TEMPORAL ACTION LOCALIZATION WITH GLOBAL SEGMENTATION MASK TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inspired by the promising results of Transformers in object detection in images, it is interesting to formulate Transformer based methods for temporal action localization (TAL) in videos. Nonetheless, this is non-trivial to adapt recent object detection transformers due to two unique challenges with TAL: (1) more complex spatio-temporal visual observations, and (2) less training data availability. In this paper, to address the above two challenges, a novel *Global Segmentation Mask Transformer* (GSMT) is proposed. Compared to object detection transformers, it is architecturally reformulated with the core idea to drive the transformer to learn *global segmentation masks* of all action instances jointly at the full video length. Supervised by such global temporal structure signals, GSMT allows to more effectively train from limited complex video data. Due to modeling TAL holistically rather than locally to each individual proposal, our model also differs significantly to the conventional proposal-based TAL methods that learn to detect local start and end points of action instances using more complex architectures. Extensive experiments show that despite its simpler design, GSMT outperforms existing TAL methods, achieving new state-of-the-art performance on two benchmarks. Importantly, it is around $\mathbf{100}\times$ faster to train and twice as efficient for inference.

1 INTRODUCTION

Temporal action localization (TAL) aims to identify the temporal interval (*i.e.*, the start and end points) and the class label of all action instances in an untrimmed video (Idrees et al., 2017; Caba Heilbron et al., 2015). All existing TAL methods rely on *proposal generation* by either regressing predefined anchor boxes (Xu et al., 2017; Chao et al., 2018; Gao et al., 2017; Long et al., 2019) (Fig 1(a)) or directly predicting the start and end times of proposals (Lin et al., 2019; Buch et al., 2017; Lin et al., 2018; Xu et al., 2020) (Fig 1(b)). In essence, once the proposals have been generated, existing TAL methods take a local view of the video and focus on each individual proposal for action instance temporal refinement and classification. Such an approach suffers from several fundamental limitations: (1) An excessive (sometimes exhaustive) number of proposals are usually required for good performance. For example, BMN (Lin et al., 2019) generates ~ 5000 proposals per video by exhaustively pairing start and end points predicted. Generating and evaluating such a large number of proposals means high computational cost for both training and inference. (2) Once the proposals are generated, the subsequent modeling is local to each individual proposal. Missing global context over the whole video can lead to sub-optimal localization.

Inspired by the success of Transformers (Vaswani et al., 2017) for object detection in images (Carion et al., 2020; Zhu et al., 2020), a few Transformer based methods for TAL have been introduced (Tan et al., 2021; Wang et al., 2021; Qing et al., 2021; Nawhal & Mori, 2021). Self-attention can model global content naturally. However, challenged by less labeled training data (*e.g.*, per-class 193 action instances in ActivityNet vs. 452 objects in COCO) and higher-dimensional observations, they mostly consider only the simpler proposal generation/refinement sub-task. Indeed, Nawhal & Mori (2021) is an exception that further introduces graph structures in the original set prediction framework of object detection transformers. Further, these methods are inferior in performance compared to top CNN based alternatives (Bai et al., 2020; Xu et al., 2020).

In this work, we address these challenges by proposing a *Global Segmentation Mask Transformer* (GSMT). Concretely, in representation learning we leverage self-attention (Vaswani et al., 2017) for

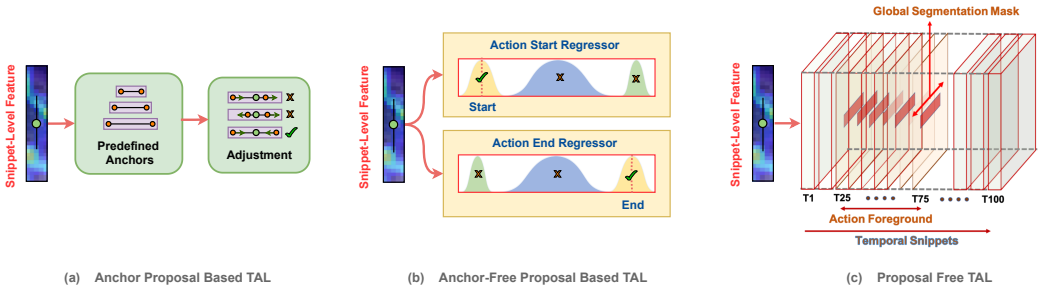


Figure 1: Conventional TAL methods are either (a) anchor-based or (b) anchor-free all needs to generate action proposals. Instead, (c) our *global segmentation mask transformer* (GSMT) model is proposal-free.

capturing necessary video-level inter-snippet relationship. Instead of predicting the start/end points of each action instance, GSMT learns to predict action segmentation masks of an entire video (Fig. 1(c)). Such masks represent the *global* temporal structure of all action instances in a video; it is thus intrinsically context-aware and more compatible with self-attentive representative learning. GSMT takes each local snippet (*i.e.*, a short sequence of consecutive frames of a video) as a predictive unit. More specifically, taking as input a snippet feature representation for a given video, GSMT then outputs the target action segmentation mask and class label concurrently. By doing so, we eliminate the need for instance query and set prediction both of which have non-trivial complexity. Hence, our GSMT is architecturally different to all the existing object detection and TAL transformers. By modeling TAL's prediction globally rather than locally, GSMT not only removes the need for proposal generation, and the associated design and computational complexity, it is also more effective. To facilitate the proposed global segmentation mask learning, we further introduce a novel boundary focused loss that pays more attention to the temporal boundary regions. During inference, once the masks and class labels are predicted, top-scoring segments can be then selected via non-maximal suppression (NMS) to produce the final TAL result.

We make the following **contributions**. **(I)** We present a novel *Global Segmentation Mask Transformer* (GSMT) method for temporal action localization. It not only eliminates the need for proposal generation, but also better tailors the self-attention of Transformers to more effectively solve the training data challenges of TAL. **(II)** This is realized by reformulating the object detection Transformers to predict global segmentation masks of action instances holistically, without a need for encoder-decoder design and set prediction. **(III)** To enhance the learning of temporal boundary, a novel boundary focused loss function is introduced. **(IV)** Extensive experiments show that the proposed GSMT method yields new state-of-the-art performance on two TAL datasets (ActivityNet-v1.3 and THUMOS'14). Importantly, our method is also significantly more efficient in both training/inference. For instance, it is 98/2.1$\times$ faster than G-TAD (Xu et al., 2020) in training and inference respectively.

2 Related Work

Although all existing TAL methods use proposals, they differ in how the proposals are generated.

Anchor-based proposal learning methods These methods generate proposal based on a predetermined set of anchors. Inspired by object detection in static images (Ren et al., 2016), R-C3D (Xu et al., 2017) proposes to use anchor boxes. It follows the structure of proposal generation and classification in design. With similar model design, TURN (Gao et al., 2017) aggregates local features to represent snippet-level features, which are then used for temporal boundary regression and classification. Later, GTAN (Long et al., 2019) improves the proposal feature pooling procedure with a learnable Gaussian kernel for weighted averaging. Recently, G-TAD (Xu et al., 2020) learns semantic and temporal context via graph convolutional networks for better proposal generation. Note that these anchor boxes are often exhaustively generated so are high in number.

Anchor-free proposal learning methods Instead of using fixed pre-designed anchor boxes, these methods directly learn to predict temporal proposals (*i.e.*, start and end times/points) (Zhao et al., 2017; Lin et al., 2018; 2019). For example, SSN (Zhao et al., 2017) decomposes an action instance into three stages (starting, course, and ending) and employs structured temporal pyramid pooling to

generate proposals. BSN (Lin et al., 2018) predicts the start, end and actionness at each temporal location and generates proposals using locations with high start and end probabilities. Later, BMN (Lin et al., 2019) additionally generates a boundary-matching confidence map to improve proposal generation. While no pre-defined anchor boxes are required, these methods often have to exhaustively pair all possible locations predicted with high scores. So both anchor-based and anchor-free TAL methods have a large quantity of temporal proposals to evaluate. This results in complex model design, high computational cost and lack of global context modeling. Our GSMT is designed to address all these limitations by being proposal-free.

Self-attention Our snippet representation is learned based on self-attention, which has been firstly introduced in Transformers for natural language processing tasks (Vaswani et al., 2017). In computer vision, non-local neural networks (Wang et al., 2018) apply the core self-attention block from transformers for context modeling and feature learning. State-of-the-art performances have been achieved in classification (Dosovitskiy et al., 2020), self-supervised learning (Chen et al., 2020), semantic segmentation (Zhang et al., 2020; Zheng et al., 2021), few-shot action recognition (Perrett et al., 2021; Zhu et al., 2021), and object tracking (Chen et al., 2021) by using such an attention model. More similar methods to our GSMT are object detection transformers (Carion et al., 2020; Yin et al., 2020; Zhu et al., 2020). Similar to this paper, several recent works (Tan et al., 2021; Wang et al., 2021; Qing et al., 2021; Nawhal & Mori, 2021) aims to leverage the transformers for TAL. They focus on either the temporal proposal generation (Tan et al., 2021) and refinement (Qing et al., 2021), or introduce elaborate query design (Nawhal & Mori, 2021). However, these methods are still less effective than top CNN alternatives. In this paper, we demonstrate for the first time the superior performance of transformer for TAL by introducing a novel global segmentation mask learning strategy, without the complexity of encoder-decoder structure and set matching.

3 METHOD

Our *Global Segmentation Mask Transformer* (GSMT) model takes as input an untrimmed video V' with a variable number of frames. The video is processed by a feature encoder (*e.g.*, a Kinetics pre-trained I3D network (Carreira & Zisserman, 2017)) into a sequence of localized snippets. To train the model, we collect a set of labeled video training set $\mathcal{D}^{train} = \{V_i, \Psi_i\}$. Each video V_i is labeled with temporal segmentation $\Psi_i = \{(\psi_j, \xi_j, y_j)\}_{j=1}^{M_i}$ where ψ_j/ξ_j denote the start/end time, y_j is the action category, and M_i is the action instance number.

Architecture As depicted in Fig. 2, the GSMT model has two key components: (1) a Transformer based snippet embedding module that learns feature representations with global temporal context (Sec. 3.1), and (2) a temporal action location head with two branches for per-snippet multi-class action classification and binary-class global segmentation mask inference, respectively (Sec. 3.2).

3.1 TRANSFORMER BASED SNIPPET EMBEDDING

Given a training video V, we first extract a feature sequence $F \in \mathbb{R}^{T \times d}$ with temporal dimension T and feature dimension d at the snippet level using a pre-trained video encoder (*e.g.*, I3D (Carreira & Zisserman, 2017)). We pre-extract the features F for both training and evaluation following the common practice (Xu et al., 2020; 2017; Lin et al., 2019). Each snippet is composed of a short sequence of (*e.g.*, 16) consecutive frames. Hence this representation contains only local spatio-temporal information while lacking global contextual information as required in TAL. To capture global context, context-aware snippet embedding is learned by the self-attention mechanism of Transformers (Vaswani et al., 2017).

Formally, we set the $Q/K/V$ of a multi-head Transformer encoder as the features $F/F/F$. The self-attentive learning between snippets is then formulated as

$$A_i = F + softmax(\frac{FW_Q(FW_K)^\top}{\sqrt{d}})(FW_V), \tag{1}$$

where $W_Q/W_K/W_V$ are learnable parameters. In a multi-head attention (MA) design, we combine a set of n_h independent heads $\{A_i\}$ to form a richer learning process. The Transformer outputs the

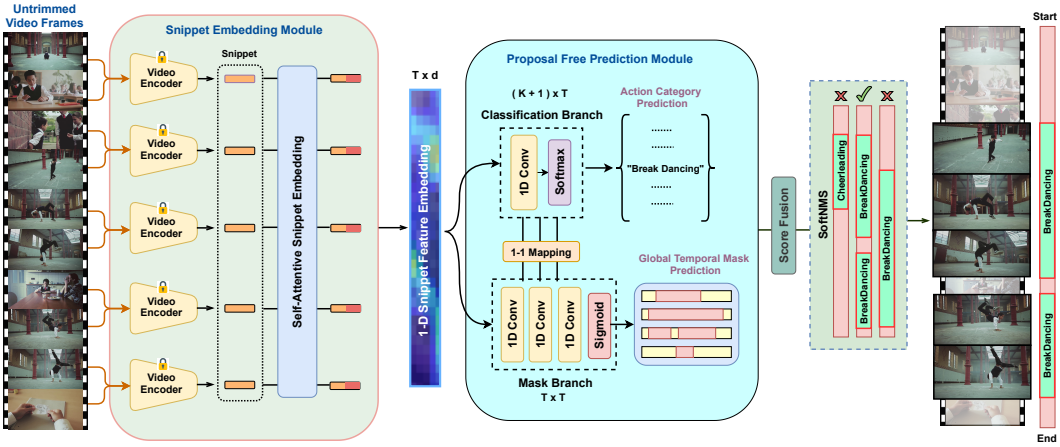


Figure 2: **Overview of our proposal-free Global Segmentation Mask Transformer (GSMT) learning architecture.** Given an untrimmed video, GSMT first extracts a sequence of T snippet features with a pre-trained video encoder (*e.g.*, I3D (Carreira & Zisserman, 2017)), and conducts self-attentive learning to obtain snippet embedding with global context. Subsequently, with each snippet embedding, GSMT classifies action class (output $\boldsymbol{P} \in \mathbb{R}^{(K+1)\times T}$ with K the action class number) and predicts full-video-long foreground mask (output $\boldsymbol{M} \in \mathbb{R}^{T\times T}$) concurrently in a two-branch design. During training, GSMT is optimized by minimizing the difference between class/mask prediction and ground-truth annotations. In inference, GSMT selects top scoring snippets from the classification output $\boldsymbol{P}$, and then thresholds the corresponding foreground masks in $\boldsymbol{M}$ to yield action instance candidates. Finally, softNMS is applied to remove redundant candidates.

snippet embedding E defined as:

$$E = MLP(\underbrace{[A_1 \cdots A_{n_h}]}_{MA}) \in \mathbb{R}^{T\times C}. \tag{2}$$

The Multi-Layer Perceptron (MLP) block has one fully-connected layer with residual skip connection. Layer norm is applied before both the MA and MLP block. We use $n_h = 4$ heads by default.

3.2 TAL HEAD: PARALLEL ACTION CLASSIFICATION AND GLOBAL SEGMENTATION MASKING

Our TAL head consists of two parallel branches: one for multi-class action classification and the other for binary-class global segmentation mask inference at the snippet level.

Multi-class action classification Given the t-th snippet $E(t) \in \mathbb{R}^c$ (*i.e.*, the t-th column of E), our classification branch predicts the probability $\boldsymbol{p}_t \in \mathbb{R}^{(K+1)\times 1}$ that it belongs to one of K target action classes or background. This is realized by a 1-D convolution layer H_c followed by a softmax normalization. Since a video has been encoded into T temporal snippets, the output of the classification branch can be expressed in column-wise as:

$$\boldsymbol{P} := softmax(H_c(E)) \in \mathbb{R}^{(K+1)\times T}. \tag{3}$$

Global segmentation mask inference In parallel to the classification branch, this branch predicts a global segmentation mask of action instances in the whole video. Formally, for each snippet $E(t)$, it outputs a mask prediction $\boldsymbol{m}_t = [q_1, \cdots, q_T] \in \mathbb{R}^{T\times 1}$ with the k-th element $q_k \in [0, 1]$ indicating the foreground probability of k-th snippet conditioned on t-th snippet. This prediction process is implemented by a stack of three 1-D convolution layers H_b as:

$$\boldsymbol{M} := sigmoid(H_b(E)) \in \mathbb{R}^{T\times T}, \tag{4}$$

where the t-th column of $\boldsymbol{M}$ is the segmentation mask prediction at the t-th snippet. With the proposed mask signal as learning supervision, our GSMT model can facilitate context-aware representation learning, which brings clear benefit on TAL accuracy (see Table 4).

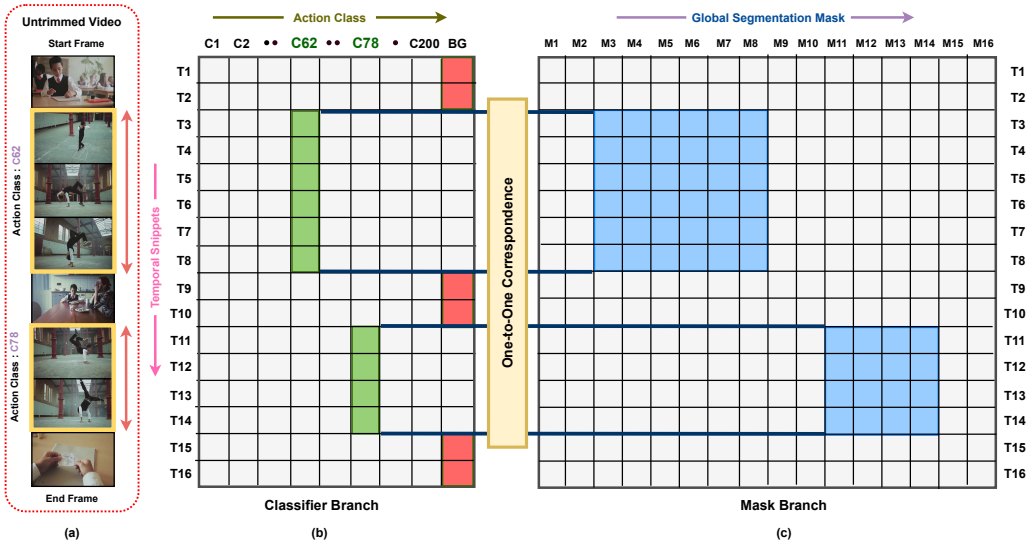


Figure 3: An illustration of label assignment (see text for more details).

It is also worthwhile noting that, by performing per-snippet class label and segmentation task inference, GSMT eliminates the set-prediction formulation of previous object detection and TAL Transformers, leading to a more compact architecture design.

3.3 Model Training and Inference

Training To train our model, the ground-truth needs to be arranged into a specific format. Concretely, given a training video with temporal intervals and action class labels (Fig 3(a)), we assign all the snippets (green squares in Fig. 3(b)) lying in action intervals as (positive) action snippets with the shared action class. All the snippets outside of action intervals are labeled as (negative) background samples (red squares in Fig. 3(b)). For an action snippet, its segmentation mask label is defined as the full binary mask of the associated action instance with the whole video length (the rows with a sequence of blue squares in Fig. 3(c)). When there are multiple action snippets involved in a specific action instance (*e.g.*, 6/4 snippets covered by the first/second action instance from class C62/C78 in Fig. 3), each will be assigned with the global segmentation mask label corresponding to this instance.

Learning objectives The classification branch is trained by a combination of cross-entropy based focal loss and a class-balanced logistic regression loss (Dong et al., 2019). For a training snippet, we denote y the ground-truth class label, $\boldsymbol{p}$ the classification output, and $\boldsymbol{r}$ the per-class regression output (discarded in inference). The loss of the classification branch is then written as:

$$L_c = \lambda_1(1-\boldsymbol{p}(y))^\gamma \log(\boldsymbol{p}_y) + (1-\lambda_1)\big(\log(\boldsymbol{r}_y) - \frac{\alpha}{|\mathcal{N}|}\sum_{k\epsilon\mathcal{N}}(\log(1-\boldsymbol{r}(k)))\big), \tag{5}$$

where $\gamma = 2$ is a focal degree parameter, $\alpha = 10$ is a class-balancing weight, and $\mathcal{N}$ specifies a set of hard negative classes at size of $K/10$ where K is the total action class number. We set the loss trade-off parameter $\lambda_1 = 0.4$.

For training the segmentation mask branch, we combine a novel boundary IOU (bIOU) loss and a dice loss (Milletari et al., 2016) to model two types of structured consistency respectively: mask boundary consistency and inter-mask consistency. Inspired by the boundary IOU metric (Cheng et al., 2021), bIOU is designed particularly to penalize incorrect temporal boundary prediction w.r.t. the ground-truth segmentation mask. Formally, for a snippet location, we denote $\boldsymbol{m} \in \mathbb{R}^{T\times 1}$ the predicted segmentation mask, and $\boldsymbol{g} \in \mathbb{R}^{T\times 1}$ the ground-truth mask. The overall segmentation mask loss is formulated as:

$$L_m = 1 - \Big(\frac{\Phi(\boldsymbol{m})\cap\Phi(\boldsymbol{g})}{\Phi(\boldsymbol{m})\cup\Phi(\boldsymbol{g})}\Big) + \frac{1}{\Phi(\boldsymbol{m})\cap\Phi(\boldsymbol{g})+\epsilon}\frac{\|\boldsymbol{m}-\boldsymbol{g}\|_2}{c} + \lambda_2\Big(1 - \frac{\boldsymbol{m}^\top\boldsymbol{g}}{\sum_{t=1}^{T}(\boldsymbol{m}(t)^2+\boldsymbol{g}(t)^2)}\Big), \tag{6}$$

where $\Phi(\cdot)$ represents a kernel of size k (7 in our default setting, see Tab 14 in Appendix A.1) used as a differentiable morphological erosion operation (Riba et al., 2020) on a mask and c specifies the

ground-truth mask length. In case of no boundary overlap between the predicted and ground-truth masks, we use the normalized L_2 loss. The constant $\epsilon = e^{-8}$ is introduced for numerical stability. We set the loss trade-off coefficient $\lambda_2 = 0.4$.

The overall objective loss function for training GSMT is defined as: $L = L_c + L_m$.

Inference At test time, we generate the action instance predictions for each test video based on the classification $\boldsymbol{P}$ and segmentation mask $\boldsymbol{M}$ predictions jointly. From $\boldsymbol{P}$ we select the top M_1 scoring snippets $S = \{S_{t_i}\}_{i=1}^{M_1}$ as action snippets, where $t_i \in [1, \cdots, T]$. For each S_{t_i}, we then obtain the segmentation mask predictions by thresholding the corresponding t_i-th column of $\boldsymbol{M}$. We have two different ways to obtain the segmentation masks depending on the characteristics of action definition. For long actions (*e.g.*, those in ActivityNet), after applying a threshold θ_i, we take the first activated snippet as the start and the last activated as the end, *i.e.*, yielding one detection per snippet per threshold. For short actions (*e.g.*, THUMOS), we output each activated subsequence of snippets as a candidate and there might be multiple candidates per snippet. To generate sufficient candidates, we apply a set of threshold values $\Theta = \{\theta_i\}$ to yield action candidates with varying lengths and confidences and combine all the outputs. For each candidate, we compute its confidence score s by multiplying the classification score from $\boldsymbol{P}$ and the maximal segmentation mask score from $\boldsymbol{M}$. Thereafter, we further filter out candidates with low confidence by a threshold θ_c, select top M_2 scoring ones, and apply SoftNMS (Bodla et al., 2017) to obtain the final predictions. In summary, once the global segmentation mask is predicted, the rest of inference is very similar to most recent TAL methods.

4 Experiments

Datasets We conduct extensive experiments on two popular TAL benchmarks. (1) *ActivityNet-v1.3* (Caba Heilbron et al., 2015) has 19,994 videos from 200 action classes. We follow the standard setting to split all videos into training, validation and testing subsets in ratio of 2:1:1. (2) *THUMOS14* (Idrees et al., 2017) has 200 validation videos and 213 testing videos from 20 categories with labeled temporal boundary and action class.

Implementation details We use two pre-extracted encoders for feature extraction, for fair comparisons with previous methods. One is a pre-trained two-stream model (Simonyan & Zisserman, 2014), with downsampling ratio 16 and stride 2. Each video feature sequence F is rescaled to $T = 100/256$ snippets for AcitivtyNet/THUMOS using linear interpolation. The other is Kinetics pre-trained I3D model (Carreira & Zisserman, 2017) with a downsampling ratio of 5. Our model is trained for 15 epochs using SGD with learning rate of $10^{-4}/10^{-5}$, weight decay of $10^{-3}/10^{-5}$ for AcitivtyNet/THUMOS respectively. The batch size is set to 200 for ActivityNet and 50 for THUMOS. During testing, we set the threshold set for segmentation mask $\Theta = \{0.1 \sim 0.9\}$ with step 0.1, and the confidence threshold $\theta_c = 0.1$. We set $M_1 = 50$ (top action snippet selection) and $M_2 = 200$ (final prediction selection). For post-processing, the SoftNMS threshold is set as 0.4/0.6 for THUMOS/ActivityNet.

4.1 Main results

Results on ActivityNet From Table 1, we can make the following observations: (1) GSMT with I3D feature achieves the best result in average mAP. Despite the fact that our model is much simpler in both architecture and loss designs compared to the existing alternatives. This validates our assumption that with proper global context modeling, explicit proposal generation is not only redundant but also less effective. (2) When using the relatively weaker two-stream (TS) features, our model remains highly strong and even surpasses I3D based BU-TAL (Zhao et al., 2020) and A2Net (Yang et al., 2020) by a large margin. (3) Compared to RTD-Net and AGTr both of which adopt the architecture of object detection Transformers, our GSMT is significantly superior in performance particularly on ActivityNet. This validates the advantage of our model formulation in exploiting the Transformer for the TAL task with smaller but more complex video training data when compared to object detection in images.

Results on THUMOS14 Similar conclusions can be drawn in general on THUMOS from Table 1. There is only one noticeable difference: We find that I3D is now much more effective than two-steam (TS), *e.g.*, around 7.8% gain in average mAP with GSMT over TS, compared with 1.2%

on ActivityNet. This is mostly likely caused by the distinctive characteristics of the two datasets in terms of action instance duration and video length.

Table 1: Performance comparison with state-of-the-art methods on THUMOS14 and ActivityNet-v1.3. The results are measured by mAP at different IoU thresholds, and average mAP in [0.3 : 0.1 : 0.7] on THUMOS14 and [0.5 : 0.05 : 0.95] on ActivityNet-v1.3. Prop. free = Proposal free. *: Using the object detection Transformer architecture.

Type	Model	Backbone	THUMOS14						ActivityNet-v1.3			
			0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
Anchor	R-C3D (Xu et al., 2017)	C3D	44.8	35.6	28.9	-	-	-	26.8	-	-	-
	TAL (Chao et al., 2018)	I3D	53.2	48.5	42.8	33.8	20.8	39.8	38.2	18.3	1.3	20.2
	GTAN (Long et al., 2019)	P3D	57.8	47.2	38.8	-	-	-	52.6	34.1	8.9	34.3
Actionness	BMN (Lin et al., 2019)	TS	56.0	47.4	38.8	29.7	20.5	38.5	50.1	34.8	8.3	33.9
	DBG (Lin et al., 2020)	TS	57.8	49.4	42.8	33.8	21.7	41.1	-	-	-	-
	G-TAD (Xu et al., 2020)	TS	54.5	47.6	40.2	30.8	23.4	39.3	50.4	34.6	9.0	34.1
	BU-TAL (Zhao et al., 2020)	I3D	53.9	50.7	45.4	38.0	28.5	43.3	43.5	33.9	9.2	30.1
	BC-GNN (Bai et al., 2020)	TS	57.1	49.1	40.4	31.2	23.1	40.2	50.6	34.8	**9.4**	34.3
	TCANet (Qing et al., 2021)	TS	60.6	**53.2**	44.6	36.8	**26.7**	-	52.2	**36.7**	6.8	35.5
	RTD-Net* (Tan et al., 2021)	I3D	68.3	62.3	51.9	38.8	23.7	-	47.2	30.7	8.6	30.8
Mixed	A2Net (Yang et al., 2020)	I3D	58.6	54.1	45.5	32.5	17.2	41.6	43.6	28.7	3.7	27.8
	GTAD+PGCN (Zeng et al., 2019)	I3D	66.4	60.4	51.6	37.6	22.9	47.8	-	-	-	-
Prop. free	AGTr* (Nawhal & Mori, 2021)	I3D	65.0	58.1	50.2	-	-	-	-	-	-	-
	GSMT	I3D	**68.5**	**62.6**	**54.3**	**43.8**	**28.7**	**51.6**	**56.0**	**36.8**	**9.4**	**36.2**
	GSMT	TS	**60.9**	52.1	**45.0**	**36.9**	24.0	**43.8**	**53.2**	34.9	9.1	**35.6**

Table 2: Model training and test cost.

Model	Epoch	Train	Test
BMN	13	9.45 hr	0.21 sec
G-TAD	11	3.91 hr	0.19 sec
GSMT	**9**	**0.04** hr	**0.09** sec

Table 3: Model parameter # and FLOPs.

Model	Params (in M)	FLOPs (in G)
BMN	5.0	91.2
GTAD	9.5	97.2
GSMT	**2.9**	**0.6**

Computational cost comparison One of the key motivations to design a proposal-free TAL model is to reduce the model training and inference cost. For comparative evaluation, we evaluate GSMT against two representative and recent TAL methods (BMN (Lin et al., 2019) and G-TAD (Xu et al., 2020)) using their released codes. All the methods are tested on the same machine with one Nvidia 2080 Ti GPU. We measure the convergence time in training and average inference time per video in testing. The two-stream video features are used. It can be seen in Table 2 that our GSMT is drastically faster, *e.g.*, $98/236\times$ for training and $2.1/2.3\times$ for testing in comparison to G-TAD/BMN, respectively. Besides, GSMT needs less epochs to converge. Table 3 also shows that our GSMT has the smallest FLOPs and the least parameter number.

4.2 Ablation Study and Further Analysis

Transformer vs. CNN We compare the Transformer with CNN for snippet feature learning. To this end, we consider two CNN designs: (1) a 1D CNN with 3 dilation rates (1, 3, 5) each with 2 layers, and (2) a multi-scale Temporal Convolutional Network MS-TCN (Farha & Gall, 2019). Each CNN design substitutes the Transformer respectively while remaining all the others. The results in Table 4 shows that the Transformer is clearly superior to both 1D-CNN and a relatively stronger MS-TCN. This suggests that our global segmentation mask learning is more compatible with self-attention models due to stronger

Table 4: Transformer vs. CNN on ActivityNet.

Network	mAP	
	0.5	**Avg**
1D CNN	46.8	26.4
MS-TCN	53.1	33.8
Transformer	**56.0**	**36.2**

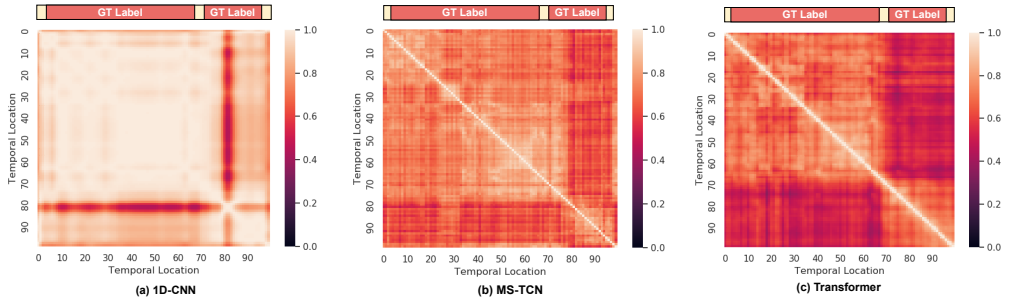


Figure 4: Inter-snippet cosine similarity in the embedding space for a random ActivityNet val video.

contextual learning capability. For qualitative analysis, we examine the cosine similarity scores of all snippet feature pairs on a random ActivityNet val video. As shown in Fig. 4, the Transformer can better identify the separation between foreground and background in the feature representation space.

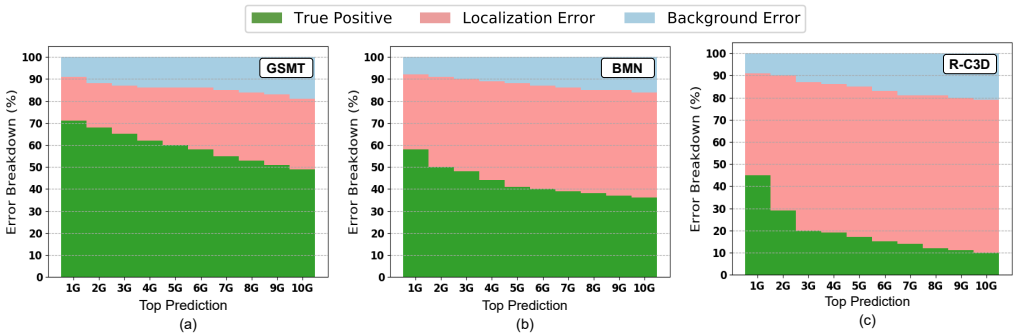


Figure 5: False positive profile of GSMT, BMN and R-C3D on ActivityNet. We use top up-to $10G$ predictions per video, where G is the number of ground truth action instances.

Proposal-based vs. proposal-free We compare our proposal-free GSMT with conventional proposal-based TAL methods BMN (Bai et al., 2020) and R-C3D (Xu et al., 2017) via false positive analysis (Alwassel et al., 2018). We sort the predictions by the scores and take the top-scoring predictions per video. Two major errors of TAL are considered: (1) *Localization error*, which is defined as when a proposal/mask is predicted as foreground, has a minimum tIoU of 0.1 but does not meet the tIoU threshold. (2) *Background error*, which is the case when a proposal/mask is predicted as foreground but its tIoU with ground truth instance is smaller than 0.1. In this test, we use ActivitityNet. We observe in Fig. 5 (a,b,c) that our GSMT identifies more precisely positive samples than BMN and R-C3D in all the cases.

Table 5: Effect of different GSMT loss objectives on ActivityNet.

Loss	mAP	
	0.5	**Avg**
GSMT (full)	56.0	36.2
w/o Focal Loss	53.8	34.2
w/o Balanced LR Loss	54.4	35.2
w/o Dice Loss	54.1	34.9
w/o bIOU Loss	53.2	34.3

Effect of various loss objectives The results in Table 5 show that each loss is beneficial for TAL's accuracy. In particular, focal loss and balanced logistic regression (LR) loss in Eq (5) and binary dice loss in Eq (6) all can tackle the imbalance problem between action and background classes, whilst our proposed boundary IOU (bIOU) loss in Eq. (6) is helpful in sharpening the foreground mask prediction. More specifically, bIOU contributes 2.8% in mAP@0.5 and 1.9% in Avg mAP, indicating the importance of temporal boundary and the effectiveness of our loss design in regulating more capacity for boundary inference. To visualize the effect of bIOU loss, we randomly select a video from ActivityNet validation set and compare the mask predictions with and without bIOU loss. Fig. 6 shows that with bIOU loss the model can more accurately capture the temporal boundary of action instance, avoiding the otherwise over-split mistakes.

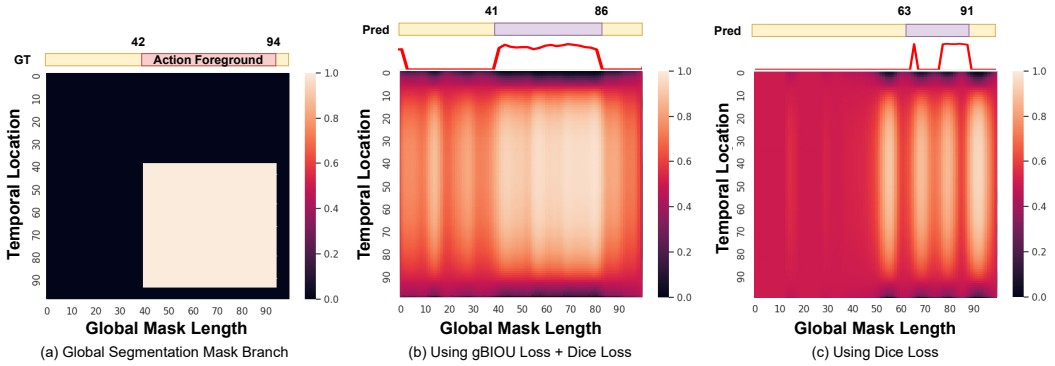


Figure 6: The effect of our bIOU loss an a random ActivitNet val video.

Table 6: Improvement analysis on ActivityNet. GT: Ground-Truth.

Model	mAP	
	0.5	Avg
GSMT	56.0	36.2
+ GT class	69.2	43.2
+ GT mask	61.0	47.0

Table 7: Analysis of network components on ActivityNet.

Model	mAP	
	0.5	Avg
GSMT(Full)	56.0	36.2
w/o Mask Branch	45.8	28.9
w/o Class Branch + UNet	49.7	31.8

Direction of improvement analysis Two subtasks are involved in TAL – temporal localization and action classification, each of which would affect the final performance. Given the two-branch design in GSMT, the performance effect of one subtask can be individually examined by simply assigning ground-truth to the other subtask's output at test time. From Table 6, the following observations can be made: (1) There is still a big scope for improvement on both subtasks. (2) Regarding the benefit from the improvement from the other subtask, the classification subtask seems to have the most to gain at mAP@0.5, whilst the localization task can benefit more on the average mAP metric. Overall, this analysis suggests that further improving the efficacy on the classification subtask would be more influential to the final model performance.

Analysis of components We can see in Table 7 that without the proposed segmentation mask branch, the model will degrade significantly, *e.g.*, a drop of 7.3% in average mAP. This is due to its fundamental capability of modeling the global temporal structure of action instances and hence yielding better action temporal intervals. Further, for GSMT we use a pre-trained UntrimmedNet (UNet) (Wang et al., 2017) as an external classifier instead of using the classification branch, resulted in a 2-stage method. This causes a performance drop of 4.4%, suggesting the advantage of our 1-stage design in both accuracy and efficiency.

5 Conclusion

In this work, we have presented a novel *Global Segmentation Mask Transformer* (GSMT) method for temporal action localization. Compared to the encoder-decoder architecture of object detection Transformers, we resort to a simpler encoder-only design, without the complexity of set prediction. To address the smaller training data challenge, we introduce the global segmentation mask learning idea for more effective self-attention representation learning in the Transformers. When compared to previous proposal based alternatives, our GSMT is significantly simpler in design with more efficient training and inference. Extensive experiments validated that the proposed GSMT yields new state-of-the-art performances on two TAL benchmarks, and with clear efficiency advantages on both model training and inference. Further, we demonstrate the superiority of our model over conventional proposal-based methods under the more realistic and more challenging cross-domain setting.

REFERENCES

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

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*, pp. 121–137. Springer, 2020.

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, pp. 5561–5569, 2017.

Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *CVPR*, 2017.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pp. 961–970, 2015.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

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

Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, 2018.

Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021.

Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *CVPR*, pp. 15334–15342, 2021.

Qi Dong, Xiatian Zhu, and Shaogang Gong. Single-label multi-class image classification by deep logistic regression. In *AAAI*, volume 33, pp. 3486–3493, 2019.

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

Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *CVPR*, pp. 3575–3584, 2019.

Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, 2017.

Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

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*, volume 34, pp. 11499–11506, 2020.

Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018.

Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: boundary-matching network for temporal action proposal generation. In *ICCV*, 2019.

Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *CVPR*, 2019.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pp. 565–571. IEEE, 2016.

Megha Nawhal and Greg Mori. Activity graph transformer for temporal action localization. *arXiv preprint arXiv:2101.08540*, 2021.

Toby Perrett, Alessandro Masullo, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. Temporal-relational crosstransformers for few-shot action recognition. In *CVPR*, 2021.

Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. In *CVPR*, pp. 485–494, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, 2016.

Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *WACV*, pp. 3674–3683, 2020.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014.

Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *ICCV*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, pp. 4325–4334, 2017.

Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. Oadtr: Online action detection with transformers. In *ICCV*, 2021.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017.

Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020.

Le Yang, Houwen Peng, Dingwen Zhang, Jianlong Fu, and Junwei Han. Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing*, 29:8535–8548, 2020.

Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *ECCV*, 2020.

Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, 2019.

Li Zhang, Dan Xu, Anurag Arnab, and Philip HS Torr. Dynamic graph message passing networks. In *CVPR*, 2020.

Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *ECCV*, pp. 539–555. Springer, 2020.

Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.

Xiatian Zhu, Antoine Toisoul, Juan-Manuel Perez-Rua, Li Zhang, Brais Martinez, and Tao Xiang. Few-shot action recognition with prototype-centered attentive learning. *arXiv preprint arXiv:2101.08085*, 2021.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint*, 2020.

A Appendix

A.1 More Ablation Studies

Failure cases and limitations of our approach A failure case is shown in Fig. 7. The segments marked in red are the wrong detections. In this example, the duration of background instance between two diving instances (21 vs 22, 24 vs 25) is very small. In such a situation, GSMT may wrongly consider the background in between as foreground. Since snippet is the smallest prediction unit, any foreground/background segments with a duration close to the snippet length will challenge any TAL methods that use snippets as input. Improving the sensitivity of our model on detecting short-duration action instances is thus part of the future work.

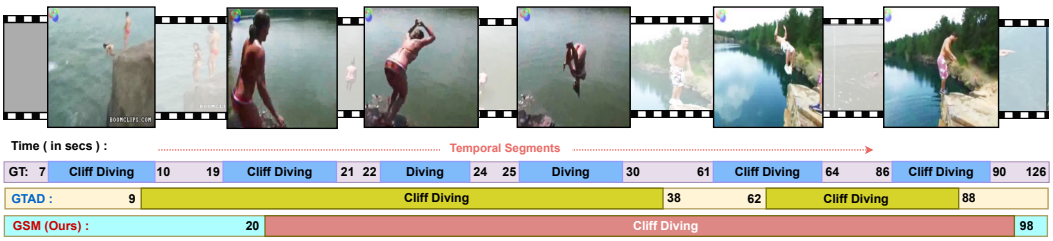


Figure 7: A failure case from THUMOS14

Ablation of component design in GSMT Our GSMT primarily consists of a Snippet Embedding Transformer and 1-D Convolution heads for classification and localization branch. We ablate the number of 1-D CNN layers for both the branch heads in Table 8. As the results suggest, only 1 layer is enough for classification branch. A plausible reason for this is that for classification it needs global information and stacking multiple 1-D CNN may affect the global information. For localization branch, it is observed that 3 layers give best performance. This is probably because for predicting the masks the network needs to process local information captured by 1-D CNNs. Additionally, we also ablate the performance of transformer design in head size. Table 9 demonstrates that the performance of GSMT improves significantly with the increase of heads in the Transformer. However, excessive heads will lead to overfitting. The performance peaks at four heads.

Cross-domain generalization The experiments so far assumed that the training and test data come from the same dataset/domain. However, in real-world applications a trained model typically needs to handle many different deployment situations out of the box. To simulate this more realistic deployment setting, we design a cross-domain experiment using a subset of classes shared by ActivityNet-v1.3 and THUMOS14. We manually match the class semantics across the two datasets and then merge those classes with same semantics but different names. This results in a total of 12 classes. G-TAD (Xu et al., 2020) is selected for comparative evaluation. We then train each model on one dataset and test on the other. We observe from Table 10 that: (1) Both models' performance degrades (vs. Table 1) under this more challenging setting due to the data distribution shift. (2) Importantly, our model's advantage over G-TAD is even bigger compared to the same-domain setting, suggesting that our model is more suited to real-world deployments. This is not surprising as simpler models often generalize better.

Table 8: Analysis of number of 1-D CNN Layers for classification and mask branches on ActivityNet.

# Layers	Class. brch		Mask brch	
	0.5	Avg	0.5	Avg
1	**56.0**	**36.2**	52.7	34.2
2	55.8	36.0	53.8	35.1
3	55.2	35.9	**56.0**	**36.2**
4	54.3	35.1	56.0	36.1
5	53.8	34.7	55.9	36.0

Table 9: Impact of the head number in the Transformer on ActivityNet.

Number of heads	mAP	
	0.5	Avg
1	53.8	34.8
2	54.6	35.0
3	55.2	35.7
4	**56.0**	**36.2**
5	55.8	35.9

Table 10: Cross-domain generalization.

Methods	**ActivityNet → Thumos**		**Thumos → ActivityNet**	
	mAP@0.5	**Avg mAP**	**mAP@0.5**	**Avg mAP**
GTAD (Xu et al., 2020)	27.5	28.2	34.5	22.1
GSMT	**32.7**	**30.3**	**43.4**	**25.6**

Transformer for existing TAL methods We examine how well existing TAL methods work with GSTM's transformer for snippet embedding. We select a representative model BMN (Lin et al., 2019) and insert our snippet embedding module right after the video encoder. As shown in Table 11, self-attention can also improve the performance of BMN, demonstrating the importance of temporal relationship modeling for temporal action localization task. However, it is still significantly inferior to our GSMT model.

Table 11: Transformer for existing TAL methods on ActivityNet.

Network	**mAP**	
	0.5	**Avg**
BMN (Lin et al., 2019)	50.1	33.9
Transformer + BMN	**51.6**	**34.8**

Performance vs. video length We additionally analyze the fine-grained performance by video length. Following (Alwassel et al., 2018), the videos of THUMOS dataset are classified into 5 different categories by the temporal duration: extra-small, small, medium, long and extra-long. We compare our GSMT with BMN (Bai et al., 2020) and R-C3D (Xu et al., 2017) on each of these 5 duration categories individually. As seen in Fig. 8, our proposal-free GSMT performs better for short, medium and extra long action instances, whilst the proposal-based methods are better in extra-small and long cases. This suggests some potential space of further improving our GSMT via taking inspirations from previous proposal methods, which will be part of our future work.

Snippet length We evaluate the impact of the video snippet length for GSMT on ActivityNet. As shown in Table 12, when the snippet length is small (*e.g.*, 50), we observe a performance drop of 4% in mAP@0.5. This may be due to that too small snippets are less capable to represent local motion patterns. We found that the length of 100 is the best, confirming the same findings as GTAD (Xu et al., 2020) and BMN (Lin et al., 2019).

Inference speed For comparison with more previous methods with no training code released, we can only compare with their reported inference speed measured in FPS. As different GPU hardware is used in previous papers, for easier comparison we translate the FPS speed according to their specification. For this comparison, I3D features on THUMOS14 are used. It is evident from Fig. 9 that our GSMT runs much faster, *e.g.*, $4/5\times$ faster than PGCN/SSTAD.

Role of positional encoding in GSMT To evaluate the effect of position encoding in GSMT on ActivityNet. As shown in Table 13, it is interesting to see that position encoding is not necessary and

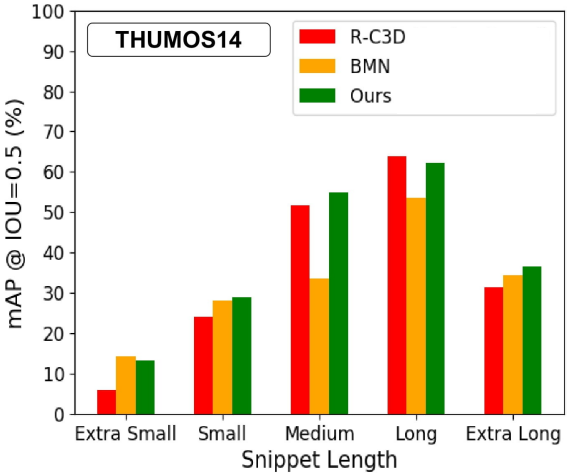


Figure 8: Fine-grained performance on video subsets with different temporal lengths on THUMOS.

Table 12: Impact of snippet length on ActivityNet

Snippet Length	mAP	
	0.5	Avg
50	52.2	33.5
100	**56.0**	**36.2**
150	55.7	36.0
200	55.1	35.8

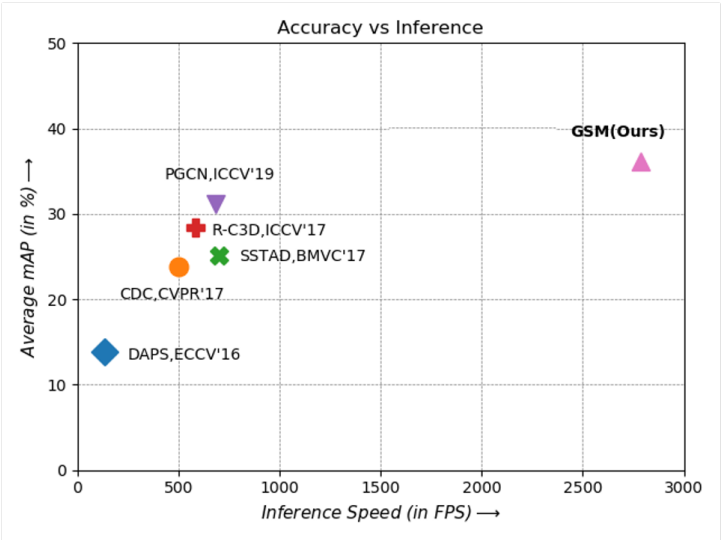


Figure 9: Translated FPS based on Titan XM.

Table 13: Effect of positional encoding in GSMT on ActivityNet.

# Position Encoding	mAP	
	0.5	Avg
No Encoding	**56.0**	**36.2**
Learned Encoding	53.9	33.4
Fixed Encoding	44.7	28.0

Table 14: Impact of kernel size on bIOU loss on ActivityNet.

Kernel Size	mAP	
	0.5	Avg
3	53.3	34.3
5	55.1	35.7
7	**56.0**	**36.2**
9	55.7	36.0

even harmful to the performance. This indicates that with our current formulation, the snippet level temporal information does not bring extra useful information.

A.2 Qualitative results

In this section, to make more visual examination we provide additional qualitative results by GTAD (Xu et al., 2020) and our GSMT model on both ActivityNetv1.3 and THUMOS14 dataset. We focus on two challenging situations: (i) a single short-duration action instance per video Fig 11 and (ii) multiple short-duration action instances per video Fig 12 . From these examples, we have a similar observation that compared to G-TAD, our proposed GSM method can localize the target action instances more accurately with a much smaller number of outputs, thus6being more efficient at inference.

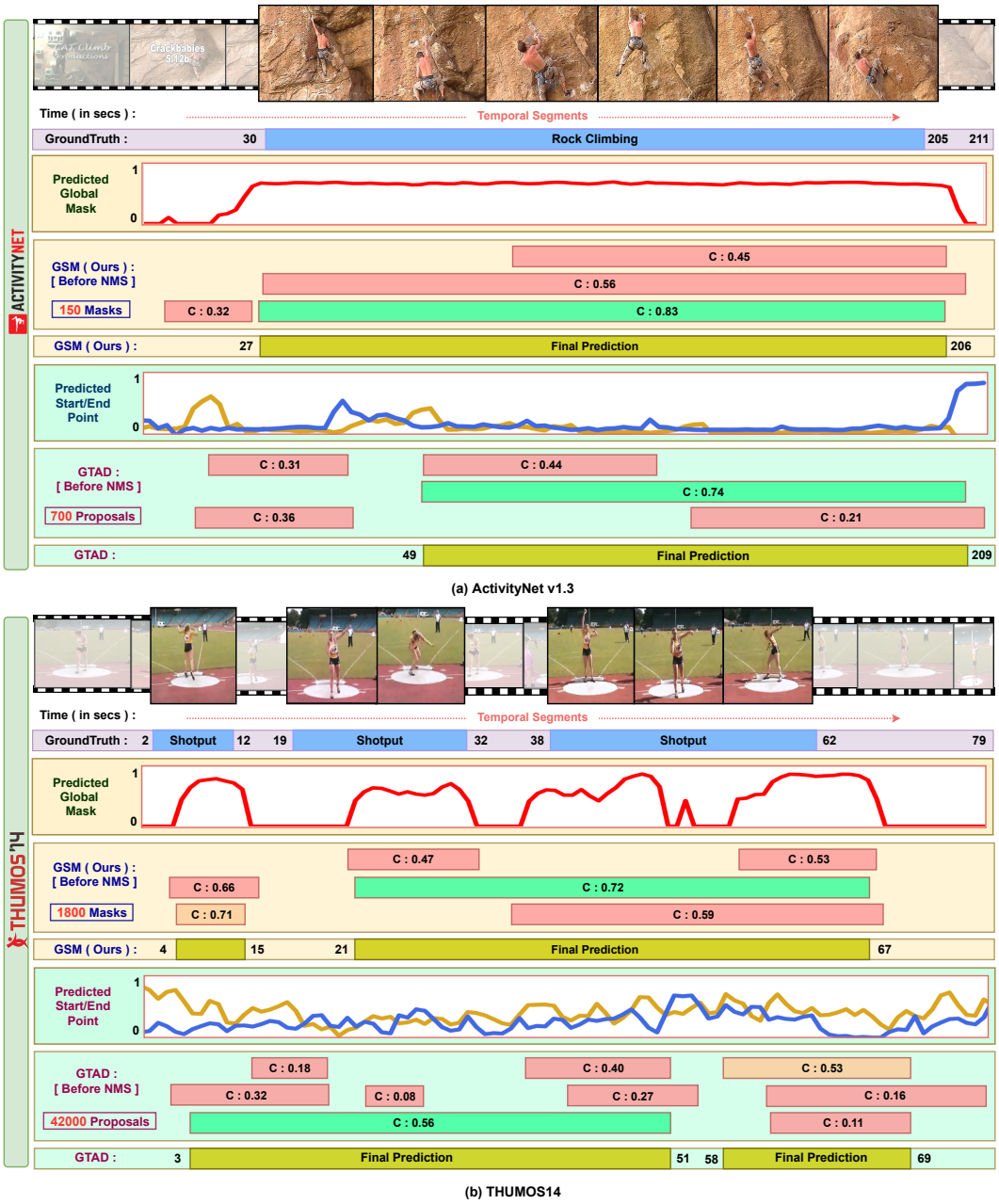
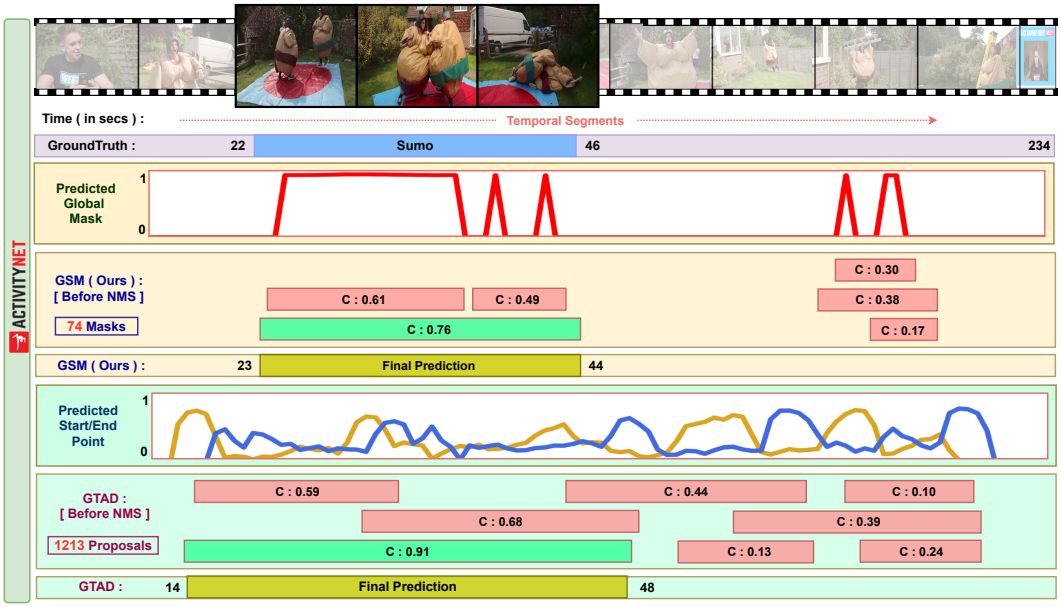
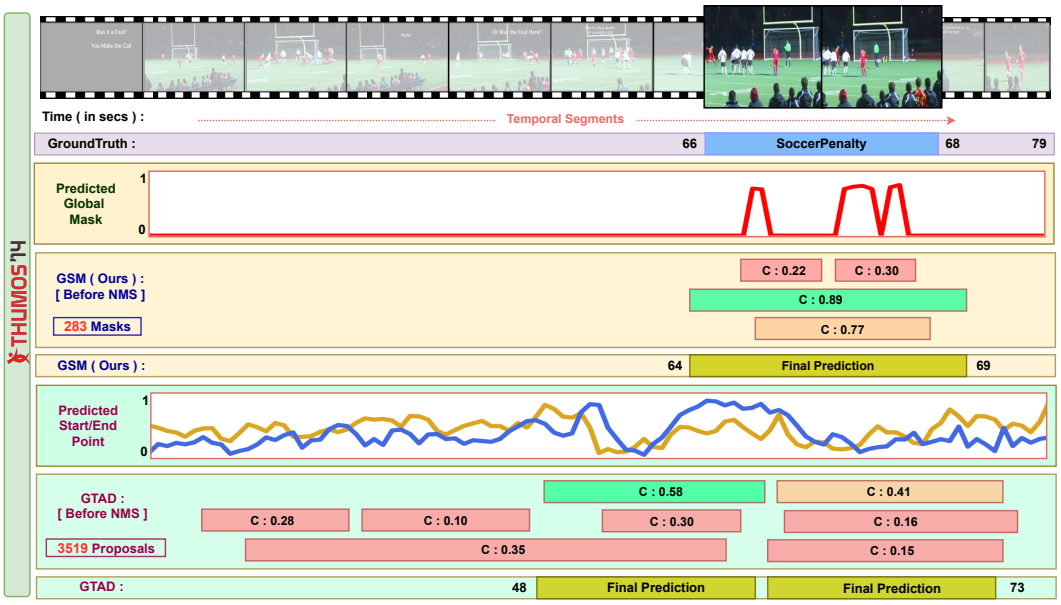


Figure 10: **Qualitative TAL result comparison** on videos from (a) ActivityNet-v1.3 and (b) Thumos14. We compare our GSMT (first 3 rows) with G-TAD Xu et al. (2020) (last 3 rows). For each method, we show a number of top action detection candidates, with the confidence score given inside each detection box. It can be seen that for both cases, our GSMT produces more accurate action instance detection with much less candidates compared to G-TAD.



(a) ActivityNet v1.3



(b) THUMOS14

Figure 11: **Qualitative TAL result comparison** on single-instance videos from (a) ActivityNet-v1.3 and (b) Thumos14.

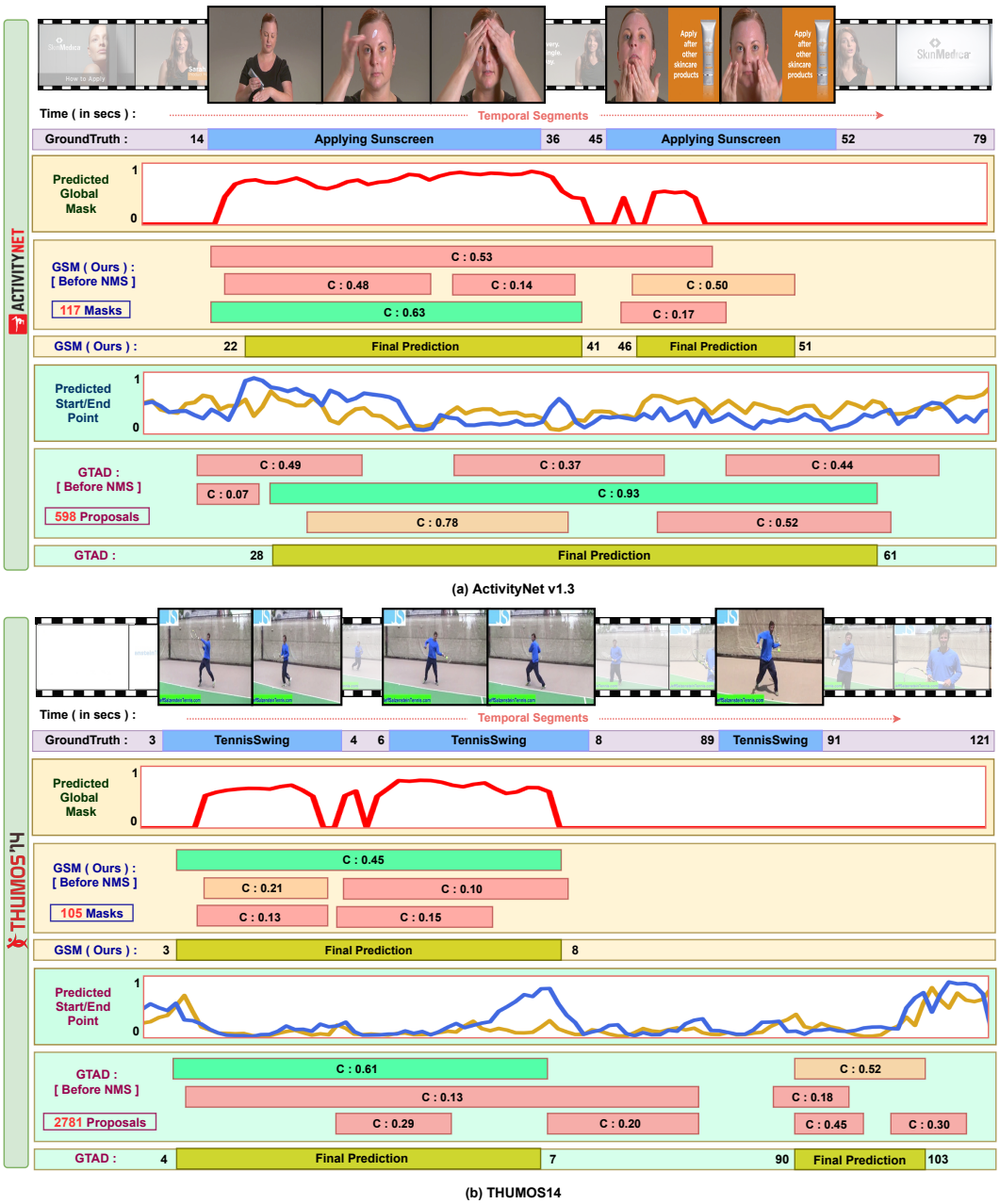


Figure 12: **Qualitative TAL result comparison** on multi-instance videos from (a) ActivityNet-v1.3 and (b) Thumos14.