Dual DETRs for Multi-Label Temporal Action Detection

Yuhan Zhu[1*] Guozhen Zhang[1*] Jing Tan[2] Gangshan Wu[1] Limin Wang[1,3,†]
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]The Chinese University of Hong Kong [3]Shanghai AI Lab
https://github.com/MCG-NJU/DualDETR

Abstract

Temporal Action Detection (TAD) aims to identify the action boundaries and the corresponding category within untrimmed videos. Inspired by the success of DETR in object detection, several methods have adapted the query-based framework to the TAD task. However, these approaches primarily followed DETR to predict actions at the instance level (i.e., identify each action by its center point), leading to sub-optimal boundary localization. To address this issue, we propose a new Dual-level query-based TAD framework, namely DualDETR, to detect actions from both instance-level and boundary-level. Decoding at different levels requires semantics of different granularity, therefore we introduce a two-branch decoding structure. This structure builds distinctive decoding processes for different levels, facilitating explicit capture of temporal cues and semantics at each level. On top of the two-branch design, we present a joint query initialization strategy to align queries from both levels. Specifically, we leverage encoder proposals to match queries from each level in a one-to-one manner. Then, the matched queries are initialized using position and content prior from the matched action proposal. The aligned dual-level queries can refine the matched proposal with complementary cues during subsequent decoding. We evaluate DualDETR on three challenging multi-label TAD benchmarks. The experimental results demonstrate the superior performance of DualDETR to the existing state-of-the-art methods, achieving a substantial improvement under det-mAP and delivering impressive results under seg-mAP.

1. Introduction

Temporal Action Detection (TAD) [4, 13, 25–27, 45, 55, 64] is one of the fundamental tasks in video understanding [10–12, 22, 44, 50, 57, 59, 62, 65], with a wide range of real-world applications in video editing [18], sports analytics [16, 23], surveillance footage analysis [56], and autonomous driving [1]. TAD aims to identify the starting and ending time of human actions, and simultaneously recognize the corresponding action categories. To address the complex real-world application scenario for TAD, we focus on the complicated Multi-label Temporal Action Detection (Multi-label TAD) [6–8, 46, 48], where diverse actions from different categories co-exist in untrimmed videos, often with *significant temporal overlaps*.

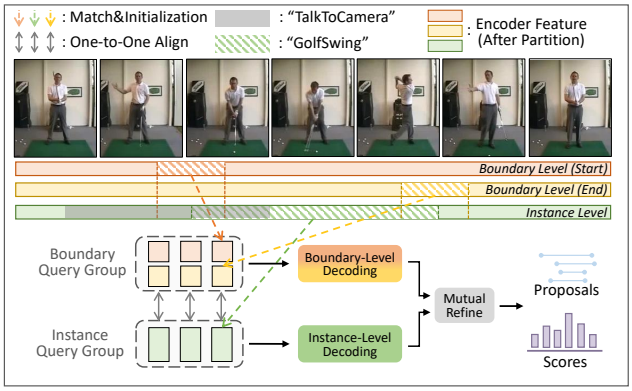

Figure 1. **DualDETR** operates at both the instance level and boundary level (start, end) by using two groups of queries, with each group corresponding to one level. To capture specific semantics at each level, we introduce a two-branch decoding structure. This structure separates the decoding process for each level, allowing queries from each group to focus on their corresponding encoder feature map. Furthermore, we propose a query alignment strategy equipped with joint initialization. This strategy aligns the queries from two groups by matching them with the same detection goal, as denoted by the bidirectional arrow.

Inspired by the success of DETR [2] in object detection, several approaches [21, 31, 40, 45–47] adopted the query-based detection pipeline and used a sparse set of learnable decoder queries to directly predict actions without NMS post-processing. These approaches commonly follow DETR to detect actions from the instance level. They identify each action by its center point and predict duration based on offsets. Although instance-level detection bene-

*: Equal Contribution. †: Corresponding author (lmwang@nju.edu.cn).

fits the capture of the important semantic frames within actions, such practice overlooks a crucial gap between two tasks: object locations are mainly decided by their centroids, whereas actions in videos are often defined by the starting and ending boundaries. Hence, these methods remain poor at the precise localization of action boundaries.

To bridge this gap, we propose a novel Dual-level query-based TAD framework (DualDETR) that integrates both instance-level and boundary-level modeling into the action decoding. As depicted in Fig. 1, DualDETR employs two groups of decoder queries, namely boundary-level query group (red and yellow) and instance-level query group (green), with each group corresponding to one level of decoding. The instance-level queries capture important semantic frames within the proposal, providing a holistic understanding of the action content. Meanwhile, the boundary-level queries focus on the details around proposal boundaries, exhibiting higher sensitivity to the salient boundary frames. Following the dual-level decoding pipeline, DualDETR can improve the action proposals by combining reliable recognition from the instance level and precise boundary refinement from the boundary level.

Simply decoding the two levels of queries via a shared decoder does not yield optimal performance. In general, decoding from boundary and instance levels requires semantics of different granularity. Using a shared decoder for dual levels will fail to focus on specific semantics at each level, hence hampering the effective decoding for both levels. To address this, we propose the two-branch decoding structure with feature partition to use a distinctive decoder for each level. Specifically, we partition the encoder feature map along the channel dimension to represent the boundary (start, end) and instance levels. The separation benefits the explicit capture of individual characteristics at each level. This design is especially helpful in multi-label TAD scenarios where different action instances overlap. For instance, as depicted in Fig. 1, the action "*GolfSwing*" starts while the action "*TalkToCamera*" is ongoing in the background. Under such complex scenario, it is challenging to accurately determine the boundaries of each action. Feature separation enables explicit cues for each action at each level to be preserved and processed in different feature maps, thus benefiting the precise localization of overlapping actions.

With the two-branch design for dual levels, we present a novel joint query initialization strategy to align queries from both levels and achieve complementary refinement of action proposals during subsequent decoding. First, we establish the alignment from action proposals predicted by the encoder. Each action proposal is paired with a starting boundary query, an ending boundary query, and an instance query. This alignment allows for a one-to-one matching between boundary and instance queries, enabling joint updates of the matched proposal during decoding. Second, similar to [30, 63], each query is constructed as a pair of position and content vectors. On top of this, instead of learning sample-agnostic priors from training [31, 45], the position and content vectors are initialized with the position and semantic priors from their matched proposal. Thanks to the joint query initialization, the position vectors guide the queries to explicitly focus on the matched proposal, while the content vectors provide semantic guidance for both pair-wise relation modeling and global feature refining.

We conduct extensive experiments on three challenging multi-label TAD benchmarks, MultiTHUMOS [58], Charades [43] and TSU [9]. Our proposed DualDETR outperforms the previous state-of-the-art methods by a large margin under detection-mAP, demonstrating its fine-grained recognition and precise localization abilities. Notably, DualDETR showcases impressive per-frame detection accuracy under the segmentation-mAP, comparing with both detection-based methods and segmentation-based methods.

In summary, our contributions are threefold:

- We identify the sub-optimal localization issue from the instance-level detection paradigm in previous query-based TAD approaches and present a novel dual-level query-based action detection framework (DualDETR).
- To facilitate effective dual-level decoding, we devise a two-branch decoding structure and joint query initialization strategy to align dual-level queries and refine proposals with complementary efforts.
- Extensive experiments demonstrate that DualDETR surpasses the previous state-of-the-art on three challenging benchmarks under det-mAP and achieves impressive results under seg-mAP, compared with both detection methods and segmentation methods.

2. Related Work

Multi-Label Temporal Action Detection. Prior studies [38] in multi-label TAD have primarily formulated the problem as a frame-wise classification (segmentation) task, with an emphasis on action class recognition rather than precise action boundary localization for all action instances. Early research [35, 36] sought to capture temporal context through carefully designed Gaussian kernels. Other works captured and modeled temporal relations with dilated attention layers [7] or a combination of convolution and self-attention blocks [8]. Coarse-Fine [20] adopted a two-stream architecture, facilitating the extraction of features from distinct temporal resolutions. MLAD [48] leveraged the attention mechanism to model actions occurring at the same and across different time steps. PointTAD [46] marked a return of multi-label TAD to the domain of action detection task [15, 39, 41, 46]. In this paper, we present a dual-level framework to further explore the potential of the query-based framework, with a specific focus on the precise localization of action instances in the multi-label TAD task.

Boundary Information in TAD. Previous studies [34, 42, 51, 53] on action boundaries primarily focused on extracting high-quality boundary features for proposal generation or evaluation. The early methods [24, 26, 27, 29, 64] employed convolutional networks to extract boundary features. MGG [32] refined proposal boundaries by identifying positions with higher boundary scores. Temporal ROI Align [17] or boundary pooling techniques were adopted by TCANet [37] and AFSD [25] to retrieve features for boundary refinement. Regarding the query-based methods, RTDNet [45] multiplied the boundary scores with the original video features. However, RTDNet encountered difficulties in achieving reliable recognition scores thus leading to unsatisfactory detection performance, leaving the appropriate way to incorporate boundary information into the query-based framework as an open problem. In this paper, we aim to tackle this problem by proposing a dual-level framework to carefully addresses these challenges with its design.

Query Formulation in DETR. The formulation of decoder queries was widely studied in the objection detection domain. DETR [2] utilized randomly initialized object queries during training to learn dataset-level object distribution. Anchor DETR [52] initialized queries based on anchor points to establish a specific detection mode. Deformable DETR [5] and Conditional DETR v2 [5] leveraged the encoder proposals to provide positional priors for decoder queries. DAB-DETR [30] formulated decoder queries with a content vector and an action vector. Upon this, DINO [63] incorporated position priors for the position vector and randomly initialized the content query during training. In this paper, we share a distinct motivation with the aforementioned object detection methods, which is to achieve effective alignment between dual-level queries.

3. Method

3.1. Preliminaries

Query-Based TAD Framework [21, 31, 40, 45–47] is proposed inspired by the success of DETR [2]. It employs a transformer architecture [49] and typically consists of an encoder and a decoder. The encoder takes video features $X \in \mathbb{R}^{T\times D}$ as input, which are extracted by a pre-trained video encoder (e.g., I3D [3]), where T and D represent the temporal length and feature dimension, respectively. The encoder employs self-attention to model snippet-level temporal relation. Following the refinement by L_E encoder layers, the decoder employs N_q action queries to simultaneously model action-level relations using self-attention and refine global features using cross-attention. Subsequently, a detection head is applied to these action queries to obtain sparse detection results without post-processing technique like Non-Maximum Suppression (NMS). During training, optimal bipartite matching is performed between predicted and ground truth action instances, enabling the calculation of classification and localization losses.

Deformable Attention [66] is proposed to address the slow convergence issue of DETR while improving its computational efficiency. In this paper, we incorporate deformable attention as a tool to explicitly guide attention localization. Let q index a query element. Given the query feature $\boldsymbol{z}_q$, a 1-d reference point $\boldsymbol{t}_q \in [0, 1]$, and the input feature map $X \in \mathbb{R}^{T\times D}$, the deformable attention is calculated as:

$$\begin{aligned}&\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{t}_q, X) = \\ &\sum_{m=1}^{M} \boldsymbol{W}_m \left[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m X\left(\boldsymbol{t}_q + \Delta \boldsymbol{t}_{mqk}\right)\right].\end{aligned} \tag{1}$$

Here, A_{mqk} represents the attention weight computed as $\text{SoftMax}(\text{Linear}(\boldsymbol{z}_q))$. m indexes the attention head and k indexes the sampling temporal points. M and K denote the number of attention heads and sampling points, respectively. $\Delta \boldsymbol{t}_{mqk}$ represents a normalized 1-d sampling offset.

3.2. Overview

Given an untrimmed video, DualDETR aims to predict a set of action instances $\Psi = \{\varphi_n = (t_n^s, t_n^e, a_n)\}_{n=1}^{N_g}$. Here, N_g represents the number of ground-truth action instances, t_n^s, t_n^e, and a_n denote the starting, ending time and the corresponding action label of action instances.

The entire pipeline is illustrated in Fig. 2. DualDETR operates on video features $X \in \mathbb{R}^{T\times D}$, that are extracted by a pre-trained feature extractor (e.g., I3D [3]). The model employs the encoder-decoder pipeline. For feature encoding, the model uses a transformer encoder with deformable attention to efficiently perform temporal modeling at the snippet level. For action decoding, we introduce a **two-branch decoding structure** based on transformer decoders to predict actions from both boundary and instance levels. Accordingly, decoder queries are divided into two groups and the encoder features are also divided along channels for dual-level cross-attention. At each branch, the decoder takes in the corresponding queries and features to make predictions. To achieve complementary refinement of proposals from both levels, we propose a **joint query initialization strategy** to align different groups of queries based on the action proposals predicted from the encoder. Each proposal is matched with a pair of boundary queries and one instance query. The content and position vectors of the queries are initialized by the feature embedding and boundary position of the matched proposal in correspondence. At the end of each layer, a **mutual refinement module** facilitates the communication between the aligned queries. Finally, the classification scores generated by the instance-level content vector, along with the proposals from the mutual refinement module, serve as the final detection results, without the need for NMS post-processing.

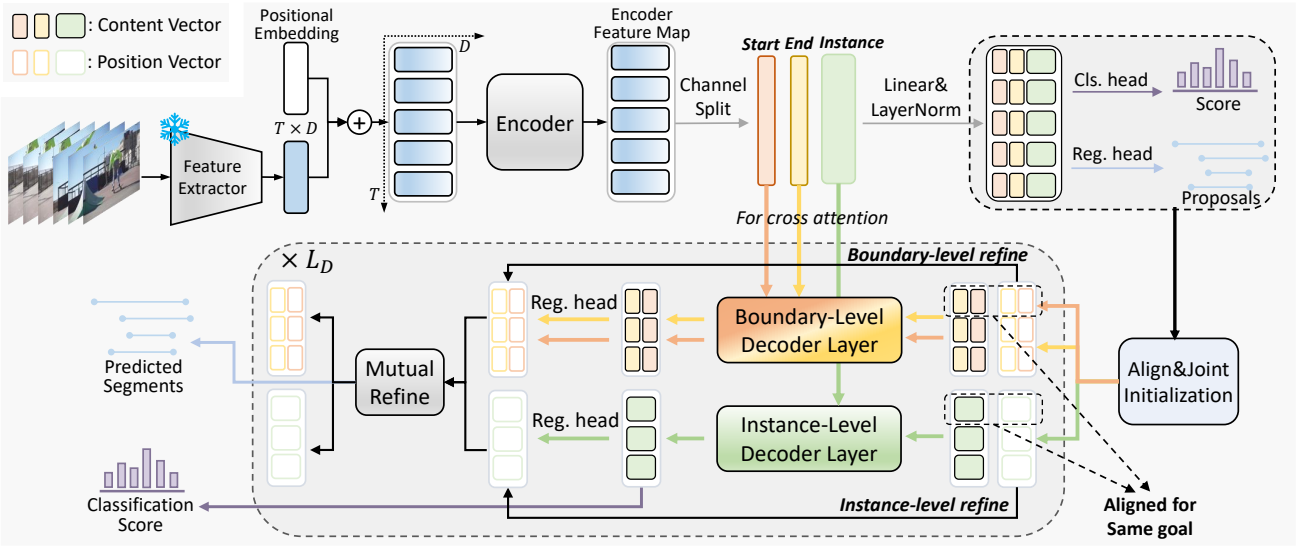

Figure 2. **Pipeline of DualDETR.** The pre-extracted video features, augmented with the positional embedding, pass through a transformer encoder to produce the encoder feature map. This map is divided along the channel dimension into separate feature maps for the boundary-level (start, end) and instance-level modeling, respectively. An auxiliary dense detection head is applied to generate encoder proposals and scores. Upon this, decoder queries are constructed using the query alignment strategy. The decoding process is performed at dual levels. Thanks to the query alignment, dual-level queries can perform a complementary refinement through the mutual refinement module. Finally, DualDETR directly output action instance predictions without NMS post-processing.

3.3. Dual-Level Query Construction

In this subsection, we present the construction of the decoder queries in our dual-level framework. We divide the decoder queries into two groups, one group for the boundary decoding branch and the other for the instance branch. Similar to [14, 30, 63], we disentangle the position and content decoding for each query by constructing it as a pair of position and content vectors. The instance-level query group, denoted as $\boldsymbol{i}$, consists of content matrix $\boldsymbol{i}^{con} \in \mathbb{R}^{N_q \times D/2}$ and position matrix $\boldsymbol{i}^{pos} \in \mathbb{R}^{N_q \times 2}$, where N_q is the number of queries, and D denotes the number of feature channels. The content vector captures high-level semantic information, while the position vector contains two normalized scalars representing the center and duration of a proposal. Similarly, the boundary-level query group consists of start and end queries, represented as $\boldsymbol{s}$ and $\boldsymbol{e}$ respectively. Each boundary query also contains content and position matrix, denoted as $\boldsymbol{s}^{con}, \boldsymbol{e}^{con} \in \mathbb{R}^{N_q \times D/4}$ and $\boldsymbol{s}^{pos}, \boldsymbol{e}^{pos} \in \mathbb{R}^{N_q \times 1}$. The position vectors contain normalized scalars representing the starting and ending times of a proposal. During the decoding process, the position vectors serve as reference points, providing explicit positional guidance in both self-attention and cross-attention. Meanwhile, the content vectors offer semantic guidance for pair-wise query relation modeling in self-attention and query refinement in cross-attention. This dual-level query corresponds to the subsequent two-barnch decoding.

3.4. Query Alignment with Joint Initialization

After constructing the action queries with two groups and decoding each group within separate branches, it is important to align the queries from both groups to facilitate their joint refinement of action proposals. This alignment enables the model to benefit from both the instance-level queries, which provide semantic guidance for recognition, and the boundary-level queries, which refine the proposal boundaries with high precision. To achieve query alignment, we first obtain proposals and classification scores by applying a detection head to the encoder feature map. These proposals, selected based on their classification scores, are then matched with the decoder queries from both groups. For example, considering the k-th selected proposal, as depicted in Fig. 3 (a), we match this proposal with the k-th instance-level query $\boldsymbol{i}_k = \{\boldsymbol{i}_k^{pos}, \boldsymbol{i}_k^{con}\}$, as well as the k-th boundary-level query $\boldsymbol{s}_k = \{\boldsymbol{s}_k^{pos}, \boldsymbol{s}_k^{con}\}, \boldsymbol{e}_k = \{\boldsymbol{e}_k^{pos}, \boldsymbol{e}_k^{con}\}$. This matching process ensures an one-to-one alignment between the instance and boundary queries, allowing them to jointly update the matched proposal during the decoding process.

Based on the matching, we propose a joint query initialization strategy to provide a good kick-start for the aligned queries and further align the queries with their matched proposal. As illustrated in Fig. 3 (b), the start and end timestamps from the k-th proposal are used to initialize the boundary-level position vectors $\boldsymbol{s}_k^{pos}$ and $\boldsymbol{e}_k^{pos}$, which can also be transformed into center and duration values to ini-

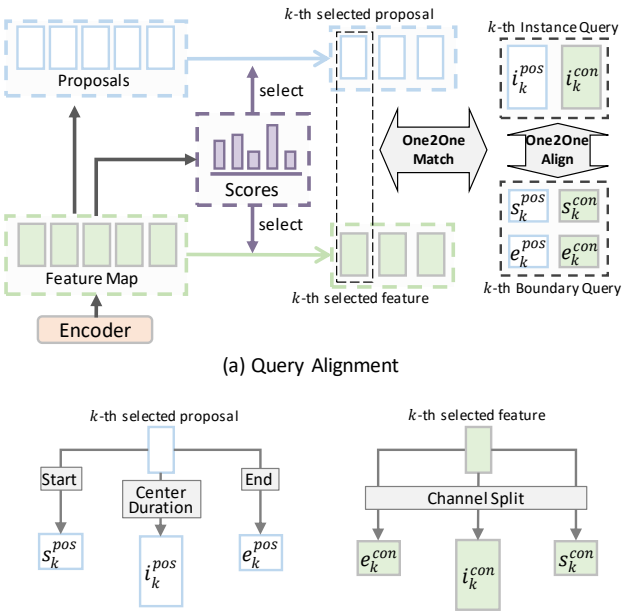

Figure 3. **Query Alignment with Joint Initialization.** (a) Instance queries and boundary queries are aligned to match with the encoder predictions in a one-to-one manner. (b) The matched encoder prediction serves as the initialization for dual-level queries.

tialize the instance-level position vectors $\boldsymbol{i}_k^{pos}$. At the same time, the k-th selected feature is employed to initialize the content vectors $\boldsymbol{s}_k^{con}$, $\boldsymbol{e}_k^{con}$, and $\boldsymbol{i}_k^{con}$ through channel splitting. This joint initialization strategy offers two benefits: (1) it further enhances the alignment between the dual-level queries, and (2) it leverages both position and semantic priors from the proposal, resulting in a better match.

3.5. Two-Branch Decoding

Before the two-branch decoding process, we partition the encoder feature map X^{Enc} into two levels: 1) the boundary-level, which consists of the starting boundary feature $X_s^{Enc} \in \mathbb{R}^{T\times(D/4)}$ and the ending boundary feature $X_e^{Enc} \in \mathbb{R}^{T\times(D/4)}$, and 2) the instance-level feature $X_i^{Enc} \in \mathbb{R}^{T\times(D/2)}$. This partition allows the queries from each level to focus on the specific semantics relevant to their respective levels. The decoder layers for both levels consist of a self-attention module [49], a deformable cross-attention module, and a feed-forward network (FFN).

Boundary-Level Decoding. The boundary-level decoder layer takes boundary-level features maps X_s^{Enc}, X_e^{Enc}, along with the content query vectors $\boldsymbol{s}^{con}, \boldsymbol{e}^{con}$, as well as the position query vectors $\boldsymbol{s}^{pos}, \boldsymbol{e}^{pos}$ as input. After the self-attention step, we employ deformable cross-attention to attend to the proposal boundaries. Specifically, we reuse the position vector $\boldsymbol{s}^{pos}, \boldsymbol{e}^{pos}$ as reference points (as described in Eq. (1)). The deformable attention attends to a small set of key sampling points around each reference point. The refinement of the content vector can be represented by:

$$\begin{aligned}\boldsymbol{s}^{con} &= \text{FFN}(\text{DeformAttn}(\boldsymbol{s}^{con}, \boldsymbol{s}^{pos}, X_s^{Enc})),\\ \boldsymbol{e}^{con} &= \text{FFN}(\text{DeformAttn}(\boldsymbol{e}^{con}, \boldsymbol{e}^{pos}, X_e^{Enc})).\end{aligned} \tag{2}$$

Subsequently, a regression head is applied to the refined content vectors to generate offsets $\Delta\boldsymbol{s}, \Delta\boldsymbol{e}$, which are used to refine the position vectors, as follows:

$$\begin{aligned}\boldsymbol{s}^{pos} &= \sigma(\Delta\boldsymbol{s} + \sigma^{-1}(\boldsymbol{s}^{pos})),\\ \boldsymbol{e}^{pos} &= \sigma(\Delta\boldsymbol{e} + \sigma^{-1}(\boldsymbol{e}^{pos})),\end{aligned} \tag{3}$$

where σ and σ^{-1} denote the `sigmoid` and `inverse sigmoid` functions, respectively. These functions are employed to ensure that the proposal coordinates remain normalized at all times.

Instance-Level Decoding. Similarly to the boundary-level decoding, the instance-level decoder layer takes instance-level feature maps X_i^{Enc}, the content query vector $\boldsymbol{i}^{con}$, and the position query vector $\boldsymbol{i}^{pos}$ as input. After applying self-attention to model the query relationships, the content query refines itself by attending to the key semantic frames within the instance-level feature. This process utilizes the instance-level position vector as the reference point, which contains the center point and duration of the proposals. The refinement can be expressed as:

$$\boldsymbol{i}^{con} = \text{FFN}(\text{DeformAttn}(\boldsymbol{i}^{con}, \boldsymbol{i}^{pos}, X_i^{Enc})). \tag{4}$$

Subsequently, a regression head is employed to generate offsets Δi for refining the position vectors:

$$\boldsymbol{i}^{pos} = \sigma(\Delta\boldsymbol{i} + \sigma^{-1}(\boldsymbol{i}^{pos})), \tag{5}$$

Mutual Refinement. After refining the decoder queries at separate levels, we introduce a mutual refinement module to achieve complementary refinement of proposals by leveraging their matched queries. This approach allows the boundary level to benefit from the robust localization of the instance level, while the instance level can leverage the precise boundary refinement of the boundary level. Specifically, we utilize the boundary-level position vector to refine the instance-level counterparts, which can be represented as:

$$\begin{aligned}\boldsymbol{i}^{pos,0} &\leftarrow \frac{\boldsymbol{i}^{pos,0} + (\boldsymbol{s}^{pos} + \boldsymbol{e}^{pos})/2}{2},\\ \boldsymbol{i}^{pos,1} &\leftarrow \frac{\boldsymbol{i}^{pos,1} + (\boldsymbol{e}^{pos} - \boldsymbol{s}^{pos})}{2},\end{aligned} \tag{6}$$

where $\boldsymbol{i}^{pos,0}$ and $\boldsymbol{i}^{pos,1}$ represent the center point and duration contained in the instance-level position vector. Similarly, we refine the boundary level as follows:

$$\begin{aligned}\boldsymbol{s}^{pos} &\leftarrow \frac{\boldsymbol{s}^{pos} + (\boldsymbol{i}^{pos,0} - \boldsymbol{i}^{pos,1}/2)}{2},\\ \boldsymbol{e}^{pos} &\leftarrow \frac{\boldsymbol{e}^{pos} + (\boldsymbol{i}^{pos,0} + \boldsymbol{i}^{pos,1}/2)}{2}.\end{aligned} \tag{7}$$

These refined position vectors are then utilized in the subsequent layers or can serve as the final predictions in the last decoder layer.

3.6. Training

Label Assignment. The predicted action set is denoted as $\hat{\Psi} = \left\{\hat{\varphi}_n = \left(\hat{t}_n^s, \hat{t}_n^e, \hat{\boldsymbol{p}}_n\right)\right\}_{n=1}^{N_q}$, where $\hat{t}_n^s, \hat{t}_n^e$ represent the starting and ending time of predicted action instances, and $\hat{\boldsymbol{p}}_n$ represents the corresponding classification scores. The ground-truth set Ψ is padded with a no-action placeholder, denoted as $\varnothing$. The cost for a permutation $\sigma \in \mathfrak{S}_{N_q}$ of query set is defined as follows:

$$\begin{aligned} \mathcal{C}_{cls}(\sigma) &= \mathcal{L}_{cls}\left(\hat{\boldsymbol{p}}_{\sigma(i)}, a_i\right), \\ \mathcal{C}_{iou}(\sigma) &= \mathcal{L}_{iou}\left((\hat{t}_{\sigma(i)}^s, \hat{t}_{\sigma(i)}^e), (t_n^s, t_n^e)\right), \\ \mathcal{C}_{L_1}(\sigma) &= \mathcal{L}_{L_1}\left((\hat{t}_{\sigma(i)}^s, \hat{t}_{\sigma(i)}^e), (t_n^s, t_n^e)\right), \end{aligned} \tag{8}$$

where $\mathcal{L}_{iou}$ represents the temporal IoU loss and $\mathcal{L}_{L1}$ denotes the L_1 distance. Focal loss [28] is utilized as $\mathcal{L}_{cls}$ following [31, 66]. The bipartite matching between two sets aims to find the permutation σ^* with the lowest cost:

$$\sigma^* = \arg\min_{\sigma} \sum_{c_i \neq \varnothing} \alpha_{cls}\mathcal{C}_{cls}(\sigma) + \alpha_{iou}\mathcal{C}_{iou}(\sigma) + \alpha_{L_1}\mathcal{C}_{L_1}(\sigma). \tag{9}$$

where c_i denotes the class label. α_{cls}, α_{iou}, and α_{L_1} are the weights of each cost, set to 6, 2, and 5 respectively, as in [31].

Loss Functions. After obtaining the best permutation σ^*, the final optimization goal can be expressed as:

$$\mathcal{L} = \sum_{i}^{N_q} \lambda_{cls}\mathcal{C}_{cls}(\sigma^*) + \mathbb{I}_{\{c_i \neq \varnothing\}}\left(\lambda_{iou}\mathcal{C}_{iou}(\sigma^*) + \lambda_{L_1}\mathcal{C}_{L_1}(\sigma^*)\right), \tag{10}$$

where $\mathbb{I}(\cdot)$ is the indicator function, and λ_{cls} and λ_{iou} and λ_{L_1} are set to 2, 2, and 5 respectively, as in [31].

4. Experiments

4.1. Dataset and Setup

Dataset. We evaluate DualDETR on three challenging datasets: (1) MultiTHUMOS [58], an extension of THUMOS14 [19], containing 413 sports videos of 65 classes. The average video length is 212 seconds, and each video has an average of 97 ground-truth instances. (2) Charades [43] comprises 9,848 videos of daily activities across 157 classes. The dataset contains an average of 6.75 action instances per video, with an average video length of 30 seconds. (3) TSU [9] is a dataset recorded in an indoor environment with dense annotations. Up to 5 actions can happen at the same moment. Additionally, the dataset also includes many long-term composite actions.

Methods	Backbone	MultiTHUMOS		Charades	
		Det_{mAP}	Seg_{mAP}	Det_{mAP}	Seg_{mAP}
R-C3D [54]	C3D	–	–	–	17.6
Super-event [36]	I3D	–	36.4	–	18.6
TGM [35]	I3D	–	37.2	–	20.6
PDAN [7]	I3D	17.3	40.2	8.5	23.7
Coarse-Fine [20]	X3D	–	–	6.1	25.1
MLAD [48]	I3D	14.2	42.2	–	18.4
CTRN [6]	I3D	–	44.0	–	25.3
MS-TCT [8]	I3D	16.2	43.1	7.9	**25.4**
PointTAD [46]	I3D	23.5	39.8	12.1	21.0
DualDETR	I3D	**32.6**	**45.5**	**15.3**	23.2

Table 1. **Comparison with state-of-the-art** multi-label TAD methods on MultiTHUMOS and Charades under **Det**ection-**mAP** (%) and **Seg**mentation-**mAP** (%).

Methods	Backbone	GFLOPs	Det_{mAP}	Seg_{mAP}
R-C3D [54]	C3D	–	–	8.7
Super-event [36]	I3D	0.8	–	17.2
TGM [35]	I3D	1.2	–	26.7
PDAN [7]	I3D	3.2	–	32.7
MS-TCT [8]	I3D	6.6	10.6	33.7
DualDETR	I3D	5.5	**20.8**	**34.8**

Table 2. **Comparison with state-of-the-art** multi-label TAD methods on the TSU dataset, where the action instances are highly overlapped. The GFLOPs are presented to evaluate the computation efficiency.

Implementation Details. Our method relies on offline extracted video features, and we utilize the two-stream I3D [3] network pre-trained on Kinetics [3] as the feature extractor. The video features are extracted at a stride of 4 frames, 8 frames, and 16 frames for MultiTHUMOS, Charades, and TSU, respectively. During training, for MultiTHUMOS and TSU, we crop each video feature sequence into windows of length 256 and 96, respectively, with a stride ratio of 0.75. During inference, the stride ratio is set to 0.25. For Charades, we directly input the entire video feature into the model, padding short videos with zeros for parallel computation. We utilize AdamW [33] optimizer with a learning rate of 2e-4 and weight decay of 0.05. Training is performed for 30 epochs on MultiTHUMOS and Charades, and 20 epochs on TSU, with the learning rate dropping to 2e-5 during the last 3 epochs.

Metrics. Following [46], we evaluate our method using two metrics: detection-mAP and segmentation-mAP. Det-mAP measures boundary localization accuracy, while seg-mAP evaluates frame-wise multi-label classification precision. We report the average mAP across tIoU thresholds $[0.1 : 0.1 : 0.9]$ and individual mAPs at each threshold.

Methods	0.1	0.3	0.5	Avg.
BSN [26]+P-GCN [60]	22.2	16.7	8.5	10.0
BSN [26]+ContextLoc [67]	22.9	18.0	10.8	11.0
AFSD [25]	30.5	23.6	14.0	14.7
TadTR [31]	48.0	41.1	29.1	27.4
ActionFormer [61]	49.0	44.5	33.3	29.6
TriDet [42]	–	–	34.3	30.7
DualDETR	**53.4**	**47.4**	**35.2**	**32.6**

Table 3. **Comparison with traditional TAD methods** on MultiTHUMOS under detection-mAP (%).

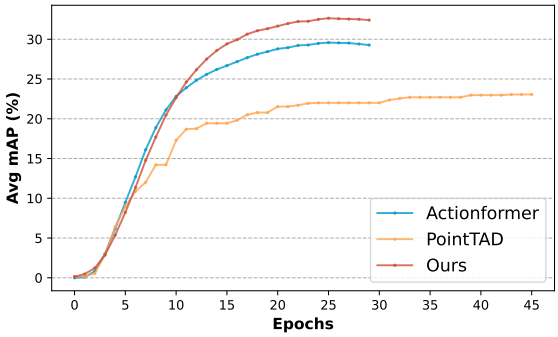

Figure 4. **Convergence curves** of DualDETR , PointTAD [46], and ActionFormer [61] on MultiTHUMOS.

4.2. Main Results

Comparison with State-of-The-Art Methods. In Tab. 1, we compare the performance of DualDETR with previous multi-label TAD methods. To calculate the segmentation-mAP metric, we follow PointTAD [46] by converting sparse prediction tuples into dense segmentation scores. Our DualDETR outperforms all previous methods by a substantial margin under detection-mAP ($+\mathbf{9.1}\%$ on MultiTHUMOS and $+\mathbf{3.2}\%$ on Charades), highlighting its exceptional boundary localization ability. Meanwhile, even when evaluated under segmentation-mAP, DualDETR delivers comparable results to methods specifically designed for the frame-wise classification task. This further emphasizes the superiority of our approach. Additionally, we present the results on the TSU dataset in Tab. 2, where action instances are highly overlapped. DualDETR still achieves remarkable performance while maintaining favorable computational efficiency.

Comparison with Traditional TAD Methods. In Tab. 3, we compare several representative methods in traditional TAD. Since MultiTHUMOS shares similar data preparation with THUMOS14, we reproduce these methods using their default hyper-parameter setting for THUMOS14, with the exception of TadTR, where the number of queries is adjusted to the same number as ours for fair comparison. While these methods demonstrate decent performance in traditional TAD, directly applying them to multi-label scenarios yields unsatisfactory results. In contrast, our DualDETR takes into account the dense overlapping scenarios in our architecture design, thus achieving superior detection performance, surpassing all these methods.

Convergence Speed. Query-based methods often encounter slow convergence issues [2, 66] compared to dense prediction methods. In Fig. 4, we compare the convergence speeds of DualDETR with PointTAD (another query-based method) and ActionFormer (dense prediction). Remarkably, our DualDETR demonstrates a favorable convergence speed, thanks to the effectiveness of our two-branch collaboration structure.

4.3. Ablation Study

We perform ablation studies on the MultiTHUMOS dataset to evaluate the effectiveness of our proposed method and investigate alternative design choices. In the tables, the default setting is colored gray.

Study on Dual-Level Design. In Tab. 4, we present the results of our ablation study, focusing on each component of DualDETR. We examine the single-level results obtained by employing either the instance-level or boundary-level detection paradigm. The instance-level detection approach yields sub-optimal results as it lacks explicit focus on boundary information. On the other hand, the boundary-level detection approach faces challenges such as incomplete detection and difficulty in obtaining reliable scores, resulting in inferior performance.

Next, we proceed to study the effectiveness of each proposed component in a step-wise manner. We first present our baseline, which simply combines instance-level and boundary-level queries into the same detection framework. Then, we incorporate our two-branch design into this framework, enabling the decoding process to focus on specific semantics at each level. This integration leads to a promising performance gain of $\mathbf{2}\%$. Furthermore, we introduce query alignment to match the dual-level queries with the encoder proposal, enabling effective collaboration. This alignment brings an additional performance gain of $\mathbf{1.51}\%$. Lastly, the joint query initialization strategy further facilitates the alignment between queries, resulting in an additional performance gain of $\mathbf{2.09}\%$.

Study on Query Initialization. Previous query-based TAD approaches typically optimize randomly initialized queries during training to learn dataset-level action distribution. In contrast, DualDETR takes advantage of position and semantic priors from the matched proposal. These priors serve two important purposes: help the decoder queries explicitly concentrate on the matched proposals, and provide an additional constraint on the aligned queries, facilitating effective collaboration. Thanks to these good qualities, Du-

Setting	0.1	0.3	0.5	0.7	0.9	Avg.
Single-Level						
Instance Level	50.22	42.92	30.69	15.72	2.28	28.62
Boundary Level	42.11	33.60	23.58	12.96	1.99	22.92
Dual-Level						
(1): Simple Combine	49.10	40.99	28.11	14.62	1.65	27.04
(2): (1)+Two-Branch	51.24	43.20	30.92	16.53	2.85	29.04
(3): (2)+Query Align	51.70	44.76	32.97	18.37	3.03	30.55
(4): (3)+Joint Init	**53.42**	**47.41**	**35.18**	**20.18**	**4.02**	**32.64**

Table 4. **Ablation Study on Dual-Level**. We first show the results of single-level detection. Following that, we present our baseline, a method simply combining two-level detection. Subsequently, we perform step-wise ablations on our proposed approaches to evaluate their effectiveness.

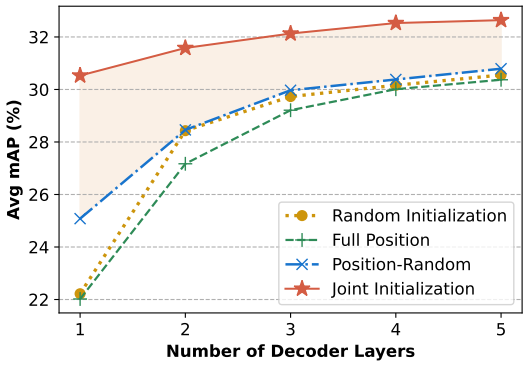

Figure 5. **Comparison of detection mAP at each decoder layer for different query initializations.** All initialization strategies are re-implemented in the DualDETR framework. The joint initialization showcased strong detection performance from early decoding stages and continues to outperform other initialization variants as the number of decoder layers increases.

alDETR enjoys prominent detection accuracy with only a few decoder layers, as depicted in Fig. 5.

Additionally, we have observed two other popular initialization methods [63, 66] in the objection detection field. Zhu et al. [66] initializes both position and content vectors with proposal predictions from the encoder (denoted as "Full Position"). However, this approach only leverages position priors for initialization, lacking crucial semantics priors necessary for fine-grained content decoding. On the other hand, Zhang et al. [63] goes back to learning the content vector during training while keeping the proposal predictions for position vector initialization (denoted as "Position Random"). Although these methods achieve superior results in object detection, adapting them to a multi-level query-based framework remains non-trivial, as shown in Fig. 5.

Alternative Choices for Mutual Refinement. Based on the design of DualDETR, we explore alternative choices for the mutual refinement module in Tab. 5. Firstly, we consider sequential refinement, which updates the position vectors in a sequential manner. This can be done either by refining the boundary-level vectors first followed by the instance-level vectors, or vice versa. Secondly, we investigate the timing of position vector updates within the decoding process. By default, the position vectors are updated at the end of each layer. We also explore the option of updating them at the end of the entire decoding process (last layer). Furthermore, we experiment with refining the content vectors during the mutual refinement process by feeding the concatenated content vectors into a feed-forward network. Overall, our default setting benefits from parallel computation and achieves superior performance. It is also worth noting that, despite the various alternative choices explored, DualDETR consistently demonstrates favorable performance, showcasing its robustness.

Setting	0.1	0.3	0.5	0.7	0.9	Avg.
Sequential Refinement						
Boundary→Instance	**53.76**	47.15	34.84	19.55	3.75	32.35
Instance→Boundary	53.36	**47.44**	**35.76**	19.60	3.84	32.50
Parallel Refinement						
Refine Position	53.42	47.41	35.18	**20.18**	4.02	**32.64**
Refine Pos (at last layer)	52.51	46.60	34.71	18.98	4.00	31.81
Refine Position&Content	52.54	46.31	34.23	19.44	**4.13**	31.82

Table 5. **Alternative design choices for mutual refinement.**

5. Conclusion

In this paper, we introduce DualDETR, a novel dual-level query-based TAD framework. DualDETR integrates both instance-level and boundary-level decoding to achieve more precise localization of temporal boundaries. To enable explicit modeling of each level's semantics, we propose a two-branch decoding structure, which allows us to capture the individual characteristics of each level. Meanwhile, to achieve complementary refinement of action proposals, we introduce query alignment, which matches dual-level queries with encoder proposals in a one-to-one manner. Furthermore, we propose the joint query initialization strategy that exploits rich priors from matched proposals, further enhancing the alignment. Thanks to the dual-level design, DualDETR outperforms existing TAD methods on various multi-label TAD benchmarks without the need for NMS post-processing.

Acknowledgements. This work is supported by the National Key R&D Program of China (No. 2022ZD0160900), the National Natural Science Foundation of China (No. 62076119, No. 61921006)), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Munirah Alyahya, Shahad Alghannam, and Taghreed Alhussan. Temporal driver action localization using action classification methods. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3318–3325, 2022. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1, 3, 7

[3] 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

[4] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1130–1139, 2018. 1

[5] Xiaokang Chen, Fangyun Wei, Gang Zeng, and Jingdong Wang. Conditional detr v2: Efficient detection transformer with box queries. *arXiv preprint arXiv:2207.08914*, 2022. 3

[6] Rui Dai, Srijan Das, and Francois Bremond. Ctrn: Class-temporal relational network for action detection. *arXiv preprint arXiv:2110.13473*, 2021. 1, 6

[7] Rui Dai, Srijan Das, Luca Minciullo, Lorenzo Garattoni, Gianpiero Francesca, and François Bremond. Pdan: Pyramid dilated attention network for action detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2970–2979, 2021. 2, 6

[8] Rui Dai, Srijan Das, Kumara Kahatapitiya, Michael S Ryoo, and François Brémond. Ms-tct: multi-scale temporal convtransformer for action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20041–20051, 2022. 1, 2, 6

[9] Rui Dai, Srijan Das, Saurav Sharma, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome untrimmed: Real-world untrimmed videos for activity detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2533–2550, 2022. 2, 6

[10] Shuangrui Ding, Maomao Li, Tianyu Yang, Rui Qian, Haohang Xu, Qingyi Chen, Jue Wang, and Hongkai Xiong. Motion-aware contrastive video representation learning via foreground-background merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9716–9726, 2022. 1

[11] Shuangrui Ding, Rui Qian, and Hongkai Xiong. Dual contrastive learning for spatio-temporal representation. In *Proceedings of the 30th ACM international conference on multimedia*, pages 5649–5658, 2022.

[12] Shuangrui Ding, Peisen Zhao, Xiaopeng Zhang, Rui Qian, Hongkai Xiong, and Qi Tian. Prune spatio-temporal tokens by semantic-aware temporal accumulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16945–16956, 2023. 1

[13] Jialin Gao, Zhixiang Shi, Guanshuo Wang, Jiani Li, Yufeng Yuan, Shiming Ge, and Xi Zhou. Accurate temporal action proposal generation with relation-aware pyramid network. In *Proceedings of the AAAI conference on artificial intelligence*, pages 10810–10817, 2020. 1

[14] Ziteng Gao, Limin Wang, Bing Han, and Sheng Guo. Adamixer: A fast-converging query-based object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2022. 4

[15] Zikai Gao, Peng Qiao, and Yong Dou. Haan: Human action aware network for multi-label temporal action detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5059–5069, 2023. 2

[16] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1711–1721, 2018. 1

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[18] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European Conference on Computer Vision*, pages 709–727. Springer, 2020. 1

[19] 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. 6

[20] Kumara Kahatapitiya and Michael S Ryoo. Coarse-fine networks for temporal activity detection in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8385–8394, 2021. 2, 6

[21] 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. 1, 3

[22] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *Computer Vision–ECCV 2020: 16th European Conference*, pages 68–84. Springer, 2020. 1

[23] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. Multisports: A multi-person video dataset of spatio-temporally localized sports actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13536–13545, 2021. 1

[24] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11499–11506, 2020. 3

[25] 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, 7

[26] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3, 7

[27] 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, 3

[28] 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

[29] Shuming Liu, Xu Zhao, Haisheng Su, and Zhilan Hu. Tsi: Temporal scale invariant network for action proposal generation. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 3

[30] 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. *arXiv preprint arXiv:2201.12329*, 2022. 2, 3, 4

[31] 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*, 31:5427–5441, 2022. 1, 2, 3, 6, 7

[32] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3604–3613, 2019. 3

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[34] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Proposal-free temporal action detection via global segmentation mask learning. In *European Conference on Computer Vision*, pages 645–662. Springer, 2022. 3

[35] AJ Piergiovanni and Michael Ryoo. Temporal gaussian mixture layer for videos. In *International Conference on Machine learning*, pages 5152–5161. PMLR, 2019. 2, 6

[36] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5304–5313, 2018. 2, 6

[37] 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. 3

[38] Faegheh Sardari, Armin Mustafa, Philip JB Jackson, and Adrian Hilton. Pat: Position-aware transformer for dense multi-label action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2988–2997, 2023. 2

[39] Jiayi Shao, Xiaohan Wang, Ruijie Quan, Junjun Zheng, Jiang Yang, and Yi Yang. Action sensitivity learning for temporal action localization. *arXiv preprint arXiv:2305.15701*, 2023. 2

[40] Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. React: Temporal action detection with relational queries. In *Computer Vision–ECCV 2022: 17th European Conference*, pages 105–121. Springer, 2022. 1, 3

[41] Dingfeng Shi, Qiong Cao, Yujie Zhong, Shan An, Jian Cheng, Haogang Zhu, and Dacheng Tao. Temporal action localization with enhanced instant discriminability. *arXiv preprint arXiv:2309.05590*, 2023. 2

[42] 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. 3, 7

[43] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference,*, pages 510–526. Springer, 2016. 2, 6

[44] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 1

[45] 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, 2, 3

[46] Jing Tan, Xiaotong Zhao, Xintian Shi, Bing Kang, and Limin Wang. Pointtad: Multi-label temporal action detection with learnable query points. In *NeurIPS*, 2022. 1, 2, 6, 7

[47] Jing Tan, Yuhong Wang, Gangshan Wu, and Limin Wang. Temporal perceiver: A general architecture for arbitrary boundary detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12506–12520, 2023. 1, 3

[48] Praveen Tirupattur, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Modeling multi-label action dependencies for temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1460–1470, 2021. 1, 2, 6

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 5

[50] 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 *European conference on computer vision*, pages 20–36. Springer, 2016. 1

[51] Qiang Wang, Yanhao Zhang, Yun Zheng, and Pan Pan. Rcl: Recurrent continuous localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Com-*

puter Vision and Pattern Recognition, pages 13566–13575, 2022. 3

[52] Y Wang, X Zhang, T Yang, and J Sun. Anchor detr: Query design for transformer-based object detection. arxiv 2021. *arXiv preprint arXiv:2109.07107*. 3

[53] Kun Xia, Le Wang, Sanping Zhou, Nanning Zheng, and Wei Tang. Learning to refactor action and co-occurrence features for temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13884–13893, 2022. 3

[54] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017. 6

[55] 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

[56] Qichao Xu, John See, and Weiyao Lin. Localization guided fight action detection in surveillance videos. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 568–573. IEEE, 2019. 1

[57] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2019. 1

[58] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126:375–389, 2018. 2, 6

[59] Zehuan Yuan, Jonathan C Stroud, Tong Lu, and Jia Deng. Temporal action localization by structured maximal sums. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3684–3692, 2017. 1

[60] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7094–7103, 2019. 7

[61] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *Computer Vision–ECCV 2022: 17th European Conference*, pages 492–510. Springer, 2022. 7

[62] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5682–5692, 2023. 1

[63] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Harry Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations*, 2022. 2, 3, 4, 8

[64] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2914–2923, 2017. 1, 3

[65] Zhiyu Zhao, Bingkun Huang, Sen Xing, Gangshan Wu, Yu Qiao, and Limin Wang. Asymmetric masked distillation for pre-training small foundation models. *arXiv preprint arXiv:2311.03149*, 2023. 1

[66] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 3, 6, 7, 8

[67] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. Enriching local and global contexts for temporal action localization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13516–13525, 2021. 7