

# Self-Feedback DETR for Temporal Action Detection

Jihwan Kim Miso Lee Jae-Pil Heo<sup>†</sup>  
 Sungkyunkwan University

{damien, dlalth557, jaepilheo}@skku.edu

## Abstract

*Temporal Action Detection (TAD) is challenging but fundamental for real-world video applications. Recently, DETR-based models have been devised for TAD but have not performed well yet. In this paper, we point out the problem in the self-attention of DETR for TAD; the attention modules focus on a few key elements, called temporal collapse problem. It degrades the capability of the encoder and decoder since their self-attention modules play no role. To solve the problem, we propose a novel framework, Self-DETR, which utilizes cross-attention maps of the decoder to reactivate self-attention modules. We recover the relationship between encoder features by simple matrix multiplication of the cross-attention map and its transpose. Likewise, we also get the information within decoder queries. By guiding collapsed self-attention maps with the guidance map calculated, we settle down the temporal collapse of self-attention modules in the encoder and decoder. Our extensive experiments demonstrate that Self-DETR resolves the temporal collapse problem by keeping high diversity of attention over all layers. Moreover, it is validated that our simple framework achieves a new state-of-the-art performance on THUMOS14 and outperforms all the DETR-based approaches on ActivityNet-v1.3.*

## 1. Introduction

Understanding videos has become fundamental as uncountable videos are produced all over devices every moment. In the first place, action recognition using trimmed video clips led the field with tremendous advances during past decades. However, unacceptable costs to snip real-world videos fostered the literature towards temporal action detection (TAD). Temporal action detection not just classifies an action but also predicts time boundaries of untrimmed video.

Pioneering methods [2, 3, 11] adopted the concept of fixed-length windows called action proposals inspired by

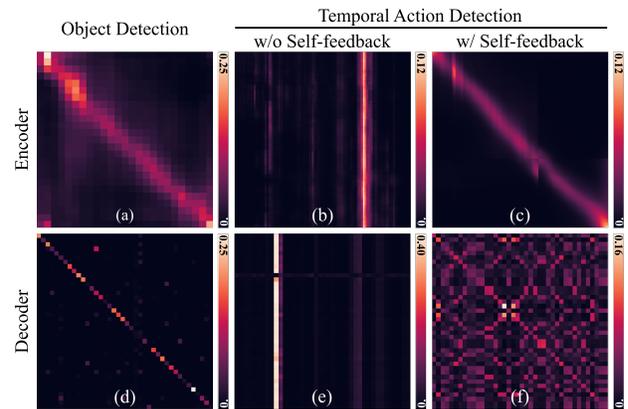


Figure 1: **Temporal collapse problem of self-attention.** The figure shows self-attention maps by DETR-based methods in object detection (OD) and temporal action detection (TAD). Each map at the top and bottom is from the last layer of the encoder and decoder, respectively. We can see that the self-attention maps of the encoder and decoder in TAD are collapsed to a small number of keys (b, e). On the other hand, those from OD and ours show high correlation for neighboring features (a, c) or query themselves (d, f).

object detection [15, 16, 36]. Meanwhile, the following approaches [20, 25, 53, 58] developed point-wise learning where they predict probabilities of start and end boundaries to solve the low-recall issue. They reached the high-recall score from more flexible action proposals by grouping each pair of start and end boundaries, but unfortunately, a bunch of generated proposals with various lengths made the ranking process more challenging. Accordingly, the previous methods heavily rely on ranking with post-processing such as non-maximum suppression (NMS) to cope with low-precision action proposals.

As DETR [4] has had a great impact on object detection, DETR-based methods for videos [22, 30, 32, 39, 44] are also introduced recently. In TAD, queries of DETR are defined as action instances of a video with their time intervals, called action queries. Here, the model learns to map these query vectors to relevant temporal features of the video to

<sup>†</sup>Corresponding author

classify and localize the actions of interest. Since there is no pre-set mapping between queries and ground-truth instances, bipartite matching associates them with minimal cost for the objectives to assign the labels. This approach can tackle the task in an end-to-end manner without any heuristics like NMS via the set-based objective.

However, it is discovered that the original DETR architecture suffers from several problems with videos and thereby does not perform well in TAD. It has been estimated that the main problem is the failure of dense attention mechanism [30, 44]. Dense attention here indicates standard attention mechanism which relates all elements without any inductive bias such as locality in convolution. To address this issue, previous DETR-based approaches in TAD revised the standard attention to boundary-sensitive module [44], deformable attention [30], or query relational attention [39].

Nevertheless, the problem of the standard attention still remains setting back DETR for TAD far behind in performance. In this paper, we confront the problem of the standard attention. Fig. 1 shows self-attention maps of DETR methods. From the figure, we find that the self-attention for TAD severely suffers from collapse to a few key elements, which we define as *temporal collapse problem*. This phenomenon implies that the model selects the shortcut to skip the self-attention to elude degeneration of the output. In contrast, self-attention in object detection and ours is highly correlated without any collapse. Hence, we point out that the temporal collapse problem in self-attention is the core to degrade DETR-based methods for TAD.

To solve the problem, we propose a new framework, Self-DETR, which provides feedback to self-attention from the encoder-decoder cross-attention. The cross-attention map contains the entire relation between the encoder and decoder features. We view the similarity between decoder queries as how much they focus on similar encoder features. Likewise, we consider the similarity between encoder features as how much they are attended by analogous decoder queries. We can obtain these two kinds of guidance by simple matrix multiplication of the cross-attention map and its transpose. From these guidance maps, the temporal collapse is relaxed by minimizing the gap of the guidance and self-attention maps. Through our extensive experiments on the public benchmarks, we validate that Self-DETR settles down the collapse by retaining high diversity of attention. As a result, Self-DETR has achieved a new state-of-the-art performance on THUMOS14, and outperforms all the DETR-based methods in ActivityNet-v1.3 without deformable attention.

To sum up, our main contributions are as follows:

- We discover the temporal collapse problem of standard attention in the DETR-based models for TAD. We reveal that the main issue is in self-attention of both the

encoder and decoder.

- We propose a novel framework, Self-DETR, which utilizes cross-attention maps to provide feedback to self-attention of the encoder and decoder to prevent the temporal collapse.
- Our extensive experiments demonstrate that Self-DETR blocks the temporal collapse efficiently by keeping high diversity of attention. Also we validate that our model reaches a new state-of-the-art performance on THUMOS14, and outperforms all the DETR-based methods on ActivityNet-v1.3.

## 2. Related Work

### 2.1. Temporal Action Detection

Temporal action detection (TAD) is the task to find a time interval of an action instance in an untrimmed video as well as classifying the instance. Early methods [2, 3, 6, 11, 13, 20, 35, 40, 41, 48, 51–53] have been realized great improvements in TAD during the last decade. As two-stage approaches had been successful in object detection [15, 16, 36], a number of methods in temporal action localization deployed multi-stage strategies [8, 14, 35, 49, 58]. As another stream of research, R-C3D [49] adopted the R-CNN architecture [36] in object detection with C3D model [45] for the first time. Similarly, TAL-Net [6] customized the Faster R-CNN architecture for TAD with dilated convolutions for Region-of-Interest (RoI) pooling.

As the subsequent work, point-wise learning has been widely introduced to generate more flexible action proposals without pre-defined temporal windows. SSN [58] and TCN [8] extended temporal context around a generated proposal to improve ranking performance. BSN [25] and BMN [24] grouped candidate start-end pairs to generate action proposals, then ranked them for final detection outputs. BSN++ [43] tackled scale imbalance problem based on BSN. Besides, graph neural networks are getting a great deal of attention in TAD [50, 54]. PGCN [54] improved ranking performance via constructing a graph of proposals based on their overlaps. GTAD [50] considered TAD as sub-graph localization problem and proposed a new framework with graph neural network. Also, TCANet [34] devised local and global temporal context aggregation. ActionFormer [55] deployed transformer encoder as backbone network, and E2E-TAD [28] studied for the end-to-end learning in TAD. AMNet [18] introduced a new framework to refine video features via action-aware attention.

### 2.2. DETR

End-to-end object DEtection with TRansformers (DETR) [4] firstly viewed object detection as a direct set prediction problem, and removed the need of the heuristic

Method	Enc. SA	Dec. SA	Dec. CA
RTD-Net [44]	MLP	Standard	Standard
TadTR [30]	Deformable	Standard	Deformable
ReAct [39]	Deformable	Heuristic	Deformable
Self-DETR	Standard	Standard	Standard

Table 1: **Comparison of baselines and Self-DETR in terms of attention methods.** The table shows the comparison in terms of the methods for self-attention and cross-attention. ‘Enc.’ and ‘Dec.’ mean the encoder and decoder, ‘SA’ and ‘CA’ indicate self-attention and cross-attention.

process, non-maximum-suppression (NMS). However, DETR required 10 times longer training time than the conventional approaches since Hungarian matching is hard to be optimized with dense attention. To cope with this issue, Deformable DETR [60] introduced sparse attention, which attends only a part of elements by learning to specify positions to focus on. Deformable attention gives locality back to DETR so that the training time is significantly reduced with performance improvement. The following DETR-based models [27, 31] further improved query representations through explicitly encoding box information, which effectively helps to stabilize training.

Transformer-based models inherently suffer from the large computational cost due to dense attention. In order to further reduce computational cost, Sparse DETR [37] introduced learnable sparsity to encoder features. To this end, they utilized encoder-decoder cross-attention maps to produce a binary mask as the guidance of sparsity. This way is quite related to ours in that they teach to sparsify encoder features by cross-attention relation as features with more attention from the decoder are more likely crucial for the task.

In TAD, DETR is also introduced recently as DETR-based models have reached a new state-of-the-art performance in object detection. RTD-Net [44] pointed out the problem of the dense attention in the DETR’s encoder, which shows nearly uniform distribution so that the self-attention layers act like over-smoothing effect. RTD-Net replaced the transformer encoder with the boundary-sensitive module to relieve the smoothing effect. On the other hand, TadTR [30] devised temporal deformable attention inspired by Deformable DETR [60] for the same problem of RTD-Net. When it comes to query relation in the decoder, ReAct [39] developed a new relation matching to enforce high correlation between queries with low-overlap and high feature similarity. This way alleviated the problem of query collapse in self-attention of the decoder since the pre-arranged relations are only permitted.

However, the problem still remains since they have detoured standard attention by deformable attention or heuristic query relation, as compared in Tab. 1. However, we directly settle down the problem of self-attention in TAD

without any deformable attention or heuristic relation.

To be specific about the problem definition, we clearly identify the problem of self-attention as temporal collapse beyond existing over-smoothing effect. It is already demonstrated that collapsed rank-1 matrix degrades performances in transformer architecture [9]. Here, we find that current situations are fully aligned as shown in Fig. 1. Therefore, we claim that the over-smooth matrix is one kind of collapsed matrix but with relatively small values.

### 3. Our Approach

In this section, we elaborate our framework, Self-DETR, which resolves the temporal collapse problem of self-attention in DETR for TAD. As shown in Fig. 2, Self-DETR follows the DETR [4] architecture. On top of it, our simple but powerful solution, self-feedback, works for guiding the self-attention layers with cross-attention map. We first explain the original DETR and differences between the original and ours in the following subsection. Afterwards, we introduce the motivation, specific method, and objective in sequence.

#### 3.1. Preliminary

**DETR.** DETR [4] is based on transformer [46] architecture and thereby has two main components: encoder and decoder. First, the transformer encoder of DETR is to learn global relations within input features. DETR uses image features from CNN as the input tokens of the encoder. There are multiple layers in the encoder to refine the input features gradually. Each layer of the encoder consists of a self-attention module and multi-layer perceptron (MLP) with layer normalization and skip connection.

On the other hand, the decoder aims to learn the relationship between the encoder features and its own inputs, learnable embedding vectors. They learn positional encoding for object detection so also called object queries. Similar to the encoder, it has several layers but also an additional cross-attention module in each layer to focus the relationship between the encoder features and the object queries. In other words, the outputs of the encoder and object queries are fed into the decoder and reinforced by self-attention and cross-attention. Finally, the output of the decoder is used to predict the class and the location of the object.

**Attention Mechanism.** Both the encoder and decoder have the attention modules [46]. They both need three variables so each has three linear layers to project inputs into three latent spaces. The projected ones are called query  $Q$ , key  $K$ , and value  $V$ , respectively. Then, the attention scores are calculated by matrix multiplication of  $Q$  and the transpose of  $K$  followed by the softmax activation function, which means how  $Q$  and  $K$  are similar. By pooling  $V$  with the scores followed by a linear projection, we can get the output

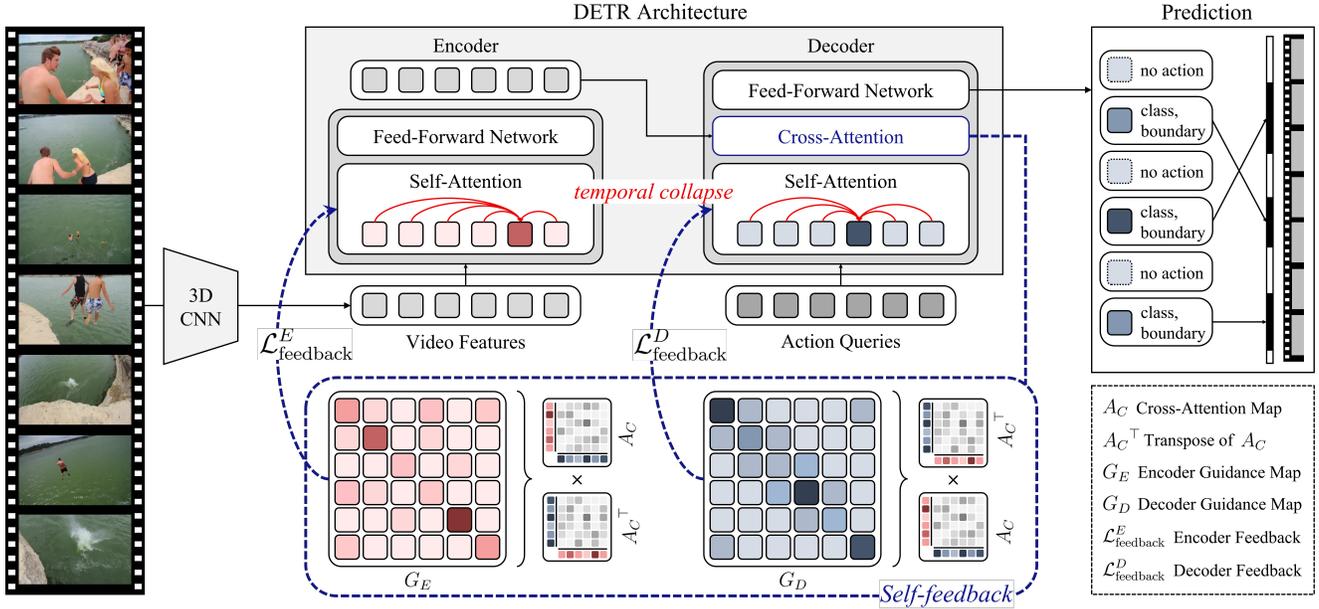


Figure 2: **Overall architecture of the proposed framework, Self-DETR.** Self-DETR is based on DETR architecture, and on top of it, we design self-feedback in order to alleviate the temporal collapse problem in the self-attention modules. We produce two types of guidance  $G_E$  and  $G_D$  for the encoder and decoder by matrix multiplication of the cross-attention map  $A_C$  and its transpose  $A_C^T$ . By applying the objectives of  $\mathcal{L}_{\text{feedback}}^E$  and  $\mathcal{L}_{\text{feedback}}^D$  to minimize the gap between the guidance and self-attention maps, the temporal collapse of self-attention disappears. This enables the model to precisely localize and classify action instances.

of the attention modules.

Formally,  $Q, K, V$  are in  $\mathbb{R}^{N_Q \times D}, \mathbb{R}^{N_K \times D}, \mathbb{R}^{N_V \times D}$ , respectively, where  $N_Q, N_K, N_V$  are the lengths of  $Q, K, V$ , and  $D$  is the number of channels. Here, we assume that the number of channels for  $Q, K, V$  are the same. We can formulate the attention mechanism as follows:

$$\text{Attention}(Q, K, V) = AV, \quad (1)$$

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right),$$

where  $A$  is the attention map,  $A^T$  indicates the transpose of  $A$ . For the self-attention module,  $Q, K, V$  are from the same input features. On the other hand,  $Q$  is from the decoder query embeddings, and  $K, V$  from the encoder features in the cross-attention module.

**DETR for TAD.** As for input to the model, we deploy features of a 3D CNN pre-trained on Kinetics [19]. Note that 3D CNN is fixed while training our model. To extract the video features, each video is fed into the 3D CNN followed by global-average pooling in spatial dimensions so that only the temporal dimension remains.

Self-DETR follows the architecture of DAB-DETR [27], but decoder queries stand for action instances, called action queries. Therefore, the decoder receives the refined video features from the encoder and relates them with action queries. Finally, the output of the decoder passes through

classification and regression heads, then final detection results are produced.

### 3.2. Guidance Maps

**Motivation.** From [9], the pure attention module itself has the bias towards a rank-1 matrix exponentially with respect to the depth of the model without skip connections. Unfortunately, the collapse problem also can be found in DETR for TAD as aforementioned Fig 1 though it has skip connections. This phenomenon implies that skip connections are not enough to slow down the collapse in TAD. It is critical to DETR-based models because the self-attention modules are just skipped for the task without learning expressive relation. Nonetheless, we observe that the cross-attention of DETR does not suffer from the temporal collapse by direct optimization from the objective. The cross-attention map contains the entire relations between encoder features and decoder queries, so we process the map to get useful information for guiding self-attention.

In this paper, we emphasize to retain the standard attention in self-attention instead of replacing it. The main benefit of keeping standard attention mechanism is that it introduces no inductive bias [10]. Convolutions or deformable attention [60] give a bias of locality so that it could lead the model to learn shortcuts. In addition, it is eventually based on heuristics where pixels resemble their neighbor-

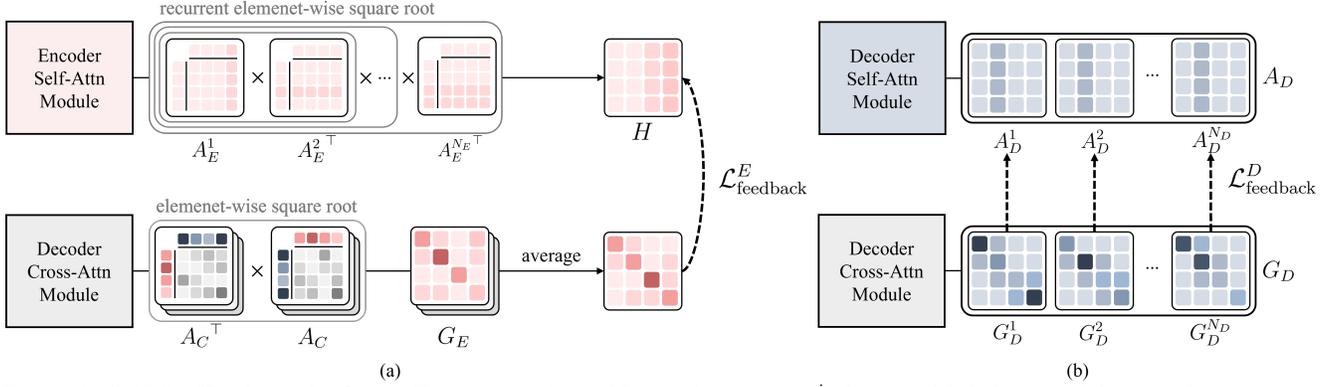


Figure 3: **Self-feedback methods.** (a) For aggregating self-attention maps  $A_E^i$  from multiple layers of the encoder, we use a series of matrix multiplication with recurrent element-wise square-root normalization. Also, we average the guidance maps  $G_E$  for the encoder from the cross-attention maps of the decoder. The guidance map is applied on the aggregated self-attention map of the encoder  $H$ . (b) On the other hand, the self-feedback for the decoder is provided layer by layer between the self-attention map  $A_D^l$  and guidance map  $G_D^l$ .

hood. The effort to remove biases sets the model free from limitations in learning [10], which is aligned with the principal of DETR [4].

**Cross-Attention.** The original goal of the cross-attention is to represent encoder features with the score of similarity between from decoder queries and encoder features. Accordingly, we usually see the map as cross-relation between decoder queries  $Q$  and encoder features  $K$ . In addition, there are two perspectives to view the cross-attention map. First, at the side of  $Q$ , they focus on the similar parts of  $K$  when they are analogous. On the other side of  $K$ , they are attended by the similar parts of  $Q$  if they are related.

We emphasize the one’s side by multiplying the cross-attention map and the transpose of it. When we denote the cross-attention map as  $A_C$ , we make the guidance map  $G_D$  for the self-attention modules of the decoder as follows:

$$G_D = \text{sqrt}(A_C A_C^\top), \quad (2)$$

where  $\text{sqrt}(\cdot)$  is element-wise square-root operation,  $A_C \in \mathbb{R}^{L_q \times L_e}$  and  $G_D \in \mathbb{R}^{L_q \times L_q}$  where  $L_q$  and  $L_e$  are the number of the query and encoder features.

Equivalently, we can obtain the guidance map  $G_E$  for the self-attention modules of the encoder as follows:

$$G_E = \text{sqrt}(A_C^\top A_C), \quad (3)$$

where  $G_E \in \mathbb{R}^{L_e \times L_e}$ .

### 3.3. Self-Feedback

Now, we have the guidance maps  $G_E$  and  $G_D$  for the self-feedback to the self-attention modules of the encoder and decoder, respectively by Eq. 2 and 3. Still, there are more options to consider for how to handle the cross-attention and self-attention maps from the multiple layers

of the encoder and decoder. This consideration is quite important because the attention map on each layer exhibits different patterns as it goes deeper. If we apply one single guidance map to force all self-attention maps to equally follow it, it will narrow down the diversity of representation. From this perspective, we deliver how to choose or aggregate the cross-attention and self-attention maps before giving the feedback, as illustrated in Fig. 3.

**Encoder.** Firstly, as for the encoder, we need to aggregate the self-attention maps of the encoder, on which we give the self-feedback. There are two possible options for aggregation: 1) average pooling, 2) matrix multiplication. One of the simplest ways is to average the self-attention maps from the encoder layers. On the other hand, we also use matrix multiplication to aggregate them. The stacked encoder layers can be viewed as the matrix multiplication of the subsequent self-attention maps when we assume there are no MLPs and skip connections. In this sense, we aggregate the self-attention maps from the multiple layers of the encoder by a series of matrix multiplication.

Formally, let us denote the self-attention map in the  $i$ -th layer of the encoder as  $A_E^i$ . Then we can describe the aggregated map as follows:

$$H^i = \text{sqrt}(H^{i-1} A_E^i{}^\top), \quad (4)$$

where  $i = [2, 3, 4, \dots, N_E]$ ,  $H^1 = A_E^1$ . The final aggregated map  $H = H^{N_E}$  with the number of encoder layers  $N_E$ .

As for the guidance map for the encoder, we simply average the cross-attention maps from the layers of the decoder. Afterwards, we obtain the guidance map  $G_E$  by Eq. 2. With the given aggregated self-attention map  $H$  of the encoder, and guidance map  $G_E$  of the decoder, we formulate the objective function of the self-feedback for the self-attention

modules of the encoder as follows:

$$\mathcal{L}_{\text{Feedback}}^E = D_{KL}(H \parallel G_E), \quad (5)$$

where  $D_{KL}$  is the Kullback–Leibler (KL) divergence.

**Decoder.** The decoder also has multiple layers, and each layer has the self-attention and cross-attention modules. On the same layer, the self-attention and cross-attention modules share the representations so their representation level is same. Therefore, we do not aggregate the attention maps from the multiple layers. Instead, we make the guidance map  $G_D$  at each layer and give it to the self-attention module on the corresponding layer.

Formally, we define the self-attention map in the  $l$ -th layer of the decoder as  $A_D^l$ . Similarly, let us denote the  $G_D$  from the  $l$ -th layer of the cross-attention module as  $G_D^l$ . We then formulate the cost function of the self-feedback for the self-attention modules of the decoder as follows:

$$\mathcal{L}_{\text{Feedback}}^D = \sum_{l=1}^{N_D} D_{KL}(A_D^l \parallel G_D^l), \quad (6)$$

where  $N_D$  is the number of decoder layers.

### 3.4. Objectives

**DETR.** Let us denote the ground-truths, and the  $M$  predictions as  $y, \hat{y} = \hat{y}_{i=1}^M$ , respectively. For the bipartite matching between the ground-truth and prediction sets, we define the optimal matching with the minimal cost to search for the permutation of  $M$  elements  $j \in J_M$  as follows:

$$\hat{j} = \arg \min_{j \in J_M} \sum_i^M \mathcal{L}_{\text{match}}(y_i, \hat{y}_{j(i)}), \quad (7)$$

where  $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{j(i)})$  is a pair-wise matching cost between  $y_i$  and the prediction with the index from  $j(i)$ , which outputs the index  $i$  from the permutation  $j$ .

Next, let us denote each ground-truth action as  $y_i = (c_i, t_i)$ , where  $c_i$  is the target class label with the background one  $\emptyset$ , and  $t_i$  is the time intervals of the start and end times. For the prediction with the index  $j(i)$ , we define the probability of the class  $c_i$  as  $\hat{p}_{j(i)}(c_i)$  and the predicted time intervals as  $\hat{t}_{j(i)}$ . Then, we define  $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{j(i)})$  as below:

$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{j(i)}) = -\mathbb{1}_{c_i \neq \emptyset} \hat{p}_{j(i)}(c_i) + \mathbb{1}_{c_i \neq \emptyset} \mathcal{L}_{\text{reg}}(t_i, \hat{t}_{j(i)}),$$

where  $\mathcal{L}_{\text{reg}}(t_i, \hat{t}_{j(i)})$  is the regression loss between the ground-truth  $t_i$  and the prediction  $\hat{t}$  with the index  $j(i)$ . The regression loss  $\mathcal{L}_{\text{reg}}$  consists of L1 and Interaction-over-Union (IoU) losses as in the DETR-based methods [30, 39, 44]. Finally, we formulate the main objective as following:

$$\mathcal{L}_{\text{DETR}}(y, \hat{y}) = \sum_{i=1}^M [-\log \hat{p}_{j(i)}(c_i) + \mathbb{1}_{c_i \neq \emptyset} \mathcal{L}_{\text{reg}}(t_i, \hat{t}_{j(i)})], \quad (8)$$

where  $\hat{j}$  is the optimal assignment from Eq. 7.

**Full Objectives.** To summarize the objectives for our framework, Self-DETR, the full objective is can be described as below:

$$\mathcal{L} = \mathcal{L}_{\text{DETR}} + \lambda_E \mathcal{L}_{\text{Feedback}}^E + \lambda_D \mathcal{L}_{\text{Feedback}}^D, \quad (9)$$

where  $\lambda_E$  and  $\lambda_D$  are the weights for the self-feedback losses for the encoder and decoder.

## 4. Experiments

### 4.1. Datasets

Our experiments are conducted on the two challenging benchmarks of temporal action detection: THUMOS14 [17] and ActivityNet-v1.3 [12].

**THUMOS14** has 1,010 and 1,574 untrimmed videos as its validation and testing samples, respectively. Specifically, 200 and 212 videos have temporal annotations in the validation and testing sets, respectively. The dataset has 20 action classes. We use the validation set for training, and the testing one for evaluation.

**ActivityNet-v1.3** contains 19,994 videos with 200 action classes. 10024, 4926, and 5044 videos are for training, validation, and testing, respectively.

### 4.2. Implementation Details

**Architecture.** We use the features of I3D [5] pre-trained on Kinetics [19]. In order for fair comparison, we are based on the size-modulated cross-attention module [27] as the baselines [30, 39] deploy deformable attention [60] which also uses the size-modulated attention. Also, Self-DETR deploys learnable anchors and the way of updating predictions iteratively as done in [30, 39]. The number of layers of the encoder and decoder is 2, and 4, respectively. The number of the queries is 40. We set the weights  $\lambda_E, \lambda_D$  of the losses of the self-feedback for the encoder and decoder as 5.

**Training.** As for both datasets, we use Adam [21] as the optimizer with the batch size of 16. For the input, we use 128 and 192 length of temporal features for THUMOS14 and ActivityNet-v1.3, respectively.

In THUMOS14, we train the framework for 120 epochs. The learning rate is decayed by 1/10 when it reaches 80 and 100 epochs. As for ActivityNet-v1.3, 20 epochs are taken for training. The learning rate decreases by cosine annealing with a warm-up of 5 epochs. In addition, we resize the features of a video with linear interpolation to 192 length.

**Inference.** We slice the temporal features with a 128-length window with overlap of 32 for THUMOS14. As for ActivityNet-v1.3, we resize the features to 192 length as done in training. Also, we use the top 100, and 200 predictions after non-maximum suppression (NMS) for the final

Method	THUMOS14						ActivityNet-v1.3			
	0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
<b>Standard Methods</b>										
BSN [25]	53.5	45.0	36.9	28.4	20.0	36.8	46.46	29.96	8.02	29.17
BMN [24]	56.0	47.4	38.8	29.7	20.5	38.5	50.07	34.78	8.29	33.85
GTAD [50]	54.5	47.6	40.2	30.8	23.4	39.3	50.36	34.60	9.02	34.09
BC-GNN [1]	57.1	49.1	40.4	31.2	23.1	40.2	50.56	34.75	9.37	34.26
BSN++ [43]	59.9	49.5	41.3	31.9	22.8	41.1	51.27	35.70	8.33	34.88
IC & IC [57]	53.9	50.7	45.4	38.0	28.5	43.3	43.47	33.91	9.21	30.12
MUSES [29]	68.9	64.0	56.9	<b>46.3</b>	31.0	53.4	50.02	34.97	6.57	33.99
CSA [42]	64.4	58.0	49.2	38.2	27.8	47.5	52.44	<u>36.69</u>	5.18	35.43
PBRNet [26]	58.5	54.6	51.3	41.8	29.5	47.1	53.96	34.97	8.98	35.01
VSGN [56]	66.7	60.4	52.4	41.0	30.4	50.2	52.38	36.01	8.37	35.07
ContextLoc [61]	68.3	63.8	54.3	41.8	26.2	50.9	56.01	35.19	3.55	34.23
AFSD [23]	67.3	62.4	55.5	43.7	31.1	52.0	52.40	35.30	6.50	34.40
DCAN [7]	68.2	62.7	54.1	43.9	<b>32.6</b>	52.3	51.78	35.98	<u>9.45</u>	35.39
Zhu <i>et al.</i> [62]	<b>72.1</b>	<b>65.9</b>	<b>57.0</b>	<u>44.2</u>	28.5	<b>53.5</b>	<b>58.14</b>	36.30	6.16	35.24
RCL [47]	70.1	62.3	52.9	42.7	30.7	51.0	54.19	36.19	9.17	<u>35.98</u>
TAGS [33]	68.6	63.8	<b>57.0</b>	<b>46.3</b>	<u>31.8</u>	52.8	<u>56.30</u>	<b>36.80</b>	<b>9.60</b>	<b>36.50</b>
<b>DETR-based Methods</b>										
RTD-Net [44]	68.3	62.3	51.9	38.8	23.7	49.0	47.21	30.68	<b>8.61</b>	30.83
TadTR [30]	62.4	57.4	49.2	37.8	26.3	46.6	49.10	32.60	8.50	32.30
ReAct [39]	<u>69.2</u>	<u>65.0</u>	<u>57.1</u>	<b>47.8</b>	<b>35.6</b>	<u>55.0</u>	<u>49.60</u>	<u>33.00</u>	<u>8.60</u>	<u>32.60</u>
Self-DETR	<b>74.6</b>	<b>69.5</b>	<b>60.0</b>	<u>47.6</u>	<u>31.8</u>	<b>56.7</b>	<b>52.25</b>	<b>33.67</b>	8.40	<b>33.76</b>

Table 2: **The comparison results with the-state-of-the-art on THUMOS14 and ActivityNet-v1.3.** The table shows the evaluation results of the two types: standard and DETR-based models. In THUMOS14, our model shows the state-of-the-art performance over all previous methods. Also, our model outperforms the existing DETR-based methods on ActivityNet.

localization results for ActivityNet-v1.3, and THUMOS14, respectively. We use SoftNMS with the NMS threshold of 0.40. For the class label, we fuse our classification scores with the top-1 video-level predictions of [59] as done in [30, 39, 44] for ActivityNet-v1.3.

### 4.3. Comparison with the State-of-the-Art

We compare the-state-of-the-art methods to evaluate our framework, Self-DETR, on THUMOS14 and ActivityNet-v1.3 datasets. Table 2 shows the comparison results with the state-of-the-art methods on THUMOS14 and ActivityNet-v1.3. On THUMOS14, Self-DETR shows a new state-of-the-art performance over all of the existing approaches. Compared to the standard methods which are not based on the set prediction, our model shows superior performances at all the IoU thresholds. As for the DETR-based methods [30, 39, 44], Self-DETR outperforms all the previous DETR-based methods in terms of the average mAP as well as the IoU thresholds of 0.3, 0.4, and 0.5.

As for ActivityNet-v1.3, Self-DETR also shows a new state-of-the-art performance among the DETR-based approaches. Interestingly, the APs at the 0.50 and 0.95 have almost reached the state-of-the-art of the standard methods.

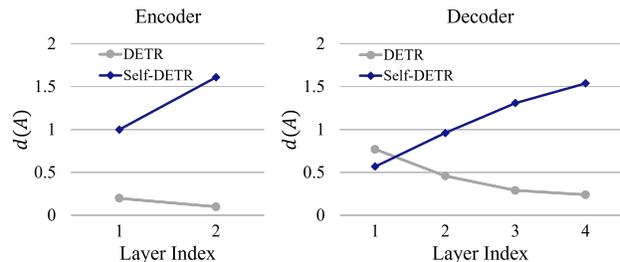


Figure 4: **Diversity of self-attention maps.** In order to further analyze the effect of the self-feedback for the temporal collapse problem, we measure the diversity defined in Eq. 10 of the self-attention maps.

This shows that the performance of the DETR-based methods has become comparable to the standard approaches.

### 4.4. Ablation Studies

**Diversity of Self-Attention.** To further analyze the effect of the self-feedback, we can measure the diversity of self-attention maps according to [9]. The diversity  $d(A)$  for the attention map  $A$  is the measure of the closeness between the attention map and a rank-1 matrix as defined as below:

$$d(A) = \|A - \mathbf{1}a^\top\|, \text{ where } a = \arg \min_{a'} \|A - \mathbf{1}a'^\top\|, \quad (10)$$

$\mathcal{L}_{\text{Feedback}}^E$	$\mathcal{L}_{\text{Feedback}}^D$	0.3	0.4	0.5	0.6	0.7	Avg.
.	.	70.5	64.3	53.9	39.3	23.8	50.3
✓	.	73.5	67.9	57.2	43.2	26.6	53.7
.	✓	73.4	67.4	58.5	44.5	28.7	54.4
✓	✓	<b>74.5</b>	<b>69.5</b>	<b>60.0</b>	<b>47.6</b>	<b>31.8</b>	<b>56.7</b>

Table 3: **Ablation on self-feedback.** To validate each self-feedback loss for the encoder and decoder, we have conducted experiments for ablating them on THUMOS14.

Encoder	Decoder	0.3	0.4	0.5	0.6	0.7	Avg.
last	layer	73.0	67.2	57.8	44.9	28.8	54.3
average	layer	74.6	68.6	59.2	46.0	<b>32.0</b>	56.1
matmul	layer	<b>74.6</b>	<b>69.5</b>	<b>60.0</b>	<b>47.6</b>	31.8	<b>56.7</b>
matmul	last	72.2	66.6	56.6	42.5	27.8	53.1
matmul	average	70.4	63.8	54.0	41.5	26.0	51.1

Table 4: **Aggregation methods for guidance.** The table shows the comparison results of the aggregation methods for self-attention maps of encoder and decoder.

where  $\|\cdot\|$  denotes the  $\ell_1, \ell_\infty$ -composite matrix norm,  $a, a'$  are column vectors of the attention map  $A$ , and  $\mathbf{1}$  is an all-ones vector. Note that the rank of  $\mathbf{1}a^\top$  is 1, and therefore, a smaller value of  $d(A)$  means  $A$  is closer to a rank-1 matrix.

Fig. 4 shows the diversity on each layer of the encoder and decoder for the baseline DETR and Self-DETR. The diversity is measured on the test set on THUMOS14 averaged over 160 randomly selected samples. As the model depth gets deeper, the diversity of the baseline decreases close to 0. However, the diversity of Self-DETR does not fall down and even increases since the self-feedback loss guides the self-attention maps.

**Ablation on Self-Feedback.** In order to validate contribution of each self-feedback for the self-attention module of the encoder and decoder, we have conducted experiments for ablating the self-feedback on the THUMOS14 dataset.

Tab. 3 shows the ablation study on the self-feedback. As reported, each self-feedback for the encoder and decoder significantly improves the overall performance at all the IoU thresholds. Moreover, when we use the self-feedback for both the encoder and decoder, the performance gain becomes larger reaching the state-of-the-art performance on THUMOS14. Through this ablation study, each of the self-feedback losses  $\mathcal{L}_{\text{Feedback}}^E$  and  $\mathcal{L}_{\text{Feedback}}^D$  is crucial to relieve the temporal collapse problem of self-attention in DETR.

**Aggregation for Guidance.** To search for the optimal aggregation method, we conducted the experiments on THUMOS14 in Tab. 4. ‘last’ means providing self-feedback only for the last self-attention layer. ‘average’ indicates average pooling self-attention maps and ‘matmul’ is matrix multiplication. Also, ‘layer’ means self-feedback to each layer with guidance from the corresponding layer of the decoder.

In the table, ‘average’ and ‘matmul’ aggregation methods show superior performances over the ‘last’ method. As

Method	0.3	0.4	0.5	0.6	0.7	Avg.
baseline	70.5	64.3	53.9	39.3	23.8	50.3
relative	69.1	63.4	53.8	40.5	24.6	50.3
identity	69.1	62.0	52.9	40.6	24.6	49.9
diversity	56.5	49.5	39.2	26.2	14.1	37.1
cross-attn.	<b>74.6</b>	<b>69.5</b>	<b>60.0</b>	<b>47.6</b>	<b>31.8</b>	<b>56.7</b>

Table 5: **Alternatives for self-feedback.** The table shows results of alternatives for self-feedback.

$\lambda_E$	$\lambda_D$	0.3	0.4	0.5	0.6	0.7	Avg.
1	5	73.1	67.2	58.6	45.1	29.7	54.7
3	5	73.8	67.8	58.5	46.4	30.6	55.4
5	5	<b>74.6</b>	<b>69.5</b>	<b>60.0</b>	<b>47.6</b>	31.8	<b>56.7</b>
5	3	73.7	67.9	59.1	47.5	<b>32.1</b>	56.1
5	1	73.4	67.1	57.7	44.3	28.3	54.2

Table 6: **Loss weights for self-feedback.** The table shows results depending on weights for self-feedback loss.

for the decoder, ‘layer’ approach outperforms ‘last’ and ‘average’ methods as representation levels of self- and cross-attention on the same layer are well compatible.

**Alternatives for self-feedback.** Table. 5 shows the results for alternatives to the cross-attention maps for self-feedback on THUMOS14. ‘relative’ stands for deploying relative attention [38] to self-attention of the encoder. Also, ‘identity’ means forcing the model to follow the identity matrix as guidance map, and ‘diversity’ indicates we give the objective to maximize the diversity defined by Eq. 10. The table shows that all alternatives do not bring a performance gain. Hence, self-feedback of our guidance map from cross-attention is not just for regularization for diversity of self-attention but for learning expressive self-relation.

**Loss Weights of Self-Feedback.** Tab. 6 shows performances for the different weights for the self-feedback losses for the encoder and decoder in THUMOS14. It shows that the weight of 5 is suitable for the encoder and decoder.

## 5. Conclusion

In this paper, we have explicitly discovered the temporal collapse problem of self-attention in DETR for TAD. To alleviate the collapse, we have proposed a new framework, Self-DETR, which re-purposes cross-attention between the encoder and decoder in DETR to produce guidance map as self-feedback for self-attention. Our extensive experiments have demonstrated that Self-DETR has resolved the collapse of self-attention preserving diversity of relation while reaching the state-of-the-art performance.

**Acknowledgement.** This work was supported in part by MSIT/IITP (No. 2022-0-00680, 2019-0-00421, 2020-0-01821, 2021-0-02068), and MSIT&KNPA/KIPoT (Police Lab 2.0, No. 210121M06).

## References

- [1] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 7
- [2] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017. 1, 2
- [3] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1914–1923, 2016. 1, 2
- [4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 1, 2, 3, 5
- [5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 6
- [6] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018. 2
- [7] Guo Chen, Yin-Dong Zheng, Limin Wang, and Tong Lu. Dcan: improving temporal action detection via dual context aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 248–257, 2022. 7
- [8] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5793–5802, 2017. 2
- [9] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, pages 2793–2803. PMLR, 2021. 3, 4, 7
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 4, 5
- [11] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision*, pages 768–784. Springer, 2016. 1, 2
- [12] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 6
- [13] Jiyang Gao, Kan Chen, and Ram Nevatia. Ctap: Complementary temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–83, 2018. 2
- [14] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3628–3636, 2017. 2
- [15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1, 2
- [16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1, 2
- [17] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. <http://cvcv.ucf.edu/THUMOS14/>, 2014. 6
- [18] Tae-Kyung Kang, Gun-Hee Lee, Kyung-Min Jin, and Seong-Whan Lee. Action-aware masking network with group-based attention for temporal action localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6058–6067, 2023. 2
- [19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4, 6
- [20] Ji-Hwan Kim and Jae-Pil Heo. Learning coarse and fine features for precise temporal action localization. *IEEE Access*, 7:149797–149809, 2019. 1, 2
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [22] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021. 1
- [23] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2021. 7
- [24] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3889–3898, 2019. 2, 7

- [25] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. [1](#), [2](#), [7](#)
- [26] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11612–11619, 2020. [7](#)
- [27] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: dynamic anchor boxes are better queries for DETR. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [3](#), [4](#), [6](#)
- [28] Xiaolong Liu, Song Bai, and Xiang Bai. An empirical study of end-to-end temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20010–20019, 2022. [2](#)
- [29] Xiaolong Liu, Yao Hu, Song Bai, Fei Ding, Xiang Bai, and Philip HS Torr. Multi-shot temporal event localization: a benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12596–12606, 2021. [7](#)
- [30] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. End-to-end temporal action detection with transformer. *arXiv preprint arXiv:2106.10271*, 2021. [1](#), [2](#), [3](#), [6](#), [7](#)
- [31] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021. [3](#)
- [32] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23023–23033, 2023. [1](#)
- [33] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Proposal-free temporal action detection via global segmentation mask learning. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 645–662. Springer, 2022. [7](#)
- [34] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 485–494, 2021. [2](#)
- [35] Haonan Qiu, Yingbin Zheng, Hao Ye, Yao Lu, Feng Wang, and Liang He. Precise temporal action localization by evolving temporal proposals. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 388–396. ACM, 2018. [2](#)
- [36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. [1](#), [2](#)
- [37] Byungseok Roh, Jaewoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse DETR: efficient end-to-end object detection with learnable sparsity. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [3](#)
- [38] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics, 2018. [8](#)
- [39] Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. React: Temporal action detection with relational queries. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pages 105–121. Springer, 2022. [1](#), [2](#), [3](#), [6](#), [7](#)
- [40] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5734–5743, 2017. [2](#)
- [41] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058, 2016. [2](#)
- [42] Deepak Sridhar, Niamul Quader, Srikanth Muralidharan, Yaoxin Li, Peng Dai, and Juwei Lu. Class semantics-based attention for action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13739–13748, 2021. [7](#)
- [43] Haisheng Su, Weihao Gan, Wei Wu, Yu Qiao, and Junjie Yan. Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2602–2610, 2021. [2](#), [7](#)
- [44] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13526–13535, October 2021. [1](#), [2](#), [3](#), [6](#), [7](#)
- [45] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. [2](#)
- [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. [3](#)
- [47] Qiang Wang, Yanhao Zhang, Yun Zheng, and Pan Pan. Rcl: Recurrent continuous localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Com-*

- puter Vision and Pattern Recognition (CVPR), pages 13566–13575, June 2022. 7
- [48] Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaoou Tang. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716*, 2017. 2
- [49] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017. 2
- [50] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020. 2, 7
- [51] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2678–2687, 2016. 2
- [52] Jun Yuan, Bingbing Ni, Xiaokang Yang, and Ashraf A Kasim. Temporal action localization with pyramid of score distribution features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3093–3102, 2016. 2
- [53] Zehuan Yuan, Jonathan C Stroud, Tong Lu, and Jia Deng. Temporal action localization by structured maximal sums. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3684–3692, 2017. 1, 2
- [54] Runhao Zeng, Wenbing Huang, Minghui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019. 2
- [55] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 492–510. Springer, 2022. 2
- [56] Chen Zhao, Ali K Thabet, and Bernard Ghanem. Video self-stitching graph network for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13658–13667, 2021. 7
- [57] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *European Conference on Computer Vision*, pages 539–555. Springer, 2020. 7
- [58] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017. 1, 2
- [59] Y Zhao, B Zhang, Z Wu, S Yang, L Zhou, S Yan, L Wang, Y Xiong, D Lin, Y Qiao, et al. Cuhk & ethz & siat submission to activitynet challenge 2017. *arXiv preprint arXiv:1710.08011*, 8, 2017. 7
- [60] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3, 4, 6
- [61] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. Enriching local and global contexts for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13516–13525, October 2021. 7
- [62] Zixin Zhu, Le Wang, Wei Tang, Ziyi Liu, Nanning Zheng, and Gang Hua. Learning disentangled classification and localization representations for temporal action localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 2, 2022. 7