An Efficient Spatio-Temporal Pyramid Transformer for Action Detection

Yuetian Weng[1], Zizheng Pan[1], Mingfei Han[1,2],
Xiaojun Chang[2,3], and Bohan Zhuang[1†]

[1] Data Science & AI, Monash University
[2] ReLER Lab, AAII, University of Technology Sydney
[3] School of Computing Technologies, RMIT University
{yuetian.weng,zizheng.pan,bohan.zhuang}@monash.edu, hmf282@gmail.com,
xiaojun.chang@uts.edu.au

Abstract. The task of action detection aims at deducing both the action category and localization of the start and end moment for each action instance in a long, untrimmed video. While vision Transformers have driven the recent advances in video understanding, it is non-trivial to design an efficient architecture for action detection due to the prohibitively expensive self-attentions over a long sequence of video clips. To this end, we present an efficient hierarchical Spatio-Temporal Pyramid Transformer (STPT) for action detection, building upon the fact that the early self-attention layers in Transformers still focus on local patterns. Specifically, we propose to use local window attention to encode rich local spatio-temporal representations in the early stages while applying global attention modules to capture long-term space-time dependencies in the later stages. In this way, our STPT can encode both locality and dependency with largely reduced redundancy, delivering a promising trade-off between accuracy and efficiency. For example, with only RGB input, the proposed STPT achieves 53.6% mAP on THUMOS14, surpassing I3D+AFSD RGB model by over 10% and performing favorably against state-of-the-art AFSD that uses additional flow features with 31% fewer GFLOPs, which serves as an effective and efficient end-to-end Transformer-based framework for action detection.

Keywords: Action Detection, Efficient Video Transformers

1 Introduction

Action detection in lengthy, real-world videos is one of the crucial tasks in many video analysis applications, *e.g.*, sports analysis, autonomous driving. Action detection aims to localize and classify the action instances appearing in untrimmed videos, which essentially depends on learning strong spatio-temporal representations from videos.

[†] Corresponding author.

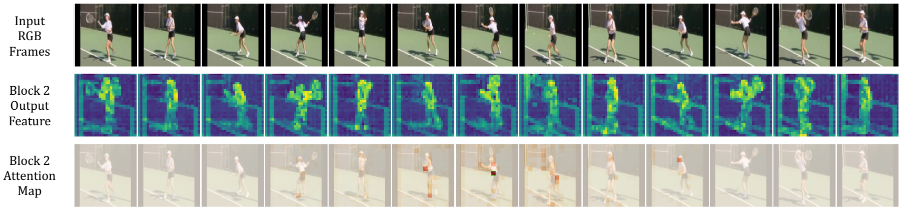

Fig. 1: Visualization of MViT [19]. We show the sampled input RGB frames, encoded feature maps and attention maps from the 2^{nd} block of MViT. We find that such global attention in the early stages actually encodes local visual patterns in each RGB frame, but it is relatively redundant in successive frames. For illustration, a target query token in the middle frame (green anchor) only attends to its nearby tokens in adjacent frames while rarely interacting with tokens in distant frames (filled with red, the more darker the color, the higher the attention score). Therefore, attending to all spatio-temporal tokens leads to huge redundancy in encoding such local patterns.

To date, the majority of action detection methods [66,32,38,54,12] are driven by 3D convolutional neural networks (CNNs), *e.g.*, C3D [56], I3D [10], to encode video segment features from video RGB frames and optical flows [71]. 3D convolution is compact and effective to aggregate contextual pixels within a small 3D region, *e.g.*, $3 \times 3 \times 3$, and thus reduce the spatio-temporal redundancy. However, the limited receptive field hinders the CNN-based models to capture long-term spatio-temporal dependencies. Alternatively, vision Transformers (ViTs) have shown the advantage [50] of capturing global dependencies via the self-attention mechanism in many computer vision tasks, such as image classification [18,55,47] and video action recognition [4,75,19,24]. Hierarchical ViTs [19] divide Transformer blocks into several stages and progressively reduce the spatial size of feature maps when the network goes deeper. However, the high-resolution feature maps of video clips in the early stages result in overlong token sequences. For instance, given an input video clip with $256 \times 96 \times 96$ RGB frames, the feature maps after the initial embedding layer requires more than 1000G Floating-point Operations (FLOPs) for a standard multi-head self-attention layer, which is impractical to train or evaluate. Therefore, how to efficiently handle spatio-temporal dependencies across overlong video frames is a fundamental challenge for action detection.

In this paper, we present an efficient Spatio-Temporal Pyramid Transformer (STPT) to tackle both spatio-temporal redundancy and long-range dependency, as illustrated in Fig. 2. Specifically, we propose a Local Spatio-Temporal Attention blocks (LSTA) to capture local patterns in the early stages while introducing Global Spatio-Temporal Attention blocks (GSTA) to handle long-range spatio-temporal relationships in the later stages. The motivation of this design comes from two aspects. First, considering the spatial dimension in videos, previous studies in CNNs and ViTs [27,65,50] have shown that shallow layers tend to capture local patterns in images (*e.g.*, texture, edges) while deep layers tend to learn high-level semantics or capture long-range dependencies. Besides, target motions across adjacent frames are subtle, which implies large temporal redundancy when encoding video representations [21,62,33]. Therefore, it naturally gives rise to the question of whether applying global attention at the early stages to encode spatio-temporal representations is necessary. Second, we empirically observe that heavy spatio-temporal redundancy exists in the shallow stages of current video Transformers [51,4,1,75,8]. We take the 2^{nd} block of MViT [19] as an example and visualize its output features as well as the attention maps in Fig. 1. We observe that the self-attention in the shallow layers mainly focuses on neighboring tokens in a small spatial area and adjacent frames, rarely attending to other tokens in distant frames. Hence, aggregating all the visual tokens via self-attention in the early stages can bring noises to the informative local representations and incurs huge computational redundancy. By leveraging LSTA in the early stages, STPT significantly alleviates spatio-temporal redundancy and inherently benefits spatio-temporal representation learning, as target motions are highly correlated in a local spatial region and temporally subtle across adjacent frames.

To encourage locality inductive bias, recent studies propose to combine convolutions [63], MLPs [46] with Transformers or restrict self-attention within local windows [41], achieving favorable performance with reduced computational complexity. Moreover, from the theoretical perspective, locality inductive bias suppresses the negative Hessian eigenvalues, thus assisting in optimization by convexifying the loss landscape [48]. Different from these methods, we are the pioneering work to build a pure Transformer model that encodes both compact spatial and temporal locality while preserving the long-range dependency for the action detection task.

Finally, the proposed efficient Transformer is equipped with a temporal feature pyramid network (TFPN) to progressively reduce the spatial and temporal dimension and enlarge the receptive field into different scales. The multi-scale spatio-temporal feature representations are further utilized to predict the temporal boundaries and categories via an anchor-free prediction and refinement module.

In summary, our contributions are in three folds:

- We propose an efficient and effective Spatio-Temporal Pyramid Transformer (STPT) for action detection, which reduces the huge computational cost and redundancy while capturing long-range dependency in spatio-temporal representation learning.
- We devise local window attention to enhance local representations while reducing the spatio-temporal redundancy in shallow layers and retain the long-range dependency in deep layers with global self-attentions, achieving a favourable balance between efficiency and effectiveness.
- Finally, we conduct extensive experiments on standard benchmarks, *i.e.*, THUMOS14 and ActivityNet 1.3, by using pure RGB frame input. Compared with

the methods that combining additional flow features, our STPT achieves state-of-the-art results with reduced computational complexity, which makes a substantial stride for Transformer on the task of video action detection.

2 Related Work

2.1 Action Detection

Action detection aims at localizing the temporal boundaries of human activities in untrimmed videos and classifying the action categories [57]. Most existing works [22,32,38] utilize CNN-based models [56,10,49,60] pretrained on large-scale datasets (*e.g.*, Kinetics400 [10]) to extract spatio-temporal representations from a stack of RGB frames and/or optical flow frames. Anchor-based methods [22,66,13] retrieve fine-grained proposals by adjusting pre-defined multi-scale anchors while actionness-guided methods [38,36,77] instead learn the boundary confidence or actionness scores at all the temporal positions of the input video, which are matched and served as proposal candidates. Another line of research resorts to multi-scale towers [13,23] or temporal feature pyramids [37,74,40] to tackle the variation of action duration, utilizing high-resolution feature maps for short actions and feature maps with large receptive field for long actions, respectively. Recently, a new anchor-free detector [35] directly predicts the distance to the action boundaries and the action category for each frame. However, the local receptive field of 3D convolutions leads to the loss of temporal dependencies on untrimmed videos. To capture the action dependencies across frames, prior works introduce graph models [72,32,3,67,76], RNNs [7,6,70], and temporal Transformers [44,54,11] to capture these temporal relationships. However, the aforementioned methods rely on pre-extracted features from 3D convolution backbones and use head-only learning manner. In contrast, our STPT devises a pure Transformer model for efficiently and effectively learning spatio-temporal representations in an end-to-end manner, which encodes local patterns and global dependency via flexible token affinity learning in shallow and deep layers, respectively.

2.2 Video Transformers

ViTs are pushing the boundaries of recent video understanding research. In particular, VTN [45], LightVideoFormer [31] and STAM [51] introduce temporal Transformers to encode inter-frame relationships over the extracted image-level feature maps. ViViT [1], TimeSformer [4] and VidTr [75] propose to factorize along spatial and temporal dimensions on the granularity of encoder, attention block or dot-product computation. Similarly, SCT [73] proposes image chunk attention and shifted attention to model spatial and temporal relationships respectively. SMAVT [8] aggregates information from tokens located at the same spatial location within a local temporal window, while SIFAR [20] turns spatio-temporal patterns in video into purely spatial patterns in images, showing an

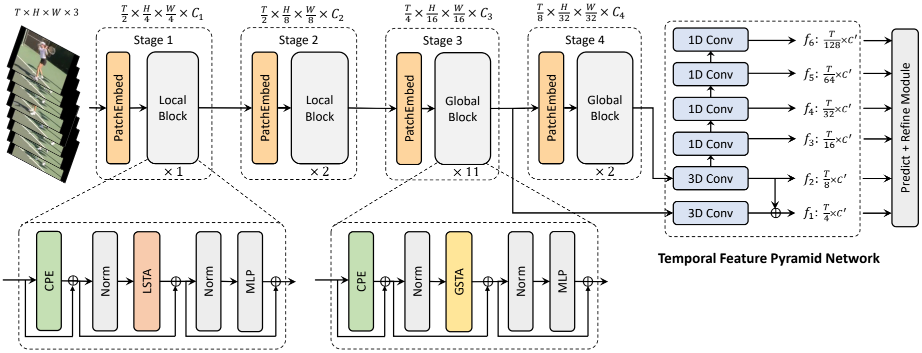
$T \times H \times W \times 3$
$\frac{T}{2} \times \frac{H}{4} \times \frac{W}{4} \times C_1$
Stage 1
PatchEmbed
Local Block
× 1
$\frac{T}{2} \times \frac{H}{8} \times \frac{W}{8} \times C_2$
Stage 2
PatchEmbed
Local Block
× 2
$\frac{T}{4} \times \frac{H}{16} \times \frac{W}{16} \times C_3$
Stage 3
PatchEmbed
Global Block
× 11
$\frac{T}{8} \times \frac{H}{32} \times \frac{W}{32} \times C_4$
Stage 4
PatchEmbed
Global Block
× 2
1D Conv
1D Conv
1D Conv
1D Conv
3D Conv
3D Conv
f_6: $\frac{T}{128} \times c'$
f_5: $\frac{T}{64} \times c'$
f_4: $\frac{T}{32} \times c'$
f_3: $\frac{T}{16} \times c'$
f_2: $\frac{T}{8} \times c'$
f_1: $\frac{T}{4} \times c'$
Predict + Refine Module
Temporal Feature Pyramid Network
CPE
Norm
LSTA
Norm
MLP
CPE
Norm
GSTA
Norm
MLP

Fig. 2: Overall architecture of the proposed spatio-temporal pyramid Transformer. Given an input video X_{in}, our STPT is utilized to encode spatio-temporal representations and detect existing action instances. In the shallow layers, local blocks that constrain self-attention into local 3D windows tend to encode local patterns while reducing spatio-temporal redundancy. In deeper layers, global blocks retain long-range dependency over the token sequence. Features from the last stages are fed to the temporal feature pyramid network to predict the temporal boundaries and action categories via an anchor-free prediction and refinement module. Please refer to Sec. 3.1 for additional details.

image classifier can undertake the task of video understanding. However, these studies lack hierarchical structure or model spatio-temporal dependencies separately, which may not be sufficient for the task of action detection. Targeting on these issues, MViT [19] presents a hierarchical Transformer to progressively shrink the spatio-temporal resolution of feature maps while expanding the channel as the network goes deeper. VideoSwin [42] proposes shifted window attention to limit the computation within a small local window, while Uniformer [33] unifies the spatio-temporal MobileNet block and self-attention and proposes an alternative multi-head relation aggregator. Different from the above methods, our model is purely Transformer based and jointly learns spatio-temporal representation. By flexibly involving locality constraint in early stages and data-specific global self-attentions in later stages, our model well addresses the challenges of spatio-temporal redundancy and dependency for action detection.

3 Method

3.1 Overall Architecture

The overall architecture of STPT is illustrated in Fig. 2. Let $X_{in} \in \mathbb{R}^{T \times H \times W \times 3}$ be an input video clip, where T, H and W represent the number, height and width of RGB frames, respectively. First, we divide the frame volume X_{in} into a series of overlapping 3D cubes, with the size of $3 \times 7 \times 7$. Then we exploit a 3D PatchEmbed module to aggregate contextual pixels from each cube and project each cube into dimension of 96, serving as the initial input for the subsequent pipeline. In practice, the 3D PatchEmbed module is a $3 \times 7 \times 7$ depth-wise convolution with stride $2 \times 4 \times 4$ and zero paddings. As a result, the input tensor X_{in} is downsampled to $\frac{T}{2} \times \frac{H}{4} \times \frac{W}{4}$.

Next, we divide the entire backbone into 4 stages. Letting $s \in [1, 2, 3, 4]$ be the index of each stage, we employ $L_s \in [1, 2, 11, 2]$ blocks at each stage, in which STPT Local Blocks are used to encode local spatio-temporal representations in the first two stages, and STPT Global Block are used to tackle long-term dependencies in the later stages. At the same time, from the 2^{nd} stage, the spatial and temporal dimensions are gradually downsampled by the 3D PatchEmbed module. Following [19], by increasing the output dimension of the final MLP layer in the previous stage, the channel dimensions are gradually expanded before the transition to the next stage. The feature maps with different resolutions from the last two stages are then fed to the temporal feature pyramid network (TFPN) to obtain multiple resolution feature maps. Finally, the prediction and refinement modules are used to predict the start frame, end frame and the category of each anchor point on the multi-scale feature maps. The detailed architecture specifications are provided in Table 1.

3.2 Block Design in STPT

Formally, each STPT Block consists of three key modules: Conditional Positional Encoding (CPE), Multi-Head Spatio-Temporal Attention (MSTA), and multi-layer perceptron (MLP). Formally, letting X_{l-1} be the input of the l-th block, each block can be formulated as

$$X_{l-1} = \mathrm{CPE}(X_{l-1}) + X_{l-1} \tag{1}$$

$$X'_{l-1} = \mathrm{MSTA}(\mathrm{LN}(X_{l-1})) + X_{l-1} \tag{2}$$

$$X_l = \mathrm{MLP}(\mathrm{LN}(X'_{l-1})) + X'_{l-1} \tag{3}$$

where $\mathrm{LN}(\cdot)$ indicates the layer normalization [2] and MLP consists of two FC layers with GELU [26] non-linearity in between. Specifically, we first leverage CPE to integrate the spatio-temporal position information into each token. Then the MSTA module, which can be the Local spatio-temporal Attention (LSTA) or Global spatio-temporal Attention (GSTA), aggregates each token with its contextual tokens, followed by an MLP to perform channel mixing. As discussed above, we encode fine and local spatial-temporal representations in the early stages using LSTA and high-level semantics with long-term dependencies in the deep layers using GSTA, respectively. In the next, we elaborate the design for each module.

Conditional positional encoding. Since actions in videos are both spatial and temporal variant, we need to explicitly encode position information for all the visual tokens. Previous works [19,1,4] commonly adopt absolute positional encodings [18]. However, the length of token sequences is much longer for action detection compared to the one pretrained for action recognition tasks (*e.g.*, $128 \times 24 \times 24$ vs. $8 \times 56 \times 56$), which makes it difficult to utilize a fixed absolute positional

Table 1: Architecture specification of STPT. For the s-th stage, we denote P_s as the patch size, S_s as the kernel stride and R_s as the reduction ratio, where each dimension corresponding to the temporal size, height and width, respectively. L_s and C_s refer to the the number of blocks and the channel dimension at the s-th stage, respectively. Furthermore, we denote W_{sl} as the window size at the l-th block in the s-th stage. We adopt an expanding ratio of 4 for all MLP layers in each block.

	Output Size	Layer Name	STPT
Stage1	$\frac{T}{2} \times \frac{H}{4} \times \frac{W}{4}$	PatchEmbed	$P_1 = 3 \times 7 \times 7$ $S_1 = 2 \times 4 \times 4$ $C_1 = 96$
		Local Block	$L_1 = 1$ $W_{11} = 8 \times 8 \times 8$ $R_1 = 2 \times 8 \times 8$
Stage2	$\frac{T}{2} \times \frac{H}{8} \times \frac{W}{8}$	PatchEmbed	$P_2 = 3 \times 3 \times 3$ $S_2 = 1 \times 2 \times 2$ $C_2 = 192$
		Local Block	$L_2 = 2$ $W_{21} = 8 \times 6 \times 6$ $W_{22} = 16 \times 4 \times 4$ $R_2 = 2 \times 2 \times 2$
Stage3	$\frac{T}{4} \times \frac{H}{16} \times \frac{W}{16}$	PatchEmbed	$P_3 = 3 \times 3 \times 3$ $S_3 = 2 \times 2 \times 2$ $C_3 = 384$
		Global Block	$L_3 = 11$ $R_3 = 2 \times 2 \times 2$
Stage4	$\frac{T}{8} \times \frac{H}{32} \times \frac{W}{32}$	PatchEmbed	$P_4 = 3 \times 3 \times 3$ $S_4 = 2 \times 2 \times 2$ $C_4 = 768$
		Global Block	$L_4 = 2$ $R_4 = 1 \times 1 \times 1$

encoding from a pretrained model. To tackle these problems, we make the spatio-temporal positional embedding conditioned on input features and extend the CPE proposed in [17] to the video domain, which can be formulated as

$$\mathrm{CPE}(X) = \mathrm{DWConv}(X), \tag{4}$$

where DWConv refers to a 3D depth-wise convolution with zero paddings. Previous works [17,28] have shown that tokens on the borders can be aware of their absolute positions when using convolutional layers with zero paddings. Therefore, the absolute position for each token can be encoded by progressively sliding convolutional kernels on the feature maps, justifying the design of our proposed CPE for introducing positional information into input features.

Local spatio-temporal attention. As discussed in Sec. 1, global attention is redundant to encode local patterns in the shallow layers, thus leading to high computational cost owing to the high-resolution feature maps. Given an embedding $X \in \mathbb{R}^{T\times H\times W\times d}$ with channel dimension d, the complexity of self-attention is $\mathcal{O}(T^2H^2W^2d)$ [58]. Here, we propose to replace self-attention with LSTA to alleviate the redundancy.

Following the design of multi-head self-attention, LSTA first projects X into query $Q \in \mathbb{R}^{T\times H\times W\times d}$, key $K \in \mathbb{R}^{T\times H\times W\times d}$, and value $V \in \mathbb{R}^{T\times H\times W\times d}$ with linear transformations. For each tensor, we evenly divide it into $w_1 \times w_2 \times w_3$ partitions (sub-windows). Without loss of generality, we assume $T\%w_1 = 0$, $H\%w_2 = 0$ and $W\%w_3 = 0$, and thus each sub-window contains $\frac{THW}{w_1w_2w_3}$ tokens. We force each query token only attends to tokens within the same local 3D window, which helps encode local patterns via joint spatial-temporal relation aggregation and reduce the computational redundancy in the early stages.

To further improve the efficiency, we follow [61,19] to reduce both the spatial and temporal resolution of the keys and values within each local window, *i.e.*, $\bar{K} = R_K(K) \in \mathbb{R}^{T'\times H'\times W'\times d}$ and $\bar{V} = R_V(V) \in \mathbb{R}^{T'\times H'\times W'\times d}$, where $\bar{K}$, $\bar{V}$ are resolution-reduced keys and values, and R_K, R_V denote two independent reduction operations (*e.g.*, depth-wise convolution). T', H' and W' are the reduced temporal dimension, height and width. Thus, each sub-window of $\bar{K}$ and $\bar{V}$ contains $\frac{T'H'W'}{w_1w_2w_3}$ tokens after being divided into $w_1 \times w_2 \times w_3$ partitions. Specifically, the computational cost for each sub-window becomes $\mathcal{O}(\frac{THW}{w_1w_2w_3} \times \frac{T'H'W'}{w_1w_2w_3} \times d)$, and the total cost of LSTA is $\mathcal{O}(\frac{T'H'W'}{w_1w_2w_3} \times THWd)$, which is significantly efficient when $\frac{T'}{w_1} \ll T$, $\frac{H'}{w_2} \ll H$ and $\frac{W'}{w_3} \ll W$ and grows linearly with THW if $\frac{T'}{w_1}$, $\frac{H'}{w_2}$ and $\frac{W'}{w_3}$ are fixed.

By applying LSTA in the early stages, the model significantly alleviates spatio-temporal redundancy and efficiently encodes local spatio-temporal representations.

Global spatio-temporal attention. To capture long-term dependencies, we employ GSTA in the deep layers. For more efficient aggregation, GSTA also uses the feature maps with reduced spatio-temporal resolution as the keys and values in the self-attention operations. Given a query token, GSTA compares it with all the tokens for aggregation. In this way, we ensure that the model captures global dependencies in the last stages. By combining LSTA in the shallow layers, the model forms an efficient and effective way of learning the spatio-temporal representations for action detection.

Relation to existing video Transformers. While [4,1] are based on space-time attention factorization, our method can encode the target motions by jointly aggregating spatio-temporal relations, without loss of spatio-temporal correspondence. Compared with MViT [19] which entirely utilizes global self-attentions, our model can resolve the long-range dependency while simultaneously reducing the local spatio-temporal redundancy. By removing the local redundancy in the early stages, our model outperforms MViT with a lower computational cost. Moreover, different from the spatio-temporal MobileNet block used in Uniformer [33], our LSTA is data-dependent [48,25] and flexible in terms of window size without introducing extra parameters, while the kernel parameters are fixed for the 3D convolutions in the spatio-temporal MobileNet block. Different from

VideoSwin [42], we do not use shifted window mechanism to get a trade-off between locality and dependency. Compared with [14,16,34] which alternatively process local and global information within each block, we encode local and global spatio-temporal representations in the shallow and deep layers separately, tackling both redundancy and dependency in a concise manner.

3.3 Temporal Feature Pyramid

Given an untrimmed video, action detection aims to find the temporal boundaries and categories of action instances, with annotation denoted by $\{\psi_n = (t_n^s, t_n^e, c_n)\}_{n=1}^N$, where N is the number of action instances. For the n-th action instance ψ_n, t_n^s, t_n^e and c_n refer to the start time, end time and action label, respectively. As shown in Fig. 2, we first forward the video input X_{in} into the backbone to encode the spatio-temporal representations. The 3D feature maps extracted from the last two stages are then fed to TFPN to obtain multi-scale temporal feature maps. The motivation comes from the fact that multi-scale feature maps contribute to tackle the variation of action duration [13,23,64]. Specifically, we construct an M-level temporal feature pyramid $\{f_m\}_{m=1}^M$, where $f_m \in \mathbb{R}^{T_m \times C'}$ and T_m is the temporal dimension of the m-th level. The TFPN contains two 3D convolution layers followed by four 1D convolutional layers to progressively forms a featural hierarchy.

After obtaining the temporal feature pyramid, an anchor-free prediction and refinement module as in [35] is utilized to predict the boundary distances and class scores at each location i on f_m. Concretely, a two-branch tower including several temporal convolutional layers is employed to map f_m into two latent representations. The latent representations are then processed by a classification head and a localization head to get the class label $\hat{y}_i^C$ and boundary distances $(\hat{b}_i^s, \hat{b}_i^e)$ for each location i, respectively. To improve the confidence of the predictions, we further adjust the boundary distances with features extracted from a small region at the coarse boundary predicted above to obtain the modified offset as $(\Delta\hat{b}_i^s, \Delta\hat{b}_i^e)$, and the refinement action category label as $\hat{y}_i^R$. To obtain high quality proposals, we additionally predict the quality confidence η following [77]. Formally, for the i-th temporal location in the m-th TFPN layer, the final predicted start time $\hat{t}_{m,i}^s$, end time $\hat{t}_{m,i}^e$ and class label $\hat{y}_{m,i}$ can be formulated in the following form:

$$\hat{t}_{m,i}^s = \hat{b}_{m,i}^s + \frac{1}{2}(\hat{b}_{m,i}^e - \hat{b}_{m,i}^s)\Delta\hat{b}_i^s, \tag{5}$$

$$\hat{t}_{m,i}^e = \hat{b}_{m,i}^e + \frac{1}{2}(\hat{b}_{m,i}^e - \hat{b}_{m,i}^s)\Delta\hat{b}_i^e, \tag{6}$$

$$\hat{y}_{m,i} = \frac{1}{2}(\hat{y}_{m,i}^C + \hat{y}_{m,i}^R)\eta_{m,i}. \tag{7}$$

In the training process, we use a multi-task loss function based on the output of coarse and refined predictions, which can be formulated as

$$\mathcal{L} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{loc}\mathcal{L}_{loc} + \lambda_q\mathcal{L}_q, \tag{8}$$

where $\mathcal{L}_{cls}$, $\mathcal{L}_{loc}$ and $\mathcal{L}_q$ are losses corresponding to the classification, boundary regression and quality confidence prediction tasks, respectively, and λ_{cls}, λ_{loc}, λ_q are hyperparameters to balance the contribution of each task to the total loss. For the classification task, we use focal loss [39] between the predicted action scores from both prediction and refinement modules and the ground-truth categories, *i.e.*, $\mathcal{L}_{cls} = \mathcal{L}^C_{focal} + \mathcal{L}^R_{focal}$. The localization loss includes tIoU (temporal Interaction over Union) loss for the predicted coarse boundaries and L1 loss for refined offsets, respectively, *i.e.*, $\mathcal{L}_{loc} = \mathcal{L}^C_{tIoU} + \mathcal{L}^R_{L1}$. For the quality prediction task, $\mathcal{L}_q$ is computed in the same way as in [35].

4 Experiments

4.1 Datasets and Settings

Datasets. We present our experimental results on the commonly-used benchmarks THUMOS14 [29] and ActivityNet 1.3 [9]. THUMOS14 dataset is composed of 413 temporally annotated untrimmed videos with 20 action categories. We use the 200 videos in the validation set for training and evaluate our method on the 213 videos in the test set. ActivityNet 1.3 is a large-scale action understanding dataset for action recognition, action detection, proposal generation and dense captioning tasks, which contains 19,994 temporally labeled untrimmed videos with 200 action categories. We follow the former setting [38] to split this dataset into training, validation and testing sets based on the proportion of 2:1:1.
Metrics. We adopt mean Average Precision (mAP) at certain tIoU thresholds as the evaluation metric. On THUMOS14, we use tIoU thresholds $\{0.3, 0.4, 0.5, 0.6, 0.7\}$; on ActivityNet 1.3, we choose 10 values in the range of $[0.5, 0.95]$ with a step size of 0.05 as tIoU thresholds following the official evaluation API.
Implementation details. We build our STPT based on MViT [19]. The pipeline and architecture specifications have shown in Fig. 2 and Table 1, respectively. Follow common practice [35], we train and evaluate our model in an end-to-end manner, which takes as input pure RGB frames without using additional optical flow features. On THUMOS14, we sample RGB frames at 10 frames per second (fps) and split a video into clips, where each clip contains 256 frames. Adjacent clips have a temporal overlap of 30 and 128 frames at training and testing, respectively. For ActivityNet 1.3, we sample a clip of 768 frames at dynamic fps for each video. We set the spatial resolution as 96×96 and use data augmentation including random crop and horizontal flipping in training. For a fair comparison, we pretrain all the models on Kinetics400 [10] for 30 epochs under the same settings following [19]. Our model is trained for 16 epochs and 12 epochs on THUMOS14 and ActivityNet 1.3, respectively, using Adam [30] with a learning rate of 5×10^{-6} for backbone and 1×10^{-4} for other modules, and the weight decay is set to 1×10^{-3} and 1×10^{-4} for the two datasets. In post-processing, we apply soft-NMS [5] to suppress redundant predictions, where the tIoU threshold is set to 0.5 for THUMOS14 and 0.85 for ActivityNet 1.3. λ_{loc} is set to 10 for THUMOS14 and 1 for ActivityNet 1.3, and λ_{cls}, λ_q is set to 1.

Table 2: Performance comparison with state-of-the-art methods on THUMOS14, We measure the performance by mAP at different tIoU thresholds and average mAP in [0.3 : 0.1 : 0.7]. "*" indicates that the models are trained in an end-to-end manner. We measure the computational cost by GFLOPs based on a clip of 256 × 96 × 96 frames for them. "Flow" indicates using optical flow features.

Methods	GFLOPs	Backbone	Flow	0.3	0.4	0.5	0.6	0.7	Avg.
BSN [38]	455.4	TS	✓	53.5	45.0	36.9	28.4	20.0	36.8
BMN [36]	455.4	TS	✓	56.0	47.4	38.8	29.7	20.5	38.5
G-TAD [67]	444.2	TSN	✓	54.5	47.6	40.2	30.8	23.4	39.3
TAL [13]	157.0	I3D	✓	53.2	48.5	42.8	33.8	20.8	39.8
A2Net [69]	157.0	I3D	✓	58.6	54.1	45.5	32.5	17.2	41.6
G-TAD+PGCN [72]	157.0	I3D	✓	66.4	60.4	51.6	37.6	22.9	47.8
BMN-CSA [53]	455.4	TS	✓	64.4	58.0	49.2	38.2	27.8	47.5
AFSD* [35]	162.2	I3D	✓	67.3	62.4	55.5	43.7	31.1	52.0
DCAN [15]	444.2	TSN	✓	68.2	62.7	54.1	43.9	**32.6**	52.3
R-C3D* [66]	453.3	C3D		44.8	35.6	28.9	-	-	-
GTAN [43]	107.0	P3D		57.8	47.2	38.8	-	-	-
AFSD* [35]	84.4	I3D		-	-	45.9	35.0	23.4	43.5
BCNet+PGCN [68]	81.8	I3D		69.8	62.9	52.0	39.8	24.0	49.7
DaoTAD [59]	81.8	I3D		62.8	59.5	53.8	43.6	30.1	50.0
Ours*	167.6	MViT [19]		68.0	62.5	54.2	43.6	30.6	51.8
	120.9	VideoSwin [42]		69.5	64.1	54.7	42.6	27.7	51.7
	116.1	TimeSformer [4]		67.6	61.9	53.0	41.9	27.9	50.5
	111.2	STPT		**70.6**	**65.7**	**56.4**	**44.6**	30.5	**53.6**

4.2 Main Results

We compare our STPT with state-of-the-art approaches on the two datasets in Table 2 and Table 3. We also report the backbone used by each method, *e.g.*, I3D [10], TS [52], TSN [60], P3D [49], and whether the optical flows are used. For models that are end-to-end trainable, we directly report the computational cost for the whole model. For methods utilizing pre-extracted features or pre-generated action proposals, we also calculate the computational cost for the offline feature extraction stage under identical input settings for fair comparison.

On THUMOS14 dataset, our STPT, which only uses RGB frames, outperforms previous RGB models by a large margin and achieves comparable performance with methods using additional optical flows, reaching mAP 70.6%, 65.7%, 56.4% at tIoU thresholds 0.3, 0.4, 0.5, respectively. Specifically, our STPT provides 53.6%, a +10.1% average mAP boost over AFSD RGB model under identical settings. Besides, our model also outperforms the two-stream AFSD at most tIoU thresholds with less computational cost (111.2 vs. 162.2 GFLOPs).

On ActivityNet, with significant reduction of computational cost, our STPT also achieves comparable performance with the existing RGB models. Specifically, STPT still obtains slightly better mAPs than AFSD RGB model on all thresholds with less computational cost (134.1 vs. 248.7 GFLOPs). Notably,

Table 3: Action localization results on ActivityNet 1.3 (validation set), measured by mAP(%) at different tIoU thresholds, and the average mAP in [0.5 : 0.05 : 0.95].

Models	GFLOPs	Backbone	Flow	0.5	0.75	0.95	Avg.
TAL [13]	471.3	I3D	✓	38.2	18.3	1.3	20.2
A2Net [69]	471.3	I3D	✓	43.6	28.7	3.7	27.8
BSN [38]	1367.0	TS	✓	46.5	30.0	8.0	30.0
BMN [36]	1367.0	TS	✓	50.1	34.8	8.3	33.9
G-TAD [67]	1367.0	TS	✓	50.4	34.6	9.0	34.1
BMN-CSA [53]	1367.0	TS	✓	52.4	36.7	5.2	35.4
AFSD* [35]	478.0	I3D	✓	52.4	35.3	6.5	34.4
DCAN [15]	1367.0	TS	✓	51.8	36.0	9.5	35.4
R-C3D* [66]	1360.0	C3D		26.8	-	-	12.7
GTAN [43]	320.0	P3D		52.6	34.1	8.9	34.3
AFSD* [35]	248.7	I3D		50.5	33.4	6.5	32.9
	172.4	MVIT [19]		50.1	32.7	5.9	32.2
	153.7	VideoSwin [42]		49.6	32.1	5.6	31.9
	140.6	TimeSformer [4]		51.1	33.3	6.0	33.1
Ours*	134.1	STPT		51.4	33.7	6.8	33.4

most previous models use optical flows to enhance motion modeling. However, the adoption of an ensemble of flow features requires pre-extracting flow features using [71], which prevents these methods from end-to-end learning and also introduces huge computational cost. In contrast, our STPT can effectively encode the spatio-temporal representations from pure RGB frames, which is completely end-to-end trainable and computationally efficient.

It is worth to note that, compared with other representative video Transformers, our STPT achieves the best mAP on THUMOS14 and ActivityNet, which demonstrates the advantages of our architecture design that applying LSTA in shallow layers and GSTA in deeper layers in realizing the trade-off between locality and dependency. We also provide more comparison with other representative video Transformers in the supplementary.

4.3 Ablation Study

Effect of LSTA. As discussed in Sec. 1, MViT [19] suffers from heavy spatio-temporal redundancy in the shallow layers. We also investigate the computational cost of each layer and find that the heavy computation is caused by the global attention in the first two stages. To this end, we replace global attention with our efficient LSTA in the first two stages. As shown in Table 2, our STPT improves baseline MViT, which actually uses global self-attention in all the stages, by 1.4% on average mAP while reducing FLOPs by 55.4G on THUMOS14. Furthermore, we also compare our LSTA with factorized space-time attention in [4,1]. As shown in Table 2 and Table 3, our LSTA leads to higher

Table 4: Effect of our architecture design principle. We evaluate the performance of several combinations of blocks in terms of mAP. L/G refers to LSTA/GSTA blocks used in each stage. All models are equipped with CPE. The number of temporal tokens is set to 128, and the window size of temporal dimension for LSTA is set to 8.

Type	GFLOPs	0.3	0.4	0.5	0.6	0.7	Avg.
LLLL	101.8	64.7	59.4	50.8	40.0	26.5	48.3
LLLG	102.7	67.6	62.1	54.2	42.2	29.4	51.3
LGGG	151.4	69.4	63.5	55.0	42.9	29.2	52.0
GGGG	167.6	68.0	62.5	54.2	43.6	30.6	51.8
LLGG	111.4	69.1	63.7	55.2	44.2	29.3	52.3

Table 5: Effect of window size in terms of temporal dimension on THUMOS. We compare the performance (in mAP) and computational cost (in GFLOPs) for different scales of local window in each LSTA block. The number of temporal tokens is set to 128. Experiments are conducted on models without using CPE.

Window size	GFLOPs	0.3	0.4	0.5	0.6	0.7	Avg.
[1,1,1]	110.4	67.0	61.5	53.6	41.3	28.3	50.3
[4,4,4]	110.6	67.8	62.4	53.4	41.6	28.1	50.6
[8,8,8]	110.8	67.6	62.5	53.1	42.1	29.1	50.9
[8,8,16]	110.8	69.5	63.6	55.6	44.9	29.5	52.7
[16,16,16]	111.1	66.3	61.1	52.7	41.5	28.7	50.1

mAP scores than other attention designs with less computational cost, indicating the effectiveness and efficiency of the proposed module.

Effect of the architecture design. To explore the effect of our architecture design principle in STPT, we investigate all the possible combinations of LSTA (L) and GSTA (G). As shown in Table 4, when only using LSTA, the computational cost is light (LLLL). However, the mAP drops dramatically, since the network lacks the capacity of learning global dependency without GSTA. A significant improvement can be observed when replacing LSTA with GSTA in the 3^{rd} stage, which indicates the importance of learning global dependency in the deeper layers for action detection. However, when applying GSTA in all stages (GGGG), the model leads to worse results and introduces heavy computational overhead (111.2G vs 166.6G). The main reason is that, without locality constraints, the model cannot extract detailed spatio-temporal patterns in the early stages. In our experiments, we choose LSTA and GSTA in the first two stages and the last two stages respectively to achieve a preferable balance between efficiency and effectiveness.

Effect of the window size. We compare the results of different window sizes in LSTA in terms of the temporal dimension. As shown in Table 5, LSTA is beneficial from the suitable window size. Moreover, it becomes equivalent to

Table 6: The effect of CPE when varying the number of temporal tokens on the performance (in mAP at different tIoU thresholds) and computational cost (in GFLOPs) on (a) THUMOS14 and (b) ActivityNet 1.3.

Length	GFLOPs	0.3	0.4	0.5	Avg.
64	83.2	68.2	62.2	53.0	50.7
64+CPE	84.7	68.2	62.5	54.3	51.1
128	110.8	69.5	63.6	55.6	52.7
128+CPE	111.2	70.6	65.7	56.4	53.6

(a) THUMOS14

Length	GFLOPs	0.5	0.75	0.95	Avg.
192	181.5	49.8	32.3	5.3	32.0
192+CPE	182.3	50.9	33.4	6.9	33.0
96	133.6	50.9	33.2	6.1	32.8
96+CPE	134.1	51.4	33.7	6.8	33.4

(b) ActivityNet 1.3

encode spatial patterns without temporal information when using only [1,1,1] in LSTA, where the model can not capture the motion variation of local patterns in the shallow layers, leading to performance drop for action detection. However, the setting of [16,16,16] leads to 0.5% average mAP drop compared to [8,8,8], showing that too large temporal window size brings noise to informative local representations, which also demonstrates the importance of involving locality inductive bias in early stages.

Effect of CPE. We verify the effectiveness of CPE in our STPT for the action detection task under the settings with different numbers of input tokens along temporal dimension. We carefully change the temporal stride in the first PatchEmbed layer, using dilated kernels to ensure more video frames are encoded into the token sequence while keeping the length of video clips identical to the setting in previous works [35]. Despite varying the number of input tokens, as shown in Table 6, CPE provides consistent performance gain on both datasets under all settings, with trivial computational cost introduced, *e.g.*, 0.9% average mAP improvement on THUMOS14 and 1.0% average mAP improvement on ActivityNet 1.3 under the setting of 128 and 192, respectively.

5 Conclusion and Future Work

In this paper, we have proposed a novel STPT, which tackles the challenges of both computational redundancy and long-range dependency in spatio-temporal representation learning for the task of action detection. Specifically, STPT applies LSTA in shallow layers to encode local patterns with reduced spatio-temporal redundancy and employs GSTA in the later stages to handle global dependencies. Extensive experiments on THUMOS14 and ActivityNet 1.3 have demonstrated that our STPT achieves a promising balance between accuracy and efficiency for the task of action detection. Future work may include extending our STPT to other dense prediction tasks in the video recognition field, *e.g.*, video segmentation and video captioning.

Acknowledgment. This work was partially supported by the NSFC under Grant (No.61972315), Shaanxi Province International Science and Technology Cooperation Program Project-Key Projects No.2022KWZ-14.

Appendix

We organize our supplementary material as follows.

- In Sec. A, we present an illustrated example of LSTA.
- In Sec. B, we compare the per-category performance of different models on THUMOS14 and provide attention visualization from the last layer on the two selected action instances.
- In Sec. C, we present additional experiment results on ActivityNet 1.3.
- In Sec. D, we provide supplementary comparison results with other representative video Transformers.

A Additional Illustration of LSTA

We further illustrate LSTA in Fig. 3. From left to right, for each input tensor of size $T \times H \times W$, we first evenly divide it into $w_1 \times w_2 \times w_3$ sub-windows, where T, H and W refer to the temporal size and height, width, respectively. Next, each query token only attends to tokens within the same 3D local window. Last, we further reduce both the spatial and temporal resolution of the keys and values within each 3D local window for better efficiency.

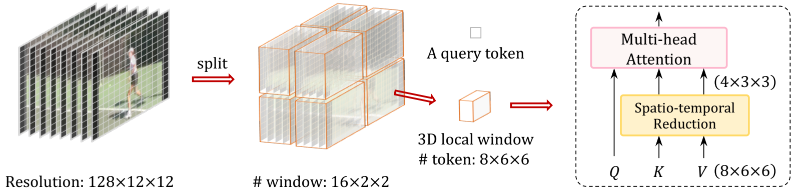

Fig. 3: An illustrated example of LSTA.

B Attention Visualization on THUMOS14

In Fig. 4, we show the per-category AP@0.5 of I3D, MViT and the proposed STPT on THUMOS14 with RGB input only. It demonstrates that our STPT surpasses the other two models in most categories. It is also notable that our STPT outperforms other models by a large margin on some action categories, *e.g.*, TennisSwing, LongJump.

We further provide attention visualization of different models on the two example action instances selected from THUMOS14 in Fig. 5 and Fig. 6. For each video instance, we show the sampled input RGB frames and use Grad-CAM to generate the corresponding attention in the last layer. Due to the local receptive field of 3D convolutions, I3D lacks the capacity of learning long-term dependencies in the videos, leading to inaccurate and irrelevant attention. Alternatively, MViT applies global attention in all the stages, which brings noises

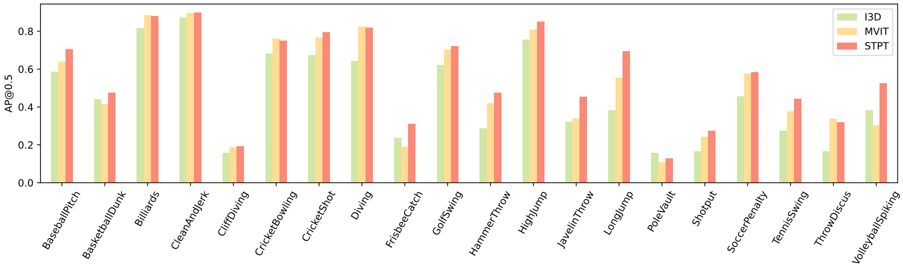

Fig. 4: Per-category AP@0.5 on THUMOS14.

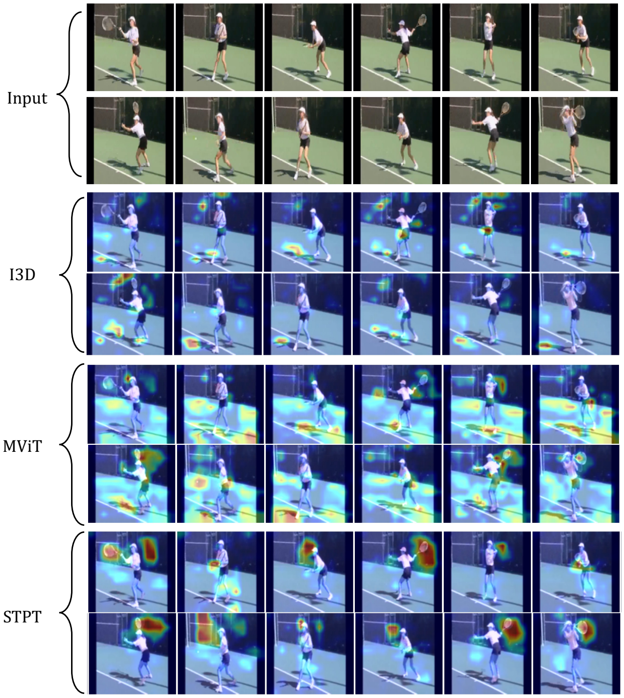

Fig. 5: Attention visualization of different models for an action instance of "TennisSwing".

to the informative spatio-temporal representations and struggles to focus on key objects or actions, *e.g.*, the tennis racket or the jumping action. Different from both cases, by flexibly involving locality constraint in early stages and applying global attention in later stages, our STPT efficiently encodes local patterns and captures global dependencies in a concise manner, enabling learning strong spatio-temporal representations from videos.

Table 7: Effect of our architecture design principle. We evaluate the performance (in mAP) and computational cost (in GFLOPs) of several combinations of blocks on ActivityNet 1.3. L/G refers to LSTA/GSTA used in each stage. Models are equipped with CPE. The number of temporal tokens is set to 96.

Type	GFLOPs	0.5	0.75	0.95	Avg.
LLLL	121.0	49.9	32.4	3.3	31.7
LLLG	126.2	50.0	32.5	3.5	31.8
LGGG	152.2	50.3	32.5	6.8	32.7
GGGG	172.4	50.1	32.7	5.9	32.2
LLGG	134.1	51.4	33.7	6.8	33.4

Table 8: Effect of window size in terms of temporal dimension. We compare the performance (in mAP) and computational cost (in GFLOPs) for different scales of a local window in each LSTA block.

Window Size	GFLOPs	0.5	0.75	0.95	Avg.
[1,1,1]	133.9	50.3	33.0	4.0	32.1
[4,4,4]	134.0	50.4	33.2	4.1	32.3
[8,8,8]	134.1	51.0	33.5	5.7	32.9
[8,8,16]	134.1	51.4	33.7	6.8	33.4
[16,16,16]	134.4	49.7	32.0	5.6	32.0

Table 9: More comparisons (mAP(%) at different tIoU thresholds) with other ViT models on THUMOS14 and ActivityNet 1.3.

Backbone	GFLOPs	0.3	0.4	0.5	0.6	0.7	Avg.
DualFormer	112.2	68.5	63.0	54.4	41.8	27.7	51.1
RegionViT	181.3	68.6	62.7	53.5	41.7	28.8	51.1
Twins	119.8	69.7	62.7	54.1	43.2	28.7	51.7
Uniformer	134.2	68.7	63.6	54.6	42.0	28.7	51.5
Ours	**111.2**	**70.6**	**65.7**	**56.4**	**44.6**	**30.5**	**53.6**

(a) THUMOS14

Backbone	GFLOPs	0.5	0.75	0.95	Avg.
DualFormer	171.1	50.7	33.1	5.4	32.7
RegionViT	241.1	51.3	33.4	6.2	32.8
Twins	140.2	51.1	33.2	5.2	32.7
Uniformer	185.5	50.7	32.8	5.2	32.6
Ours	**134.1**	**51.4**	**33.7**	**6.8**	**33.4**

(b) ActivityNet 1.3

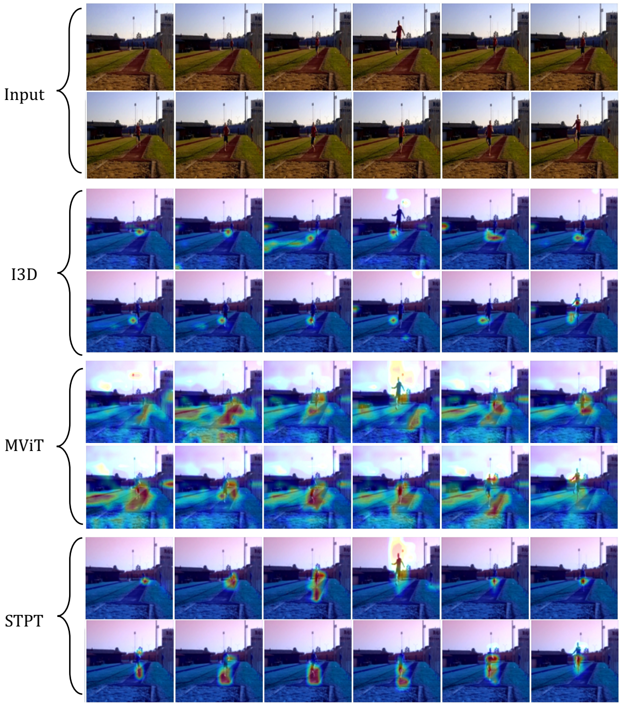

Fig. 6: Attention visualization of different models for an action instance of "LongJump".

C More Results on ActivityNet 1.3

We provide more results on ActivityNet 1.3 in Table 7 and Table 8. In Table 7, we report the performance and the computational cost of all the possible combinations of LSTA (L) and GSTA (G) on ActivityNet 1.3. Without using GSTA, the model LLLL is computationally efficient but lacks the capacity of learning global dependency, resulting in the lowest mAP scores compared with other structures. Alternatively, applying GSTA in all the stages leads to heavy computational cost (134.1G *vs*. 172.4G). However, the mAP scores drop at all thresholds as the model cannot extract detailed spatio-temporal patterns in the early stages. Thus, for ActivityNet, we choose LSTA and GSTA in the first two stages and the last two stages respectively, in order to achieve a favourable balance between efficiency and effectiveness. As shown in Table 8, with comparable FLOPs, the model with the window size of [8,8,16] for LSTA outperforms the other settings at all thresholds.

D Comparison with other video Transformers

Compared with DualFormer, RegionViT and Twins, which alternatively process local and global information within each block, we leverage local LSTA in the early stages to remove local redundancy and utilize global GSTA in the deeper layers to model the long-term dependencies. Different from the factorized space-time attention used in RegionViT, we encode the target motions by jointly aggregating spatio-temporal relations. As shown in Table 9, our STPT consistently achieves better performance with fewer FLOPs than other methods on both datasets.

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: ICCV. pp. 6836–6846 (2021)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Bai, Y., Wang, Y., Tong, Y., Yang, Y., Liu, Q., Liu, J.: Boundary content graph neural network for temporal action proposal generation. In: ECCV. pp. 121–137 (2020)
4. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML. pp. 813–824 (2021)
5. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: ICCV. pp. 5561–5569 (2017)
6. Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., Niebles, J.C.: End-to-end, single-stream temporal action detection in untrimmed videos. In: BMVC (2019)
7. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Carlos Niebles, J.: Sst: Single-stream temporal action proposals. In: CVPR. pp. 2911–2920 (2017)
8. Bulat, A., Perez Rua, J.M., Sudhakaran, S., Martinez, B., Tzimiropoulos, G.: Space-time mixing attention for video transformer. In: NeurIPS. vol. 34 (2021)
9. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: CVPR. pp. 961–970 (2015)
10. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. pp. 4724–4733 (2017)
11. Chang, S., Wang, P., Wang, F., Li, H., Feng, J.: Augmented transformer with adaptive graph for temporal action proposal generation. arXiv preprint arXiv:2103.16024 (2021)
12. Chang, X., Liu, W., Huang, P.Y., Li, C., Zhu, F., Han, M., Li, M., Ma, M., Hu, S., Kang, G., et al.: Mmvg-inf-etrol@ trecvid 2019: Activities in extended video. In: TRECVID (2019)
13. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster r-cnn architecture for temporal action localization. In: CVPR. pp. 1130–1139 (2018)
14. Chen, C.F., Panda, R., Fan, Q.: Regionvit: Regional-to-local attention for vision transformers. In: ICLR (2022)
15. Chen, G., Zheng, Y., Wang, L., Lu, T.: DCAN: improving temporal action detection via dual context aggregation. In: AAAI (2022)

16. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting the design of spatial attention in vision transformers. In: NeurIPS (2021)
17. Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., Shen, C.: Conditional positional encodings for vision transformers. arXiv preprint arXiv:2102.10882 (2021)
18. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
19. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. In: ICCV. pp. 6824–6835 (2021)
20. Fan, Q., Panda, R., et al.: An image classifier can suffice for video understanding. arXiv preprint arXiv:2106.14104 (2021)
21. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV. pp. 6202–6211 (2019)
22. Gao, J., Yang, Z., Chen, K., Sun, C., Nevatia, R.: Turn tap: Temporal unit regression network for temporal action proposals. In: ICCV. pp. 3628–3636 (2017)
23. Gong, G., Zheng, L., Mu, Y.: Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. In: ICME. pp. 1–6 (2020)
24. Han, M., Zhang, D.J., Wang, Y., Yan, R., Yao, L., Chang, X., Qiao, Y.: Dual-ai: Dual-path actor interaction learning for group activity recognition. In: CVPR (2022)
25. Han, Q., Fan, Z., Dai, Q., Sun, L., Cheng, M.M., Liu, J., Wang, J.: On the connection between local attention and dynamic depth-wise convolution. In: ICLR (2022)
26. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv: Learning (2016)
27. Hou, Q., Cheng, M.M., Hu, X., Borji, A., Tu, Z., Torr, P.H.S.: Deeply supervised salient object detection with short connections. In: CVPR (2017)
28. Islam*, M.A., Jia*, S., Bruce, N.D.B.: How much position information do convolutional neural networks encode? In: ICLR (2020)
29. Jiang, Y.G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: Action recognition with a large number of classes (2014)
30. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
31. Koot, R., Lu, H.: Videolightformer: Lightweight action recognition using transformers. arXiv preprint arXiv:2107.00451 (2021)
32. Li, J., Liu, X., Zong, Z., Zhao, W., Zhang, M., Song, J.: Graph attention based proposal 3d convnets for action detection. In: AAAI. pp. 4626–4633. No. 04 (2020)
33. Li, K., Wang, Y., Peng, G., Song, G., Liu, Y., Li, H., Qiao, Y.: Uniformer: Unified transformer for efficient spatial-temporal representation learning. In: ICLR (2022)
34. Liang, Y., Zhou, P., Zimmermann, R., Yan, S.: Dualformer: Local-global stratified transformer for efficient video recognition. arXiv preprint arXiv:2112.04674 (2021)
35. Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Learning salient boundary feature for anchor-free temporal action localization. In: CVPR. pp. 3320–3329 (2021)
36. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: Bmn: Boundary-matching network for temporal action proposal generation. In: ICCV. pp. 3889–3898 (2019)
37. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: ACM MM. pp. 988–996 (2017)

38. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. In: ECCV. pp. 3–19 (2018)
39. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2980–2988 (2017)
40. Liu, Q., Wang, Z.: Progressive boundary refinement network for temporal action detection. In: AAAI. pp. 11612–11619 (2020)
41. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)
42. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. arXiv preprint arXiv:2106.13230 (2021)
43. Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., Mei, T.: Gaussian temporal awareness networks for action localization. In: CVPR. pp. 344–353 (2019)
44. Nawhal, M., Mori, G.: Activity graph transformer for temporal action localization. arXiv preprint arXiv:2101.08540 (2021)
45. Neimark, D., Bar, O., Zohar, M., Asselmann, D.: Video transformer network. In: ICCV. pp. 3163–3172 (2021)
46. Pan, Z., Zhuang, B., He, H., Liu, J., Cai, J.: Less is more: Pay less attention in vision transformers. In: AAAI (2022)
47. Pan, Z., Zhuang, B., Liu, J., He, H., Cai, J.: Scalable vision transformers with hierarchical pooling. In: ICCV. pp. 377–386 (2021)
48. Park, N., Kim, S.: How do vision transformers work? In: ICLR (2022)
49. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: ICCV. pp. 5533–5541 (2017)
50. Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A.: Do vision transformers see like convolutional neural networks? In: NeurIPS (2021)
51. Sharir, G., Noy, A., Zelnik-Manor, L.: An image is worth 16x16 words, what is a video worth? arXiv preprint arXiv:2103.13915 (2021)
52. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. NeurIPS (2014)
53. Sridhar, D., Quader, N., Muralidharan, S., Li, Y., Dai, P., Lu, J.: Class semantics-based attention for action detection. In: ICCV. pp. 13739–13748 (2021)
54. Tan, J., Tang, J., Wang, L., Wu, G.: Relaxed transformer decoders for direct action proposal generation. In: ICCV. pp. 13526–13535 (2021)
55. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML. pp. 10347–10357 (2021)
56. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV. pp. 4489–4497 (2015)
57. Vahdani, E., Tian, Y.: Deep learning-based action detection in untrimmed videos: A survey. arXiv preprint arXiv:2110.00111 (2021)
58. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS. vol. 30 (2017)
59. Wang, C., Cai, H., Zou, Y., Xiong, Y.: Rgb stream is enough for temporal action detection. arXiv preprint arXiv:2107.04362 (2021)
60. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks for action recognition in videos. IEEE TPAMI **41**(11), 2740–2755 (2019)
61. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: ICCV. pp. 568–578 (2021)

62. Wang, Y., Chen, Z., Jiang, H., Song, S., Han, Y., Huang, G.: Adaptive focus for efficient video recognition. In: ICCV. pp. 16249–16258 (2021)
63. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. arXiv preprint arXiv:2103.15808 (2021)
64. Wu, J., Sun, P., Chen, S., Yang, J., Qi, Z., Ma, L., Luo, P.: Towards high-quality temporal action detection with sparse proposals. arXiv preprint arXiv:2109.08847 (2021)
65. Wu, Z., Su, L., Huang, Q.: Cascaded partial decoder for fast and accurate salient object detection. In: CVPR. pp. 3907–3916 (2019)
66. Xu, H., Das, A., Saenko, K.: R-c3d: Region convolutional 3d network for temporal activity detection. In: ICCV. pp. 5783–5792 (2017)
67. Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-tad: Sub-graph localization for temporal action detection. In: CVPR. pp. 10156–10165 (2020)
68. Yang, H., Wu, W., Wang, L., Jin, S., Xia, B., Yao, H., Huang, H.: Temporal action proposal generation with background constraint. In: AAAI. vol. 36, pp. 3054–3062 (2022)
69. Yang, L., Peng, H., Zhang, D., Fu, J., Han, J.: Revisiting anchor mechanisms for temporal action localization. IEEE TIP pp. 8535–8548 (2020)
70. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. In: CVPR. pp. 2678–2687 (2016)
71. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l 1 optical flow. In: Joint pattern recognition symposium. pp. 214–223 (2007)
72. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: ICCV. pp. 7094–7103 (2019)
73. Zha, X., Zhu, W., Xun, L., Yang, S., Liu, J.: Shifted chunk transformer for spatio-temporal representational learning. In: NeurIPS. vol. 34 (2021)
74. Zhang, D., Dai, X., Wang, X., Wang, Y.: S3D: single shot multi-span detector via fully 3d convolutional networks. In: BMVC. p. 293 (2018)
75. Zhang, Y., Li, X., Liu, C., Shuai, B., Zhu, Y., Brattoli, B., Chen, H., Marsic, I., Tighe, J.: Vidtr: Video transformer without convolutions. In: ICCV. pp. 13577–13587 (2021)
76. Zhao, C., Thabet, A.K., Ghanem, B.: Video self-stitching graph network for temporal action localization. In: ICCV. pp. 13658–13667 (2021)
77. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: ICCV. pp. 2914–2923 (2017)