Object Detection Difficulty: Suppressing Over-aggregation for Faster and Better Video Object Detection

Bingqing Zhang
Renmin University of China
Beijing, China
bingqing.zhang@ruc.edu.cn

Sen Wang
The University of Queensland
Brisbane, Australia
sen.wang@uq.edu.au

Yifan Liu
The University of Adelaide
Adelaide, Australia
yifanliu0926@gmail.com

Brano Kusy
Data 61, CSIRO
Brisbane, Australia
brano.kusy@data61.csiro.au

Xue Li
The University of Queensland
Brisbane, Australia
xueli@itee.uq.edu.au

Jiajun Liu*
Data 61, CSIRO
Brisbane, Australia
jiajun.liu@csiro.au

ABSTRACT

Current video object detection (VOD) models often encounter issues with over-aggregation due to redundant aggregation strategies, which perform feature aggregation on every frame. This results in suboptimal performance and increased computational complexity. In this work, we propose an image-level Object Detection Difficulty (ODD) metric to quantify the difficulty of detecting objects in a given image. The derived ODD scores can be used in the VOD process to mitigate over-aggregation. Specifically, we train an ODD predictor as an auxiliary head of a still-image object detector to compute the ODD score for each image based on the discrepancies between detection results and ground-truth bounding boxes. The ODD score enhances the VOD system in two ways: 1) it enables the VOD system to select superior global reference frames, thereby improving overall accuracy; and 2) it serves as an indicator in the newly designed ODD Scheduler to eliminate the aggregation of frames that are easy to detect, thus accelerating the VOD process. Comprehensive experiments demonstrate that, when utilized for selecting global reference frames, ODD-VOD consistently enhances the accuracy of Global-frame-based VOD models. When employed for acceleration, ODD-VOD consistently improves the frames per second (FPS) by an average of 73.3% across 8 different VOD models without sacrificing accuracy. When combined, ODD-VOD attains state-of-the-art performance when competing with many VOD methods in both accuracy and speed. Our work represents a significant advancement towards making VOD more practical for real-world applications. The code will be released at https://github.com/bingqingzhang/odd-vod.

CCS CONCEPTS

• **Computing methodologies → Object detection**.

---

*Corresponding author

KEYWORDS

Video Object Detection, Efficient Video Perception, Object Detection Metrics, Feature Aggregation / Fusion

1 INTRODUCTION

Video Object Detection (VOD) [28, 46, 52] focuses on identifying objects within videos by utilizing rich spatial and temporal data. This task holds significant importance in the field of multimedia. VOD models typically build upon the success of modern Still Image Object Detectors (SIODs) [3, 34, 35, 51], sampling a series of reference frames and aggregating them to support the frame being processed. This approach has proven effective for enhancing frame feature representation and improving object detection performance. Consequently, many state-of-the-art models focus on designing aggregation modules, such as SELSA [46] and LRM in MEGA [4]. However, feature aggregation operations applied to every frame can introduce high computational costs and decrease detection speed, which ultimately limits the practical applicability of VODs in real-life applications. We refer to this as the *over-aggregation* problem. Moreover, low-quality reference frames may not provide any benefits in the VOD aggregation step due to their limited information content, resulting in suboptimal performance.

Methods for addressing the speed-accuracy trade-off in VODs have been proposed and can be categorized into two types: plug-in and unified methods. Plug-in methods [7] are employed alongside existing VOD models, while unified methods [22, 31, 50] combine two distinct strategies, aggregation and propagation, within the VOD algorithm. In key frames, unified methods use feature aggregation to improve detection accuracy, while in non-key frames, a propagation module updates motion information. Although unified methods alleviate the over-aggregation problem, their key-frame selection strategy or scheduler is overly simplistic, relying on either interval-based [50] or heuristic sampling [31]. Meanwhile, the quality of reference frames remains largely unexplored in VOD literature.

Figure 1: An overview of ODD-VOD. The dotted lines represent optional operations. There are five ODD-based components: ODD Metric, ODD Predictor for predicting ODD scores of testing frames. ODD Scheduler (ODD-A) switches between VOD and SIOD to accelerate the inference speed. ODD-based Global Reference Frame Selector (OGRFS) selects low-ODD-score frames (easy to detect) as global reference frames in the training stage (ODD-C) and inference stage (ODD-B) for better accuracy. The ODD-* labels are created for easier reference in the rest of the paper.

Figure 2: The predicted ODD scores have an inverse correlation with detection quality on two detectors. If the ODD score of a frame is low, the SIOD (Faster R-CNN) will have the same detection results as the video detector (SELSA). As the ODD score increases (making detection more difficult), the VOD will have significantly better results.

In this paper, we present an efficient VOD framework called Object Detection Difficulty-based VOD (ODD-VOD) to address the over-aggregation problem and enhance detection accuracy. The underlying intuition is twofold: first, feature aggregation is only necessary for frames that pose *difficulties* for SIODs; second, feature aggregation yields the most significant improvements over SIODs when the aggregated images are of high quality and easily detected by SIODs. To quantify the difficulty of an image for SIODs, we propose the ODD metric, which measures a ground-truth ODD score given a training image, its annotations, and its detection results from a pre-trained image detector. We then train an ODD Predictor that supports the ODD-VOD pipeline (Figure 1). The main technical contributions of our proposed framework are as follows:

(1) **ODD Metric**: We formulate a non-trivial metric to quantify the ODD score for an image.

(2) **ODD Predictor**: We first train an ODD Predictor module using quantified ODD scores to measure the detection difficulty of testing frames.

(3) **ODD-A**: We then propose an ODD Scheduler that switches between VOD and SIOD for a given input frame to increase inference speed.

(4) **ODD-B/C**: We present ODD-based Global Reference Frame Selector (OGRFS). As a plug-in method, OGRFS utilizes ODD scores to guide the selection of global reference frames, thereby improving the quality of aggregated features.

OGRFS can be utilized in two ways. It can be directly applied during inference without retraining the VOD model, resulting in improved detection accuracy (ODD-B). To further enhance detection accuracy, OGRFS can be employed during both training and inference stages (ODD-B+C).

Figure 2 illustrates how the proposed ODD score captures the difficulty levels for an SIOD and how this subsequently reflects the benefits of VOD/feature aggregation. We display the quantified ODD scores and the outputs of two detectors on a test video snippet. There is a strong inverse correlation between the predicted ODD scores and the accuracy of detection results from Faster R-CNN [35] (a typical SIOD). When the ODD scores are low (easy to detect), there is little difference between Faster R-CNN and SELSA [46] (VOD). As the ODD score increases (becoming harder to detect due to occlusion, blurriness and so on), Faster R-CNN starts to produce inaccurate object localization (blue box) and eventually misclassifies the object (red boxes) or even misses detections (purple box) (see more cases in Section 4.8). In our ODD-VOD, detection accuracy and speed can be simultaneously optimized using ODD scores. When a frame has a low ODD score, an SIOD can quickly process it without sacrificing accuracy and bypass VOD/feature aggregation. When the ODD score is higher, we can switch to VOD to detect these frames, maximizing aggregation benefits; furthermore, in the aggregation process, we use reference frames with low ODD scores for improved results.

In the Experiments section, we present extensive empirical results demonstrating that our proposed method can achieve significant speed improvements and enhanced accuracy across a range of widely-used VOD models.

2 RELATED WORK

Object Detection in Still Images As one of the most critical computer vision tasks, object detection has been extensively studied in academia and industry. Detectors in still images can be classified into two types: two-stage and one-stage. Two-stage detectors first extract objects from background areas and then determine their categories and positions [2, 9, 15, 35, 47]. On the other hand, one-stage detectors [14, 27, 30, 34, 44] can be faster than the two-stage ones. These detectors regard the detection task as a regression problem to obtain the types and positions of bounding boxes. Recent years, applying the transformer [43] to object detection is becoming a new research hotspot [3, 29, 32, 48].

Object Detection in Videos Unlike object detection in still images, video object detectors utilize spatial and temporal information in frames. Based on the different types of reference frames, existing methods can be classified into three categories: local-frame-based,

Figure 3: Training Pipeline for ODD Predictor. It takes four steps: 1) training an SIOD; 2) obtaining detection results of training images; 3) quantifying ODD scores; 4) using training images and ODD scores to train the ODD Predictor.

global-frame-based, and both-frame-based. Local-frame-based methods aggregate temporal information of nearby frames [1, 11, 26, 52, 53]. Global-frame-based methods select a range of reference frames in the whole videos and seek to enhance the features with semantic information [8, 16, 19, 28, 33, 38, 46]. Apart from the cite reference frames from the local or global position, MEGA [4] samples frames from both positions to obtain semantic information and motion information. However, aggregating on each frame may cause the over-aggregation problem, which brings high computational costs.

Therefore, efficient VODs are designed to reduce aggregation costs. Plug-in efficient methods [7] can be attached to other VODs for decreasing reference frames. Unifed-based efficient methods [17, 22, 31, 42, 50] combine dense (aggregation) and sparse (propagation) detectors to achieve faster speed. However, propagation between frames is vulnerable and sensitive to changes in objects' appearance and positions. Our method replaces the sparse detector with an SIOD to avoid the object motion problem.

3 METHOD

We first formally define the concept of ODD, and describe the process to quantify the ODD scores from the training set. We then present the details of ODD-VOD and its components.

3.1 Definition of Object Detection Difficulty

The most prevalent metric for evaluating the performance of various object detectors is the Mean Average Precision (mAP). However, mAP is a dataset-level metric, which makes it unsuitable for directly assessing the detection results on individual images. To address this limitation, we introduce the concept of Object Detection Difficulty (ODD), an image-level metric for object detection performance.

We propose an image-level metric for object detection difficulty, derived from the results of a SIOD model and the ground-truth bounding boxes in the training set. This image-level metric serves as the ground-truth signal for training an ODD Predictor, which can then estimate an ODD score for each image, making it a valuable tool for training or inference.

After obtaining the detection results from the SIOD model and the ground-truth bounding boxes for an image in the training set, we classify each predicted bounding box into four categories: positive, negative, near-positive, and multi-positive. If a bounding box has the maximum Intersection over Union (IoU) with a ground-truth

bounding box, and the IoU exceeds the positive threshold (t_1), we label this result as positive. If a bounding box's IoU is greater than t_1 but not the maximum, we classify this result as multi-positive. If a bounding box's IoU does not reach the positive threshold but is numerically close (e.g., t_1 is 0.5 and the IoU is 0.49), we designate this result as a near-positive sample. The concept of near-positive samples is inspired by the sampling strategy of Region Proposal Network (RPN) [35], which assigns positive samples in the second stage. Furthermore, the IoU of a near-positive sample should be larger than the near-positive threshold (t_2). Finally, if the IoU of a predicted bounding box is smaller than t_2, we categorize this result as negative. Among these categories, positive, near-positive, and multi-positive results contribute positively to the ground-truth ODD score.

Then, we can use a unified formula to define the weighted sample (ws). It can be written as:

$$ws(p) = \sum_{l \in L} \sum_{max(IoU_i)} C_{l,i} \mathbf{1}_{\mathbf{P}}(p) + \sum_{l \in L} \sum_{IoU_i \in [t_2, t_1]} \frac{1}{2} C_{l,i} \mathbf{1}_{\mathbf{NR}}(p)$$
$$+ \sum_{l \in L} \sum_{IoU_i \in [t_1, 1]} C_{l,i} \mathbf{1}_{\mathbf{M}}(p) + \sum_{l \in L} \sum_{IoU_i \in [0, t_2)} C_{l,i} \mathbf{1}_{\mathbf{N}}(p), \quad (1)$$

where L is the label of different objects, C is the confidence score of the detection result, t_1 is the positive threshold and t_2 is the near-positive threshold which is mentioned above. $\mathbf{1}_{\mathbf{P}}$, $\mathbf{1}_{\mathbf{NR}}$, $\mathbf{1}_{\mathbf{M}}$ and $\mathbf{1}_{\mathbf{N}}$ are indicator functions, which are used to determine whether the current result belongs to the corresponding sample (positive, near-positive, multi-positive and negative sample correspondingly). The output of this function has two values, 0 and 1. In addition, there is a parameter p in equation 1. When p is equal to 1, $\mathbf{1}_{\mathbf{P}}$, $\mathbf{1}_{\mathbf{NR}}$ and $\mathbf{1}_{\mathbf{M}}$ will work to find all positive samples. And when p is 0, ws will calculate weighted negative samples. The weighted sample is an important measurement to calculate the output of a detector on one image. Borrowing the idea of the F1-score, we also use harmonic means to balance different samples.

First, the weighted precision (wp) can be defined as:

$$wp = \begin{cases} \dfrac{ws(p=1)}{ws(p=1)+ws(p=0)} \\ \\ 1, \text{if no gt bbox} \end{cases} \quad (2)$$

Here the denominator cannot be 0 due to the structure of object detectors. And the weighted recall (wr) can be defined as:

$$wr = \begin{cases} \dfrac{ws(p=1)}{max(\text{total\_gt\_sample}, ws(p=1))}, \\ \\ 1, \text{if no gt bbox} \end{cases} \quad (3)$$

where total_gt_sample is the total number of ground truth proposals in one image. Finally, we can define the ODD with wp and wr:

$$ODD = 1 - 2 \cdot \frac{wp \times wr}{wp + wr + \varepsilon}, \quad (4)$$

where ε is a tiny value to prevent the denominator from being 0. The value range of ODD is between 0 and 1. If the ODD score is high, it means the current image is hard to detect for the object detector and vice versa.

Figure 4: ODD Scheduler for faster detection. The input is predicted ODD scores from ODD Predictor. Then, the ODD Scheduler will use the ODD score to determine which detector to detect the current frame (Dispatching). After generating results, ODD Scheduler will collect them together (Collecting). In addition, ODD-B (OGRFS in inference) can be used to improve detection accuracy.

3.2 ODD-VOD Framework

An overview of the ODD-VOD framework is presented in Figure 1. The basic input of the framework is the quantified ODD scores (ODD ground truth scores), which can be calculated from an SIOD and the training dataset. ODD Predictor is used to predict ODD scores of testing frames, which is an important input for the ODD Scheduler. The ODD Scheduler (ODD-A) is deployed in a hybrid detection pipeline for faster speed. OGRFS can be used in the training stage (ODD-C) and inference stage (ODD-B), which select lowest k ODD score global reference images for better detection accuracy. We now introduce the ODD-based components in detail.

Training the ODD Predictor

The ODD Predictor is a **novel head** attached to an SIOD model, and its training serves as the initial stage for the ODD-VOD framework. The primary goal of the ODD Predictor is to estimate the ODD score for a given input frame.

Four steps are involved in training an ODD Predictor, as illustrated in Figure 3. First, an SIOD model is trained on the training dataset, or a pre-trained detector can be used. Second, the trained SIOD model is executed to produce detection results. Then, with the detection and ground-truth results collected, the method described in Section 3.1 is employed to generate the ODD score set, which serves as the supervision signal for ODD Predictor training. Finally, the ODD Predictor is trained with the backbone frozen, ensuring that the training process does not affect the object detection (DET) head.

The ODD Predictor is a lightweight auxiliary head with four layers: one layer of convolutional network with 3 by 3 kernel, one layer of adaptive average pooling with 7 by 7 kernel, and two fully connected layers. For a video frame x_i, the ODD score y_i can be calculated as:

$$y_i = ODD(f(x_i)), \tag{5}$$

where f is the detector's backbone and x_i is the training image. y_i ranges from 0 to 1. And we use the smooth L1 loss [15] to optimize

Figure 5: OGRFS for Better Detection. Frames with lower ODD scores tend to yield more accurate detection outcomes and can serve as reliable reference frame for other hard-to-detect frames during feature aggregation, thus improving overall performance.

the ODD Predictor:

$$ODDloss(z_i) = \begin{cases} 0.5z_i^2, \text{if } |z_i| < 1 \\ |z_i| - 0.5, \text{otherwise} \end{cases} \tag{6}$$

Here, $z_i = gt_i - 10y_i$. gt_i is the ground truth ODD score of frame x_i, which can be obtained in step 3. Note that we magnify the value of y_i by 10 times to increase the convergence speed when calculating the ODD loss.

ODD Scheduler for Faster Detection (ODD-A)

Figure 4 illustrates the working process of the ODD Scheduler, which functions as a hybrid detection pipeline. It comprises two object detectors: a Still Image Object Detector (SIOD) and a Video Object Detector (VOD). While the SIOD can detect objects more rapidly but with reduced detection accuracy, the VOD can achieve superior detection results, albeit at a slower speed.

During the inference stage, the ODD Predictor first assigns scores to the frames. If the predicted ODD score for the current frame is below the ODD threshold (indicating an easy-to-detect frame), the SIOD detection head will directly process it. Otherwise, the ODD Scheduler will delegate this frame (considered hard-to-detect) to the Video Object Detector, representing the dispatching process. The ODD Scheduler subsequently collects all detected results in their original order.

Thus, the ODD-VOD framework processes a video in two rounds. In the first round, the SIOD equipped with the ODD Predictor evaluates the entire video and performs detection when necessary. In the second round, the VOD detects the remaining frames.

OGRFS for Better Detection (ODD-B/C)

For global-frame-based VODs, selecting global frames from the entire video as reference frames for aggregation is essential. However, existing sampling strategies are typically naive. The most common strategy involves randomly selecting several frames as global frames. In ODD-VOD, we propose the ODD-based Global Reference Frame Selection (OGRFS) method as a more sophisticated selection strategy, as illustrated in Figure 5. Given a number k as a parameter for the reference frames to be aggregated, OGRFS selects

Table 1: Main results for the full ODD-VOD framework (ODD-Full, i.e. ODD-A+[B+C], all ODD components used when applicable to the VOD model). We use ResNet-50 backbone for all models. The bold numbers mean settings on which acceleration is achieved without sacrificing accuracy, and the biggest lossless acceleration rate is shown in the rightmost column. On the other hand, when the ODD Threshold is set to near 0, we can get maximum accuracy improvement in this table, and the corresponding mAP gains over the original VOD models are given in the second rightmost column.

VOD Model	ODD-Strategy		ODD Theshold												Original Results	Lossless Acc. Rate
			0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.25	0.2	0.15	0.1	0.05		
FGFA [52]	ODD-A	mAP	72.1	72.8	73.1	73.4	73.8	71.2	74.5	**74.7**	**74.7**	**74.7**	**74.7**	**74.7**	74.7	↑104.2%
		FPS	29.6	27.5	25.9	23.8	19.4	17.2	15.3	**14.5**	**13.6**	**13.1**	**12.2**	**11.4**	7.1	
SELSA [46]	ODD-A+B+C	mAP	73.3	74.4	75.0	75.8	76.7	77.6	**78.7**	**79.2**	**79.5**	**79.7**	**79.8**	**80.0**	78.4	↑72.4%
		FPS	34.1	33.7	30.7	30.3	25.6	22.7	**20.0**	**19.1**	**18.1**	**16.6**	**15.4**	**14.2**	11.6	
TROIA [16]	ODD-A+B+C	mAP	73.9	75.5	76.3	76.7	77.7	78.8	79.7	**79.9**	**80.1**	**80.2**	**80.4**	**80.6**	79.8	↑66.7%
		FPS	28.7	24.6	22.5	19.7	15.9	13.0	10.8	**10.0**	**9.1**	**8.4**	**7.6**	**6.9**	6.0	
RDN(local) [11]	ODD-A	mAP	72.0	72.5	72.7	73.0	73.7	74.2	74.8	75.1	75.3	75.5	75.6	**75.7**	75.7	↑13.9%
		FPS	26.6	23.1	21.4	19.0	15.7	13.6	11.4	10.8	9.8	9.5	8.9	**8.2**	7.2	
RDN(global) [4]	ODD-A+B+C	mAP	73.1	74.1	74.9	75.6	76.5	**77.6**	**78.9**	**79.2**	**79.4**	**79.7**	**79.9**	**80.1**	77.6	↑70.3%
		FPS	36.2	34.5	33.0	31.5	27.9	**24.7**	**22.7**	**21.7**	**19.5**	**18.7**	**17.6**	**16.0**	14.5	
MEGA [4]	ODD-A+B+C	mAP	72.3	73.0	73.4	74.2	75.2	76.1	**77.2**	**77.6**	**77.9**	**78.2**	**78.4**	**78.5**	77.0	↑112.5%
		FPS	25.5	21.9	19.5	18.2	14.3	12.4	**10.2**	**9.6**	**8.7**	**8.2**	**7.7**	**6.9**	4.8	

Table 2: Main results for ODD-VOD. We use ResNet-101 backbone for all models.

Model		ODD Threshold						Original Results	Acc. Rate
		0.8	0.4	0.2	0.15	0.1	0.05		
FGFA	mAP	76.9	77.1	77.6	**77.8**	**77.8**	**77.8**	77.8	↑98.4%
	FPS	25.7	16.7	12.5	**12.1**	**11.1**	**10.6**	6.3	
SELSA	mAP	78.4	80.2	**81.7**	**82.1**	**82.5**	**82.7**	81.5	↑50.5%
	FPS	28.3	17.9	**15.8**	**15.0**	**13.9**	**12.8**	10.5	
TROIA	mAP	80.6	81.5	**82.7**	**82.9**	**83.6**	**83.9**	82.6	↑62.7%
	FPS	20.5	11.7	**8.3**	**7.5**	**6.8**	**6.2**	5.1	
RDN (Local)	mAP	77.2	78.9	80.5	81.2	81.5	**81.7**	81.7	↑18.0%
	FPS	21.6	12.3	8.9	7.8	7.8	**7.2**	6.1	
RDN (Global)	mAP	80.4	81.5	**82.3**	**82.8**	**82.3**	**83.6**	81.7	↑45.6%
	FPS	29.6	18.2	**16.6**	**16.1**	**15.7**	**15.2**	11.4	
MEGA	mAP	80.8	81.7	82.8	**83.2**	**83.9**	**84.1**	82.9	↑86.1%
	FPS	19.6	10.9	7.4	**6.7**	**5.9**	**5.3**	3.6	

the frames with the k lowest ODD scores to create a global frame pool.

OGRFS can be employed in two distinct ways. One approach is to directly incorporate OGRFS into the VOD inference stage (ODD-B), which enhances detection accuracy without re-training the models. Alternatively, OGRFS can be further utilized to train global-frame-based VOD models (ODD-C), resulting in superior detection accuracy compared to the inference-only version. In summary, OGRFS helps avoid selecting deteriorated frames as global reference frames during both training and inference stages.

4 EXPERIMENTS

4.1 Experiment Setup

Dataset and Evaluation. We conduct our experiments on the ImageNet VID dataset [36], which is the most widely used benchmark for video object detection [4, 46, 52]. The dataset consists of 3,862 video snippets for training and 555 video snippets for validation. All video frames are fully annotated with bounding boxes and object categories, covering over 30 categories.

To evaluate the performance of detection results, we follow the common practice in VOD [37, 49] and report mean Average Precision (mAP) at 0.50 IoU on the validation set as the accuracy metric and runtime as the speed metric. It is worth noting that runtime may vary due to system factors. Therefore, we typically take the average of multiple measurements. Moreover, two types of runtime are widely used in previous work: runtime (ms) and runtime (FPS). We choose runtime (FPS) because we believe that FPS provides a more intuitive understanding of a model's inference speed.

Training and Inference Details. We implemented our code using PyTorch 1.12.0. For the main experiments, we employed the MMTracking toolbox [6], a widely-used open-source platform for video object detection, to implement the methods and compare their performance. We utilized Faster R-CNN with ResNet-50 and ResNet-101 [23] backbones, initialized with ImageNet pre-trained weights, as the still image object detector. In the generalizability experiments (Section 4.3), we implemented our ODD-VOD framework using the corresponding official codebases [37, 49]. It is important to note that image preprocessing varies across different methods. For YOLOV experiments, we resized input frames to 576×576. In contrast, for other experiments, we resized input frames to 1000×600.

For training, we employed three NVIDIA GeForce RTX 3090 GPUs to train the ODD predictor, Faster R-CNN, and the VOD models with OGRFS (ODD-B) when applicable, in parallel. The process unfolded as follows: first, we trained Faster R-CNN for 7 epochs and then generated the ODD ground truth scores on the training dataset, with the near-positive threshold (t_1) set to 0.3 and the positive threshold (t_2) set to 0.5. After generating the ODD ground truth scores, we froze the backbone to train the ODD Predictor. We set the training iteration to 220k. The initial learning rate was set to 3.75×10^{-3} and was divided by 10 at 110k and 165k iterations, respectively. Subsequently, we trained several VOD

Table 3: Generalisability for ODD Metric in Different ODD-VOD Combinations

Still Image Detector	Video Object Detector		ODD Threshold					Original Results
			0.7	0.4	0.2	0.1	0.05	
Deformable DETR (Swin-Base)	TransVOD Lite (Swin-Base)	mAP	88.3	89.1	89.9	90.1	90.1	90.1
		FPS	16.1	14.7	13.0	11.7	11.0	9.9
YOLOX-s	YOLOV-s	mAP	73.0	74.8	76.3	77.5	77.9	77.9
		FPS	193.8	184.0	176.9	170.3	166.1	161.3
YOLOX-l	YOLOV-l	mAP	77.7	79.2	81.2	82.7	83.4	83.4
		FPS	142.0	135.5	125.9	120.1	117.3	113.0
YOLOX-x	YOLOV-x	mAP	83.1	84.3	85.2	85.6	85.6	85.6
		FPS	105.7	95.5	90.1	84.4	82.5	78.1
YOLOX-s	YOLOV-x	mAP	74.5	79.2	82.4	83.6	84.3	85.6
		FPS	182.6	151.4	128.0	103.7	91.1	78.1

Table 4: OGRFS for Global-based VOD Training & Inference.

Model	ODD-B		✔	✔
	ODD-C			✔
	Backbone		mAP@0.50	
RDN(Global) [4]	ResNet-50	77.6	79.1	80.1
	ResNet-101	82.5	83.0	83.6
TROIA [16]	ResNet-50	79.8	80.1	80.6
	ResNet-101	82.6	83.4	84.0
SELSA [46]	ResNet-50	78.4	79.7	80.1
	ResNet-101	81.5	81.8	82.7

models with OGRFS. The training iteration remained the same as for the ODD Predictor. We followed the dataset protocols widely used in [4, 24, 37, 52].

In the main experiments, we evaluated ODD-VOD's performance on six different VODs to test its effectiveness: FGFA [52], SELSA [46], MEGA [4], RDN(local) [11], RDN(global), and Temporal RoI Align (TROIA) [16]. To obtain comparable results for inference speed, we utilized a single NVIDIA GeForce RTX 3090 GPU to evaluate the ImageNet VID dataset and set the batch size in Faster R-CNN and VOD to 1. In the generalizability experiments, we directly reused ODD predicted scores obtained from the main experiments and combined ODD-VOD with the two latest state-of-the-art models, namely YOLOV (YOLO series) [37] and TransVOD_Lite (DETR series) [49], to achieve better performance. The batchsize of SIODs was aligned with the original models, i.e., 1 for SELSA, 12 for TransVOD_Lite and 32 for YOLOV.

4.2 Main Results

Tables 1 and 2 present the detection accuracy and speed on various VOD models with different ODD thresholds, employing two different backbones (ResNet-50 and ResNet-101). The ODD strategy in Table 1 indicates the ODD components (ODD-A, ODD-B, and ODD-C) included in the models. In Table 2, the corresponding ODD strategies remain consistent. The right column displays the results of the original VOD model. Among these VOD models, FGFA and RDN (local) are local-frame-based methods; SELSA, TROIA, and RDN (global) are global-frame-based; MEGA is both-frame-based. These six models represent all types of typical precision-oriented VODs. The bold numbers in the tables signify the accuracy lossless ODD settings, which are recommended speed-accuracy trade-offs.

The ODD threshold is used by ODD-A for frame assignment to either an SIOD or a VOD process. If the predicted ODD score falls below the ODD threshold, Faster R-CNN will detect the frame. Otherwise, the frame will be processed by the VOD. Consequently, as the ODD threshold decreases, detection accuracy increases while detection speed slows down. The ODD threshold can be flexibly adjusted to cater to different application requirements. Furthermore, we find that setting the ODD Threshold to 0.2 allows most SIOD-VOD combinations to achieve lossless acceleration. Therefore, **0.2 is a recommended ODD threshold** for our ODD-VOD framework.

Furthermore, Table 1 validates the existence of the *over-aggregation* phenomenon. When we set the ODD threshold to a specific number or below, we can achieve the same detection results with a

faster inference speed. For instance, in FGFA detection, when we set the ODD threshold to 0.25 for ResNet-50 and 0.15 for ResNet-101, ODD-VOD can correspondingly achieve the same mAP as the initial results with 104% and 98.4% inference speed, respectively. For global-frame-based and both-frame-based methods (SELSA, TROIA, RDN (global), and MEGA), OGRFS makes the aggregation operation more efficient, resulting in better detection accuracy and faster detection outcomes. For example, when we set the ODD threshold to 0.3, SELSA can achieve a 79.2 mAP and 20.0 FPS (compared to the initial results: 78.4 mAP and 11.6 FPS).

4.3 Generalisability for ODD-VOD (ODD-A).

In the main experiments, we use Faster R-CNN as the SIOD to verify the effectiveness of ODD-A with the defined ODD metric in different VODs. We then further explore the generalizability of ODD-VOD on various combinations of SIODs and VODs. Here we directly reuse the predicted ODD scores mentioned earlier. We include Deformable DETR [51] and YOLOX [14] as alternative SIOD options to Faster R-CNN and test both SIODs on the very recent and competitive VOD models, TransVOD Lite [49] and YOLOV [37], to validate the generalizability and flexibility of ODD-VOD. Table 3 displays the detection results at different ODD thresholds. We observe consistent performance gains for both SIODs combined with TransVOD Lite and YOLOV. This confirms that the ODD-A component in ODD-VOD is robust to the choice of SIOD. Therefore, we argue that the proposed ODD metric is suitable for various detectors (detector-agnostic) because most detectors share common challenging detection cases, such as small and blurry objects.

4.4 Evaluation of OGRFS (ODD-B/C)

Table 4 shows the effects of ODD for global-frame-based VOD training and inference. The results show that the OGRFS can improve detection accuracy even with only a pre-trained model in the inference stage (ODD-B). We use SELSA and TROIA (78.4 and 79.8 mAP correspondingly) with the model and weights directly downloaded from the Open-MMLab [1]. OGRFS achieves better detection accuracy (79.7 and 80.1 mAP correspondingly) without retraining. This means ODD-B can be a low-cost plug-in for better accuracy, for any pretrained VOD model in this category. Furthermore, if we use OGRFS for both retraining and inference (ODD-B+C), ODD-VOD outperforms all existing VOD models consistently.

---

[1]https://mmtracking.readthedocs.io/en/latest/model_zoo.html

4.5 Design of the ODD Quantifying Metric.

In Section 3.1, we defined the categorized output bounding boxes into four kinds of results: positive, negative, near-positive, and multi-positive. The notion of positive and negative results is widely used in F1-score and mAP. In this ablation study, we prove the effectiveness of near-positive and multi-positive results. Table 5 shows the detection results (accuracy and speed) of ODD-based SELSA with different components with the ODD threshold from 0.7, 0.4 to 0.2. When we use all components, the detection results can perform best in most cases. If we use two of them, the performance can be better than only using positive & negative detections.

Table 5: Ablation Study for Object Detection Difficulty Design

Component		positive&negative	✔	✔	✔
		near-positive		✔	✔
		multi-positive			✔
ODD Threshold	0.7	mAP	74.8	75.0	**75.1**
		FPS	30.5	**31.4**	30.7
	0.4	mAP	77.1	77.3	**77.7**
		FPS	21.9	19.6	**22.7**
	0.2	mAP	79.0	**79.3**	79.2
		FPS	17	14.5	**18.1**

4.6 Comparison with Efficient VOD Methods

Comparison with Plug-in Efficient VODs.

Plug-in efficient VOD methods can be attached to other video object detectors for faster detection speed without significant accuracy losses. The proposed ODD-VOD framework addresses the issue of over-aggregation by alternating between the SIOD and VOD, which is a typical plug-in approach. However, only a few methods are designed in plug-in form. To the best of our knowledge, DFA [7] is also a plug-in efficient method addressing this challenge by designing a plug-in module to dynamically aggregate features for off-the-shelf video object detectors. It can reduce the computational cost and improve the detection speed when attached to other VOD models. It is worth mentioning that some plug-in VOD methods [8, 20] also exist which enhance feature aggregation for better detection accuracy. However, these methods are not considered efficient approaches because they sacrifice detection speed to achieve higher precision. In Table 6, we compare ODD-VOD with DFA (including Vanilla DA and Deformable DA) to show the detection results changes, including FPS and mAP. Since DFA is deployed on a different hardware platform, one reasonable way is to compare the gains of the base VOD models after the enhancement frameworks are applied. Overall, ODD-enhanced methods can achieve the best results without accuracy losses, while DFA will lead to a decline in detection accuracy. This indicates that DFA can partially alleviate the over-aggregation problem for speedups, but it comes at the cost of the model's accuracy.

Comparison with Unified Efficient VOD Methods.

Unified methods deeply optimize the process of feature aggregation to obtain better detection speed. In Table 7, we compare our ODD (Plug-in trade-off) with some representative state-of-the-art

Table 6: Comparison with plug-in efficient VODs. Since the results of Vanilla DA and Deformable DA are reported from different platforms, we compare relative gains here

Methods	Backbone	FPS Gains	mAP Gains
FGFA + Vanilla DA	ResNet-50	↑ 2.1 $_{5.8\rightarrow7.9}$	↓ 0.4 $_{74.3\rightarrow73.9}$
FGFA + Deformable DA	ResNet-50	↑ 1.8 $_{5.8\rightarrow7.6}$	↓ 0.2 $_{74.3\rightarrow74.1}$
FGFA + ODD (ours)	ResNet-50	↑ 7.4 $_{7.1\rightarrow14.5}$	0.0 $_{74.7\rightarrow74.7}$
SELSA + Vanilla DA	ResNet-50	↑ 4.4 $_{5.0\rightarrow9.4}$	↓ 1.4 $_{77.9\rightarrow76.5}$
SELSA + Deformable DA	ResNet-50	↑ 3.8 $_{5.0\rightarrow8.8}$	↓ 0.4 $_{77.9\rightarrow77.5}$
SELSA + ODD (ours)	ResNet-50	↑ 8.4 $_{11.6\rightarrow20.0}$	↑ 0.3 $_{78.4\rightarrow78.7}$
TROIA + Vanilla DA	ResNet-50	↑ 2.4 $_{1.5\rightarrow3.9}$	↓ 1.2 $_{79.0\rightarrow77.8}$
TROIA + Deformable DA	ResNet-50	↑ 2.1 $_{1.5\rightarrow3.6}$	↓ 0.2 $_{79.0\rightarrow78.8}$
TROIA + ODD (ours)	ResNet-50	↑ 4.0 $_{6.0\rightarrow10.0}$	↑ 0.1 $_{79.8\rightarrow79.9}$
FGFA + Vanilla DA	ResNet-101	↑ 2.4 $_{5.1\rightarrow7.5}$	↓ 0.4 $_{77.6\rightarrow77.2}$
FGFA + Deformable DA	ResNet-101	↑ 2.0 $_{5.1\rightarrow7.1}$	↓ 0.1 $_{77.6\rightarrow77.5}$
FGFA + ODD (ours)	ResNet-101	↑ 5.8 $_{6.3\rightarrow12.1}$	0.0 $_{77.8\rightarrow77.8}$
SELSA + Vanilla DA	ResNet-101	↑ 3.0 $_{4.5\rightarrow7.5}$	↓ 1.3 $_{81.3\rightarrow80.0}$
SELSA + Deformable DA	ResNet-101	↑ 2.6 $_{4.5\rightarrow7.1}$	↓ 0.3 $_{81.3\rightarrow81.0}$
SELSA + ODD (ours)	ResNet-101	↑ 4.5 $_{10.5\rightarrow15.0}$	↑ 0.6 $_{81.5\rightarrow82.1}$
TROIA + Vanilla DA	ResNet-101	↑ 2.4 $_{1.2\rightarrow3.6}$	↓ 0.6 $_{82.4\rightarrow81.8}$
TROIA + Deformable DA	ResNet-101	↑ 2.1 $_{1.2\rightarrow3.3}$	↓ 0.4 $_{82.4\rightarrow82.0}$
TROIA + ODD (ours)	ResNet-101	↑ 2.4 $_{5.1\rightarrow7.5}$	↑ 0.3 $_{82.6\rightarrow82.9}$

Table 7: Comparison with SOTA VOD Methods

Model	Backbone	FPS	mAP@0.5
Precision-Oriented VOD			
SELSA [46]	ResNet-50	11.6	78.4
FGFA [52]	ResNet-101	6.3	77.8
SELSA [46]	ResNet-101	10.5	81.5
TROIA [16]	ResNet-101	5.1	82.6
LWDN [25]	ResNet-101	20(X)	76.3
MAMBA [41]	ResNet-101	11.1(RTX)	80.8
MINet [12]	ResNet-101	7.5(V)	80.2
LRTR [39]	ResNet-101	10(X)	80.6
DSFNet [28]	ResNet-101	-	84.1
EBFA [18]	ResNet-101	-	84.8
TransVOD Lite [24, 49]	SwinBase	9.9	**90.1**
Unified Efficient VOD			
SparseVOD [21]	ResNet-50	14.4(A100)	80.3
DFF [53]	ResNet-101	39.8(V100)	73.5
OGEMN [10]	ResNet-101	14.9(1080Ti)	76.8
QueryProp [22]	ResNet-101	26.8(X)	82.3
THP [50]	ResNet-101	13.0(K40)	78.6
DorT [31]	ResNet-101	7.8(X)	75.8
PSLA [17]	ResNet-101	30.8(V)/18.7(X)	77.1
LSTS [40]	ResNet-101	23.0(V)	77.2
EOVOD [42]	YOLOX-m	50.5(V100)	74.5
YOLOV [37]	YOLOX-s	**161.3**	77.9
YOLOV [37]	YOLOX-x	78.1	85.6
ODD-VOD (ours)			
SELSA+ODD	ResNet-50	20.0	78.7
FGFA+ODD	ResNet-101	12.1	77.8
SELSA+ODD	ResNet-101	15.8	81.7
TROIA+ODD	ResNet-101	6.6	82.7
TransVOD Lite+ODD	SwinBase	11.7	**90.1**
YOLOV+ODD	YOLOX-s	**166.1**	77.9
YOLOV+ODD	YOLOX-x	84.4	85.6

methods including some latest methods, e.g. [21, 22, 37]. The remarks in column FPS mean different GPU platforms. Table 7 has three parts. The top part lists some precision-oriented VOD models which are designed to achieve better detection accuracy but may encounter the over-aggregation problem. The middle part displays some unified efficient VOD methods focusing on detection speed. The bottom part shows our methods. Finally, our ODD-VOD is competitive with all these methods and can achieve state-of-the-art performance when integrated with the latest methods, such as TransVOD Lite+ODD and YOLOV+ODD.

Table 8: Proportion of Frames Processed by SIOD Detection Head.

ODD Thresh	0.9	0.7	0.5	0.3	0.2	0.1
Proportion	90.8%	84.0%	71.5%	52.2%	39.6%	25.0%

4.7 Time Efficiency

In comparison to other efficient VOD approaches, ODD-VOD is able to accelerate the detection speed directly from the base SIOD models. Consequently, we investigate the percentage of frames processed by the SIOD detection head under various ODD threshold settings (refer to Table 8). It is evident that the ODD metric tends to assign higher scores to input frames. Remarkably, even when the ODD threshold is set to 0.1, approximately one-quarter of the frames are directly processed by the SIOD detection head, resulting in favorable speed-accuracy trade-offs for the majority of VOD methods.

We further compute the GFLOPs for each component of our ODD-VOD method. The GFLOPs of Faster R-CNN (r50) equipped with an ODD predictor were 131.63, with the detection head contributing 45.09 and the ODD predictor (head) contributing a mere 0.13. Consequently, the ODD head accounted for only 0.1% of the model complexity, exerting negligible impact on detection speed. The VODs employed in our primary experiment were considerably more complex than SIOD; for example, SELSA had a GFLOP of 324.2. In summary, the ODD predictor can be regarded as a *cost-effective* means to bridge the gap between SIOD and VOD.

4.8 Visualization

Figure 6 shows more visualization results on the relationship between predicted ODD scores and the results for two different detectors (Faster R-CNN for a typical SIOD and SELSA for a typical VOD). Predicted ODD score could accurately characterize the detection difficulty for Faster R-CNN in most cases. When the predicted ODD score is below 0.2, Faster R-CNN mainly outputs correct proposals. This phenomenon can also be proved by Table 1: when we set the ODD threshold to 0.2, our ODD-VOD (ODD-Full) can achieve the same or better detection accuracy and faster detection speed compared with the original video object detector. When the predicted ODD score is 0.7 or above, these frames will likely be challenging for Faster R-CNN to detect. Therefore, the feature aggregation module can get maximum detection benefit for the frames with ODD scores of 0.7 or above.

Furthermore, as an interpretation of what the ODD score is really measuring, we find that many frames with higher ODD scores

Figure 6: Visualization of the Relationship between Predicted ODD Scores and Detection Results.

exhibit motion blur, video defocus, part occlusion, rare poses, poor illumination, small object scales, and other attributes that affect detection difficulty. A high ODD score could be derived from a single or a combination of such attributes. Therefore, we argue that there is no simple heuristic/rule-based method to measure detection difficulty. Instead, the proposed ODD Predictor manages to capture that complex signal with the proper supervision from the ODD Metric, and is able to produce robust ODD predictions across different scenarios.

5 CONCLUSION

In this paper, we define an image-level metric named Object Detection Difficulty (ODD) to measure the difficulty of object detection for a given image. We also design an ODD-VOD framework using ODD scores to suppress over-aggregation for faster video object detection speed. In addition, using ODD scores, a module in ODD-VOD called OGRFS helps the global-frame-based VOD models select better reference frames for better detection accuracy. Finally, extensive experiments demonstrate that our method can achieve faster and better video object detection than other VOD methods tested.

ACKNOWLEDGMENTS

REFERENCES

[1] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. 2018. Object Detection in Video with Spatiotemporal Sampling Networks. *ArXiv* abs/1803.05549 (2018).

[2] Zhaowei Cai and Nuno Vasconcelos. 2017. Cascade R-CNN: Delving Into High Quality Object Detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017), 6154–6162.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-End Object Detection with Transformers. *ArXiv* abs/2005.12872 (2020).

[4] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. 2020. Memory Enhanced Global-Local Aggregation for Video Object Detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 10334–10343.

[5] Manri Cheon, Sung-Jun Yoon, Byungyeon Kang, and Junwoo Lee. 2021. Perceptual Image Quality Assessment with Transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2021), 433–442.

[6] MMTracking Contributors. 2020. MMTracking: OpenMMLab video perception toolbox and benchmark. https://github.com/open-mmlab/mmtracking.

[7] Yiming Cui. 2022. DFA: Dynamic Feature Aggregation for Efficient Video Object Detection. *ArXiv* abs/2210.00588 (2022).

[8] Yiming Cui, Liqi Yan, Zhiwen Cao, and Dongfang Liu. 2021. TF-Blender: Temporal Feature Blender for Video Object Detection. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 8118–8127.

[9] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *ArXiv* abs/1605.06409 (2016).

[10] Hanming Deng, Yang Hua, Tao Song, Zongpu Zhang, Zhengui Xue, Ruhui Ma, Neil Martin Robertson, and Haibing Guan. 2019. Object Guided External Memory Network for Video Object Detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 6677–6686.

[11] Jiajun Deng, Yingwei Pan, Ting Yao, Wen gang Zhou, Houqiang Li, and Tao Mei. 2019. Relation Distillation Networks for Video Object Detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 7022–7031.

[12] Jiajun Deng, Yingwei Pan, Ting Yao, Wen gang Zhou, Houqiang Li, and Tao Mei. 2021. MINet: Meta-Learning Instance Identifiers for Video Object Detection. *IEEE Transactions on Image Processing* 30 (2021), 6879–6891.

[13] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. 2020. Image Quality Assessment: Unifying Structure and Texture Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44 (2020), 2567–2581.

[14] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. 2021. YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430* (2021).

[15] Ross B. Girshick. 2015. Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), 1440–1448.

[16] Tao Gong, Kai Chen, Xinjiang Wang, Qi Chu, Feng Zhu, Dahua Lin, Nenghai Yu, and Huamin Feng. 2021. Temporal ROI Align for Video Object Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 1442–1450.

[17] Chaoxu Guo, Bin Fan, Jie Gu, Q. Zhang, Shiming Xiang, Véronique Prinet, and Chunhong Pan. 2019. Progressive Sparse Local Attention for Video Object Detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 3908–3917.

[18] Liang Han, Pichao Wang, Zhaozheng Yin, F. Wang, and Hao Li. 2020. Exploiting Better Feature Aggregation for Video Object Detection. *Proceedings of the 28th ACM International Conference on Multimedia* (2020).

[19] Mingfei Han, Yali Wang, Xiaojun Chang, and Y. Qiao. 2020. Mining Inter-Video Proposal Relations for Video Object Detection. In *European Conference on Computer Vision*.

[20] Khurram Azeem Hashmi, Alain Pagani, Didier Stricker, and Muhammad Zeshan Afzal. 2022. BoxMask: Revisiting Bounding Box Supervision for Video Object Detection. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (2022), 2029–2039.

[21] Khurram Azeem Hashmi, Didier Stricker, and Muhammamd Zeshan Afzal. 2022. Spatio-Temporal Learnable Proposals for End-to-End Video Object Detection. *ArXiv* abs/2210.02368 (2022).

[22] Fei He, Naiyu Gao, Jian Jia, Xin Zhao, and Kaiqi Huang. 2022. QueryProp: Object Query Propagation for High-Performance Video Object Detection. *ArXiv* abs/2207.10959 (2022).

[23] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 770–778.

[24] Lu He, Qianyu Zhou, Xiangtai Li, Li Niu, Guangliang Cheng, Xiao Li, Wenxuan Liu, Yunhai Tong, Lizhuang Ma, and Liqing Zhang. 2021. End-to-End Video Object Detection with Spatial-Temporal Transformers. In *Proceedings of the 29th ACM International Conference on Multimedia*. 1507–1516.

[25] Zhengkai Jiang, Peng Gao, Chaoxu Guo, Qian Zhang, Shiming Xiang, and Chunhong Pan. 2019. Video Object Detection with Locally-Weighted Deformable Neighbors. In *AAAI Conference on Artificial Intelligence*.

[26] Ruibing Jin, Guosheng Lin, Changyun Wen, Jianliang Wang, and Fayao Liu. 2020. Feature Flow: In-network Feature Flow Estimation for Video Object Detection. *Pattern Recognit.* 122 (2020), 108323.

[27] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. 2020. *ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements.* https://doi.org/10.5281/zenodo.4154370

[28] Lijian Lin, Haosheng Chen, Honglun Zhang, Jun Liang, Yu Li, Ying Shan, and Hanzi Wang. 2020. Dual Semantic Fusion Network for Video Object Detection. *Proceedings of the 28th ACM International Conference on Multimedia* (2020).

[29] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. 2022. DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. In *International Conference on Learning Representations.* https://openreview.net/forum?id=oMI9PjOb9Jl

[30] W. Liu, Dragomir Anguelov, D. Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. 2015. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*.

[31] Hao Luo, Wenxuan Xie, Xinggang Wang, and Wenjun Zeng. 2018. Detect or Track: Towards Cost-Effective Video Object Detection/Tracking. In *AAAI Conference on Artificial Intelligence*.

[32] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. 2021. Conditional DETR for Fast Training Convergence. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[33] Yijun Qian, Lijun Yu, Wenhe Liu, Guoliang Kang, and Alexander G. Hauptmann. 2020. Adaptive Feature Aggregation for Video Object Detection. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. https://doi.org/10.1109/wacvw50321.2020.9096948

[34] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *ArXiv* abs/1804.02767 (2018).

[35] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2015), 1137–1149.

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

[37] Yuheng Shi, Naiyan Wang, and Xiaojie Guo. 2022. YOLOV: Making Still Image Object Detectors Great at Video Object Detection. *arXiv preprint arXiv:2208.09686* (2022).

[38] Mykhailo Shvets, Wei Liu, and Alexander C. Berg. 2019. Leveraging Long-Range Temporal Relationships Between Proposals for Video Object Detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 9755–9763.

[39] Mykhailo Shvets, Wei Liu, and Alexander C. Berg. 2019. Leveraging Long-Range Temporal Relationships Between Proposals for Video Object Detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 9755–9763.

[40] Mykhailo Shvets, Wei Liu, and Alexander C. Berg. 2019. Leveraging Long-Range Temporal Relationships Between Proposals for Video Object Detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 9755–9763.

[41] Guanxiong Sun, Yang Hua, Guosheng Hu, and Neil Martin Robertson. 2020. MAMBA: Multi-level Aggregation via Memory Bank for Video Object Detection. In *AAAI Conference on Artificial Intelligence*.

[42] Guanxiong Sun, Yang Hua, Guosheng Hu, and Neil Martin Robertson. 2022. Efficient One-Stage Video Object Detection by Exploiting Temporal Consistency. In *European Conference on Computer Vision*.

[43] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *ArXiv* abs/1706.03762 (2017).

[44] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696* (2022).

[45] Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13 (2004), 600–612.

[46] Haiping Wu, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. 2019. Sequence Level Semantics Aggregation for Video Object Detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 9216–9224.

[47] Hongkai Zhang, Hong Chang, Bingpeng Ma, Naiyan Wang, and Xilin Chen. 2020. Dynamic R-CNN: Towards High Quality Object Detection via Dynamic Training. *ArXiv* abs/2004.06002 (2020).

[48] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. 2022. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. arXiv:2203.03605 [cs.CV]

[49] Qianyu Zhou, Xiangtai Li, Lu He, Yibo Yang, Guangliang Cheng, Yunhai Tong, Lizhuang Ma, and Dacheng Tao. 2022. TransVOD: End-to-end Video Object Detection with Spatial-Temporal Transformers. *arXiv preprint arXiv:2201.05047* (2022).

[50] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Towards High Performance Video Object Detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017), 7210–7218.

[51] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv preprint arXiv:2010.04159* (2020).

[52] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Flow-Guided Feature Aggregation for Video Object Detection. *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 408–417.

[53] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. 2016. Deep Feature Flow for Video Recognition. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 4141–4150.

A RELATION WITH IMAGE QUALITY ASSESSMENT

Another existing work similar to our proposed ODD is called Image Quality Assessment (IQA) [5, 13, 45], which is widely used to assess image quality. However, ODD has enormous differences from IQA essentially. First, ODD pays attention to the objects in images, while IQA evaluates both the foreground and background in images, including the sky and clouds in images. Second, the influences on detection accuracy are not only about the quality of frames, other factors like object scales, and rare poses can also make detection difficult. The ODD score can comprehensively quantitatively measure all these factors (See Section 4.8). In addition, the definition of IQA is more subjective, relying on humans' judgment, while ODD is defined from the perspective of an SIOD.

B OBJECT DETECTION DIFFICULTY METRIC: THE ALGORITHM DESCRIPTION

Algorithm 1 Calculate the ODD ground truth score for an image

---

Input: PredBbox, PredLabel, GtBbox, GtLabel
Parameter: Threshold: near-positive (t_1), positive (t_2)
Output: *ODD*

1: PosList, NearList, MultiList, NegList := []
2: **for** l in **CONCAT**(PredLabel, GtLable) **do**
3:    PredBboxL, GtBboxL := **GetBboxWithLabel**(l)
4:    PredScoreL := **GetScoreWithLabel**(l)
5:    IoU_l := **CalculateIoU**(PredBboxL, GtBboxL)
6:    **for** i in [0 ... NUM(PredBboxL)] **do**
7:       Weight := **PredScoreL(i)**
8:       **if** IoU_{li} is $max(IoU_l)$ **and** $IoU_{li} \geq t_2$ **then**
9:          APPEND(PosList, Weight)
10:       **else if** $IoU_{li} \geq t_2$ **then**
11:          APPEND(MultiList, Weight)
12:       **else if** $t_1 \leq IoU_{li} \leq t_2$ **then**
13:          APPEND(NearList, Weight)
14:       **else**
15:          APPEND(NegList, Weight)
16:       **end if**
17:    **end for**
18: **end for**
19: wp, wr := **CalPR**(PosList, NearList, MultiList, NegList)
20: $ODD := 1 - 2 \cdot \frac{wp \times wr}{wp + wr + \varepsilon}$
21: **return** ODD

In Section 3.1, we formally define the concept of Object Detection Difficulty (ODD) metric with equations (1 to 4). Equation 1 is used to divide the detection results of an SIOD into four categories and compute the wp. The form of the other three equations is very similar to that of the F1-Score.

The procedure for calculating the ODD ground truth score is described in Algorithm 1. Note that the ODD score is an image-level measurement. Therefore, the algorithm will return the detection difficulty for one image. The algorithm's input is the same as calculating mAP, which includes prediction results (PredBbox, PredLabel with its confidence) and ground truth (GtBbox and GtLabel). In addition, the algorithm has two hyperparameters: near-positive threshold (t_1) and positive threshold (t_2). These two parameters are used to differentiate positive, near-positive, and multi-positive samples, which have been mentioned in Section 3.1 of our paper. After running the algorithm, we can obtain the corresponding ODD score.

C EFFECT OF ODD STRATEGY FOR GLOBAL-FRAME-BASED VOD

Table 9 shows three VOD models with different ODD strategies. Recall that ODD-A uses ODD Scheduler with ODD Threshold for faster detection speed and ODD-B/C uses ODD-based Global Reference Frame Selector (OGRFS) for better detection accuracy in both training and inference stages. Note that using ODD-B/C will not reduce the detection speed. Therefore, the ODD-full version (ODD-A+B+C) ODD-VOD can get the best speed-accuracy trade-off but need to retrain the VOD model. ODD-A+B strategy can also get a good trade-off without retraining VOD models, which provides an additional option for how ODD-VOD can be used.

Table 9: Results for Global-frame-based VOD with different ODD strategies. We use ResNet-50 as the backbone for all models.

Model	ODD Strategy		ODD Theshold						Original Results	Lossless Acc. Rate
			0.8	0.4	0.3	0.2	0.15	0.1		
SELSA [46]	A	mAP	73.5	76.3	77.2	78.1	78.2	**78.4**	78.4	32.8%
	A+B	mAP	74.2	77.4	**78.5**	79.2	79.4	79.6		72.4%
	A+B+C	mAP	74.4	77.6	**78.7**	79.5	79.7	79.8		72.4%
		FPS	33.7	22.7	20.0	18.1	16.6	15.4	11.6	-
TROIA [16]	A	mAP	73.8	77.5	78.2	79	79.4	79.6	79.8	15%
	A+B	mAP	75.4	78.3	79.0	79.7	**79.9**	**80**		40%
	A+B+C	mAP	75.5	78.8	79.7	**80.1**	**80.2**	80.4		51.7%
		FPS	24.6	13.0	10.8	9.1	8.4	7.6	6.0	-
MEGA [4]	A	mAP	72.5	75.4	76.2	76.8	**77**	**77**	77.0	70.8%
	A+B	mAP	72.9	76	**77.0**	77.8	78	78.2		112.5%
	A+B+C	mAP	73	76.1	**77.2**	77.9	78.2	78.4		112.5%
		FPS	21.9	12.4	10.2	8.7	8.2	7.7	4.8	-