This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

DiGIT: Multi-Dilated Gated Encoder and Central-Adjacent Region Integrated Decoder for Temporal Action Detection Transformer

Ho-Joong Kim Yearang Lee Jung-Ho Hong Seong-Whan Lee*

Dept. of Artificial Intelligence, Korea University, Seoul, Korea

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

Abstract

In this paper, we examine a key limitation in query-based detectors for temporal action detection (TAD), which arises from their direct adaptation of originally designed architectures for object detection. Despite the effectiveness of the existing models, they struggle to fully address the unique challenges of TAD, such as the redundancy in multi-scale features and the limited ability to capture sufficient temporal context. To address these issues, we propose a multi-dilated gated encoder and central-adjacent region integrated decoder for temporal action detection transformer (DiGIT). Our approach replaces the existing encoder that consists of multi-scale deformable attention and feedforward network with our multi-dilated gated encoder. Our proposed encoder reduces the redundant information caused by multi-level features while maintaining the ability to capture fine-grained and long-range temporal information. Furthermore, we introduce a central-adjacent region integrated decoder that leverages a more comprehensive sampling strategy for deformable cross-attention to capture the essential information. Extensive experiments demonstrate that DiGIT achieves state-of-the-art performance on THUMOS14, ActivityNet v1.3, and HACS-Segment. Code is available at: https://github.com/Dotori-HJ/DiGIT

1. Introduction

Temporal action detection (TAD) is crucial for video understanding and supports a wide range of real-world applications, including video surveillance, summarization, and retrieval. TAD aims to detect action instances within untrimmed videos by identifying action classes along with their start and end times. The most existing TAD methods [15, 27, 36, 47] are a snippet-based approach, which utilizes pre-extracted features to address long durations. This approach does not require the computational cost for the backbone network at the detection stage, enabling the detector to address a comprehensive length of features at once. The existing methods can be divided into three approaches:

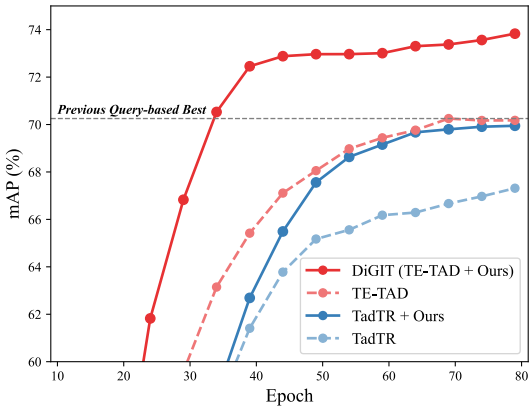


Figure 1. **Convergence curves with InternVideo2 [41] features on THUMOS14 [14].** Our method boosts the previous query-based detectors like TE-TAD [15] and TadTR [27].

anchor-based [3, 21, 22, 29, 43, 49], anchor-free [6, 8, 20, 34, 36, 44, 47], and query-based [15, 27, 35, 37, 53].

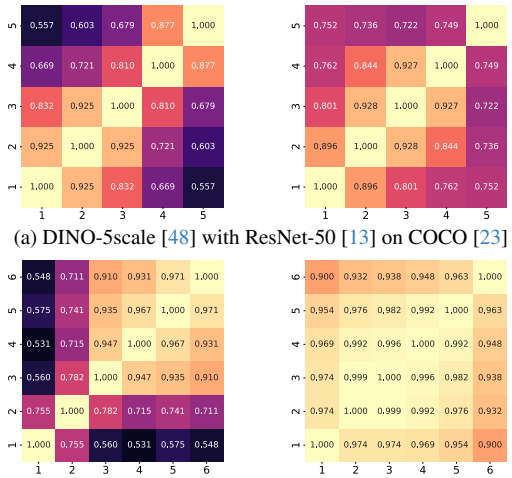
Query-based detectors, inspired by DETR [4], have attracted interest because of their potential to eliminate reliance on hand-crafted components, such as the sliding window and non-maximum suppression (NMS). This capability derives from their adoption of a set-prediction mechanism, which enables an end-to-end detection process through a one-to-one matching paradigm. Among them, TE-TAD [15] enables a full end-to-end detection process by reformulating the coordinate representation based on recent DETR-based network architectures, such as multi-scale deformable attention [52]. However, despite these advancements, query-based detectors still rely on originally designed architectures for object detection, which is not fully suited to addressing the unique challenges of TAD. We identify two limitations within the encoder and decoder structures of existing query-based TAD models. (1) In the encoder, simply utilizing single-scale [27, 53] feature or multi-scale [15] features fails to extract the meaningful features needed to capture the diverse temporal scale information and distinct feature representations. (2) In the decoder, the deformable cross-attention mechanism focuses on the central regions of reference points, overlooking surrounding areas essential for accurately detecting the action instances.

*Corresponding author

In the encoder, existing query-based detectors employ either single-scale [27, 53] or multi-scale [15] features, but both approaches have inherent limitations. The single-scale approach [27, 53] processes feature at a single resolution throughout the encoder and decoder, intuitively restricting the model to capture varying durations. The multi-scale approach [15] combines features across multiple resolutions, enhancing the ability of the model to detect different lengths of actions by aggregating broader contextual information. However, despite the advantage of the multi-scale approach, it struggles to capture distinct feature representations at each level, resulting in highly correlated features across scales. Fig. 2 illustrates this issue by comparing layer-wise CKA [31] similarities on the pre-encoder and post-encoder features between an object detection model (DINO [48]) and a TAD model (TE-TAD [15]). As shown in Fig. 2 on the left, the pre-encoder features of TE-TAD show high similarity among high-level features (3-6) compared to DINO. This is due to the repeated use of single convolutional projections, where the final-layer feature is downsampled to produce multi-scale features. These highly similar pre-encoder features of TE-TAD cause the multi-scale deformable attention to propagate redundant information across levels during encoding. Consequently, as shown in Fig. 2(b) on the right, the post-encoder features show high similarity across levels compared to DINO. This result suggests that utilizing multi-scale features from the initial stage leads to excessive redundancy.

In the decoder, deformable cross-attention relies on sampling points near the center of reference points, typically determined by multiplying the reference width by 0.5. However, this center-focused approach restricts the model from capturing the contextual information from the surrounding region, which is crucial for classifying action instances and determining their start and end boundaries. Fig. 3 shows the challenges of center-focused sampling through the example of actions like *LongJump* and *HighJump*, where both involve a similar running motion. When the model focuses only on the central motion (red box), identifying these two actions is challenging because the surrounding context (gray frames), such as the final landing motion, provides essential cues for identifying them. Furthermore, relying solely on the running motion makes it challenging to determine accurate start and end boundaries. These observations suggest that the center-focused sampling strategy is insufficient for capturing the full context of action instances.

In this paper, we propose a multi-dilated gated encoder and a central-adjacent region integrated decoder for temporal action detection transformer (DiGIT). First, we introduce a multi-dilated gated encoder (MDGE), which replaces the previous multi-scale deformable attention and feedforward network in the encoder. MDGE utilizes multi-dilated convolutions to capture diverse feature representations across multiple receptive fields, reducing redundant information in multi-scale features while preserving the benefits of utilizing multi-scale information. Additionally, we present a central-adjacent region integrated decoder (CAID), which combines both central- and adjacent-region sampling based on the deformable cross-attention mechanism. By incorporating these two types of information, CAID enables the detector to capture a complete contextual view for each detection query. Extensive experiments demonstrate that DiGIT achieves state-of-the-art performance on popular benchmarks, and our method is adaptable to existing query-based detection frameworks.



(a) DINO-5scale [48] with ResNet-50 [13] on COCO [23]

(b) TE-TAD [15] with InternVideo2 [41] on THUMOS14 [14]

Figure 2. **Layer-wise CKA similarity comparison between object detection and TAD.** The left and right sides are extracted from pre-encoder and post-encoder features, respectively. The 1–5 or 1–6 labels on each axis correspond to the number of multi-scale feature levels used in the respective models.

(a) *LongJump*

(b) *HighJump*

Figure 3. **Challenges of center-focused sampling in action distinction.** Each sequence shows seven evenly sampled frames across the action duration, using examples from THUMOS14 [14].

Our contributions are summarized as three-fold:

- We propose DiGIT that combines multi-dilated gated encoder and central-adjacent region integrated decoder to address the unique challenges of TAD.
- As shown in Fig. 1, our method consistently achieves faster convergence and improves performance when applied to existing query-based detectors.
- Our experiments demonstrate that our DiGIT achieves state-of-the-art performance on THUMOS14, ActivityNet v1.3, and HACS-Segment.

2. Related Work

Action Recognition Action recognition is a fundamental task in video analysis. It involves classifying video sequences into specific action categories. I3D [5] extends the inception network by incorporating 3D convolutions, while R(2+1)D [38] improves efficiency by decomposing 3D convolutions into separate 2D spatial and 1D temporal operations. TSP [2] introduces temporal channel shifting to model temporal dynamics effectively without adding computational overhead. VideoMAEv2 [40] leverages masked reconstruction pretraining method based on transformer architecture for robust video representation learning. InternVideo2 [41] leverages both large-scale training data and a highly scalable model, further enhancing video representation learning. These models are utilized in various downstream tasks like TAD as a feature extraction method.

Anchor-free Detector Anchor-free detectors [6, 8, 18, 20, 34, 36, 44, 47] provide flexibility in localizing action instances by utilizing an asymmetric modeling approach. ActionFormer [47] improves TAD performance by leveraging a transformer-based architecture to capture long-range dependencies in video data. TriDet [36] utilizes the trident prediction scheme and its proposed architecture. ActionMamba [6] improves the ActionFormer detector by utilizing Mamba [12] architecture at the temporal feature extraction. However, despite these advancements, anchor-free detectors still require hand-crafted components such as NMS to remove redundant proposals.

Query-based Detector Query-based detectors, drawing inspiration from DETR [4], employ a set-prediction mechanism that minimizes dependence on hand-crafted components, eliminating the necessity for NMS. RTD-Net [37] and ReAct [35] utilize query-based detection approaches; however, they do not fully address one-to-one matching in their architectures. TadTR [27] introduces cross-window fusion, applying NMS only to overlapping areas in sliding windows, which partially reduces dependency on hand-crafted components. DualDETR [53] divides the decoder into separate branches for instance-level and boundary-level decoding, whereas our CAID does not address split branch but unifies comprehensive range information within a single decoder. Both TadTR and DualDETR still rely on sliding windows, which require NMS to handle redundant areas, thereby limiting their applicability as a fully end-to-end detector. In contrast, TE-TAD [15] achieves a fully end-to-end approach for TAD by reformulating coordinate representation, removing the need for hand-crafted components like NMS and sliding windows. Building on TE-TAD, we propose DiGIT, which introduces MDGE, a multi-scale adapter, and CAID for the decoder. Although our DiGIT is primarily based on TE-TAD, our method is designed to be adaptable across various query-based detectors by replacing the encoder with MDGE and decoder with CAID.

3. Method

3.1. Preliminary

Let $X \in \mathbb{R}^{T_0 \times C}$ represents the video feature sequence extracted by the backbone network, where T_0 denotes the temporal length of the sequence, and C corresponds to the feature dimension. Each element of the sequence, denoted as $X = \{x_t\}_{t=1}^{T_0}$, is associated with a snippet at timestep t, with each snippet covering a few consecutive frames. These snippets are processed using a pre-trained backbone network such as I3D [5] or InternVideo2 [41]. Each video contains multiple action instances, each defined by start and end timestamps s and e, as well as its action class c. Formally, the set of action instances in a video is expressed as $\mathcal{A} = \{(s_n, e_n, c_n)\}_{n=1}^{N}$, where N denotes the total number of action instances, and s_n, e_n, and c_n represent the start time, end time, and action class of the n-th instance, respectively. The primary objective of TAD is to predict the set of action instances $\mathcal{A}$ for a given video.

The query-based detectors [15, 27, 53] employ N_q queries to detect action instances. The set of queries is represented as $\mathcal{Q}^{(0)} = \{F_q^{(0)}, (c_q^{(0)}, d_q^{(0)})\}_{q=1}^{N_q}$, where $F_q^{(0)}$ is the initial embedding of the q-th query, $c_q^{(0)}$ and $d_q^{(0)}$ denote the initial center and width reference point of the q-th query, respectively. These queries interact with the encoded features through deformable cross-attention layers in the decoder, where they are iteratively refined across each layer l:

$$\mathcal{Q}^{(l)} = \text{Decoder}^{(l)}(\mathcal{Q}^{(l-1)}) \quad l = 1, \ldots, L_D, \tag{1}$$

where L_D denotes the number of decoder layer. The final refined queries $Q^{(L_D)}$ are then used for final predictions $\hat{\mathcal{A}}$ obtained through the classification and regression heads.

3.2. DiGIT

In this part, we describe our method mainly based on TE-TAD, but our DiGIT can be applied to various query-based detectors by simply replacing the previous encoder and decoder architecture. The overall architecture of DiGIT is illustrated in Fig. 4. Our method mainly addresses three parts: (1) the multi-dilated gated encoder (MDGE), (2) the multi-scale adapter, which converts the single-scale feature into multi-scale features, providing diverse scale information for query selection and decoder, and (3) the central-adjacent region integrated decoder (CAID).

Embedding We project input features X using a single convolutional neural network to align them with the dimension of the transformer architecture.

$$Z^{(0)} = \text{LayerNorm}(\text{Conv}(X)), \tag{2}$$

where $Z^{(0)} \in \mathbb{R}^{D \times T_0}$ represented the embedded features of the detector. Here, D denotes the width of the encoder and decoder. In contrast to TE-TAD, we do not address the multi-scale features at the initial and encoding stages.

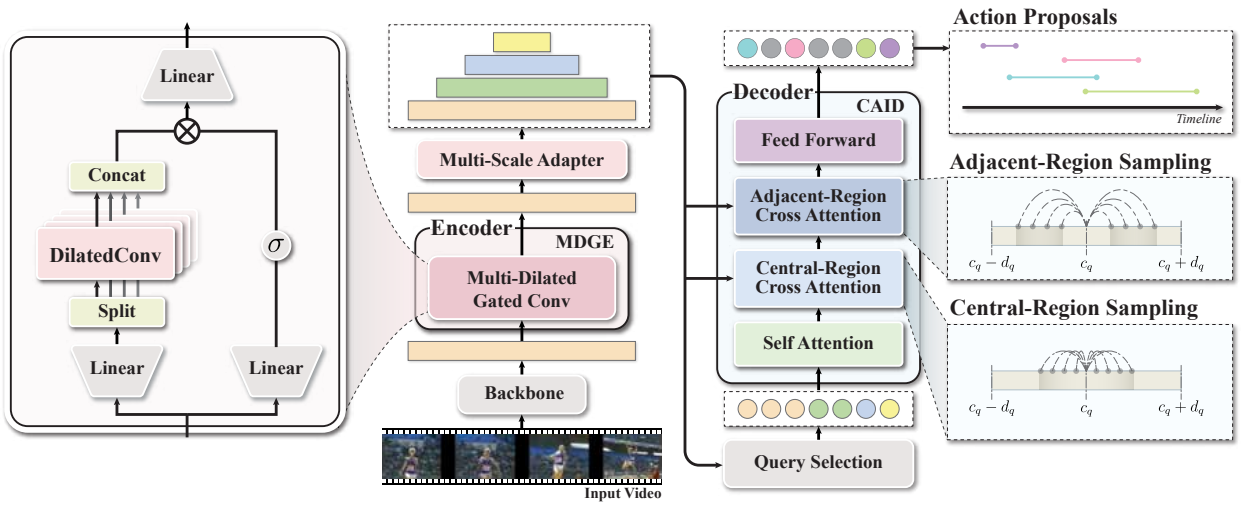


Figure 4. **Overview of DiGIT.** Our model processes video features through MDGE to capture distinct feature representations utilizing various receptive fields. Subsequently, CAID captures both central and adjacent region information, enhancing action boundary regression and classification. For simplicity, residual connection and layer normalization are omitted.

Multi-Dilated Gated Encoder (MDGE) As discussed in Sec. 1, both single-scale and multi-scale approaches have inherent limitations. Single-scale methods struggle to capture long temporal dependencies. Conversely, multi-scale methods can capture a broader range of temporal scales but cause highly correlated features across each scale.

To address the limitations of both approaches, we introduce MDGE that replaces the previous multi-scale deformable encoder structure. Inspired by a previous work [7], MDGE applies multi-dilated convolutions to capture diverse receptive fields within a single encoder. This approach enables the model to extract short-term and long-term temporal features without relying on a multi-scale structure. Furthermore, inspired by gating mechanism [46], MDGE selectively filters out redundant information, retaining only the most relevant features across the different receptive fields. In the following, we describe the detailed structure of MDGE, which is composed of L_E multi-dilated gated convolution layers.

At each encoder layer l, the input features $Z^{(l-1)}$ from the previous layer are first projected into two paths:

$$Z^{(l)}_{\text{conv}} = \text{Linear}(Z^{(l-1)}), \quad Z^{(l)}_{\text{gate}} = \text{Linear}(Z^{(l-1)}), \quad (3)$$

where $Z^{(l)}_{\text{conv}} \in \mathbb{R}^{D_h \times T_0}$ and $Z^{(l)}_{\text{gate}} \in \mathbb{R}^{D_h \times T_0}$. Here, D_h represents the hidden dimension of the feedforward network within the transformer architecture. Instead of utilizing the feedforward network in the encoder, we expand the feature dimension before applying the dilated convolution. This approach retains a similar parameter to a single transformer layer that consists of an attention layer and a feedforward network. Each transformed features, $Z^{(l)}_{\text{conv}}$ and $Z^{(l)}_{\text{gate}}$, are then processed in separate paths independently for multi-dilated convolution and gating mechanism, respectively.

The transformed features for multi-dilated convolution $Z^{(l)}_{\text{conv}}$ are split along the channel dimension into N_d equal subsets, denoted as $Z^{(l,i)}_{\text{conv}} \in \mathbb{R}^{(D_h/N_d) \times T_0}$, where N_d is the number of parallel dilated convolutions. Each subset is processed by a dilated convolution with a different dilation rate, increasing from 1 up to N_d. The output features for each subset are expressed as follows:

$$Z^{(l,i)}_{\text{dilated}} = \text{DilatedConv}_{(l,d_i)}(Z^{(l,i)}_{\text{conv}}) \quad i = 1, \ldots, N_d, \quad (4)$$

where d_i is the dilation rate for the i-th convolution, set as $d_i = i$. The increasing dilation rates provide varying receptive field sizes, allowing the model to capture both short- and long-term temporal features simultaneously. Our encoder has two main hyperparameters: the number of dilated convolutions N_d and the kernel size.

Subsequently, the outputs of the dilated convolutions $Z^{(l,i)}_{\text{dilated}}$ are concatenated along the channel dimension:

$$Z^{(l)}_{\text{concat}} = \text{Concat}(Z^{(l,1)}_{\text{dilated}}, \ldots, Z^{(l,N_d)}_{\text{dilated}}). \quad (5)$$

Following this concatenation, we apply a gating mechanism to selectively retain relevant features:

$$Z^{(l)} = Z^{(l)}_{\text{concat}} \odot \sigma(Z^{(l)}_{\text{gate}}), \quad (6)$$

where σ is an activation function and $\odot$ denotes element-wise multiplication. Specifically, we use the SiLU activation function for the gate activation.

Consequently, the encoding process across L_E layers of MDGE can be summarized as follows:

$$Z^{(l)} = \text{MDGE}^{(l)}(Z^{(l-1)}) \quad l = 1, \ldots, L_E. \quad (7)$$

Our MDGE design enables the encoder to capture diverse temporal relations by leveraging dilated convolutions and a gating mechanism without requiring explicit multi-scale features at each layer.

Multi-Scale Adapter & Query Selection TE-TAD [15] employs a two-stage approach with multi-scale features for query selection. To take advantage of the two-stage approach and utilize multi-scale features for query selection, we introduce a multi-scale adapter that converts the single-scale feature to multi-scale features. Our multi-scale adapter utilizes the encoder output feature $Z^{(L_E)}$ to generate multi-scale representations. Specifically, $Z^{(L_E)}$ is progressively downsampled to produce a set of features at multiple levels, with each subsequent level reduced to half the temporal length of the previous one. We denote the L levels of multi-scale features as follows:

$$F^{(l)} = \text{DownSample}(Z^{(L_E)}) + E^{(l)} \quad l = 1, \dots, L, \quad (8)$$

where each $F^{(l)} \in \mathbb{R}^{T_l \times D}$ represents the resized feature at level l, with T_l being half the length of the previous level. To enable the decoder to distinguish between these multi-scale levels, we add a level-specific embedding $E^{(l)} \in \mathbb{R}^{1 \times D}$, applied consistently across all time steps. These multi-scale features are then utilized for the query selection process. Unlike the adaptive query selection (AQS) in TE-TAD [15] that enforces a strict uniform sampling of queries across the video, we apply a top-k selection approach based on binary classification scores from the encoder.

Subsequently, we utilize the top-k indices to retrieve both the query embeddings and the corresponding reference points (c_q, d_q) based on a time-aligned query generation method [15] that assigns temporal center points c_q and widths d_q, aligning them with their respective positions in the video. Additionally, the query embedding for each selected query is obtained by linearly projecting the corresponding top-k multi-scale features. The input embedding of a decoder is denoted as follows:

$$F_q^{(0)} = \text{LayerNorm}(\text{Linear}(F_{\text{topk}})), \quad (9)$$

where F_{topk} denotes the top-k selected features.

Central-Adjacent Region Integrated Decoder (CAID) The previous query-based methods [15, 27] apply temporal deformable cross-attention that applies the center-focused sampling strategy. As discussed in Sec. 1, the center-focused sampling does not provide sufficient information for detecting the action instances. To address this issue, we introduce CAID, which combines central- and adjacent-region cross-attention. In standard deformable cross-attention, features are sampled around each reference point, defined by its center c_q and duration d_q. The sampling offset Δp_{mqk} for each head m and sampling point k is computed as:

$$\Delta p_{mqk} = \text{Linear}(F_q) = WF_q + b, \quad (10)$$

where W and b are learnable parameters for obtaining the sampling offset by linear projection. Generally, W is initialized to zero and b within the range $[-1, 1]$. Using the computed offset Δp_{mqk}, sampling points p_{mqk} are determined based on the reference points as:

$$p_{mqk} = c_q + 0.5 w_q \Delta p_{mqk}, \quad (11)$$

where the factor $0.5 w_q$ ensures the sampling points are initially positioned close to the center of each reference point. In our approach, we do not change how to obtain the sampling points when addressing central- and adjacent-region sampling. We change the initialization method for the bias value b, which determines the initial sampling points.

As shown in Fig. 4, our CAID contains two cross-attention layers sequentially: central-region cross-attention and adjacent-region cross-attention. For central-region cross-attention, the initial bias of sampling offsets b are initialized uniformly within the range $[-1, 1]$, which is identical to the previous methods [15, 19, 25, 27, 48, 52]. As Eq. (11), this initialization constrains the initial bias for sampling points b within $[-0.5, 0.5]$, focusing on the central region of the reference points. For adjacent-region cross-attention, the bias of sampling offsets b are adjusted to capture surrounded points of central-region cross-attention. Specifically, half of the sampling offsets are initialized within the range $[-1.5, -0.5]$ to focus on the left, while the other half are initialized within the range $[0.5, 1.5]$ to focus on the right. This adjustment, combined with Eq. (11), constraints in the initial sampling points p_{mqk} within $[-0.75, -0.25]$ and $[0.25, 0.75]$. Overall, we sequentially apply self-attention, central-region cross-attention, adjacent-region cross-attention, and feedforward network across all L_D layers, as shown in Fig. 4.

3.3. Training and Inference

Training Following the previous works [15, 27], we use the bipartite matching loss [4]. The total loss $\mathcal{L}_{total}$ is defined as follows:

$$\mathcal{L}_{total}(\mathcal{A}, \hat{\mathcal{A}}) = \sum_{i=1}^{N_q} \mathcal{L}_{match}(\mathcal{A}_i, \hat{\mathcal{A}}_{\pi(i)}), \quad (12)$$

where $\mathcal{L}_{match}$ denotes the bipartite matching loss, which considers both classification and regression loss between ground truth $\mathcal{A}_i$ and predicted action instances $\hat{\mathcal{A}}_{\pi(i)}$ from the last layer of the decoder. The permutation indices $\pi(i)$ are obtained through bipartite matching [17]. This cost function $\mathcal{L}_{match}$ is a composite of the classification, and regression loss. We use focal loss [24] for the classification loss to effectively manage class imbalance. For the regression loss, our method incorporates GIoU [33] and a log-ratio distance loss [15].

Inference Following the previous work [15] that removes the need for post-processing steps, the predictions of DiGIT from the final layer of the decoder $\hat{\mathcal{A}}$ are directly used. For a fair comparison, we report both raw prediction results and NMS applied results.

Training Type	Head Type	Method	Feature	THUMOS14						ActivityNet v1.3			
				0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
Full	-	AFSD [20]	I3D [5]	67.3	62.4	55.5	43.7	31.1	52.0	52.4	35.3	6.5	34.4
		TALLFormer [8]	Swin-B [28]	76.0	-	63.2	-	34.5	59.2	54.1	36.2	7.9	35.6
		ViT-TAD [45]	ViT-B [9]	85.1	80.9	74.2	61.8	45.4	69.5	55.8	38.5	8.8	37.4
		AdaTAD [26]	VideoMAEv2-g [40]	**89.7**	**86.7**	**80.9**	**71.0**	**56.1**	**76.9**	**61.7**	**43.4**	**10.9**	**41.9**
Head only	Anchor -free	TriDet [36]	I3D [5] / R(2+1)D [38]	83.6	80.1	72.9	62.4	47.4	69.3	54.7	38.0	8.4	36.8
		DyFADet [44]	I3D [5] / R(2+1)D [38]	84.0	80.1	72.7	61.1	47.9	69.2	58.1	39.6	8.4	38.5
		ActionFormer [47]	I3D [5] / R(2+1)D [38]	82.1	77.8	71.0	59.4	43.9	66.8	54.7	37.8	8.4	36.6
		ActionFormer [47]	InternVideo2 [41]	82.3	81.9	75.1	65.8	50.3	71.9	61.5	**44.6**	**12.7**	41.2
		ActionMamba [6]	InternVideo2 [41]	**86.9**	**83.1**	**76.9**	**65.9**	**50.8**	**72.7**	**62.4**	43.5	10.2	**42.0**
	Query -based	RTD-Net [37]	I3D [5] / TSN [39]	68.3	62.3	51.9	38.8	23.7	49.0	47.2	30.7	8.6	30.8
		ReAct [35]	I3D [5] / TSN [39]	69.2	65.0	57.1	47.8	35.6	55.0	49.6	33.0	8.6	32.6
		Self-DETR [16]	I3D [5]	74.6	69.5	60.0	47.6	31.8	56.7	52.3	33.7	8.4	33.8
		TadTR [27]	I3D [5] / R(2+1)D [38]	74.8	69.1	60.1	46.6	32.8	56.7	53.6	37.5	10.6	36.8
		DualDETR [53]	I3D [5]	82.9	78.0	70.4	58.5	44.4	66.8	52.6	35.0	7.8	34.3
		TE-TAD [15]	I3D [5] / R(2+1)D [38]	83.3	78.4	71.3	60.7	45.6	67.9	54.2	38.1	10.6	37.1
		DiGIT†	I3D [5] / R(2+1)D [38]	81.6	77.7	70.3	60.5	48.4	67.7	54.3	38.4	10.6	37.2
		DiGIT	I3D [5] / R(2+1)D [38]	83.6	79.6	71.9	61.5	48.6	69.0	54.4	38.2	10.7	37.3
		TadTR [27]	InternVideo2 [41]	84.8	79.3	70.4	58.2	43.8	67.3	57.1	38.8	11.0	38.2
		TadTR [27] + Ours	InternVideo2 [41]	86.1	81.8	73.7	61.7	46.3	69.9	60.2	41.0	11.2	40.5
		TE-TAD [15]	InternVideo2 [41]	84.3	81.1	73.7	62.6	49.5	70.3	61.3	41.8	10.9	41.1
		DiGIT†	InternVideo2 [41]	85.7	82.3	75.6	65.6	51.4	72.1	58.9	43.4	11.4	41.3
		DiGIT	InternVideo2 [41]	**87.6**	**84.2**	**77.6**	**67.3**	**52.5**	**73.8**	**62.0**	**43.1**	**11.3**	**42.0**

Table 1. **Performance comparison with state-of-the-art methods on THUMOS14 and ActivityNet v1.3.** In cases marked with †, our method does not utilize NMS. For TadTR + Ours, we additionally apply our MDGE and CAID on TadTR.

4. Experiments

4.1. Setup

Datasets We conduct experiments on three datasets: THUMOS14 [14], ActivityNet v1.3 [10], and HACS-Segment [50]. THUMOS14 consists of 20 action classes with 200 validation and 213 test videos, containing 3,007 and 3,358 action instances, respectively. ActivityNet v1.3 is a large-scale dataset with 200 action classes, including 10,024 videos for training and 4,926 videos for validation. HACS-Segment is another large-scale TAD dataset with extensive annotations, covering 200 activity classes similar to ActivityNet v1.3. It provides 37,613 videos for training and 5,981 videos for validation. These datasets provide a rigorous evaluation environment for our method, containing diverse actions and scenes.

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 are set at [0.3:0.7:0.1], while for ActivityNet v1.3 and HACS-Segment, 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 We describe the implementation details for each dataset in the Supplementary Sec. A.

4.2. Main Results

THUMOS14 Table 1 contains a comparison with the state-of-the-art methods on THUMOS14. Our DiGIT shows consistent improvements over TadTR [27] and TE-TAD [15] on both I3D [5] and InternVideo2 [41] features. Even without applying NMS, our method outperforms the existing query-based detectors. Furthermore, our model outperforms the existing snippet-based head-only training methods, and our DiGIT shows a comparable performance even compared to the full training method.

ActivityNet v1.3 Following the conventional approach [15, 27, 36, 47], the external classification score is used to evaluate ActivityNet v1.3. The pre-extracted classification scores are combined with class-agnostic predictions from a binary detector to obtain class labels. For R(2+1)D [38] and InternVideo2 [41] features, classification results from CUHK [42] and InternVideo2 [41] are incorporated to obtain class scores, respectively. As demonstrated in Table 1, DiGIT achieves consistent improvements over TadTR and TE-TAD on ActivityNet v1.3. Furthermore, DiGIT demonstrates competitive performance compared to other types of state-of-the-art methods, demonstrating effectiveness across various datasets and feature extractors.

HACS-Segment Table 2 presents a comparison of our model with state-of-the-art methods on HACS-Segment. Our model achieves significant improvements over previous methods, establishing a new state-of-the-art performance, including mAP at higher IoU thresholds of 0.75 and 0.95. The stronger performance at the 0.95 IoU threshold indicates that DiGIT excels in precise action localization. Moreover, the superior performance of DiGIT on this larger dataset, compared to THUMOS14 and ActivityNet v1.3, demonstrates its scalability and robustness across varying data sizes and complexities.

Head Type	Method	Feature	mAP			
			0.5	0.75	0.95	Avg.
Anchor -based	SSN [51]	I3D [5]	28.8	18.8	5.3	19.0
	G-TAD [43]	I3D [5]	41.1	27.6	8.3	27.5
	BMN [22]	SlowFast [11]	52.5	36.4	10.4	35.8
	TCANet [32]	SlowFast [11]	54.1	37.2	11.3	36.8
Anchor -free	TALLFormer [8]	Swin-B [28]	55.0	36.1	11.8	36.5
	TriDet [36]	I3D [5]	54.5	36.8	11.5	36.8
	TriDet [36]	SlowFast [11]	56.7	39.3	11.7	38.6
	TriDet [36]	VideoMAEv2-g [40]	62.4	44.1	13.1	43.1
	DyFADet [44]	SlowFast [11]	57.8	39.8	11.8	39.2
	DyFADet [44]	VideoMAEv2-g [40]	64.0	44.8	14.1	44.3
	ActionFormer [47]	InternVideo2 [41]	62.6	44.6	12.7	43.3
	ActionMamba [6]	InternVideo2 [41]	**64.0**	**45.7**	**13.3**	**44.6**
Query -based	TadTR [27]	I3D [5]	47.1	32.1	10.9	32.1
	TadTR [27]	InternVideo2 [41]	54.2	38.8	12.8	37.8
	TadTR [27] + Ours	InternVideo2 [41]	55.0	40.0	13.8	39.0
	TE-TAD [15]	InternVideo2 [41]	60.4	45.6	16.5	44.1
	DiGIT†	InternVideo2 [41]	61.1	47.5	**17.8**	45.5
	DiGIT	InternVideo2 [41]	**62.4**	**47.9**	17.6	**45.9**

Table 2. **Performance comparison with state-of-the-art methods on HACS-Segment.** In cases marked with †, our method does not utilize NMS. For TadTR + Ours, we additionally apply our MDGE and CAID on TadTR.

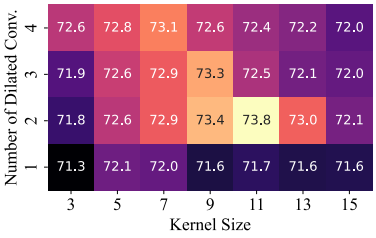


Figure 5. **Ablation study on MDGE with InternVideo2 features on THUMOS14.** The heatmap shows mAP values for different combinations of kernel size and number of dilated convolutions.

4.3. Further Analysis

Ablation Study on MDGE Fig. 5 shows an ablation study on hyperparameters of MDGE, specifically the impact of kernel size and the number of dilated convolution layers N_d on mAP performance. Excluding the configuration with a single dilated convolution that consistently yields lower mAP values, configurations with multiple dilated layers demonstrate improved performance. These results indicate that addressing diverse scale information is crucial for the encoder by utilizing our multi-dilated convolution that enables the model to capture diverse temporal relations. Furthermore, compared to the TE-TAD [15] baseline of 70.4 in Table 4, our MDGE consistently improves performance regardless of hyperparameter variations, demonstrating robustness across different configurations.

Ablation Study on CAID Table 3 demonstrates the effects of integrating the proposed adjacent-region cross-attention (ACA) into the decoder operation sequence. We evaluate several configurations, including variations of central-region cross-attention (CCA) with expanded initialization ranges (1.5× and 2.0×). CCA (1.5×) and CCA (2.0×) (Rows #2 and #3) show that simply increasing the initialization range of CCA to cover a broader area, similar range to CAID, does not show the improved performance compared to the original setting (Row #1). This indicates that simply expanding the initialization range does not benefit the detector. Moreover, when using ACA alone (Row #4), we observe a significant decrease in performance, underscoring the importance of balancing adjacent-region information with central-region information. Configurations that use two successive CCA layers (Row #5) show only marginal improvements, suggesting that additional central-region cross-attention alone does not significantly enhance performance. The highest mAP score is achieved by the sequence of SA → CCA → ACA → FFN, which combines central-region cross-attention and adjacent-region cross-attention, which is our CAID (Row #6).

	Decoder Sequece	mAP@AVG
#1	SA → CCA → FFN	72.5
#2	SA → CCA (1.5×) → FFN	72.6
#3	SA → CCA (2.0×) → FFN	71.8
#4	SA → ACA → FFN	69.2
#5	SA → CCA → CCA → FFN	72.7
#6	SA → CCA → ACA → FFN	**73.8**
#7	SA → ACA → CCA → FFN	73.2

Table 3. **Analysis of decoder operation sequences using InternVideo2 features on THUMOS14.** SA refers to the self-attention layer. CCA denotes central-region cross-attention (as used in previous works such as [15, 27]). CCA (1.5× and 2.0×) indicates an expansion of the initial sampling range of central-region cross-attention by 1.5 and 2.0 times, respectively, to cover a broader area around the reference points. ACA refers to adjacent-region cross-attention, and FFN represents the feedforward network.

Baseline	Enc.	Dec.	MDGE	CAID	mAP			
					0.3	0.5	0.7	Avg.
TadTR [27]	S	S			84.8	70.4	43.8	67.3
			✓		84.9	71.9	44.7	68.4
				✓	84.4	70.8	45.4	67.9
			✓	✓	86.1	73.7	46.3	69.9
TE-TAD [15]	M	M			84.3	73.7	49.5	70.3
				✓	85.2	74.6	49.9	70.8
	S	M			85.4	73.8	49.1	70.4
			✓		87.0	75.8	51.9	72.5
				✓	85.5	74.6	51.0	71.4
			✓	✓	**87.6**	**77.6**	**52.5**	**73.8**

Table 4. **Ablation study on the contributions of each component using InternVideo2 features on THUMOS14.** The first row for TadTR [27] and TE-TAD [15] represents the baseline performance. Enc. and Dec. refer to how scale information is handled in the encoder and decoder. S: single-scale. M: multi-scale.

Component Contribution Analysis Table 4 provides an analysis of the performance contributions from each proposed component, evaluated using InternVideo2 [41] features on THUMOS14 [14]. The first row for each method shows the baseline performance without our proposed enhancements, allowing for a direct comparison with subsequent configurations. Notably, converting multi-scale en-

Baseline	AQS [15]	mAP			
		0.3	0.5	0.7	Avg.
TE-TAD [15]		85.6	73.7	47.6	70.3
	✓	82.5	71.8	48.4	68.7
DiGIT		85.8	76.4	51.5	72.2
	✓	**87.6**	**77.6**	**52.5**	**73.8**

Table 5. **Ablation study on the query selection method using InternVideo2 features on THUMOS14.**

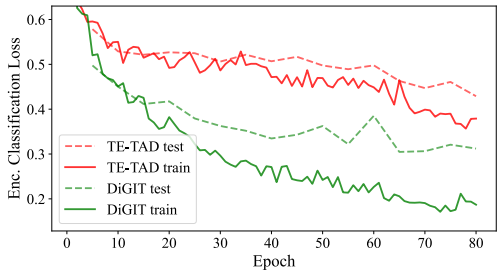


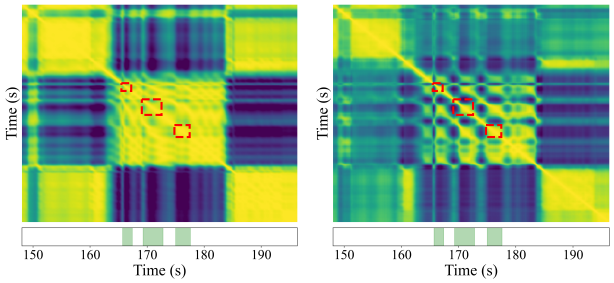
Figure 6. **Comparison of training and testing loss for the encoder classification matching loss using InternVideo2 features on THUMOS14.** Our method significantly accelerates the convergence of the encoder.

coding to single-scale encoding alone in TE-TAD [15] results in minimal performance change (70.3 → 70.4). This result aligns with our motivation, as shown in Fig. 2, which is that the previous multi-scale features contain redundant information. Furthermore, both MDGE and CAID show consistent improvements in both TadTR and TE-TAD, demonstrating the robustness and effectiveness of our approach across different baseline architectures.

Effect of MDGE In TE-TAD [15], adaptive query selection (AQS) is employed to sample initial queries across the entire video sequence uniformly. This process enforces a strict uniform selection of queries to prevent queries from being overly concentrated in certain areas, as might happen with a simple top-k selection. However, our observations suggest that this strict uniform condition is unnecessary when using a well-trained encoder, as provided by our MDGE. As shown in Table 5, removing AQS results in higher mAP scores for DiGIT, whereas the performance of TE-TAD declines without AQS. Furthermore, as illustrated in Fig. 6, DiGIT exhibits faster convergence and consistently lower loss values. These results indicate that our MDGE enhances the representational ability of the encoder and can reduce the heuristic part of the query selection, which positively impactsoverall detection performance.

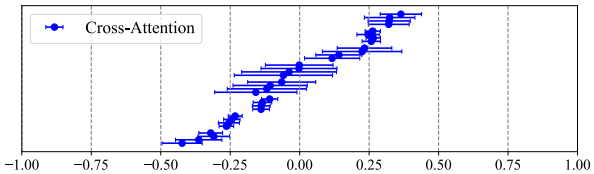
4.4. Qualitative Results

Visualization of Cosine Similarity on Encoder Output Features Fig. 7 presents a comparison of cosine similarity matrices for encoder output features between TE-TAD and DiGIT. Our DiGIT shows improved feature discriminability, indicating that MDGE captures distinct temporal patterns more effectively compared to TE-TAD [15].

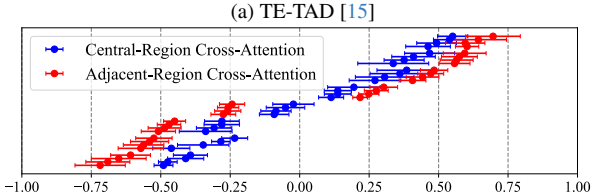


(a) TE-TAD [15] (b) DiGIT

Figure 7. **Comparison of cosine similarity on encoder output features between TE-TAD and DiGIT.** Top: represents cosine similarity, with red boxes indicating regions of similarity among features within the ground truth. Bottom: displays the ground truth action timeline for reference. Sample taken from THUMOS14.



(a) TE-TAD [15]



(b) DiGIT

Figure 8. **Visualization of sampling offsets in cross-attention layers on THUMOS14.** Each point denotes the mean value across the dataset, with error bars indicating the standard deviation.

Visualization of Sampling Offsets Fig. 8 shows the distribution of sampling offsets relative to the center and width of each action query. The values 0, -1, and 1 correspond to the center c_q, the start boundary $c_q - d_q$, and the end boundary $c_q + d_q$, respectively. This visualization shows that our method gathers information from diverse sampling points, covering both central and adjacent regions. Furthermore, while deformable attention allows learnable offsets, they remain close to their initial points, indicating the importance of the initial value of b in Eq. (10).

5. Conclusion

In this paper, we propose a multi-dilated gated encoder and central-adjacent region integrated decoder for temporal action detection transformer (DiGIT). MDGE offers diverse receptive fields while maintaining a single-scale encoding structure by utilizing multi-dilated convolutions. CAID provides essential information to precisely detect action instances by focusing on both the central- and adjacent- regions of action instances. Extensive experiments demonstrate that DiGIT outperforms the previous query-based methods. Furthermore, our method consistently improves when integrated with the existing query-based detectors.

Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant, funded by the Korea government (MSIT) (No. RS-2019-II190079, Artificial Intelligence Graduate School Program (Korea University), No. RS-2024-00457882, AI Research Hub Project, and No. RS-2022-II220984, 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. 12

[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. 3, 12

[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

[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, 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. 3, 6, 7, 12

[6] Guo Chen, Yifei Huang, Jilan Xu, Baoqi Pei, Zhe Chen, Zhiqi Li, Jiahao Wang, Kunchang Li, Tong Lu, and Limin Wang. Video mamba suite: State space model as a versatile alternative for video understanding. *arXiv preprint arXiv:2403.09626*, 2024. 1, 3, 6, 7, 12, 13

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017. 4

[8] 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, 3, 6, 7

[9] 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 *International Conference on Learning Representations*, 2021. 6

[10] 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. 6, 12, 14

[11] 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. 7

[12] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[14] 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. 1, 2, 6, 7, 12, 14

[15] Ho-Joong Kim, Jung-Ho Hong, Heejo Kong, and Seong-Whan Lee. TE-TAD: Towards full end-to-end temporal action detection via time-aligned coordinate expression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18837–18846, 2024. 1, 2, 3, 5, 6, 7, 8, 12, 13

[16] Jihwan Kim, Miso Lee, and Jae-Pil Heo. Self-feedback DETR for temporal action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10286–10296, 2023. 6

[17] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2 (1-2):83–97, 1955. 5

[18] Yearang Lee, Ho-Joong Kim, and Seong-Whan Lee. Text-infused attention and foreground-aware modeling for zero-shot temporal action detection. In *Advances in Neural Information Processing Systems*, 2024. 3

[19] 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. 5

[20] 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, 3, 6

[21] 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

[22] 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, 7

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014. 2

[24] 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. 5, 12, 14

[25] 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 *International Conference on Learning Representations*, 2022. 5

[26] Shuming Liu, Chen-Lin Zhang, Chen Zhao, and Bernard Ghanem. End-to-end temporal action detection with 1b parameters across 1000 frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18591–18601, 2024. 6

[27] 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, 2, 3, 5, 6, 7, 12, 13

[28] 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. 6, 7

[29] 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

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 12, 14

[31] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2021. 2

[32] 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. 7

[33] 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. 5, 12, 14

[34] 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. 1, 3

[35] 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, 6

[36] 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, 3, 6, 7

[37] 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, 6

[38] 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. 3, 6, 12

[39] 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

[40] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560, 2023. 3, 6, 7

[41] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Chenting Wang, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. InternVideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*, 2024. 1, 2, 3, 6, 7, 12

[42] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang. Cuhk ethz siat submission to activitynet challenge 2016. *arXiv preprint arXiv:1608.00797*, 2016. 6

[43] 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, 7

[44] Le Yang, Ziwei Zheng, Yizeng Han, Hao Cheng, Shiji Song, Gao Huang, and Fan Li. DyFaDet: Dynamic feature aggregation for temporal action detection. In *Proceedings of the European Conference on Computer Vision*, pages 305–322, 2024. 1, 3, 6, 7

[45] Min Yang, Huan Gao, Ping Guo, and Limin Wang. Adapting short-term transformers for action detection in untrimmed videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18570–18579, 2024. 6

[46] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019. 4

[47] 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, 3, 6, 7, 12, 13

[48] 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. 2, 5

[49] 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

[50] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. HACS: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8668–8678, 2019. 6, 12, 14

[51] 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. 7

[52] 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. 1, 5

[53] Yuhan Zhu, Guozhen Zhang, Jing Tan, Gangshan Wu, and Limin Wang. Dual DETRs for multi-label temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18559–18569, 2024. 1, 2, 3, 6