   [4]Shanghai AI Lab

yangminmcg1011@hotmail.com, gaohuan@inchitech.com, ping.guo@intel.com, lmwang@nju.edu.cn

https://github.com/MCG-NJU/ViT-TAD

Abstract

Vision Transformer (ViT) has shown high potential in video recognition, owing to its flexible design, adaptable self-attention mechanisms, and the efficacy of masked pre-training. Yet, it remains unclear how to adapt these pre-trained short-term ViTs for temporal action detection (TAD) in untrimmed videos. The existing works treat them as off-the-shelf feature extractors for each short-trimmed snippet without capturing the fine-grained relation among different snippets in a broader temporal context. To mitigate this issue, this paper focuses on designing a new mechanism for adapting these pre-trained ViT models as a unified long-form video transformer to fully unleash its modeling power in capturing inter-snippet relation, while still keeping low computation overhead and memory consumption for efficient TAD. To this end, we design effective cross-snippet propagation modules to gradually exchange short-term video information among different snippets from two levels. For inner-backbone information propagation, we introduce a cross-snippet propagation strategy to enable multi-snippet temporal feature interaction inside the backbone. For post-backbone information propagation, we propose temporal transformer layers for further clip-level modeling. With the plain ViT-B pre-trained with VideoMAE, our end-to-end temporal action detector (ViT-TAD) yields a very competitive performance to previous temporal action detectors, riching up to 69.5 average mAP on THUMOS14, 37.40 average mAP on ActivityNet-1.3 and 17.20 average mAP on FineAction.

1. Introduction

As an important task in video understanding, temporal action detection (TAD) [6, 13, 23] aims to localize all action instances and recognize their categories in a long untrimmed video. Most TAD methods [17, 18, 41, 44–46]
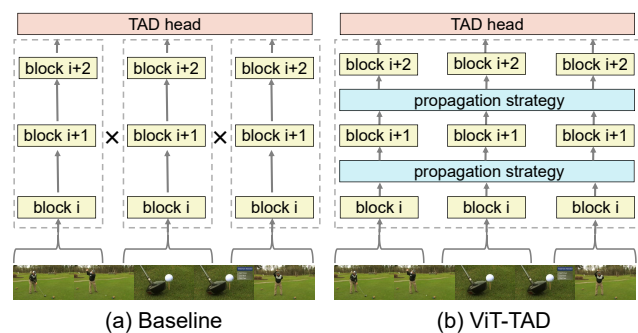
✉: Corresponding author.

Figure 1. **Different input processing between baseline and our ViT-TAD.** The dashed box illustrates the feature modeling within the backbone. In contrast to the baseline approach which models each snippet individually, our ViT-TAD allows snippets to collaboratively interact with each other during the modeling process within the backbone.

rely on the pre-trained action recognition networks (backbones) to extract short-term features for each snippet and then apply the TAD heads on top of feature sequence for action detection in long-form videos. In this pipeline, the feature extracted from the backbone is of great importance in the final TAD performance. To obtain powerful features, the existing TAD methods [41, 44–46] have tried different kinds of backbones, from CNN-based backbones [8, 39] to Transformer-based ones [12, 25]. Recently, the transformer [36] has become a promising alternative to CNN in modern TAD pipeline design, thanks to its effective self-attention operations. Furthermore, the flexible vision transformer (ViT) [11] enables self-supervised masked pre-training on extensive video datasets [34], resulting in a more robust video representation learner.

Several TAD methods [9, 10, 41, 46] have tried to apply Transformer-based backbones, such as VideoSwin [25] and MViT [12], to the task of TAD. Unlike CNN, these transformer backbones [11, 12, 25] present challenges when applied directly to untrimmed video modeling due to the

quadratic computational cost associated with self-attention operations [36]. Therefore, the existing methods often treat the transformer network as the **off-the-shelf** feature extractor for each snippet independently, as shown in Fig 1(a), thereby neglecting the intricate temporal relationships across snippets and failing to fully harness the end-to-end representation learning power of Transformers for TAD. Meanwhile, the hierarchical transformer architecture of VideoSwin or MViT they choose which is hard to benefit from the powerful masked video pre-training [34] on large-scale unlabeled videos. Therefore, it still remains unclear *how to effectively adapt the short-term* **plain ViT** *to untrimmed video action detection with the ability of capturing* **cross-snippet temporal structure** in an end-to-end manner.

To this end, we focus on building a simple and general temporal action detector based on the plain ViT backbone [11]. Rather than employing the pre-trained short-term ViTs as the snippet feature extractors, we design an efficient mechanism of adapting them to model longer video consisting of multiple snippets within a unified transformer backbone, as shown in Fig 1(b). Following the success of ViT in object detection [15], we first divide video into non-overlapping snippets and apply both intra-snippet and inter-snippet self-attention operations to keep a trade-off between representation power and computational cost. In this way, our unified transformer backbone can model multiple snippets as a whole and capture the fine-grained and holistic temporal structure for the TAD task.

Specifically, we apply dense self-attention on all tokens within each snippet to capture their spatiotemporal relations in intra-snippet blocks and propose a cross-snippet propagation module to aggregate global temporal information in a position-wise manner in inter-snippet blocks. These two kinds of blocks are stacked alternatively to gradually exchange temporal information within long videos. In addition, inspired by the design of DETR [7], we devise the post-backbone information propagation module after our Transformer-based backbone, which is composed of several temporal transformer layers to aggregate global temporal information. This post-backbone propagation module can effectively enlarge the temporal receptive field and capture global context. Equipped with simple TAD head such as BasicTAD [44] and AFSD [17], our final temporal action detector, termed as **ViT-TAD**, enjoys a simple yet effective design with end-to-end training. In particular, our ViT-TAD can embrace the powerful self-supervised masked pre-training [34] and yield state-of-the-art performance on the challenging datasets THUMOS14 and ActivityNet-1.3. In summary, our contributions are as follows:

- We introduce ViT-TAD, the first end-to-end TAD framework that utilizes the plain ViT backbone. Through the incorporation of a straightforward inner-backbone in-

formation propagation module, ViT-TAD can effectively treat multiple video snippets as a unified entity, facilitating the exchange of temporal global information.
- With a simple TAD head and careful implementation, we can train our ViT-TAD in an end-to-end manner under the limited GPU memory. This simple design fully unleashes the modeling power of the transformer and embraces the strong pre-training of VideoMAE [34].
- The extensive experiments on THUMOS14 [13], ActivityNet-1.3 [6] and FineAction [23] demonstrate that our simple ViT-TAD outperforms the previous state-of-the-art end-to-end TAD methods.

2. Related Work

Transformer in Action Recognition. Action recognition is an important task in video understanding. With the success of the self-attention mechanism in computer vision, several works tried to apply the transformer to their structures. Specifically, VTN [27] and STAM [30] introduced temporal transformers to encode frame-level relationships between features. ViViT [2] and TimeSformer [4] factorized along spatial and temporal dimensions on the granularity of the encoder. SMAVT [5] aggregated information from tokens at the same spatial location within a local temporal window. Some works [12, 25] tried to reintroduce hierarchical designs into transformer inspired by ConvNet. Among them, MViT [12] presented a hierarchical transformer structure to progressively shrink the spatiotemporal resolution of feature maps and increase channels as the network structure goes deeper. VideoSwin [25] used shifted window attention to enable information propagation inspired by Swin Transformer [24]. Furthermore, with the development of self-supervised learning [34], Transformer-based methods benefit from larger training data and achieve better results than CNN-based methods on the action recognition task.

Transformer in Temporal Action Detection. With the development of transformer, more and more TAD approaches began to apply it or its variants into TAD head [22, 32, 33, 45] or backbone [9, 10, 41, 46]. Different from increasingly successful applications of transformer in TAD head, few works seriously explored the application of transformer in the backbone. Existing works [9, 10, 46] treated Transformer-based backbones as "black box" and applied them as short-term feature extractors. STPT [41] attempted to explore the inner modeling of a hierarchical Transformer-based backbone by inserting new blocks in it. However, it needs additional pre-training on the action recognition dataset which fails to embrace the benefits of pre-trained big models [28, 40]. In this work, we try to design a new mechanism for adapting a pre-trained short-term snippet-level non-hierarchical ViT-based backbone as a unified clip-
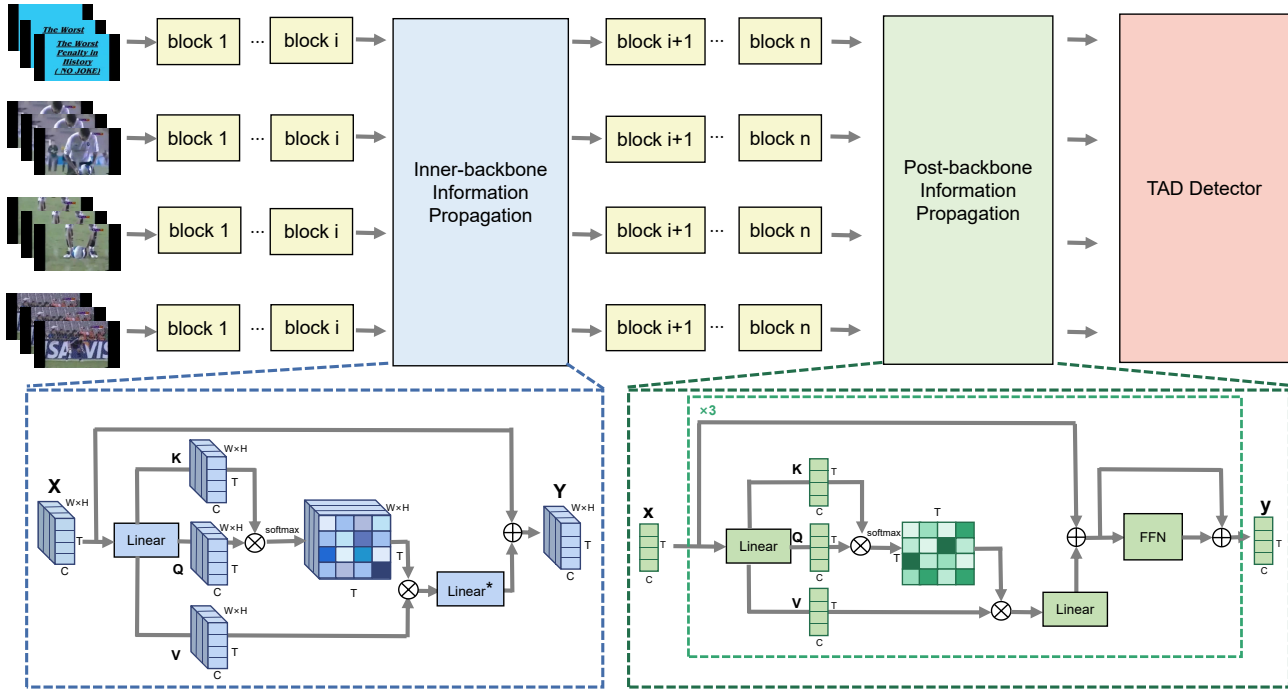
Figure 2. **Overview of ViT-TAD.** We suppose the ViT-based backbone has n blocks and divide them into several subsets. Each subset has i blocks. We divide a video clip into several snippets and send each into the backbone for feature extraction. We perform temporal feature interaction among all snippets through the inner-backbone information propagation strategy. We further conduct clip-level modeling to refine clip-level features through the post-backbone information propagation strategy. $*$ means the last layer is initialized as zero.

level video transformer.

End-to-End Temporal Action Detection. Due to GPU memory constraints, most existing methods treated TAD [18, 19, 43, 47] as a two-step approach separating feature modeling and action detection. Since temporal features are extracted using backbones that are pre-trained on the action recognition dataset, there is a gap between these two different tasks, leading to sub-optimal performance. In order to address this gap, incorporating the backbone into the network's training process forms an end-to-end TAD framework [10, 17, 20, 41, 46]. Simultaneously updating the network weights of both parts in the two-step approach will easily exceed the GPU memory limit. To address this issue, some methods [17, 20, 41] tried to downscale the frame resolution, while others tried to keep the resolution of the original video frame through improvements in training mechanisms. Among them, TALLFormer [10] attempted to build an offline memory bank to store features that cannot be updated synchronously and only update a portion of features during end-to-end training. Re^{2}TAL [46] built a backbone with reversible modules to save memory consumption. The concurrent work AdaTAD [21] achieved the purpose of saving video memory by only training the adapter and freezing the backbone. Due to the existence

of positional encoding, how to fine-tune the TAD model when spatiotemporal resolution differs from the pre-trained Transformer-based model is a problem. TALLFormer [10] and Re^{2}TAL [46] circumvented this problem by keeping the same spatiotemporal resolution as the pre-trained backbone, while STPT [41] pre-trained its own backbone. We adopt the downscaling frame resolution strategy and use the operation from DeiT [35] to fix the positional encoding across resolutions to better use the pre-trained weight of the ViT-based backbone.

3. ViT-TAD

Overview. We describe the details of our ViT-TAD pipeline shown in Fig 2. Formally, for each input video $V \in R^{T \times H \times W \times 3}$, where T, H and W represent the number, height and width of RGB frames respectively. We divide it into N_s non-overlapping snippets $s = \{s_j\}_{j=1}^{N_s}$, where $s_j \in R^{(T/N_s) \times H \times W \times 3}$. Then we feed each non-overlapping snippet s_j into consecutive blocks of the ViT-based backbone. In the baseline approach (shown in Fig 1(a)) which is adopted by the existing methods [10, 46], each snippet has no interaction with each other during fine-tuning, so we propose *inner-backbone information propagation* module to enable cross-snippet interaction between all snippets. Given
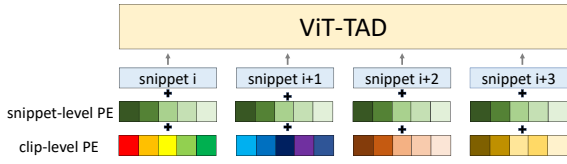
Figure 3. **Temporal positional encoding for all snippets.** Original input consists of a snippet and its snippet-level PE. We add a learnable additional temporal PE, called clip-level PE here. PE is short for positional encoding.

that the inner-snippet and cross-snippet modeling process within the backbone is gradual and sparse, resulting in relatively small receptive fields, it becomes necessary to enhance the final snippet-level features by effectively expanding their temporal receptive field. So we propose the *post-backbone information propagation* module composed of several temporal transformer layers to refine snippet-level temporal features with the global context. Finally, the enhanced features will be sent to the TAD head for action detection, producing K predicted actions $\{(s_i, e_i, c_i, p_i)\}_{i=1}^K$ as final detection results, where s_i, e_i, c_i, p_i represent the start timestamp, end timestamp, category and confidence score of ith action instance.

3.1. Inner-backbone Information Propagation

We design a cross-snippet propagation module for inner-backbone information exchange. Unlike hierarchical backbone structures like VideoSwin [25], we do not shift the temporal windows across layers. We use very few blocks that can go across snippets to allow information propagation. Inspired by ViT-Det [15], we evenly split a pre-trained ViT-based backbone into several subsets, and then insert a cross-snippet propagation module after the last block of each subset to enable information propagation. For the simplicity of the method, we adopt two simple but useful blocks as our cross-snippet propagation modules, coined as *Local Propagation Block* and *Global Propagation Block*.

Local Propagation Block. We choose the bottleneck architecture, comprising three 3D convolutions and an identity shortcut, as our local block. The last normalization layer within this block is initialized to zero, establishing an initial identity state for the block. This design permits integration into any place without breaking the pre-trained model. Although the cross-snippet modeling capacity of convolutions is limited, it can still exchange cross-snippet information gradually via the stacking of multiple such modules.

Global Propagation Block. Self-attention mechanism is naturally suitable for long-term modeling, so we choose it to build a global block shown in Fig 2. To keep low computation overhead and memory consumption, we perform temporal self-attention among all video snippets after the last block of each stage. It is noteworthy that spatial dimensions do not participate in self-attention where width(W) and height(H) are directly flattened to batch dimension. Specifically, we obtain the sequence of clip-level spatiotemporal features $X \in R^{(W \times H) \times T \times C}$ and split them into $W \times H$ temporal features $x \in R^{T \times C}$, where $X = \text{concat}(x_1, x_2, ..., x_{W \times H})$ and each x represents the clip-level temporal feature of a certain spatial location. For x in each spatial location, we formulate our self-attention mechanism as:

$$q = xw_q, k = xw_k, v = xw_v,$$
$$y = \text{softmax}(qk^T/\sqrt{C_q})v, \quad (1)$$

where $w_q \in R^{C \times C_q}$, $w_k \in R^{C \times C_k}$, $w_v \in R^{C \times C_v}$, $C_q = C_k = C/m$ and m is the number of attention head. Our ViT-TAD complexity is reduced from $O(H^2W^2L_s^2N_sC)$ to $O(HWT^2C)$ where $L_s = T/N_s$. Finally, we obtain $Y = \text{concat}(y_1, y_2, ..., y_{W \times H}) \in R^{(W \times H) \times T \times C}$. After the application of multi-head dot-product attention, a linear layer is required to attend to information from distinct representation sub-spaces and positions jointly. Similar to the local propagation block, the last linear layer in the global block is initialized as zero and an identity shortcut is employed. Since each spliced short-term snippet in the clip shares the same temporal positional encoding, concatenating snippet-level positional encoding cannot capture the long-term temporal order of the time series. Therefore, we propose additional clip-level positional encoding to depict the temporal sequence's overall order within the video. As shown in Fig 3, we add a learnable clip-level positional encoding to each temporal position of the snippet.

3.2. Post-backbone Information Propagation

Given that modeling snippet-level features in the backbone with an inner-backbone information propagation strategy is a sparse and gradual process. we have designed a post-backbone information propagation module to ensure adequate feature interaction among snippets and incorporate clip-level modeling. Inspired by the encoder designed by DETR [7], we build L temporal transformer encoder layers to allow snippets to interact with each other, as shown in Fig 2. It is worth noting that the spatial dimension of each snippet feature has been squeezed with average pooling before this propagation module. Formally, given the video features F concatenated by several snippets of feature provided by backbone, several temporal transformer encoder layers are adopted to enhance these features: $F^i = \text{encoder}(F^{i-1})$, where $i \in [1, L]$ and L is the number of encoder layers, $F^0 = F$ and $L = 3$ in our experiment.

3.3. TAD Head

Our proposed ViT-TAD framework is a simple and general temporal action detection pipeline. In principle, it

is compatible with any TAD head for converting features into predictions. Following the previous practice in TAD, we choose a one-stage anchor-free method from Basic-TAD [44] on THUMOS14 and FineAction, a two-stage anchor-free method from AFSD [17] on ActivityNet-1.3.

Discussion with ViT-Det. In spirit, our work is similar to ViT-Det [15] which also has similar findings in the field of object detection where restricted window attention is not enough to capture object information in a global view. However, dynamic motion scenes are different from static object scenes, so our ViT-TAD behaves differently in several ways. *First*, temporal continuity makes action content highly correlated with long-range context, making the global block perform better than the local block which is different from ViT-Det. *Second*, we design post-backbone propagation to enhance further temporal context modeling based on the first finding, which is missing in ViT-Det. *Finally*, unlike spatial downsampling in ViT-Det, our ViT-TAD exhibits the same temporal downsampling rate and pattern as the previous CNN-based TAD method (e.g., BasicTAD [44]). This is the reason why we directly use the same FPN with BasicTAD without specific exploration, but ViT-Det needs further study. *In addition*, the TAD community often tend to avoid designing end-to-end detection pipeline, but instead devise complicated temporal modules on outdated pre-extracted features (e.g., I3D of 6 years ago). Building a neat end-to-end TAD pipeline and embracing more powerful models is more urgent and we make a meaningful attempt to facilitate TAD research in this direction.

4. Experiments

Datasets and Evaluation Metric. We perform extensive experiments on THUMOS14 [13], ActivityNet-1.3 [6] and FineAction [23] to demonstrate the effectiveness of ViT-TAD. **THUMOS14** is a commonly-used dataset in TAD, containing 200 validation videos and 213 test videos with labeled temporal annotations from 20 categories. **ActivityNet-1.3** is a large-scale dataset containing 10,024 training videos and 4,926 validation videos belonging to 200 activities. **FineAction** is a newly collected large-scale fine-grained TAD dataset containing 57,752 training instances from 8,440 videos and 24,236 validation instances from 4,174 videos and 21,336 testing instances from 4,118 videos. Following previous work, we report the **mean average precision (mAP)** with tIoU thresholds [0.3:0.1:0.7] for THUMOS14 and [0.5:0.05:0.95] for ActivityNet-1.3 and FineAction. Avg is average mAP on these thresholds.

Implementation Details. For THUMOS14 and FineAction, we sample each video clip with temporal windows of 32 seconds covering 99.7% of action instances for THUMOS14 and 48 seconds covering 97.0% of action instances for FineAction. We sample each video clip with 256 frames for THUMOS14 and 384 frames for FineAction at 8 FPS

prop. strat.	blk. num	0.3	0.4	0.5	0.6	0.7	Avg
none	-	74.6	70.2	62.6	51.3	38.4	59.4
global	4	78.4	74.5	66.6	53.4	38.7	**62.3**
	12	76.5	71.6	63.3	51.9	38.4	60.3
local	4	75.9	71.6	64.7	53.3	38.3	60.8
	12	77.1	72.8	65.4	54.2	40.6	**62.0**

Table 1. **Study on inner-backbone information propagation**. Both propagation strategies and different numbers of blocks are explored. "none" is our baseline without propagation.

and then divide each clip into 16 and 24 snippets each, while each snippet consists of 16 frames. Due to the limited GPU memory, we resize frame's original size into short-128 (the short side of the frame is set to 128) and set the crop size (the size of the cropped images) to 112×112 for THUMOS14 in ablation experiments and report final results in short-180 original size and 160×160 cropping size. Inspired by DeiT [35], we need to adapt the positional embeddings to smaller spatial resolution with bicubic interpolation. For TAD head, we choose a one-stage anchor-free method from BasicTAD [44] for its astonishing detection performance. We train the model using SGD with a momentum of 0.9 and weight decay of 0.0001 on 8 TITAN Xp GPUs. The batch size is set to 2 for each GPU. For ActivityNet-1.3, we resize all videos to 768 frames and then treat these frames as a video clip. Similarly, we divide each clip into 48 snippets and each snippet consists of 16 frames. For TAD head, we choose a two-stage anchor-free method from AFSD [17] for its more accurate boundary regression. We resize the frame's original size into short-180 and set the crop size to 160×160. We adopt positional embeddings to smaller spatial resolution with bicubic interpolation. We train the model using AdamW [26] with a learning rate of 0.0002 and weight decay of 0.0001 on 8 TITAN Xp GPUs. The batch size is set to 1 for each GPU.

4.1. Ablation Study

In this section, we perform ablation experiments for ViT-TAD on THUMOS14. Both ViT-S and ViT-B we adopted are pre-trained on Kinetics-400 [14] provided by [38].

Study on Inner-backbone Information Propagation. Table 1 ablates our inner-backbone information propagation strategy. These propagation blocks are placed in the backbone evenly and kernel size is set to $(3 \times 3 \times 3)$ for the local propagation block. We compare our global and local propagation blocks with the baseline, and both of them perform better than the baseline in 4-block settings. We further explore the number of propagation blocks and find that the local strategy benefits from having more blocks, but the global strategy does the opposite. There are two plausible explanations for this. First, the model's ability to learn valuable information is hampered by frequent global interactions. Sec-

temp. pos. enc.	0.3	0.5	0.7	Avg
✓	78.4	66.6	38.7	**62.3**
	77.9	64.4	38.8	61.3

(a) Effect of clip-level temporal positional encoding

prop. strategy	0.3	0.5	0.7	Avg	Mem.
none	74.6	62.6	38.4	59.4	8.3GB
1D	78.4	66.6	38.7	**62.3**	9.3GB
3D	78.0	65.5	39.7	62.2	27GB

(b) Comparison between 1D strategy and 3D strategy. Both strategies are shown in Fig 4.

propagation location	0.3	0.5	0.7	Avg
evenly 4 blocks	78.4	66.6	38.7	**62.3**
first 4 blocks	75.2	64.1	38.5	60.1
last 4 blocks	77.6	66.0	39.6	61.8

(c) Study on the locations of global blocks. Here we choose 4 global blocks strategy.

Table 2. **Study on global propagation strategy**. "none" is our baseline without propagation.

prop. strategy	use p-b	0.3	0.4	0.5	0.6	0.7	Avg
none		74.6	70.2	62.6	51.3	38.4	59.4
none	✓	77.3	73.5	65.1	52.6	37.7	**61.2**
4 global blocks		78.4	74.5	66.6	53.4	38.7	62.3
4 global blocks	✓	78.7	74.1	66.2	55.5	40.3	**63.0**
12 local blocks		77.1	72.8	65.4	54.2	40.6	**62.0**
12 local blocks	✓	77.5	73.3	65.9	52.7	38.2	61.5

Table 3. **Study on Post-backbone Information Propagation**. p-b is short for post-backbone information propagation.

ond, the self-attention operation cannot use prior information like the convolutional operator. To enhance temporal interaction, a more in-depth exploration is warranted.

Study on Global Propagation Strategy. The design of the global block is more flexible than the local block and requires further discussion. As shown in Table 2a, additional clip-level positional encoding can enable the model to recognize clip-level feature sequences after snippet splicing, leading to better detection results. To save computational consumption, we disassemble the complete cross-snippet spatiotemporal relationship modeling into cross-snippet temporal modeling and inner-snippet spatiotemporal modeling. For vivid comparison, our global propagation block is referred to as 1D (temporal only) strategy shown in Fig 4(b), and applying clip-level spatiotemporal modeling on the last block of each subset in the backbone is referred to as 3D (temporal+spatial) strategy (Fig 4(a)). Both 1D and 3D strategies obtain similar detection results while 3D strategy causes more memory consumption shown in Table 2b, inferring that concentrating on clip-level temporal modeling is sufficient. We further study where global blocks should be located in the backbone. By default 4 global blocks are placed evenly. We compare by placing them in the first or last 4 blocks instead. As is shown in Table 2c, evenly 4 blocks configuration performs best, and last 4 blocks configuration follows. This is in line with the observation in ViT [11] that ViT has a longer attention distance in later blocks and is more localized in earlier ones. Premature temporal interaction of individual snippet features yields limited benefit. Finally, we adopt 4 evenly placed global blocks as the final configuration of the global propagation block.

Study on Post-backbone Information Propagation. Post-backbone information propagation strategy is used for sufficient feature interaction among snippets. It shows that only applying such a strategy ("none" in Table 3) is also effective. Equipped with post-backbone information propa-
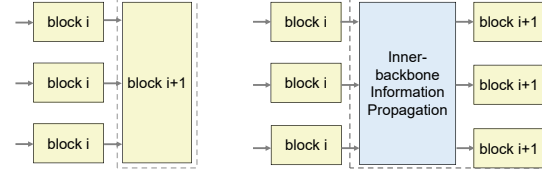
(a) 3D Strategy　　　(b) 1D Strategy

Figure 4. **Comparison between 1D and 3D Propagation Strategy.** (a) The 3D setting: the $(i+1)$th block takes all snippets as input and directly applies spatiotemporal self-attention to the whole video clip. (b)The 1D setting: global propagation block is inserted between consecutive backbone blocks.

Method	details	0.3	0.4	0.5	0.6	0.7	Avg
ActionFormer [45]	ViT-S$_{224\times224}^{30FPS}$	75.2	70.0	62.0	48.7	33.4	57.9
	ViT-S*	75.6	70.3	63.4	52.4	39.6	**60.3**
TriDet [31]	ViT-B$_{224\times224}^{30FPS}$	82.9	78.5	72.0	61.2	46.2	**68.2**
	ViT-B$_{160\times160}^{8FPS}$	77.0	72.2	63.8	51.6	35.1	60.0
BasicTAD(baseline) [44]	ViT-S$_{160\times160}^{8FPS}$	74.6	70.2	62.6	51.3	38.4	59.4
ViT-TAD	ViT-S$_{160\times160}^{8FPS}$	78.7	74.1	66.2	55.5	40.3	63.0
ViT-TAD	ViT-B$_{160\times160}^{8FPS}$	85.1	80.9	74.2	61.8	45.4	**69.5**

Table 4. **Comparison between ViT-TAD and other TAD pipelines.** ViT-S* means BasicTAD fine-tunes that backbone feature in an end-to-end manner. Details include the choice of backbone, frame resolution, and frame rate.

Method	Backb.	0.3	0.4	0.5	0.6	0.7	Avg
TALLFormer [10]	Swin-B	76.0	-	63.2	-	34.5	59.2
TALLFormer (w.o. proposed module)	ViT-B	78.9	75.0	67.6	56.1	37.9	63.1
TALLFormer (w. proposed module)	ViT-B	81.0	77.0	70.7	58.9	42.7	**66.1**

Table 5. **Effectiveness of proposed module in ViT-TAD to TALLFormer.** Proposed module means both inner-backbone and post-backbone propagation blocks.

gation, we then compare the detection results for both local and global blocks with the best configuration each in the inner-backbone strategy. As is shown in Table 3, we could see that the global propagation strategy obtains better detection results (from 62.3% to 63.0%), while the detection results of the local propagation strategy drop (from 62.0% to 61.5%), so we employ the global propagation strategy for both inner-backbone and post-backbone information propagation strategies.

Comparison between ViT-TAD and other TAD Pipelines. In this section, we compare ViT-TAD with four convincing methods of TAD, namely ActionFormer [45], BasicTAD [44], TALLFormer [10] and Tridet [31]. Here we migrate the feature modeling capabilities of ViT-TAD to

Methods	details	GPU	FPS	Avg
ActionFormer [45]	ViT-B$_{224×224}^{30FPS}$	3090	268	65.1
TriDet [31]	ViT-B$_{224×224}^{30FPS}$	3090	239	68.2
PBRNet [20]	I3D$_{96×96}^{10FPS}$	1080Ti	<1488	47.1
AFSD [17]	I3D$_{96×96}^{10FPS}$	1080Ti	<3259	52.0
e2e-tadtr [22]	res50-SlowFast$_{112×112}^{10FPS}$	TITAN Xp	5076	54.2
BasicTAD [44]	SlowOnly$_{112×112}^{3FPS}$	TITAN Xp	7143	54.5
ViT-TAD	ViT-S$_{112×112}^{8FPS}$	TITAN Xp	2135	63.0
ViT-TAD	ViT-S$_{160×160}^{8FPS}$	TITAN Xp	1349	64.3
ViT-TAD	ViT-B$_{160×160}^{8FPS}$	TITAN Xp	845	**69.5**

Table 6. **Runtime comparison with other TAD methods.** Details include the choice of backbone, frame resolution, and frame rate.

prop. strategy	details	Avg	params	train mem.	FLOPs	FPS
none	ViT-S$_{112×112}^{8FPS}$	59.4	25.62M	8.39G	104.59G	2204
4 global blocks+p-b	ViT-S$_{112×112}^{8FPS}$	**63.0**	33.35M	9.33G	108.80G	2135
	ViT-S$_{160×160}^{8FPS}$	**64.3**	33.35M	23.56G(4.35G*)	220.59G	1349
	ViT-B$_{160×160}^{8FPS}$	**69.5**	131.25M	8.97G*	866.73G	845

Table 7. **Practical performance of backbone adaptation strategies.** p-b is short for post-backbone information propagation. Details include backbone, frame resolution, and frame rate. ∗ means checkpoint training strategy is used.
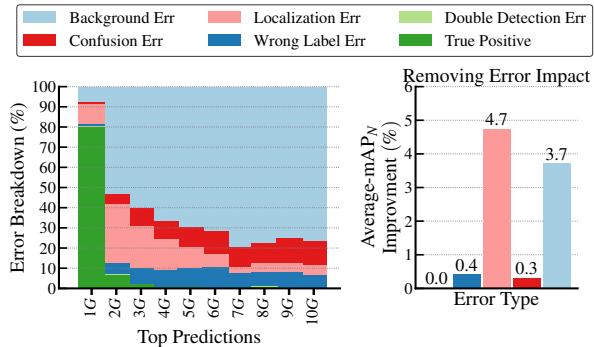

Figure 5. **Error analysis of our ViT-TAD.** There are error rates of 5 types on top-10G predictions, where G denotes the number of ground truths.

these methods to further explore the reasons why ViT-TAD achieves good detection results. Shown in Table 4, we use the feature extracted by ViT-S for ActionFormer, obtaining an average mAP of 57.9% on THUMOS14. While BasicTAD achieves an average mAP of 59.4% with a lower frame rate and resolution. To eliminate the benefit of end-to-end training manner, we treat the backbone fine-tuned by BasicTAD as a feature extractor for ActionFormer, making the detection accuracy of ActionFormer reach an average mAP of 60.3%. When we further adopt our ViT-TAD based on BasicTAD, it can achieve an average mAP of 63.0%, even higher than ActionFormer with fine-tuned features from BasicTAD. When we further compare our ViT-TAD with the state-of-the-art TAD method TriDet that inputs pre-extracted features, its performance is worse than our ViT-TAD when using pre-extracted ViT-B features based on 30 FPS frame rate and 224×224 frame resolution (compare 68.2% with 69.5%), and even worse under fair input conditions (compare 60.0% with 69.5%). Therefore we can draw several conclusions. First, the Actionformer's detector is stronger than BasicTAD. Second, BasicTAD achieves better results after using the proposed modules in ViT-TAD. Third, ViT-TAD outperforms the state-of-the-art TAD method TriDet under fair input conditions. To further verify the module's effectiveness proposed in ViT-TAD, we migrate it to another end-to-end TAD method TALLFormer that handles clip-level temporal modeling. Shown in Table 5, our proposed modules improve TALLFormer's detection results when both are applied with ViT-B provided by [38] (from 63.1% to 66.1%). In summary, ViT-TAD has significant advantages over the above TAD methods, and its performance is better than the current non-end-to-end TAD method TriDet. At the same time, the modules proposed in ViT-TAD can be adopted by other TAD methods to achieve better detection results.

Runtime Comparison with other TAD Methods. Here we list the runtime comparison with other TAD methods shown in Table 6. The time cost of optical flow extraction makes PBRNet [20] and AFSD [17] lower than the reported speed. TriDet [31] and ActionFormer [45] need to extract features of the complete video in advance, so we consider the inference speed of both feature extraction and model in-

ference. Since our method is based on ViT, its inference speed cannot be compared with convolution-based methods [22, 44], but our ViT-TAD achieves the best detection results while ensuring fast inference speed among methods using ViT.

4.2. Analysis

Efficiency Analysis of ViT-TAD. We compare the differences between baseline and variants of ViT-TAD. As is shown in Table 7, ViT-TAD introduces additional training parameters and consumes more memory due to the introduction of multiple blocks in the backbone. Thanks to checkpoint training strategy (marked by ∗ in Table 7), it can still run successfully with limited computing resources.

Error Analysis. To analyze the limitations of our model, we provide false positive error chart [1] of our ViT-TAD on THUMOS14 dataset shown in Fig 5. We obtain quite high true positive rate on Top-1G predictions. As is mentioned by BasicTAD [44], one-stage anchor-free methods suffer from "Background Error" due to limited anchors causing more predictions failing to match ground truth. A more precise regression loss design is needed.

4.3. Visualization

We provide heatmap of action "TennisSwing" in THUMOS14 for both baseline and ViT-TAD detectors to visualize inner-backbone information propagation strategy's impact on frame-level modeling. We use Grad-CAM [29] to generate corresponding heatmap in backbone's last layer. As is shown in Fig 6, due to the integration of global tem-
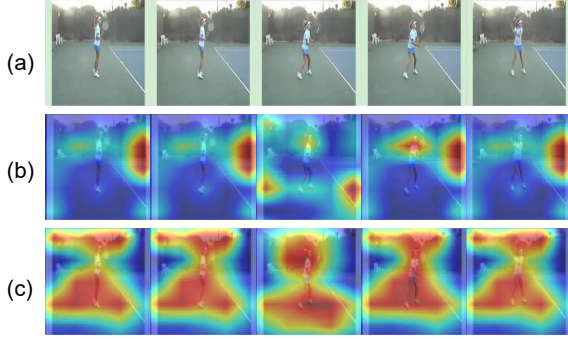
Figure 6. **Heatmap visualization.** (a): Original input frames. (b): Heatmap from baseline method. (c): Heatmap from ViT-TAD.

Method	Backb.	Flow	0.5	0.75	0.95	Avg
BSN [19]	TSN	✔	46.45	29.96	8.02	30.03
ReAct [32]	TSN	✔	49.60	33.00	8.60	32.60
BMN [18]	TSN	✔	50.07	34.78	8.29	33.85
BC-GNN [3]	TSN	✔	50.56	34.75	9.37	34.26
G-TAD [43]	TSN	✔	50.36	34.60	9.02	34.09
RTD-Net [33]	I3D	✔	47.21	30.68	8.61	30.83
VSGN [47]	TSN	✔	52.38	36.01	8.37	35.07
ActionFormer [45]	I3D	✔	53.50	36.20	8.20	35.60
TriDet [31]	I3D	✔	54.50	36.80	11.50	36.80
TriDet [31]	SlowFast	✘	56.70	39.30	11.70	**38.60**
STPT [41]$_{96\times96}$	MViT	✘	51.40	33.70	6.80	33.40
AFSD [17]$_{96\times96}$	I3D	✔	52.40	35.30	6.50	34.40
PBRNet [20]$_{96\times96}$	I3D	✔	53.96	34.97	8.98	35.01
e2e-tadtr [22]$_{112\times112}$	SlowFast-R50	✘	50.47	35.99	10.83	35.10
TALLFormer [10]$_{224\times224}$	Swin-B	✘	54.10	36.20	7.90	35.60
Re^{2}TAL [46]$_{224\times224}$	Swin-T	✘	54.75	37.81	9.03	36.80
Re^{2}TAL [46]$_{224\times224}$	SlowFast-R101	✘	55.25	37.86	9.05	37.01
ViT-TAD$_{160\times160}$	ViT-S	✘	55.09	37.81	8.75	36.69
ViT-TAD$_{160\times160}$	ViT-B	✘	55.87	38.47	8.80	**37.40**

Table 9. **Comparison with state-of-the-art methods on ActivityNet-1.3.** The subscript indicates the spatial resolution. **Flow** denotes whether each method uses optical flow as input.

Method	Backbone	Flow	0.5	0.75	0.95	Avg
BMN [18]	I3D	✔	14.44	8.92	3.12	9.25
DBG [16]	I3D	✔	10.65	6.43	2.50	6.75
G-TAD [43]	I3D	✔	13.74	8.83	3.06	9.06
ActionFormer [45]	I3D	✘	-	-	-	13.20
VideoMAE V2 [38]	ViT-G	✘	29.07	17.66	5.07	**18.24**
BasicTAD [44]$_{160\times160}$	SlowOnly-R50	✘	24.34	10.57	0.43	12.15
ViT-TAD$_{160\times160}$	ViT-B	✘	32.61	15.85	2.68	17.20

Table 10. **Comparison with state-of-the-art methods on Fine-Action.** The subscript indicates the spatial resolution. **Flow** denotes whether each method uses optical flow as input.

Method	Backb.	Flow	0.3	0.4	0.5	0.6	0.7	Avg
BSN [19]	TSN	✔	53.5	45.0	36.9	28.4	20.0	36.8
BMN [18]	TSN	✔	56.0	47.4	38.8	29.7	20.5	38.5
DBG [16]	TSN	✔	57.8	49.4	39.8	30.2	21.7	39.8
BC-GNN [3]	TSN	✔	57.1	49.1	40.4	31.2	23.1	40.2
G-TAD [43]	TSN	✔	66.4	60.4	51.6	37.6	22.9	47.8
RTD-Net [33]	I3D	✔	58.5	53.1	45.1	36.4	25.0	43.6
VSGN [47]	TSN	✔	66.7	60.4	52.4	41.0	30.4	50.2
ReAct [32]	TSN	✔	69.2	65.0	57.1	47.8	35.6	55.0
ActionFormer [45]	I3D	✔	82.1	77.8	71.0	59.4	43.9	66.8
ActionFormer [45]	ViT-B	✘	80.5	75.7	68.6	57.9	42.6	65.1
TriDet [31]	I3D	✔	83.6	80.1	72.9	62.4	47.4	69.3
VideoMAE V2 [38]	ViT-G	✘	84.0	79.6	73.0	63.5	47.7	**69.6**
PBRNet [20]$_{96\times96}$	I3D	✔	58.5	54.6	51.3	41.8	29.5	47.1
AFSD [17]$_{96\times96}$	I3D	✔	67.3	62.4	55.5	43.7	31.1	52.0
STPT [41]$_{96\times96}$	MViT	✘	70.6	65.7	56.4	44.6	30.5	53.6
DaoTAD [37]$_{112\times112}$	res50-I3D	✘	62.8	59.5	53.8	43.6	30.1	50.0
e2e-tadtr [22]$_{112\times112}$	SlowFast-R50	✘	69.4	64.3	56.0	46.4	34.9	54.2
TALLFormer [10]$_{224\times224}$	Swin-B	✘	76.0	-	63.2	-	34.5	59.2
TALLFormer [10]$_{224\times224}$	ViT-B	✘	78.9	75.0	67.6	56.1	37.9	63.1
BasicTAD [44]$_{160\times160}$	SlowOnly-R50	✘	75.5	70.8	63.5	50.9	37.4	59.6
Re^{2}TAL [46]$_{224\times224}$	Swin-T	✘	77.0	71.5	62.4	49.7	36.3	59.4
Re^{2}TAL [46]$_{224\times224}$	SlowFast-R101	✘	77.4	72.6	64.9	53.7	39.0	61.5
BasicTAD [44](baseline)$_{112\times112}$	ViT-S	✘	74.6	70.2	62.6	51.3	38.4	59.4
ViT-TAD$_{112\times112}$	ViT-S	✘	78.7	74.1	66.2	55.5	40.3	63.0
ViT-TAD$_{160\times160}$	ViT-S	✘	79.8	75.2	68.4	56.4	41.7	64.3
ViT-TAD$_{160\times160}$	ViT-B	✘	85.1	80.9	74.2	61.8	45.4	**69.5**

Table 8. **Comparison with state-of-the-art methods on THUMOS14.** The subscript indicates the spatial resolution. **Flow** denotes whether each method uses optical flow as input.

poral information, the model prioritizes the regions of the image that are relevant for action recognition, encompassing the athlete's body and the pertinent background during the action. When the model is restricted to short-term observations, it may lead to the model diverting its attention to irrelevant or imprecise areas.

4.4. Comparison with the State of the Art

We compare our ViT-TAD with the previous state-of-the-art methods on THUMOS14 [13], ActivityNet-1.3 [6] and FineAction [23]. We classify these methods into end-to-end methods (lower part of tables) and non-end-to-end methods (upper part of tables). For THUMOS14, the results are shown in Table 8. Our ViT-TAD with ViT-B outperforms all TAD methods except VideoMAE V2 which adopts ViT-G. For ActivityNet-1.3, the results are shown in Table 9. We attempt to adapt our model to predict binary action proposals and obtain the detection results by applying video-level ac-

tion classifiers [42]. We get competitive results with ViT-S (36.69%) and state-of-the-art results with ViT-B (37.40%) among all end-to-end TAD methods. For FineAction, the results are shown in Table 10. We get good results with ViT-B (17.20%), only slightly lower than VideoMAE V2 [38].

5. Conclusion

In this paper, we have presented a simple TAD framework (ViT-TAD) based on the plain ViT backbone. Our ViT-TAD incorporates the inner-backbone propagation module and post-backbone propagation module to capture more fine-grained temporal information across different snippets and global contexts. We perform in-depth ablation studies on the design of different components in ViT-TAD. With the simple TAD head and powerful masked video pre-training, our ViT-TAD yields a state-of-the-art performance compared with other end-to-end methods on the challenging datasets THUMOS14, ActivityNet-1.3 and FineAction. We hope it will serve as a new TAD baseline for future research.

References

[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *ECCV*, pages 256–272, 2018. 7

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, pages 6836–6846, 2021. 2

[3] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *ECCV*, pages 121–137. Springer, 2020. 8

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 2

[5] Adrian Bulat, Juan Manuel Perez Rua, Swathikiran Sudhakaran, Brais Martinez, and Georgios Tzimiropoulos. Space-time mixing attention for video transformer. *NIPS*, 34:19594–19607, 2021. 2

[6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 1, 2, 5, 8

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 2, 4

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 1

[9] Shimin Chen, Chen Chen, Wei Li, Xunqiang Tao, and Yandong Guo. Faster-tad: Towards temporal action detection with proposal generation and classification in a unified network. *CoRR*, abs/2204.02674, 2022. 1, 2

[10] Feng Cheng and Gedas Bertasius. Tallformer: Temporal action localization with a long-memory transformer. In *ECCV*, pages 503–521. Springer, 2022. 1, 2, 3, 6, 8

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021. 1, 2, 6

[12] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6824–6835, 2021. 1, 2

[13] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014. 1, 2, 5, 8

[14] 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. *CoRR*, abs/1705.06950, 2017. 5

[15] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, pages 280–296. Springer, 2022. 2, 4, 5

[16] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *AAAI*, pages 11499–11506, 2020. 8

[17] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *CVPR*, pages 3320–3329, 2021. 1, 2, 3, 5, 7, 8

[18] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, pages 3889–3898, 2019. 1, 3, 8

[19] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, pages 3–19, 2018. 3, 8

[20] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *AAAI*, volume 34, pages 11612–11619, 2020. 3, 7, 8

[21] Shuming Liu, Chen-Lin Zhang, Chen Zhao, and Bernard Ghanem. End-to-end temporal action detection with 1b parameters across 1000 frames. *arXiv preprint arXiv:2311.17241*, 2023. 3

[22] Xiaolong Liu, Song Bai, and Xiang Bai. An empirical study of end-to-end temporal action detection. In *CVPR*, pages 19978–19987. IEEE, 2022. 2, 7, 8

[23] Yi Liu, Limin Wang, Yali Wang, Xiao Ma, and Yu Qiao. Fineaction: A fine-grained video dataset for temporal action localization. *TIP*, 31:6937–6950, 2022. 1, 2, 5, 8

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 2

[25] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, pages 3202–3211, 2022. 1, 2, 4

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*. OpenReview.net, 2019. 5

[27] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *CoRR*, abs/2102.00719, 2021. 2

[28] AJ Piergiovanni, Weicheng Kuo, and Anelia Angelova. Rethinking video vits: Sparse video tubes for joint image and video learning. *CoRR*, abs/2212.03229, 2022. 2

[29] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017. 7

[30] Gilad Sharir, Asaf Noy, and Lihi Zelnik-Manor. An image is worth 16x16 words, what is a video worth? *CoRR*, abs/2103.13915, 2021. 2

[31] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. Tridet: Temporal action detection with relative boundary modeling. In *CVPR*, pages 18857–18866, 2023. 6, 7, 8

[32] Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. React: Temporal action detection with relational queries. In *ECCV*, pages 105–121. Springer,

2022. 2, 8

[33] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *ICCV*, pages 13526–13535, 2021. 2, 8

[34] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *NIPS*, 35:10078–10093, 2022. 1, 2

[35] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 3, 5

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 30, 2017. 1, 2

[37] Chenhao Wang, Hongxiang Cai, Yuxin Zou, and Yichao Xiong. RGB stream is enough for temporal action detection. *CoRR*, abs/2107.04362, 2021. 8

[38] 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 *CVPR*, pages 14549–14560, 2023. 5, 7, 8

[39] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016. 1

[40] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. Internvideo: General video foundation models via generative and discriminative learning. *CoRR*, abs/2212.03191, 2022. 2

[41] Yuetian Weng, Zizheng Pan, Mingfei Han, Xiaojun Chang, and Bohan Zhuang. An efficient spatio-temporal pyramid transformer for action detection. In *ECCV*, pages 358–375. Springer, 2022. 1, 2, 3, 8

[42] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang. Cuhk & ethz & siat submission to activitynet challenge 2016. *arXiv preprint arXiv:1608.00797*, 2016. 8

[43] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, pages 10156–10165, 2020. 3, 8

[44] Min Yang, Guo Chen, Yin-Dong Zheng, Tong Lu, and Limin Wang. Basictad: an astounding rgb-only baseline for temporal action detection. *CVIU*, 232:103692, 2023. 1, 2, 5, 6, 7, 8

[45] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *ECCV*, pages 492–510. Springer, 2022. 2, 6, 7, 8

[46] Chen Zhao, Shuming Liu, Karttikeya Mangalam, and Bernard Ghanem. Re2tal: Rewiring pretrained video backbones for reversible temporal action localization. In *CVPR*, pages 10637–10647, 2023. 1, 2, 3, 8

[47] Chen Zhao, Ali K. Thabet, and Bernard Ghanem. Video self-stitching graph network for temporal action localization. In *ICCV*, pages 13638–13647, 2021. 3, 8