Supplementary Material
MS-TCT: Multi-Scale Temporal ConvTransformer for Action Detection

Rui Dai[1,2], Srijan Das[3], Kumara Kahatapitiya[3], Michael S. Ryoo[3], François Brémond[1,2]
[1]Inria [2]Université Côte d'Azur [3]Stony Brook University
[1,2]{name.surname}@inria.fr

Overall: In the following sections, we provide further experimental results on MS-TCT along four aspects: (1) temporal action relation, (2) blurred videos, (3) heat-map branch (4) hyper-parameters. In addition, we provide the limitation of our method and more details on the Temporal Encoder architecture.

1. Analysis of Temporal Action Relations

Firstly, we analyse how different types of layers in the stages affect the range of temporal relations. We utilize the action-conditional metrics [3] for this analysis as it provides the dependencies between the video tokens at different temporal ranges. Similar to Section 4.3 in the main paper, we construct three types of stage based on the temporal encoder: Pure Convolution, Pure Transformer and ConvTransformer (i.e., MS-TCT). As shown in table 1, we find that Pure Convolution is better than Pure Transformer for local temporal dependencies ($\tau = 5$), but for the long-term dependencies ($\tau = 100$), the Transformer based model achieves better performance. The ConvTransformer benefits from both layers thus achieving better performance on both short and long term dependencies.

Table 1. Studies on temporal dependencies. Evaluated with action conditional mAP [3] on Charades using RGB.

Stage-Type	$\tau = 5$	$\tau = 100$
Pure Convolution	26.4	28.7
Pure Transformer	24.6	30.8
ConvTransformer	**28.9**	**33.1**

2. Performance on Blurred Videos

For the protection of personal privacy, the face of all the subjects is blurred on TSU [2]. The proposed MS-TCT outperforms the state-of-the-art methods on this dataset, showing that MS-TCT is not relying on the information of person id to conduct the action detection.

3. More studies on Heat-map Branch

We first study how the location of the heat-map branch affects the detection performance. Precisely, instead of feeding the output features from all stages (i.e., MS-TCT), here, we provide only the stage 1 or stage 4 features to the heat-map branch. In table 2, we find that having the heat-map branch either in the early stage (i.e., stage $= 1$) or at late stage (i.e., stage $= 4$) can boost the performance of MS-TCT compared to MS-TCT without the heat-map branch. The model achieves the overall best performance, while exploiting features from all the stages (i.e., F_v) to the heat-map branch.

Table 2. Location of heat-map branch on Charades dataset using only RGB. Stage indicates the features from which stage is fed to the heat-map branch. $\times$ indicates not having the heat-map branch and *All* indicates that we fed features from all the stages to the heat-map branch (i.e., similar to MS-TCT).

Stage	mAP (%)
$\times$	24.1
1	24.9
4	24.7
All	**25.4**

We then perform a qualitative analysis of the action detection performance for MS-TCT, with or without the heat-map branch. As shown in figure 2, we find that while having the heat-map branch (i.e., MS-TCT), the prediction is more continuous (e.g., *sitting in bed*, *putting a pillow*). This reflects that with the heat-map branch, the tokens in MS-TCT are embedded with the instance-center relative position. Therefore, the tokens in the instance, especially the ones close to the center region, are well detected.

4. More Studies on Hyper-parameters

In this section, we further study the hyper-parameters of MS-TCT model on the Charades dataset.

Study on the number of heads H. Multi-head attention layer divides the channels into several groups. Each group

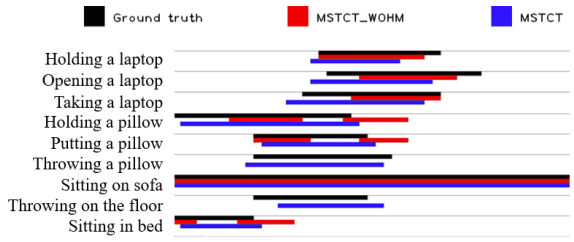

Figure 1. Qualitative study for the heat-map branch. Blue: the proposed method MS-TCT. Red: MS-TCT without the heat-map branch.

of features is sent to an attention head to model the global temporal relation. While changing the number of heads, we find that the FLOPs number remains the same, thanks to the group operations. With more heads, more complex relationships can be modelled. However, increasing the heads reduces the number of channels processed by each head. As a balance between the number of channels for each head and the number of relations to model, we set the number of heads H to 8.

Table 3. Study on number of heads in Global Relational Blocks on Charades dataset using only RGB.

# Heads H	GFLOPs	mAP (%)
1	6.6	24.6
4	6.6	25.1
8	6.6	**25.4**
16	6.6	25.3

Study on the number of Blocks B. We then study how the number of Global-Local Relational Blocks B affects the network performance. From table 4, we find that the FLOPs number is increasing with the number of blocks. The network can achieve better performance when more Global and Local layers are involved for the temporal modeling. As a balance between FLOPs and the number of blocks, we utilize a three-block architecture.

Study on the kernel size K for Temporal Convolution. After that, we study how the kernel size of the temporal convolution in the Local Relational Block affects the action detection performance. In table 5, we find that removing the temporal convolution layer in the Local Relational Block causes a significant drop in the performance. While having the convolution layer, there is not a large difference between the model with different kernel sizes. As a larger kernel size results in more weight parameters, in this work we choose the kernel size as 3.

Number of Tokens T. We randomly select consecutive T tokens for each video in the training phase and utilize the sliding window at inference. Here, we have studied how the number of tokens T affects the action detection performance. When T is set to 128, 256 and 512 tokens, MS-TCT achieves 25.0%, 25.4% and 25.5% on Charades. There is no significant difference in the action detection performance while changing the number of input tokens. However, increasing the number of tokens T in MS-TCT increases the FLOPs. For the trade-off between the computation cost and performance precision, we set T to 256 tokens, which corresponds to 2048 frames (about 86 sec.) of video.

Table 4. Study on number of Global-Local Relational Blocks for each stage on Charades dataset using only RGB.

# Block B	GFLOPs	mAP (%)
1	3.4	24.3
2	5.0	24.7
3	6.6	25.4
4	8.2	25.5

Table 5. Study on the kernel size K for the temporal convolutional layer in Local Relational Block on Charades dataset using only RGB. $\times$ indicates removing the temporal convolution layer in the Local Relational Block.

# Kernel Size K	mAP (%)
$\times$	22.3
3	25.4
5	25.1
7	25.4

Table 6. Study on the number of stage N for the temporal encoder.

N (#Stage)	1	2	3	4	5
Charades mAP	20.4	22.9	24.6	25.4	25.6

Study on the kernel size N for Temporal Encoder. Finally, we analyse the hyper-parameter N in table 6. Number of Stages (N) determines the level of semantic information in our representations. Our experiment show that a 4-stage structure strikes a balance between the performance and model size.

5. Method Limitation

Although MS-TCT has outperformed state-of-the-art methods on three challenging datasets, the performance is still relatively low (e.g., less than 30% on Charades). One of the reasons is that the Visual Encoder and the Temporal Encoder in MS-TCT are not optimized jointly in our network, due to hardware limitation. Our future work will focus on modeling the temporal and spatial relations end-to-end for long untrimmed videos.

6. Which actions benefit the most?

In order to quantify the action types which are the most benefited from MS-TCT, we present performance w.r.t. 3 action-class characteristics: # instances, intra-class variance of duration and normalized instance duration [1]. Note that, we normalize the length of the instance by the duration of

the video to have the normalized instance duration. Firstly, we find that as MS-TCT does not have a specific design for imbalanced data, this model is troubled in few-sampled action classes. Secondly, we notice that MS-TCT can perform better in the action class with high intra-class variance of duration. Finally, by the analysis of action classes with different instance duration, we find that MS-TCT can consistently detect both long and short instances.

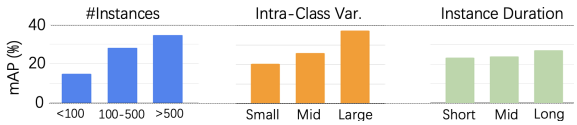

Figure 2. The sensitivity of MS-TCT's mAP to three action characteristics.

7. Temporal Encoder Architecture

To better understand the computation flow, table 7 shows the detailed architecture along with the input and output feature size of our Temporal Encoder Module. We have also allocated different hyper-parameters as H, B for different stages. However, we do not observe further improvements.

References

[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 256–272, 2018. 2

[2] Rui Dai, Srijan Das, Saurav Sharma, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota Smarthome Untrimmed: Real-World Untrimmed Videos for Activity Detection. *arXiv preprint arXiv:2010.14982*, 2020. 1

[3] Praveen Tirupattur, Kevin Duarte, Yogesh Rawat, and Mubarak Shah. Modeling multi-label action dependencies for temporal action localization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1

Table 7. Temporal Encoder architecture. The input and out feature size is following $T \times D$ format, where number of tokens (T) on the left and feature dimension D on the right. Linear layer is the kernel size 1 convolution. For the hyper-parameters: H: heads, K: kernel size, S: stride, P: zero-padding rate. Note that, for brevity, number of blocks (B) is not reflected in this table. Each Stage contains 3 Global-Local Relational Blocks, i.e., the set of a Global Relational Block and a Local Relational Block repeated 3 times for each stage.

Stage	Components	Learnable layers	Hyper-parameters	Input size	Output size
Stage 1	Temp Merge	Temp Convolution	K: 3, S: 1, P: 1	256×1024	256×256
	Global Relation	Multi Head-Attention	H: 8	256×256	256×256
	Local Relation	Linear	K: 1, S: 1, P: 0	256×256	256×2048
		Temp Convolution	K: 3, S: 1, P: 1	256×2048	256×2048
		Linear	K: 1, S: 1, P: 0	256×2048	256×256
Stage 2	Temp Merge	Temp Convolution	K: 3, S: 2, P: 1	256×256	128×384
	Global Relation	Multi Head-Attention	H: 8	128×384	128×384
	Local Relation	Linear	K: 1, S: 1, P: 0	128×384	128×3072
		Temp Convolution	K: 3, S: 1, P: 1	128×3072	128×3072
		Linear	K: 1, S: 1, P: 0	128×3072	128×384
Stage 3	Temp Merge	Temp Convolution	K: 3, S: 2, P: 1	128×384	64×576
	Global Relation	Multi Head-Attention	H: 8	64×576	64×576
	Local Relation	Linear	K: 1, S: 1, P: 0	64×576	64×4608
		Temp Convolution	K: 3, S: 1, P: 1	64×4608	64×4608
		Linear	K: 1, S: 1, P: 0	64×4608	64×576
Stage 4	Temp Merge	Temp Convolution	K: 3, S: 2, P: 1	64×576	32×864
	Global Relation	Multi Head-Attention	H: 8	32×864	32×864
	Local Relation	Linear	K: 1, S: 1, P: 0	32×864	32×6912
		Temp Convolution	K: 3, S: 1, P: 1	32×6912	32×6912
		Linear	K: 1, S: 1, P: 0	32×6912	32×864