TE-TAD: Towards Full End-to-End Temporal Action Detection via Time-Aligned Coordinate Expression

Ho-Joong Kim[1] Jung-Ho Hong[1] Heejo Kong[2] Seong-Whan Lee[1*]

[1]Dept. of Artificial Intelligence, Korea University, Seoul, Korea

[2]Dept. of Brain and Cognitive Engineering, Korea University, Seoul, Korea

{hojoong_kim, jungho-hong, hj_kong, sw.lee}@korea.ac.kr

Abstract

In this paper, we investigate that the normalized coordinate expression is a key factor as reliance on hand-crafted components in query-based detectors for temporal action detection (TAD). Despite significant advancements towards an end-to-end framework in object detection, query-based detectors have been limited in achieving full end-to-end modeling in TAD. To address this issue, we propose TE-TAD, a full end-to-end temporal action detection transformer that integrates time-aligned coordinate expression. We reformulate coordinate expression utilizing actual timeline values, ensuring length-invariant representations from the extremely diverse video duration environment. Furthermore, our proposed adaptive query selection dynamically adjusts the number of queries based on video length, providing a suitable solution for varying video durations compared to a fixed query set. Our approach not only simplifies the TAD process by eliminating the need for hand-crafted components but also significantly improves the performance of query-based detectors. Our TE-TAD outperforms the previous query-based detectors and achieves competitive performance compared to state-of-the-art methods on popular benchmark datasets. Code is available at: https://github.com/Dotori-HJ/TE-TAD

1. Introduction

Temporal action detection (TAD) plays an essential role in video understanding and its numerous real-world applications, such as video surveillance, video summarization, and video retrieval. TAD aims to recognize and localize actions within untrimmed video sequences by identifying the class labels with precise start and end times of action instances. Recently, TAD methods can be mainly divided by three approaches: anchor-based [3, 15, 16, 21, 23, 31, 34], anchor-free [6, 14, 27, 32], and query-based [18, 26, 28] detector.

Query-based detectors, inspired by DETR [4], have attracted interest because of their potential to eliminate reliance on hand-crafted components, such as the sliding window and non-maximum suppression (NMS). This potential derives from adopting a set-prediction mechanism, which aims to provide an end-to-end detection process by utilizing a one-to-one matching paradigm. Despite these advantages, query-based detectors encounter significant challenges in two aspects: (1) they show decreased performance when dealing with extended temporal coverage, often making them a less favorable option compared to anchor-free detectors, and (2) due to limited extended temporal coverage, the reliance on the sliding window approach leads to redundant proposals and necessitates the use of NMS.

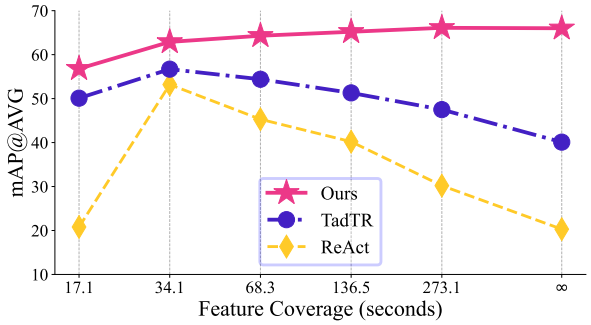


Figure 1. Performance comparison of query-based detectors across various feature coverages on THUMOS14, demonstrating how extending the feature coverage impacts detection performance, as measured by mean Average Precision (mAP@AVG). The full end-to-end setting that without coverage constraints is denoted by ∞.

To demonstrate these issues, we conduct experiments by increasing the feature coverages on existing query-based detectors. Fig. 1 demonstrates the change in performance across various feature coverage. Except for unrestricted feature coverage denoted by ∞, feature coverage is calculated from the window size of the sliding window method. As shown in the graph, even though wide feature coverage is beneficial to capturing longer context, the performance of existing query-based detectors diminishes in mean Average Precision (mAP) as feature coverage increments. This result indicates that there are significant limitations to query-based detectors in scalability to temporal length, despite efforts to address long durations [32] in TAD.

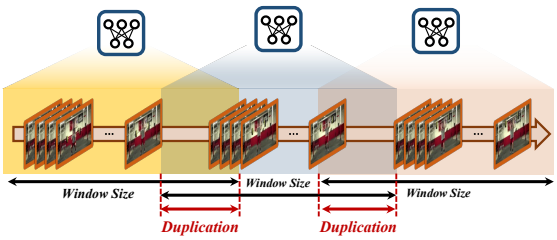
*Corresponding author

Furthermore, Fig. 2 shows the second issue, illustrating the limitations of the sliding window approach. As shown in Fig. 2(a), the sliding window is limited to address long-range duration due to its limited window size. Moreover, the sliding window approach contains overlapping areas to prevent predictions from being truncated in the middle. These overlapping areas generate duplicate predictions, necessitating the use of NMS to filter the false positive cases. This reliance on NMS contradicts the set-prediction goal of minimizing hand-crafted components, thus hindering the achievement of a fully end-to-end TAD. To address these issues, we investigate why extended temporal coverage adversely affects the performance of query-based detectors (Sec. 3.3). Our investigation reveals that the conventional use of normalized coordinate expressions is a significant factor, disturbing the achievement of a full end-to-end TAD.

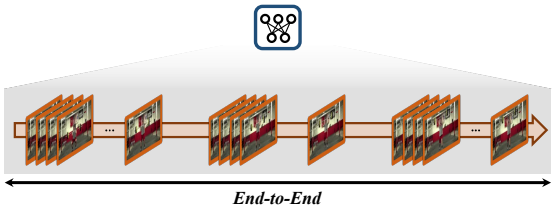
In this paper, we propose a full end-to-end temporal action detection transformer that integrates time-aligned coordinate expression (TE-TAD), which reformulates normalized coordinate expression to actual timeline video values. Our reformulation enables the query-based detector to effectively address length-invariant modeling by avoiding the distortion of the normalizing process, which not only enhances the detection performance but also simplifies the detection process. Our TE-TAD stabilizes the training process of the query-based detector when dealing with extended videos; our approach shows significant improvements and completely removes the reliance on hand-crafted components such as the sliding window and NMS. Furthermore, we introduce an adaptive query selection that effectively addresses various video lengths, dynamically adjusting the number of queries in response to the temporal length of each video. In contrast to relying on a fixed set of queries, our TE-TAD provides a suitable approach to process diverse video lengths. Our approach shows significant improvements compared to previous query-based detectors and achieves competitive performance with state-of-the-art methods on popular benchmark datasets: THUMOS14 [10], ActivityNet v1.3 [8], and EpicKitchens [7].

Our contributions are summarized as three-fold:

- We propose a full end-to-end temporal action detection transformer that integrates time-aligned coordinate expression (TE-TAD), which preserves the set-prediction mechanism and enables a full end-to-end modeling for TAD by eliminating the hand-crafted components.
- Our approach introduces a length-invariant mechanism to query-based detectors, significantly improving scalability in handling varying lengths of videos.
- Our TE-TAD significantly outperforms the previous query-based detectors and achieves competitive performance compared to state-of-the-art methods, even without hand-crafted components such as the sliding window and NMS.



(a) Sliding window



(b) End-to-end setting

Figure 2. Comparison between sliding window and end-to-end settings. The sliding window generates redundant proposals in the duplicated area.

2. Related Work

Action Recognition Action Recognition is a foundational task in video understanding, categorizing video sequences into distinct action classes. Notable models include I3D [5], which enhances the inception network with 3D convolutions, and R(2+1)D [29], separating 3D convolutions into 2D spatial and 1D temporal parts for efficient processing. TSP [2] introduces temporal channel shifting for effective temporal modeling without extra computation. VideoSwin [20], employing the SwinTransformer [19] architecture, excels in complex video data recognition. These models serve as backbones for extracting video features, directly impacting performance in subtasks like TAD.

Anchor-based Detector Anchor-based detectors [3, 15, 16, 21, 23, 31, 34] leverage predefined anchor boxes to generate action proposals. These hand-designed anchors hinder the diverse range of action instances because of a lack of flexibility in the localization of the action instances. This approach inherently limits and necessitates additional post-processing steps to discard redundant proposals because they model the one-to-many assignment training strategy.

Anchor-free Detector Anchor-free detectors [6, 14, 25, 27, 32, 35] offer more flexibility in action instance localization compared to anchor-based detectors by adopting an asymmetric modeling approach. For instance, ActionFormer [32] significantly enhances TAD performance by employing a transformer-based architecture that captures long-range video dependencies. TriDet [27] demonstrates superior performance in TAD by employing a trident prediction scheme and their proposed convolution-based architecture. Despite their improvements, anchor-free detectors rely on hand-crafted components to remove redundant proposals using NMS because they adopt a one-to-many assignment train-

ing manner. In contrast, our approach directly addresses a one-to-one matching scheme, eliminating the use of NMS.

Query-based Detector Query-based detectors, inspired by DETR [4], introduce a set-prediction mechanism, thereby reducing the reliance on hand-crafted components, which ideally prevents the need for NMS. However, existing query-based detectors still require NMS because their model design inherently breaks the one-to-one matching paradigm. RTD-Net [28] utilizes a one-to-many matching to mitigate the slow convergence issue associated with the detection transformer. This approach inherently breaks the one-to-one assignment. ReAct [26] modify the decoder's self-attention, called relational attention, only adopt self-attention between their defined relations. This partial adoption of the decoder's self-attention disturbs the set-prediction mechanism because they cannot capture the whole context of queries. Furthermore, previous query-based detectors [18, 26, 28] adopt the sliding window method that contains overlapping areas, causing redundant proposals. Furthermore, TadTR [18] deals with one-to-many matching at training loss. TadTR employs cross-window fusion (CWF), which applies NMS to overlapping areas to remove redundant proposals. In contrast, our approach entirely preserves the one-to-one matching paradigm, which enables a full end-to-end modeling.

3. Our Approach

3.1. Overview

In this section, we first discuss about the limitations of existing query-based detectors in TAD, focusing on the normalized coordinate expression. The normalized coordinate expression, used in existing models, causes matching instability and sensitivity, especially in extended video scenarios. Subsequently, to introduce our TE-TAD, we describe the reformulation of normalized coordinate expression to timeline coordinate expression and adaptive query proposals to ensure the length-invariant modeling. The overall architecture of TE-TAD is illustrated in Fig. 5.

3.2. Preliminary

Let $X \in \mathbb{R}^{T_0 \times C}$ denote the video feature sequence extracted by the backbone network, where T_0 is the temporal length of the features, and C is the dimension of the video feature. Each element in the video feature sequence represented as $X = \{x_t\}_{t=1}^{T_0}$, corresponds to a snippet at timestep t, with each snippet comprising a few consecutive frames. These snippets are processed using a pre-trained backbone network such as I3D [5] or SlowFast [9]. Each video contains numerous action instances, and each action instance contains start and end timestamps s and e, along with its action class c. Formally, the set of action instances in a video is represented as $\mathcal{A} = \{(s_n, e_n, c_n)\}_{n=1}^{N}$, where

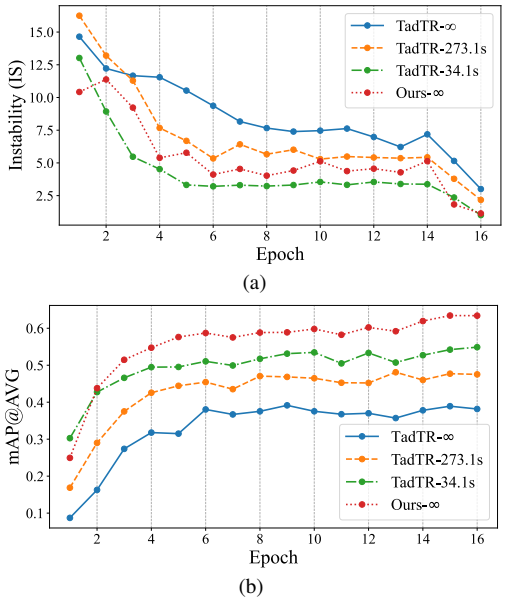


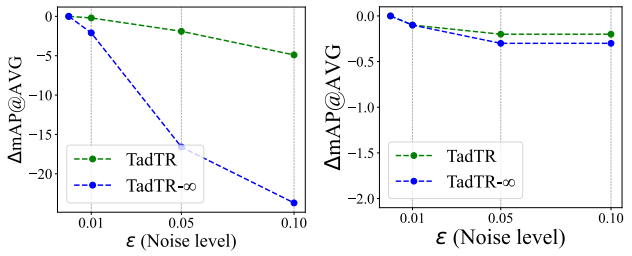
Figure 3. Comparative analysis of instability and detection performance on THUMOS14: (a) variance in instance matching quantified by IS; (b) the performance change in mAP. Feature coverage lengths are denoted by the values following the dash (-). The symbol ∞ denotes an unrestricted end-to-end setting.

N is the number of action instances, and s_n and e_n are the start and end timestamps of an action instance, respectively, and c_n is its action class. The main goal of TAD is to accurately predict the set of action instances $\mathcal{A}$ for any given video. Previous query-based detectors [18, 26, 28] typically compute predicted values of center $\hat{c}$ and width $\hat{d}$ using a sigmoid function (σ). Consequently, existing models decode predicted start $\hat{s}$ and end $\hat{e}$ timestamps by $\sigma(\hat{c}) - \sigma(\hat{d})$ and $\sigma(\hat{c}) + \sigma(\hat{d})$ to start and end timestamps, respectively.

3.3. Exploring Key Issues

Matching Instability As discussed in Sec. 1, extended video lengths significantly influence the performance of existing query-based methods. To investigate this issue, we conduct a comparative study on the instability (IS) [13] and the detection performance across different feature coverage scenarios on THUMOS14. Here, IS is a quantitative measurement of the inconsistency of matching during the training process. For query-based detectors, fluctuations in matched targets compel the model to be learned from different values for the same input, leading to performance degradation. In the TadTR setting, smaller window sizes have more steps per epoch due to generating more sliding windows, whereas larger window sizes yield fewer steps per epoch. This mismatch in the number of steps results that with fewer updates per epoch, there is inherently less change to the model, which shows lower instability. For a fair comparison, we match the number of iteration steps per epoch, aligning with TadTR-34.1s (original TadTR).

Fig. 3 illustrates the instability and detection performance across diverse feature coverage. As feature coverage increases, we observe a rise in IS across the training epoch, indicating less stable instance matching, which leads to a decline in detection performance. This analysis underscores the challenge of maintaining consistent learning when with extended temporal lengths. Furthermore, a direct comparison between TadTR models and our method reveals that Ours-∞ maintains a level of stability comparable to TadTR-34.1s. This indicates that our approach significantly stabilizes the training process relative to TadTR-∞. Moreover, our method demonstrates similar levels of matching instability yet shows significant improvements under more challenging conditions for matching problems, even when compared to models with shorter feature coverage like TadTR-34.1s and TadTR-273.1s.



(a) $\sigma(\hat{c}+\epsilon)$ (b) $\sigma(\hat{d}+\epsilon)$

Figure 4. Analysis of noise tolerance on predicted value. The noise level ϵ sampled from uniform distribution and injected before the *sigmoid* function.

Sensitivity to Localize Action Instances To further explore this issue, we investigate the sensitivity of model prediction measured by adopting minor perturbations in predictions of localization. We inject a small scale of noise sampled from a uniform distribution before adopting the *sigmoid* function that normalizes coordinates within a [0, 1] range. Fig. 4 shows that even minimal noise injection significantly affects the prediction when applying the center prediction value. The noise $\epsilon \sim \text{Uniform}(-\alpha, \alpha)$ is sampled from the uniform distribution. As illustrated in Fig. 4(a), the model's sensitivity to small shifts is evident from the significant decline in ΔmAP@AVG upon introducing noise to the center predicted value. Notably, with noise levels of ± 0.01 and ± 0.1, after processing through the *sigmoid* function (σ), the maximal shifts are confined within ± 0.0025 and ± 0.025 in the normalized coordinate space, respectively. These findings highlight that even minor output variations amplify in extended videos, leading to considerable drops in performance due to the heightened sensitivity in the normalized coordinate framework. To address this issue, we introduce a time-aligned coordinate expression that is not normalized, thereby ensuring independence from video length and reducing sensitivity.

3.4. TE-TAD

This part describes our TE-TAD for a full end-to-end TAD. We adopt the TadTR [18] architecture as a baseline method, including encoder, decoder, and temporal deformable attention architecture. Starting with the baseline, we mainly address three aspects: (1) adopting multi-scale and two-stage methods from the previous methods in object detection to bridge the performance gap between query-based and anchor-free detectors, (2) reformulating coordinate expression utilizing the actual time values to address extended video length in an end-to-end setting, and (3) proposing an adaptive query selection that dynamically adjusts the number of queries based on the diverse length of videos.

Embedding & Multi-Scale Features We project input features X using a single convolutional neural network to align them with the dimension of the transformer architecture. The projection maps the input feature X to the embedded feature $Z_1 \in \mathbb{R}^{D \times T_1}$, where D denotes the channel dimension of the encoder and decoder transformer architecture. The temporal length T_1 of the embedded features Z_1 remains the same as the original T_0. Subsequently, following previous approaches [27, 32], we incorporate multi-scale generation to effectively address varying lengths of actions. Unlike utilizing a transformer [32], we employ a single convolution layer with a stride of 2 to produce features at each scale level as follows:

$$Z_l = \text{LayerNorm}_l(\text{Conv}_l(Z_{l-1})), l \in \{2, ..., L\}, \quad (1)$$

where $Z_l \in \mathbb{R}^{C \times T_l}$ represents the embedded features at each level l, and L denotes the total number of feature levels. Each subsequent level l has a temporal length T_l that is half of the temporal level of the previous level T_{l-1}. The LayerNorm_l and Conv_l denote l-th layer normalization and convolutional neural networks, respectively. We do not apply any activation function in this process to deliver the raw feature representations to the transformer detector.

Time-Aligned Query Generation Our method follows a two-stage approach [33, 37] that generates initial action proposals using the transformer encoder. The previous two-stage approach [37] provides the reference to the encoder's outputs. The transformer encoder predicts a binary foreground score, $p^{(0)}$, and segment offsets $\Delta c^{(0)}$ and $\Delta d^{(0)}$ to refine segments based on reference, for each time t and level l. We define reference for center c^{ref} and width d^{ref} predictions, aligning with the real timeline of the video. For each scale level l, the reference for the center is computed as follows:

$$c^{\text{ref}} = \left\{ t \times \frac{f}{w \times 2^{l-1}} + \frac{w \times 2^{l-1}}{2} \right\}_{t=1}^{T_l}, \; l \in \{1, 2, \ldots, L\}, \quad (2)$$

where f denotes the frame-per-second rate of the video, and w represents step size for feature extracting that indicates

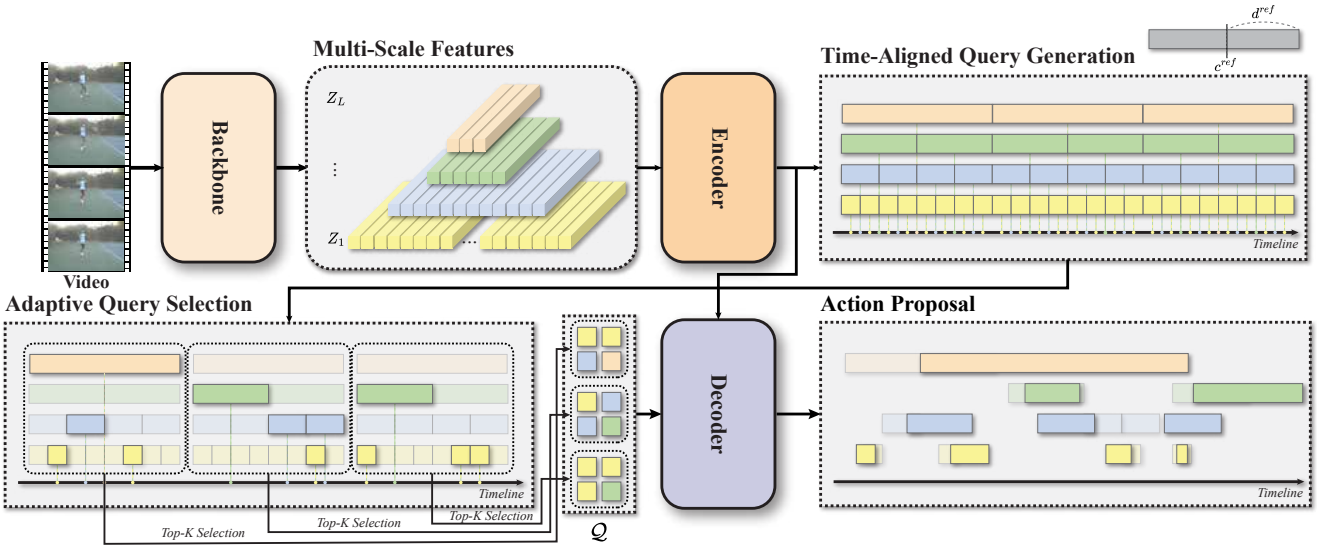


Figure 5. Overview of the TE-TAD. Starting with video input, the architecture processes through a backbone for feature extraction, generating multi-scale features Z. These are encoded and subsequently passed through an adaptive query selection, aligning with the video timeline for initial query generation. The decoder refines these queries layer-by-layer, culminating in the refinement of action proposals.

how many frames to step when extracting features. The factor 2^{l-1} fits the temporal lengths to the embedded features from the multi-scale generation, corresponding with the actual timeline. Subsequently, the reference for the width is computed as follows:

$$d^{\text{ref}} = \alpha \cdot f \times 2^{l-1},\ l \in \{1, 2, \ldots, L\}, \tag{3}$$

where α is the base scale for encoder proposals, which adjusts the length of the reference width length. Consequently, the time-aligned queries are decoded as follows:

$$(\hat{c}^{(0)}, \hat{d}^{(0)}) = (c^{\text{ref}} + \Delta c^{(0)} \cdot d^{\text{ref}}, \exp(\ln(d^{\text{ref}}) + \Delta d^{(0)})) \tag{4}$$

where $\hat{c}^{(0)}$ and $\hat{d}^{(0)}$ denote the center and width of the proposals, respectively. Our approach utilizes the scaling of the center offsets with width d^{ref} and exp at the proposals to address the scale-invariant approach. The decoder then refines these initial locations of queries after the proposed adaptive query selection.

Adaptive Query Selection In a conventional two-stage approach in object detection, a fixed top-k selection method based on binary class predictions $p^{(0)}$ is typically employed. However, this static method may not be optimal for TAD, where the number and duration of action instances within videos vary significantly. In TAD, there's often a direct correlation between the length of a video and the number of action instances. Typically, longer videos contain more action instances, while shorter videos have fewer. This variation presents a challenge for the fixed selection method, which fails to adapt to video content characteristics.

Our method divides videos into sectors of a base length T_{sector}. We then perform top-k selection for each sector. This individual selection allows the detector to adapt to the video length, preventing the selection of only some parts when dealing with long videos. In practice, we redistribute any remaining timesteps to ensure that no sector at the end part of the video is smaller than T_{sector}. The total number of sectors, S, is calculated by dividing the number of the first layer's feature T_1 by T_{sector}, applying a floor function. For each sector s, where $s \in \{1, 2, \ldots, S\}$, we select the top-$k$ proposals from all levels based on the encoder's binary class scores within that sector. We define a subset of encoder output scores, P_s, for each sector, and then select the top-k proposals from this subset. The adaptive query selection is represented as follows:

$$\mathcal{Q} = \bigcup_{s=1}^{S} \left\{ (\hat{c}_{t,l}^{(0)}, \hat{d}_{t,l}^{(0)}) \mid (t,l) \in \text{indices of top-}K \text{ in } P_s \right\}, \tag{5}$$

where $\mathcal{Q} = \{(\hat{c}_q^{(0)}, \hat{d}_q^{(0)})\}_{q=1}^{N_q}$ is the aggregated set of selected queries. Here, K is the number of queries selected from each sector, and the total number of queries N_q equals the sum of the top-k proposals across all sectors, denoted as $N_q = \sum_{s=1}^{S} K$.

Time-Aligned Segment Refinement Existing query-based models [18, 26, 28] employ the *sigmoid* function to express normalized coordinates from the range 0 to 1. Moreover, TadTR [18] utilizes the refining step in the decoder of the transformer using the predicted center and width of each layer. The layer-wise segment refinement step in the normalized coordinate expression is defined as $\sigma(\sigma^{-1}(\hat{c}_q^{(n-1)}) + \Delta c_q^{(n)})$ and $\sigma(\sigma^{-1}(\hat{d}_q^{(n-1)}) + \Delta d_q^{(n)})$ for center prediction and width prediction, respectively. This refinement step restricts the values within a $[0, 1]$ but enables layer-by-layer updates. We reorganize the previous segment refinement step without the normalized expression.

In our approach, given selected queries $\mathcal{Q}$ are utilized for segment refinement. Formally, segment refinement of each layer is as follows:

$$\hat{c}_q^{(n)} = \hat{c}_q^{(n-1)} + \Delta c_q^{(n)} \cdot \hat{d}_q^{(n-1)}, \; n \in \{1, 2, \ldots, L_D\}, \quad (6)$$

$$\hat{d}_q^{(n)} = \exp\left(\ln(\hat{d}_q^{(n-1)}) + \Delta \hat{d}_q^{(n)}\right), \; n \in \{1, 2, \ldots, L_D\}, \quad (7)$$

where $\hat{c}_q^{(n)}$ and $\hat{d}_q^{(n)}$ represent the predicted outputs of the center point and width from the decoder, respectively, of n-th layer for each query. Similar to the encoder case, we utilize the scaling of the center offsets with width $d^{(n)}$ and exp function at the proposals to address the scale-invariant approach. The start $\hat{s}^{(L_D)}$ and end $\hat{e}^{(L_D)}$ timestamps for predictions are decoded as $\hat{c}^{(L_D)} - \hat{d}^{(L_D)}$ and $\hat{c}^{(L_D)} + \hat{d}^{(L_D)}$, respectively. Consequently, the final predicted proposals is defined as $\hat{\mathcal{A}} = (\hat{s}^{(L_D)}, \hat{e}^{(L_D)}, \hat{p}^{(L_D)})$, where L_D is the number of decoder layer, and $\hat{p}^{(L_D)}$ is the prediction of confidence score of last layer of decoder action class.

3.5. Training and Inference

Training We follow the standard bipartite matching loss [4]. The total loss $\mathcal{L}_{total}$ is defined as follows:

$$\mathcal{L}_{total}(\mathcal{A}, \hat{\mathcal{A}}) = \sum_{i=1}^{N_q} \mathcal{L}_{match}(\mathcal{A}_i, \hat{\mathcal{A}}_{\pi(i)}), \quad (8)$$

where $\mathcal{L}_{match}$ is the bipartite matching loss that incorporates both the classification probabilities and the distance between the ground truth and the predicted segments, and π represent permutation indices obtained by bipartite matching [12]. This cost function, $\mathcal{L}_{match}$, is a composite of the classification loss, and the regression loss. For the classification loss, we use the focal loss [17], which effectively addresses the class imbalance issue. For the regression loss, our model utilizes DIoU [36] and log-ratio distance for the width. DIoU evaluates the relative center distance and GIoU [24] once, while the log-ratio compares the widths relatively. Following the previous query-based approaches [4, 18, 37], we utilize the auxiliary decoding loss at every decoder layer for a more effective learning process. Furthermore, we do not apply any other losses without bipartite matching loss, such as actionness loss [18] or action classification enhancement loss [26]. More detailed descriptions of each loss are described in the supplementary materials.

Inference TE-TAD introduces a significant innovation by completely removing the need for common post-processing steps, such as NMS and temporal scaling. This is a direct consequence of our model's unique ability to work with an end-to-end approach and actual timeline values of the video, streamlining the inference process. The predictions from the final layer of the decoder $\hat{\mathcal{A}}$ are directly used.

4. Experiments

4.1. Setup

Datasets We conduct experiments on three datasets: THUMOS14 [10], ActivityNet v1.3 [8], and EpicKitchens [7]. THUMOS14 comprises 20 action classes with 200 and 213 untrimmed videos in the validation and test sets, containing 3,007 and 3,358 action instances, respectively. ActivityNet v1.3 are large-scale datasets with 200 action classes. They consist of 10,024 videos for training and 4,926 videos for validation, respectively. EpicKitchens, a first-person vision dataset, includes two sub-tasks: noun and verb. It contains 495 and 138 videos with 67,217 and 9,668 action instances for training and test, with 300 and 97 action classes for nouns and verbs, respectively. These datasets contain diverse actions and scenes, providing a rigorous evaluation setup for our method.

Evaluation Metric We follow the standard evaluation protocol for all datasets, utilizing mAP at different intersections over union (IoU) thresholds to evaluate TAD performance. The IoU thresholds for THUMOS14 and EpicKitchens are set at [0.3:0.7:0.1] and [0.1:0.5:0.1] respectively, while for ActivityNet v1.3, the results are reported at IoU threshold [0.5, 0.75, 0.95] with the average mAP computed at [0.5:0.95:0.05].

Implementation Details To ensure the clarity and focus of our main manuscript, detailed descriptions of the hyperparameters and experimental environments are provided in the supplementary materials.

Type	Method	NMS	mAP					
			0.3	0.4	0.5	0.6	0.7	Avg.
Anchor -based	BSN [15]*	✓	53.5	45.0	36.9	28.4	20.0	36.8
	BMN [16]*	✓	56.0	47.4	38.8	29.7	20.5	38.5
	BC-GNN [3]*	✓	57.1	49.1	40.4	31.2	23.1	40.2
	G-TAD [31]*	✓	54.5	47.6	40.3	30.8	23.4	39.3
	VSGN [34]*	✓	66.7	60.4	52.4	41.0	30.4	50.2
	TCANet [23]*	✓	60.6	53.2	44.6	36.8	26.7	44.3
Anchor -free	AFSD [14]	✓	67.3	62.4	55.5	43.7	31.1	52.0
	MENet [35]‡	✓	70.7	65.3	58.8	49.1	34.0	55.6
	TALLFormer [6]†	✓	76.0	-	63.2	-	34.5	59.2
	ActionFormer [32]	✓	82.1	77.8	71.0	59.4	43.9	66.8
	TriDet [27]	✓	**83.6**	**80.1**	**72.9**	**62.4**	**47.4**	**69.3**
Query -based	RTD-Net [28]	✓	68.3	62.3	51.9	38.8	23.7	49.0
	ReAct [26]	✓	69.2	65.0	57.1	47.8	35.6	55.0
	Self-DETR [11]	Δ	74.6	69.5	60.0	47.6	31.8	56.7
	TadTR [18]	Δ	74.8	69.1	60.1	46.6	32.8	56.7
	TE-TAD (Ours)	✗	81.7	76.6	69.5	59.3	44.8	66.4
	Ours w/ NMS	✓	**83.3**	**78.4**	**71.3**	**60.7**	**45.6**	**67.9**

Table 1. Performance comparison with state-of-the-art methods on THUMOS14. * and † denote TSN [30] and Swin-B [20] backbones, respectively. ‡ represents R(2+1)D [29]. Others employ I3D [5] backbone. The symbol Δ indicates partial adoption of NMS.

4.2. Main Results

THUMOS14 Table 1 provides a comparison with the state-of-the-art methods on THUMOS14. Our TE-TAD shows a

Type	Method	Feature	mAP			
			0.5	0.75	0.95	Avg.
Anchor -based	BSN [15]	TSN [30]	46.5	30.0	8.0	30.0
	BMN [16]	TSN [30]	50.1	34.8	8.3	33.9
	BC-GNN [3]	TSN [30]	50.6	34.8	9.4	34.3
	G-TAD [31]	TSN [30]	50.4	34.6	9.0	34.1
	VSGN [34]	TSN [30]	52.4	36.0	8.4	35.1
	TCANet [23]	TSN [30]	52.3	36.7	6.9	35.5
Anchor -free	AFSD [14]	I3D [5]	52.4	35.3	6.5	34.4
	TALLFormer [6]	Swin-B [20]	54.1	36.2	7.9	35.6
	ActionFormer [32]	R(2+1)D [29]	**54.7**	37.8	8.4	36.6
	TriDet [27]	R(2+1)D [29]	**54.7**	38.0	8.4	36.8
	MENet [35]	R(2+1)D [29]	**54.7**	**38.4**	**10.5**	**37.7**
Query -based	RTD-Net [28]	TSN [30]	47.2	30.7	8.6	30.8
	ReAct [26]	TSN [30]	49.6	33.0	8.6	32.6
	TadTR [18]	R(2+1)D [29]	53.6	37.5	**10.6**	36.8
	TE-TAD (Ours)	R(2+1)D [29]	54.0	**38.2**	**10.6**	37.0
	Ours w/ NMS	R(2+1)D [29]	**54.2**	38.1	**10.6**	**37.1**

Table 2. Comparison with state-of-the-art methods on ActivityNet v1.3.

significant margin of improvement over other query-based detectors, even without NMS. While TadTR achieves an average mAP of 56.7% with partial NMS utilization through their proposed cross window fusion (CWF) denoted by **Δ**, our method without NMS achieves a superior performance average mAP of 66.4%. This significant improvement indicates the length-invariant capability of our TE-TAD model because THUMOS14 contains extremely diverse lengths of videos. Furthermore, even compared to anchor-free detectors, our method demonstrates competitive performance.

ActivityNet v1.3 Following the conventional approach [18, 27, 32], the external classification score is used to evaluate ActivityNet v1.3. The pre-extracted classification scores are combined with predictions from binary detectors to obtain class labels. Table 2 presents a performance comparison of our TE-TAD with state-of-the-art approaches. While our TE-TAD method exhibits a slight improvement in query-based detectors on ActivityNet v1.3 compared to the significant gains shown in Table 1, this is reflective of the intrinsic characteristics of the dataset. ActivityNet v1.3 does not contain the diverse length of the video relative to THUMOS14. Despite this different condition, our approach still demonstrates an improvement on ActivityNet v1.3, showing performance improvement even though it does not align with the primary issues our TE-TAD aims to resolve.

EpicKitchens Table 3 shows the comparison with state-of-the-art methods on EpicKitchens. Our model achieves competitive performance without relying on NMS, indicating TE-TAD robustness in diverse and complex action detection scenarios. EpicKitchen contains an extremely diverse length of action instances, like THUMOS14. The results indicate the robustness of TE-TAD in handling a wide range of action lengths and complexities. The comparable performance of TE-TAD is meaningful in query-based approaches, a relatively less explored field than the more established anchor-free methods.

Task	Method	NMS	mAP					
			0.1	0.2	0.3	0.4	0.5	Avg.
Verb	BMN [16]	✓	10.8	8.8	8.4	7.1	5.6	8.4
	G-TAD [31]	✓	12.1	11.0	9.4	8.1	6.5	9.4
	ActionFormer [32]	✓	26.6	25.4	24.2	22.3	19.1	23.5
	ASL [25]	✓	27.9	-	25.5	-	19.8	24.6
	TriDet [27]	✓	**28.6**	**27.4**	**26.1**	**24.2**	**20.8**	**25.4**
	TE-TAD (Ours)	✗	27.0	25.9	24.6	22.9	**20.0**	24.1
	Ours w/ NMS	✓	**27.9**	**26.8**	**25.4**	**23.4**	**20.0**	**24.7**
Noun	BMN [16]	✓	10.3	8.3	6.2	4.5	3.4	6.5
	G-TAD [31]	✓	11.0	10.0	8.6	7.0	5.4	8.4
	ActionFormer [32]	✓	25.2	24.1	22.7	20.5	17.0	21.9
	ASL [25]	✓	26.0	-	23.4	-	17.7	22.6
	TriDet [27]	✓	**27.4**	**26.3**	**24.6**	**22.2**	**18.3**	**23.8**
	TE-TAD (Ours)	✗	26.0	24.8	23.2	20.8	**18.3**	22.6
	Ours w/ NMS	✓	**26.3**	**25.2**	**23.2**	**21.0**	18.2	**22.8**

Table 3. Comparison with state-of-the-art methods on EpicKitchens. All methods employ SlowFast [9] as a backbone.

Type	Method	NMS	mAP@AVG
Anchor-free	ActionFormer [32]	✗	43.2 (-23.6)
		✓	66.8
	TriDet [27]	✗	44.9 (-24.4)
		✓	69.3
Query-based	ReAct [26]	✗	19.8 (-35.7)
		✓	55.0
	TadTR [18]	✗	53.1 (-3.6)
		Δ	56.7
	TE-TAD (Ours)	✗	66.4 (-1.5)
		✓	67.9

Table 4. Effect of NMS on the mAP across various anchor-free and query-based methods on THUMOS14. The value in parentheses represents the decrease in mAP when NMS is not applied. The symbol **Δ** indicates partial adoption of NMS.

4.3. Further Analysis

Impact of NMS In Table 4, we evaluate how NMS influences the performance of various TAD methods on THUMOS14. NMS is particularly crucial for anchor-free detectors, which employ a one-to-many assignment strategy that leads to duplicated predictions for the same instance. Furthermore, even though ReAct [26] is a query-based approach, removing NMS at the ReAct significantly affects the performance. This is why ReAct adopts partial self-attention in the decoder called relational queries, which does not address whole queries. As shown in Table 6, our approach also drops the performance without NMS when removing the decoder's self-attention layer. This result indicates that addressing whole queries by the decoder's self-attention is crucial to preserving the set-prediction mechanism and full end-to-end modeling. Furthermore, our proposed method exhibits a minimal performance decrement of only -1.5 when NMS is excluded. This indicates that our approach effectively achieves full end-to-end modeling.

Component Contribution Analysis Table 5 shows the incremental impact of each key component in our TE-TAD on THUMOS14. The performance is measured without NMS. We conduct experiments based on full end-to-end TadTR, referred to as TadTR-∞ in Sec. 3.3. Starting with the TadTR-∞ baseline, incorporating multi-scale features and

Baseline	Multi-Scale	Two-stage [35]	TE	AQS	mAP@AVG
TadTR-34.1s w/ NMS					56.7
	✓				56.5
		✓			57.3
	✓	✓			57.0
TadTR-∞ w/o NMS					40.2
	✓				42.6
	✓	✓			43.3
			✓		59.5
		✓		✓	46.1
	✓	✓	✓		63.6
	✓	✓	✓	✓	66.4

Table 5. Analysis of contributions of each component on THUMOS14. The all-empty check is the denoted baseline.

	Encoder	Decoder		mAP@AVG	
	Self-attn.	Self-attn.	Cross-attn.	w/o NMS	w/ NMS
#1		✓	✓	61.2	63.8
#2	✓		✓	53.0	63.4
#3	✓	✓		0.1	0.2
#4	✓	✓	✓	66.1	67.7

Table 6. Analysis of the specific roles of self-attention and cross-attention layers in the encoder and decoder, and their impact with and without utilizing NMS.

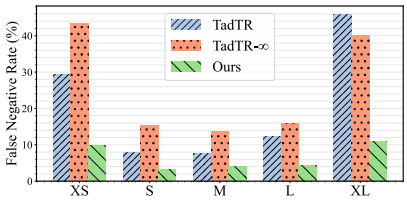


Figure 6. Comparison of false negative rates by instance lengths (XS, S, M, L, XL) on THUMOS14.

a two-stage approach shows slight improvements. Nonetheless, these adoptions of improved methods for query-based detectors do not reach the performance level of the original TadTR. However, including our time-aligned expression (TE) significantly improves performance, demonstrating the value of our time-aligned representations. Finally, incorporating the adaptive query selection (AQS) mechanism shows the highest performance.

Role of Each Attention To clarify the role of encoder and decoder architecture, we conduct experiments for removing the attention layers of the encoder and decoder. Table 6 shows the individual contributions of the encoder and decoder layers' attention. As shown in Row #1, removing self-attention in the encoder slightly decreases the detection performance, indicating the encoder's self-attention affects representational ability. Row #2 reveals that the decoder's self-attention is the core role of the set-prediction mechanism by showing the performance degradation without using NMS. Finally, Row #3 shows that removing the decoder's cross-attention does not work because only location information is provided to the decoder, which does not capture the content information without cross-attention.

False Negative Analysis To further compare with the baseline method, we evaluate the false negative rate on THUMOS14. Fig. 6 shows the false negative rates across varying action instance sizes: extra small (XS), small (S), medium (M), large (L), and extra large (XL) based on DETAD [1]. These results show that even though TadTR-∞ shows the worse overall performance in mAP, TadTR-∞ more captures the XL cases, indicating the shorter feature coverage cannot capture the long duration of instances. These results show that our method significantly reduces false negatives, particularly in XS and XL cases.

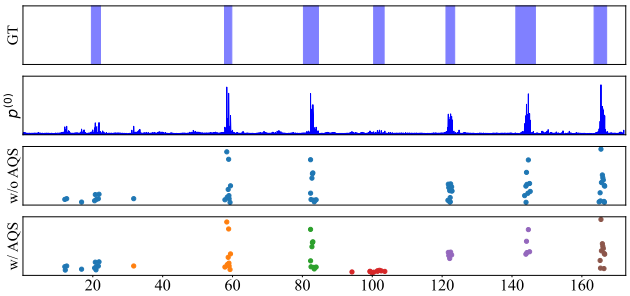


Figure 7. Comparison between query selection method. Ground truth (GT) is illustrated in the first row. The second row shows encoder prediction scores $p^{(0)}$. The third and fourth row visualizes the generated queries at the top 50 points and 10 points for each sector by AQS, respectively. Different colors in the fourth row represent proposals selected by different sectors.

Effetiveness of Adaptive Query Selection Fig. 7 compares the query selection method by denoting the center point c^{ref} of selected queries. Selected points both w/o AQS and w/ AQS are generated by the same encoder output scores. As illustrated in Fig. 7, the fixed top-k proposal method misses the fourth ground location as a foreground candidate. When a foreground proposal is too distant from actual action instances, it necessitates extensive refinement from the decoder. Refining the segments from missed candidates leads to an additional workload for the decoder layers. In contrast, our AQS method successfully captures the part missed by w/o AQS. By dividing the video into sectors and selecting queries within these local units, AQS dynamically adjusts the number of queries and more accurately detects foreground candidates.

5. Conclusion

In this paper, we propose a full end-to-end temporal action detection transformer that integrates time-aligned coordinate expression, called TE-TAD, which eliminates reliance on hand-crafted components such as the sliding window and NMS. By aligning coordinate expression with the actual video timeline, our model not only simplifies the detection process but also significantly enhances the performance of query-based detectors. Furthermore, our TE-TAD has a length-invariant property by combining the proposed time-aligned coordinate expression and adaptive query selection, showing the potential of query-based detectors.

Acknowledgement This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program (Korea University), No. 2021-0-02068, Artificial Intelligence Innovation Hub, and No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation).

References

[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *Proceedings of the European Conference on Computer Vision*, pages 256–272, 2018. 8

[2] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 3173–3183, 2021. 2

[3] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision*, pages 121–137, 2020. 1, 2, 6, 7

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020. 1, 3, 6

[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. 2, 3, 6, 7

[6] Feng Cheng and Gedas Bertasius. Tallformer: Temporal action localization with long-memory transformer. In *Proceedings of the European Conference on Computer Vision*, pages 503–521, 2022. 1, 2, 6, 7

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, and Will Price. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, 2020. 2, 6

[8] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 2, 6

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019. 3, 7

[10] 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. 2, 6

[11] Jihwan Kim, Miso Lee, and Jae-Pil Heo. Self-feedback detr for temporal action detection. In *ICCV*, pages 10286–10296, 2023. 6

[12] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6, 1

[13] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 3

[14] 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. 1, 2, 6, 7

[15] 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*, pages 3–19, 2018. 1, 2, 6, 7

[16] 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. 1, 2, 6, 7

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 6, 1

[18] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. End-to-end temporal action detection with transformer. *IEEE Transactions on Image Processing*, 2022. 1, 3, 4, 5, 6, 7

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[20] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 2, 6, 7

[21] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 344–353, 2019. 1, 2

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 1

[23] 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. 1, 2, 6, 7

[24] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 6, 1

[25] Jiayi Shao, Xiaohan Wang, Ruijie Quan, Junjun Zheng, Jiang Yang, and Yi Yang. Action sensitivity learning for temporal

action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2, 7

[26] Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. React: Temporal action detection with relational queries. In *Proceedings of the European Conference on Computer Vision*, pages 105–121, 2022. 1, 3, 5, 6, 7

[27] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. Tridet: Temporal action detection with relative boundary modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18857–18866, 2023. 1, 2, 4, 6, 7

[28] 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*, pages 13526–13535, 2021. 1, 3, 5, 6, 7

[29] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2, 6, 7

[30] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision*, pages 20–36, 2016. 6, 7

[31] 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. 1, 2, 6, 7

[32] Chenlin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 492–510, 2022. 1, 2, 4, 6, 7

[33] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *Proceedings of the European Conference on Computer Vision*, 2022. 4, 1

[34] 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. 1, 2, 6, 7

[35] Zixuan Zhao, Dongqi Wang, and Xu Zhao. Movement enhancement toward multi-scale video feature representation for temporal action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13555–13564, 2023. 2, 6, 7

[36] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12993–13000, 2020. 6, 1

[37] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020. 4, 6, 1