TransVOD: End-to-End Video Object Detection with Spatial-Temporal Transformers

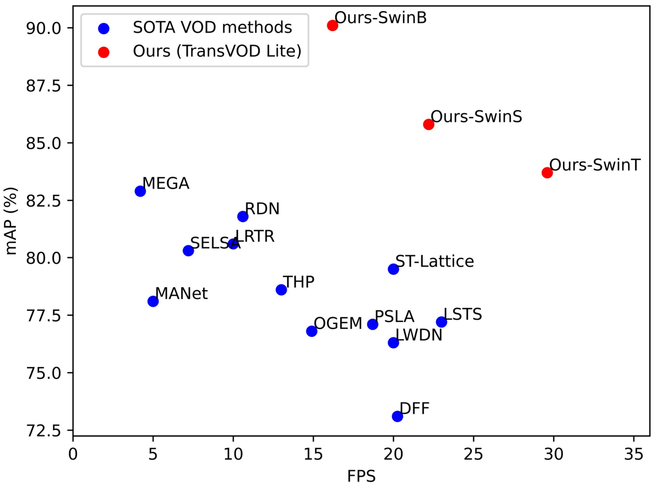
Qianyu Zhou*, Xiangtai Li*, Lu He*, Yibo Yang, Guangliang Cheng,
Yunhai Tong, Lizhuang Ma†, Dacheng Tao, *Fellow*, IEEE

Abstract—Detection Transformer (DETR) and Deformable DETR have been proposed to eliminate the need for many hand-designed components in object detection while demonstrating good performance as previous complex hand-crafted detectors. However, their performance on Video Object Detection (VOD) has *not* been well explored. In this paper, we present **TransVOD**, the *first end-to-end* video object detection system based on spatial-temporal Transformer architectures. The first goal of this paper is to streamline the pipeline of VOD, effectively removing the need for many hand-crafted components for feature aggregation, *e.g.,* optical flow model, relation networks. Besides, benefited from the object query design in DETR, our method does not need complicated post-processing methods such as Seq-NMS. In particular, we present a temporal Transformer to aggregate both the spatial object queries and the feature memories of each frame. Our temporal transformer consists of two components: Temporal Query Encoder (TQE) to fuse object queries, and Temporal Deformable Transformer Decoder (TDTD) to obtain current frame detection results. These designs boost the strong baseline deformable DETR by a significant margin (3 %-4 % mAP) on the ImageNet VID dataset. TransVOD yields comparable performances on the benchmark of ImageNet VID. Then, we present two improved versions of TransVOD including TransVOD++ and TransVOD Lite. The former fuses object-level information into object query via dynamic convolution while the latter models the entire video clips as the output to speed up the inference time. We give detailed analysis of all three models in the experiment part. In particular, our proposed TransVOD++ sets a new state-of-the-art record in terms of accuracy on ImageNet VID with 90.0 % mAP. Our proposed TransVOD Lite also achieves the best speed and accuracy trade-off with 83.7 % mAP while running at around 30 FPS on a single V100 GPU device. Code and models will be available for further research.

Index Terms—Video Object Detection, Vision Transformers, Scene Understanding, Video Understanding.

1 Introduction

Object detection is a fundamental task in computer vision and achieves a huge progress with deep convolution neural networks [1], [2], [3], [4], [5], [6], [7], [8], [9]. It enables various applications in the real world, *e.g.,* autonomous driving. Recently, DETR like detectors [10], [11] remove complex components such as NMS which makes the object detection a sparse set prediction problem and achieve competitive results. However, all these still-image detectors cannot be directly applied to video data, due to the appearance deterioration and changes of video frames, *e.g.,* motion blur, part occlusion, and rare poses. Thus, video object detection (VOD) aims to detect all objects given a video clip. Previous video object detection approaches mainly leverage temporal information in two different manners. The first one relies on post-processing of temporal information to make the object detection results more coherent and stable [12], [13], [14], [15]. These methods usually apply a still-image detector to obtain detection results, then associate the results. Another line of approaches [16], [17], [18], [19], [20], [21], [22], [23], [24] exploits the feature aggregation of temporal information. Specifically, they mainly improve features of the current frame by aggregating that of adjacent frames or entire clips to boost the detection performance via specific operator design. In this way, the problems such as motion blur, part occlusion, and fast appearance change can be well solved. In particular, most of these methods [22], [23], [24], [25] use two-



Video object detection (VOD) results on ImageNet VID

Fig. 1: Speed and Accuracy trade-off of video object detection (VOD) results in ImageNet VID. The blue points plot the state-of-the-art (SOTA) VOD methods, and the red ones are our proposed method TransVOD Lite, achieving the **best** trade-off between the speed and accuracy with different backbones. SwinB, SwinS and SwinT mean Swin Base, Small and Tiny respectively.

- * *indicates equal contribution.* † *indicates corresponding author.*
- *Q. Zhou, L. He and L.Ma are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: {zhouqianyu, 147258369}@sjtu.edu.cn, ma-lz@cs.sjtu.edu.cn).*
- *X. Li and Y. Tong are with the School of Artificial Intelligence, Peking University, Beijing 100871, China (e-mail: lxtpku@pku.edu.cn).*
- *Y. Yang and D. Tao are with JD Explore Academy, Beijing, China.*
- *G. Cheng is with the SenseTime Research, Beijing, China (e-mail: guangliangcheng2014@gmail.com).*

stage detector Faster-RCNN [1] or R-FCN [4] as the still-image baseline.

Despite the gratifying success of these approaches, most of the two-stage pipelines for video object detection are over sophisticated, requiring many hand-crafted components, *e.g.,* optical flow model [26], [27], [28], [29], [30], recurrent neural network [23], [25], [31], deformable convolution fusion [21], [32], [33], relation networks [23], [34], [35]. In addition, most of them need complicated post-processing methods by linking the same object across the video to form tubelets and aggregating classification scores in the tubelets to achieve the state-of-the-art performance [12], [13], [14], [15]. Meanwhile, there are also several related studies [16], [17], [33], [36], [37], [38] focusing on real-time video object detection. However, these works still need sophisticated designs. Thus, it is *in desperate need to build* ***a simple yet effective*** *VOD framework in a fully end-to-end manner*.

Transformers [10], [11], [39], [40], [41] have shown promising potential in computer vision. Especially, DETR [10], [11] simplifies the detection pipeline by modeling the object queries and achieving comparative performance with highly optimized CNN-based detectors. However, given a video clip, such static detectors cannot handle motion blur, part occlusion, video defocus, or rare poses well due to the lack of temporal information, which will be shown in the experiment part. Thus, how to model the temporal information in a long-range video clip is a very critical problem.

In this paper, our goal is to extend the DETR-like object detection into the video object detection domain. Our insights are four aspects. Firstly, we observe that the video clip contains rich inherent temporal information, *e.g.,* rich visual cues of motion patterns. Thus, it is natural to view video object detection as a sequence-to-sequence task with the advantages of Transformers [42]. The whole video clip is like a sentence, and each frame contributes similarly to each word in natural language processing. Transformers can not only be used in inner each frame to model the interaction of each object but also be used to link objects along the temporal dimension. Secondly, object query is one key component design in DETR [10] which encodes instance-aware information. The learning process of DETR can be seen as the grouping process: grouping each object into an object query. Thus, these query embeddings can represent the instances of each frame and it is natural to link these sparse query embeddings via another temporal transformer. Thirdly, the output memory from the DETR transformer encoder contains rich spatial information which can also be modeled jointly with query embeddings along the temporal dimension. Fourthly, adopting clip-level inputs of Transformers can speed up the detection process in a video which is needed in many real application cases.

Motivated by these facts, we propose TransVOD, a new end-to-end video object detection framework based on a spatial-temporal Transformer architecture. Our TransVOD views video object detection as an end-to-end sequence decoding/prediction problem. For the current frame, as shown in Fig. (2)(a), it takes multiple frames as inputs and directly outputs the current frame detection results via a Transformer-like architecture. In particular, we design a novel temporal Transformer to link each object query and memory encoding outputs simultaneously. Our proposed temporal Transformer mainly contains three components: Temporal Deformable Transformer Encoder (TDTE) to encode the multiple frame spatial details, Temporal Query Encoder (TQE) to fuse object query in one video clip, and Temporal Deformable Transformer Decoder (TDTD) to obtain the final detection results of the current frame. TDTE efficiently aggregates spatial information via Temporal Deformable Attention and avoids background noise. TQE first adopts a coarse-to-fine strategy to select relevant object queries in one clip and fuse such selected queries via several self-attention layers [42]. TDTD is another decoder that takes the outputs of TDTE and TQE, and outputs the final detection results. These modules are shared for each frame and can be trained in an end-to-end manner. We carry out extensive experiments on ImageNet VID dataset [43]. Compared with the single-frame baseline [11], our TransVOD achieves significant improvements (2%-4% mAP).

Based on the TransVOD framework, which is published in ACM MM 2021 [44], we present two improved versions including TransVOD++ and TransVOD Lite. For TransVOD++, regarding that there exists large redundancy in both the number of object queries and the targets, we present a hard query mining (HQM) strategy to sample the hardest queries during the training inspired from the hard pixels mining in image object detection and segmentation [5], [45], [46], as shown in Fig. 2(b). Moreover, we present a novel query and RoI fusion (QRF) module via dynamic convolutions. In this way, the object-level appearance information is injected into each query and TDTE can be avoided since the spatial fusion can be replaced with QRF. Compared with previous TransVOD, we find both improvements lead to better results with faster speed. Moreover, when deploying the Vision Transformer backbone [47], we present a simply-aligned fusion to fuse multi-scale features for TDTD. After adopting Swin base as the backbone, our TransVOD++ achieves 90% mAP on the ImageNet VID dataset and suppress previous works by a significant margin (5-6%) with a simpler pipeline. Our method is **the first to achieve 90% mAP on ImageNet VID dataset**.

Inherited from TranVOD, we present TransVOD Lite, aiming at real-time VOD and modeling the VOD task as a sequence-to-sequence prediction problem which is adopted in machine translation [42]. The pipeline is shown in Fig. 2(c). In particular, given a window size T (T can be chosen in 8, 16), we take multiple frames as inputs and obtain multiple frame results simultaneously. Then, one video clip results can be obtained in a temporal window manner. In this way, we can fully use the memory of GPU to speed up inference time. Our TransVOD Lite can boost the single image baseline by 2-3% mAP but with a faster speed (4x-6x). After adopting the Swin Transformer, as shown in Fig. 1, our methods achieve the best speed and accuracy trade-off. Our methods lead to a significant margin (3%-4%mAP, 5-15 FPS) compared with previous VOD methods in both speed and accuracy. Our best model can achieve 83.7 % mAP while running at around 30 FPS. In summary, following the TransVOD framework, we present TransVOD++ and TransVOD Lite. Both models set new state-of-the-art results on the challenging ImageNet VID dataset in two different settings: accuracy for non-real-time models and best speed-accuracy trade-off on real-time models.

2 Related work

Video Object Detection. VOD task requires detecting objects in each frame and linking the same objects across frames. State-of-the-art methods typically develop sophisticated pipelines to tackle it. In general, VOD task can be divided into two directions: *improving detection accuracy via temporal fusing* and *performing real-time video object detection while keeping the accuracy*.

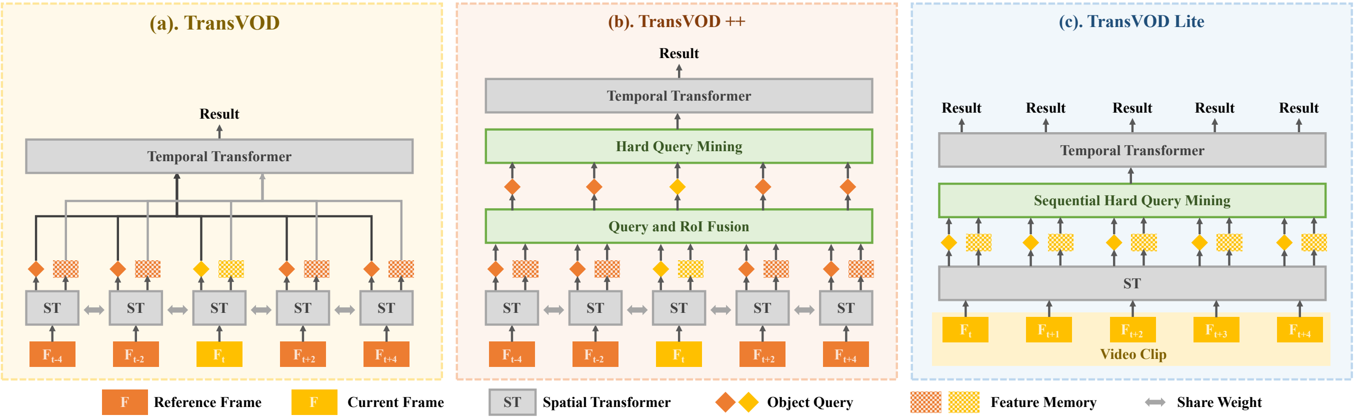


Fig. 2: Illustration of our proposed TransVOD series. (a) Original TransVOD: our network is based on spatial Transformer which outputs spatial object query and feature memory of each frame. We propose a temporal Transformer to link both the spatial object queries and feature memories in a temporal dimension to obtain the results of the current frame. The final detection results are obtained via a shared feed-forward network (FFN). (b) Based on TransVOD, our TransVOD++ add two improvements including Hard Query Mining (HQM) and Query and RoI Fusion module (QRF). (c) Inherited from TransVOD, our TransVOD Lite models the VOD task as a sequence-to-sequence prediction problem, and **directly** outputs all the detection results of the entire sequence in the window via Sequential Hard Query Mining (SeqHQM).

For the first aspect, most previous works [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [27], [28], [29] to amend this problem is feature aggregation that enhances per-frame features by aggregating the features of nearby frames. Earlier works adopt flow-based warping to achieve feature aggregation. Specifically, FGFA [27] and THP [29] both utilize the optic flow from FlowNet [48] to model the motion relation via different temporal feature aggregation strategies. To calibrate the pixel-level features with inaccurate flow estimation, MANet [28] dynamically combines pixel-level and instance-level calibration according to the motion. Nevertheless, these flow-warping-based methods have several disadvantages: 1) Training a model for flow extraction requires large amounts of flow data, which may be difficult and costly to obtain. 2) integrating a flow network and a detection network into a single model may be challenging due to multi-task learning. Another line of attention-based approaches [21], [23], [31], [32], [33], [49] utilize self-attention [50] and non-local [51] to capture long-range dependencies of temporal contexts. SELSA [49] treats video as a bag of unordered frames and proposes to aggregate features in the full-sequence level. STSN [32] and TCENet [21] propose to utilize deformable convolution to aggregate the temporal contexts within a complicated framework with so many heuristic designs. RDN [34] introduces a new design to capture the interactions across the objects in spatial-temporal context. LWDN [33] adopts a memory mechanism to propagate and update the memory feature from key frames to key frames. OGEMN [31] present to use object-guided external memory to store the pixel and instance-level features for further global aggregation. MEGA [23] considers aggregating both the global information and local information from the video and presents a long-range memory. Despite the great success of these approaches, most of the pipelines for video object detection are too sophisticated, requiring many hand-crafted components, *e.g.,* extra optic flow model, memory mechanism, or recurrent neural network. In addition, most of them need complicated post-processing methods such as Seq-NMS [12], Tubelet rescoring [13], Seq-Bbox Matching [14] or REPP [15] by linking the same object across the video to form tubelets and aggregating classification scores in the tubelets to achieve the state-of-the-art. Instead, our previous work TransVOD builds a *simple and end-to-end trainable* VOD framework without these designs. Beyond that, our improved version TransVOD++ incorporates more appearance information into query design and simplifies the pipeline by removing the temporal encoder (TDTE) of origin TransVOD. It achieves better results than TransVOD and state-of-the-art performances on the ImageNet VID dataset.

For the second aspect, starting from DFF [26], several works [16], [17], [33], [36], [37], [38], [52], [53] focus on real-time video object detection while keeping accuracy unchanged or even improved. In general, most of these works also perform specific architecture design with many hand-crafted components and human prior such as object-level tracker in [16], patchwork cell with attention in [54] and Convolutional LSTMs in [37]. Our proposed TransVOD Lite models the entire VOD pipeline as a sequence to sequence problem, as Transformer did in machine translation [42]. It achieves significant improvements over the strong image baseline along with a faster speed.

Vision Transformers. Recently, Vision Transformers [10], [11], [40], [41], [47], [55] make a great progress the computer vision. It can be mainly divided into two directions: replacing CNN backbone with Transformer-Like architecture [40], [47], [56], [57] and using object query to represent instance for scene understanding [10], [11], [58], [59], [60]. Our work is related to the second part. DETR [10] builds a fully end-to-end object detection system based on Transformers, which largely simplifies the traditional detection pipeline. It also achieves on par performances compared with highly-optimized CNN-based detectors [1]. However, it suffers from slow convergence and limited feature spatial resolution, Deformable DETR [11] improves DETR by designing a deformable attention module, which attends to a small set of sampling locations as a pre-filter for prominent key elements out of all the feature map pixels. Our work is inspired by DETR [10] and Deformable DETR [11]. The above works show the effectiveness of Transformers in image object detection tasks. There are several con-current works that applied Transformer into video understanding, *e.g.,* Video Instance Segmentation (VIS) [61], multi-object tracking (MOT). TransTrack [41] introduces a query-key mechanism into the multi-object tracking model, while Trackformer [55] directly adds track query for MOT. However, both only leverage limited temporal information, *i.e.,* just the previous frame. We suppose that this way can not fully use enough temporal contexts from a video clip. VisTR [39] views the VIS task as a direct end-to-end parallel sequence prediction problem. The targets of a clip are disrupted in such an instance sequence, and directly performing target assignment is not optimal. Instead, we aim to link the outputs of the spatial Transformer, *i.e.,* object query, through a temporal Transformer, which acts in a completely different way from VisTR [39]. To our knowledge, there are no prior applications of Transformers to video object detection (VOD) tasks so far. It is intuitive to see that the Transformers' advantage of modeling long-range dependencies in learning temporal contexts across multiple frames for VOD task. Our previous work TransVOD [44], leverages both the spatial Transformer and the temporal Transformer, and then provide an affirmative answer to that. In this paper, based on the TransVOD framework, we provide two extra solutions including TransVOD++ and TransVOD Lite. The former aims to improve the performance of TransVOD while keeping inference efficiency, while the latter carry out real-time VOD detection with much faster inference speed.

Object Tracking. Most single object tracking methods [62], [63], [64] adopt Siamese-like tracking pipelines, where the search patch calculates the correlation with multiple temporal features. Most multiple object tracking [65], [66], [67] adopts tracking by detection pipeline with extra appearance learning. Different from these works, our TransVOD Lite utilizes multiple frames in an offline manner, which can fully explore the entire temporal context of video clips.

3 Method

Overview. We will first review the previous work including both DETR [10] and Deformable DETR [11] in Sec. 3.1. Then, we will give detailed descriptions of our previous proposed TransVOD framework in Sec. 3.2. It contains three key components: Temporal Deformable Transformer Encoder (TDTE), Temporal Query Encoder (TQE), and Temporal Deformable Transformer Decoder (TDTD). Then, we present two advanced versions of our TransVOD framework including TransVOD++ (Sec. 3.3) and TransVOD Lite (Sec. 3.4). Finally, we describe the loss functions and details of inference in Sec. 3.5.

3.1 Revisiting DETR and Deformable DETR

DETR [10] treats object detection as a set prediction problem. A CNN backbone [68] extracts visual feature maps $f \in \mathbb{R}^{C \times H \times W}$ from an image and H, W are the height and width of the visual feature map, respectively. The visual features augmented with position embedding f_{pe} would be fed into the encoder of the Transformer. Self-attention would be applied to f_{pe} to generate the key, query, and value features K, Q, V to exchange information between features at all spatial positions. Let Ω_q and Ω_k indicate the set of query and key elements, respectively. Then, $q \in \Omega_q$ denotes the query element and $k \in \Omega_k$ denotes the key element, respectively, which indexes the query feature $z_q \in R^C$, and key feature $x_k \in R^C$, where C denotes the dimension of the feature. Then, the multi-head attention feature is calculated as follows:

$$\text{MultiHeadAttn}(z_q, x) = \sum_{m=1}^{M} W_m \big[\sum_{k \in \Omega_k} A_{mqk} \cdot W'_m x_k \big], \quad (1)$$

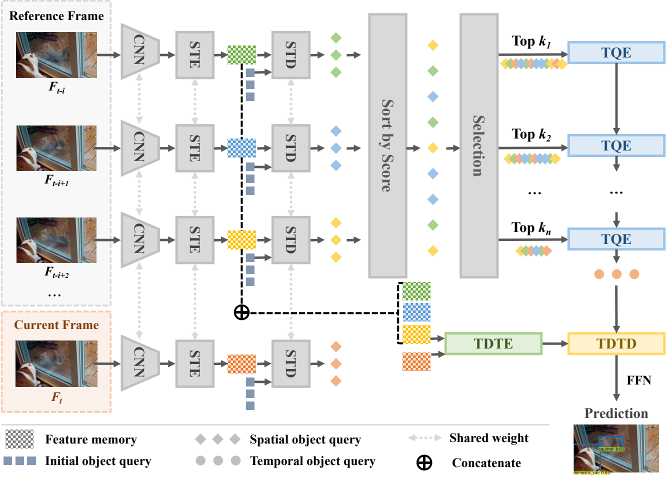


Fig. 3: **The whole pipeline of TransVOD.** A shared CNN backbone extracts features of multiple frames. Next, a series of shared Spatial Transformer Encoders (STE) produce the feature memories and these memories are linked and fed into Temporal Deformable Transformer Encoder (TDTE). Meanwhile, the Spatial Transformer Decoder (STD) decodes the spatial object queries. Naturally, we use a Temporal Query Encoder (TQE) to model the relations of different queries and aggregate these queries, thus we can enhance the object query of the current frame. Both the temporal query and the temporal memories are fed into the Temporal Deformable Transformer Decoder (TDTD) to learn the temporal contexts across different frames.

where m indexes the attention head, $W'_m \in R^{C_v \times C}$ and $W_m \in R^{C \times C_v}$ are learnable weights ($C_v = C/M$ by default). The attention weights A_{mqk} are normalized as:

$$A_{mqk} \propto \exp\{\frac{z_q^T U_m^T V_m x_k}{\sqrt{C_v}}\}, \sum_{k \in \Omega_k} A_{mqk} = 1, \quad (2)$$

in which $U_m, V_m \in R^{C_v \times C}$ are learnable weights. The features z_q and x_k are the concatenation/summation of element contents and positional embeddings in practice. The decoder's output features of each object query are then further transformed by a Feed-Forward Network (FFN) to output class score and box location for each object. Given box and class prediction, the Hungarian algorithm is applied between predictions and ground-truth box annotations to identify the learning targets of each object query for one-to-one matching. Deformable DETR [11] replaces the multi-head self-attention layer with a deformable attention layer to efficiently sample local pixels rather than all pixels. Moreover, to handle missing small objects, they also propose a cross attention module that incorporates multi-scale feature representation. Due to the fast convergence and computation efficiency, we adopt Deformable DETR as our still image Transformer detector.

3.2 TransVOD Framework

The overall TransVOD architecture is shown in Fig. 3. It takes multiple frames in a video clip as inputs and outputs the detection results for the current frame. It contains four main components: Spatial Transformer for single frame object detection to extract both object queries and compact features representation (memory for each frame), Temporal Deformable Transformer Encoder (TDTE) to fuse memory outputs from Spatial Transformer, Temporal Query Encoder (TQE) to link objects in each frame along the temporal dimension and Temporal Deformable Transformer Decoder (TDTD) to obtain final output for the current frame.

Spatial Transformer. We choose the recent Deformable DETR [11] as our still image detector. In particular, to simplify complex designs in [11], we *do not* use multi-scale features in both encoders and decoders. We only use the last stage of the backbone as the input of the deformable Transformer. The modified detector includes Spatial Transformer Encoder (STE) and Spatial Transformer Decoder (STD), which encodes each frame F (including Reference Frame and Current Frame) into two compact representations: spatial object query Q and memory encoding E.

Temporal Deformable Transformer Encoder. The goal of the Temporal Deformable Transformer Encoder is to encode the spatial-temporal feature representations and provide the location cues for the final decoder output. Since most adjacent features contain similar appearance information, using naive Transformer encoder [10], [42] directly may bring much extra computation (most useless computation on object background). Deformable attention [11] samples only partial information efficiently according to the learned offset field. Thus, we can link these memory encodings E_t through the this operation in a temporal dimension. The core idea of the temporal deformable attention modules is that we only attend to a small set of key sampling points around a reference efficiently. The multi-head temporal deformable attention (TempDefAttn) is as follows:

$$\text{TempDefAttn}(z_q, \hat{p}_q, \{x^l\}_{l=1}^{L}) = \sum_{m=1}^{M} W_m \big[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \, x^l(\phi_l(\hat{p_q}) + \Delta p_{mlqk}) \big], \quad (3)$$

where m indexes the attention head, l indexes the frame sampled from the same video clip, and k indexes the sampling points, and Δp_{mlqk} and A_{mlqk} indicate the sampling offset and attention weights of the k^{th} sampling point in the l^{th} frame and the m^{th} attention head, respectively. A_{mlqk} denotes the scalar attention weight in the range of $[0,1]$, normalized by $\sum_{l=1}^{L}\sum_{k=1}^{K} A_{mlqk} = 1$. $\Delta p_{lmqk} \in R^2$ are of 2-d real numbers with unconstrained range. Since $p_q + \Delta p_{mlqk}$ is fractional, we apply bilinear interpolation in [69] for computing $x(p_q + \Delta p_{mlqk})$. For each frame l, both Δp_{mlqk} and A_{mlqk} are calculated by feeding the the query feature z_q to a linear projection of $3MK$ channels, where the first $2MK$ channels encode the sampling offsets Δp_{mlqk}, and the remaining MK channels are fed to a $\mathrm{Softmax}$ function to obtain the attention weights A_{mlqk}. Here, we use normalized coordinates $\hat{p}_q \in [0,1]^2$ for the clarity of scale formulation, in which $(0,0)$ and $(1,1)$ indicate the top-left and the bottom-right image corners, respectively. $\phi_l(\hat{p}_q)$ re-scales the normalized coordinates $\hat{p}_q$ to the input feature map of l-th frame. The multi-frame temporal deformable attention samples LK points from L feature maps instead of K points from single-frame feature maps. There exist total M attention heads in each TDTE layer.

Temporal Query Encoder. As mentioned in the previous part, learnable object queries can be regarded as the non-geometric anchors, which automatically learns the statistical features of the whole still image datasets during the training process. It means that the spatial object queries are not related to temporal contexts across different frames. Thus, we propose a *simple yet effective* encoder to measure the interactions between the objects in the current frame and the objects in reference frames.

Our key idea is to link these spatial object queries in each frame via a temporal Transformer, and thus learn the temporal contexts across different frames. We name our module Temporal Query Encoder (TQE). TQE takes all the spatial queries from reference frames to enhance the spatial output query of the current frame, and it outputs the temporal query for the current frame. Moreover, inspired from [34], we design a coarse-to-fine spatial object query aggregation strategy to progressively schedule the interactions between the current object query and the reference object queries. The benefit of such a coarse-to-fine design is that we can reduce the computation cost to some extent.

Specifically, we combine the spatial object query from all reference frames, denoted as Q_{ref}. Then, we perform the scoring and selection in a coarse-to-fine manner. Specifically, we use an extra Feed Forward Network (FFN) to predict the class logits and after that, we calculate the sigmoid value of that: $p = Sigmoid[FFN(Q_{ref})]$. Then, we sort all the reference points by p value and select the top-confident k values from these reference points. Considering that the shallow layer may learn more detailed information, while there is less information in the deep layers, we perform a coarse-to-fine selection. In other words, the shallow layers should select more confident queries, and the last layers should choose less credible object queries. The selected values are fed to feature refiners to interact with the object queries extracted from different frames, calculating the co-attention with the output of the current frame. The decoder layers with cross-attention modules play the role of a cascade feature refiner which updates output queries of each spatial Transformer iteratively. The refined temporal object query is the input of the Temporal Deformable Transformer Decoder.

Temporal Deformable Transformer Decoder. This decoder aims to obtain the current frame output according to both outputs from TDTE (fused memory encodings) and TQE (temporal object queries). Given the aggregated feature memories $\hat{E}$ and the temporal queries $\hat{O}_q$, our Temporal Deformable Transformer Decoder (TDTD) performs co-attention between online queries and the temporal aggregated features. The deformable co-attention [11] of the temporal decoder layer is shown as follows:

$$\text{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \big[\sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \big], \quad (4)$$

where m indexes the attention head, k indexes the sampled keys, and K is the total number of the sampled keys ($K \ll HW$). p_{mqk} and A_{mqk} indicate the sampling offset and attention weight of the k^{th} sampling point in the m^{th} attention head, respectively. The attention weight $A_{mqk} \in [0,1]$, normalized by $\sum_{k=1}^{K} A_{mqk} = 1$. $\Delta p_{mqk} \in R^2$ are of 2-d real numbers with unconstrained range. Due to the fact that $p_q + \Delta p_{mqk}$ is fractional, we also adopt bilinear interpolation in computing $x(p_q + \Delta p_{mqk})$ following [69]. Both Δp_{mqk} and A_{mqk} are obtained via linear projection over the query feature z_q. In our implementation, the query feature z_q is fed to a linear projection operator. The output of TDTD is sent to one feed-forward network (FFN) for the final classification and box regression as the detection results of the current frame.

3.3 TransVOD++

Compared with previous work, despite TransVOD simplifying the pipeline of VOD, it has several limitations. Firstly, it contains heavy computation costs in TDTE. Secondly, the performance of TransVOD is still limited. To solve these problems, we present TransVOD++ which contains the following improvements including Query and RoI Fusion (QRF), Hard Query Mining (HQM), and a strong backbone. The pipeline is shown in Fig 4.

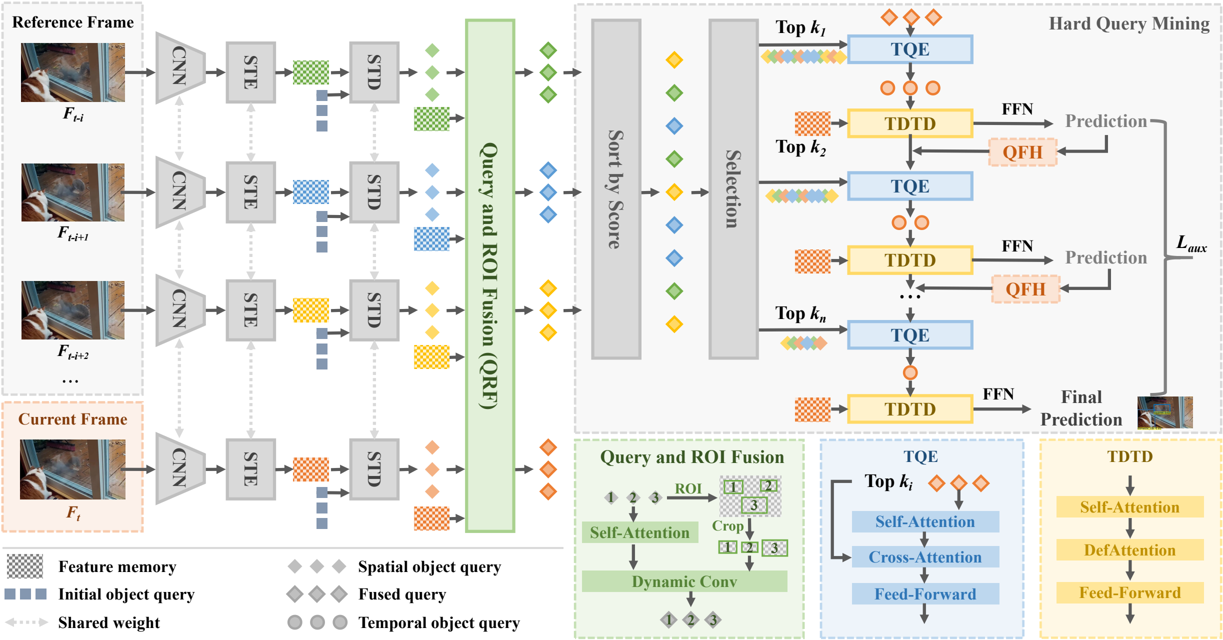


Fig. 4: **The whole pipeline of TransVOD++.** Compared with the original TransVOD, it add the Query and RoI Fusion (QRF) and Hard Query Mining (HQM) module. To avoid redundant spatial information in TDTE, we present QRF by fully injecting the object-level appearance information into each object query. Then, to dynamically reduce the query number and target number, we present HQM for mining the hardest query with multiple TDTD modules and multiple auxiliary TDTD loss functions. Details of Query Filter Head (QFH) is illustrated in Fig. 5.

Query and RoI Fusion. Previous works [23], [70] show that the region features are more useful and contain more precise appearance information for temporal fusion. Our motivation is to replace the TDTE with features in region of interest (RoI) via the proxy strategy where each RoI feature is injected into each query, thus fully utilizing the object-level appearance information to enhance the object query.

In particular, given the detection boxes from spatial Transformers, we get the region of interest (RoI) of each frame in a video clip. Then, according to those RoIs and the feature memory from the spatial Transformer Encoder (STE), we could calculate the RoI feature E_{cur}^{RoI} and E_{ref}^{RoI} of the current frame and the reference frames, respectively. Next, the cropped RoI features are used to weigh each query via the transformation of MLP, as shown in the green part of Fig. 4. The current RoI feature E_{cur}^{RoI} is aggregated onto the current query to generate the enhanced current query $\hat{Q}_{cur}$, where feature aggregation is conducted through dynamic convolutions. Similarly, for each reference frame, the reference RoI features E_{ref}^{RoI} of the i_{th} frame are fused with the reference query of the i_{th} frame.

$$\hat{Q}_{ref}^{j+1} = \begin{cases} \mathrm{QRF}(Q_{ref}^{j}, E_{ref}^{RoI}), \ \text{if } j = 1 \\ \mathrm{QRF}(\hat{Q}_{ref}^{j}, E_{ref}^{RoI}), \ \text{otherwise} \end{cases} \tag{5}$$

where Q_{ref}^{j} denotes the spatial object query of the reference frame before the j_{th} temporal query encoder (TQE), and $\hat{Q}_{ref}^{j}$ denotes the temporal object query before the j_{th} TQE module.

The details of the "Query and RoI fusion" module are described as follows: given the object query and RoI feature memory, we first feed the object query to a multi-head self-attention layer to reason about the relations between objects. Then, each RoI feature will interact with the corresponding object query to filter out ineffective bins and outputs the final object query. Inspired from [71], we carry out two consecutive 1×1 convolutions with ReLU activation function for light design. The k_{th} object query generates dynamic parameters of these two convolutions for the corresponding k_{th} RoI feature via a linear projection.

Finally, the aggregated reference queries $\hat{Q}_{ref}^{j}$ are used to enhance the aggregated current query $\hat{Q}_{cur}^{j}$ via a temporal query encoder (TQE), thus learning the temporal contexts across different frames, as shown in Eq. 6.

$$\hat{Q}_{cur}^{j} = \mathrm{TQE}(\hat{Q}_{cur}^{j}, \hat{Q}_{ref}^{j}) \tag{6}$$

Hard Query Mining. Considering that both the spatial object queries and temporal object queries contain much redundant information across the dataset, for example, 300 queries reflect the temporal appearance distributions of 30 categories, and those queries need to match more than 300 ground truths during the training procedure, and there is no need to maintain so many object queries/targets in both the spatial and temporal dimension. As such, we are motivated to selectively reduce the redundancy of query number and target number in the training of temporal Transformers, and meanwhile, we mine the hardest query in both the current frame and the reference frames.

Concretely, the current query and the reference queries are fed forward to the class embedding layer, *i.e.,* a linear classification layer with sigmoid activation. Then, those reference queries are concatenated in the dimension of the query number. Inherited from TransVOD, we adopt the coarse-to-fine object query aggregation strategy to progressively model the relationships between the current query and the reference queries via TQE module.

The differences between TransVOD++ and TransVOD lie in several aspects. Firstly, in contrast to TransVOD that only selects the reference query, our TransVOD++ selects not only the reference query but also the current query. Both of them are treated differently in a coarse-to-fine manner, thus reducing the computation cost in the temporal Transformer. Secondly, compared to TransVOD, we add a Temporal Defomrable Transformer Decoder (TDTD) after each TQE module and supervise the object

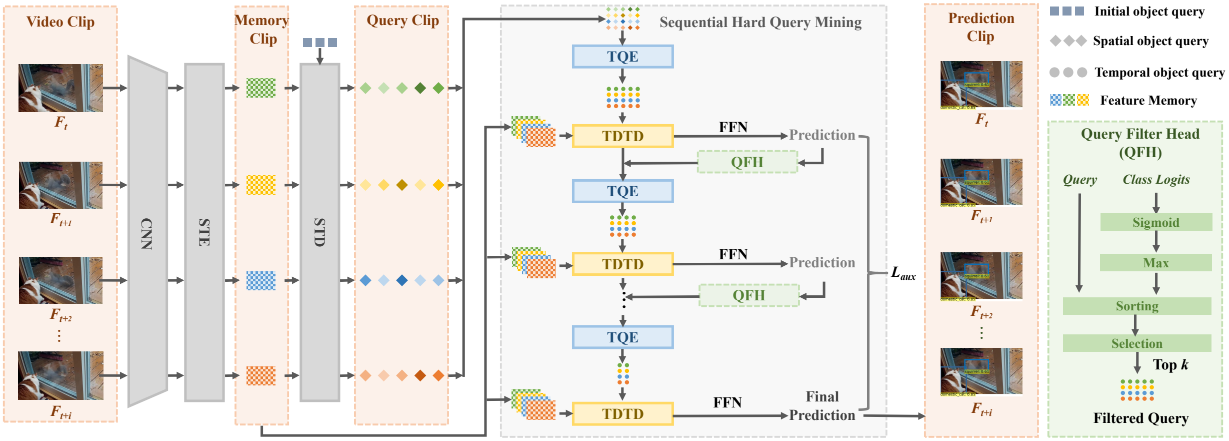


Fig. 5: **The whole pipeline of TransVOD Lite.** Compared with TransVOD and TransVOD++, TransVOD Lite aims at real-time video object detection. It takes multiple frames as inputs and outputs all the results simultaneously. We propose Sequential Hard Query Mining (SeqHQM) to mine the hardest query in a video clip for selectively reducing the redundancy of sequential object queries and targets in the training of temporal Transformers via Query Filter Head (QFH) and auxiliary TDTD loss functions $\mathcal{L}_{aux}$.

query with different query numbers via an **auxiliary TDTD loss**, denoted as $\mathcal{L}_{aux}$. We found it helpful to use auxiliary TDTD losses $\mathcal{L}_{aux}$ in temporal Transformer during training, especially to help the model output the correct number of objects of each class. We add prediction FFNs and Hungarian loss after each TDTD module. All prediction FFNs share their parameters.

Strong Backbone. We further adopt Swin Transformer as the strong backbone network. However, Swin Transformer generates multi-scale features adopted with FPN-like framework [72] which is not suitable for our TransVOD framework. We propose a simple yet effective solution via fusing multi-scale features into one scale where we directly add multi-scale features into one scale.

3.4 TransVOD Lite

Despite TransVOD and TransVOD++ make the VOD pipeline much simpler, the inference time is still limited due to multiple frame query fusing. As mentioned in Section 2, the inference time is also critical for real application. To embrace the advantage of modeling sequence data in transformer [39], [50], we present TransVOD Lite where it takes multi frames as inputs and output detection results of all frames directly, as shown in Fig. 5.

Direct Multiple Frame Predictions. In TransVOD Lite, we abandon the feature aggregation paradigm, which requires much more computation costs in terms of time and memory space. Instead, a sequence of video clips is fed as input and output a sequence of results. As shown in Figure 5, TransVOD Lite inherits the Hard Query Mining from the TransVOD++ and spatial-temporal transformer design in TransVOD including TQE, TDTE. The main difference is that TransVOD Lite directly outputs the multiple frame prediction with a hyper-parameter T_w which is the temporal window size of the input clip or the number of the input frames. When T_w is larger, the inference speed is faster while the memory is increased. In this way, we can fully use the memory of GPU to speed up the inference time. We provide detailed experiments on the effect of choosing T_w in the experiment part.

Sequential Hard Query Mining. Different from TransVOD and TransVOD++, we do not need to discriminate whether an object query is the reference query or the current query for filtering, all object queries in the whole sequence are equally selected in a coarse-to-fine manner, thus increasing the speed, *e.g.,* FPS, to T_w times in temporal Transformer than original TransVOD, where T_w denotes the temporal window size in a given clip. We name our method "sequential hard query mining" (SeqHQM). For example, $T_w = 12$ means the input frames are 12 in the video clip, and then we need to generate the results of those 12 frames, if each frame has 300 object queries, there are 3600 object queries in total. There is no doubt that there exists large redundant information of those large number of object queries, and it is necessary to dynamically reduce the computation costs to boost the inference speed, as well as achieve good results in modeling the temporal motion.

In particular, given a sequence of spatial object query Q_{seq}, we fed it into a Query Filter Head (QFH), which select the most confident object query and filter those redundant object query in a video sequence. The number of object queries and targets is dynamically decreasing to reduce the computation redundancy. We implement the QFH differently before the k_{th} TQE module. If $k = 1$, we use the class embedding layer of the spatial Transformer to generate class logits and go through a sigmoid activation function. If $k > 1$, the class logits are generated through the learnable temporal class embedding layer then with a sigmoid activation function. Next, we compute the maximum probability and select the top k confident query by sorting and selection in a coarse-to-fine manner, which is illustrated in the green part of Fig. 5. Similar to TransVOD++, we add a TDTD after each TQE module and supervise the object query with different query numbers via an **auxiliary TDTD loss**, denoted as $\mathcal{L}_{aux}$. $\mathcal{L}_{aux}$ is essential to help the model output the correct number of objects of each class. We add prediction FFNs and Hungarian loss after each TDTD module. All prediction FFNs share their parameters.

3.5 Loss Functions and Inference

Loss functions. Original DETR [10] avoids post-processing and adopts a one-to-one label assignment rule. Following [10], [11], [73], we match predictions from STD/TDTD with ground truth by Hungarian algorithm [74] and thus the entire training process of spatial Transformer is the same as original DETR. The temporal Transformer uses similar loss functions given the box and class prediction output by two FFNs. The matching cost is defined as the loss function. Following [10], [11], [71], the loss function is:

$$\mathcal{L}_{aux} = \sum_{j=1}^{J} \left[\lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{L1} \cdot \mathcal{L}_{L1} + \lambda_{giou} \cdot \mathcal{L}_{giou} \right], \quad (7)$$

TABLE 1: Comparison with the state-of-the-art methods on ImageNet VID using ResNet 50 as the backbone.

Methods	Base Detector	mAP (%)
Single Frame Baseline [1]	Faster-RCNN	71.8
DFF [26]	Faster-RCNN	70.4
FGFA [27]	Faster-RCNN	74.0
RDN [34]	Faster-RCNN	76.7
MEGA [23]	Faster-RCNN	77.3
Single Frame Baseline [11]	Deformable DETR	76.0
TransVOD	Deformable DETR	**79.9**
TransVOD++	Deformable DETR	**80.5**

where J denotes the total number of TDTD modules in the temporal Transformers, where J = 1 for TransVOD and J = 3 for TransVOD++ and TransVOD Lite in all experiments. $\mathcal{L}_{cls}$ represents focal loss [5] for classification. $\mathcal{L}_{L1}$ and $\mathcal{L}_{giou}$ represent L1 loss and generalized IoU loss [75] in for localization. λ_{cls}, λ_{L1} and λ_{giou} are coefficients of them. We balance these loss functions following the same setting in [11]. For TransVOD Lite, we apply such a loss function for all input frames.

Inference for TransVOD Lite. In TransVOD Lite, the window size of a given video is defined as T_w and the interval between the two adjacent frames within one clip is denoted as I_w, respectively. Given a video $V = \{F_1, F_2, \cdots, F_N\}$, we first expand the video size to the integer multiples of T_w as: $\hat{N} = \lceil \frac{N}{T_w} \rceil T_w$. Then, for each expanded video, we divide the video into two parts and adopt different sampling strategies for these two parts.

As for the first part, the clip is normal where the interval of different frames is I_w. The index of the first frame in each video clip is $S = T_w I_w i + j$, where $i \in \{0, 1, \cdots, K-1\}$, $j \in \{1, \cdots, I_w - 1\}$, $K = \lfloor \frac{\hat{N}}{T_w I_w} \rfloor$. We input the normal clip sequentially with window size T_w and interval size I_w to feed the clip into the model. For the second part, the frames are not divisible by $T_w I_w$. The index of the first frame is the clip is $T_w k + 1$. There are $\hat{N} - TWk$ frames in this clip. Those frames are randomly divided into $\frac{\hat{N}}{T_w} - KI_w$ video clips, with the size of each clip as T_w.

For example, if $N = 10$, $T_w = 4$, $I_w = 2$, the video $V = \{F_1, F_2, \cdots, F_{10}\}$, we first expand the video size from $N = 10$ to $\hat{N} = 12$, $\hat{v} = \{F_1, F_2, \cdots, F_{10}, F_{11}, F_{12}\}$, where $F_{12} = F_{11} = F_{10}$. Then, we will split the video into two parts: the first part includes two normal clips, $\hat{C}_1 = \{F_1, F_3, F_5, F_7\}$, and $\hat{C}_2 = \{F_2, F_4, F_6, F_8\}$. The left part includes: $\hat{C}_3 = \{F_9, F_{10}, F_{11}, F_{12}\}$. $\hat{C}_1$ and $\hat{C}_2$ are directly input to the model and $\hat{C}_3$ are randomly shuffled and sent to the model.

In contrast, we introduce another sampling strategy using random shuffling. We find that if we first randomly shuffle the $\hat{v}$ and split it to $\frac{\hat{N}}{T_w}$ clips, our model could model the temporal motions better due to the large view of the video. The empirical evidence perceived by the human visual system illustrates that when people are not certain about the identity of an object, they would seek to find a distinct object from other frames that share high semantic similarity with the current object and assign them together. Regarding that Transformers are effective in modeling the long-range dependencies, if we randomly shuffle the video, we could increase the data diversity and fully utilize the global information of the video. The effectiveness of both strategies is demonstrated in the experimental part.

TABLE 2: Comparison with the state-of-the-art on ImageNet VID. Most methods use ResNet 101 as the backbone. $\star$ denotes using Swin-Base as backbone.

Methods	Base Detector	mAP(%)
Single Frame Baseline [76]	R-FCN	73.6
DFF [26]	R-FCN	73.0
AdaScale [77]	R-FCN	75.5
D&T [52]	R-FCN	75.8
FGFA [27]	R-FCN	76.3
LWDN [33]	R-FCN	76.3
IFF-Net [30]	R-FCN	77.1
SCNet [78]	R-FCN	77.9
AFA [79]	R-FCN	77.9
THP [29]	R-FCN	78.6
STSN [32]	R-FCN	78.9
PSLA [25]	R-FCN	80.0
OGEMN [31]	R-FCN	80.0
STMN [80]	R-FCN	80.5
TCENet [21]	R-FCN	80.3
MAMBA [24]	R-FCN	80.8
Single Frame Baseline [1]	Faster RCNN	76.7
ST-Lattice [22]	Faster RCNN	79.0
BFAN [81]	Faster RCNN	79.1
STCA [82]	Faster RCNN	80.3
SELSA [49]	Faster RCNN	80.3
MINet [83]	Faster RCNN	80.6
LRTR [35]	Faster RCNN	81.0
RDN [34]	Faster RCNN	81.8
TROI [84]	Faster RCNN	82.0
MEGA [23]	Faster RCNN	82.9
HVRNet [18]	Faster RCNN	83.2
TF-Blender [85]	Faster RCNN	83.8
DSFNet [20]	Faster RCNN	84.1
MAMBA [24]	Faster RCNN	84.6
EBFA [19]	Faster RCNN	84.8
CFA-Net [86]	Faster RCNN	85.0
Single Frame Baseline [87]	CenterNet	73.6
CHP [88]	CenterNet	76.7
Single Frame Baseline [11]	Deformable DETR	78.3
TransVOD Lite	Deformable DETR	80.5
TransVOD++	Deformable DETR	82.0
TransVOD++*	Deformable DETR	**90.0**

4 Experiment

Overview. In this section, we first introduce the experimental setup for the VOD task, including the dataset and evaluation protocols. Then, we present the implementation details of TransVOD. We also compare our proposed method with several other state-of-the-art VOD methods in various settings. Then, we perform several ablation studies and analyses for all three models on the ImageNet VID [43] validation set. Finally, we provide both visualization results and analysis to our models.

4.1 Experimental Setup

Datasets. We empirically conduct experiments on the ImageNet VID dataset [43] which is a large-scale benchmark for video object detection. It contains 3862 training videos and 555 validation videos with annotated bounding boxes of 30 classes. Since the ground truth of the official testing set is not publicly available, we follow the widely adopted setting in previous works [27], [28], [34], [49] where we train our models using a combination of ImageNet VID and DET datasets [43] and measure the performance on the validation set using mean average precision (mAP) metric.

Implementation detail. We use ResNet-50 [68] and ResNet-101 [68] as the network backbone. Following original Deformable DETR [11], the optimizer is AdamW [89] with batch size 2, initial

Transformer's learning rate 2×10^{-4}, the backbone's 2×10^{-5}, and weight decay 10^{-4}. All Transformer weights are initialized with Xavier init [90], and the backbone ImageNet-pretrained [91] model with frozen batch-norm layers [92]. The number of initial object query is set as 300. Following [16], [36], [79], we pre-train our image detector on the COCO dataset [93]. Following previous work [23], we use the same data augmentation including random horizontal flip, random resizing the input images such that the shortest side is at least 600 while the longest at most 1000. We train the network for 7 epochs and the learning rate drops at the 5-th and 6-th epochs when using the ResNet-50 and ResNet-101 as the backbone network. We also use Swin Transformer [47] as the backbone. The inference time is calculated on a single V100 GPU card. In the inference phase, we do not need any sophisticated post-processing method, which largely simplifies the pipeline of VOD. Both the spatial Transformer encoder and the spatial Transformer decoder's weight are fixed for better convergence when training the temporal Transformer.

4.2 Main Results

We first compare our proposed TransVOD and TransVOD++ using ResNet-50 backbone in Table 1. Then we present the detailed results with the previous state-of-the-art methods in Table 2. Finally, we compare the real-time models in Table 3.

Results using ResNet-50 backbone. As shown in Table 1, the results under the same backbone ResNet-50 demonstrate that our proposed TransVOD achieves the best performance against the state-of-the-art methods by a large margin. In particular, the mAP can achieve 79.9 % with ResNet-50, which makes 2.6 % absolute improvement over the best competitor MEGA [23]. Our proposed TransVOD++ further improves the original TransVOD by 0.6 %, achieving 80.5 % on the ImageNet VID validation set.

Results with stronger backbone. We further report stronger backbone results to compare with the state-of-the-art methods in Table 2. When equipped with a stronger backbone ResNet-101, the mAP of our TransVOD++ is further boosted up to 82.0%, which outperforms most state-of-the-art methods [26], [27], [28], [29], [77]. Specifically, our model is remarkably better than FGFA [27] (76.3% mAP) and MANet [28] (78.1% mAP), which both aggregate features based on optical flow estimation, and the mAP improvements are +5.6% mAP and +3.8% mAP respectively. When compared with some relation-based methods (LRTRN [35] (81.0% mAP), RDN [34] (81.8% mAP), SELSA [49] (80.3 % mAP)), our method also shows its superiority in case of detection precision. Moreover, our proposed method boosts the strong baseline *i.e.,* deformable DETR [11] by a significant margin (**3%** $\sim$ **4% mAP**). After adopting Swin Base (SwinB) as the backbone, our TransVOD++ achieve **90.0 % mAP** and it outperforms previous works by a large margin (about 4 % $\sim$ 5 % mAP). We will detail the setting in the following parts.

Results using TransVOD Lite In Table 3, we report our TransVOD Lite model with previous real-time models. As shown in that table, using the ResNet-101 backbone, our method achieves the best speed and accuracy trade-off. After adopting Swin-Tiny as the backbone, our TranVOD Lite achieves **83.7 % mAP** while running at nearly 30 FPS. Our best TransVOD Lite model with a Swin base backbone can achieve 90.1 % mAP while running at 15.0 FPS. Furthermore, the parameter count (46.9M) is fewer than other video object detectors (*e.g.,* around 100M in [26]), which also indicates that our method is more friendly for mobile devices.

TABLE 3: **Performance comparison with the state-of-the-art real-time VOD methods on ImageNet VID validation set**. In terms of both accuracy and speed, Our method outperforms most of them and has fewer parameters than existing models.

Model	mAP (%)	Runtime (FPS)	#Params (M)	Backbone
DFF [26]	73.1	20.25	97.8	Res101
D &T [77]	75.8	7.8	-	Res101
LWDN [33]	76.3	20	77.5	Res101
OGEMNet [31]	76.8	14.9	-	Res101
THP [29]	78.6	13.0	-	Res101+DCN
RDN [34]	81.8	10.6	-	Res101
SELSA [49]	80.3	7.2	-	Res101
LRTR [35]	80.6	10	-	Res101
PSLA [25]	77.1	18.7	63.7	Res101
PSLA [25]	80.0	13.3	72.2	Res101+DCN
LSTS [17]	77.2	23.0	64.5	Res101
LSTS [17]	80.1	21.2	65.5	Res101+DCN
TransVOD Lite	80.5	**32.3**	74.2	Res101
TransVOD Lite	83.7	29.6	**46.9**	SwinT
TransVOD Lite	85.8	22.2	68.3	SwinS
TransVOD Lite	**90.1**	14.9	106.3	SwinB

4.3 Ablation Study and Analysis

Overview. In this section, we demonstrate the effect of key components in our proposed methods including TransVOD, TransVOD++ and TransVOD Lite. For TransVOD, we adopt ResNet-50 as the backbone. For TransVOD++ and TransVOD Lite, we adopt Swin Transformer as the backbone.

4.3.1 *Ablation for TransVOD*

Effectiveness of each component in TransVOD. Table 4(a) summarizes the effects of different design components on the ImageNet VID dataset. Temporal Query Encoder (TQE), Temporal Deformable Transformer Encoder (TDTE), and Temporal Deformable Transformer Decoder (TDTD) are three key components of our TransVOD. The single-frame baseline Deformable DETR [11] is 76.0%. By only adding TQE, we boost the mAP with an additional +2.9 %, which demonstrates that TQE can effectively measure the interaction among the objects in different video frames. And then, by adding the TDTD and TDTE sequentially, we boost the mAP with an additional +0.4% and +0.6%, achieving 79.3% and 79.9%, respectively. These improvements show the effects of individual components of our TransVOD.

Number of encoder layers in TDTE. Table 5(a) illustrates the ablation study on the number of encoder layers in TDTE. We observe that when the number of TDTE encoder layers are larger than 1, it brings no significant benefits to the final performance. This experiment also proves the claim that aggregating the feature memories in a temporal dimension via deformable attention is useful for learning the temporal contexts across different frames.

Number of encoder layers in TQE. Table 5(b) shows the ablation study on the number of encoder layers in TQE. It shows that the best result occurs when the number query layer is set to 5. When the number of layers is up to 3, the performance is basically unchanged. Thus, we use 3 encoder layers in our final method.

Number of decoder layers in TDTD. Table 5(c) illustrates the ablation study on the number of decoder layers in TDTD. The basic setting is 4 reference frames, 1 encoder layer in TQE, and 1 encoder layer in TDTE. The results indicate that only one decoder layer in TDTD is needed.

Number of top k object queries in TQE. To verify the effectiveness of our coarse-to-fine Temporal Query Aggregation strategy,

TABLE 4: Ablation studies of TransVOD on ImageNet VID using ResNet 50 as the backbone.

(a) Effect of each component. TDTE: Temporal Deformable Transformer Encoder. TQE: Temporal Query Encoder. TDTD: Temporal Deformable Transformer Decoder.

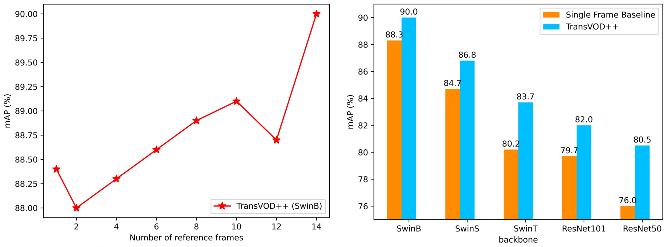
Single Frame Baseline	TDTE	TQE	TDTD	mAP (%)
✓				76.0
✓	✓		✓	77.1
✓		✓		78.9
✓		✓	✓	79.3
✓	✓	✓	✓	79.9

(b) Ablation of top k spatial object query numbers with three encoder layers. Our coarse-to-fine strategy has better results.. Best view it in color.

k1	30	30	30	50	50	80	80
k2	20	20	30	30	50	50	80
k3	10	20	30	20	50	20	80
mAP(%)	79.7	79.6	79.3	79.6	79.5	79.9	79.7

TABLE 5: Ablation studies on TransVOD: number of encoder layers N_{TDTE} in TDTE, number of encoder layers N_{TQE} in TQE, number of decoder layers N_{TDTD} in TDTD and top k spatial query in TQE with one decoder layer.

(a) Number of encoder layers N_{TDTE} in TDTE.						
N_{TDTE}	0	1	2	3	4	
mAP(%)	77.0	77.7	77.6	77.8	77.7	
(b) number of encoder layers N_{TQE} in TQE						
N_{TQE}	1	2	3	4	5	6
mAP(%)	78.8	79.4	79.6	79.6	79.7	79.7
(c) number of decoder layers N_{TDTD} in TDTD.						
N_{TDTD}	1	2	3	4	5	6
mAP (%)	78.2	77.7	77.1	76.2	74.8	72.3
(d) top k spatial query in TQE with one decoder layer.						
k	25	50	100	200	300	
mAP(%)	78.0	78.1	78.3	77.9	77.7	
(d) Number of reference frames N_{ref}.						
N_{ref}	2	4	8	10	12	14
mAP(%)	77.7	78.3	79.0	79.1	79.0	79.3



(a). Effect on the number of reference frames

(b). Improvements over different baselines

Fig. 6: Ablations of TransVOD++: (a). Effect on the number of reference frames N_{ref} using Swin Base as the backbone. (b) Improvements over the different single frame baseline.

we conduct ablation experiments in Table 5(d) and Table 4(b) to study how they contribute to the final performance. All the experiments in each table are conducted under the same setting. The first experiment is that when we use 1 encoder layer in TQE with 4 reference frames, the best performance is achieved when we choose the top 100 spatial object queries for each reference frame. The second experiment is conducted in a multiple TQE encoder layers case, *i.e.,* 3 encoder layers in TQE. We denote the Fine-to-fine (F2F) selection by using a small number of spatial object queries in each TQE encoder layer. coarse-to-coarse (C2C) means selecting a large number of spatial object queries when performing the aggregation in each layer. Our proposed coarse-to-fine aggregation strategy is using a larger number of spatial object queries in the shallow layers and a smaller number of spatial object queries in the deep layers to conduct the query aggregation. The results in Table 4(b) show that our coarse-to-fine aggregation strategy is superior to both the coarse-to-coarse selection and fine-to-fine selection.

Number of reference frames in TransVOD. Table 5(d) illustrates the ablations on number of reference. The basic setting is 3 encoder layers in TQE, 1 encoder layer in TDTE, and 1 decoder layer in TDTD. As shown in Table 5(d), the mAP improves when the number of reference frames increases, and it tends to stabilize when the number is up to 8. We set the reference frames to 8 for both TransVOD and TransVOD++.

4.3.2 Ablation for TransVOD++

Effect of each component in TransVOD++ on strong baseline. In Table 6(a), we verify the effectiveness of each component in TransVOD++ on a strong baseline. Adding RoI and Query Fusion results in 1.4 % mAP improvements while applying Hard Query Mining leads to extra 0.3 % mAP improvements and 1.6 % mAP improvements on small objects. This proves that our proposed Hard Query Mining is suitable for detecting small objects.

Effect of reference frames in TransVOD++. In Fig. 6 (a), we show the effect of reference frames in TransVOD++ where we find the best reference frames is 14. This is different from the original TransVOD. We argue that utilizing more RoI information rather than full-frame fusion in the temporal dimension leads to better results. This finding is consistent with previous works [23], [34], [86] focusing on RoI-wised fusion in Faster-RCNN framework. We set the number of reference frames to 14 by default.

Improvements over different baselines. In Fig. 6 (b), we show the imporvements over different single frame baselines including Swin Transformer [47] and ResNet [68]. Swin Base, Swin Small, and Swin Tiny are abbreviated as SwinB, SWinS, SwinT, respectively. Our proposed TransVOD++ can boost the gain over 2.0%-4.0% mAP on various baselines.

Effect of multi-level feature fusion. In Table 6(b), we show the improvements on multi-level feature fusion. In total, there is a 0.6 % mAP$_{50}$ gain. However, there is a more significant gain (2.3) on mAP$_{50:95}$ which indicates multi-scale information leads to more accurate detection results. Thus, we adopt the simple multi-level feature fusion as the default settings when adopting Swin Transformer as the backbone for both TransVOD++ and TransVOD Lite.

Effect of COCO pre-training using Swin base. Following [16], [36], [79], we pre-train our image detector on the COCO dataset [93]. As shown in Table 6(c), removing COCO pre-training leads to a huge drop. This indicates our TransVOD needs more training examples to achieve better performance and this is also consistent with original vision transformer [40]. Thus, we pre-train both TransVOD++ and TransVOD Lite on the COCO dataset

TABLE 6: Ablation studies of TransVOD++ on ImageNet VID using Swin Transformer Base (SwinB) as the backbone.

(a) Effect of each component of TransVOD++

Component	(a)	(b)	(c)
Single Frame Baseline RoI and Query Fusion Hard Query Mining	✓	✓ ✓	✓ ✓ ✓
mAP$_{50}$ (%)	88.3	89.7	**90.0**
mAP$_{50:95}$ (%)	67.3	67.2	**67.8**
mAP$_{50:95}$ (%) (small)	14.2	16.0	**17.6**
mAP$_{50:95}$ (%) (medium)	39.0	41.5	**42.1**
mAP$_{50:95}$ (%) (large)	73.9	73.7	**74.4**

(b) Effect of multi-level feature fusion.

Component	(a)	(b)
Single Frame Baseline Multi-level feature fusion	✓	✓ ✓
mAP$_{50}$ (%)	87.7	88.3
mAP$_{50:95}$ (%)	65.0	67.3
mAP$_{50:95}$ (%) (small)	12.2	14.2
mAP$_{50:95}$ (%) (medium)	35.3	39.0
mAP$_{50:95}$ (%) (large)	72.2	73.9

(c) Effect of COCO pre-training.

Component	(a)	(b)
Single Frame Baseline COCO pre-training	✓	✓ ✓
mAP$_{50}$ (%)	44.8	88.3
mAP$_{50:95}$ (%)	28.5	67.3
mAP$_{50:95}$ (%) (small)	4.5	14.2
mAP$_{50:95}$ (%) (medium)	12.8	39.0
mAP$_{50:95}$ (%) (large)	33.3	73.9

TABLE 7: Ablation studies of TransVOD Lite on ImageNet VID.

(a) Effect of window size T_w with Swin Tiny as backbone.

T_w	AP$_{50}$	AP$_{50:95}$	AP$_S$	AP$_M$	AP$_L$	FPS
1	76.6	55.1	12.6	31.5	63.7	16.5
2	79.1	56.7	12.4	34.1	65.0	21.7
4	80.9	57.9	12.1	35.1	66.0	23.5
6	81.5	58.3	14.3	35.8	66.4	22.9
8	82.1	58.6	13.7	36.3	66.6	22.5
10	82.3	58.7	13.7	35.9	66.7	29.2
12	82.7	59.0	13.7	**36.6**	**67.0**	30.1
14	82.5	58.8	14.4	**36.6**	66.8	**32.2**
15	**83.7**	**66.2**	**14.7**	35.1	67.3	29.6

(b) Effect of window size T_w with Swin Base as backbone.

T_w	AP$_{50}$	AP$_{50:95}$	AP$_S$	AP$_M$	AP$_L$	FPS
1	85.4	64.1	13.8	39.1	72.0	10.6
2	87.6	66.2	14.2	41.2	74.0	12.7
4	88.6	66.5	**14.9**	42.4	74.1	14.2
6	89.2	67.0	14.2	42.3	74.5	15.2
8	88.9	66.7	14.3	42.6	74.2	15.4
10	88.8	66.4	14.4	42.6	74.0	15.6
12	**90.1**	**67.7**	13.7	**43.1**	**75.3**	**16.2**
14	88.9	66.7	14.4	42.3	74.2	15.4
15	90.0	67.3	**14.9**	41.6	74.9	15.0

(c) Effect of interval size I_w using Swin Base as backbone.

(a) Interval size I_w when window size $T_w = 4$.					
$I_w(T_w = 4)$	1	4	8	12	Randomly Shuffle
mAP(%)	86.3	86.9	87.2	87.5	88.6
(a) Interval size I_w when window size $T_w = 8$.					
$I_w(T_w = 8)$	1	4	8	12	Randomly Shuffle
mAP(%)	86.6	87.3	88.0	88.2	88.9
(b) Interval size I_w when window size $T_w = 12$.					
$I_w(T_w = 12)$	1	4	8	12	Randomly Shuffle
mAP(%)	86.9	88.0	88.7	89.3	90.1

(d) Ablation of top k query numbers in SeqHQM using ResNet-101 as backbone where the window size is set to 14.

k1 k2 k3	30 20 10	30 20 20	50 30 25	80 50 30	100 80 50	100 80 30
mAP$_{50}$(%)	78.4	78.4	78.8	80.4	80.3	79.9
mAP$_{50:95}$(%)	56.2	56.4	56.5	58.3	58.2	58.0
mAP$_S$(%)	11.1	11.3	11.4	10.1	10.0	10.0
mAP$_M$(%)	30.8	31.2	31.4	29.1	29.3	28.6
mAP$_L$(%)	65.0	65.1	65.2	65.4	65.2	65.1

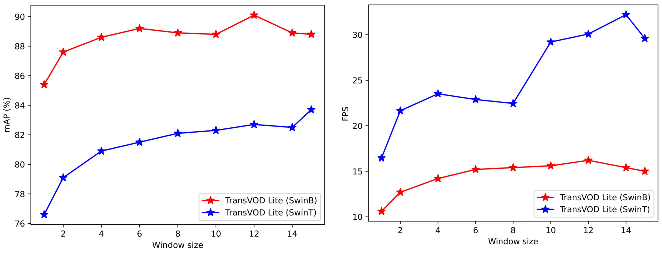
by default. This observation shows that our models have the potentiality to scale up on large video object detection datasets.

4.3.3 Ablation for TransVOD Lite

Effect of window size in TransVOD Lite. In Fig. 7 (a) and Fig. 7 (b), we show the effect of window size on both accuracy and inference time where the interval mode is randomly shuffled within the window for all experiments. As shown in these figures, increasing window size leads to both accuracy improvements and FPS increase for both Swin Tiny and Swin base as backbones. In Table 7(a) and Table 7(b), we detail the results of the above figures. We choose the best window size T_w as 15 for all models.

Effect of interval size and mode in TransVOD Lite. In Table 7(c), we show the effect of interval size between frames in each fixed window. For different window sizes, increasing the interval size leads to better results. This indicates that fusing more global temporal information leads to better results. However, adopting our proposed randomly shuffled strategy results in the best performance on different window sizes. This is mainly because random shuffles increase the diversity of each frame. For example, the global and local temporal information can exist in one window. Moreover, during training, the frames are randomly selected from each clip. Thus randomly shuffled inputs share the same distribution with training examples. Thus we report the final performance using such settings. Moreover, as shown in Table 7(c), even with the sequential inputs, our methods can still achieve the best performance compared with methods in Table 3.

Ablation on query numbers in Sequential Hard Query Mining. In Table 7(d), we perform ablation studies on Sequential Hard Query Mining (SeqHQM) in TransVOD Lite. From the table, we find the best hyper-parameter with 80, 50, 30 queries for each stage. We use that setting for all the TransVOD Lite models.



(a). Effect of window size on mAP (%)

(b). Effect of window size on FPS

Fig. 7: Effect of the temporal window size T_w of a video clip on the mean Average Precision (mAP) (a) and on the Frame Per Second (FPS) (b) in TransVOD Lite using Swin Base and Swin Tiny as the backbone, respectively.

4.4 Visualization and Analysis

Visual detection results. As shown in Fig. 10, we show the visual detection results of still image detector Deformable DETR [11]

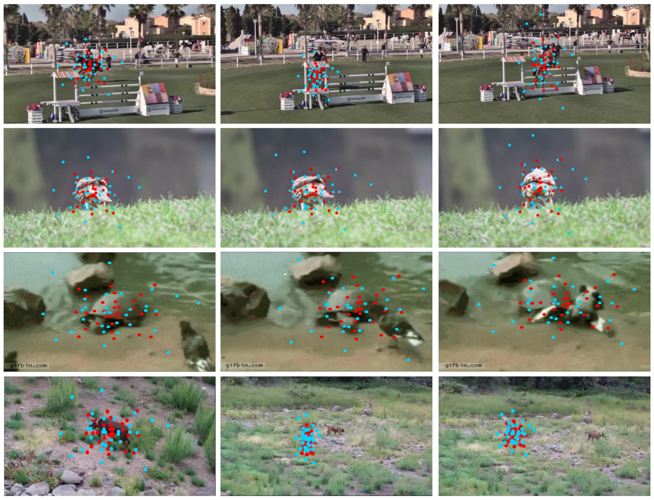


Fig. 8: The visualization of the deformable cross-attention in the last spatial Transformer decoder layer and temporal Transformer decoder layer. We visualize the sampling locations of the temporal object query and corresponding spatial object query in one picture. Each sampling point of the temporal object query is marked as a red-filled circle, while the blue circle represents the sampling point of the spatial query.

and our proposed TransVOD in odd and even rows, respectively. The still image detector is easy to cause false detection (*e.g.,* turtle detected as a lizard) and missed detection (*e.g.,* zebra not detected), in the case of motion blur, part occlusion. Compared with still image detectors, our method can effectively model the long-range dependencies across different video frames to enhance the features of the detected image. Thus, our proposed TransVOD can not only increase the confidence of correct prediction but also effectively reduce the number of cases that are missed or falsely detected. Moreover, as shown in Fig. 10 (b), our proposed TransVOD Lite shows the more confident scores than the still image detector.

Visual sampling locations of object query in TransVOD. To further explore the advantages of TQE, we visualize the sampling locations of both spatial object query and temporal object query in Fig. 8. The sample locations indicate the most relevant context for each detection. As shown in the figure, for each frame in each clip, our temporal object query *has more concentrated and precise results* on fore-ground objects while the original spatial object query has more diffuse results. This proves that our temporal object query is more suitable for detecting objects in video. This explains the effectiveness of our temporal query fusion.

Failure case analysis. In Fig. 9, we present several failure cases using our best TransVOD Lite model. The first two rows show the missing detection problems. The first is mainly due to the larger motion blur and the second is caused by the various background change. The last two rows show the false detection where a car is detected as a bus. This is caused by the large occlusion. Both cases show that tackling occlusion and more stable temporal modeling are needed for further work.

5 Conclusion

In this paper, we proposed a novel video object detection framework, namely TransVOD, which provides a new perspective of feature aggregation by leveraging spatial-temporal Transformers. TransVOD effectively removes the need for many hand-crafted components and complicated post-processing methods. Our core idea is to aggregate both the spatial object queries and the memory encodings in each frame via temporal Transformers. Our TransVOD boosts the strong baseline deformable DETR by a significant margin (3%-4% mAP) on the ImageNet VID dataset. To our knowledge, our work is the **first one** that applies the Transformer to video object detection tasks. Based on the TransVOD framework, we present two advanced versions, namely TransVOD++ and TransVOD Lite. The former improves the performance of TransVOD via better Query and RoI fusion (QRF), and hard query mining (HQM) to fully utilize the object-level information, and dynamically reduce the number of object queries and targets. The latter focuses on real-time video object detection by modeling VOD as a sequence-to-sequence prediction problem via Sequential Hard Query Mining (SeqHQM). Both models set new state-of-the-art results on the ImageNet VID dataset on two different settings: accuracy for non-real-time models and best speed-accuracy trade-off on real-time models. Our method is the first work that achieves 90% mAP on ImageNet VID dataset.

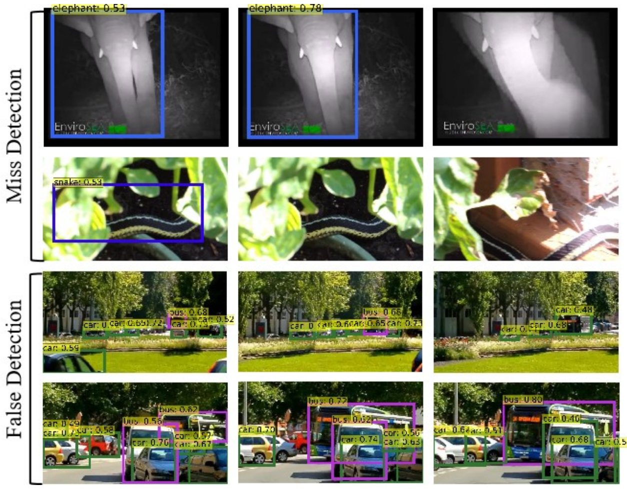


Fig. 9: Failure case analysis. First and second row: miss detection. Third and fourth row: false detection. The results are obtained via our TransVOD Lite with Swin Base backbone.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[2] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[4] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[6] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9627–9636.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

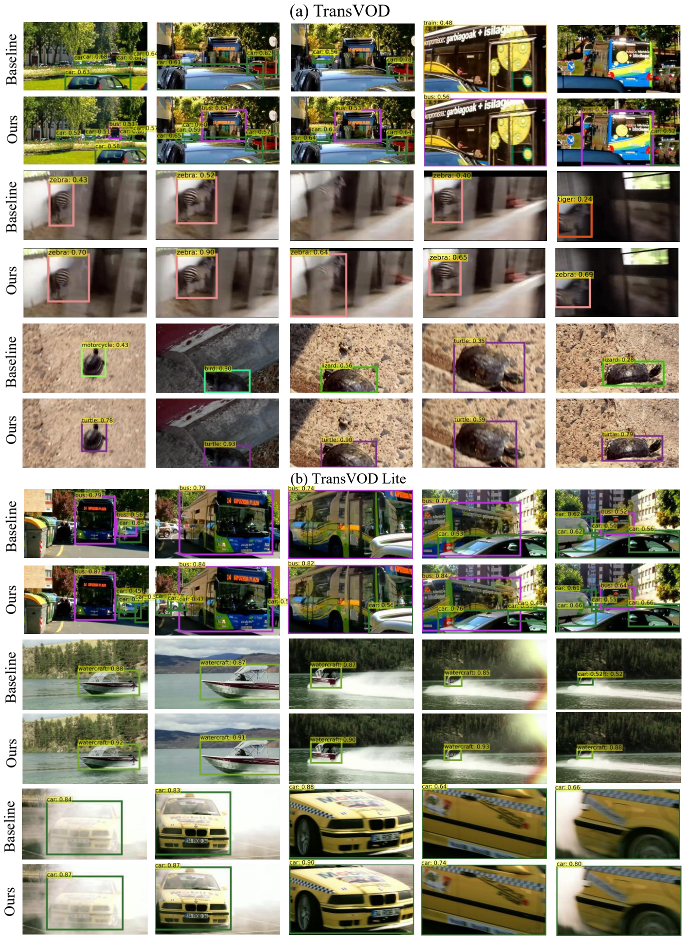


Fig. 10: The visualization results of single frame baseline method [11] and TransVOD (a), TransVOD Lite (b) in different scenarios. Compared with single frame baseline, our proposed TransVOD and TransVOD lite show better and consistent detection results in the cases of part occlusion (top two rows of (a) and (b)), motion blur (middle two rows of (a) and (b)) and rare pose (last two rows of (a) and (b)), respectively. Best view it on the screen and zoom in.

[8] Z. Cai and N. Vasconcelos, “Cascade r-cnn: High quality object detection and instance segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[9] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, “Object detection from scratch with deep supervision,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 2, pp. 398–412, 2019.

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[11] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” *arXiv preprint arXiv:2010.04159*, 2020.

[12] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, “Seq-nms for video object detection,” *arXiv preprint arXiv:1602.08465*, 2016.

[13] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang *et al.*, “T-cnn: Tubelets with convolutional neural networks for object detection from videos,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, 2017.

[14] H. Belhassen, H. Zhang, V. Fresse, and E.-B. Bourennane, “Improving video object detection by seq-bbox matching.” in *VISIGRAPP (5: VISAPP)*, 2019, pp. 226–233.

[15] A. Sabater, L. Montesano, and A. C. Murillo, “Robust and efficient post-processing for video object detection,” *arXiv preprint arXiv:2009.11050*, 2020.

[16] C.-H. Yao, C. Fang, X. Shen, Y. Wan, and M.-H. Yang, “Video object detection via object-level temporal aggregation,” in *European conference on computer vision*. Springer, 2020, pp. 160–177.

[17] Z. Jiang, Y. Liu, C. Yang, J. Liu, P. Gao, Q. Zhang, S. Xiang, and C. Pan, “Learning where to focus for efficient video object detection,” in *European Conference on Computer Vision*. Springer, 2020, pp. 18–34.

[18] M. Han, Y. Wang, X. Chang, and Y. Qiao, “Mining inter-video proposal relations for video object detection,” in *European Conference on Computer Vision*. Springer, 2020, pp. 431–446.

[19] L. Han, P. Wang, Z. Yin, F. Wang, and H. Li, “Exploiting better feature aggregation for video object detection,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1469–1477.

[20] L. Lin, H. Chen, H. Zhang, J. Liang, Y. Li, Y. Shan, and H. Wang, “Dual semantic fusion network for video object detection,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1855–1863.

[21] F. He, N. Gao, Q. Li, S. Du, X. Zhao, and K. Huang, “Temporal context enhanced feature aggregation for video object detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 10 941–10 948.

[22] K. Chen, J. Wang, S. Yang, X. Zhang, Y. Xiong, C. C. Loy, and D. Lin, “Optimizing video object detection via a scale-time lattice,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7814–7823.

[23] Y. Chen, Y. Cao, H. Hu, and L. Wang, “Memory enhanced global-local aggregation for video object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 337–10 346.

[24] G. Sun, Y. Hua, G. Hu, and N. Robertson, “Mamba: Multi-level aggregation via memory bank for video object detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2620–2627.

[25] C. Guo, B. Fan, J. Gu, Q. Zhang, S. Xiang, V. Prinet, and C. Pan, “Progressive sparse local attention for video object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3909–3918.

[26] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, “Deep feature flow for video recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2349–2358.

[27] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, “Flow-guided feature aggregation for video object detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 408–417.

[28] S. Wang, Y. Zhou, J. Yan, and Z. Deng, “Fully motion-aware network for video object detection,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 542–557.

[29] X. Zhu, J. Dai, L. Yuan, and Y. Wei, “Towards high performance video object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7210–7218.

[30] R. Jin, G. Lin, C. Wen, J. Wang, and F. Liu, “Feature flow: In-network feature flow estimation for video object detection,” *Pattern Recognition*, vol. 122, p. 108323, 2022.

[31] H. Deng, Y. Hua, T. Song, Z. Zhang, Z. Xue, R. Ma, N. Robertson, and H. Guan, “Object guided external memory network for video object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6678–6687.

[32] G. Bertasius, L. Torresani, and J. Shi, “Object detection in video with spatiotemporal sampling networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 331–346.

[33] Z. Jiang, P. Gao, C. Guo, Q. Zhang, S. Xiang, and C. Pan, “Video object detection with locally-weighted deformable neighbors,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8529–8536.

[34] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, “Relation distillation networks for video object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7023–7032.

[35] M. Shvets, W. Liu, and A. C. Berg, “Leveraging long-range temporal relationships between proposals for video object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9756–9764.

[36] M. Liu, M. Zhu, M. White, Y. Li, and D. Kalenichenko, “Looking fast and slow: Memory-guided mobile video object detection,” *arXiv preprint arXiv:1903.10172*, 2019.

[37] M. Liu and M. Zhu, “Mobile video object detection with temporally-aware feature maps,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5686–5695.

[38] K. Chen, J. Wang, S. Yang, X. Zhang, Y. Xiong, C. C. Loy, and D. Lin, “Optimizing video object detection via a scale-time lattice,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7814–7823, 2018.

[39] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," *arXiv preprint arXiv:2011.14503*, 2020.

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *The International Conference on Learning Representations (ICLR)*, 2021.

[41] P. Sun, Y. Jiang, R. Zhang, E. Xie, J. Cao, X. Hu, T. Kong, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple-object tracking with transformer," *arXiv preprint arXiv:2012.15460*, 2020.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[44] L. He, Q. Zhou, X. Li, L. Niu, G. Cheng, X. Li, W. Liu, Y. Tong, L. Ma, and L. Zhang, "End-to-end video object detection with spatial-temporal transformers," in *Proceedings of the 29th ACM International Conference on Multimedia (ACM MM)*, 2021, p. 1507–1516.

[45] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 761–769.

[46] Z. Wu, C. Shen, and A. v. d. Hengel, "High-performance semantic segmentation using very deep fully convolutional networks," *arXiv preprint*, 2016.

[47] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *ICCV*, 2021.

[48] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[49] H. Wu, Y. Chen, N. Wang, and Z. Zhang, "Sequence level semantics aggregation for video object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9217–9225.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[51] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[52] T.-W. Chin, R. Ding, and D. Marculescu, "Adascale: Towards real-time video object detection using adaptive scaling," *arXiv preprint arXiv:1902.02910*, 2019.

[53] X. Li, H. He, H. Ding, K. Yang, G. Cheng, J. Shi, and Y. Tong, "Improving video instance segmentation via temporal pyramid routing," *arXiv preprint arXiv:2107.13155*, 2021.

[54] Y. Chai, "Patchwork: A patch-wise attention network for efficient object detection and segmentation in video streams," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3414–3423, 2019.

[55] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," *arXiv preprint arXiv:2101.02702*, 2021.

[56] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[57] J. Zhang, X. Li, Y. Wang, C. Wang, Y. Yang, Y. Liu, and D. Tao, "Eatformer: Improving vision transformer inspired by evolutionary algorithm," *arXiv preprint arXiv:2206.09325*, 2022.

[58] X. Li, W. Zhang, J. Pang, K. Chen, G. Cheng, Y. Tong, and C. C. Loy, "Video k-net: A simple, strong, and unified baseline for video segmentation," in *CVPR*, 2022.

[59] X. Li, S. Xu, Y. Yang, G. Cheng, Y. Tong, and D. Tao, "Panoptic-partformer: Learning a unified model for panoptic part segmentation," in *Eur. Conf. Comput. Vis.*, 2022.

[60] S. Xu, X. Li, J. Wang, G. Cheng, Y. Tong, and D. Tao, "Fashionformer: A simple, effective and unified baseline for human fashion segmentation and recognition," in *Eur. Conf. Comput. Vis.*, 2022.

[61] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5188–5197.

[62] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *(CVPR)*, 2021.

[63] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV*, 2016.

[64] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *CVPR*, 2018.

[65] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *ICCV*, 2019.

[66] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *ECCV*, 2020.

[67] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Eur. Conf. Comput. Vis.*, 2020.

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[69] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[70] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3588–3597.

[71] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang *et al.*, "Sparse r-cnn: End-to-end object detection with learnable proposals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 454–14 463.

[72] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[73] R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-end people detection in crowded scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2325–2333.

[74] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[75] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.

[76] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *arXiv preprint arXiv:1605.06409*, 2016.

[77] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3057–3065.

[78] F. Wang, Z. Xu, Y. Gan, C.-M. Vong, and Q. Liu, "Scnet: Scale-aware coupling-structure network for efficient video object detection," *Neurocomputing*, vol. 404, pp. 283–293, 2020.

[79] Y. Qian, L. Yu, W. Liu, G. Kang, and A. G. Hauptmann, "Adaptive feature aggregation for video object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, 2020, pp. 143–147.

[80] F. Xiao and Y. J. Lee, "Video object detection with an aligned spatial-temporal memory," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 485–501.

[81] Y. Wu, H. Zhang, Y. Li, Y. Yang, and D. Yuan, "Video object detection guided by object blur evaluation," *IEEE Access*, vol. 8, pp. 208 554–208 565, 2020.

[82] H. Luo, L. Huang, H. Shen, Y. Li, C. Huang, and X. Wang, "Object detection in video with spatial-temporal context aggregation," *arXiv preprint arXiv:1907.04988*, 2019.

[83] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Minet: Meta-learning instance identifiers for video object detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 6879–6891, 2021.

[84] T. Gong, K. Chen, X. Wang, Q. Chu, F. Zhu, D. Lin, N. Yu, and H. Feng, "Temporal roi align for video object recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1442–1450.

[85] Y. Cui, L. Yan, Z. Cao, and D. Liu, "Tf-blender: Temporal feature blender for video object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 8138–8147.

[86] L. Han, P. Wang, Z. Yin, F. Wang, and H. Li, "Class-aware feature aggregation network for video object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2021.

[87] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[88] Z. Xu, E. Hrustic, and D. Vivet, "Centernet heatmap propagation for real-time video object detection," in *European Conference on Computer Vision*. Springer, 2020, pp. 220–234.

[89] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[90] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[91] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255.

[92] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[93] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014.