Feature Aggregated Queries for Transformer-based Video Object Detectors

Yiming Cui
University of Florida
cuiyiming@ufl.edu

Abstract

Video object detection needs to solve feature degradation situations that rarely happen in the image domain. One solution is to use the temporal information and fuse the features from the neighboring frames. With Transformer-based object detectors getting a better performance on the image domain tasks, recent works began to extend those methods to video object detection. However, those existing Transformer-based video object detectors still follow the same pipeline as those used for classical object detectors, like enhancing the object feature representations by aggregation. In this work, we take a different perspective on video object detection. In detail, we improve the qualities of queries for the Transformer-based models by aggregation. To achieve this goal, we first propose a vanilla query aggregation module that weighted averages the queries according to the features of the neighboring frames. Then, we extend the vanilla module to a more practical version, which generates and aggregates queries according to the features of the input frames. Extensive experimental results validate the effectiveness of our proposed methods: On the challenging ImageNet VID benchmark, when integrated with our proposed modules, the current state-of-the-art Transformer-based object detectors can be improved by more than 2.4% on mAP and 4.2% on AP_{50}. Code is available at https://github.com/YimingCuiCuiCui/FAQ.

1. Introduction

Object detection is an essential yet challenging task which aims to localize and categorize all the objects of interest in a given image [14, 50, 98]. With the development of deep learning, extraordinary processes have been achieved in static image object detection [3, 14, 22, 42, 47, 71]. Existing object detectors can be mainly divided into three categories: two-stage [3, 29, 32, 46, 65], one-stage [47, 52, 57, 62–64, 72, 73] and query-based models [4, 27, 56, 66, 71, 103]. For better performance, two-stage models generate a set of proposals and then refine the prediction results, like R-CNN families [15, 26, 32, 65]. However, these two-stage object detectors usually suffer from a low inference speed. Therefore, one-stage object detectors are introduced to balance the efficiency and performance, which directly predicts the object locations and categories based on the input image feature maps, like YOLO series [62–64, 69] and FCOS [72, 73]. Recently, query-based object detectors have been introduced, which generate the predictions based on a series of input queries and do not require complicated post-processing pipelines like NMS [2, 55, 60]. Some typical example models are DETR series [4, 56, 66, 103] in Figure 1(a) and Sparse R-CNN series [24, 35, 71].

With the existing approaches getting better performance on the image domain, researchers began to extend the tasks to the video domain [10, 41, 67, 75, 83, 85]. One of the most challenging issues of video object detection is handling the feature degradation caused by motion, which rarely appears in static images. Since videos provide informative temporal hints, post-processing-based video object detectors are proposed [1, 31, 39, 40, 68]. As shown in Figure 1(c), these methods first apply image object detectors on every individual frame and then associate the prediction results. However, since the image object detectors and the post-processing pipelines are not optimized jointly, these models usually suffer from poor performance.

Besides post-processing methods, feature-aggregation-based models [6, 13, 30, 34, 38, 82, 100, 104] are introduced to improve the feature representations for video object detection. These approaches first weighted average the features from the neighboring frames and then fed the aggregated features into the task heads for the final prediction, as shown in Figure 1(b). The pipeline for weighted averaging is usually based on feature similarity [6, 79, 82, 104, 105] or learnable networks [13, 34, 100]. Since Transformer-based models perform better on image object detection, researchers have begun extending them to the video domain [34, 76, 100]. TransVOD families [34, 100] introduce a temporal Transformer to the original Deformable-DETR [103] to fuse both the spatial and temporal information to handle the feature degradation issue. Similarly, PTSEFormer [76] introduces progressive feature aggregation modules to the current Transformer-based image object detectors to boost

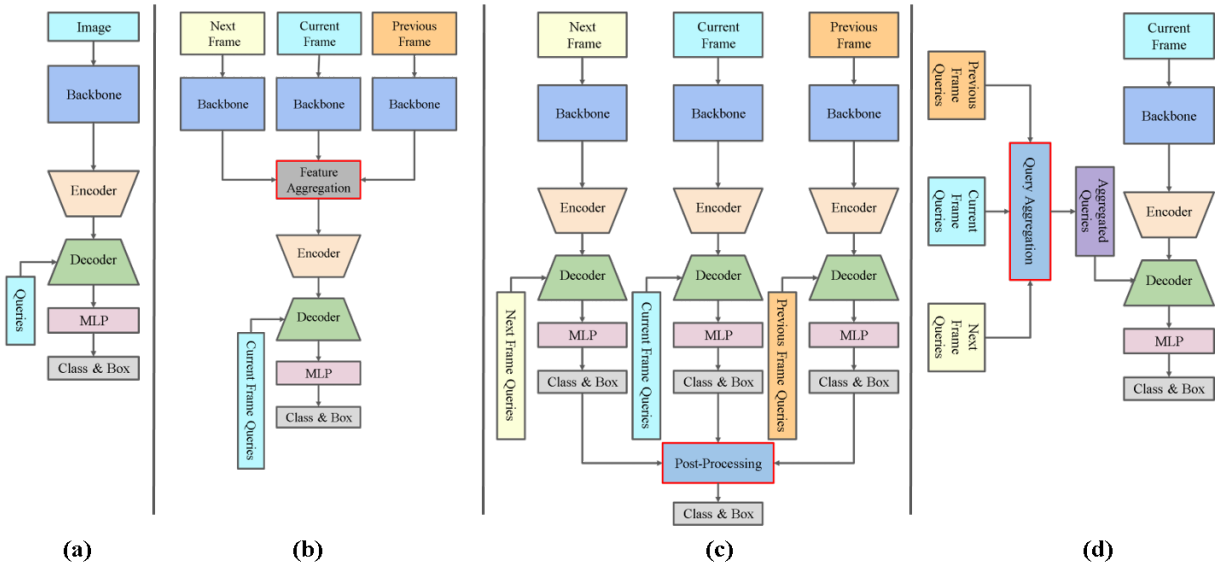

Figure 1. The differences between the existing works and ours. (a) Transformer-based object detectors. (b) Feature-aggregation based video object detectors. (c) Post-processing based video object detectors. (d) Ours. Previous works can be divided into feature-aggregation based (b) and post-processing based (c) models. For Transformer-based models, these works either enhance the features used for detection or the prediction results of each frame. In contrast, our methods (d) pay attention to the aggregation of queries for those Transformer-based object detection models to handle the feature degradation issues.

the performance. Following the TransVOD series [34, 100], we use Transformer-based object detectors as the baseline models in this work.

Unlike the existing models, we take a deeper look at the Transformer-based object detectors and find out the unique properties of their designs. We notice that the queries of Transformer-based object detectors play an essential role in the final prediction performance. Therefore, different from the existing works, which apply different modules to aggregate features (Figure 1(b)) or detection results in every single frame (Figure 1(c)), we introduce a module to aggregate the queries for the Transformer decoder, as shown in Figure 1(d). The existing TransVOD families [34, 100] initialize the spatial and temporal queries randomly regardless of the input frames and then aggregate them after several Transformer layers. Unlike them, our models focus on initializing the object queries and enhancing their qualities of Transformer-based approaches for better performance. By associating and aggregating the initialization of the queries with the input frames, our models can achieve a much better performance compared to the TransVOD families [34, 100] and PTSEFormer [76]. Meanwhile, our methods can be integrated into most of the existing Transformer-based image object detectors to be adaptive to the video domain task. Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to focus on the initialization of queries and aggregate them based on the input features for Transformer-based video object detectors to balance the model efficiency and performance.

- We design a vanilla query aggregation (VQA) module, which enhances the query representations for the Transformer-based object detectors to improve their performance on the video domain tasks. Then we extend it to a dynamic version, which can adaptively generate the initialization of queries and adjust the weights for query aggregation according to the input frames.

- Our proposed method is a plug-and-play module which can be integrated into most of the recent state-of-the-art Transformer-based object detectors for video tasks. Evaluated on the ImageNet VID benchmark, the performance of video object detection can be improved by at least 2.0% on mAP when integrated with our proposed modules.

2. Related Works

Image object detectors. Image object detection requires the model to accurately predict the location and category of the objects in the input image [14, 22, 42, 57, 97, 99]. R-CNN families [15, 32, 65, 65] introduce the basic framework for two-stage object detectors, where proposals are first roughly predicted and then refined for better performance. Following R-CNN families, multiple two-stage works [3, 15, 18, 46] are proposed to improve the perfor-

mance of two-stage object detectors. Besides the detection accuracy, one-stage object detectors [47, 53, 57, 72, 73] are introduced, which generate the predictions without the need for region proposals to improve the inference speeds. Meanwhile, anchor-free based models [23, 61, 72, 73] and point-based methods [22, 42, 101] are proposed, which provides different perspectives to the object detection fields. However, all the models mentioned above require the preprocessing pipeline like anchor design [3, 57, 65] or postprocessing like NMS [55, 60].

Recently, query-based object detectors [4, 17, 19, 24, 27, 36, 56, 66, 71, 81, 87, 103] are proposed, which removes the complicated anchor designs or post-processing pipelines in the traditional models. Among them, Transformer-based models, especially DETR [4] is a pioneer work which treats the object detection task as a bipartite matching problem and introduces a sequence-to-sequence model with 100 randomly initialized queries. Following DETR, multiple works [5, 5, 26, 37, 45, 56, 58, 66, 91, 103] are proposed to improve the inference speed, performance, or convergence efficiency of DETR. For one direction [26, 56, 58, 81, 87], prior knowledge is introduced to the queries for faster convergence. In another way, modulated self-attention modules with fewer operations are proposed to improve the convergence speed of DETR [17, 56, 66, 92, 103]. Meanwhile, multiple training strategies [5, 37, 45, 91] are introduced to improve the convergence speed and performance of the DETR series models. In this work, we mainly focus on Transformer-based image object detectors and design modules to extend them for the tasks in the video domain.

Video object detectors. Depending on the pipeline to improve the detection performance, existing methods can be divided into post-processing [31, 39, 40, 68] and feature-aggregation [6, 12, 13, 34, 49, 51, 54, 76, 79, 82, 84, 96, 100, 104] based models. Post-processing based video object detectors extend the models from the image domain by merging the prediction results according to the temporal information. For example, Seq-NMS [31] associates the bounding boxes from different frames with the IoU threshold; T-CNN [40] links the object detection results from each frame according to optical flows. These methods are not trained end-to-end, and the performances are always sub-optimal. Feature-aggregation based models enhance the feature representations of the current frame by fusing those from the neighboring frames. FGFA [104] and MANet [79] weight average the features from the neighboring frames after being warped based on optical flows. SELSA [82], and Temporal ROI Align [30] fuse the neighboring frames according to their semantic similarity. MEGA [6] takes both temporal information and semantic similarity into account and aggregates local and global features jointly. Different from the methods mentioned above, which generate weights for aggregation using cosine similarity, TF-Blender [13] applies a learnable network to predict the weights.

All the methods mentioned above are designed for the classic object detectors like Faster R-CNN [65], or CenterNet [22]. Recently, with DETR series [4, 26, 66, 103] introduced for the image domain tasks, researchers have begun to focus on how to use these Transformer-based models for video object detection. Among them, TransVOD series [34, 100] introduces two key modules to boost the performance: Temporal Query Encoder to fuse object queries and Temporal Deformable Transformer Decoder (TDTD) to obtain current frame detection results. Similarly, PTSEFormer [76] proposes the Spatial Transition Awareness Model to fuse the temporal and spatial information for a better prediction. These models still follow the feature-aggregation based prototype, where those of the neighboring frames enhance features of the current frame before being fed into the task heads for the final prediction. Unlike these methods mentioned above, we aggregate the queries of those Transformer-based object detectors for the video tasks, which the current researchers have never investigated.

Dynamic models. Dynamic networks aim to adjust the inference paths according to the inputs selectively. Slimmable networks [44, 88, 89] are models which can adaptively select the computational complexity based on the inputs without the need for retraining. In terms of applications for object detection, neural architecture search (NAS) is one of the widely used approaches. NAS-FPN [28] introduces NAS to optimize designing FPNs for object detection according to the input image. In other work like [77, 78, 86], NAS is applied to the existing image object detectors to improve the performance. Besides NAS, recent works have begun using dynamic models [17, 25, 59, 70, 90, 95, 102] to improve efficiency dynamic convolutions [7, 11, 20, 21, 80], dynamic heads [16, 93, 94], and dynamic proposals [14], which are introduced to the existing object detectors to balance the performance and inference speed. For video object detection, DFA [8, 9] proposes a model which can dynamically select the frames used to aggregate the features according to the input frames. This work mainly focuses on dynamically aggregating the queries for the Transformer-based object detectors according to the input frames.

3. Preliminary

We first review the pipeline of the existing Transformer-based object detectors for videos: Given an input frame $\boldsymbol{I}$, the multi-scale features extracted by the backbone like ResNet [33] are denoted as $\boldsymbol{F}$, which are then fed into a Transformer encoder $\mathcal{N}_{enc}$. Next, the outputs of $\mathcal{N}_{enc}$ are fed into a Transformer decoder $\mathcal{N}_{dec}$ together with n randomly initialized queries $\boldsymbol{Q} \in \mathbb{R}^{n \times f}$, where n and f denote the number of queries and length of every query respectively. The outputs of the Transformer decoder are then fed into a task head $\mathcal{N}_t$ for the final prediction $\boldsymbol{P} =$

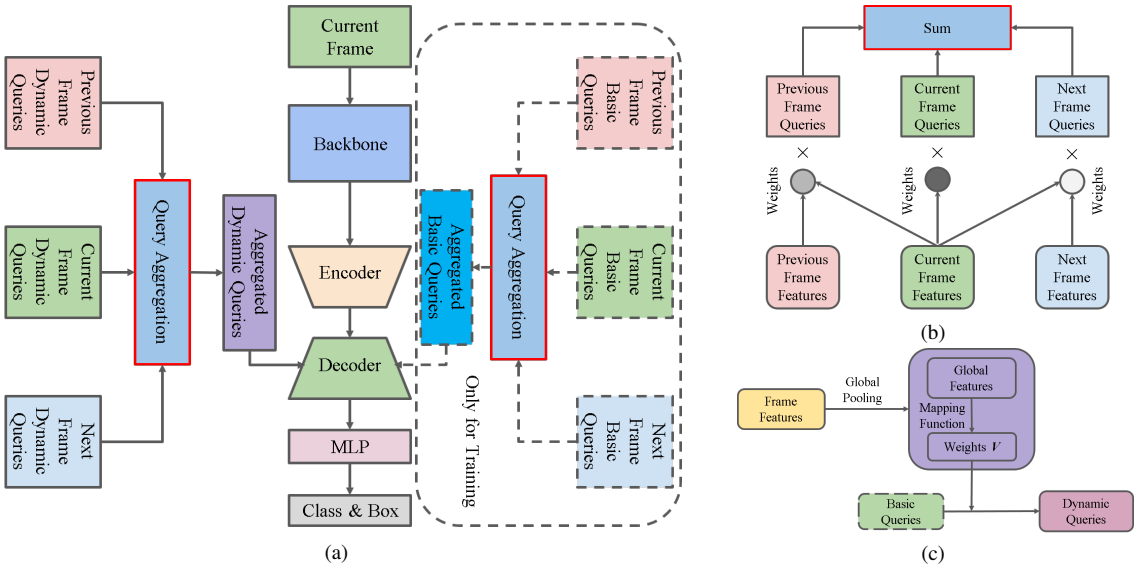

Figure 2. Framework of the proposed method. (a) Dynamic query aggregation with both basic queries and dynamic queries. (b) Details of query aggregation. (c) The process to generate the dynamic queries based on the basic queries and input frame features.

$\{(\boldsymbol{b}_i, \boldsymbol{c}_i), = 1, 2, \ldots, n\}$, where $\boldsymbol{b}_i, \boldsymbol{c}_i$ represent the corresponding locations and categories of the predicted bounding boxes. The process above can be summarized as follows:

$$\boldsymbol{P} = \mathcal{N}_t\left(\mathcal{N}_{dec}\left(\mathcal{N}_{enc}\left(\boldsymbol{F}\right), \boldsymbol{Q}\right)\right) \tag{1}$$

The predictions $\boldsymbol{P}$ are then matched with the ground truths $\boldsymbol{Y}$ using the Hungarian Algorithm [4] for bipartite matching, and the final loss is the summation of all the frames, as Equation 2.

$$\mathcal{L} = \sum \mathcal{L}_{\text{Hungarian}}\left(\boldsymbol{P}, \boldsymbol{Y}\right) \tag{2}$$

To handle the issue of feature degradation in the video frames, existing models enhance the representations of different parts in Equation 1: Post-processing based models [31, 39, 40, 43] associate the predictions $\boldsymbol{P}$ from different frames using temporal hints like optical flows or object tracking to improve the performance of detection at each frame, as Figure 1(c). Feature-aggregation based approaches [34, 79, 82, 100, 104] enhance the feature representations for object detection by weighted averaging the features $\boldsymbol{F}$ neighboring frames, as Figure 1(b). Different from the methods mentioned above, we focus on improving the quality of queries $\boldsymbol{Q}$ by aggregation, as Figure 1(d), which are the unique properties of Transformer-based models compared with the classic approaches. The query aggregation operation is only applied on the first decoder layer.

4. Query Aggregation

4.1. Vanilla Query Aggregation

In terms of how to aggregate the query $\boldsymbol{Q} \in \mathbb{R}^{n \times f}$ of frame $\boldsymbol{I}$, a naive vanilla idea is to weighted average the queries $\boldsymbol{Q}_i$ from the neighboring frame $\boldsymbol{I}$, where $\boldsymbol{I}_i \in \mathcal{N}\left(\boldsymbol{I}\right)$[1] and the size of $\mathcal{N}\left(\boldsymbol{I}_i\right)$ is l, as Figure 2 (b). Within one batch, $l \times n$ queries are randomly initialized and shared across the training data during the training process. Within the neighborhood $\mathcal{N}\left(\boldsymbol{I}_i\right)$, the queries are different. Therefore, the aggregated query $\Delta\boldsymbol{Q}^v \in \mathbb{R}^{n \times f}$ for the current frame $\boldsymbol{I}$ is represented as:

$$\Delta\boldsymbol{Q}^v = \sum_{\forall \boldsymbol{I}_i \in \mathcal{N}(\boldsymbol{I})} w_i^v \boldsymbol{Q}_i, \tag{3}$$

where $w_i^v \in \mathbb{R}$ is the learnable weights for aggregation. A simple idea to generate the learnable weights is based on the cosine similarity of the input frame features, shown as the dots and arrows in Figure 2(b). Following the existing feature-aggregation based video object detectors [79, 82, 104], we generate w_i^v according to Equation 4. We will discuss more ways to aggregate the queries in the experiment section.

$$w_i^v = \frac{\alpha(\boldsymbol{F})\beta(\boldsymbol{F}_i)}{|\alpha(\boldsymbol{F})||\beta(\boldsymbol{F}_i)|}, \tag{4}$$

where α, β are mapping functions and $|\cdot|$ denotes the normal. The corresponding features of the current frame $\boldsymbol{I}$ and its neighbors $\boldsymbol{I}_i$ are denoted as $\boldsymbol{F}$ and $\boldsymbol{F}_i$. Therefore, the process in Equation 1 and 2 are updated as:

$$\begin{aligned} \boldsymbol{P}^v &= \mathcal{N}_t\left(\mathcal{N}_{dec}\left(\mathcal{N}_{enc}\left(\boldsymbol{F}\right), \Delta\boldsymbol{Q}^v\right)\right) \\ \mathcal{L}^v &= \sum \mathcal{L}_{\text{Hungarian}}\left(\boldsymbol{P}^v, \boldsymbol{Y}\right), \end{aligned} \tag{5}$$

[1]For simplicity, $\boldsymbol{I}$ is also considered to be within the neighboring frames $\mathcal{N}\left(\boldsymbol{I}\right)$.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	FPS
Two-stage Object Detectors							
Faster R-CNN [65] + DFF [105]	42.7	70.3	45.7	5.0	17.6	48.7	17.3
Faster R-CNN [65] + FGFA [104]	47.1	74.7	52.0	5.9	22.2	53.1	14.9
Faster R-CNN [65] + SELSA [82]	48.7	78.4	53.1	8.5	26.3	54.5	14.1
Faster R-CNN [65] + Temporal RoI Align [30]	48.5	79.8	52.3	7.2	26.5	54.4	10.5
DETR-based Object Detectors							
SMCA-DETR [26]	53.5	74.2	59.6	7.6	25.7	60.5	13.4
SMCA-DETR [26] + TransVOD [100]	$54.6_{\uparrow 1.1}$	$78.5_{\uparrow 4.3}$	$61.0_{\uparrow 1.4}$	$9.1_{\uparrow 1.5}$	$27.0_{\uparrow 1.3}$	$61.4_{\uparrow 0.9}$	10.1
SMCA-DETR [26] + Ours	$55.8_{\uparrow 2.3}$	$79.1_{\uparrow 4.9}$	$62.7_{\uparrow 3.3}$	$9.3_{\uparrow 1.7}$	$28.7_{\uparrow 3.0}$	$62.6_{\uparrow 2.1}$	10.9
Conditional-DETR [58]	53.7	74.7	60.1	7.7	25.9	60.6	13.1
Conditional-DETR [58] + TransVOD [100]	$54.8_{\uparrow 1.1}$	$78.6_{\uparrow 3.9}$	$61.3_{\uparrow 1.2}$	$9.6_{\uparrow 1.9}$	$27.5_{\uparrow 1.6}$	$61.9_{\uparrow 1.3}$	9.9
Conditional-DETR [58] + Ours	$56.1_{\uparrow 2.4}$	$79.2_{\uparrow 4.5}$	$63.0_{\uparrow 2.9}$	$8.8_{\uparrow 1.1}$	$29.0_{\uparrow 3.1}$	$63.0_{\uparrow 2.4}$	10.3
DAB-DETR [56]	54.2	75.3	61.3	8.9	26.8	61.2	12.0
DAB-DETR [56] + TransVOD [100]	$56.4_{\uparrow 2.2}$	$77.2_{\uparrow 1.9}$	$63.7_{\uparrow 2.4}$	$10.1_{\uparrow 1.2}$	$28.9_{\uparrow 2.1}$	$63.5_{\uparrow 2.3}$	8.7
DAB-DETR [56] + Ours	$58.0_{\uparrow 3.8}$	$79.0_{\uparrow 3.7}$	$65.5_{\uparrow 4.2}$	$12.0_{\uparrow 3.1}$	$30.1_{\uparrow 3.1}$	$65.1_{\uparrow 3.9}$	9.2
Deformable-DETR [103]	55.4	76.2	62.2	10.5	27.5	62.3	15.3
Deformable-DETR [103] + TransVOD [100]	$58.1_{\uparrow 2.7}$	$79.1_{\uparrow 2.9}$	$64.7_{\uparrow 2.5}$	$11.0_{\uparrow 0.5}$	$29.9_{\uparrow 2.4}$	$65.2_{\uparrow 2.9}$	12.1
Deformable-DETR [103] + Ours	$60.1_{\uparrow 4.7}$	$81.7_{\uparrow 5.5}$	$66.9_{\uparrow 4.7}$	$13.2_{\uparrow 2.7}$	$33.1_{\uparrow 5.6}$	$66.9_{\uparrow 4.6}$	12.3

Table 1. Performance comparison with the recent state-of-the-art video object detection approaches on ImageNet VID validation set [67]. The AP_{50} here is the mAP evaluation metric in most of the existing works like TransVOD [100] and PTSEFormer [76].

where $\boldsymbol{P}^v$ denotes the prediction results with the aggregated queries $\Delta\boldsymbol{Q}^v$.

4.2. Dynamic Query Aggregation

The vanilla query aggregation module has an issue that these neighboring queries $\boldsymbol{Q}_i$ are randomly initialized, which are not related to their corresponding frames $\boldsymbol{I}_i$. Therefore, the neighboring queries $\boldsymbol{Q}_i$ do not provide enough temporal or semantic information to eliminate the feature degradation issues caused by fast motion. Though the weights w_i^v used for aggregation are related to the features $\boldsymbol{F}_i$ and $\boldsymbol{F}$, there are not enough constraints on the quantities of those randomly initialized queries $\boldsymbol{Q}_i$.

Therefore, we propose to update the vanilla query aggregation module to a dynamic version, which adds constraints to the queries and can adjust the weights according to the neighboring frames. For implementation, the simple idea is to generate the queries $\boldsymbol{Q}_i$ directly from the features $\boldsymbol{F}_i$ of the input frame. However, experiments show us that this way is challenging to train and always gets a worse performance. Unlike the naive idea mentioned above, we generate the new queries adaptive to the input frame from the randomly initialized queries. We first define two kinds of query vectors, shown with dashed and solid lines in Figure 2 (a) and (c): basic queries $\boldsymbol{Q}_i^b \in \mathbb{R}^{n\times f}$ and dynamic queries $\boldsymbol{Q}_i^d \in \mathbb{R}^{m\times f}$, where $n = rm$ and r is set to be 4 by default. During the training and inference processes, we generate the dynamic queries from the basic queries according to the features $\boldsymbol{F}_i, \boldsymbol{F}$ of the input frames as:

$$\boldsymbol{Q}_i^d = \mathcal{M}\left(\boldsymbol{Q}_i^b, \boldsymbol{F}_i, \boldsymbol{F}\right), \tag{6}$$

where $\mathcal{M}(\cdot)$ is the mapping function to build the relationship of the basic query $\boldsymbol{Q}_i^b$ and the dynamic one $\boldsymbol{Q}_i^d$ according to $\boldsymbol{F}$ and $\boldsymbol{F}_i$. Here we give a default example of the implementation of $\mathcal{M}(\cdot)$ and will analyze its design in the experiment section in detail. We first divide the basic queries $\boldsymbol{Q}_i^b$ into m groups, where each group has r queries. Then, for each group, we use the same weights $\boldsymbol{V} = \{v_j, j = 1, 2, \ldots, r\}$, $\boldsymbol{V} \in \mathbb{R}^{r\times m}$ to weighted average the queries in the current group, as Equation 7.

$$\boldsymbol{Q}_i^d = \sum_{j=1}^{r} v_j \boldsymbol{Q}_{ij}^b, \tag{7}$$

where $\boldsymbol{Q}_{ij}^b$ denotes the j-th basic query in the current group of $\boldsymbol{Q}_i^b$. To build the relationship between the dynamic queries $\boldsymbol{Q}_i^d$ and their corresponding frame $\boldsymbol{I}_i$, we generate the weights $\boldsymbol{V}$ using the global features of $\boldsymbol{I}_i$, as Figure 2(c), denoted as:

$$\boldsymbol{V} = \mathcal{G}\left(\mathcal{A}\left(\boldsymbol{F}_i\right)\right), \tag{8}$$

where $\mathcal{A}$ is a global pooling operation to reduce the feature resolution and generate the global-level features, and $\mathcal{G}$ is a mapping function to project the pooled features to the dimension of $r \times m$. Therefore, the process of aggregating

Model	VQA	DQA	Loss	AP
A				55.4
B	✓			56.7
C		✓		58.5
D			✓	58.9
E		✓	✓	60.1

Table 2. Analysis of the proposed modules.

the queries dynamically based on the features of the input frames can be updated as follows:

$$\Delta \boldsymbol{Q}_i^d = \sum_{\forall \boldsymbol{I}_i \in \mathcal{N}(\boldsymbol{I})} w_i^d \boldsymbol{Q}_i^d \tag{9}$$

During the training, as shown in Figure 2(a), we aggregate both the dynamic queries $\boldsymbol{Q}_i^d$ and basic queries $\boldsymbol{Q}_i^b$ with the same weights w_i^d and generate the corresponding prediction $\boldsymbol{P}^d$ and $\boldsymbol{P}^b$, as Equation 10.

$$\begin{aligned} \Delta \boldsymbol{Q}^b &= \sum_{\forall \boldsymbol{I}_i \in \mathcal{N}(\boldsymbol{I})} w_i^d \boldsymbol{Q}_i^b \\ \boldsymbol{P}^b &= \mathcal{N}_t\left(\mathcal{N}_{dec}\left(\mathcal{N}_{enc}\left(\boldsymbol{F}\right), \Delta \boldsymbol{Q}^b\right)\right) \\ \boldsymbol{P}^d &= \mathcal{N}_t\left(\mathcal{N}_{dec}\left(\mathcal{N}_{enc}\left(\boldsymbol{F}\right), \Delta \boldsymbol{Q}^d\right)\right) \end{aligned} \tag{10}$$

In the training phase of Figure 2, the dynamic query is generated from the same basic query group with the dashed line surrounding it. We calculate the bipartite matching losses for both $\boldsymbol{P}^b$ and $\boldsymbol{P}^d$ and use a hyperparameter γ to balance their influence as Equation 11.

$$\mathcal{L} = \sum \mathcal{L}_{\text{Hungarian}}\left(\boldsymbol{P}^d, \boldsymbol{Y}\right) + \gamma \sum \mathcal{L}_{\text{Hungarian}}\left(\boldsymbol{P}^b, \boldsymbol{Y}\right) \tag{11}$$

During the inference time, we only use the dynamic queries $\boldsymbol{Q}_i^d$ and their corresponding predictions $\boldsymbol{P}^d$ as the final outputs, which introduce only a little extra computational complexity to the original models. We will discuss and analyze the model complexity in the experiment part. The reason why to optimize basic queries in parallel in the training phase and discard them in the inference phase are: 1. It is because of the inference speed. If these two kinds of queries are used in the inference phase, the inference speed will decrease. We do not want to improve the performance by increasing the number of queries at the sacrifice of inference speeds. 2. Our key idea is to aggregate the queries with the guidance of the input frames. Therefore, the dynamic queries are what we finally want for the Transformer decoder. The basic queries are used to help ease the difficulty of training.

5. Experiments

5.1. Experimental Setup

We evaluate our proposed methods on the ImageNet VID benchmark [67] with the recent state-of-the-art Transformer-based object detection models [26,56,58,103]. Following the pipeline of TransVOD [34, 100], we first pre-train our models on the MS COCO [48] and then fine-tune on the combination of ImageNet VID and DET datasets. All the models are trained on 8 Tesla A100 GPUs, and during the training and inference processes, 14 neighboring frames are used for aggregation.

5.2. Main Results

In this section, we conduct experiments on the dynamic query aggregation modules with the current Transformer-based object detectors on the ImageNet VID benchmark [67]. The experiments of vanilla query aggregation will be provided in the following section. For a fair comparison, we use the same experimental setups and compare these models with and without our proposed modules integrated. The default backbone is ResNet-50 [33]. We summarize the results in Table 1. Visualization examples are provided in the supplementary materials.

For most of the DETR-based object detectors, the performance can be improved by at least 2.4% on the metric of mAP compared with those not integrated with our proposed modules. The performance is even better than those integrated with TransVOD [100] by a large margin, which validates the effectiveness of our dynamic query aggregation modules. When considering the objects' sizes, we notice that the performance of large or medium objects is much better than that of small objects. We argue that this is because the process to generate the dynamic queries $\boldsymbol{Q}_i^d$ is only based on the global features, which lack enough information for the small objects.

5.3. Model Analysis

We conduct experiments with Deformable-DETR [103] on the ImageNet VID benchmark [67] in this section to study the design of our proposed modules, mainly the dynamic query aggregation module. More experiment results are provided in the supplementary materials.

Analysis of $\mathcal{M}$. We conduct experiments to analyze the design of $\mathcal{M}$, as Table 3. Model A is our default setting where only one $\boldsymbol{V}$ is generated, and r is set to be 4. Model B and C increase the number of weight matrix $\boldsymbol{V}$ to be 2 and 4. In detail, multiple outputs are provided in each group of $\boldsymbol{Q}_i^b$ instead of generating only one $\boldsymbol{Q}_i^d$. Model D replaces the $\mathcal{G}$ and $\mathcal{A}$ operations in Equation 8 to MLPs, and Model E generates $\boldsymbol{Q}_i^d$ from $\boldsymbol{Q}_i^b$ and $\boldsymbol{F}_i$ using multi-head cross attention. Models F and G follow model A but have different ways of grouping the basic queries. Model F shuffles the

Model	AP	AP_{50}	AP_{75}	Extra Param
A	60.1	81.7	66.9	37K
B	60.3	82.0	67.0	74K
C	60.5	82.3	67.2	147K
D	59.3	80.7	65.8	49K
E	58.2	79.3	64.9	26K
F	60.0	81.7	66.8	37K
G	60.1	81.6	67.0	37K

Table 3. Analysis of different designs on $\mathcal{M}$.

Aggregation Methods	AP	AP_{50}	AP_{75}
Cosine Similarity	58.7	79.3	64.9
Simple Networks	59.5	80.2	65.3
Transformer	60.1	81.7	66.9

Table 4. Analysis of different ways to aggregate queries.

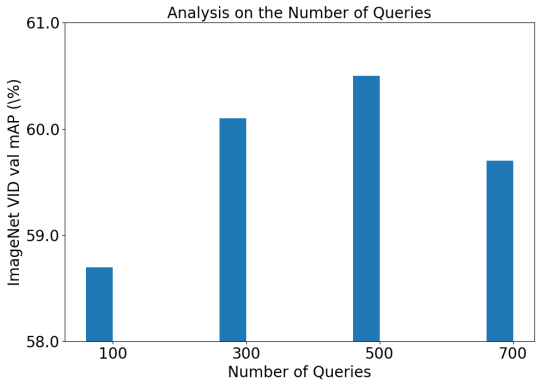

Figure 3. Model analysis on the number of queries. The default model is Deformable-DETR [103] with ResNet-50 as the backbone.

queries before grouping, and model G randomly clusters non-consecutive queries into a group. From the table, we notice that by increasing the numbers of $\boldsymbol{V}$, there is a slight improvement in the performance at the sacrifice of more extra parameters. For models D and E, the performance drops a little bit. We argue that this is because local features from $\boldsymbol{F}_i$ will bring misleading information and make the model difficult to optimize. For models F and G, there is not much difference with model A because the basic queries $\boldsymbol{Q}_i^b$ are randomly initialized, bringing robustness to our models.

Analysis of aggregation process. In section 4, we provide a way to implement the aggregation process based on cosine similarity as the existing works [6, 79, 82, 104]. Here, we analyze different ways to aggregate the queries as Table 4. Besides cosine similarity, we use simple learnable networks as TF-Blender [13] and Transformer as TransVOD [34, 100]. For a fair comparison, all the models are trained with 14 neighboring frames and tested on the ImageNet VID benchmark `val` split. For the implementation details, we use the same structures as TF-Blender [13] and TransVOD [34, 100]. The table shows that cosine similarity based aggregation has the worst performance compared to learnable simple networks and Transformers. By default, we use the Transformer to aggregate the queries in our work.

Analysis of each component. We conduct experiments to study the effects of each proposed component. We denote the original Deformable-DETR [103] with ResNet-50 as the backbone of model A. Then, we introduce the vanilla query aggregation modules with 14 neighboring frames to the original Deformable-DETR to get model B. For model C, we change the vanilla query aggregation module to the dynamic version without the extra loss. To validate the effectiveness of our proposed modules, we conduct experiments on mode D, where we use two groups of randomly initialized $\boldsymbol{Q}_i^b$ and $\boldsymbol{Q}_i^d$. We do not build a relationship between them. Therefore, there is only a loss for these two groups of queries but no relation to the input frames. Finally, we integrate the dynamic query aggregation module and the extra loss to get model E. The results are summarized in Table 2.

From the table, we notice that by only using the vanilla query aggregation module, the performance can be improved by 1.3% compared with the original Deformable-DETR without aggregation. However, this is worse than the original TransVOD [100] model. Updating the vanilla query aggregation module to the dynamic version increases the performance by 1.8%, which is better than the original TransVOD. We argue that by introducing the contents of the inputs into the queries, the performance can be improved by a large margin. Model D removes the relations between the basic and dynamic queries but leaves both losses for them. It also performs well, though different than model E, which contains both the dynamic query aggregation module and the extra losses. From the experiments, we notice that by either introducing the dynamic query aggregation or adding two separate groups of randomly initialized queries, though they are not related to the inputs, the performance can also be improved. By combining these two modules, our proposed methods achieve the best performance.

Analysis of the number of queries. We conduct experiments with dynamic query aggregation on the number of queries as Figure 3. From the figure, by increasing the number of queries, the performance of the video object detection will be improved accordingly. However, when the number of queries is more than 500, the performance begins to be saturated. We argue that this is because there are enough queries to represent the objects in the input frames, and some are redundant.

Analysis of model complexity. Here, we analyze the model complexity of our proposed modules to the existing object detectors. These methods mainly have four compu-

Model	r	mAP	AP_{50}
Deformable-DETR	-	55.4	76.2
Deformable-DETR + Ours	1	58.7	80.9
	2	59.4	81.2
	4	60.1	81.7
	8	59.7	81.5

Table 5. Analysis of the effect of r. Experiments are conducted with Deformable-DETR [103] using ResNet-50 as the backbone.

tational loads: 1. Feature extraction network from the backbone $\mathcal{N}_{ex}$; 2. Transformer encoder networks $\mathcal{N}_{enc}$; 3. Transformer decoder networks $\mathcal{N}_{dec}$; 4. task network $\mathcal{N}_{tk}$. Therefore, the total computational complexity is:

$$\mathcal{O}\left(\mathcal{N}_{tk}\right)+\mathcal{O}\left(\mathcal{N}_{dec}\right)+\mathcal{O}\left(\mathcal{N}_{enc}\right)+\mathcal{O}\left(\mathcal{N}_{ex}\right) \tag{12}$$

In our proposed models, we introduce a tiny network $\mathcal{M}$ to generate the dynamic queries $\boldsymbol{Q}_i^d$. Therefore, during the training process, the computational complexity of our model is defined as:

$$\begin{aligned}&\mathcal{O}\left(\mathcal{N}_{tk}\right)+\mathcal{O}\left(\mathcal{N}_{dec}\right)*(r+1)\\&+\mathcal{O}\left(\mathcal{N}_{enc}\right)+\mathcal{O}\left(\mathcal{M}\right)+\sum\mathcal{O}\left(\mathcal{N}_{ex}\right),\end{aligned} \tag{13}$$

where $r+1$ represents the extra loss calculated for aggregation (both $\boldsymbol{Q}_i^d$ and $\boldsymbol{Q}_i^b$). Typically, in the Transformer-based object detectors, $\mathcal{O}\left(\mathcal{M}\right) \ll \mathcal{O}\left(\mathcal{N}_{ex}\right) \approx \mathcal{O}\left(\mathcal{N}_{dec}\right) < \mathcal{O}\left(\mathcal{N}_{enc}\right)$. Therefore, our model does not increase too many computational loads to the existing models. When it comes to the inference process, the computational complexity is updated to be:

$$\mathcal{O}\left(\mathcal{N}_{tk}\right)+\mathcal{O}\left(\mathcal{N}_{dec}\right)+\mathcal{O}\left(\mathcal{N}_{enc}\right)+\mathcal{O}\left(\mathcal{M}\right)+\sum\mathcal{O}\left(\mathcal{N}_{ex}\right), \tag{14}$$

since only the dynamic queries $\boldsymbol{Q}_i^d$ are taken into account. The increasing computational load is affordable since the impact of $\mathcal{O}\left(\mathcal{M}\right)+\sum\mathcal{O}\left(\mathcal{N}_{ex}\right)$ is negligible compared to those of the other networks structures like $\mathcal{O}\left(\mathcal{N}_{dec}\right)$.

Analysis of r. We conduct experiments to analyze the effect of r on the final performance. By default, m is set to be 300 for the dynamic queries so that the number of queries will not affect the final performance during the inference time. We change the value of r to use different numbers of basic queries $\boldsymbol{Q}_i^b$ to generate the same number of dynamic queries $\boldsymbol{Q}_i^d$, and the results are summarized in Table 5. The table shows that the performance is the worst when r is set to 1. We argue that a limited number of basic queries are not enough to generate the dynamic queries adaptive to the input frames. However, the performance will be saturated when r is set to be 8. We think this is because there are enough and even redundant basic queries to generate the dynamic ones.

Figure 4. Visualization of dynamic queries with TSNE [74]. Please zoom in for better visualization.

Analysis of $\boldsymbol{Q}_i^b$ and $\boldsymbol{Q}_i^d$. To better understand our proposed method, we analyze and visualize the queries from the original models and the dynamic queries from our models. We choose 100 video clips from the ImageNet VID benchmark [67] and sample 14 frames from each video. We generate the corresponding queries based on the input frames and visualize them using the TSNE [74] as Figure 4. For the original model, the queries are always the same regardless of the input frames[2]. Regarding our dynamic queries, as Figure 4, queries within the same video clips share similar representations, like the clusters on the top and left, which are better and easier to improve the performance of video object detection.

6. Conclusion

In this paper, we discuss the unique property of the existing Transformer-based image object detectors and introduce a plug-and-play module designed specifically for these models for the video domain tasks. We first introduce a vanilla version to aggregate the queries for the decoders of the existing Transformer-based models to improve the performance of video object detection. Then, we extend the vanilla query aggregation module to a dynamic version which builds the relationships between the queries and the features of the input frame. Extensive experiments demonstrate that, when integrated with our proposed modules, the current state-of-the-art Transformer-based image object detectors can perform much better on the video object detection task. We believe our proposed modules can bring some light to the Transformer-based models for the video tasks.

[2]It is the same with our basic queries. Since all the input frames share the same queries, we do not visualize them.

References

[1] Hatem Belhassen, Heng Zhang, Virginie Fresse, and El-Bay Bourennane. Improving video object detection by seq-bbox matching. In *VISIGRAPP*, 2019. 1

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, 2017. 1

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2019. 1, 2, 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 3, 4

[5] Qiang Chen, Xiaokang Chen, Gang Zeng, and Jingdong Wang. Group detr: Fast training convergence with decoupled one-to-many label assignment. *arXiv preprint arXiv:2207.13085*, 2022. 3

[6] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *CVPR*, 2020. 1, 3, 7

[7] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *CVPR*, 2020. 3

[8] Yiming Cui. Dfa: Dynamic feature aggregation for efficient video object detection. *arXiv preprint arXiv:2210.00588*, 2022. 3

[9] Yiming Cui. Dynamic feature aggregation for efficient video object detection. In *ACCV*, 2022. 3

[10] Yiming Cui, Zhiwen Cao, Yixin Xie, Xingyu Jiang, Feng Tao, Yingjie Victor Chen, Lin Li, and Dongfang Liu. Dg-labeler and dgl-mots dataset: Boost the autonomous driving perception. In *WACV*, 2022. 1

[11] Yiming Cui, Xin Liu, Hongmin Liu, Jiyong Zhang, Alina Zare, and Bin Fan. Geometric attentional dynamic graph convolutional neural networks for point cloud analysis. *Neurocomputing*, 432:300–310, 2021. 3

[12] Yiming Cui, Yi Shen, Xin Zhang, Yan Wang, and Miao Zhang. Electric differential control for electric vehicles based on emd method. In *I2MTC*, 2016. 3

[13] Yiming Cui, Liqi Yan, Zhiwen Cao, and Dongfang Liu. Tf-blender: Temporal feature blender for video object detection. In *ICCV*, 2021. 1, 3, 7

[14] Yiming Cui, Linjie Yang, and Ding Liu. Dynamic proposals for efficient object detection. *arXiv preprint arXiv:2207.05252*, 2022. 1, 2, 3

[15] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *NeurIPS*, 29, 2016. 1, 2

[16] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *CVPR*, 2021. 3

[17] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *ICCV*, 2021. 3

[18] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 2

[19] Bin Dong, Fangao Zeng, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Solq: Segmenting objects by learning queries. *NeurIPS*, 2021. 3

[20] Guimin Dong, Mingyue Tang, Lihua Cai, Laura E Barnes, and Mehdi Boukhechba. Semi-supervised graph instance transformer for mental health inference. In *ICMLA*, 2021. 3

[21] Guimin Dong, Mingyue Tang, Zhiyuan Wang, Jiechao Gao, Sikun Guo, Lihua Cai, Robert Gutierrez, Bradford Campbell, Laura E Barnes, and Mehdi Boukhechba. Graph neural networks in iot: A survey. *ACM Transactions on Sensor Networks (TOSN)*, 2022. 3

[22] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019. 1, 2, 3

[23] Kaiwen Duan, Lingxi Xie, Honggang Qi, Song Bai, Qingming Huang, and Qi Tian. Corner proposal network for anchor-free, two-stage object detection. In *ECCV*, 2020. 3

[24] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *ICCV*, 2021. 1, 3

[25] Mingfei Gao, Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. Dynamic zoom-in network for fast object detection in large images. In *CVPR*, 2018. 3

[26] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *ICCV*, 2021. 1, 3, 5, 6

[27] Ziteng Gao, Limin Wang, Bing Han, and Sheng Guo. Adamixer: A fast-converging query-based object detector. In *CVPR*, 2022. 1, 3

[28] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *CVPR*, 2019. 3

[29] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 1

[30] Tao Gong, Kai Chen, Xinjiang Wang, Qi Chu, Feng Zhu, Dahua Lin, Nenghai Yu, and Huamin Feng. Temporal roi align for video object recognition. In *AAAI*, 2021. 1, 3, 5

[31] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S Huang. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465*, 2016. 1, 3, 4

[32] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6

[34] Lu He, Qianyu Zhou, Xiangtai Li, Li Niu, Guangliang Cheng, Xiao Li, Wenxuan Liu, Yunhai Tong, Lizhuang Ma, and Liqing Zhang. End-to-end video object detection with spatial-temporal transformers. In *ACM Multimedia*, 2021. 1, 2, 3, 4, 6, 7

[35] Qinghang Hong, Fengming Liu, Dong Li, Ji Liu, Lu Tian, and Yi Shan. Dynamic sparse r-cnn. In *CVPR*, 2022. 1

[36] Jie Hu, Liujuan Cao, Yao Lu, ShengChuan Zhang, Yan Wang, Ke Li, Feiyue Huang, Ling Shao, and Rongrong Ji. Istr: End-to-end instance segmentation with transformers. *arXiv preprint arXiv:2105.00637*, 2021. 3

[37] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. *arXiv preprint arXiv:2207.13080*, 2022. 3

[38] Zhengkai Jiang, Yu Liu, Ceyuan Yang, Jihao Liu, Peng Gao, Qian Zhang, Shiming Xiang, and Chunhong Pan. Learning where to focus for efficient video object detection. In *ECCV*, 2020. 1

[39] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2896–2907, 2017. 1, 3, 4

[40] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. *CVPR*, 2016. 1, 3, 4

[41] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *CVPR*, 2020. 1

[42] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018. 1, 2, 3

[43] Byungjae Lee, Enkhbayar Erdenee, Songguo Jin, Mi Young Nam, Young Giu Jung, and Phill Kyu Rhee. Multi-class multi-object tracking using changing point detection. In *ECCV*, 2016. 4

[44] Changlin Li, Guangrun Wang, Bing Wang, Xiaodan Liang, Zhihui Li, and Xiaojun Chang. Dynamic slimmable network. In *CVPR*, 2021. 3

[45] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, 2022. 3

[46] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 2

[47] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1, 3

[48] 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 *ECCV*, 2014. 6

[49] Dongfang Liu, Yiming Cui, Zhiwen Cao, and Yingjie Chen. Indoor navigation for mobile agents: A multimodal vision fusion model. In *IJCNN*, 2020. 3

[50] Dongfang Liu, Yiming Cui, Zhiwen Cao, and Yingjie Chen. A large-scale simulation dataset: Boost the detection accuracy for special weather conditions. In *IJCNN*, 2020. 1

[51] Dongfang Liu, Yiming Cui, Yingjie Chen, Jiyong Zhang, and Bin Fan. Video object detection for autonomous driving: Motion-aid feature calibration. *Neurocomputing*, 409:1–11, 2020. 3

[52] Dongfang Liu, Yiming Cui, Xiaolei Guo, Wei Ding, Baijian Yang, and Yingjie Chen. Visual localization for autonomous driving: Mapping the accurate location in the city maze. In *ICPR*, 2021. 1

[53] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *CVPR*, 2021. 3

[54] Dongfang Liu, Yiming Cui, Liqi Yan, Christos Mousas, Baijian Yang, and Yingjie Chen. Densernet: Weakly supervised visual localization using multi-scale feature aggregation. In *AAAI*, 2021. 3

[55] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd. In *CVPR*, 2019. 1, 3

[56] 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. 1, 3, 5, 6

[57] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 1, 2, 3

[58] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, 2021. 3, 5, 6

[59] Qi Ming, Zhiqiang Zhou, Lingjuan Miao, Hongwei Zhang, and Linhao Li. Dynamic anchor learning for arbitrary-oriented object detection. In *AAAI*, 2021. 3

[60] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *ICPR*, 2006. 1, 3

[61] Zhengquan Piao, Junbo Wang, Linbo Tanga, Baojun Zhao, and Wenzheng Wang. Accloc: Anchor-free and two-stage detector for accurate object localization. *Pattern Recognition*, 126:108523, 2022. 3

[62] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1

[63] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017. 1

[64] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[65] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 1, 2, 3, 5

[66] Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse detr: Efficient end-to-end object detection with learnable sparsity. *arXiv preprint arXiv:2111.14330*, 2021. 1, 3

[67] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 1, 5, 6, 8

[68] Alberto Sabater, Luis Montesano, and Ana C Murillo. Robust and efficient post-processing for video object detection. In *IROS*, 2020. 1, 3

[69] Mohammad Javad Shafiee, Brendan Chywl, Francis Li, and Alexander Wong. Fast yolo: A fast you only look once system for real-time embedded object detection in video. *arXiv preprint arXiv:1709.05943*, 2017. 1

[70] Lin Song, Yanwei Li, Zhengkai Jiang, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng. Fine-grained dynamic head for object detection. *NeurIPS*, 2020. 3

[71] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *CVPR*, 2021. 1, 3

[72] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, 2019. 1, 3

[73] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: A simple and strong anchor-free object detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 3

[74] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8

[75] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019. 1

[76] Han Wang, Jun Tang, Xiaodong Liu, Shanyan Guan, Rong Xie, and Li Song. Ptseformer: Progressive temporal-spatial enhanced transformer towards video object detection. *arXiv preprint arXiv:2209.02242*, 2022. 1, 2, 3, 5

[77] Ning Wang, Yang Gao, Hao Chen, Peng Wang, Zhi Tian, Chunhua Shen, and Yanning Zhang. Nas-fcos: Fast neural architecture search for object detection. In *CVPR*, 2020. 3

[78] Ning Wang, Yang Gao, Hao Chen, Peng Wang, Zhi Tian, Chunhua Shen, and Yanning Zhang. Nas-fcos: Efficient search for object detection architectures. *International Journal of Computer Vision*, 129(12):3299–3312, 2021. 3

[79] Shiyao Wang, Yucong Zhou, Junjie Yan, and Zhidong Deng. Fully motion-aware network for video object detection. In *ECCV*, 2018. 1, 3, 4, 7

[80] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 3

[81] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, 2022. 3

[82] Haiping Wu, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Sequence level semantics aggregation for video object detection. In *ICCV*, 2019. 1, 3, 4, 5, 7

[83] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 1

[84] Liqi Yan, Qifan Wang, Yiming Cui, Fuli Feng, Xiaojun Quan, Xiangyu Zhang, and Dongfang Liu. Gl-rg: Global-local representation granularity for video captioning. *arXiv preprint arXiv:2205.10706*, 2022. 3

[85] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 1

[86] Lewei Yao, Hang Xu, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Sm-nas: Structural-to-modular neural architecture search for object detection. In *AAAI*, 2020. 3

[87] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021. 3

[88] Jiahui Yu and Thomas Huang. Autoslim: Towards one-shot architecture search for channel numbers. *arXiv preprint arXiv:1903.11728*, 2019. 3

[89] Jiahui Yu and Thomas S Huang. Universally slimmable networks and improved training techniques. In *ICCV*, 2019. 3

[90] Hongkai Zhang, Hong Chang, Bingpeng Ma, Naiyan Wang, and Xilin Chen. Dynamic r-cnn: Towards high quality object detection via dynamic training. In *ECCV*, 2020. 3

[91] 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. *arXiv preprint arXiv:2203.03605*, 2022. 3

[92] Jingyi Zhang, Jiaxing Huang, Zhipeng Luo, Gongjie Zhang, and Shijian Lu. Da-detr: Domain adaptive detection transformer by hybrid attention. *arXiv preprint arXiv:2103.17084*, 2021. 3

[93] Miao Zhang, Wei Guo, Yiming Cui, Fei Shen, and Yi Shen. Manifold learning based supervised hyperspectral data classification method using class encoding. In *IGARSS*, 2016. 3

[94] Miao Zhang, Zheqi Lin, Yiming Cui, Fei Shen, and Yi Shen. Multiclassification method for hyperspectral data based on chernoff distance and pairwise decision tree strategy. In *IGARSS*, 2016. 3

[95] Miao Zhang, Jie Liu, Yifei Wang, Yongri Piao, Shunyu Yao, Wei Ji, Jingjing Li, Huchuan Lu, and Zhongxuan Luo. Dynamic context-sensitive filtering network for video salient object detection. In *ICCV*, 2021. 3

[96] Xin Zhang, Yiming Cui, Yan Wang, Mingjian Sun, and Hengshan Hu. An improved ae detection method of rail defect based on multi-level anc with vss-lms. *Mechanical Systems and Signal Processing*, 99:420–433, 2018. 3

[97] Zhengming Zhang and Renran Tian. Studying battery range and range anxiety for electric vehicles based on real travel demands. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 65, pages 332–336. SAGE Publications Sage CA: Los Angeles, CA, 2021. 2

[98] Zhengming Zhang, Renran Tian, and Zhengming Ding. Trep: Transformer-based evidential prediction for pedestrian intention with uncertainty. 1

[99] Zhengming Zhang, Renran Tian, Rini Sherony, Joshua Domeyer, and Zhengming Ding. Attention-based interrelation modeling for explainable automated driving. *IEEE Transactions on Intelligent Vehicles*, 2022. 2

[100] Qianyu Zhou, Xiangtai Li, Lu He, Yibo Yang, Guangliang Cheng, Yunhai Tong, Lizhuang Ma, and Dacheng Tao. Transvod: End-to-end video object detection with spatial-temporal transformers. *arXiv preprint arXiv:2201.05047*, 2022. 1, 2, 3, 4, 5, 6, 7

[101] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019. 3

[102] Mingjian Zhu, Kai Han, Changbin Yu, and Yunhe Wang. Dynamic feature pyramid networks for object detection. *arXiv preprint arXiv:2012.00779*, 2020. 3

[103] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1, 3, 5, 6, 7, 8

[104] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, 2017. 1, 3, 4, 5, 7

[105] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *CVPR*, 2017. 1, 5